Los Angeles Metro Bus Data Analysis Using GPS Trajectory and Schedule Data

Kien Nguyen, Jingyun Yang, Yijun Lin, Jianfa Lin, Yao-Yi Chiang, and Cyrus Shahabi

Integrated Media Systems Center, University of Southern California

Contents

Abstract

With the widespread installation of location-enabled devices on public transportation, a large amount of bus trajectory data is being generated. Using the location data of the Los Angeles Metro bus system, along with publicly available bus schedule data, a set of data processing and analyses were performed to measure the performance of the public transportation system in Los Angeles by a number of metrics such as travel-time reliability, on-time performance, bus bunching, and travel-time estimation. The data analysis results are demonstrated in a web-based application via web services. The developed algorithms and system provide powerful tools to detect issues and improve the efficiency of public transportation systems.

Acronyms

ADMS Archived Traffic Data Management System. 6

Caltrans the California Department of Transportation. 5

GPS Global Positioning System. 2, 5–15, 17–21, 24, 30

GTFS General Transit Feed Specification. 9, 30

ID Identifier. 7–10, 13

LA Los Angeles. 5, 6, 8, 9, 21, 30

METRANS METRANS Transportation Center. 6

RIITS The Regional Integration of Intelligent Transportation Systems. 6, 8

URL Uniform Resource Locator. 29

US United States. 5

1 Introduction

Inefficient traffic conditions have been a major issue for many big cities. The issue has been a significant cost of time and money for citizens and visitors. According to the Transportation Statistics Annual Report from the Bureau of Transportation Statistics in 2016 [3], the average annual delay per commuter rose from 37 hours in 2000 to 42 hours in 2014, a 13.5 percent increase. Traffic conditions in Los Angeles (LA) are even worse. According to the TomTom Traffic Index, LA is ranked the most congested city in the United States (US), the 12th most congested city worldwide, with a typical half-hour commute taking 81 percent longer during evening peak periods and 60 percent longer during the morning peak.

Fortunately, with the widespread establishment of public transportation in LA, traffic congestion is expected to be alleviated. Also, in recent years, a vast number of location-enabled sensors (e.g., GPS) have been installed on buses running in LA, from which public transportation performance can be calculated and analyzed. Improving the performance and reliability of public transportation vehicles has been one of the primary objectives for the California Department of Transportation (Caltrans). The wealth of data collected from bus GPS trajectories can offer an unprecedented opportunity for analysis of public transportation systems towards reducing operating costs and increasing ridership. At our research center, USC Data Science Institute, we have been collecting both bus sensor data for the past six years and transit schedule data publicly available from LA Metro. These datasets enable us to perform various kinds of analyses on the performance of LA metro buses.

The primary objective of this project is to develop a system that can process massive amounts of GPS trajectories from public transportation vehicles and implement statistical algorithms to analyze a variety of public transportation-system performance metrics such as travel-time reliability, on-time performance, bus bunching, and travel-time estimation. This project includes fundamental research in the mining and correlation of real-time and historical bus GPS trajectory datasets in LA County that have been collected and archived in our database over the past six years. Furthermore, to demonstrate the benefits of our research, we have developed a proof-of-concept web-based application enabling the access and visualization of the performance metrics of public transportation vehicles. This research exploits the real-world LA traffic sensor and bus GPS datasets collected from The Regional Integration of Intelligent Transportation Systems (RIITS) under our current Archived Traffic Data Management System (ADMS) project with the METRANS Transportation Center (METRANS).

Building on our current data management and trajectory analysis research help developed the algorithms and a system that can process and analyze large GPS datasets for this project. The inputs to our algorithms and system are the GPS datasets collected from buses in LA County and several other related datasets (e.g., timetables, road network topology data, and traffic sensor data). In particular, the research focuses on developing several novel components to (i) clean and transform GPS datasets; (ii) map-match and integrate the cleaned datasets to the road network and timetables; and (iii) mining algorithms to compute performance and reliability metrics such as travel-time reliability, bus bunching, and bus arrival-time estimation.

This report presents our process and the results of cleaning and analyzing the bus sensor and schedule data and demonstrates our system including web services and a web-based dashboard. First, we formulate definitions of relevant terms that will be used throughout the report. Next, in order to perform data analysis on trajectory data of LA metro buses, we need to first clean the bus GPS data since the sensor data are sampled in a low frequency (around 3 - 4 minutes) and can contain errors in attributes such as direction, progress, and schedule of the buses. Here we present our data preprocessing steps, which cleans and prepares data for our analyses. Third, we present our arrival time estimation algorithm, which is an essential step for all analyses. Then, we show our data analysis process and the results we obtained from the process. Finally, we describe the implementation of the system and use example cases of the dashboard.

2 Problem Definition

Definition 1. *Bus GPS records: A bus GPS record p_i includes the route Identifier (ID) of the bus ($p_i.route$), an identifier which includes bus ID and run ID ($p_i.identifier$), bus direction ($p_i.dir$), latitude and longitude ($p_i.lat$, $p_i.lon$), and time of the record ($p_i.time$).*

Definition 2. *GPS data: GPS data is a set of GPS records $P = \{p_1, p_2, \ldots, p_N\}$.*

Definition 3. *GPS trip: A GPS trip is a sequence of consecutive GPS records $\tau = \{p_1, p_2, \ldots, p_n\}$ that satisfy the property $\forall p_i.p_j \in P, p_i.identifier = p_j.identifier$ and $\forall i \in [0, n-1], p_i.time < p_j.time$*

Definition 4. *Bus stop: A bus stop (or stop) s_i contains information about a specific bus stop, including identification ($s_i.identifier$), latitude and longitude ($s_i.lat$, $s_i.lon$)*

Definition 5. *Scheduled Trip: A scheduled trip is a sequence of stop-times $\phi : \{st_1, st_2, \ldots, st_n\}$ which indicates the progress of a GPS trip of a bus from one stop to the next stop. Every record st_i in a scheduled trip has the same route ID ($st_i.route$), expected arrival time ($st_i.time$) and a unique stop sequence ($p_i.seq$) depicting the progress of the trip. Each scheduled trip ϕ_i is also associated to a trip ID ($\phi_i.trip$) and a service ID ($\phi_i.service$), which indicates days of the week that the trip would operate.*

Definition 6. *Trajectory Schedule Mapping: Find a scheduled trip for each GPS trip. More formally, find a mapping $F : T \mapsto \Phi$ such that $\forall \pi_j \in T, \exists \phi_i \in \Phi : \phi_i = F(\pi_j)$*

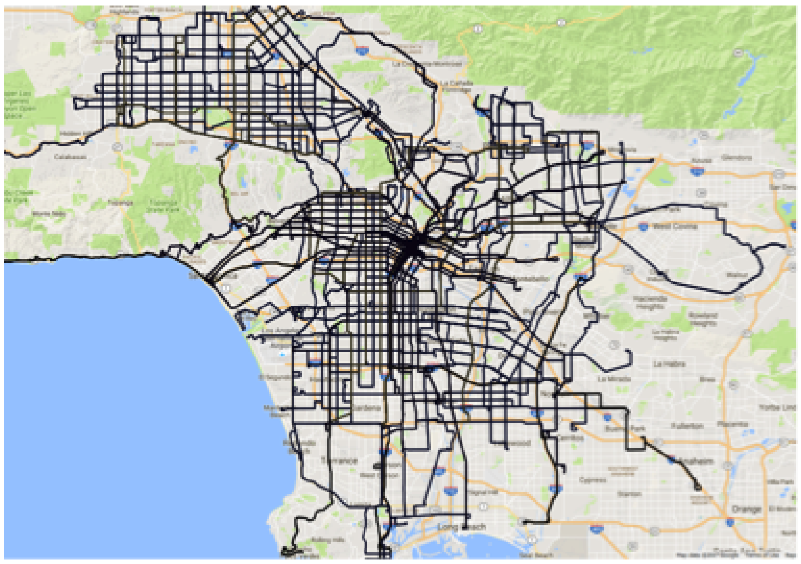


Figure 1: Examples of route maps of all bus trajectories in LA.

