ELSEVIER

CrossMark

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 52 (2015) 67 – 74

Procedia Computer Science

The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015)

Deriving public transportation timetables with large-scale cell phone data

Christopher Horn, Roman Kern

Know-Center Gmbh, 8010 Graz, Austria

Abstract

In this paper, we propose an approach to deriving public transportation timetables of a region (i.e. country) based on (i) large-scale, non-GPS cell phone data and (ii) a dataset containing geographic information of public transportation stations. The presented algorithm is designed to work with movements data, which are scarce and have a low spatial accuracy but exists in vast amounts (large-scale). Since only aggregated statistics are used, our algorithm copes well with anonymized data. Our evaluation shows that 89% of the departure times of popular train connections are correctly recalled with an allowed deviation of 5 minutes. The timetable can be used as feature for transportation mode detection to separate public from private transport when no public timetable is available.

Keywords: Cell phone data; Transportation Mode Detection; Public Transportation;Timetable;

1. Introduction

When conducting transportation mode detection for given movement trajectories, it is necessary to use features that are as expressive as possible. For example, Stenneth et al.[1] used the distance to the next bus stop location as a feature for their classifier. Another feature that may help to distinguish between public and individual transportation is based on public transportation timetables: the closer the travel time approximates the departure and arrival times of a public transportation facility, the more likely its trajectory is to originate from this facility. However, it might not be possible to obtain the official timetable of a region by automatic means. In the paper, we propose an approach to inferring the train timetable of a given region from movement-based datasets and locations of train stations.

Our dataset consists of anonymised cell phone events that are based on interactions with a cell phone providers infrastructure rather than on GPS data or sensor data directly recorded on the cell phone. Since such events are

[*] Corresponding author. Tel.: +43 316 810-30866 ;
E-mail address: chorn@know-center.at

doi:10.1016/j.procs.2015.05.026

infrequent and have a low spatial accuracy, this task is particularly challenging. The discrepancy between the observed location and the true location can be two orders of magnitude bigger than in existing GPS-based systems. Depending on the users behaviour, the frequency may vary from a single event every 6 hours to several events per minute. The cell phone data was pre-processed to avoid the identification of individual users by removing all personal information from the dataset and replacing the phone number with a hash value that was valid for 24 hours. This allowed us to establish anonymous movement trajectories for up to 24 hours. In addition, to prevent the exposure of individual users and their positions, the dataset included obfuscated and spurious dummy events.

Although our dataset and task were quite specific, the results should apply to a much bigger class of location- and movement-aware data analytics scenarios. For example, they may offer insights into the service-privacy trade-off that applies to many location-based services, some of which deliberately cloak the users privacy profile and insert dummy trajectories (e.g., k-anonymity algorithms and path confusion approaches, such as Never Walk Alone). Gruteser and Grunwald[2] proposed various algorithms that meet certain anonymity requirements by decreasing the spatial or the temporal resolutions. Our work demonstrates the usefulness of even highly obfuscated movement datasets. In our evaluation, we provide a deep analysis not only of the individual parameters of our algorithm, but also measure how much input data is required to achieve a certain accuracy.

2. Related work

Extensive research of transportation mode detection was performed using high-sampling GPS[1,3,4,5] or GPS data in connection with on-device sensors, such as an accelerometer[6]. In our work, we employ low-sampling and low-accuracy cell phone data.

Wang et al.[7] used CDRs to infer the transportation mode based on the travel time. They used a k-Nearest-Neighbor (kNN)-based clustering approach to distinguish between the different types of transportation modes (cars, public transportation, walking). In our setting, the travel times of car and train travellers partly overlapped, making a distinction based solely on the travel time unfeasible. Sohn et al.[8] applied manually collected GSM traces to distinguish between the three mobility modes (“stationary”, “walking” and “driving”). Unlike other approaches, this one relies on the signal strength and the change between two consecutive measurements rather than on the geographic coordinates of the cell towers.

In the field of public transportation modelling, Aguilera et al.[9] used cell phone data to measure passenger flows in the Paris transit system. In addition to travel times, they derived occupancy rates and origin-destination flows and established that in 80% of cases the occupancy rates estimated by GSM data corresponded to the actual ones. Calabrese et al.[10] fused cell phone data with the location data of public transportation, which allowed the authorities to better understand the movement patterns of pedestrians and buses.

