

Accuracy and Effectiveness of GTFS Transit Feeds for Trip Planning in Public Transit Networks

By

MAHMOUD ABUSALIM

Marshallplan Research Paper

Spatial Information Management

Carinthia University of Applied Sciences

School of Engineering & IT

Department of Geoinformation & Environmental Technologies

Supervisors

*FH-Prof. Mag. Dr. MSc. MAS Gernot Paulus, School of Spatial Information
Management, Carinthia University of Applied Sciences*

*Dr. Hartwig Henry Hochmair, Associate Professor, Geomatics Program
University of Florida | UF · Ft. Lauderdale Research & Education Center*

Villach, 2 November 2020

Abstract

*This master thesis analyses and evaluates the effectiveness of GTFS (General Transit Feed Specification) Realtime feeds of transit data for planning a multi-modal trip in a public transportation network. Realtime feeds can be used to enhance the delay prediction. For now, BCT has established delay prediction using Realtime feeds internally which predicting the delays up to 60 minutes ('Broward County Transit' 2020). In this work, different data sources, such as GTFS static and GTFS Realtime feeds and OpenStreetMap (OSM) road data, will be integrated into a Routing Graph within the Open Trip Planner (OTP) framework. This facilitates multi-modal trip planning in a transit network under consideration of static timetables information as well as planning of current and future trips to observe transit delays at the time of planning the trip. Moreover, the quality and completeness of GTFS Realtime feed information will be analyzed and visualized in total and per vehicle type. In addition, the accuracy of delay predictions provided as part of GTFS Realtime feed information will be assessed, visualized for all vehicle types. Finally, the delay data will be compared with and without outliers showing how the data distribution changed. Two study areas were proposed for this research, namely the Boston Metropolitan Area (Massachusetts Bay Transportation Authority **MBTA**) and Broward County, Florida (Broward County Transit, **BCT**). MBTA has been chosen as it has bigger area coverage and several transporting modes (Tram, Subway, Bus, Rail & Ferry).*

Keywords: *GTFS Realtime, GTFS static, OTP, OSM, delay prediction, public transit, transportation network, TripPlanning, MBTA, BCT.*

Acknowledgment

I would like to convey my utmost sincere appreciation and thanks to my supervisors FH-Prof. Mag. Dr. MSc. MAS Gernot Paulus and Dr. Hartwig Henry Hochmair for giving me this opportunity and aiding me with their expertise and support throughout this Master thesis. I would also like to give my gratitude to the Marshall Plan Foundation and the FH Kärnten international office for facilitating the means to complete this research at the University of Florida. Thanks to Alexandra Maurer and Sarah Kern for helping me with the paperwork and the visa application. I would like to give a special thank you to my family and friends for supporting me throughout the years and my wife for always being by my side. Lastly, I would like to thank the staff and colleagues at the University of Florida Research Center for enthusiastic support throughout the project.

Table of Content

<i>Abstract</i>	2
<i>Acknowledgment</i>	3
<i>List of Tables</i>	7
<i>Introduction</i>	8
1.1 <i>Motivation</i>	8
1.2 <i>General Goals</i>	8
1.3 <i>Detailed description of the research problem</i>	9
1.4 <i>Research questions</i>	10
1.5 <i>Methodological Considerations</i>	11
1.6 <i>Workflow</i>	12
1.7 <i>Relevance and expected results</i>	13
2 <i>State of the Art and Literature review</i>	14
2.1 <i>Public transportation</i>	14
2.1.1 <i>Schedule timetable</i>	15
2.1.2 <i>Open Data</i>	16
2.1.3 <i>Public transportation standards</i>	21
2.1.4 <i>Usage of GTFS in public transit</i>	22
2.1.5 <i>GTFS-Realtime</i>	24
2.2 <i>OpenTripPlanner</i>	28
2.2.1 <i>Graph structure</i>	29
2.2.2 <i>Graph creation using shapefile or OSM</i>	31
2.3 <i>Related work in Quality Assessment of Open Realtime Data for Public Transportation in the Netherlands</i>	32
3 <i>Methodology</i>	33
3.1 <i>Setting up the OTP</i>	33
3.2 <i>Transit Data</i>	34
3.3 <i>Realtime Data Impacts Evaluation</i>	36
3.3.1 <i>Realtime types and structure</i>	36
3.3.2 <i>Evaluation of Realtime information</i>	36
3.3.3 <i>Definition and concept of scenarios</i>	36
3.4 <i>Study area and geodata</i>	38

3.4.1	<i>Boston Metropolitan Area (Massachusetts Bay Transportation Authority MBTA)</i>	38
3.4.2	<i>Broward County, Florida (Broward County Transit, BCT).....</i>	40
4	<i>Implementation.....</i>	42
4.1	<i>Static Data Preparation.....</i>	42
4.1.1	<i>Static Data Collection</i>	42
4.1.2	<i>Static Data Importing.....</i>	42
4.2	<i>MBTA Realtime Transit Data Completeness & Accuracy Evaluation</i>	44
4.2.1	<i>Realtime Data Collection</i>	44
4.2.1	<i>Realtime Data Transformation.....</i>	45
4.2.2	<i>Realtime Data Importing.....</i>	47
4.2.1	<i>Realtime Data Filtering</i>	47
5	<i>Results</i>	49
5.1	<i>Descriptive statistics of all collected data</i>	51
5.1.1	<i>Descriptive statistics of the collected data according to vehicle type.....</i>	51
5.2	<i>VehiclePositions feeds and TripUpdates feeds' Stops distribution in Boston.</i>	53
5.3	<i>Static and real-time information Comparison.....</i>	54
5.4	<i>Delay analysis.....</i>	57
5.4.1	<i>Total Delay Analysis</i>	57
5.4.2	<i>Stops Delay Analysis</i>	58
5.4.3	<i>Tram, Streetcar & Light rail Stops Delay Analysis.....</i>	59
5.4.4	<i>Subway & Metro Stops Delay Analysis.....</i>	60
5.4.5	<i>Rail Stops Delay Analysis</i>	60
5.4.6	<i>Trips Delay Analysis</i>	61
5.4.7	<i>Tram, Streetcar & Light rail Trips Delay Analysis.....</i>	61
5.4.8	<i>Subway & Metro Trips Delay Analysis</i>	61
5.4.9	<i>Rail Trips Delay Analysis.....</i>	62
	<i>Delay Data Comparison</i>	63
6	<i>Summary.....</i>	66
6.1	<i>Discussion.....</i>	66
6.2	<i>Conclusion.....</i>	68
6.3	<i>Future work</i>	71
7	<i>References</i>	72

List of Figures

Figure 1 Workflow	12
Figure 2 Timetable of bus transportation from Vienna Airport (Schwechat) Platform 4 to Bratislava Main Bus Station (Bratislava AS) (“FEBS Workshop,” 2013).	15
Figure 3 Timetable of a tour from Hermagor to Villach train station and every stop in between (“B&B Sonntagsho,” 2017)	16
Figure 4 Transportation Open Data Benefits (Kaufman, 2012)	18
Figure 5 A GTFS dataset from MBTA transit agency consists of several text files within a ZIP file	22
Figure 6 how is Protocol Buffers used to share data across languages (Masina, 2019, p. 101)	27
Figure 7 The Protocol Buffer compiler auto-generates code to exchange binary GTFS-Realtime messages	27
Figure 8 Architecture of the OpenTripPlanner (South Tyrol Free Software Conference, 2019b).	29
Figure 9 Simple representation of the pattern-based Graph Structure in OTP (GraphStructure, 2015)	30
Figure 10 Edge-based representation of the Graph Structure (Paris Open Source Summit, 2012)	30
Figure 11 OTP directory at Maven Centra (Basic Tutorial - OpenTripPlanner, 2020)	33
Figure 12 OTP Data Flow Model (South Tyrol Free Software Conference, 2019a)	33
Figure 13 OTP Instance running in Broward 2020	34
Figure 14: General Transit Feed Specification (GTFS) relations (Introduction to tidytransit, 2019)	35
Figure 15 General Transit Feed Specification (GTFS) table relations (MARTA Developer Resources, 2019)	35
Figure 16 MBTA subway map (Subway Schedules & Maps MBTA, 2020)	38
Figure 17 FREQUENCIES FOR BUS ROUTES, RAPID TRANSIT LINES AND FERRY ROUTES (MBTA, 2020)	39
Figure 18 GTFS dataset from MBTA transit agency consists of several text files within a ZIP file and GTFS-Realtime files	40
Figure 19 an overview of BCT's service area (Broward County Florida, 2020)	41
Figure 20 Static data in the PostgreSQL Database tables and zip file tables	42
Figure 21 Creating GTFS Static Data tables in PostgreSQL	43
Figure 22 Stops distribution through Boston	43
Figure 23 Python codes that used to download the TripUpdates and the VehiclePositions files	44
Figure 24 Collected Data Analysis Steps	50
Figure 25 Total collected data vs Stops (in Percentage)	52
Figure 26 VehiclePositions count when Current Status = 1 out of all VehiclePositions in Percentage	52
Figure 27 VehiclePositions feeds distribution per stop	53
Figure 28 TripUpdates feeds distribution per stop	54
Figure 29 All vehicles arrival delay with outliers	57
Figure 30 All vehicles departure delay with outliers	57
Figure 31 all stops delay average vs median	58
Figure 32 Positive and Negative Delay Distribution Per Stop	59
Figure 33 Tram, Streetcar & Light rail stops average delay	59

Figure 34 Subway & Metro stops average delay -----	60
Figure 35 Rails stops average delay -----	60
Figure 36 Tram, Streetcar & Light Rail Trips Delay Average-----	61
Figure 37 Subway & Metro Trips Arrival Delay Average-----	62
Figure 38 Rail Trips arrival delay average-----	62
Figure 39 All vehicles arrival delay with outliers -----	63
Figure 40 All vehicles arrival delay without outliers -----	63
Figure 41 All vehicles departure delay with outliers-----	63
Figure 42 All vehicles departure delay without outliers-----	63
Figure 43 Arrival delay with outliers per vehicle type-----	64
Figure 44 Arrival delay without outliers per vehicle type-----	64
Figure 45 Departure delay with outliers per vehicle type-----	64
Figure 46 Departure delay without outliers per vehicle type -----	64

List of Tables

Table 1 The number of applications and ridership in several U.S. cities (“Transit transparency,” 2012)	20
Table 2 Ridership by Mode and Quarter 2016 Present (APTAAAdmin, 2020).....	20
Table 3 Common data standards and file formats used by transportation agencies for different purposes ... Error! Bookmark not defined.	
Table 4 Google Transit Feed Specification tables (Kizoom and Miller, 2008).....	23
Table 5 VehiclePositions table structure in the PostgreSQL database.....	45
Table 6 TripUpdates table structure in the PostgreSQL database	46
Table 7 TripUpdates table after importing it into the PostgreSQL database	47
Table 8 Extract of a trip from VehiclePositions table	48
Table 9 Extract of a trip from Stop_times table	48
Table 10 Descriptive statistics of the collected data.....	51
Table 11 Descriptive statistics of the collected data according to vehicle type.....	51
Table 12 Comparison of stops included in the VehiclePositions and TripUpdates feeds	55
Table 13 Comparison of trips included in the VehiclePositions and TripUpdates feeds	55
Table 14 Comparison of box plots information of arrival and departure delay	58
Table 15 Comparison of arrival delay data with and without outliers	65
Table 16 Comparison of departure delay data with and without outliers.....	65

Introduction

1.1 Motivation

Providing users with transit data updates in Realtime greatly enhances their experience. Providing up-to-date information about current arrival and departure times allows users to smoothly plan their trips. As a result, in case of an unfortunate delay, a rider would be relieved to know that they can stay home a little bit longer. Realtime feeds could also be used to enhance the delay prediction.

