Imperial College London

Department of Computing

MEng Individual Project Interim Report

Active Delay Warning Transport Application

Author:
Yangfan Zhang

Supervisor:
Dr. Peter McBrien

May 21, 2015

Abstract

abstract

¡Abstract here¿ The abstract is a very brief summary of the report's contents. It should be about half a page long. Somebody unfamiliar with your project should have a good idea of what it's about having read the abstract alone and will know whether it will be of interest to them. Note that the abstract is a summary of the entire project including its conclusions. A common mistake is to provide only introductory elements in the abstract without saying what has been achieved.

Acknowledgments

¡Acknowledgements here¿ Peter McBrien, Susan

¡Name here¿

¡Month and Year here¿
National Institute of Technology Calicut

Contents

Chapter 1

Introduction

The London bus network carries 2.4 billion passengers a year, more than the rest of England combined [1].

The bus arrival times published by Transport for London (TfL) are currently widely available on digital live bus arrivals signs at more than 2,500 bus stops [2]. Passengers can also check this information by sending a text message with the bus stop code, as well as doing a quick search online or on mobile applications.

Passengers chose travel mode based on travel time and convenience. Find evidence for this assumption. How do people choose travel mode / whether to take bus?

Scenario, a passenger chose to take a bus, then only realises delay at the end of the journey. Would have chosen another mode of travel should he know of the delay earlier.

Passengers rely on the bus arrival times to plan their journey, by factoring in the waiting time when choosing the buses to take. Current London journey planning software takes the journey start time, start location, and destination as input, and recommends routes consisting of a variety of travel mode, with an estimate travel time for each suggested journey. Such popular planners include Google maps [3], Citymapper [4] and TfL Journey Planner [5].

However, the accuracy of the bus arrival times published is affected by many external factors. For example, when there is heavy traffic, the buses are likely to be delayed by a difference significant enough for a change in passengers' route picking. Yet, this delay information is not reflected in the arrival time data or the estimated journey time early enough for the passengers to make a decision to choose an alternate route. As a result, passengers waste time waiting for buses that come much later than expected, or chosing to board a bus that takes far longer than the estimated journey

time. Although the average bus delay is 1 minute, there was a 16.6 chance of waiting for more than 10 minutes [11].

This can be avoided if passengers are informed of the delays in bus arrival times in advance. Such delays can predicted by analysing the historical delays. This is achieved by collecting data from the TfL live bus arrivals Application Program Interface (API) stream feed[2], and estimate the average journey time required between two locations on a given time of the day. Next, a bus arrival time table with delays during various time windows over a week can be crafted and fine tuned incrementally. We compared the travel times in this timetable with the official travel times published by the TfL Open Data to find out the delays in bus arrival times.

Chapter 2

Background

2.1 London Bus Network

The bus network in London is one of the largest and most accessible in the world. It is carrying a staggering number of passengers, with more than 2.4 billion journeys in 2013/14, which was more than any year since 1959 [1].

On an average day between 2005 and 2010, about 14% of the trips made by London residents were by bus [7]. They spent on average 14 minutes per day on these bus trips.

There are currently 19,345 bus stops, and 680 routes served by 8,765 buses daily in London[8].

produce graph for this, number of stops, routes and buses

2.1.1 Bus Network Performance

TfL published the following figures in the second quarter 2014/2015 buses performance data [6].

For the high frequency services, the average scheduled wait was 4.86 minutes, the average excess wait was 0.94 minutes, and the average actual wait was 5.80 minutes. While passengers could expect the buses to come within 10 minutes 83.4% of the time, there was 15.1% chance of waiting for 10-20 minutes, 1.3% chance of waiting for 20-30 minutes, and 0.2% chance of waiting for more than 30 minutes.

For the low frequency services, 87% of the buses services were on time, and 11.4% were 5-15 minutes late.

For the night buses, 84.5% of the services were on time. The average excess wait was 0.68 minutes.

produce graph here

The bus arrivals might be affected by traffic congestion, staff availability, and engineering problems or mechanical breakdown [11].

2.2 Transport for London Open Data

why does TfL release open data? Find evidence

TfL provides free, open data consisting of around 30 feeds and API [9].

There are 3 forms of data, static data files which rarely change, feeds that refreshed at regular intervals, and API that enable a query to receive a bespoke response, depending on the parameters supplied.

Over 5,000 developers have registered for the open data[9], and around 200 travel apps are powered by it [1].