However, to the best of our knowledge, no research that uses large-scale cell phone data to derive a public transportation timetable has been performed to date.

3. Methodology

In this section we describe the proposed approach to deriving the public transportation timetable of a region (i.e., a country) based solely on map data and non-GPS cell phone data. The assumption was that it was possible to use a large-scale movement dataset to identify bursts of travellers that in turn indicate public transportation movements, even if the single trajectories failed to reliably detect such movements. This assumption is visualised in Figure 1 that shows the transitions between nearby checkpoints: Figure 1 (a) the checkpoints are two motorway junctions and Figure 1 (b) the checkpoints are two railway stations. Unlike the transitions between two motorway junctions, those between two railway stations indicate multiple bursts (spikes). Assuming that those bursts are moving trains, we can derive an accurate train timetable based on a burst detection algorithm. In the remainder of this paper, we demonstrate our method using train timetables as an example.

Obtaining a train timetable of a region required the following: (i) a dataset containing geolocations for public transportation, (ii) a large-scale movement dataset, (iii) an algorithm to calculate the transitions, and (iv) a burst detection algorithm to derive the timetable.

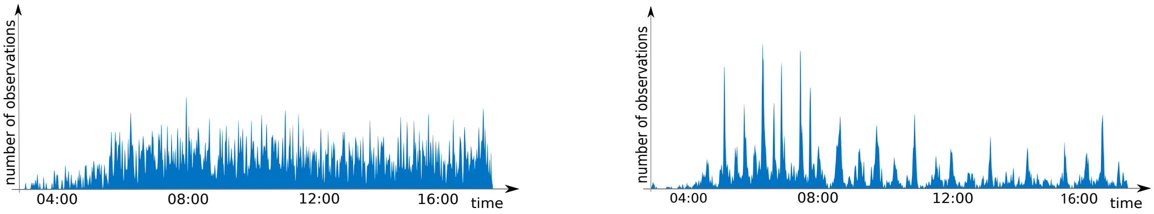


Fig. 1: Transitions between two checkpoints based on cell phone data. (a) Number of transitions between two motorway junctions (Wiener Neustadt and Wien Vösendorf), where the spikes do not follow an apparent pattern.; (b) Number of transitions between two railway stations (Wiener Neustadt and Wien Meidling), with distinctive burst, which could coincide with train movements. Data for June 9, 2014.

3.1. Geolocation dataset for public transportation

As input, our algorithm requires a list of public transportation stations or stops and their geolocation. Due to the size of some of the stations (e.g., metropolitan train stations) and the low accuracy of cell phone data, the location may not always be highly accurate. In our case, a checkpoint was a railway station with a radius of 1000 meters.

Definition 3.1. Checkpoint c: A checkpoint is a geographic entity with a given radius r, and may contain zero, one or more cell towers.

3.2. Large-scale movement dataset

A dataset that contained movement data was required to compute trajectories, which should be long enough to cover most of the public transportation connections. Although individual trajectories may have had spurious or dummy events to cloak the true movement, the aggregation of many trajectories had to be proportional to the true traffic situation. Furthermore, the accuracy and reliability of the movement data did not have to be at a level of GPS data. Since no personal information was included, the dataset was fit to protect the privacy of the users (which is a legal requirement in many countries).

Definition 3.2. Cell phone event e: An event is issued by the cell monitoring system when a certain user actively uses his/her cell phone or changes the location area. It contains the anonymous identification number, timestamp, event type (e.g., voice call, short message) and position of the cell tower (i.e., longitude and latitude).

Definition 3.3. Cell phone trajectory g: A trajectory is an ordered list E of sequential cell phone events e triggered by the same user.

3.3. Algorithm for calculating transitions

Algorithm 1 describes how to extract transitions between given checkpoints. As stated above, the movements data is an unordered list E of cell phone events e. First, these events are grouped by user and sorted by time, resulting in a list G of movement trajectories. As indicated by previous research, those trajectories may contain outliers[11], which can be removed by an appropriate outlier detection algorithm (i.e. Kalman Filter).

Next, the list of intersecting checkpoints is determined for each trajectory, and the transition between two consecutive checkpoints is recorded. To filter out waiting times of persons at the stations, the transition is only inserted when the current checkpoint is different to the previous checkpoint. This ensures that every transition consists of the last observation at the source station and the very first observation at the destination station. Unlike GPS-based trajectories, cell phone trajectories have a low sampling rate (up to hours). In this case, not every trajectory intersecting a checkpoint is necessarily observed within radius r of that checkpoint. To resolve this issue, a continuous movement for every user is simulated. The trajectories are discretised with respect to a constant time interval u (e.g. 30 seconds) and in each time unit the current position is derived via a linear interpolation between the previous and the next observation.