GTFS Realtime is a feed specification that allows public transportation agencies to provide real-time updates about their fleet. Broward County Transit (BCT) has tested the delay prediction Realtime feed internally in predicting the delays up to 60 minutes ('Broward County Transit' 2020).

Such an accomplishment can enhance the public transportation experience for people by providing more correct timing information about public transport services.

1.2 General Goals

The overall goal of this project is to evaluate the usability of GTFS Realtime data feeds provided by public transportation service providers for multi-modal trip planning. This work aims to utilize the open-source trip planning framework Open Trip Planner (OTP) with its ability to combine different data sources into routing graphs, including GTFS static, as well as GTFS dynamic data feeds. OTP is an open-source and collaborative mapping platform that facilitates the integration of GTFS Realtime feeds for planning a public transit trip. The routing graph forms the basis for multi-modal trip planning (Jariyasunant *et al.* 2010) that considers the Realtime delay information for public transport systems. Such an accomplishment can enhance the public transportation experience for people by providing more correct timing information about public transport services.

1.3 Detailed description of the research problem

Creating Providing users with current and future expected transit vehicles delays both for departure and arrival improves their travel experience since it facilitates more accurate planning of their trips in the case of travel delays. Public transportation agencies that use GTFS Realtime feed specifications provide their fleet Realtime updates to users and developers. GTFS Realtime is a GTFS extension, which is an open data format for public transportation schedules and associated geographic information designed by Google and several public transit application developers ('GTFS Realtime Overview | Realtime Transit' 2020). Recently, sharing predicted delay information of transit vehicles in Realtime through online feeds is a technical means for several public transit agencies. Incorporating GTFS Realtime could enhance the accuracy of the transport applications, such as trip planners, and will make these trip planners more reliable and competitive compared to other ways of transportation. Until now, the effectiveness of Realtime feeds' is still not well utilized. Earlier studies focusing on simulations and observed delays in bus networks revealed that using such novel methods for trip planning render only marginal improvement of route effectiveness compared to trip planning based on static information methods (Hickman and Wilson 1995). Another study found that the Realtime feeds are incomplete and predicted delays are erroneous (Steiner *et al.* 2015), possibly due to the early stage of implementation of that system, which rendered it unsuitable for Realtime routing applications and comparison to static trip planning (Steiner 2014).

On the one hand, the Massachusetts Bay Transportation Authority (MBTA) provides has GTFS Realtime data that provides trip updates, live alerts, vehicle location, and arrival-prediction data in an industry-standard format as separate files. GTFS Realtime is best for retrieving data for the entire system at once in relatively small packages but must be spatially extrapolated using the agency's GTFS data to be meaningful. Their data includes subway, bus, commuter rail, and ferry ('MBTA' 2020).

On the other hand, Broward County Transit (BCT) recently equipped all its buses with positioning and telemetric units, which allows Realtime tracking of its bus fleet. Based on this, BCT had a GTFS Realtime feed system developed through a third-party vendor. This system provides delay prediction for up to 60 minutes. Because the feed is still in the testing phase, it has not been yet released for public consumption. It is, however, expected to become available soon.

GTFS Realtime data feeds can be integrated into routing computations within OTP. OTP can combine different data sources into a routing graph, which builds on time table data and spatial information about routes and stops from static GTFS data, GTFS Realtime data feeds for public transit delays, and OSM data extracts that connect the transit with the road network. The routing graph is then used as the basis for multi-modal trip planning that considers the Realtime delay information.

These statements lead to the following research objectives:

- I. Integration of different data sources (GTFS, OSM) into a routing graph within the OTP framework
- II. Evaluation of the accuracy of delay prediction through comparison with posterior trip characteristics for a selected set sample set of trips and different user scenarios
- III. Exploration of the accuracy of existing arrival and departure prediction methods, e.g. artificial neural networks on public transport Realtime feeds by comparing these results with the VehiclePositions feeds from GTFS Realtime data feed.
- IV. Potential combination of a public transport Realtime information system and incident management system for BCT since in that system Realtime data feeds can currently not remove schedule times for buses that have been suspended e.g. due to mechanical or other issues

1.4 Research questions

This earlier problem description leads to the following questions:

- What is the feasible accuracy of the public transport delay prediction when comparing it with a posteriori trip characteristics (e.g. the observed vehicle positions at a later point of time) for a selected set sample set of trips (e.g. metro and bus trips for selected lines) and different user scenarios (e.g. have or not have access to a smartphone during a trip)?
- What is the feasibility of applying existing bus arrival prediction methods, e.g. artificial neural networks on public transport Realtime feeds? and, if feasible, compare these results with the accuracy of the presently used GTFS Realtime data feeds.

- How can we combine a public transport Realtime information system with an incident management system (as Realtime data feeds currently do not have the ability to remove schedule times for buses that have been suspended e.g. due to mechanical or other issues)?

1.5 Methodological Considerations

Setting up the OTP, the transformation of the GTFS feed, the combination of different data sources into a routing graph, as well as data extraction of OSM that connect the transit with the road network is considered. The routing graph is then used as the basis for multi-modal trip planning under consideration of Realtime delay information next to the evaluation of delay prediction. Also, the feasibility to apply existing bus arrival prediction methods on MBTA or BCT Realtime feeds and compare these results with the accuracy of the presently used by MBTA or BCT GTFS Realtime data feed. So far, most of the local transit companies only provide static GTFS although, some agencies offer Realtime location display for buses, most of their data are not downloadable. Only a few bigger cities provide Realtime GTFS data which comes in the Protocol buffer (Pb) format ('Developer Guide | Protocol Buffers' 2020) and always includes trip updates, service alerts, and vehicle positions as separate files. Finally, a separate accuracy assessment of delay predictions is added for the different transit modes (metro, bus, etc.). This works better for those large metropolitan areas than in Broward County which has only bus and train. However, a very important aspect is the availability, consistency, and completeness of the data provided. Moreover, supplied data should contain a complete table of timing information without missing time slots and the data should constantly fill over a sufficient period so prediction output can be as accurate as possible. It is worth mentioning that several prediction methods can be used such as Artificial Neural Networks ((Chien *et al.* 2002), (Lin *et al.* 2013)), decision support systems (Sun *et al.* 2016), Combinatorial Optimization (Kroon *et al.* 2016), etc.

1.6 Workflow

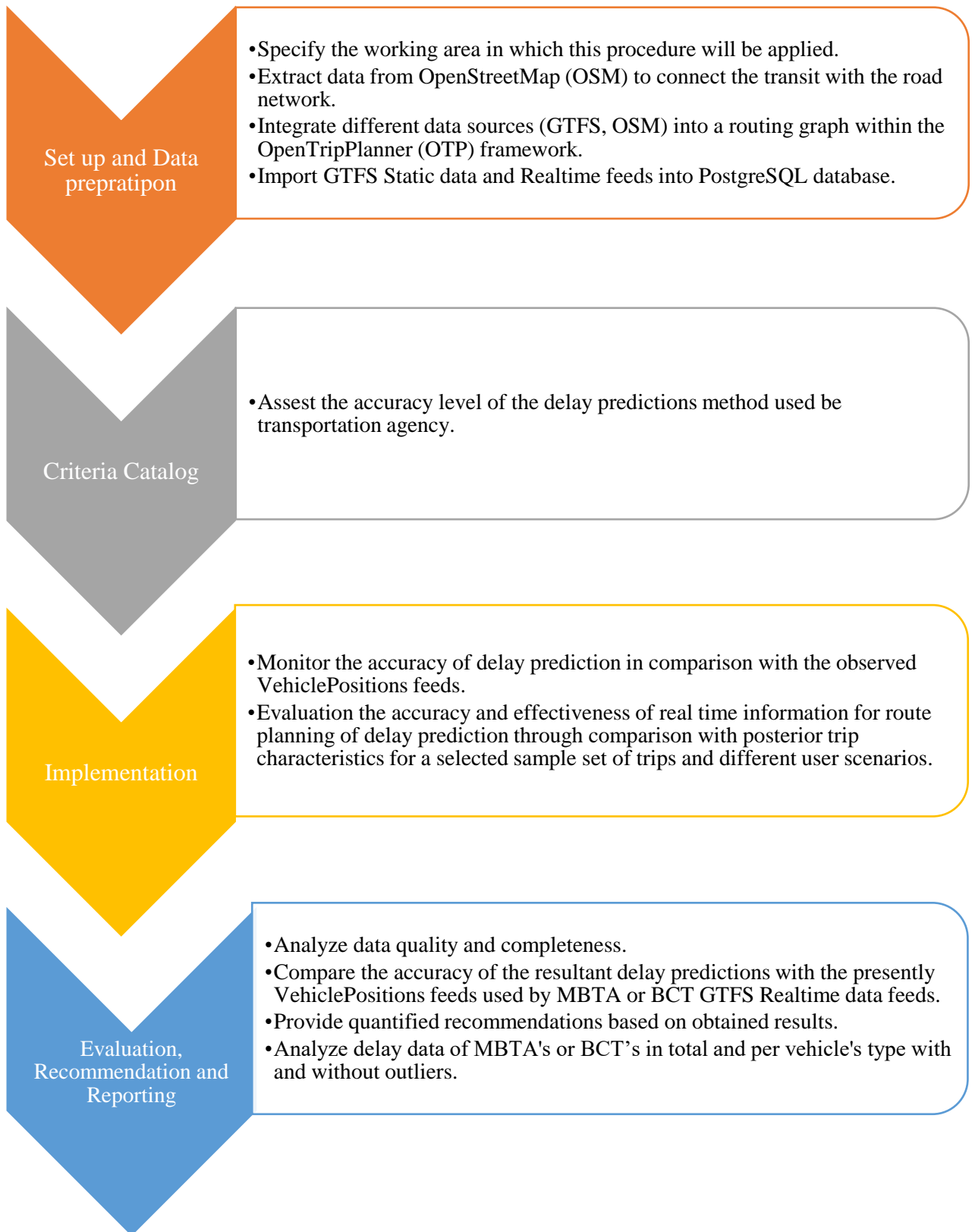


Figure 1 Workflow

1.7 Relevance and expected results

Due to the improvement of IT infrastructure, geo-positioning methods, and communication technology, public transit agencies nowadays have the technical means to share current and predicted delay information of their transit vehicles in Realtime through online feeds. This could improve the accuracy of transport applications, such as trip planners, and thus make public transit more competitive compared to other means of transportation.

The expected results of this research project are:

- ❖ Integration of different data sources (GTFS, OSM) into a routing graph within the OTP framework.
- ❖ Evaluation of the accuracy of the public transport delay prediction through comparison with a posteriori trip characteristics (e.g. the observed vehicle positions at a later point of time) for a selected set sample set of trips (e.g. metro and bus trips for selected lines) and different user scenarios (e.g. have or not have access to a smartphone during a trip).
- ❖ Exploration of the feasibility to apply existing bus arrival prediction methods, e.g. artificial neural networks on public transport Realtime feeds and, if feasible, compare these results with the accuracy of the presently used GTFS Realtime data feeds.
- ❖ The potential combination of a public transport Realtime information system and incident management system as Realtime data feeds currently do not have the ability to remove schedule times for buses that have been suspended e.g. due to mechanical or other issues.

