Evaluating Citywide Bus Service Reliability Using Noisy GPS Data

Shen Wang*, Brian Mac Namee*

Abstract— **An increasing number of people use smartphone applications to plan their trips. Unfortunately, for various reasons, bus trips suggested by such applications are not as reliable as other trip types (e.g. by car, on foot, or by bicycle), which can result in excessive waiting time, or even the need to revise a planned trip. Traditional punctuality-based bus service reliability metrics do not capture route deviations, which are especially frequent in rapid changing urban environments due to rapidly changing road conditions caused by traffic congestion, road maintenance, etc. The prevalence of GPS data allows buses to be tracked and route deviations to be captured. We use such data to propose and calculate a novel reliability score for bus trips. This score is a linear weighted combination of distance, time, and speed deviations from an expected, pre-defined bus trip. GPS trajectory data is large and noisy which makes it challenging to process. This paper also presents an efficient framework that can de-noise and semantically split raw GPS data by pre-defined bus trips in citywide. Finally, the paper presents a comparative case study that applies the proposed reliability score to publicly available open bus data from Rio de Janeiro in Brazil and Dublin in Ireland.**

I. INTRODUCTION

Bus trips suggested by smartphone applications are not as reliable as other transportations (e.g. by car, on foot, or by bicycle). This is partly due to frequent bus re-routing to avoid unexpected events, and out-of-date assignment of bus stops and schedules, and leads to excessive waiting time or even changes in planned trips for bus passengers.

Reliability is a key aspect of bus service quality for both passengers and operators [1]. Traditionally, the measurement of bus service reliability is focused on punctuality, as it is assumed that bus routes rarely change. However, because road conditions are changing more frequently than ever before, bus drivers and operators often need to make temporary re-routing decisions to avoid unexpected events (e.g. accidents, parades, or demonstrations). For example, in an observational study of bus drivers in the city of Changsha in China, Wang et al. [2] found that 20.2% of 7,611 observed bus stops were not made at the planned bus stops along a pre-defined route. Similarly two studies by Mizra et al. [3] and Bessa et al. [4] also show that bus services do not follow their predefined route more often than the acceptable level in Karachi, Pakistan, and Rio de Janeiro, Brazil respectively. Bus route violations directly affect the completion of a desired trip, which is more serious that delayed arrival. Chen et al. [5] proposed a new evaluation methodology considering both punctuality and route-adherence. The data used in that study, however, is collected by surveyors manually over a 3-day test period and so is not cost-effective for long-term evaluation. Furthermore their results are mainly designed for bus operators to improve their management efficiency, rather than planning reliable trips for passengers. As pointed out in [6], there is no commonly recognized measurement of bus route reliability proposed from a user experience perspective.

The prevalence of GPS data enables tracking buses at an affordable cost with wide coverage and low granularity, and has spurred significant research. For example, Fabio et al. [7] use bus GPS data to correct potential errors in bus schedules. Rudy et al. [8] applied a dimensionality reduction based trajectory similarity algorithm to infer bus line numbers given anonymized bus GPS data. Bessa et al. [4] propose systems to detect and visualize bus route violations in Rio de Janeiro using a convolutional neural network.

To the best of our knowledge none of the existing work on using publicly available GPS bus tracking data is focused on measuring bus service reliability, which is vital for bus operators and commuters. Also, most of the existing work focuses on only a small number of bus lines and predominantly uses GPS data from only one city. This is not surprising as large-scale spatio-temporal data from different cities typically has different structures, levels of granularity, and quality levels—as pointed out in [9]. This does not facilitate smart city managers learning from successful counterparts, and leads to excessive data pre-processing time (e.g. up to 79%[1] of the time of a data scientist on a typical project) that could be saved for mining more useful insights from data.

The following contributions are made in this paper to address the aforementioned problems:

- **An efficient semantic pre-processing framework for raw GPS data that enables citywide bus service analysis.** Our pre-processing framework applies a minimum bounding rectangle (MBR)-based filter to remove GPS outliers when a bus is not operating any of its pre-defined trips. We consider a set of pre-defined bus stops as a bus trip, which has much fewer points than defined route shapes. Compared with the method of calculating distance for each defined route shape point [10], ours is much quicker and keeps the trajectory points of reasonable route deviations. Moreover, to achieve citywide bus trip segmentation for all tracked buses, compared with [7] that uses the unreliable raw GPS data attributes to parse bus trips, our framework splits and matches massive raw trajectories to all possible origin-destination pairs of predefined bus trips.