Processing all trajectories yields in a large set of transitions between checkpoints (e.g. railway stations). Based on the assumption that the bursts indicate a train moving between stations, train departures between two arbitrary railway stations can be calculated (Algorithm 2). The algorithm iterates over all source and destination checkpoints, identifies transitions between the pairs and extracts the departure times by applying a burst detection algorithm described in Section 3.4 below. The resulting system can derive departure times between two not necessarily neighbouring train stations.

Algorithm 1: Algorithm for calculating transitions between checkpoints

```
Data: $E$: List of cell phone events;
$C$: List of checkpoints
Result: $S$: List of transitions between checkpoints
$G \leftarrow$ create trajectories from $E$;
for $g \in G$ do
    $g \leftarrow$ remove outliers from $g$;
    $c_{prev} \leftarrow$ undefined;
    for $t = 0$ to $T$ step $\Delta t$ do
        $p \leftarrow$ interpolate position on $r$ at time $t$
        for $c \in C$ do
            if distance$(p, c) < r$ then
                if $c_{prev} \neq undefined \wedge c_{prev} \neq c$ then
                    $S \leftarrow S \cup \{(c_{prev}, c, t)\}$;
                end
                $c_{prev} \leftarrow c$;
            end
        end
    end
end
return $S$
```

Algorithm 2: Algorithm for detecting departures between checkpoints

```
Data: $S$: List of transitions;
$C$: List of checkpoints
Result: $D$: List of departure times between
        checkpoints
for $c_{source} \in C$ do
    for $c_{dest} \in C \setminus \{c_{source}\}$ do
        $S' \leftarrow \{(c_1, c_2, t) \in S : c_1 = c_{source} \wedge c_2 = c_{dest}\}$;
        $B \leftarrow$ detect bursts in $S'$;
        if $|B| > 0$ then
            $D \leftarrow D \cup \{(c_{source}, c_{dest}, B)\}$;
        end
    end
end
return $D$;
```

3.4. Burst detection algorithm

For detecting bursts in transitions, we used a burst detection algorithm with a sliding window approach. The input of the algorithm was a list of transition times S, and the output was a list of detected bursts B.

Algorithm 3 describes the burst detection. First, the average window sum is calculated. Next, the time period T is divided into windows of size z. For every window, the number of observations is counted. If it exceeds the average window sum by factor f, the highest observation value within this window is selected as burst b, and the window is shifted one unit after the detected burst. If no burst is detected, the window is moved $z + 1$ units forward.

Definition 3.4. Burst b: the highest value within the burst window. A burst window occurs when the number of observations (i.e., transitions) within that window exceeds the average number of observations (a).

Following two parameters can be configured: (i) the size (time unit) of the sliding window (z) and (ii) the factor by which the observations in a given window has to exceed the average number of observations in order to be considered a burst (f).

Definition 3.5. Burst window size z: defines the size of the sliding window which is used to partition time period T.

Definition 3.6. Burst detection factor f: defines by which factor the window sum must exceed the average window sum to constitute a burst. A lower value indicates more bursts (i.e., potential departures).

As input, the algorithm requires a complete list of transitions, which may not be available under some scenarios, e.g., real-time processing. To derive the timetable in such cases, one can either (i) calculate the average window sum based on historic data, (ii) use a fixed value or (iii) employ a more sophisticated burst detection algorithm.

Algorithm 3: Burst detection algorithm

```
Data: List of transitions $S$;
Result: List of bursts $B$
$h \leftarrow$ histogram $h[k] = |\{(c_1, c_2, t) \in S : t = k\}|$;
$a \leftarrow$ average window sum $\frac{z}{T} \sum_{i=1}^{T/z} \sum_{k=iw}^{(i+1)z-1} h[k]$
for $t = 0$ to $T$ step $z$ do
    $b \leftarrow$ window sum $\sum_{k=t}^{t+z-1} h[k]$;
    if $b > a \cdot f$ then
        $m \leftarrow$ max value from burst window $W_s$
        $B = B \cup (m)$
        $s = W_m + 1$
    end
end
return $B$;
```

4. Evaluation

In this section we describe the datasets used, the evaluation methodology and the results of our evaluation runs. For the evaluation purposes, we selected Austria, a country with a size of over 80.000km^2 and a population of 8.5 millions, as region. Austria has a cell phone penetration rate of 156% and a modern road and train network infrastructure. According to official information of an Austrian railroad company OEBB, 96.7% of the trains were on schedule in 2014[1].