2.2.1 Live Bus Arrivals API Stream

The Live Bus Arrivals API Stream provides the predicted time until a bus is expected to arrive at a stop. These predictions are available for the next 30 minutes at any point in time. For example, at 9am, the stream will provide predicted bus arrivals up to 9.30am on the same day. This data is refreshed every 30 seconds [10].

The base URL is `http://countdown.api.tfl.gov.uk/interfaces/ura/stream_V1`. In order to collect bus arrival data for analysis, we supplied the following parameters which specify the fields returned by the API.

2.2.2 Bus Stop Locations and Routes

The TfL Open Data provides network information on the location of all bus stops in London, and the sequence of bus stops in every bus route.

This data is in the Comma Separated Values (CSV) format.

2.2.3 Journey Planner Bus Timetables

In order to calculate the delays in bus arrival times, we need the official bus travel times between stops for reference. This data was extracted from the Journey Planner Bus Timetables[17] as part of the TfL Open Data. The timetables contians information on bus schedules including stops, routes, departures times, departure frequencies, operational notes, as well as the days on which the services run.

The timetables uses the Extensible Markup Language (XML) [13] format, with the schema defined in TransXChange [12], the UK nationwide standard

for exchanging bus schedules and related data. We used the General schema version 2.1[14][15] for this project.

Data Structure

The TransXChange model has seven basic concepts[16]:

insert definitions here

1. *Service*,
2. *Registration*,
3. *Operator*,
4. *Route* specifies an ordered list of *StopPoints*.
5. *StopPoint* contains reusable declarations of the stops used by the routes and journey patterns of the schedule. All StopPointRef instances elsewhere in a document are resolved against the contents of the StopPoints element.
6. *JourneyPattern* specifies an ordered list of links between the *StopPoints*, giving relative times between each stop.
7. *VehicleJourney* specifies the list of stops at specific absolute passing times.

iBus System maybe include this part?

2.2.4 iBus System

The actual arrival time of buses on a route at any given bus stop for the selected day are made available by the London Buses iBus system.

Currently, the routes were selected as the first of the New Routemaster (New Bus for London). Data will be published once a week [11].

This data is stored in the CSV format. There are two potential uses for this data.

- We can integrate it into the arrivals table to improve the precision of the arrivals data. Since the current entries in the arrivals table contain estimated bus arrival times, the integrated of the iBus data which contains real bus arrival times will likely boost the precision of the data in arrival table.

- We can compare the predicted delays with the iBus data for performance evaluation.

2.2.5 Summary on Available Data

There is no data to directly give predictions on delay.

2.3 Current Apps that use TfL Open Data

2.3.1 TfL Journey Planner

- name of app
- company/app logo
- year released
- version
- platform
- main features

2.3.2 Google Maps

2.3.3 Citymapper

2.3.4 Summary on current Apps

No app that gives predictions and delay warnings on travel time

2.4 Literature review on similar research done in other areas of the world

2.5 Our Approach / Suggestion

We try to find a solution for theses two problems by providing a service of bus travel time predictions, and a demo application to show case the use of such a data service.

Chapter 3

Concept Design

3.1 Overview

TfL releases average travel time between stops. However, this data is too general and sometimes not accurate.

The general idea to predict delays is that the average travel time between 2 stops depends mainly on the day of the week and the hour.

3.2 Historical Timetable

We collect historical bus arrival tiems, and culculate the average travel time between 2 neighbouring bus stops by all routes, on given day and hour

3.3 Current Timetable

We collect the past 1 hour of bus arrival times, and calculate the average travel time between neighbouring bus stops.

3.4 Reference Timtable

Get the official average travel time from journey planner timetable

3.5 Assumption

The historical timetable should converge to reference timetable.

When the current timetable is very different from reference timetable, there's delay.

Chapter 4

Data Generation

4.1 Overview

UML diagrams

4.2 Development Environment

4.2.1 Virtual Machine

4.2.2 Databases

MySQL

4.3 Generating Historical Timetable

- data collection
- data storing
- data processing
- DB optimisation
- python scripts and SQLs
- resulting table name and schema

Column Name	Type	Default
id(Primary)	int(11)	Auto Increment
stop_code_lbsl	varchar(64)	
route	varchar(64)	
vehicle_id	varchar(64)	
trip_id	varchar(64)	
arrival_date	date	
arrival_time	timestamp	NULL
expire_time	timestamp	NULL
recorded_time	timestamp	Current Timestamp

Table 4.1: delay_arrivals Table Schema

Parameters supplied

- *StopID* This is the alphanumeric identifier of a bus stop. It is also known as stop_code_lbsl.
- *LineName* This is the route number that is displayed on the front of the bus on any publicity advertising the route.
- *VehicleID* The unique identifier of the vehicle.
- *TripID* The identifer of the specific trip that the prediction is for.
- *EstimatedTime* This is the predicted time of arrival for the vehicle at a specific stop.
- *ExpireTime* This is the time at which the corresponding prediction is no longer valid and should stop being displayed.