*Centre for Applied Data Analytics Research, University College Dublin, Ireland. {shen.wang, brian.macnamee}@ucd.ie

[1]2016 Data Science Report: http://visit.crowdflower.com/data-science-report.html

- **A method to quantify bus service reliability for both passengers and operators.** It is commonly recognized that a reliable bus trip should stop at each of its predefined stops on time. We propose a measure of bus service reliability that, given the real GPS trajectory of a bus trip, selects the trajectory points that are geographically nearest to each pre-defined bus stop, and calculates their deviation in time, distance, and speed of selected trajectory points from what is expected. A weighted linear combination of these deviations are combined to calculate a reliability score of a bus stop for a certain travelled bus trip. This metric can effectively capture the rerouting behaviours, unpunctual arrivals, and bus stop skippings that often leads to unreliable bus trips. Passengers can query the reliability of nearby bus stops for a certain bus line to decide if they need to start a bus trip now or later. Operators can check the overall reliability for all tracked buses and study unreliable bus lines to consider if any changes need to be made.
- **An evaluation of our bus service reliability method for two cities.** We apply our proposed methodology to process raw bus GPS and GTFS data, and evaluate bus service reliability for two different cities: Rio de Janeiro (Rio) in Brazil and Dublin in Ireland. These are the only two cities that we are aware of that have relatively complete open datasets on their bus services. To the best of our knowledge, this is the first study of bus service reliability using GPS data from different cities.

II. Overview

This section provides an overview of the datasets used in our study, and the general framework used. In these descriptions the following key terminology is used:

- **Bus line**: A bus line, b, is the commonly recognized name used to identify a bus route that is provided by a bus operator. For example, the "46A" and "145" are bus lines in Dublin in Ireland. The bus line is equivalent to the *route short name* defined in the GTFS standard.
- **Bus stop**: A bus stop, s, is a predefined place where passengers can board and alight a bus. For a specific bus trip, t, a bus stop, s, has two attributes: ts indicates the expected arrival time at the bus stop on that trip, and loc indicates the geographic location of the bus stop.
- **O/D pair**: An O/D pair, OD, is a tuple that defines an origin and destination bus stop pair, $(s_1, s_{|t|})$ on a bus line. In general, a bus line has two reversible O/D pairs. For example, in Dublin, the "46A" bus line has one O/D pair from "Dun Laoghaire" to "Phoenix Park", and another from "Phoenix Park" back to "Dun Laoghaire". However, a non-negligible number of bus lines found in our datasets (in total around 50 in both cities used in this study) have more than two O/D pairs. Moreover, some circular bus lines in Rio have only one O/D pair where origin and destination are roughly in the same place. Therefore, in this paper, an O/D pair is used to describe one of the defined "directions" of a bus line.
- **Bus trip**: A bus trip, t, defines a spatio-temporal sequence connecting a pre-defined O/D pair. For a specific bus line, b, an O/D pair, OD, often has multiple pre-defined bus trips, $\boldsymbol{t}_{bOD}$, that have varying departure times, $s_1.ts$. Instead of using route shape sequences to define a bus trip, we use a sequence of predefined bus stops $\{s_1, s_2, ..., s_{|t|}\}$ to describe the "spatio-temporal sequence" for a bus trip, t. Thus, multiple bus trips can also differ in the amount and order of bus stops.