2 State of the Art and Literature review

2.1 Public transportation

Oxford Learner's Dictionaries explained the system of buses, trains, etc. provided by the government or by companies, which people use to travel from one place to another. ('Oxford Advanced Learner's Dictionary' 2020)

Mass transit, also called mass transportation, or public transportation, the movement of people within urban areas using group travel technologies such as buses and trains. The essential feature of mass transportation is that many people are carried in the same vehicle (e.g., buses) or the collection of attached vehicles (trains). This makes it possible to move people in the same travel corridor with greater efficiency, which can lead to lower costs to carry each person or—because the costs are shared by many people—the opportunity to spend more money to provide better service or both.

Mass transit systems may be owned by private, profit-making companies or by governments or quasi-government agencies that may not operate for profit. Whether public or private, many mass transportation services are subsidized because they cannot cover all their costs from fares charged to their riders. Such subsidies assure the availability of mass transit, which contributes to making cities efficient and desirable places in which to live. The importance of mass transportation in supporting urban life differs among cities, depending largely on the role played by its chief competitor, the private automobile.

People travel to meet their needs for subsistence (to go to work, to acquire food and essential services), for personal development (to go to school and cultural facilities), and for entertainment (to participate in or watch sporting events, to visit friends). The need for travel is derived because people rarely travel for the sake of travel itself; they travel to meet the primary needs of daily life. Mobility is an essential feature of urban life, for it defines the ability to participate in modern society.

Travelers make rational choices of the modes they use, each choosing the one that serves him or her best, although best may be viewed differently by each traveler. Transportation services in a city define the alternatives from which travelers must choose, the activities

available to them, and the places to which they can go. The transportation available to an individual is the collective result of government policies, the overall demand for travel in the region, competition among different modes, and the resources available to each individual to buy services. Urban transportation services directly affect the character and quality of urban life, which can differ among individuals who have access to different kinds and amounts of transportation services. ('Mass transit' 2017)

2.1.1 Schedule timetable

A schedule timetable shows the times at which public transportation modes such as railroad trains, airplanes, buses, etc., arrive and depart ('Dictionary.com' 2020). It comes as a printed or electronic document. This information can help people to get knowledge about arrival and departure times from different stations to finally plan a trip. The document may consist of several movements on a particular trip, as well as latencies between trips. Timetables are available in different representations. The most common format is the Matrix format, where you can see the stations in the rows, and service times in the columns. Figure 2 shows a timetable of bus transportation from Vienna, Austria Line 8A Künigberg / ORF-Zentrum - Meidling station, Eichenstrasse.

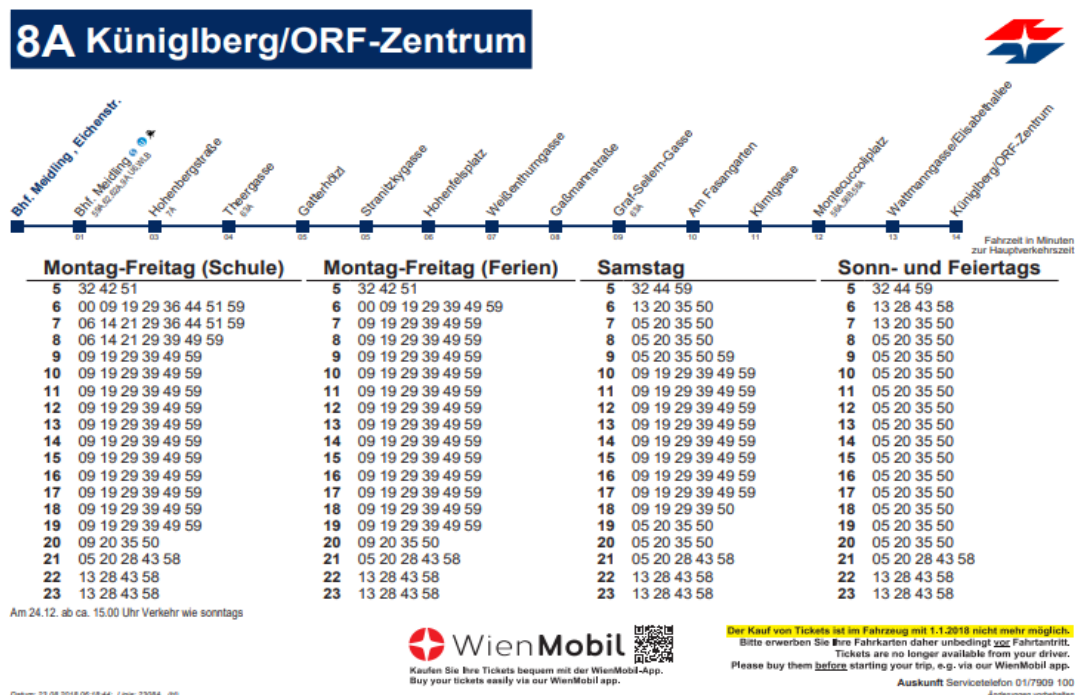


Figure 2 Timetable of bus transportation from Vienna Airport (Schwechat) Platform 4 to Bratislava Main Bus Station (Bratislava AS) ("FEBS Workshop," 2013).

Figure 3 shows another representation of the ÖBB train timetable of a tour from Hermagor, Austria train station to Villach, Austria train station, and every stop in between.

S-Bahn ÖBB															
ÖBB S4 Hermagor → Villach															
	SO-SO	SO-SO	MO-FR	SO-SO	MO-FR	SO-SO	MO-FR	SO-SO	MO-FR	SO-SO	MO-FR	SO-SO	MO-FR	SO-SO	MO-FR
	4802	4806	4808*	4810	4812*	4814	4816*	4818	4820*	4822	4824*	4826	4828*	4830	4832
Hermagor	05:42	07:42	08:42	09:42	10:42	11:42	12:42	13:42	14:42	15:42	16:42	17:42	18:42	19:42	20:42
Vellach-Khünburg	05:45	07:45	08:45	09:45	10:45	11:45	12:45	13:45	14:45	15:45	16:45	17:45	18:45	19:45	20:45
Pressegger See	05:48	07:48	08:48	09:48	10:48	11:48	12:48	13:48	14:48	15:48	16:48	17:48	18:48	19:48	20:48
Görtschach-Förolach	05:53	07:53	08:53	09:53	10:53	11:53	12:53	13:53	14:53	15:53	16:53	17:53	18:53	19:53	20:53
St.Stefan-Vorderberg	06:01	08:01	09:01	10:01	11:01	12:01	13:01	14:01	15:01	16:01	17:01	18:01	19:01	20:01	21:01
Emmersdorf i. G.	06:05	08:05	09:05	10:05	11:05	12:05	13:05	14:05	15:05	16:05	17:05	18:05	19:05	20:05	21:05
Nötsch	06:11	08:11	09:11	10:11	11:11	12:11	13:11	14:11	15:11	16:11	17:11	18:11	19:11	20:11	21:11
Arnoldstein	06:23	08:23	09:23	10:23	11:23	12:23	13:23	14:23	15:23	16:23	17:23	18:23	19:23	20:23	21:23
Pöckau	06:26	08:26	09:26	10:26	11:26	12:26	13:26	14:26	15:26	16:26	17:26	18:26	19:26	20:26	21:26
Neuhaus a.d. Gail	06:29	08:29	09:29	10:29	11:29	12:29	13:29	14:29	15:29	16:29	17:29	18:29	19:29	20:29	21:29
Fürnitz	06:33	08:33	09:33	10:33	11:33	12:33	13:33	14:33	15:33	16:33	17:33	18:33	19:33	20:33	21:33
Villach Warmbad	06:37	08:37	09:37	10:37	11:37	12:37	13:37	14:37	15:37	16:37	17:37	18:37	19:37	20:37	21:37
Villach West	06:40	08:40	09:40	10:40	11:40	12:40	13:40	14:40	15:40	16:40	17:40	18:40	19:40	20:40	21:40
Villach HBF	06:44	08:44	09:44	10:44	11:44	12:44	13:44	14:44	15:44	16:44	17:44	18:44	19:44	20:44	21:44

* MO - FR: Werktag außer Samstag

Figure 3 Timetable of a tour from Hermagor to Villach train station and every stop in between (“B&B Sonntagsho,” 2017)

Timetables can have different forms such as printed books, folders, posters, visualized in homepages, mobile apps, electronic boards in central stations, published as Open Data (OD), and even sent by Short Message Service (SMS) (Steiner *et al.* 2015).

2.1.2 Open Data

Open data is data that can be freely used, re-used, and redistributed by anyone - subject only, at most, to the requirement to attribute and share alike (‘Open Data Handbook’ 2019).

The most important characteristics of Open Data are:

- **Availability and Access:** the data must be available as a whole and at no more than a reasonable reproduction cost, preferably by downloading over the internet. The data must also be available in a convenient and modifiable form.
- **Re-use and Redistribution:** the data must be provided under terms that allow re-use and redistribution including the intermixing with other datasets.

- **Universal Participation:** everyone must be able to use, re-use, and redistribute - there should be no discrimination against fields of endeavor or persons or groups. For example, ‘non-commercial’ restrictions that would prevent ‘commercial’ use, or restrictions of use for certain purposes (e.g. only in education), are not allowed.

It is so important to be clear about what open means and why this definition is used, there is a simple definition: *interoperability*.

Interoperability means the ability of diverse systems and organizations to work together (inter-operate). In this case, it is the ability to interoperate - or intermix - different datasets.

Interoperability is important because it allows for different components to work together. This ability to componentize and to ‘plug together’ components is essential to building large, complex systems. Without interoperability, this becomes near impossible — as evidenced in the most famous myth of the Tower of Babel where the (in)ability to communicate (to interoperate) resulted in the complete breakdown of the tower-building effort (‘Open Data Handbook’ 2019).

Organizations should give access to their internal data in a usable format, which allows both interested individuals and application programmers to benefit from it. Giving this access generate better communications between transportation organizations and their customers, causing improved services and travel experience. The profits of opening up data include better efficient travel (with an enhanced ability to find optimal routes while on the go), a greater understanding of investment (serving to possibly promote enhanced funding), and crowdsourced analysis capabilities (potentially helping detect schedule improvements or errors in stop locations/names, for instance). The data, which would normally consist of sets as schedules, routes, budgetary information, ridership numbers, traffic numbers, and road conditions, should be published in both historical and real-time for both analysis and prediction. The data must be released similar to its original format, except for security-sensitive data. Open data includes sharing information for augmented travel, management, and future improvements (Kaufman 2012). Figure 4 shows the advantages of open data in the transportation context.

Typical Transportation Data Benefits

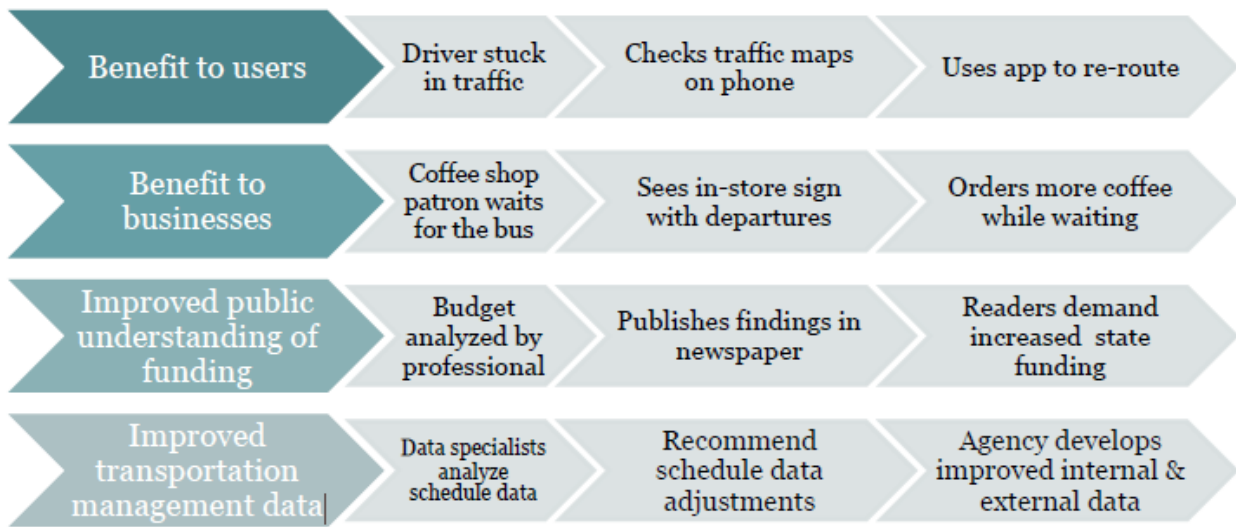


Figure 4 Transportation Open Data Benefits (Kaufman, 2012)

2.1.2.1 Costs of Opening Data

Open data use existing internal data, hence the costs of generating it are modest.