3 Datasets

3.1 GPS dataset

The GPS dataset used in this study came from RIITS, a communication system that supports the real-time transformation of information sponsored by LA Metro. Currently, 144 bus routes were operated and recorded by Metro. Approximately one million records are sent each day on average. The time interval of each sensor data report is about 3 minutes. This report shows the results of the analyses on data from January 01, 2016 to October 11, 2017.

The main fields in the GPS dataset are route ID, bus ID, run ID, record time, latitude and longitude. Examples of route maps and bus stops in LA are plotted in Figure 1 and Figure 2. A sample of original GPS datasets is shown in Table 1.

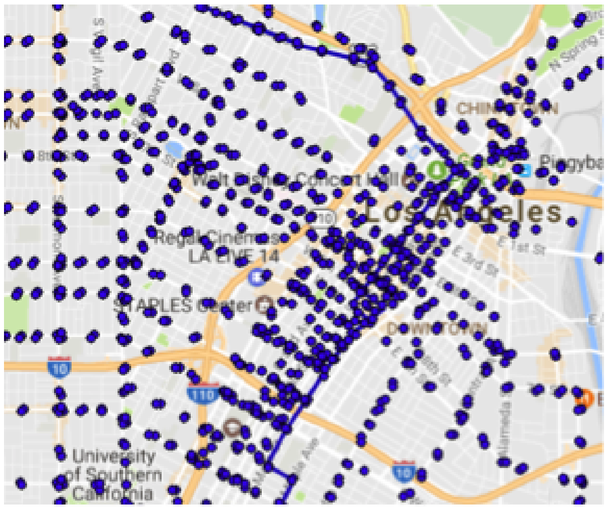


Figure 2: Examples of bus stops in LA.

Table 1: Example of GPS Dataset

Route	Bus ID	Run ID	Direction	Record Time	Latitude	Longitude
10	5765	5602	2	1/1/16 20:58	34.086176	-118.382024
10	5765	5602	2	1/1/16 21:04	34.086176	-118.382056
10	5765	5602	2	1/1/16 21:07	34.086176	-118.381980
10	5765	5602	2	1/1/16 21:10	34.086176	-118.382024
10	5765	5602	2	1/1/16 21:13	34.086149	-118.381980

3.2 Schedule dataset

The schedule dataset used in this study is provided by LA Metro in the General Transit Feed Specification (GTFS) format. It offers a total of 38719 trips of 146 bus routes in LA. The GTFS dataset used in this study is the December 2017 version, available at the LA Metro website [1].

The original GTFS data in this study is preprocessed into one integrated and dimensionally reduced dataset with only the crucial information needed for the matching process. The processed dataset includes fields such as corresponding route ID, trip ID, stop sequence, stop name, stop arrival time, latitude and longitude. A sample of the schedule dataset is shown in Table 2.

Table 2: Example of GPS Dataset.

Route	Trip ID	Service	Direction ID	Stop Seq	Stop ID	Stop Name	Arrival Time	Latitude	Longitude
10	37095669	Weekday	0	1	6000	Santa Monica / Larrabee	10:26:00	34.085548	-118.382774
10	37095669	Weekday	0	2	2192	San Vicente / Melrose	10:31:00	34.08205	-118.383644
10	37095669	Weekday	0	3	3206	Melrose / Huntley	10:32:00	34.08186	-118.380524
10	37095669	Weekday	0	4	3216	Melrose / La Cienega	10:33:00	34.081833	-118.376678
10	37095669	Weekday	0	5	3224	Melrose / Orlando	10:34:00	34.083279	-118.373421

4 Data Preprocessing

4.1 Overview

In this study, we have a set of GPS records that are not grouped or ordered in any way. In other words, there are only p_i data but not τ_i data in the GPS dataset. Thus, in order to perform the mapping from bus GPS data to stops and schedules, we first have to split the records into trips (i.e. τ_i). Also, the bus direction data (i.e. $p_i.dir$) in the GPS data is not reliable. This means that for some $i \in [0, |P|)$, $p_i.dir$ is either incorrect or undefined. Thus, a bus direction recovery process must be performed to correct all incorrect and undefined entries.

Figure 3 shows the steps of data pre-processing. First, raw GPS records separated by route, identifier, and date are mapped to stop sequences in the Trip Progress Mapping step. Then, in the Trajectory Splitting step, they are

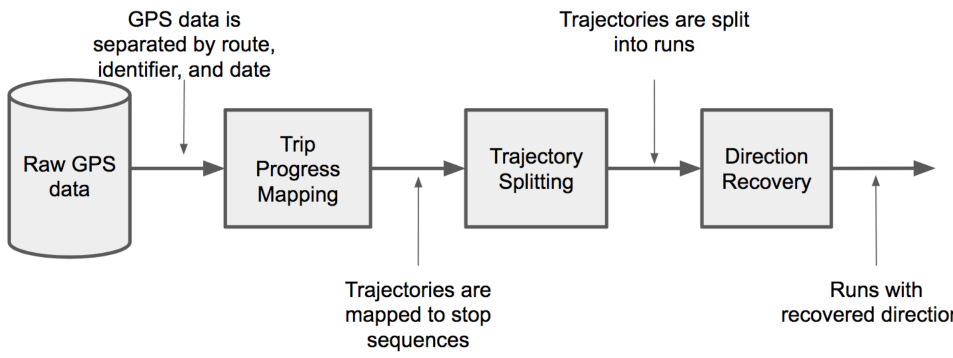

GPS data is separated by route, identifier, and date
Trajectories are split into runs
Raw GPS data
Trip Progress Mapping
Trajectory Splitting
Direction Recovery
Trajectories are mapped to stop sequences
Runs with recovered directions

Figure 3: Pre-processing steps

split into different trajectories where each trajectory represents a run of a bus from start to end stations. The idle points are also removed from these trajectories. Next, the direction of each record in each trajectory is recovered in the Direction Recovery step.

4.2 Trip Progress Mapping

To split the records into different trips, we first have to know the bus positions in the trajectory. Thus, for the GPS records of the same route, identifier, and date, we have to extract the trip progress of buses from location data of the records. Our approach is based on the assumption that every valid trip in the dataset running the same route should travel a sub-trajectory of the route trajectory with a non-strictly increasing or decreasing stop sequence.

More formally, for every route r', there exists a route trajectory T' such that for every record p_i, $p_i \in P$, $r_i = r'$, there is a stop $t_j \in T'$ such that $D(p_i, t_j) < \epsilon$ where ϵ is a fixed value greater than 0. To do the mapping, we select a parent trip of a route r by selecting the trip with the longest cumulative distance for the route. Then, for every p_i running route r, we define a mapping function $g : (lat, lon) \to seq$ that maps any location to

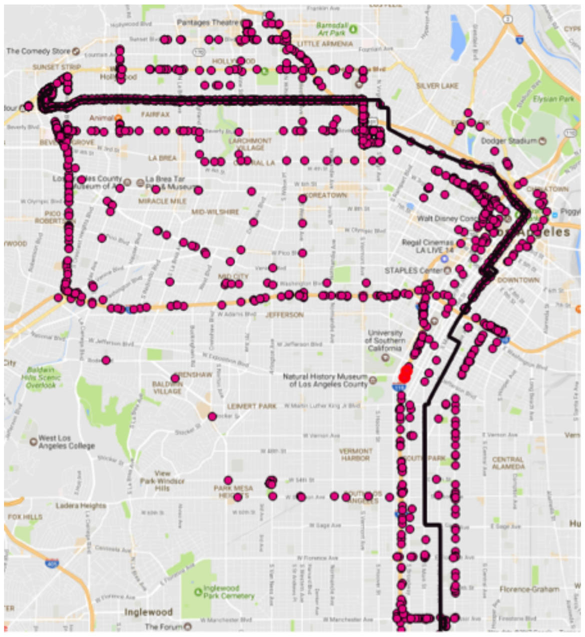


Figure 4: Route 10 outliers on the map.

the stop sequence of the record in the parent trajectory of the route that is closest of the location, and assign $g(p_i.lat, p_i.lon)$ to p_i. Also, if the distance between the GPS point and the stop's location is larger than a threshold (e.g., 400 meters), the GPS point is considered as an outlier and removed from the process. Figure 4 shows the Route 10 outliers on the map. The black line is the trajectory of Route 10, and the red dots are the outlier records. It can be seen from the figure that most of the outlier records are far away from the route trajectory.