4.1. Public transportation geolocation dataset

The coordinates of the railway stations in Austria were extracted using OpenStreetMap's Overpass API by issuing following query: "`area[name="Österreich"]; (node[railway="station"](area);); out`". The result contained 845 railway stations of different railroad operators and is visualised in Figure 2.

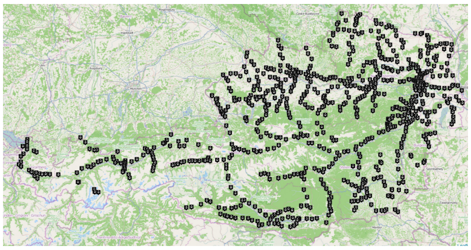


Fig. 2: Visualisation of all extracted railway stations in Austria.

4.2. Large-scale movements dataset

For this research, an Austrian mobile carrier provided access to their mobile network data. Events were issued when a user actively used his/her cell phone (e.g. made calls, sent/received text messages) or changed the location area. In addition, a ping event was issued every 6 hours. All personal information was removed from the dataset. To create user movements trajectories, a random, unique hash value (ID) whose value changed every 24 hours was

[1] http://www.oebb.at/de/Services/Puenktlichkeitsstatistik/

assigned to every user. Besides the ID, the dataset also contained information on the event type (e.g., voice call, text message), the timestamp and the location of the cell tower that issued the event.

We received three weeks' worth of cell phone data (June 6, 2014 - June 27, 2014) and identified 25.408.871 trajectories that intersected at least two train stations. Applying the above-described methodology for calculating the transitions resulted in 57.940.912 transitions between railway stations. To efficiently process this vast data amount, we embedded the algorithms in the Map/Reduce paradigm, and employed a cluster running Apache Hadoop[2].

The derived timetable was compared to the official timetable of the largest Austrian railway company, OEBB[3]. We manually selected ten representative routes, three of which are amongst the busiest commuting routes. Due to the topology in Austria, most routes and their origin- and destination stations are located next to much-travelled highways, which means that the data include both train and car travellers. However, our approach should allow us to distinguish between these travel types and derive a correct train timetable.

4.3. Overall performance

We calculated the average precision, recall and F1 value of all routes. Precision P and recall R are defined as $P(n) = \frac{D_C}{D_E}$ and $R(n) = \frac{D_C}{D_R}$, where n denotes the current route, D_C the number of correctly extracted departures, D_E the number of total extracted departures and D_R the number of real departures. The F1 score is defined as the harmonic mean of precision and recall. An extracted departure time was considered to be correct if it did not deviate more than Δt minutes from the official departure time.

In addition, we performed a detailed evaluation of selected low, medium and high traffic tracks. The results are presented in Figure 3. With an allowed deviance of 5 minutes, 49% of the extracted times are correct, and 89% of the official departure times are detected. With an allowed deviance of 10 minutes, the number of correctly extracted times increases to 70%, with 85% of the departures recalled correctly. Details of the evaluation are listed in Table 1.