However, there are costs, such as:

- Converting data to mainstream formats.
- Web service for hosting data.
- Personnel time to update and maintain data as needed.
- Personnel time to liaise with data users.

These costs vary, depending upon the size and scale of the transportation services covered. It should be noted that these costs are lower than any internal attempt to create applications internally; as written by the *Boston Globe* about the Massachusetts Department of Transportation's data releases, "This approach is a smart 21st-century alternative to hiring some consultant who develops inelegant software at exorbitant costs." ('MBTA: App judgment' 2020). It should be noted that the costs involved should not prevent openness: for public transportation agencies, data was developed using public funds and should be accessible for

public use and analysis. The returns on these investments are manifold, reaching large numbers of people with modest effort.

The benefits of improved travel outweigh the maintenance costs; organizations must think holistically about their missions, like providing mobility resources and the tools to use them, rather than by departmental line budgets, to embrace data openness fully (Kaufman 2012).

2.1.2.2 Open Data in Use

Transportation data is used worldwide for policy analysis and smartphone applications. Currently, there are 1280 agencies in 674 locations worldwide providing GTFS data ('OpenMobilityData - Public transit feeds from around the world' 2020). Of the half-million applications in the Apple App Store, many thousands are transit related. Realtime transit data is available in dozens of locations via the NextBus system, and others via Google's new standard in four US cities (Boston, Portland, OR, San Diego, and San Francisco). In a recent Transit Cooperative Research Program study, of 28 transit agencies surveyed, 13 relied on third-party applications to disseminate real-time information, but only three agencies relied on internally-developed applications (Transit Cooperative Research Program *et al.* 2011). Those 13 agencies have embraced the potential of open data to inform customers through any medium; they cite the benefits of releasing data as:

- Free development of mobile applications.
- Increased ridership.
- Improved customer service.
- Time saved by agencies in developing customized applications.
- More accurate applications.
- Positive image for agencies (Kaufman 2012).

Also, using the data, developers can create applications for users in foreign languages, for those with vision disabilities, and highly localized information for particular neighborhoods. Table 1 shows the number of applications and ridership in several U.S. cities in 2012 (Kaufman 2012).

Table 1 The number of applications and ridership in several U.S. cities (“Transit transparency,” 2012)






City	Portland	Boston	Chicago	Washington	New York
Agency					
Av. Weekday ridership (2011)	325,400	1,292,000	1,717,200	1,446,400	10,287,600
Number of apps (2011)	44	47	22	11	58
Ratio (app/riders)	1/7,000	1/27,000	1/78,000	1/131,000	1/177,000
Developer relationship	strong	strong	medium	weak	weak
Marketing push	light	light	medium	light	heavy
Data release	2006/2007	2009	2009/2010	2010	2011/2012

Table 2 shows the latest report of the American Public Transportation Association (APTA) which shows the number of Ridership by Mode and Quarter from 2016 to Present in U.S. cities.

Table 2 Ridership by Mode and Quarter 2016 Present (APTAdmin, 2020)

Quarter	Year	Total Ridership (000s)	Heavy Rail (000s)	Light Rail (000s)	Commuter Rail (000s)	Trolleybus (000s)	Bus (000s)	Demand Response (000s)	Other (000s)
2016	Q1	2,581,519	954,021	129,998	118,443	23,736	1,260,104	52,262	42,956
	Q2	2,644,957	1,000,979	139,745	126,086	22,055	1,253,289	53,673	49,130
	Q3	2,598,388	968,259	140,828	126,937	21,454	1,234,299	52,858	53,752
	Q4	2,583,028	970,509	137,435	125,573	20,540	1,231,932	52,039	45,000
2017	Q1	2,496,395	931,040	133,650	120,348	20,325	1,198,305	51,036	41,690
	Q2	2,569,443	987,582	137,555	125,486	21,242	1,195,562	52,607	49,409
	Q3	2,498,636	939,136	138,167	125,717	20,956	1,169,820	52,001	52,840
	Q4	2,526,237	955,801	132,902	125,522	20,507	1,194,916	51,809	44,780
2018	Q1	2,415,212	901,386	126,690	119,895	19,919	1,155,076	50,825	41,421
	Q2	2,525,653	956,916	133,600	126,224	20,354	1,185,691	53,789	49,079
	Q3	2,457,352	916,002	134,175	126,603	20,300	1,154,284	52,905	53,084
	Q4	2,508,574	938,141	132,516	126,360	20,498	1,191,954	53,395	45,710
2019	Q1	2,365,284	871,380	123,555	122,422	20,133	1,135,243	51,854	40,697
	Q2	2,526,794	970,735	127,948	130,688	20,329	1,174,619	54,035	48,441
	Q3	2,511,387	965,979	127,082	132,147	20,554	1,161,052	54,088	50,484

Note: Ridership numbers are readjusted in each quarter based on the new Fact Book ridership number and data changes and updates from agencies.

There are two barriers to providing transportation data to the public: firstly, the collection and maintenance of these data are very difficult and costly, and secondly, it is hard to create a commercial business out of it because no one will pay for such service (Lyoen *et al.* 2010). Most transit information is currently locked into proprietary formats and systems and cannot be

easily shared, viewed, updated, or co-mingled without permission from the vendor and expert data analysis. This limits the ability of transit agencies and others to provide information resources, such as trip planning tools, that support the use of transit or other alternatives (Hillsman and Barbeau 2011).

Nevertheless, to make aware of communication between different transport systems, standards have become more and more important. (Soares and Martins 2013) described transport systems as: “Distributed systems with very complex information requirements. So, the full interoperability of these systems can only be achieved through the existence and implementation of adequate standards, properly conceived by experts, tested, and understood by practitioners. A strong standard for transportation data is important for the safe and efficient operation of the systems. In the last decades, several international projects and international standardization bodies have gathered efforts to develop a set of basic standards and procedures to ensure the interoperability of public transport, enabling the effective sharing of information between the different transport systems.”

2.1.3 Public transportation standards

Error! Reference source not found. describes common data standards and file formats used by transportation agencies for different purposes.

Table 3 Table 3 Common data standards and file formats used by transportation agencies for different purposes

	Champion	Where it's used	Applicable data sets	Examples	More information ¹¹
Data Standards					
GTFS	Google	Worldwide	Schedule data	Train line schedule	https://developers.google.com/transit/gtfs/
GTFS-realtime	Google	Select US & European cities	Real-time data	“Train arriving in 3 min”	https://developers.google.com/transit/gtfs-realtime/
SIRI	European Committee for Standardization	European cities	Real-time data	“Train arriving in 3 min”	http://bustime.mta.info/wiki/Developers/SIRIIntro
TransXchange	UK Gov	UK Buses	Bus schedules & data	Bus route schedule	http://www.dft.gov.uk/transxchange/
DATEX 2	European Commission	European Cities	Traffic data & Management	Delays on Route 4	http://www.datex2.eu/content/datex-background
File Formats					
CSV	Many	Worldwide	Data tables	Historic on-time data	http://www.ehow.com/how_5091077_us_e_csv_files.html
TXT	Many	Worldwide	Text	Textual information	http://en.wikipedia.org/wiki/Text_file
GIS	Many	Worldwide	Geographic mapping	Subway station entrances	http://en.wikipedia.org/wiki/GIS_file_formats
KML	Google	Worldwide	Google Maps & Earth	GIS road outlines	https://developers.google.com/kml/documentation/
XML	Many	Worldwide	Large data sets	Traffic numbers	http://www.w3schools.com/xml/xml_what_is.asp

Note that files in the formats above are used both within the data standards (for example, a GTFS data set is a series of specific TXT files) and on their own (to convey text or information separately, like budgetary information in a CSV file). It's essential to choose a file format that is both easily convertible within your transportation agency and useful to the majority of readers and data developers (that includes converting Microsoft Excel spreadsheets to CSV, and Word documents to TXT (Kaufman 2012).

2.1.4 Usage of GTFS in public transit

The General Transit Feed Specification, or GTFS, has become the most popular world-wide data format to describe fixed-route transit services. Many transit agencies have created and published GTFS data with the primary purpose being integration with Google Maps. However, GTFS data can power many other different types of transit and multimodal software applications, including multimodal trip planning, timetable creation, mobile apps, visualization, accessibility, analysis tools for planning, real-time information, and interactive voice response (IVR) (Antrim and Barbeau 2013).

2.1.4.1 Overview of GTFS

GTFS represents a fixed-route schedule, route, and bus stop data in a series of comma-delimited text files compressed into a ZIP file. Figure 5 shows the contents of a GTFS ZIP file from the Boston Metropolitan Area (Massachusetts Bay Transportation Authority MBTA) and the contents of the stops.txt file within it which contains information about the name, ID, and location of every stop.

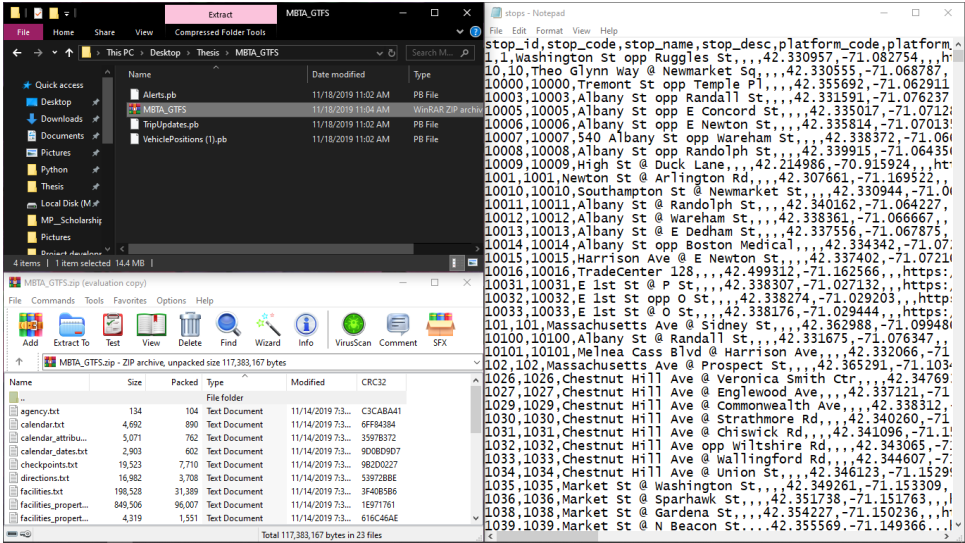


Figure 5 A GTFS dataset from MBTA transit agency consists of several text files within a ZIP file

The routes.txt file contains information about the transit agency's routes, both calendar.txt, and calendar_dates.txt files contain schedule information. Information about the order of visitation of bus stops for a particular route according to a particular schedule is provided in the trips.txt and stop_times.txt files. Spatial representation of a route alignment to be accurately drawn on a map is included in the shapes.txt file (Antrim and Barbeau 2013).