A. Datasets

The datasets used in this study are bus GPS trajectory data and General Transit Feed Specification (GTFS)[2] data which defines bus stops, routes, schedules, and fares. A public feed of the bus GPS trajectory data from Rio[3] is published as a snapshot of the latest status (i.e. time stamp, location, speed, bus line, vehicle id) of all buses running in the city, and it is updated every minute. We collected a sample of this feed from 00:00:00 to 23:59:59 on April 19, 2016. This dataset contains 6,363 buses operating 446 bus lines with 3,729,833 GPS points and its average sampling interval is 95 seconds. Public Dublin Bus GPS datasets[4] can be downloaded directly for specified time periods. We extract a dataset that covers 00:00:00 to 23:59:59 on November 06, 2012 and is composed of 1,765,912 GPS points of 835 buses running 76 bus lines. In contrast to the Rio dataset, the data from Dublin does not include vehicle speeds, but a much shorter average sampling interval of 20 seconds.

GTFS datasets for both Rio[5] and Dublin[6] follow the same standard. We extract information of bus schedules and bus stop locations from GTFS datasets for outlier removal, trip segmentation, and reliability calculation.

B. Framework

The framework we propose in this paper consists of three stages: pre-processing, trip segmentation, and reliability calculation. Pre-processing is used to remove noisy data points, estimate speeds, and transform the data into a unified data format. Trip segmentation splits a long sequence of raw bus trajectory data to match one of a set of pre-defined bus trips. Reliability calculation is used to compute the similarity between actual bus trajectory data and pre-defined trips.

To allow the framework to operate on data from multiple cities we addressed inconsistencies in both the GPS data and the GTFS data. The Rio GPS data includes the speed of each bus at every GPS trajectory point. The Dublin GPS data, however, does not include speeds. On the other hand, the Dublin GPS data has a trip direction field that the Rio data does not include. However, as the speed readings in the Rio data are noisy and the trip directions in the Dublin data were not found to be trustworthy, we estimate speed and apply

[2]http://developers.google.com/transit/gtfs/

[3]http://data.rio/dataset/gps-de-onibus

[4]https://data.dublinked.ie/dataset/dublin-bus-gps-sample-data-from-dublin-city-council-insight-project

[5]http://data.rio/dataset/onibus-gtfs

[6]https://data.dublinked.ie/dataset/dublin-bus-gtfs-data

our trip segmentation algorithm for both datasets. In relation to the GTFS data, when our trip segmentation algorithm is applied, a lot of matches in the Rio dataset in both shapes and departure time are found. A comparable match rate is achieved only in shapes in the Dublin datasets, not in the exact departure time. We solve this problem by normalizing the first stop departure time as 0, and using elapsed time instead of departure time for all subsequent stops. Finally, as the sampling rate and noise level of GPS datasets, as well as road conditions (i.e. may affect speed), are very different between the two cities in our study, we have tuned a set of unified thresholds to achieve good performance for both cities.

The three stages of the framework are described in detail in the following sections.

III. Pre-processing

To prepare data for analysis, we remove noisy data points, estimate speed, and transform data format by addressing data sparsity, GPS outliers, varying sampling rates, and duplicate data. These tasks are described in detail below.

A. Data Sparsity

In the datasets used in this paper some bus lines defined in GTFS data cannot be found in bus GPS trajectory data and some bus lines reported in GPS trajectory data do not have corresponding service information in GTFS data. In this paper, we only study bus lines with both GPS and GTFS data available. This is 66 (out of 76) bus lines in Dublin and 328 (out of 446) bus lines in Rio.

B. GPS Outliers

There are two kinds of GPS outliers observed in our datasets. The first is caused by signal interference in the GPS transceiver itself, which can lead to a large differences between GPS readings and actual locations. This happens, for example, when a bus is travelling through a tunnel or an area surrounded by high buildings or mountains. To remove such outliers, we set a speed threshold, $th.spd$, of $80m/s$ which a bus is highly unlikely to reach for two consecutive GPS trajectory points.

The second GPS outlier type arises when buses are not operating their expected trips. For instance, a bus could still record GPS data en-route to a depot for maintenance after completing a trip. The occurrence of this type of outlier is far more frequent than the first type. We propose an effective and efficient filter to remove these GPS outliers, and still retain trajectory data from valid reroutings, using a *minimum bounding rectangle* (MBR). This approach is based on the assumption that, even if re-routed, buses respect most of their pre-defined trips.