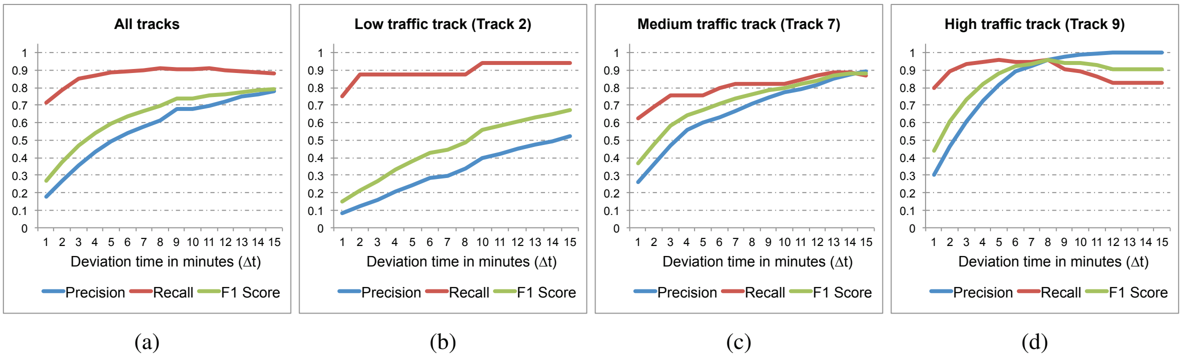


Fig. 3: Performance evaluation of the proposed approach. (a) is the average for all 10 tracks, and (b), (c) and (d) are low, medium and high traffic tracks, respectively. The x-axis denotes the allowed deviation from the official departure times in minutes (Δt). (Parameter settings: z=2, f=2)

4.4. Amount of required input data

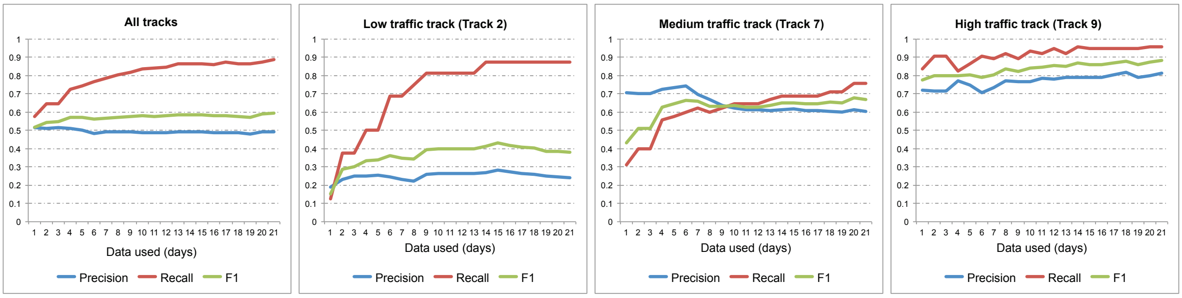
We also evaluated how the amount of used input data impacts the correctness of the derived timetable. Will it make a significant difference if only one day of data is used, compared to three weeks of data? To answer this question, we split our dataset into days and carried out the same evaluation with increasing numbers of days. Figure 4 shows the results. While the recall increases with the amount of input data, the precision does not seem to be affected. We also compared high, medium and low traffic tracks. For the high traffic track, good results were achieved, even with one-day data. For the low traffic track, the results (especially the recall) improved with the increase in the amount of

[2] http://hadoop.apache.org/
[3] http://www.oebb.at/en/

Table 1: Tracks used and results of the evaluation (Parameter settings: Δt = 5, z=2, f=2). Traffic indicates the occupancy rates of the tracks. High traffic tracks have a high number of train departures per day and a short interval between those departures. N denotes the number of observed trajectories that intersected both the source- and destination stations.

ID	From - To	Traffic	Trains	Interval (minutes)	Duration (minutes)	Distance (km)	N	P	R	F1
1	Graz - Wien Meidling	Low	20	60	164	191	320	0,25	0,95	0,40
2	Innsbruck - Bregenz	Low	16	70	268	190	336	0,24	0,88	0,38
3	Korneuburg - Wien Handelskai	High	63	30	24	17	25187	0,67	0,86	0,75
4	Kufstein - Innsbruck	High	75	30	57	78	7282	0,39	0,90	0,55
5	Liezen - Graz	Low	16	120	101	145	770	0,28	0,94	0,44
6	Linz - Wien Westbahnhof	Medium	53	30	94	180	2703	0,72	0,94	0,81
7	Salzburg - Bischofshofen	Medium	45	30	61	64	277	0,60	0,76	0,67
8	Selzthal - Linz	Low	10	120	107	112	285	0,16	0,90	0,27
9	St. Poelten - Wien	High	74	20	34	63	9727	0,81	0,96	0.88
10	Wiener Neustadt - Wien	High	97	12	53	54	6146	0,81	0,77	0,81
	Average							**0,49**	**0,89**	**0,63**

data days. However, the precision did not exceed 30%, regardless of how many days of data were used. A deeper analysis shows that since few trajectories are moving along this track, the average number of observations in the burst detection window is very low. As a result, the threshold is exceeded even when few trajectories are observed, resulting in more false positives and thus lower precision.



(a) (b) (c) (d)

Fig. 4: Effect of data used on precision, recall and F1 score. (a) is the average value for all 10 tracks, and (b), (c) and (d) are the values for one specific track. (Parameter settings: Δt = 5, z=2, f=2).