The graph of mapping g is shown in Figure 5. Therefore, after completing the trip progress mapping, every record $p_i \in P$ will have a stop sequence $p_i.seq$ associated with it depicting the buses' progress in the trip.

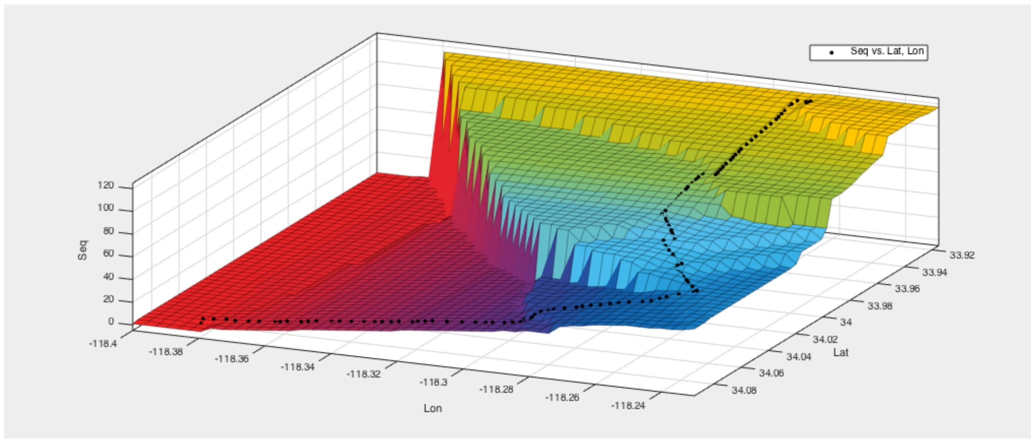


Figure 5: The graph of mapping function g.

4.3 Trajectory Splitting

In bus trajectory splitting, we traverse through the set of GPS records with the same route ID, identifier, and running date. For each record p_i we traverse through, we associate a value $p_i.trend$ to the record such that

$$p_i.trend = sgn\left(g\left(p_i.lat, p_i.lon\right) - g\left(p_{i-1}.lat, p_{i-1}.lon\right)\right) \tag{1}$$

The recursive definition associates every record with whether the trip progress at this record increases compared to the previous one. After completing this step, the trend values of all the p_i's are reviewed. If for a p_i, the largest integer j s.t. $j < i; p_j.trend \neq 0$ and the smallest integer k s.t. $k > i; p_k.trend \neq 0$ satisfies $p_j.trend = p_k.trend$, then it follows that $p_{j+1}, p_{j+2}, \cdots, p_{k-1}$ belong to the same trip as p_j and p_k. Thus, p_m is set to p_j (p_k) for all $j \leq m \leq k$. After this step, every group of consecutive records with the same trend value that is not zero should belong to a trip. Thus, at this point, we can split trips into groups with consecutive records with the same trend value (which we call a "run"), and delete groups with values zero at the beginning or end of each run (as they are identified as idle points).

When buses are not running or in intermission, the bus location data is

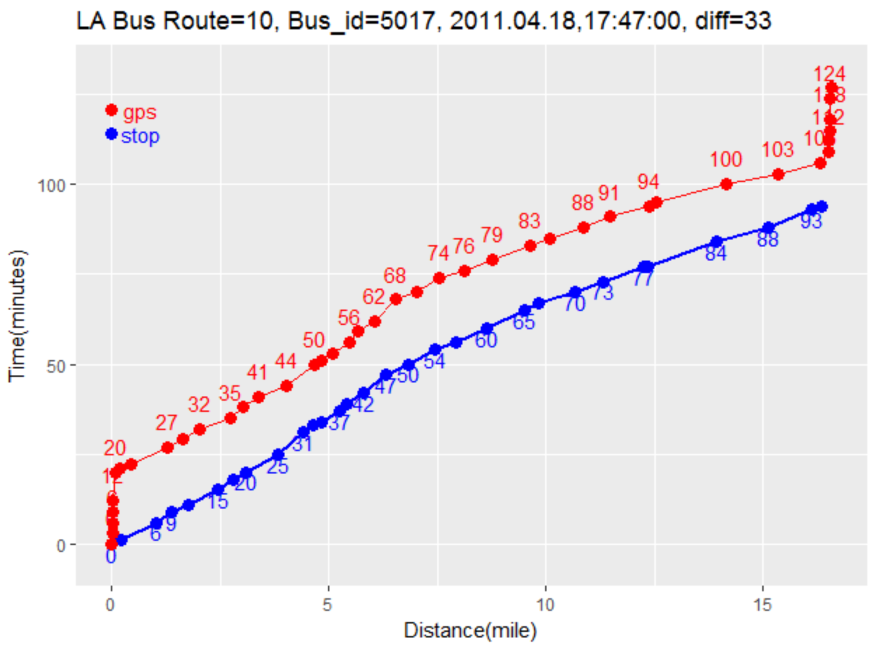


Figure 6: Example of trip with idle points.

still being recorded. These records are recognized as idle points and should be removed from the dataset before the matching process. The presence of idle points can distort travel time information and skew our analyses because the actual departure time of a trip is the time recorded in the last moment before the bus starts moving, not the time recorded in an idle point of data. Thus, it is very important that we eliminate all records of idle points before we perform the matching or any analysis.

Figure 6 illustrates an sample of idle points in GPS trajectory data at the start and end points of a trip. It can be seen that there are lots of data points with a cumulative distance of approximately 0 and 18.5 miles at either end of the trip.

Then, all runs having too few GPS records (e.g., fewer than five records) are removed. This will complete the trip splitting process and remove the idle points between the trips.

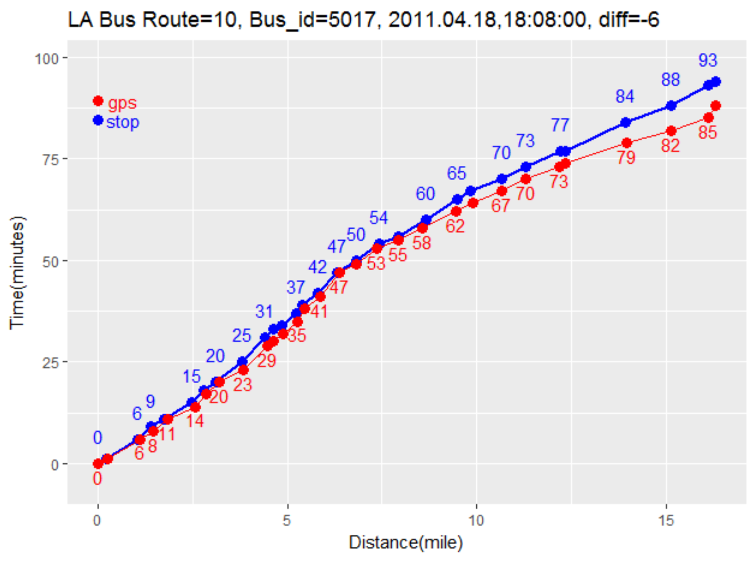


Figure 7: Space-time plot after bus trajectory splitting, direction recovery and removing idle points.

4.4 Direction Recovery

Now, for every run split, there should be a unique trend value associated with all the runs. We can then map the trend value to the representation of direction value in p_i. This will complete the direction recovery process.

4.5 Results

Figure 7 shows an example of space-time plot after bus trajectory splitting, direction recovery and removing idle points.