As shown in Fig. 1, to detect outliers we first retrieve all possible bus stops on a given bus line, and calculate two location points $(x_{max}, y_{max}), (x_{min}, y_{min})$, based on the maximum and minimum latitude and longitude values. To tolerate deviation from expected routes we add $th.dist = 1km$ to calculate $(x_{mbr}, y_{mbr}), (x'_{mbr}, y'_{mbr})$ as two end points of the MBR. For all raw trajectory points running a specific bus line, we exclude those are outside the MBR for the same bus line.

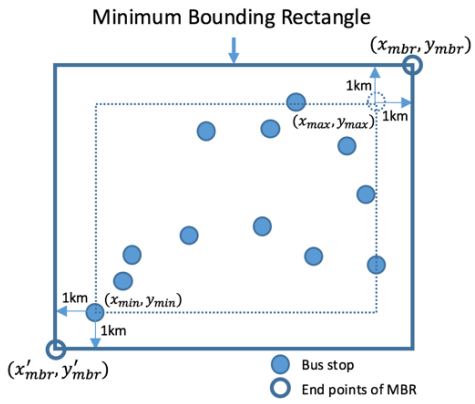


Fig. 1. Minimum bounding rectangle used to filter GPS outliers

C. Varying Sampling Rates

Varying sampling rates in GPS trajectory data is caused by missing data or power saving features of GPS devices. For each bus trip, we set a threshold time interval, $th.ts$, between consecutive trajectory points of 30 minutes. This threshold should be long enough to tolerate missing data, but not so long as to affect the evaluation accuracy. A gap between trajectory points above this threshold indicates that one trip has stopped and another new one has started. If a whole trip contains less than $th.len = 10$ trajectory points, we dump it as it is too sparse to get trustworthy results.

D. Duplicate Data

We observed many complete duplicate data lines in the Rio GPS trajectory datasets, which we removed. Sometimes data lines that have different timestamps but the same location are generated (for example when a bus is stationary). In these cases we keep the first and the last such data lines and remove those in between.

IV. Trip Segmentation

For each bus line, b, we segment the long sequence of cleaned bus trajectory data $\boldsymbol{g}'_{bv}$ generated by vehicle v into several bus trips each of which matches one of the pre-defined O/D pairs for this bus line. Our trip segmentation algorithm, shown in Fig. 2, has two objectives: (1) to find the exact subset of trajectory points that compose the trip, and (2) to match one pre-defined O/D pair to this trip. In order to minimise the complexity of our algorithm these two objectives are achieved in a single pass through the GPS trajectory data.

A bus line can have multiple pre-defined origins associated with it. The first phase of our algorithm, "finding origin", selects the most likely origin for a bus trip based on the GPS trajectory data. It does this by calculating the great circle distance to each of the possible origins associated with a bus line, b, for each point in a GPS trajectory sequence starting

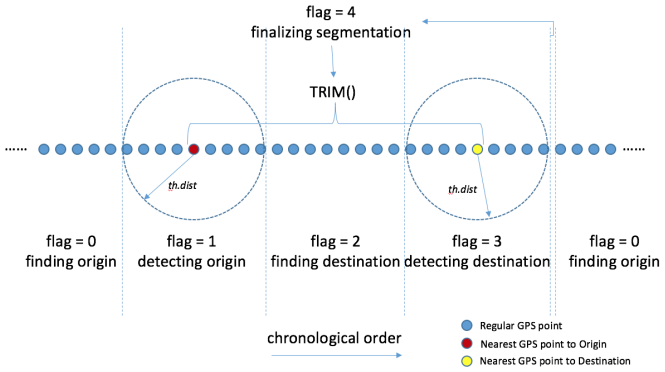


Fig. 2. Diagram of trip segmentation algorithm

at the beginning of the sequence. Once the distance from a trajectory point to an origin associated with b that is less than a specified threshold, $th.dist$, is found that origin, O, is selected as the origin for the bus trip.

The next phase, "detecting origin", records the great circle distances between the GPS trajectory points that are close to the origin, O, for later processing (i.e. trim). The distance from each GPS trajectory point to O is calculated. As long as these distances remain less than the threshold $th.dist$ the GPS points and distances are placed on the buffer t_{gps}.