2.1.4.2 Overview of Current GTFS Components

GTFS is used for the static exchange of schedule data and public transport stops. Each version of schedule data is sent as a set of CSV tables, encapsulated as a zip file. Table 4 shows the CSV tables which make up the GTFS.

Table 4 Google Transit Feed Specification tables (Kizoom and Miller, 2008)

Google File		Contents	CEN Transmodel Concept
agency.txt	Required	Information about the transit agency.	OPERATOR, AUTHORITY
stops.txt	Required	Information about individual locations where vehicles pick up or drop off passengers.	SCHEDULED STOP, POINT, STOP PLACE (IFOPT), TARIFF ZONE
routes.txt	Required	Information about a transit organization's routes. A route is a group of trips that are displayed to the rider as a single service.	LINE
trips.txt	Required	Information about scheduled service along a particular route. Trips consist of two or more stops that are made at regularly scheduled intervals.	VEHICLE JOURNEY
stop_times.txt	Required	Lists the times that a vehicle arrives at and departs from individual stops for each trip along a route.	Call, PASSING TIME, STOP POINT IN JOURNEY, PATTERN (SERVICE PATTERN), (ROUTE LINK - distance)
calendar.txt	Required	Defines service categories. Each category indicates the days that service starts and ends as well as the days that service is available.	DAY TYPE PERIOD, DAY OF WEEK
calendar_dates.txt	Optional	Lists exceptions for the service categories defined in the calendar.txt file.	OPERATING DAY
fare_attributes.txt	Optional	Defines fare information for a transit organization's routes.	FARE ELEMENT PRICE
fare_rules.txt	Optional	defines the rules for applying fare information for a transit organization's	FARE ELEMENT, DISTANCE MATRIX
shapes.txt	Optional	Provides rules for drawing lines on a map to represent a transit organization's routes.	ROUTE, ROUTE LINK / PROJECTION
frequencies.txt	Optional	Provides the headway (time between trips) for routes with a variable frequency of service.	(Frequency)
transfers	Optional	Provides additional rules for making connections between routes.	CONNECTION LINK, SERVICE JOURNEY INTERCHANGE, SERVICE JOURNEY PATTERN INTERCHANGE, DEFAULT INTERCHANGE

2.1.4.3 Applications Based on GTFS

Different types of applications based on GTFS are accessible when an agency creates a GTFS feed and share it with the public.

Following is an overview of these different types of applications:

- Trip planning and maps: applications that assist a transit customer in planning a trip from one location to another using public transportation.
- Ridesharing: applications that assist people in connecting with potential ridesharing matches.
- Timetable creation: create a printed list of the agency’s schedule in a timetable format.
- Mobile applications: applications for mobile devices that provide transit information.
- Data visualization: applications that provide graphic visualizations of transit routes, stops, and schedule data.
- Accessibility: applications that assist transit riders with disabilities in using public transportation.
- Planning analysis: applications that assist transit professionals in assessing the current or planned transit network.
- Interactive Voice Response (IVR): applications that provide transit information over the phone via an automated speech recognition system.
- Real-time transit information: applications that use GTFS data along with a real-time information source to provide estimated arrival information to transit riders (Antrim and Barbeau 2013).

2.1.5 GTFS-Realtime

“GTFS-Realtime is a feed specification that allows public transportation agencies to provide realtime updates about their fleet to application developers. It is an extension to GTFS (General Transit Feed Specification), an open data format for public transportation schedules and associated geographic information. GTFS-Realtime was designed around ease of implementation, good GTFS interoperability, and a focus on passenger information” (‘GTFS Realtime Overview | Realtime Transit’ 2020).

The initial Live Transit Updates partner agencies, several transit developers, and Google designed the specification through a partnership. The specification is published under the Apache 2.0 License ('GTFS Realtime Overview | Realtime Transit' 2020).

The following types of information are currently supported by the specification:

- **Trip updates** (delays, cancellations, changed routes):

"Bus X is delayed by 5 minutes"

Trip updates stand for fluctuations in the timetable. These updates provide a predicted arrival or departure for stops along the route. Trip updates can also support more complex scenarios where trips are canceled, added to the schedule, or even re-routed.

- **Service alerts** (stop moved, unforeseen events affecting a station, route, or the entire network)

"Station Y is closed due to construction"

Service alerts stand for higher-level problems with a particular entity and are generally in the form of a textual description of the disruption.

They could be problems with:

- Stations
- Lines
- The whole network
- etc.

A service alert usually consists of some text which describes the problem, which also allows for URLs for more information as well as more structured information to help to understand who will be affected by this service alert.

- **VehiclePositions** (information about the vehicles including location and congestion level)

"This bus is at position X at time Y"

VehiclePositions stands for a few basic pieces of information about a particular vehicle on the network.

Most important are the latitude and longitude the vehicle is at, but data on current speed and odometer readings from the vehicle can also be used.

A feed may, although not required to, combine entities of different types. Feeds are frequently updated and served via HTTP. The file is in a regular binary format, so it can be hosted and served by any type of web server (other transfer protocols might be used as well). Alternatively, a response to a valid HTTP GET request, web application servers could also be used to return the feed. There are no constraints on how frequently nor on the exact method of how the feed should be updated or retrieved.

Because GTFS Realtime allows presenting the actual status of an agency's fleet, the feed needs to be updated regularly - preferably whenever new data comes in from an agency's Automatic Vehicle Location system.

2.1.5.1 Data format

The GTFS Realtime data exchange format is based on Protocol Buffers.

Protocol Buffers (Protobuf) are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data – think XML, but smaller, faster, and simpler. The data structure is defined in a [gtfs-Realtime .proto](#) file, which is then used to generate source code to easily read and write structured data from and to a variety of data streams, using a variety of languages such as Java, C++, or Python.

Figure 6 shows how is Protocol Buffers used to share data across languages.

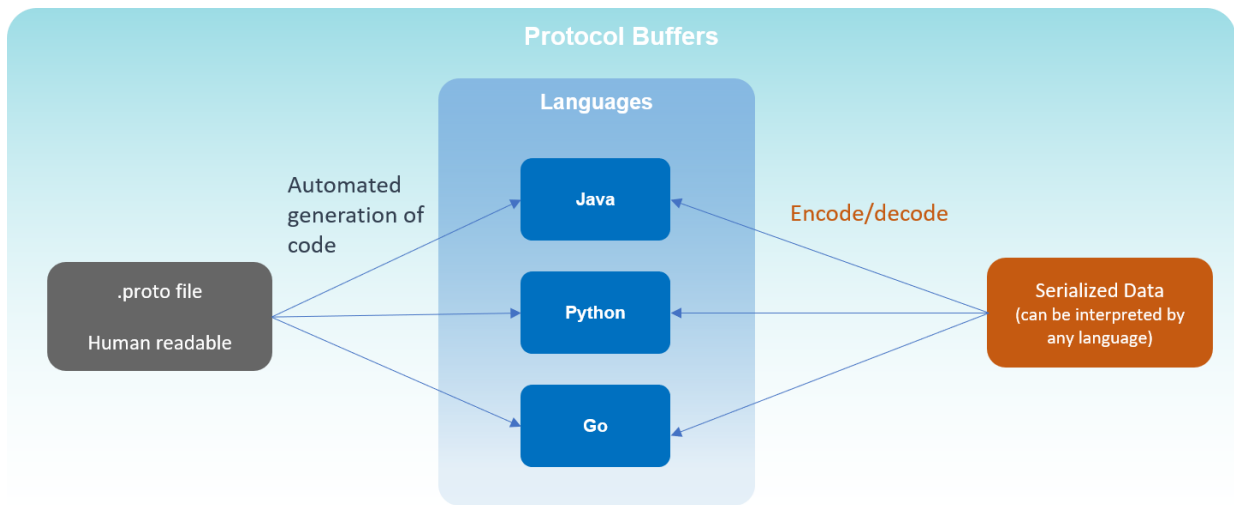


Figure 6 how is Protocol Buffers used to share data across languages (Masina 2019, p.101)

Figure 7 shows the Protocol Buffer compiler and how it auto-generates code to exchange binary GTFS-Realtime messages.

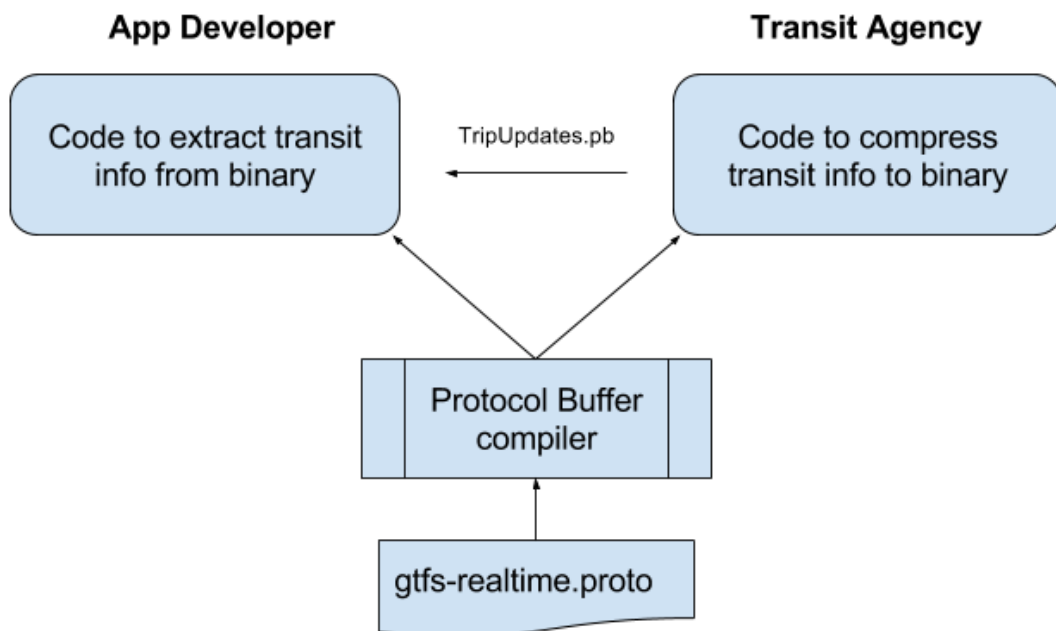


Figure 7 The Protocol Buffer compiler auto-generates code to exchange binary GTFS-Realtime messages

2.1.5.2 Data structure

The hierarchy of elements and their type definitions are specified in the [gtfs-Realtime .proto](#) file.

This text file is used to generate the necessary libraries of programming language. These libraries provide the classes and functions needed for generating valid GTFS Realtime feeds. The libraries make feed creation easier and also ensure that only valid feeds are produced ('GTFS Realtime Overview | Realtime Transit' 2020).

2.2 OpenTripPlanner

OpenTripPlanner (OTP) is an open-source multi-modal trip planner, which runs on Linux, Mac, Windows, or potentially any platform with a Java virtual machine. OTP is released under the LGPL license. The code is under active development with a variety of deployments around the world. It is a family of open-source software projects that provide passenger information and transportation network analysis services. The core server-side Java component finds itineraries combining transit, pedestrian, bicycle, and car segments through networks built from widely available, open standard OpenStreetMap and GTFS data. This service can be accessed directly via its web API or using a range of Javascript client libraries, including modern reactive modular components targeting mobile platforms.