4.5. Parameter evaluation

We evaluated how changing the configuration parameters of the algorithms affected the results. To that end, we defined reasonable values for each parameter and performed an evaluation using every possible combination of the settings. The following values were selected: z = [2,5,10], f = [1..5], Δt = [1..10]. The results indicate that for the burst window size z, a value of 2 yielded the best performance (z = 2: P=0.44, R=0.90; z = 5: P=0.41, R=0.91; z = 10: P=0.40, R=0.87). Moreover, the effect of the burst detection factor f was evaluated. A lower value resulted in a higher recall and a lower precision, and a higher value decreased the recall and increased the precision (f = 1: P=0.44, R=0.90, f = 5: P=0.58, R=0.57). As mentioned above, a higher value detected fewer trains, but with a higher confidence.

5. Conclusion

In this paper, we presented an approach to deriving a public transportation timetable of a region using large-scale cell phone data and a list of public transportation stations extracted from OpenStreetMap. The challenge is that, unlike GPS data, cell phone data have a low spatial accuracy and observations do not occur frequently.

Following our approach, we could correctly recall 89% of departure times with an allowed deviance of 5 minutes. Moreover, we found that that for high traffic tracks, 80% of departures could be detected based on only one day of movement data. For low traffic tracks, however, more than one weeks worth of movement data was required to achieve a similar recall rate. To increase the relatively low precision for low traffic tracks, one may consider the dwell time on railway stations, based on the assumption that at their origin and destination railway stations train travellers have a longer dwell time than car travellers.

We believe that public transportation timetables obtained from movement data provide a valuable alternative when no automatically-derived official timetables are available. For future research, we plan to use the derived timetable to build a probabilistic model of transportation mode detection to distinguish public and private transportation based on the spatial position, departure time and travel duration. Furthermore, as a logical next step, the area below the respective spikes within the data can be used to derive an estimate of the number of passengers for each of the public transportation connections.

6. Acknowledgements

The Know-Center is funded within the Austrian COMET Program under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. L. Stenneth, O. Wolfson, P. S. Yu, B. Xu, Transportation mode detection using mobile phones and gis information, in: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS ’11, ACM, New York, NY, USA, 2011, pp. 54–63.
2. M. Gruteser, D. Grunwald, Anonymous usage of location-based services through spatial and temporal cloaking, in: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, MobiSys ’03, ACM, New York, NY, USA, 2003, pp. 31–42.
3. F. Biljecki, H. Ledoux, P. van Oosterom, Transportation mode-based segmentation and classification of movement trajectories, International Journal of Geographical Information Science 27 (2) (2013) 385–407.
4. S. Wang, Y. Chen, Z. Chen, Recognizing Transportation Mode on Mobile Phone using Probability Fusion of Extreme Learning Machines, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems 21 (2013) 13–22.
5. D. J. Patterson, L. Liao, D. Fox, H. Kautz, Inferring High-Level Behavior from Low-Level Sensors, UbiComp 2003 Ubiquitous Computing 2864 (2003) 73–89.
6. S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M. Srivastava, Using mobile phones to determine transportation modes, ACM Transactions on Sensor Networks 6 (2) (2010) 1–27.
7. H. Wang, F. Calabrese, G. Di Lorenzo, C. Ratti, Transportation mode inference from anonymized and aggregated mobile phone call detail records, 13th International IEEE Conference on Intelligent Transportation Systems (2010) 318–323.
8. T. Sohn, A. Varshavsky, A. LaMarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, E. de Lara, Mobility detection using everyday gsm traces, in: Proceedings of the 8th International Conference on Ubiquitous Computing, UbiComp’06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 212–224.
9. V. Aguiléra, S. Allio, V. Benezech, F. Combes, C. Milion, Using cell phone data to measure quality of service and passenger flows of Paris transit system, Transportation Research Part C: Emerging Technologies 43 (2014) 198–211.
10. F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, C. Ratti, Real-time urban monitoring using cell phones: A case study in Rome, IEEE Transactions on Intelligent Transportation Systems 12 (2011) 141–151.
11. C. Horn, S. Klampfl, M. Cik, T. Reiter, Detecting Outliers in Cell Phone Data, Transportation Research Record: Journal of the Transportation Research Board 2405 (2014) 49–56.