5 Arrival Time Estimation

In this section, our goal is to estimate the arrival time at each stop by combining the GPS data and schedule data. The reason is that most of the GPS points are not recorded when buses arrived at the stops. This is an essential

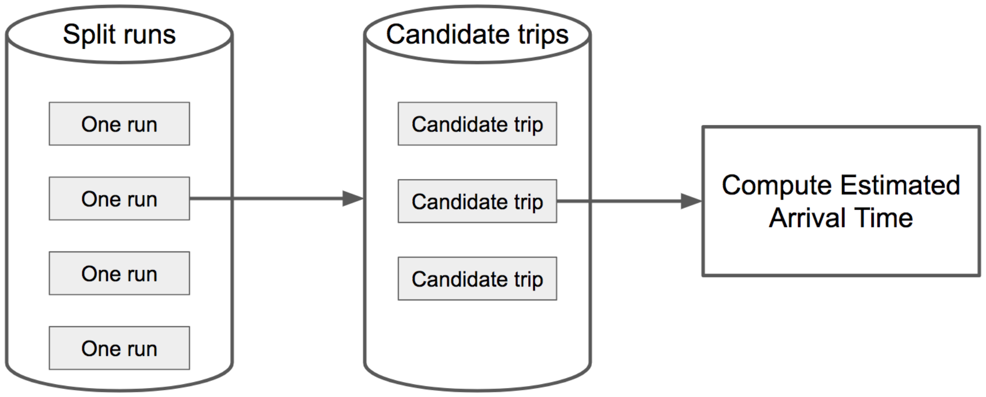


Figure 8: The steps to estimate arrival time

step because arrival time of a bus to a stop can give us the information to compare the estimated arrival time and the schedule, which in turn can help us calculate many different measurements of the data analytics process.

The main steps to estimate arrival time are shown in Figure 8. The general idea is that for each split run, we find its candidate trips (schedules), and then we compute the estimated arrival time for the stops in each candidate of that run. Algorithms 1 and 2 show the pseudo-code for the main algorithms. Next, we will present details of the algorithm for the bus arrival time estimation.

Algorithm 1 Arrival Time Estimation Algorithm

```
1: function busArrivalTimeEstimation(allSplitRuns, trips)
2:     for aRun in allSplitRuns do
3:         tripCandidates ← findTripCandidates(aRun, trips)
4:         for aTrip in tripCandidates do
5:             computeEstimatedArrival(aRun, aTrip)
6:         end for
7:     end for
8: end function
```

First, for a run of a bus of a specific route, all the trips of that route

Algorithm 2 Trip Candidate Selection Algorithm

```
1: function FINDTRIPCANDIDATES(aRun, trips)
2:     select trips covering the time span of the run
3:     for aTrip in trips do
4:         distance ← sum of distances from GPS points to their nearest stops
5:     end for
6:     rank trips by distance
7:     select trips with distance ≤ distance of the closest trip + threshold
8: end function
```

that have a start and end time covering the start and end time of the run are considered as candidate trips. The purpose is to reduce the number of trips to which the run may belong. For example, if the run GPS lasted from 2016-8-20 08:00:00 to 2016-8-20 08:57:00, we should not choose the trips (schedules) in the afternoon or evening.

Then, for each candidate trip, we calculate the distance from the GPS run to the trip. The distance is calculated based on the sum of the distances from each GPS record of the run to its closest stop of the trip. For example, in Figure 9, we have a GPS run containing three GPS points G1, G2, and G3, and a candidate trip T1 with four stops. As the algorithm goes, we will find S1, S2, S3 as the closest stop to G1, G2, G3 with a distance of d1, d2, d3, respectively. Then, the sum of those distances (d1 + d2 + d3) can be considered as the distance from the GPS run to the trip T1.

Next, the candidate trips are sorted by their distance to the GPS run. Then, the candidate trips with the distance smaller than a threshold compared to the closest candidate are selected. For example, in Figure 10, we have T1 as the closest candidate trip, but also have another candidate trip T2 with four stops (S1, S2, S5, S6) and the distance from the GPS run to T2 is (d1 + d4 + d5). If (d1 + d4 + d5) is less than (d1 + d2 + d3) plus a threshold (e.g. 500 meters), T2 would be considered as the candidates for this run; otherwise, T2 would not be considered. Thresholding by distance

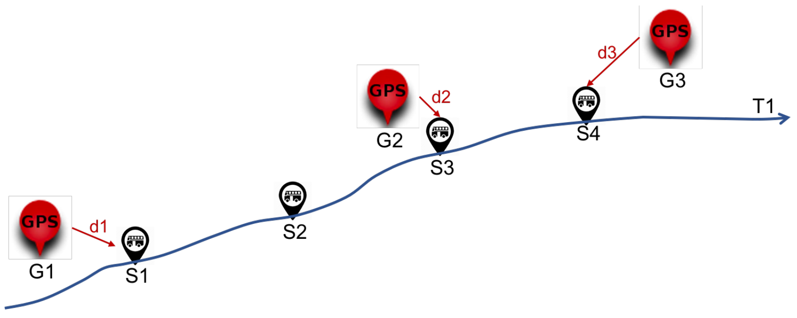


Figure 9: An example of calculating distance from GPS points to their nearest stop

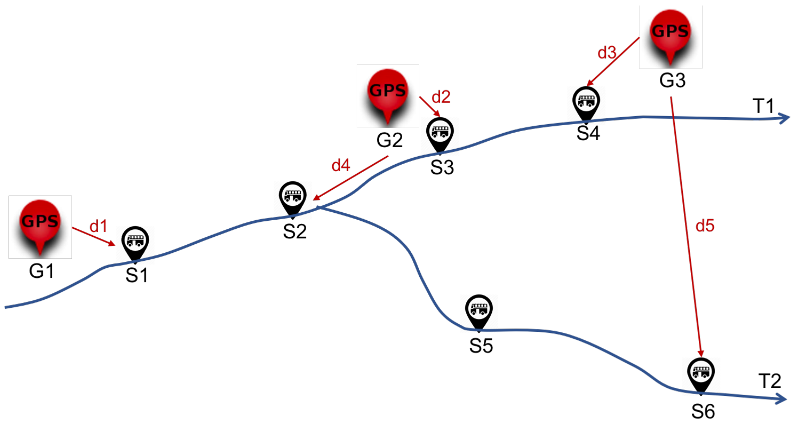


Figure 10: An example of selecting close candidate trips

is helpful because there can be several trips that share the same stops with different schedule times.

Then the GPS run can be used to estimate arrival time for all of those trips. In the example in Figure 11, three trips T1, T2, and T3 have the same bus stops with different schedule times starting at 7:00 am, 7:10 am, and 7:20 am, respectively, and the same GPS run can be used to estimate arrival time for all of these trips.

Next, for each candidate trip of a GPS run, we estimate the arrival time of the bus for each scheduled time at a stop by finding the GPS records of

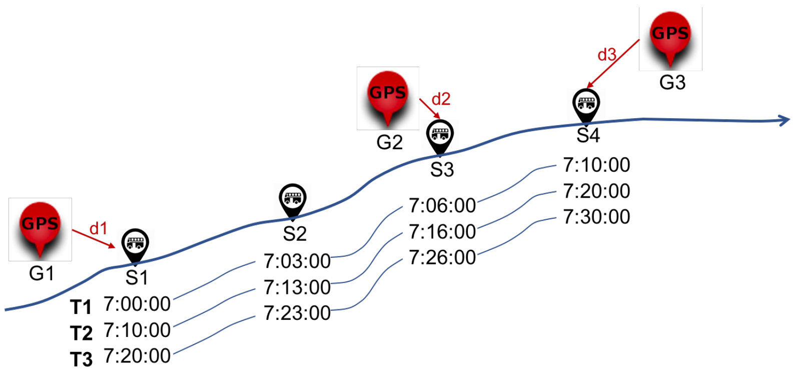


Figure 11: An example of the same stops with difference schedule time

the run that are the first records coming before and after the location of the stop, then calculating distance from the stop to those two GPS records, assuming that the bus ran at a constant speed between those two GPS records, and finally estimating arrival time based on distances and time at the GPS records. For example, in Figure 12, we would like to estimate arrival time for stop S1 which is scheduled at 7:00 am. We find the two GPS points G1 and G2 that are the first records coming before and after S1. We then calculate the distance from G1 and G2 to S1. Using timestamp of G1 and G2 and their distances to S1, we can estimate arrival time of that bus when arriving at S1. Once we have the estimated arrival time, by comparing with the scheduled time (e.g., 7:00 am in this case), we can calculate the delay time at a certain stop and schedule. Algorithm 3 shows the pseudo-code for the estimated arrival time computation algorithm.