The resulting URL is `http://countdown.api.tfl.gov.uk/interfaces/ura/stream_V1?ReturnList=StopID,LineName,VehicleID,TripID,EstimatedTime,ExpireTime`.

Each data entry contains an estimated arrival time for each bus journey at a given bus stop. This estimated arrival time is stored in the database via an UPDATE statement, which ensures that only the latest estimated arrival times per journey per bus stop are stored. To avoid having data being overwritten, only the entries that have been changed in the recent 10 minutes can be updated. This has been achieved through running a daemon process written in python on a virtual host. Table 4.1 shows the schema for arrivals table.

Column Name	Type	Comments
id(Primary Key)	int(11)	Auto Increment
route	varchar(64)	The route name
run	int(11)	The route direction
sequence	int(11)	The sequence of the bus stop in the route
stop_code_lbsl	varchar(64)	The internal bus stop identifier
bus_stop_code	varcher(64)	The public code for the bus stop
naptan_atco	varchar(64)	The national identifier of the bus stop
stop_name	varchar(64)	The name of the bus stop

Table 4.2: delay_bus_sequences Table Schema

Column Name	Type	Comments
id(Primary Key)	int(11)	Auto Increment
route	varchar(64)	The bus route
start_stop	varchar(64)	The stop_code_lbsl for the start stop
end_stop	varcher(64)	The stop_code_lbsl for the end stop

Table 4.3: delay_neighbours Table Schema

Assumption

We assume that the actual bus arrival time is the midpoint between the last estimated arrival time, and the system time when the clear signal ($ExpireTime = 0$) is received.

4.4 bus locations

We imported the bus sequences into the delay_bus_sequences table (Table 4.2)

Question: Can I skip some columns of the table schema, as those columns have not been used in the project, but just storing as reference for now?

Additionally, we extracted information on all pairs of neighbouring bus stops and the routes that serve between them. We save this information in the delay_neighbours table (Table 4.3). See sample data in Table 4.4.

id	route	start_stop	end_stop
18433	30	10002	11469
44878	N19	10002	11469
47128	N41	10002	29772
8653	19	10002	11469

Table 4.4: Sample data in delay_neighbours Table

start_stop	end_stop	route	trip_id	day	start_time	end_time	length_in_secs
10002	11469	19	346930	Monday	1/12/2015 12:34:16	1/12/2015 12:35:28	72
10002	11469	4	547334	Monday	1/12/2015 12:39:55	1/12/2015 12:41:14	79
10002	11469	19	346961	Monday	1/12/2015 12:40:04	1/12/2015 12:41:53	109
10002	11469	30	226765	Monday	1/12/2015 12:45:07	1/12/2015 12:47:11	124
10002	11469	4	547676	Monday	1/12/2015 12:47:17	1/12/2015 12:49:12	115
10002	11469	19	346976	Monday	1/12/2015 12:48:21	1/12/2015 12:50:09	108
10002	11469	30	276872	Monday	1/12/2015 12:54:05	1/12/2015 12:55:08	63

Figure 4.1: List of journey time from stop 10002 to stop 11469

Finding the average travel time between neighbouring stops

update this part

To experiment with the queries, we selected one pair of the neighbouring stops (10002, 11469), and listed the time required to travel from stop 10002 to stop 11469 by finding the difference in arrival times for each journey. Sample entries of this list is shown in Figure 4.1.

We then calculated the average journey time required to travel from 10002 to 11469 for each hour in each week of the day. This information is stored as a timetable, which would be used for further analysis.

Figure4.2 shows the timetable generated. Each cell indicates the average journey time required to travel from stop 10002 to stop 11469 at a give hour of a give week of day. The **NULL** values are due to a current databases performance issue. This will be resolved later.

We plan to construct a timetable this way for each pair of the neighbouring bus stop.