Launched in 2009, the project has attracted a thriving community of users and developers, receiving support from public agencies, startups, and transportation consultancies alike. OTP powers regional and national journey planning services around the world, as well as several popular multi-city mobile applications ('OpenTripPlanner' 2020).

OTP consists of three main components of the framework: "Graph Building", "Routing Engine" and "Web Interface" (Figure 8). The "Graph Building" generates an object called "Graph.obj", which is the routing graph. The settings for that process like data source paths and restrictions are included in a configurations file. This graph is used by the "Routing Engine" to calculate a route after a request from the "Web Interface". The route, the map and additional information like route instructions will be visualized in the "Web Interface" when the route is computed as described in (Weyrer *et al.* 2013).

Architecture

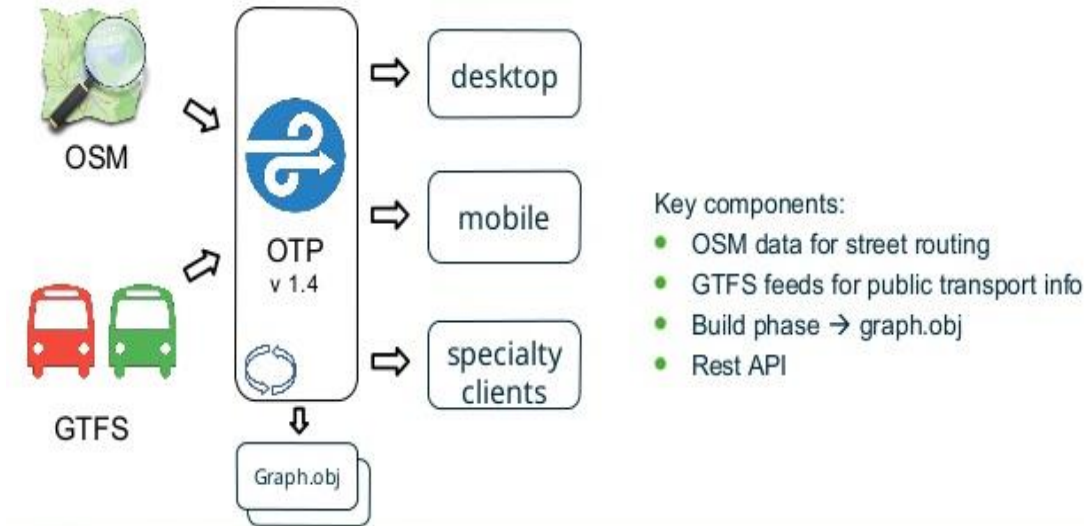


Figure 8 Architecture of the OpenTripPlanner (South Tyrol Free Software Conference 2019a).

2.2.1 Graph structure

In Open Trip Planner, directed graphs are representing all the street and transport networks. A small extract of an OTP graph for New York, containing both street and public transit information (Figure 9). In the OTP ("master branch") version, the model has shifted from a pattern-based graph into an edge-based graph (Figure 10). Based on the directed graph, the street segments consist of two edges, each edge indicates one direction. This specifies all travel modes on one street (e.g. pedestrians in both directions, cars in one-way streets). Each transit stop is linked to a street vertex and has at least one node that is connected to a transit trip. If several platforms exist, or if the station entrances are far away from each other, multiple nodes could be connected to a single transit stop. In such a case, the street edge of the transit trip will be split. The transit trips are represented by the Board, Alight, Dwell, and Hop edges. When different trips have the same sequence of stops, these trips can be grouped to time-dependent PatternBoard, PatternAlight, PatternDwell, and Pattern Hop edges. These edges' weight functions vary according to the time at which they are traversed. For example, a PatternBoard edge will search for the next departure time for its associated trip pattern and vary its weight accordingly. Figure 9 illustrates two different subway services with two different trip patterns to pass through the same stops, but they branched out elsewhere in the city ('GraphStructure' 2015).

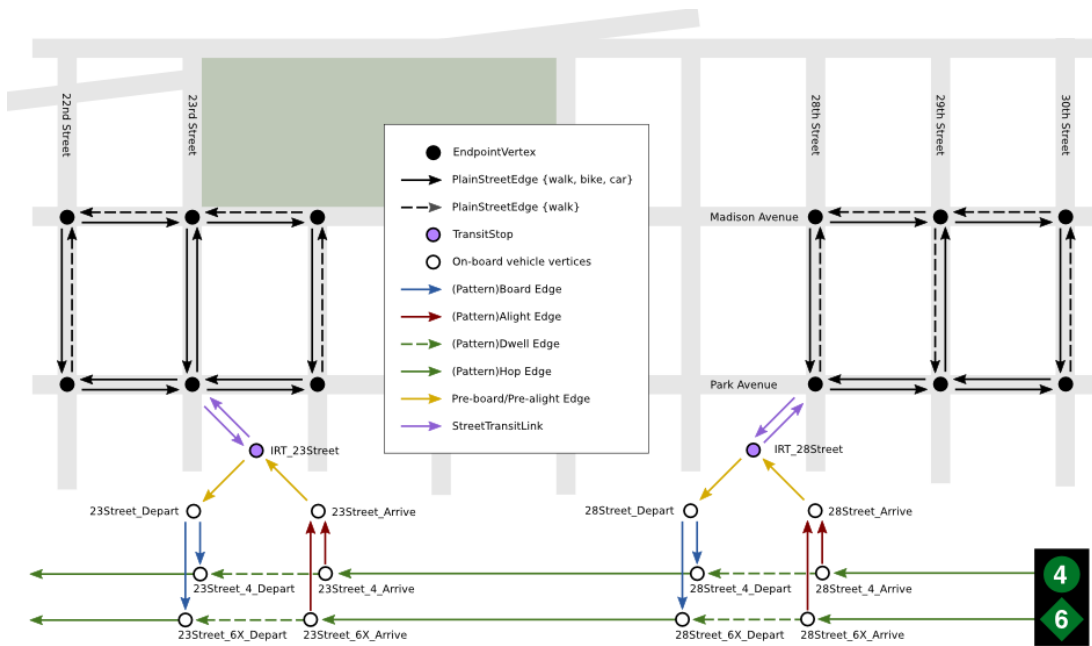


Figure 9 Simple representation of the pattern-based Graph Structure in OTP ('GraphStructure' 2015)

The pattern-based representation from Figure 9 was changed with the edge-based representation in Figure 10 to enable an advanced performance such as turn restrictions. To apply this change, each vertex of the original graph is replaced by an edge and vice versa. A vertex means being on a street segment, where being on an edge means changing the street segment through a road junction. This makes applying turning restrictions to individual turns as well as different travel modes possible. In this case, the computation of the shortest path could loop itself, which needs to be considered ('GraphStructure' 2015).

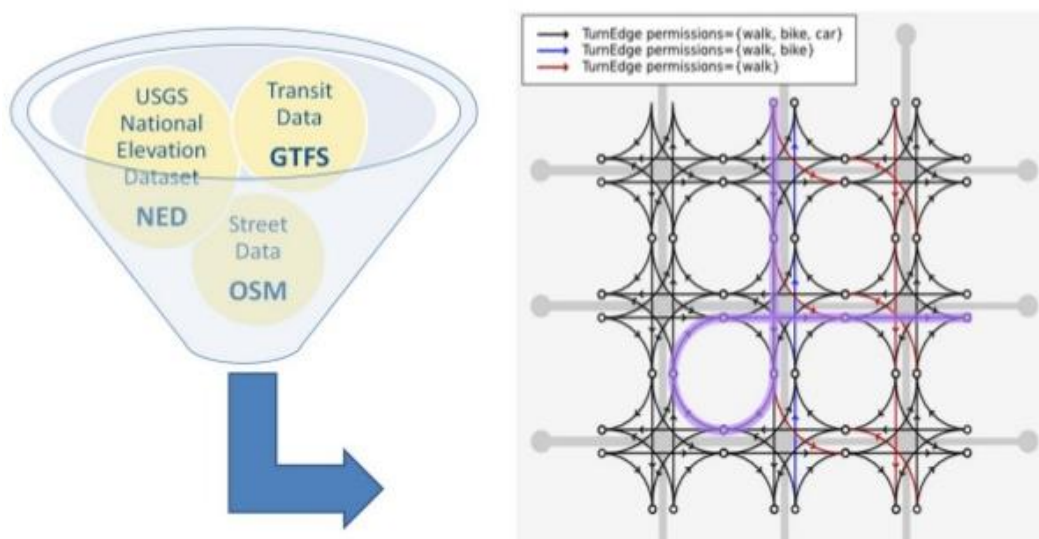


Figure 10 Edge-based representation of the Graph Structure (Paris Open Source Summit, 2012)

2.2.2 Graph creation using shapefile or OSM

The graph builder configuration file describes what data and settings are used to build the graph.

2.2.2.1 Data sources

Many data sources are used by the trip planner: GTFS for transit data, OpenStreetMap or shapefiles for street data, and the National Elevation Dataset for elevation data. Each data source has its configuration options ('GraphBuilder' 2014).

2.2.2.2 Street segments

Streets are divided into segments. A segment is a part of a street between two consecutive intersections or between the intersection and dead end. Consecutive street segments that share attributes can be stored together as one "way" for OSM, which is not a problem. In Opposition, a street segment may be broken at an overpass even if there is no intersection in shapefiles, which requires special handling to ensure that, even when two streets meet at the same point, they don't intersect when one passes over the other('GraphBuilder' 2014).

2.2.2.3 Permissions

Defines types of users who can traverse through a given street segment. For example, to make certain street segments traversable by bikes only in one direction, an attribute can be used. Alternatively, to make certain segments (e.g., highways) inaccessible to pedestrians, a street type attribute could be used. Street traversal permissions are defined in `org.opentripplanner.routing.edgetype.StreetTraversalPermission` (Javadoc) and are:

- NONE
- ALL
- PEDESTRIAN
- BICYCLE
- PEDESTRIAN_AND_BICYCLE
- PEDESTRIAN_AND_CAR
- BICYCLE_AND_CAR
- CAR ('GraphBuilder' 2014)

2.3 Related work in Quality Assessment of Open Realtime Data for Public Transportation in the Netherlands

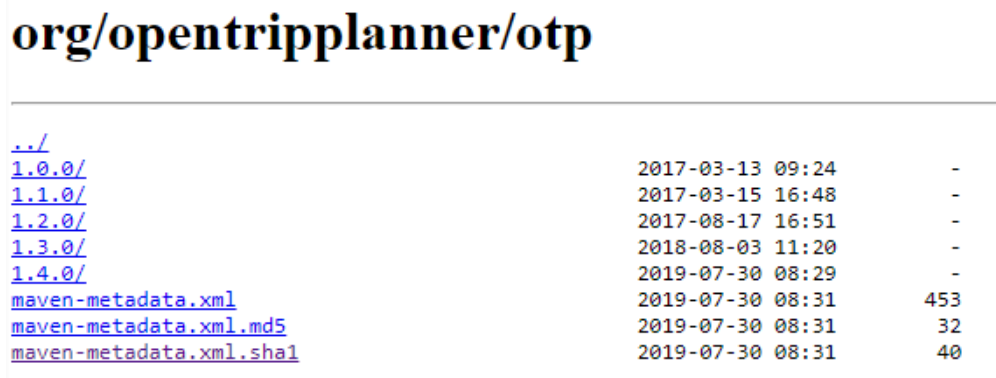
In a previous study done using Realtime feeds with access introduced by OVapi (accessible at <http://gtfs.ovapi.nl/>) for buses from different transit agencies in the Netherlands at the end of January 2014, using the Realtime format (Steiner *et al.* 2015). The data quality of Realtime public transit data provided as GTFS-Realtime feeds for the Netherlands was analyzed. Since OVapi covers a large geographic area and delivers data in GTFS-Realtime feed format, (Steiner *et al.* 2015) focused on the data quality evaluation of the OVapi real-time information and analyzed data completeness and temporal accuracy of the Realtime feeds. OTP was also used to illustrate the effect of GTFS-Realtime information on selected computed bus routes. Trip computations were also published as a Web service to be called within other applications through the API. However, it was not yet possible to download the Realtime information of the entire network at a given time, which was necessary for building customized routing applications. Thus, in Europe at that time only OVapi provided Open Access to GTFS-Realtime data, whereas it was already provided by several US transit agencies, such as BART (Oakland, CA), TriMet (Oregon, WA), or MARTA BUS (Atlanta, GA).