6 Data Analytics

6.1 Travel-Time Reliability (or On-Time Performance)

Travel-time reliability (or on-time performance) indicates the percentage of seeing a bus at a scheduled time of a trip at a stop. Based on the definition

Algorithm 3 Estimated Arrival Time Computation Algorithm

```
1: function computeEstimatedArrival(aRun, candidateTrips)
2:     for aTrip in candidateTrips do
3:         for aStopTime in aTrip do
4:             find the closest GPS points to aStopTime
5:             estimate arrival time
6:         end for
7:     end for
8: end function
```

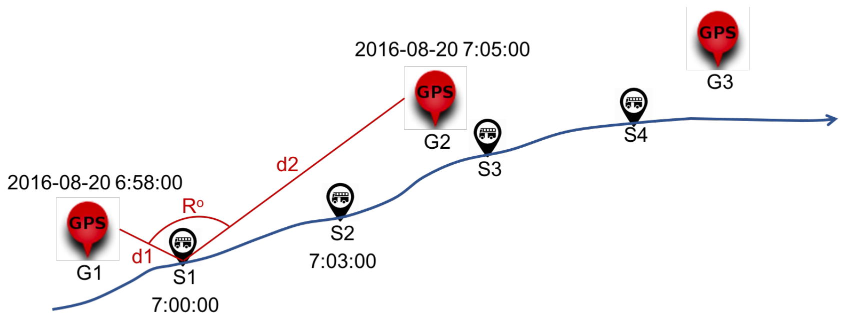


Figure 12: An example of the same stops with difference schedule time

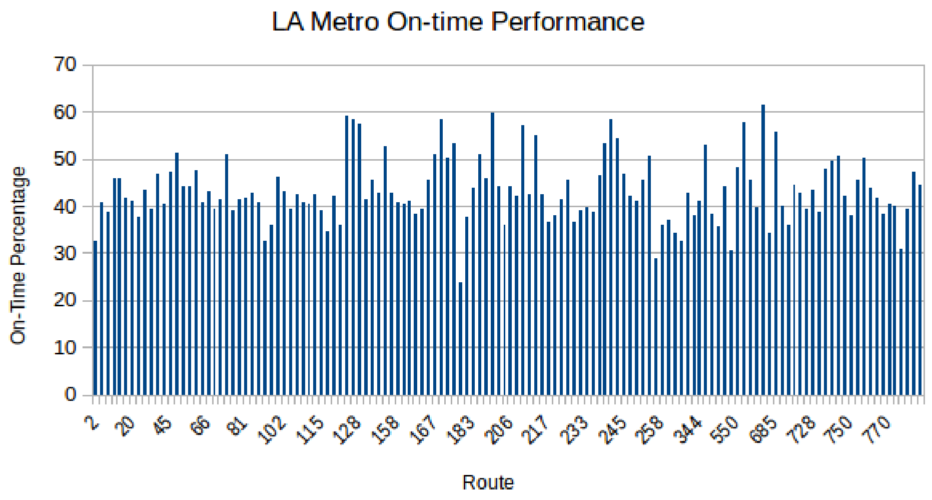


Figure 13: Average travel-time reliability (or on-time performance) for all routes - for brevity, only some route names are shown in x-axis

of LA Metro [2], the range of one minute early and five-minutes late is used as the range of on-time achievement. Thus, from estimated arrival times of buses to a specific schedule and the definition of on-time achievement, a scheduled time at a stop is marked as on-time if there is an estimated arrival time of a bus lying between the one minute early and five-minute late range. Then, the percentage of on-time scheduled times are calculated for each route, for each stop on a route, and for each scheduled time (trip) of a stop of a route. Figure 13 shows the average travel-time reliability (or on-time performance) in percentage for all routes.

6.2 Travel-Time Deviation

The travel-time deviation is the difference between a scheduled time of a trip at a stop and the estimated arrival time of a bus based on GPS data. All the values of the travel-time deviation are then averaged to show the average travel-time deviation for each route, stop of a route, and scheduled time

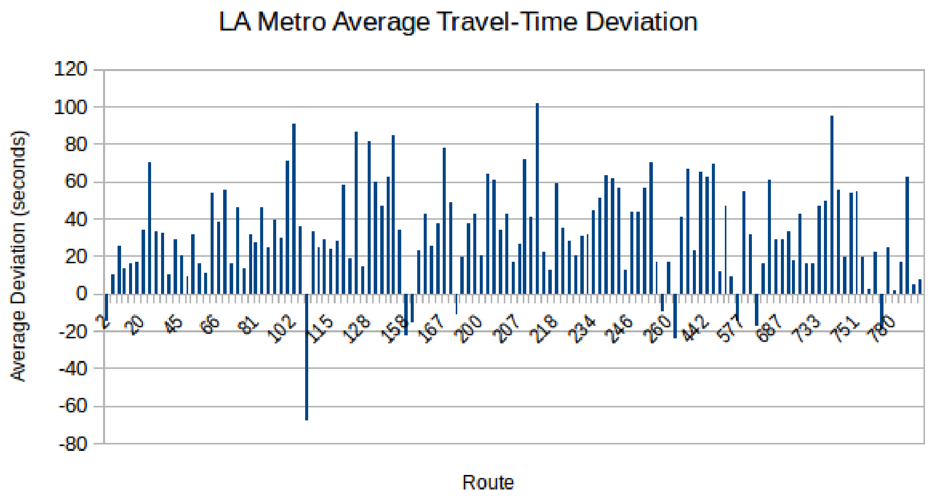


Figure 14: Average travel-time deviation for all routes - for brevity, only some route names are shown in x-axis

(trip) of a stop on a route. Figure 14 shows average travel-time deviation in seconds for all routes.

6.3 Bus Bunching

For many different reasons such as traffic jams or crowded passengers, some buses may run slower than normal buses, thus creating the situation where there are several buses of the same route running too close to each other. This situation is called bus bunching.

Therefore, bus bunching can be calculated as the number of buses arriving within the one minute early and five-minute late range at a scheduled time of a trip at a stop. If there is more than one arrived buses, a bus bunching event is marked. Then the percentage of bus bunching can be calculated for each route, for each stop of a route, and for each scheduled time (trip) of a stop of a route. Figure 15 shows average bus bunching in percentage for all routes.

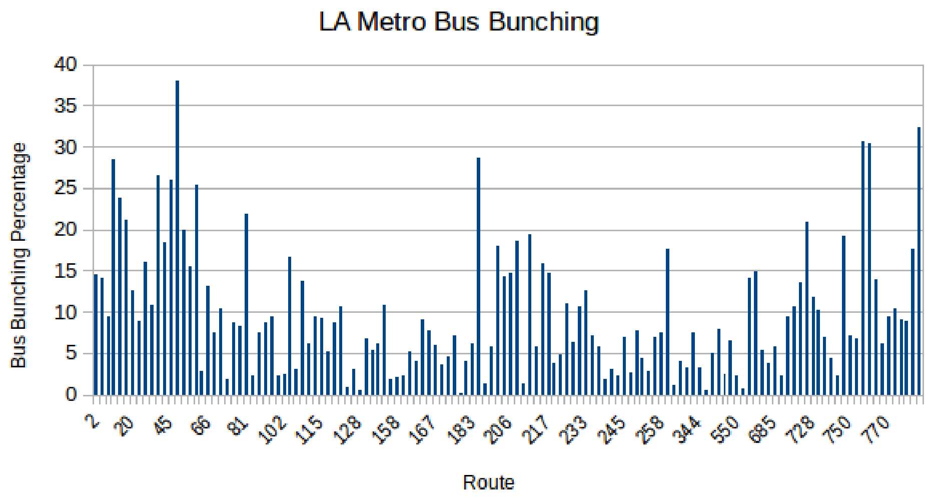


Figure 15: Average bus bunching for all routes - for brevity, only some route names are shown in x-axis

6.4 Travel-Time Estimation (Waiting Time Estimation)

Travel-time (or waiting time) estimation shows the average time a passenger has to wait for the bus to arrive at the scheduled time of a trip at a stop. This estimation can also be calculated from the estimated arrival times by choosing the minimum positive value of delay time as the time a passenger has to wait. Then, the average travel-time estimation can be evaluated for each route, for each stop of a route, and for each scheduled time (trip) of a stop of a route. Figure 16 shows average waiting time in seconds for all routes.