Once a trajectory point that is further from O than $th.dist$ is found the algorithm moves to the next phase, "finding destination". Even though the origin has been found there may still be multiple destinations associated with a particular origin on a bus line. This phase of the algorithm finds the most likely destination, D, based on the trajectory data in the same way that the most likely origin was found. Each trajectory point process in this phase is also added to t_{gps}.

Once D has been found the algorithm moves into the "detecting destination" phase. During this phase the distance from each trajectory point to D is calculated and these points and distances are stored in t_{gps}. Once a trajectory point that is further from D than $th.dist$ is found the algorithm moves to the final phase, "finalizing segmentation". On reaching this phase the algorithm has processed all of the trajectory points that are likely to belong to a bus trip and has stored these in t_{gps}. This phase trims t_{gps} to the exact subset of trajectory points that belong to the trip. It does this by finding the GPS trajectory points in t_{gps} that have the lowest distances to O and D, respectively, and trimming the trip to the trajectory points between these two. Finally, t_{gps} is reset to empty, and the algorithm returns to the "finding origin" phase to start next trip segmentation.

V. Trip Reliability Score

We measure bus trip reliability at the level of a bus stop. A reliable bus trip should *stop* at scheduled bus stops on time. This allows a bus trip to be considered reliable even it does not strictly follow a route but visits all stops, as passengers plan their trips only on bus stops. Calculating a reliability score begins with a segmented trajectory sequence $g''_{b_i v_i OD_i} = \{g_1, g_2, ..., g_i, ...\}$, and its matched pre-defined trips $t_{b_i OD_i} = \{t_1, t_2, ..., t_i, ...\}$ running on the same bus line, b_i, and the same O/D pair, OD_i. The reliability score of a travelled bus trip trajectory $g''_{b_i v_i OD_i}$ to one of its predefined bus stops s_i is then based on a linear combination of weighted deviation in terms of distance, time, and speed of the GPS trajectory point g_{min} that is the nearest to s_i :

$$R_s(g''_{b_i v_i OD_i}, s_i) = 1.0 - \boldsymbol{d}^T \boldsymbol{w} \tag{1}$$

where $\boldsymbol{d} = [d_{dist}, d_{time}, d_{spd}]^T$ are the deviations in distance, time and speed from g_{min} to s_i; and $\boldsymbol{w} = [w_{dist}, w_{time}, w_{spd}]^T$ is a set of weights that sum to 1.

In this study, we assigned $w_{dist} = 0.6$, $w_{time} = 0.3$, $w_{spd} = 0.1$. This is to reflect the importance preference that if a bus never passes an expected bus stop, a bus trip is at its higher risk than delayed arrival (i.e. at least you can wait), or fast go-through (i.e. you can stop it by waiving hands). Ideally, between g_{min} and s_i, the most reliable bus trip should have 0 meters distance, 0 seconds arrival time delay, and 0 m/s speed deviations. The larger these deviations are, the lower the reliability score should be. To keep the final reliability score in the range of [0, 1], we normalize each type of deviation by setting thresholds on each (2000 meters, 30mins, and 40m/s respectively) and perform a linear scaling based on these thresholds (if the deviation is larger that the given threshold it is set to 1.0).

For each of pre-defined trips of bus line b_i operating direction OD_i, $t_i = \{s_1, ..., s_i, ..., s_{|t_i|}\}$, the reliability score of $g''_{b_i v_i OD_i}$ to t_i is the average of the reliability score between $g''_{b_i v_i OD_i}$ and each pre-defined bus stop $s_i \in t_i$:

$$R_t(g''_{b_i v_i OD_i}, t_i) = \frac{1}{|t_i|} \sum_{i=1}^{|t_i|} R_s(g''_{b_i v_i OD_i}, s_i) \tag{2}$$

Finally, the reliability score of $g''_{b_i v_i OD_i}$ to $t_{b_i OD_i}$ is the one with the maximum reliability score among all candidate trips $t_i \in t_{b_i OD_i}$:

$$R_{OD}(g''_{b_i v_i OD_i}, t_{b_i OD_i}) = \arg\max_{t_i} R_t(g''_{b_i v_i OD_i}, t_i) \tag{3}$$

VI. Evaluation

In this section, we summarize the comparison of existing bus GPS data processing frameworks with ours. Then, to show the quality of our proposed bus reliability measurement, we visualize the trajectories with different reliability scores and compare them with predefined trips. Finally, we demonstrate results when applying our framework to analyse bus service reliability in Dublin and Rio.