4.4.1 Generating Reference Timetable

Journey Planner Bus Timetables

Need more detailed explanation. Walk through an example with xml code

Each xml file contains bus schedule information for a route. For each day of the week, there are predefined number of VehicleJourneys running the give

hour	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
0	NULL	NULL	75.1	56	63.5	62.5	NULL
1	NULL	NULL	68.4	60.6	69.6	54	NULL
2	NULL	NULL	50.5	32.5	58.5	86.5	NULL
3	NULL	NULL	NULL	64	69	51.5	NULL
4	NULL	NULL	NULL	88.5	62	46	NULL
5	NULL	NULL	NULL	61	73	59.3	NULL
6	NULL	NULL	NULL	69	79.4	75.1	NULL
7	NULL	NULL	NULL	88.1	81.7	84.3	NULL
8	NULL	NULL	NULL	90.2	118.8	116.7	NULL
9	NULL	NULL	NULL	82	90.9	101.7	NULL
10	NULL	NULL	NULL	74.4	90.8	145.6	NULL
11	NULL	NULL	NULL	111.2	97.9	124.2	NULL
12	NULL	95.5	267.3	169.9	146.5	114.7	NULL
13	NULL	125.4	245.3	168.2	104.7	NULL	NULL
14	NULL	90.4	NULL	123.8	85.7	NULL	NULL
15	NULL	120.7	113	153.1	87.2	NULL	NULL
16	NULL	168.1	125.1	134.1	100.5	NULL	NULL
17	NULL	113	110.4	124.6	126.5	NULL	NULL
18	NULL	127.1	110.9	179.8	148.6	NULL	NULL
19	NULL	115.5	149.3	243.7	160	NULL	NULL
20	NULL	87.7	96	140.5	85.2	NULL	NULL
21	NULL	80.1	94	94.5	85	NULL	NULL
22	NULL	75	93.3	80.4	100.7	NULL	NULL
23	NULL	54.6	75.6	78	73.7	NULL	NULL

Figure 4.2: Average journey time in seconds from stop 10002 to stop 11469 for each hour of each day of week

route throughout the day. We used the VehicleJourneys as a starting point to retrieve the departure time of the actual vehicle from the terminal. We then retrieve the corresponding JourneyPattern, and obtained the travel time between each neighbouring stops on the route for the given vehicle journey, to compute the cumulative travel times throughout the route.

The above computation was performed on each xml file to generate the actual arrival time and travel time for each vehicle trip at each stop in the route throughout the day. The results of this computation was stored in the delay_tfl_timetable table(Table 4.5).

Column Name	Type	Comments
id(Primary Key)	int(11)	Auto Increment
route	varchar(64)	The bus route
day	varchar(32)	The day of week for the vehicle journey
run	int(11)	The route direction
sequence	int (11)	The sequence of the bus stop in the route
stop_name	varchar(64)	The name of the bus stop
naptan_atco	varchar(64)	The national identifier of the bus stop
arrival_time	datetime(6)	The expected arrival time for the given vehicle at the current bus stop
travel_time	int(11)	The travel time in seconds from the previous stop in the route to the current stop
cumulative_travel_time	int(11)	The travel time in seconds from the terminal to the current stop
departure_time_from_origin	datetime(6)	The departure time of the given vehicle from the terminal

Table 4.5: delay_tfl_timetable Table Schema

Chapter 5

Serving the Data as REST API

5.1 Overview

Objective of setting this up - to enable easy and fast query of the predictions I generated

5.2 Frameworks

5.2.1 Django Framework

- what is it
- why did I choose it
- what are the available options and why do I choose the current one

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design[18].

5.2.2 Django REST Framework

5.3 API Endpoints

- address of the endpoint
- parameters supplied
- return result format

5.3.1 Historical & Current

5.3.2 Reference

5.3.3 Arrival

5.4 Summary of data service

Met the first objective by supplying data on bus travel time predictions

Chapter 6

Mobile Application for Active Delay Warning

6.1 Motivation

Demo App to use the data for effective delay warning
Write from the user point of view
to demonstrate what the predictions can be used for non-technical users
easy to use interface
easy to understand data presentation
personalised warning feature

6.1.1 AngularJS

6.2 Contribution

save users time make informed decisions when choosing travel mode

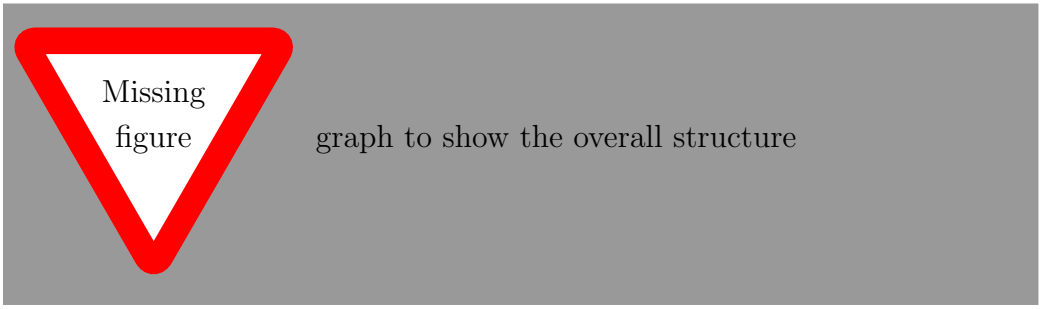
Chapter 7

Technical Details

Missing Testing section

7.1 Overview

- Used Django Framework to build the backend that retrieves data from a MySQL database.
- Used AngularJS with Twitter Bootstrap for frontend development