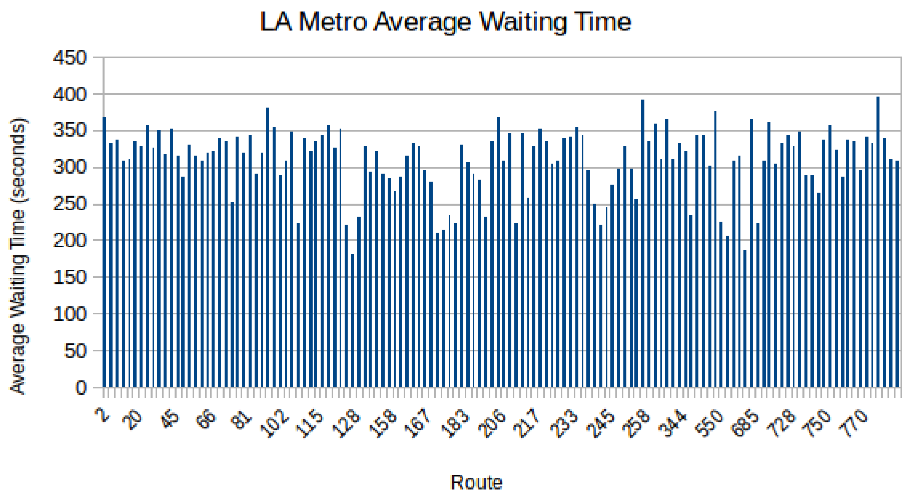


Figure 16: Average waiting time for all routes - for brevity, only some route names are shown in x-axis

7 System Implementation

7.1 Overview

The overview of system implementation is shown in Figure 17. The GPS Database stores all GPS data. The Arrival Time Estimation Service fetches GPS data from the GPS database, estimates arrival times, and stores them into the Estimated Arrival Time Database, from which the Analytics Service then can fetch data and calculate different analytics such as travel-time reliability, travel-time deviation, bus bunching, and travel-time estimation, and stores results in the Analytics Database. This analytics can be fetched by the Web Service to serve data to other agencies such as a Web Page. In the next section, different parts of the system will be explained.

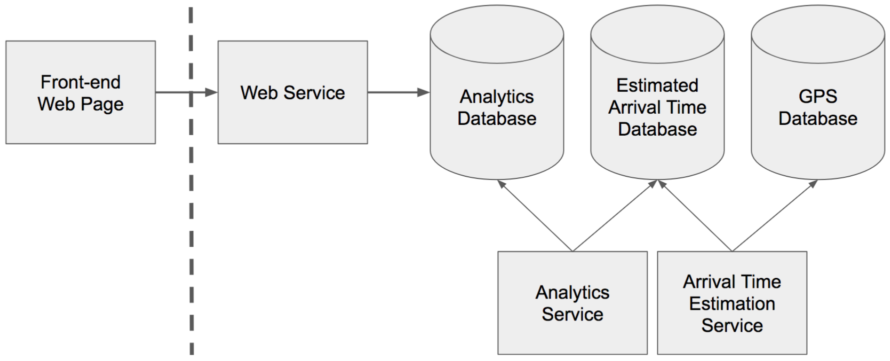


Figure 17: System Implementation Overview

7.2 Metrans Dashboard

The Metrans Dashboard is the web front-end for users to interact with analyses. Figure 18 shows the start page of the Dashboard with the welcome messages.

When a user clicks on the "Dashboard" button on the top-left corner of the Dashboard, she is navigated to the Overview page, which is shown in Figure 19. In the Overview page, on the left, the list of all bus routes is shown, and users can utilize it to view more information about a specific route. In the center is the map with the shape of routes displayed, and on the right is the overall information of all routes.

The "Basic Information" box shows basic information of our dataset, such as the number of bus routes is 136, the number of stops is 14002, and so on. The box also shows overall information such as the average travel-time reliability over all routes is about 42%. Below the box are some graphs showing our measurements, which are Deviation, Bus Bunching, Reliability, and Waiting Time, grouped by the hour of the day, the month, and day of the week.

When a user clicks on a specific route on the list on the left, she is navigated to a route-specific page where more information of the selected

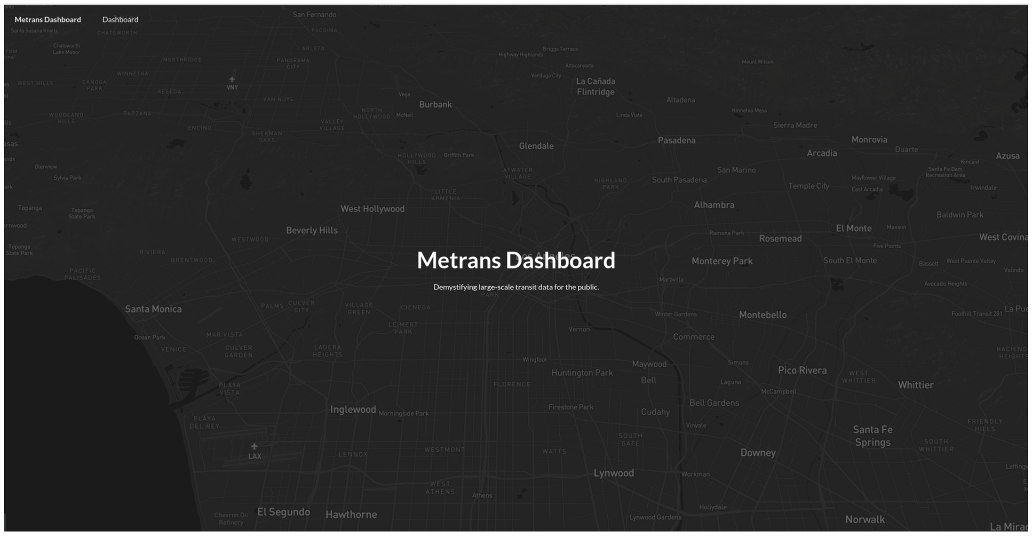
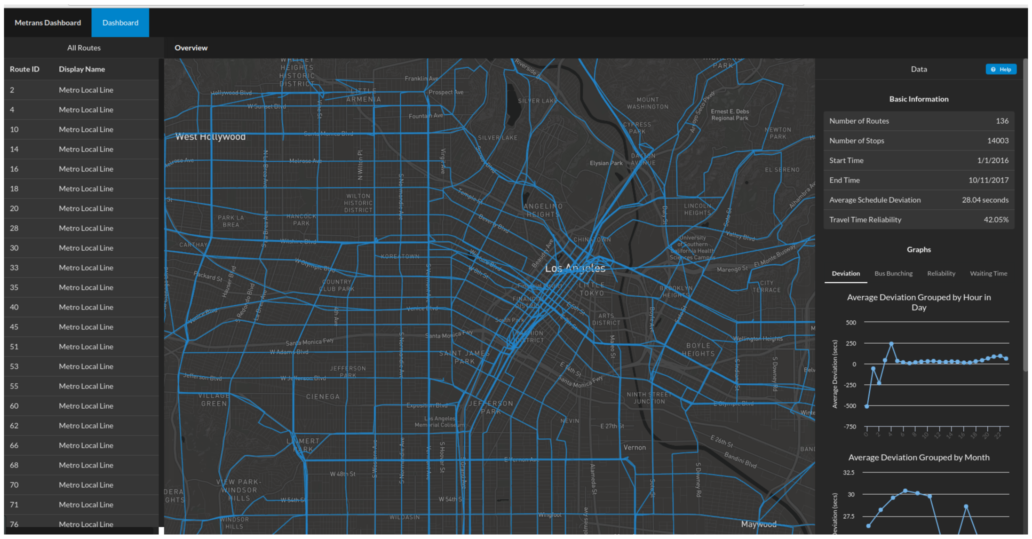