A. Processing Framework

As we have not found any detailed description to a comparable algorithm in the literature, we summarize the main features of existing processing frameworks in comparison to ours in Table I. The key improvement of our framework is that it can achieve processing of all GPS tracked buses (i.e. citywide) in two different cities by automatically dealing

with all possible O/D pairs in predefined bus trips with near-unified threshold settings.

One exception for unified threshold settings is due to the fact that the sampling rate of Dublin bus GPS data is almost 5 times faster than Rio bus GPS data. We set the distance threshold $th.dist$ to 1km for Rio data to capture enough trajectory points showing movement patterns, while the setting $th.dist$ in Dublin is 300 meters to achieve accurate trip segmentation.

TABLE I
BUS GPS DATA PROCESSING FRAMEWORKS COMPARISON

	Usapiens[10]	Fabio et al.[7]	Ours
Pre-processing	within 300 meters around each pre-defined route shape point	70 km/h speed threshold for consecutive GPS points pair	pre-defined bus stop point based MBR + 288km/h speed threshold
Initial Trip Segmentation	10800 seconds temporal threshold	900 seconds temporal threshold	1800 seconds temporal threshold
Final Trip Segmentation	Manually chosen pre-defined bus routes from GTFS	GPS data field "direction" manually by drivers	automatic trip segmentation for all possible OD pairs defined in GTFS
Bus lines	less than 10	5	66: Dublin; 328: Rio
Datasets	Rio	Dublin	Dublin & Rio

B. Reliability Score

We present 6 bus trajectories comparisons in Fig. 4 to show typical cases of high and low reliability score respectively. In (a), we show a case that even the number of full trajectory is not sufficient, it still reaches high reliability score as long as the deviation is small for all nearest trajectory points to each predefine bus stop. In (b), we show that the proposed reliability score is robust to some data loss, as long as all the existing trajectory points can match a certain amount of pre-defined bus stops. In (c), we show that the increase in reliability score when sufficient full trajectory points are present compared to the same bus line shown in (b). In (d), we show a low reliability score case in which the predefined bus trip information might be wrong, as an excessive deviation can be found in geo-distance, arrival time delay, and speed. In (e), we show a second low reliability score case due to arrival time delay mainly which might be caused by a detour in the bottom right corner. In (f), we show a third low reliability score case mainly due to reroute, even when low delay and speed achieved for all matched bus stops.

C. Reliability Evaluation in Rio and Dublin

The results of reliability score for all bus trips in Dublin and Rio are summarized in Table II. The reliability score of Dublin is higher than the one of Rio in terms of minimum, mean, 25th percentile, 50th percentile, and 75th percentile, ranges from 4% to 18%. Moreover, a t-test shows a significant difference between the reliability scores for the two cities at the 95% confidence level. Thus a conclusion can be drawn that the bus trip in Dublin is more reliable (i.e. adhere to pre-defined bus schedules) that the one in Rio.

TABLE II
CITYWIDE BUS RELIABILITY SCORE COMPARISON

City	trips	mean	std	min	25%	50%	75%	max
Rio	21503	0.833	0.125	0.193	0.753	0.865	0.938	0.992
Dublin	4036	0.897	0.122	0.227	0.865	0.964	0.975	0.991

As shown in Fig. 3, we investigate the relationship between bus trip length and reliability score. We got Pearson correction coefficient 0.02 and -0.36 in Dublin and Rio, respectively. This indicates that the reliability score is stable for all bus trip length in Dublin, while in Rio, the reliability drops modestly when the trip length increases, which is in line with the findings in Beijing [5]. We also show the