Results showed that the data quality was not satisfactory up to May 2014, and trip delay data and a large percentage of vehicle positions were missing, which was necessary for reliable Realtime trip planning applications. Analyzing user benefits of Realtime information for trip planning was one of the goals, but in addition to data scarcity, this was impossible due to OTP data handling. For instance, some problems like returning sub-optimal routes were caused for scenario 5 during the integration of VehiclePositions data into the OTP. Furthermore, the assessment of usability for smartphone users with repeated computations along the trip (scenarios 3 and 4) required a modification of the OTP code (Steiner *et al.* 2015).

3 Methodology

3.1 Setting up the OTP

OpenTripPlanner is written in Java and distributed as a single runnable JAR file. These JARs are deployed to the Maven Central repository and available in [the OTP directory at Maven Central](#) in several versions (Figure 11). OSMCONVERT can be used to convert and process OpenStreetMap files. It masters fewer functions than the commonly-used Osmosis: for example, there is no way to access a database with OSMCONVERT. (‘Osmconvert - OpenStreetMap Wiki’ 2020)



File Name	Date	Size
./		
1.0.0/	2017-03-13 09:24	-
1.1.0/	2017-03-15 16:48	-
1.2.0/	2017-08-17 16:51	-
1.3.0/	2018-08-03 11:20	-
1.4.0/	2019-07-30 08:29	-
maven-metadata.xml	2019-07-30 08:31	453
maven-metadata.xml.md5	2019-07-30 08:31	32
maven-metadata.xml.sha1	2019-07-30 08:31	40

Figure 11 OTP directory at Maven Center (‘Basic Tutorial - OpenTripPlanner’ 2020)

Different types of data are building a multimodal graph in OTP (Figure 12), afterwards the desired route is calculated using this data besides the route parameters of the user.

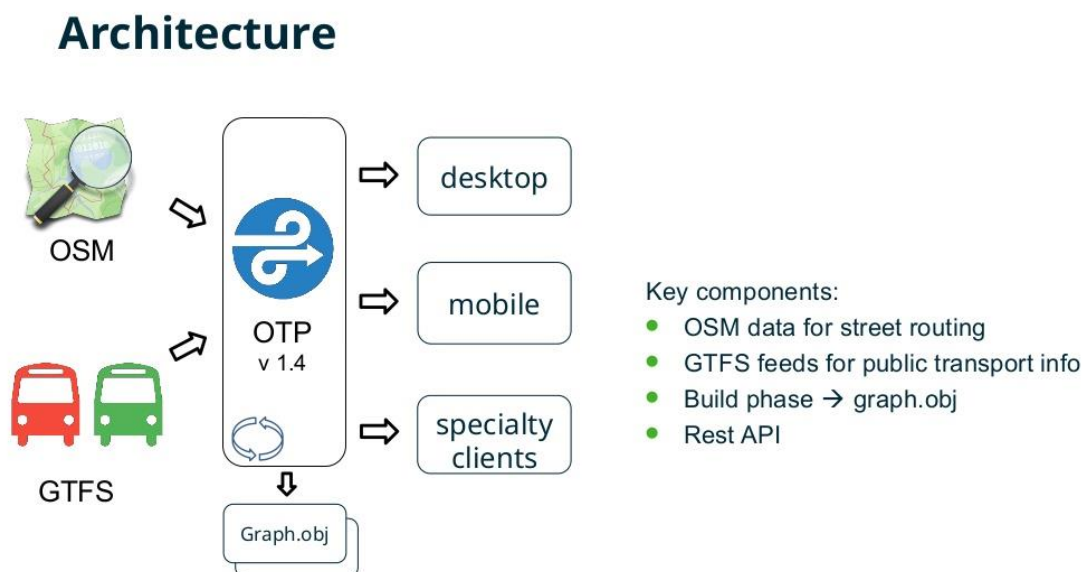


Figure 12 OTP Data Flow Model (South Tyrol Free Software Conference 2019b)

The following figure (Figure 13) illustrates a trip in the OTP interface.

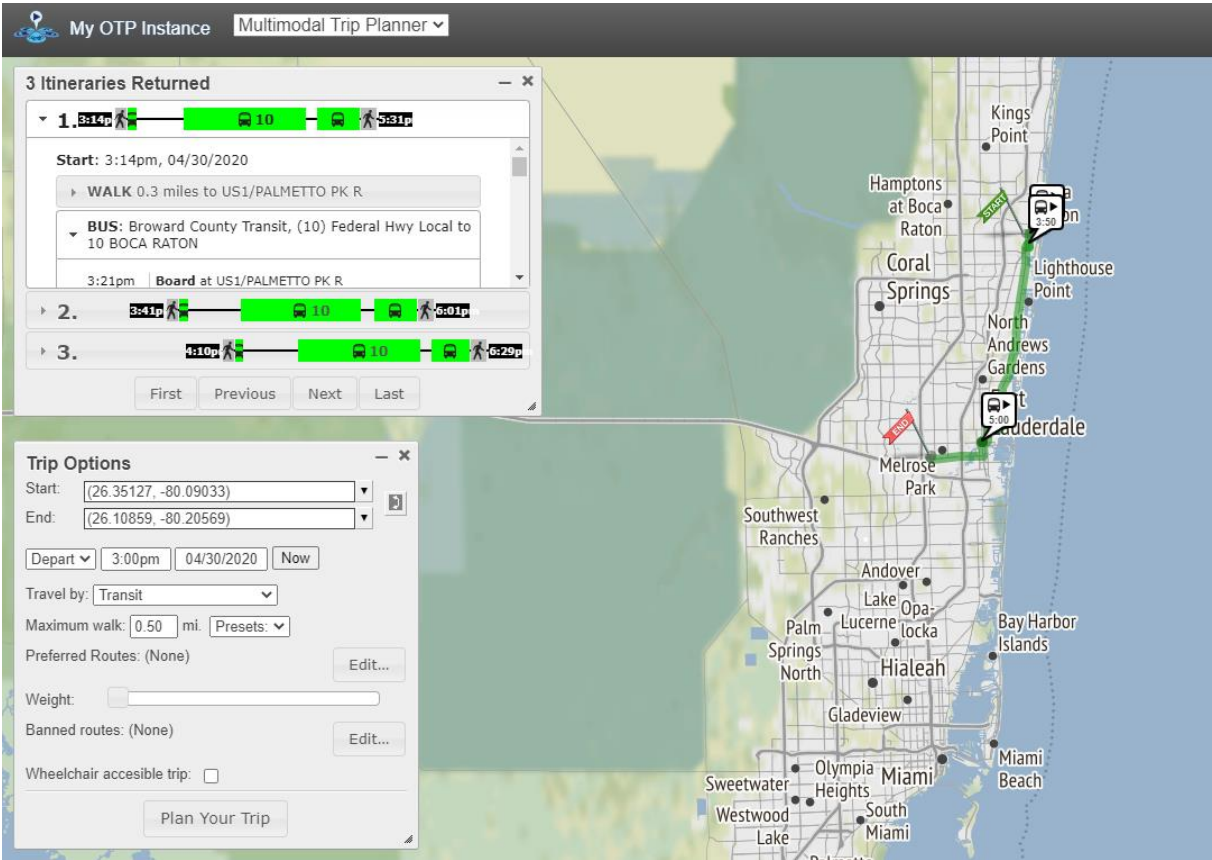


Figure 13 OTP Instance running in Broward 2020

If the public transport agency provides raw timetables instead of a standardized GTFS feed, the timetables should be converted into the GTFS. OSM will be used for the creation of the multimodal graph. Afterward, the integration of a user round trip option into OTP.

3.2 Transit Data

The availability of transit data in the GTFS format should be considered. In case this information is not available in the desired format, this data should be prepared using available schedule information.

As mentioned before, each GTFS feed consists of a set of unique CSV-text files (Figure 14) and compressed in a ZIP package. All these required files must have exactly the column headings described previously to provide the information in a standardized way, the same way as a traditional database.

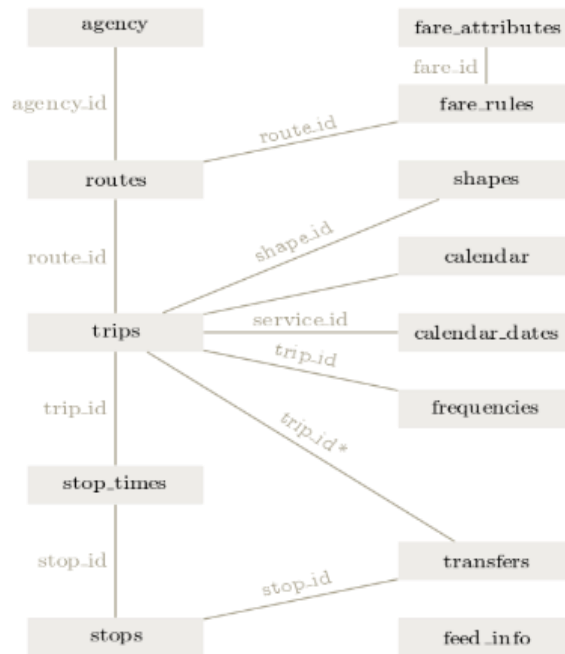


Figure 14: General Transit Feed Specification (GTFS) relations ('Introduction to tidytransit' 2019)

The following figure (Figure 15) demonstrates the General Transit Feed Specification (GTFS) tables relations.