Figure 18: Metrans Dashboard Welcome Page

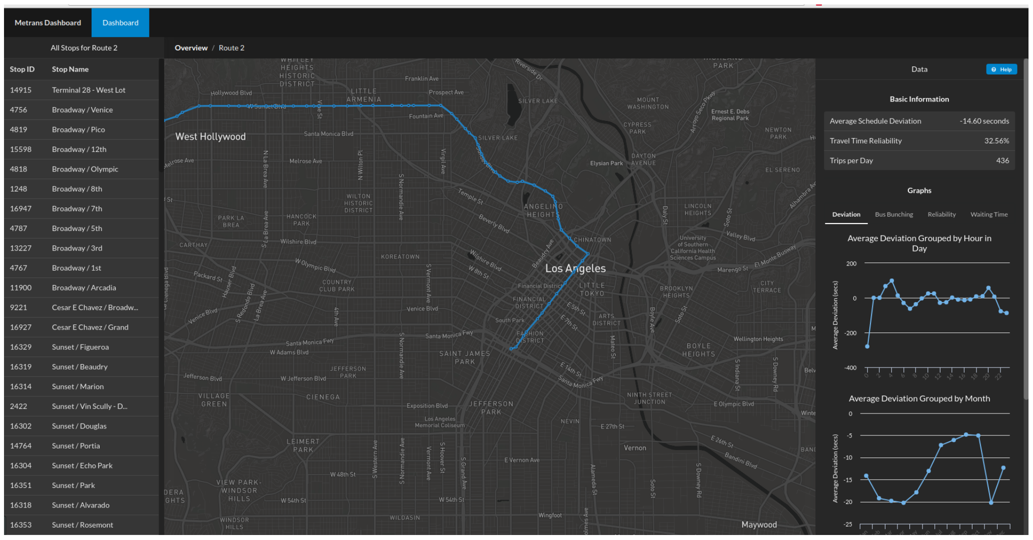


Route ID	Display Name
2	Metro Local Line
4	Metro Local Line
10	Metro Local Line
14	Metro Local Line
16	Metro Local Line
18	Metro Local Line
20	Metro Local Line
28	Metro Local Line
30	Metro Local Line
33	Metro Local Line
35	Metro Local Line
40	Metro Local Line
45	Metro Local Line
51	Metro Local Line
53	Metro Local Line
55	Metro Local Line
60	Metro Local Line
62	Metro Local Line
66	Metro Local Line
68	Metro Local Line
70	Metro Local Line
71	Metro Local Line
76	Metro Local Line

Basic Information

Number of Routes	136
Number of Stops	14003
Start Time	1/1/2016
End Time	10/11/2017
Average Schedule Deviation	28.04 seconds
Travel Time Reliability	42.05%

Figure 19: Metrans Dashboard Overview Page



All Stops for Route 2

Stop ID	Stop Name
14915	Terminal 28 - West Lot
4756	Broadway / Venice
4819	Broadway / Pico
15598	Broadway / 12th
4818	Broadway / Olympic
1248	Broadway / 8th
16947	Broadway / 7th
4787	Broadway / 5th
13227	Broadway / 3rd
4767	Broadway / 1st
11900	Broadway / Arcadia
9221	Cesar E Chavez / Broadw...
16927	Cesar E Chavez / Grand
16329	Sunset / Figueroa
16319	Sunset / Beaudry
16314	Sunset / Marion
2422	Sunset / Vin Scully - D...
16302	Sunset / Douglas
14764	Sunset / Portia
16304	Sunset / Echo Park
16351	Sunset / Park
16318	Sunset / Alvarado
16353	Sunset / Rosemont

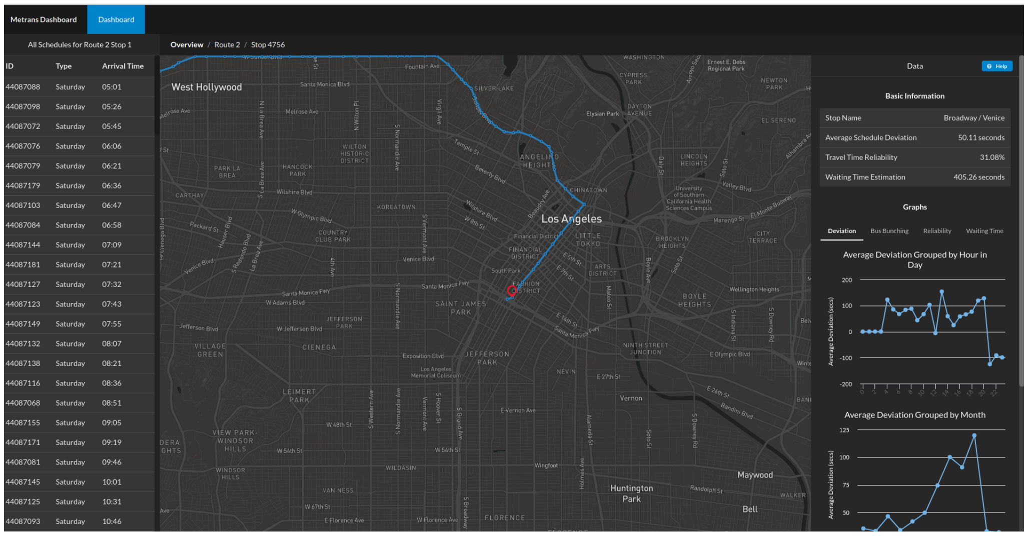
Basic Information

Average Schedule Deviation	-14.60 seconds
Travel Time Reliability	32.56%
Trips per Day	436

Figure 20: Metrans Dashboard Route Information Page

route is shown. Figure 20 shows a route-specific page when a user clicks on Route 2. The left side of the page shows the list of all stops of the route, the shape of the route is displayed on the map in the center, and more information is shown on the right similar to the Overview page. Above the map in the center is the navigation panel, which tells the users where she is in the process of exploring the Dashboard. In this example, the user is viewing information for Route 2. The navigation panel is available to navigate back and forth between the pages. For example, if the user wants to go back to the Overview page, he or she can click on the Overview part of the navigation panel to be directed to the Overview page.

Next, when the user clicks on a specific stop, she will be directed to a stop-specific page for that stop. Figure 21 shows the stop-specific page when the user clicks on Stop 4756 of Route 2. The left side shows the list of all schedule trips for Stop 4756 of Route 2 with the Service days (e.g., Saturday, Weekday) and the scheduled arrival of the trip. In the center is the location of the stop marked on the map, and the right side is more information on the statistics for the stop.