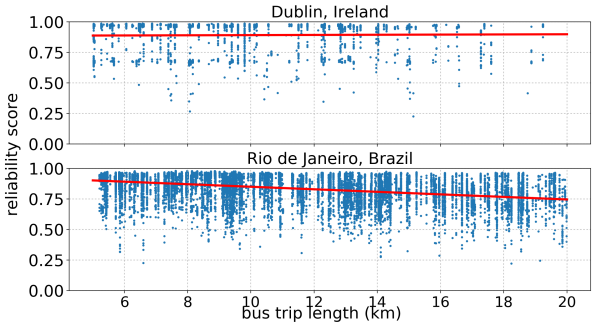


Fig. 3. Reliability score v.s. bus trip length

changes of reliability score with different parts of a day in Fig. 5. The general suggestion for planning a journey is to avoid the peak hour of the day. Specifically, morning peak hours in Dublin, and evening peak hours in Rio. In particular, night buses in Rio (i.e. Dublin does not operate) should be avoided as it shows the lowest reliability score.

Analysed reliability score results are also organised by each bus stop for passengers to plan their trips. For example, if passengers live near the Corduff bus stop shown in Fig. 6 (a), who want to take bus line 38 from IBM Campus to City Centre, our framework provides reliability results of this bus stop for all departure times of a day, as shown in Fig. 6 (b). Therefore, they probably need to avoid or pay less trust on buses leaving around 5pm, as the real bus schedule may be subject to the traffic when people finished their work.

VII. CONCLUSIONS AND FUTURE WORK

This paper proposed a set of methodologies to efficiently de-noise and process raw GPS data of all tracked buses and their corresponding GTFS data in different cities. A citywide comparative case study on bus service reliability of Dublin and Rio is conducted by applying the proposed methodologies. We find that the defined reliability score captures the key factors (i.e. re-routings, delayed arrivals, and missed bus stops) that lead to unreliable bus trips.

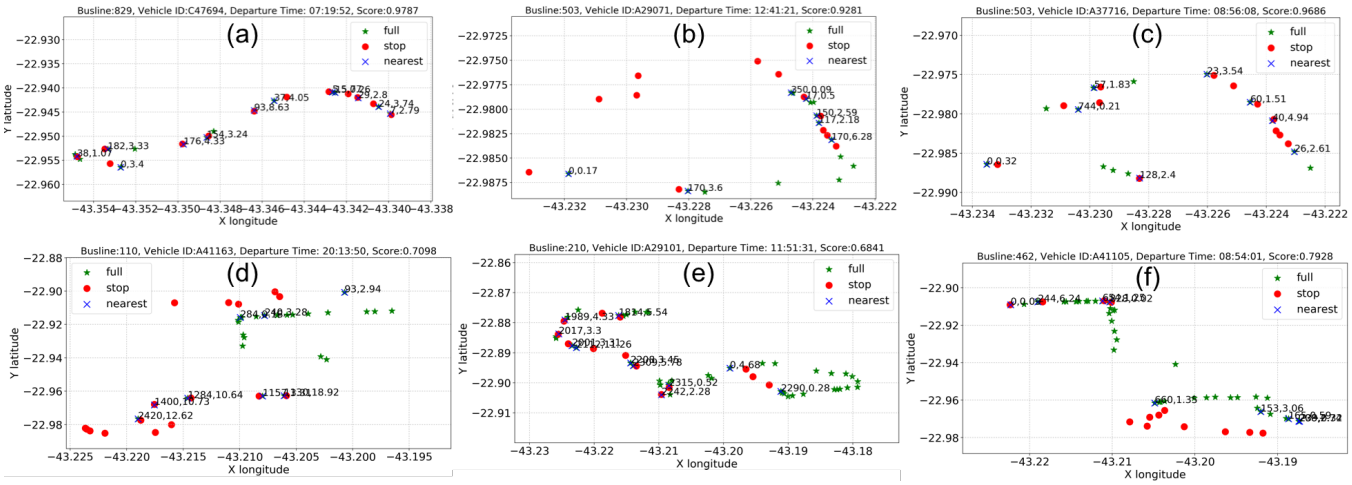


Fig. 4. Bus trajectory comparison in Rio. ("star" represents full bus trajectory points; "circle" represents predefined bus stops; "cross" represents the nearest bus trajecotry points (subset of "star") to each of its predefined bus stops, it also displays "time_deviation(seconds), speed(m/s)")