graph to show the overall structure

7.2 Daemen Script Management

7.3 Backend Deployment

7.4 Frontend Deployment

7.5 Software Project Management

7.5.1 Jenkins

7.5.2 Trello

Chapter 8

Evaluation

8.1 Compare predictions to real data

We can compare the predictions generated to the live bus arrivals data stored in the arrivals table. We can calculate the standard deviation of the difference in the predicted time and actual arrival time. This value will be the direct indicator of the accuracy of the predictions.

8.2 User feedback

We can get students that take buses regularly to use the mobile application to test the accuracy of the delay predictions. The user feedback gathered will be the second input for evaluation.

8.3 Accuracy

How accurate are the predictions?

8.4 Performance

How reliable is the service?

8.5 User feedback on UI

How effective is it at warning delay? ¿¿¿¿¿¿¿ Stashed changes

Chapter 9

Future Work

¡Future work here¿

Chapter 10

Conclusion

¡Conclusion here¿

Acronyms

API Application Program Interface. 2, 4

CSV Comma Separated Values. 4

TfL Transport for London. 1, 2, 4

XML Extensible Markup Language. 4

Glossary

TransXChange the UK nationwide standard for exchanging bus schedules and related data. 4, 5

References

[1] Transport for London Annual Report and Statement of Accounts 2013/14, http://tfl.gov.uk/cdn/static/cms/documents/annual-report-2013-14.pdf [visited on 27/01/2015]

[2] Transport for London Live Bus Arrivals, http://www.tfl.gov.uk/modes/buses/live-bus-arrivals [visited on 27/01/2015]

[3] Google Maps, https://www.google.co.uk/maps [visited on 30/01/2015]

[4] CityMapper, https://citymapper.com/london [visited on 30/01/2015]

[5] TfL Plan A Journey, http://tfl.gov.uk/plan-a-journey/ [visited on 30/01/2015]

[6] Transport for London Buses Network Performance Second Quarter 2014/15, http://tfl.gov.uk/cdn/static/cms/documents/network-performance-latest-quarter.pdf [visited on 27/01/2015]

[7] Travel in London, Supplementary Report: London Travel Demand Survey (LTDS), http://www.tfl.gov.uk/cdn/static/cms/documents/london-travel-demand-survey.pdf [visited on 28/01/2015]

[8] Transport for London Bus Stops Locations and Routes Open Data, https://www.tfl.gov.uk/info-for/open-data-users/our-feeds [visited on 28/01/2015]

[9] Transport for London Open Data, https://www.tfl.gov.uk/info-for/open-data-users/our-open-data [visited on 28/01/2015]

[10] Transport for London Live Bus & River Bus Arrivals API Interface Documentation, https://www.tfl.gov.uk/cdn/static/cms/documents/tfl-live-bus-river-bus-arrivals-api-documentation-v16.pdf [visited on 28/01/2015]

[11] Transport for London Buses Performance Data, https://www.tfl.gov.uk/corporate/publications-and-reports/buses-performance-data#on-this-page-5 [visited on 28/01/2015]

[12] TransXChange, https://www.gov.uk/government/collections/transxchange [visited on 13/05/2015]

[13] XML Schema Language, http://www.w3.org/TR/xmlschema-2/ [visited on 13/05/2015]

[14] TransXChange Downloads & Schema, https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/391783/Downloads_Schema-2015-01-05_-_Updated.pdf [visited on 13/05/2015]

[15] TransXChange Schema 2.1 xsd, http://www.transxchange.org.uk/schema/2.1/TransXChange_general.xsd [visited on 13/05/2015]

[16] TransXChange Schema Guide 2.1 & 2.2a, http://81.17.70.199/transxchange/schema/2.1/guide/TransXChangeSchemaGuide-2.1-v-45.pdf [visited on 13/05/2015]

[17] Transport for London Open Data Feeds, https://www.tfl.gov.uk/info-for/open-data-users/our-open-data [visited on 13/05/2015]

[18] Django Framework, https://www.djangoproject.com/ [visited on 13/05/2015]