All Schedules for Route 2 Stop 1

ID	Type	Arrival Time
44087088	Saturday	05:01
44087098	Saturday	05:26
44087072	Saturday	05:45
44087076	Saturday	06:06
44087079	Saturday	06:21
44087179	Saturday	06:36
44087103	Saturday	06:47
44087084	Saturday	06:58
44087144	Saturday	07:09
44087181	Saturday	07:21
44087127	Saturday	07:32
44087123	Saturday	07:43
44087149	Saturday	07:55
44087132	Saturday	08:07
44087138	Saturday	08:21
44087116	Saturday	08:36
44087068	Saturday	08:51
44087155	Saturday	09:05
44087171	Saturday	09:19
44087081	Saturday	09:46
44087145	Saturday	10:01
44087125	Saturday	10:31
44087093	Saturday	10:46

Basic Information

Stop Name	Broadway / Venice
Average Schedule Deviation	50.11 seconds
Travel Time Reliability	31.08%
Waiting Time Estimation	405.26 seconds

Figure 21: Metrans Dashboard Stop Information Page

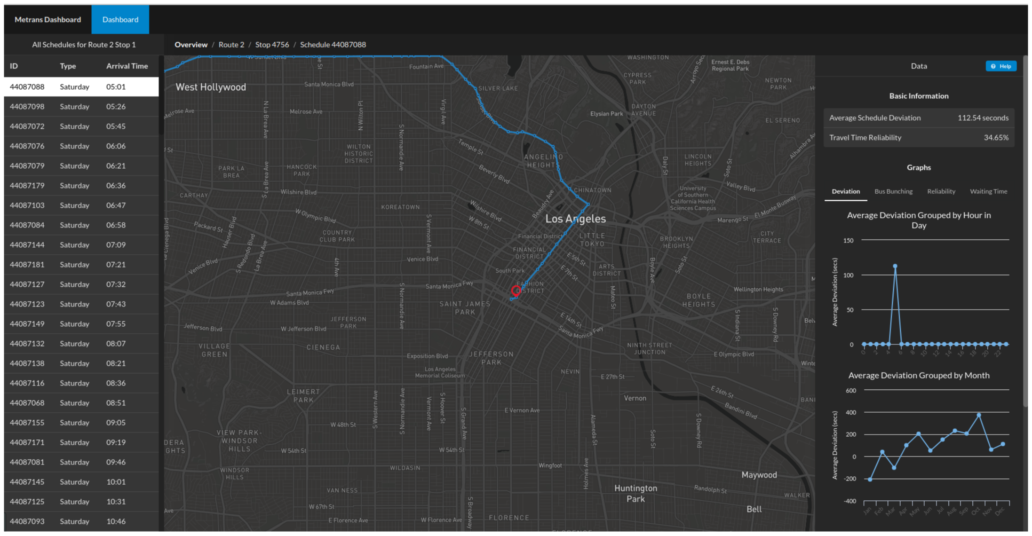
When the user clicks on a specific trip of a stop on the list, the stop-specific page changes the information of the stop on the right side to the information of the selected-trip. For example, in Figure 22, the user selected the first trip on the list of Stop 4756. Then the selected trip is highlighted on the list, and the information on the right side is changed to show the information of trip 44087088.

At any time, the user can use the navigation panel above the center map to navigate between different pages. For example, if the user clicks on the Route 2 part of the navigation panel on Figure 22, she will be directed to the route-specific page of Route 2 as shown in Figure 20.

7.3 Web Service

The web service serves as the portal for other parties to retrieve information for all the statistics. The Dashboard is an example of a website retrieving information from the web services and displayed to users.

Currently, for the proof-of-concept implementation, the web service is


Metrans Dashboard
Dashboard
Overview / Route 2 / Stop 4756 / Schedule 44087088
Data
Help
Graphs
Deviation
Bus Bunching
Reliability
Waiting Time
Average Deviation Grouped by Hour in Day
Average Deviation (secs)
Average Deviation Grouped by Month

All Schedules for Route 2 Stop 1

ID	Type	Arrival Time
44087088	Saturday	05:01
44087098	Saturday	05:26
44087072	Saturday	05:45
44087076	Saturday	06:06
44087079	Saturday	06:21
44087179	Saturday	06:36
44087103	Saturday	06:47
44087084	Saturday	06:58
44087144	Saturday	07:09
44087181	Saturday	07:21
44087127	Saturday	07:32
44087123	Saturday	07:43
44087149	Saturday	07:55
44087132	Saturday	08:07
44087138	Saturday	08:21
44087116	Saturday	08:36
44087068	Saturday	08:51
44087155	Saturday	09:05
44087171	Saturday	09:19
44087081	Saturday	09:46
44087145	Saturday	10:01
44087125	Saturday	10:31
44087093	Saturday	10:46

Basic Information

Average Schedule Deviation	112.54 seconds
Travel Time Reliability	34.65%

Figure 22: Trip-specific information is shown on the right side when users click on a specific trip

deployed at the base Uniform Resource Locator (URL) `http://gdt.usc.edu:8888/metransws/main`. Figure 3 show the list of the services currently provided by our Web Service. The URL mapping shows the URL part that should be added to the base URL to get the information as described in the Description column. For example, to get the longest shapes of all routes, the full should be `http://gdt.usc.edu:8888/metransws/main/shape/all`.

The part in the curved brackets should be replaced by the specific data. For example, {route_id} should be replaced by a specific route number (e.g. 10). So, the full URL to get stops of route 10 would be `http://gdt.usc.edu:8888/metransws/main/list/route/10`.

The "type" for the query to get analytics of a type includes "deviation, busbunching, reliability, waitingtime", which would return travel-time deviation, bus bunching, travel-time reliability, and waiting time estimation, respectively. For example, to get bus bunching analytics for route 10, stop 6000, the full URL would be `http://gdt.usc.edu:8888/metransws/main/stats?type=busbunching&route=10&stop=6000`, and a sample response is:

Table 3: List of the services provided by the Web Service

#	Description	URL Mapping
1	Get The Longest Shapes of All the Routes	/shape/all
2	Get All Shapes of a Route	/shape/route/{route_id}
3	Get Names of All the Routes	/list/routes
4	Get Stops of a Route	/list/route/{route_id}
5	Get Trips for a Specific Stop	/list/route/{route_id}/stop/{stop_id}
6	Get Basic Info for Overall/Route/Stop/Trip	/basicinfo?route={route_id} &stop={stop_id}&trip={trip_id}
7	Get Analytics of a Type for a Route/Stop/Trip	/stats?type={type}&route={route_id} &stop={stop_id}&trip={trip_id}

```
{"day":[0.0,0.0,0.0,0.0,0.0,0.040,0.148,0.119,0.120,0.039,0.024,
0.055,0.142,0.026,0.124,0.052,0.054,0.051,0.090,0.1333,0.027,
0.006,0.002,0.003],
"month":[0.172,0.048,0.055,0.045,0.050,0.062,0.058,0.059,
0.0769,0.0722,0.05928,0.053],
"week":[0.0360,0.108,0.087,0.090,0.082,0.095,0.0304]}
```

8 Conclusion and Future Directions

The report describes the process of data cleaning and analytics for LA Metro bus lines from GPS data and schedule (GTFS) data. The tangible outcome includes two types of assets. The application assets include (i) an interactive web-based application that enables visualization, querying, and analysis of performance metrics of buses in LA County based on real-time and historical GPS data; and (ii) a suite of web-services that implements our statistical bus performance detection algorithms. A remote user can call these services to query data hosted on our servers. The second type of assets are the data mining algorithms to effectively analyze the performance of public transportation vehicles based on their GPS trajectories.

Together the applications demonstrate how the algorithms developed from this project will help to increase the efficiency of the public transportation systems. For example, the applications can be used by city transportation agencies to quickly identify the problem with bus lines, such as delays possibly caused by driver behavior or bus technical problems or quantify the delays in bus lines caused by construction in the city. Even long-term policy decisions can be made to rearrange bus timetables. Also, the web application will benefit riders to have a better understanding and access to travel-time delays and reliability.

Future directions include building on the current project outcomes to develop a comprehensive analytic framework on public transportation data. The current web and service applications, while useful, still provide limited types of analytics to the user. We envision a comprehensive analytic framework that will include (i) a web interface that the users can customize their queries to test scenarios and policy hypothesis as well as visualize the results and (ii) the algorithms that will recommend possible improvement to the transportation system and use user feedback in an active learning environment to advance the recommendation.

References

[1] Los Angeles Metro. GTFS data. `https://developer.metro.net/introduction/gtfs-data/download-metros-gtfs-data/`. [Online; accessed 14-March-2018].

[2] Los Angeles Metro. Service Performance Indicators. `https://www.metro.net/about/metro-service-changes/service-performance-indicators/`. [Online; accessed 12-March-2018].

[3] United States Department of Transportation. Transportation Statistics Annual Report. `https://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/transportation_statistics_annual_report/index.html`. [Online; accessed 12-March-2018].