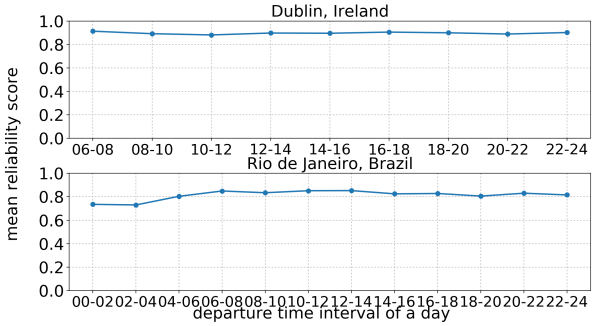


Fig. 5. Mean reliability score v.s. departure time of a day

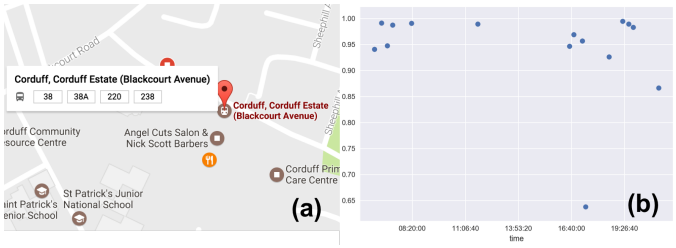


Fig. 6. (a) Corduff bus stop in Dublin; (b) reliability score v.s. departure time at Corduff bus stop serving No. 38 bus line from IBM Campus to City centre

For the future work, we plan to introduce bus headway into reliability measurement. Besides, a more intelligent method is required to infer whether a low reliability score is caused by incorrect GTFS data, or only happened in a single trip.

Acknowledgements

The authors would like to acknowledge the support of Enterprise Ireland and IDA Ireland for funding this research under the Technology Centres Programme (TC 20130013).

A part of this research was supported by the IBM Research Intern Program. We thank Takashi Imamichi of IBM Research - Tokyo for discussions.

References

[1] Chapman, "Bus reliability: Definition and measurement," *Research Report 18. Transport Operations Research Group*, 1976.

[2] Q. Wang, W. Zhang, R. Yang, Y. Huang, L. Zhang, P. Ning, X. Cheng, D. C. Schwebel, G. Hu, and H. Yao, "Common traffic violations of bus drivers in urban china: An observational study," *PLOS ONE*, vol. 10, no. 9, 2015.

[3] S. Mirza, M. Mirza, H. Chotani, and S. Luby, "Risky behavior of bus commuters and bus drivers in karachi, pakistan." *Accident Analysis & Prevention*, vol. 31, no. 4, pp. 329–333, 1999.

[4] A. Bessa, F. d. M. Silva, R. F. Nogueira, E. Bertini, and J. Freire, "Riobusdata: Outlier detection in bus routes of rio de janeiro," *arXiv preprint arXiv:1601.06128*, 2016.

[5] X. Chen, L. Yu, Y. Zhang, and J. Guo, "Analyzing urban bus service reliability at the stop, route, and network levels," *Transportation Research Part A: Policy and Practice*, vol. 43, no. 8, pp. 722–734, 2009.

[6] A. Gittens and A. Shalaby, "Evaluation of bus reliability measures and development of a new composite indicator," in *94th Transportation Research Board Annual Meeting, Washington D.C.*, 2015.

[7] F. Pinelli, F. Calabrese, and E. Bouillet, "A methodology for denoising and generating bus infrastructure data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 1042–1047, 2015.

[8] R. Raymond and T. Imamichi, "Bus trajectory identification by map-matching," in *Pattern Recognition (ICPR), 2016 IEEE International Conference on*. IEEE, 2016.

[9] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 1905–1914.

[10] M. R. Vieira, L. Barbosa, M. Kormksson, and B. Zadrozny, "Usapiens: A system for urban trajectory data analytics," in *2015 16th IEEE International Conference on Mobile Data Management*, vol. 1, 2015, pp. 255–262.