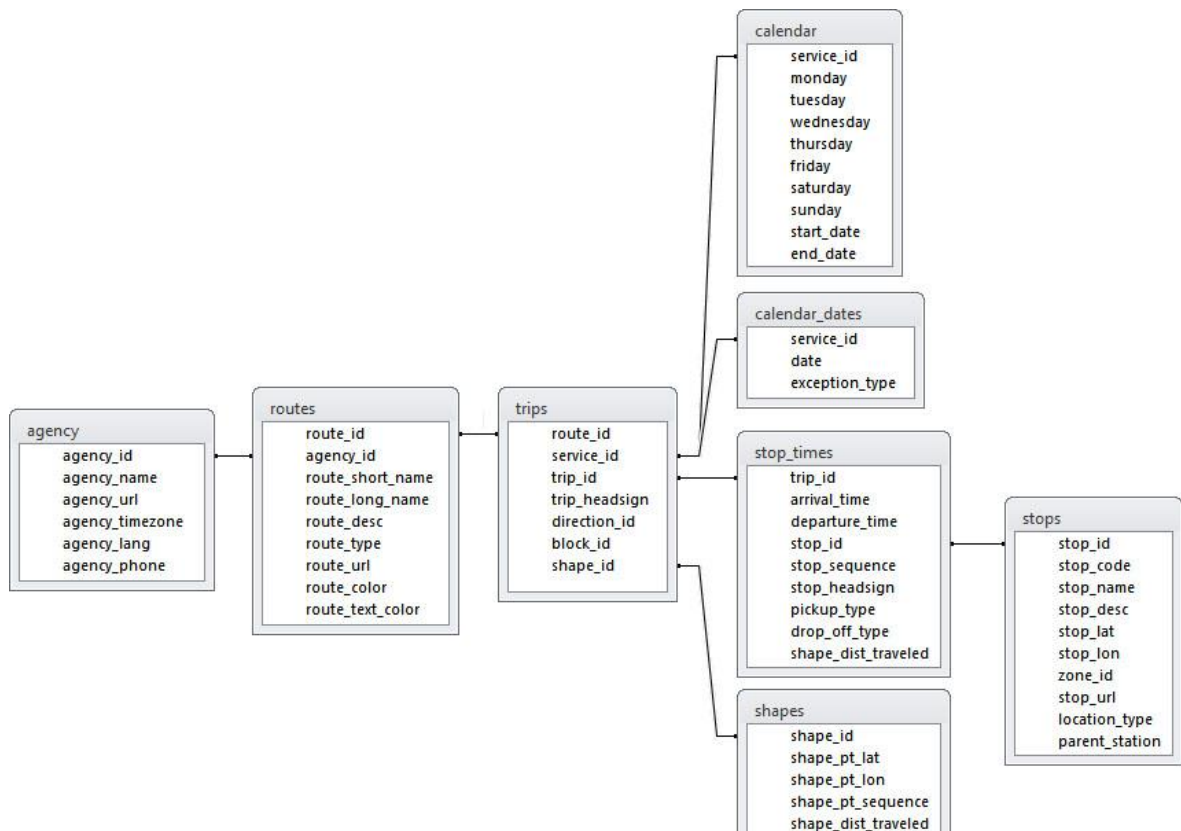


Figure 15 General Transit Feed Specification (GTFS) table relations ('MARTA Developer Resources' 2019)

3.3 Realtime Data Impacts Evaluation

Public transportation systems' real-time data and their impact on the trip planning system user will be analyzed. This will be done using the OpenTripPlanner (OTP) software as it is capable of all further measures and analysis.

3.3.1 Realtime types and structure

As described before GTFS-Realtime specification has three different feed types that are published as protocol buffers. Protocol buffers are a flexible, efficient, automated mechanism for serializing structured data. The information is encoded into a binary format, which makes it easy to read and write information in different programming languages ('Developer Guide | Protocol Buffers' 2020).

3.3.2 Evaluation of Realtime information

Collected feeds should be decoded and stored in a central spatial database. The original GTFS feed should be imported to the central spatial database as well. To create a modified GTFS feed that includes collected timestamps the recorded real-time positions could be used. Again, OTP can handle both the original and modified GTFS feed, with and without TripUpdates delay information, for analyzing differences in route times. Moreover, to obtain variations, the predicted and observed delays could be compared. Following five different scenarios are defined, to show differences in traveling times of different passenger types.

3.3.3 Definition and concept of scenarios

Five scenarios of a single passenger who wants to move from point A to point B were defined for trip planning and traveling simulation for this research:

1. Print trip on paper, without consideration of trip updates:

In this scenario, it is assumed that the user will have a printed route and will follow exactly the written instructions since he is not aware of the area. A user trip will not include any trip updates. According to this, delay information will not be considered in the computation. So, if the user missed the planned bus, s\he must wait for the following one that departs at the same route described on the plan.

2. Print trip on paper, with consideration of trip updates:

This scenario follows the same behavior as the first one, except the trip updates are considered in the initial calculation. Therefore, during the route calculation stage, future delay estimates are integrated, which could have an impact on the traveling time.

3. A smartphone without Realtime route computation:

It is assumed that the passenger uses a smartphone for the initial route calculation in the third scenario. The user follows the route displayed on the screen until s\he misses one bus because of unexpected delays. In this situation, the passenger uses the smartphone to calculate a route from the current position to the destination. This procedure is done every time until the target is reached.

4. Smartphone with Realtime route computation:

In this situation, a "real-time simulation" of the fastest route is supposed to be calculated. As a navigation system of a car continuously considers traffic jams, accidents, and other conditions to visualize the fastest route, it is assumed to follow the instance using TripUpdates. The concept is to constantly calculate the fastest route and inform the user in case of a newly calculated route, where s\he must change buses. As the trip planner would calculate a walking route to the next stop while the bus is on tour, the continuous re-calculation is supposed to be performed at each stop. This process is done until the passenger reaches the desired place.

5. Optimal route calculation using modified GTFS feed:

This scenario represents the "optimal" route that could be used by the passenger. The routing graph has timetable information from the modified GTFS feed and is only true for routes that are in the past. As the modified GTFS feed includes the observed arrival and departure times which were collected during the real-time data collection phase, the calculated route in this scenario is supposed to be the "optimal" one the passenger could have taken.

All these scenarios have the same route request look. Differences between scenarios are in the used routing graph, the use of TripUpdates, and if requests will be renewed recursively in case of a missed bus. The request is sent to the port of the local running OTP instance and contains the optional parameters such as route mode, optimization criteria, and maximal walking distance. Besides, the origin and destination coordinates will be attached alongside the date and desired time.

3.4 Study area and geodata

For this research, one of the following two suggested study areas will be considered: *Boston Metropolitan Area* and *Broward County, Florida*. Both provide *GTFS* feeds.

3.4.1 Boston Metropolitan Area (Massachusetts Bay Transportation Authority MBTA)

The Massachusetts Bay Transportation Authority, more commonly known as the T, is one of the oldest public transit systems in the United States. It's also the largest transit system in Massachusetts. As a division of the Massachusetts Department of Transportation (MassDOT), the MBTA provides subway, bus, Commuter Rail, ferry, and paratransit service to eastern Massachusetts and parts of Rhode Island ('MBTA' 2020).

3.4.1.1 Transit Systems

- Subway Lines

The Red, Orange, Blue, and Green subway lines provide fast, easy connections to and from Boston and surrounding cities, including Cambridge, Newton, Revere, and Quincy, as Figure 16 shows.



Figure 16 MBTA subway map ('Subway | Schedules & Maps | MBTA' 2020)

3.4.1.2 MBTA GTFS Feeds

Both GTFS static and GTFS-Realtime specification feeds are available to be downloaded (Figure 18).

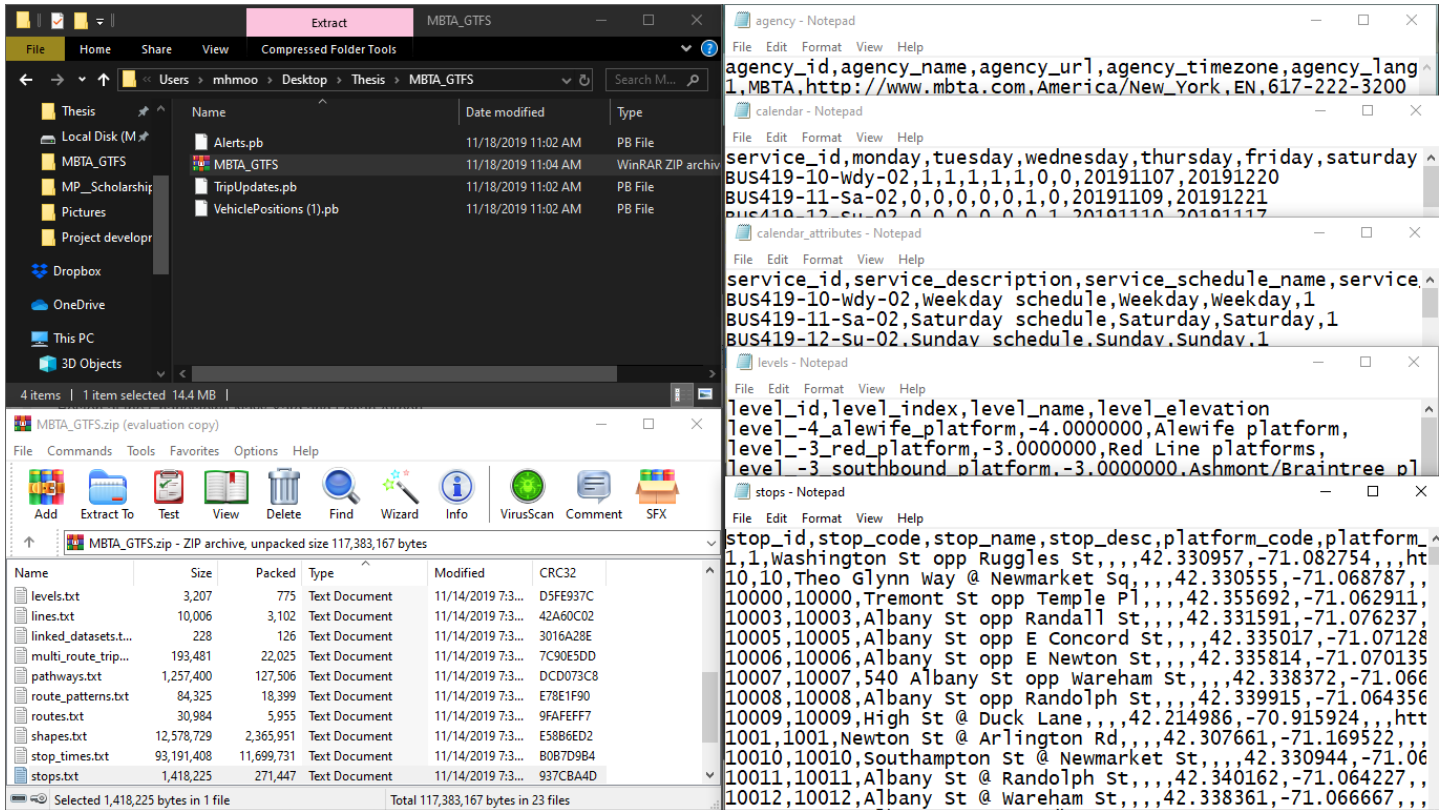


Figure 18 GTFS dataset from MBTA transit agency consists of several text files within a ZIP file and GTFS-Realtime files

3.4.2 Broward County, Florida (Broward County Transit, BCT).

Broward County Transit (also known as BCT) is the public transit authority in Broward County, Florida. It is the second-largest transit system in Florida after Miami-Dade Transit. It currently operates the only public bus system in Broward County. Besides serving Broward County, it also serves portions of Palm Beach County and Miami-Dade County, where it overlaps its service with Miami-Dade Transit and Palm Tran (Figure 19). Broward Realtime comes in a different format (*TXT*, *JSON* & *XML*), which need to be encoded and converted to *Protobuf* format to be used.



Figure 19 an overview of BCT's service area ('Broward County Transit' 2020)

4 Implementation

4.1 Static Data Preparation

The static data was collected then imported into the PostgreSQL database to be used with the Realtime data for the analysis.

4.1.1 Static Data Collection

Boston Metropolitan Area (MBTA) GTFS-Static data were downloaded to be combined with the Realtime data on Wednesday, June 24, 2020, for the period from June 16, 2020, to August 29, 2020, Summer 2020.

4.1.2 Static Data Importing

After downloading the digital timetable of the whole transit network as a GTFS feed, tables were created in the PostgreSQL database with the same names in the static zip file (Figure 20) to import the data to these tables to be used in the analysis, taking into account the type of each column of these tables and adding *geom* column into tables such as **Stops** (Figure 21) to be visualized in the map (Figure 22).

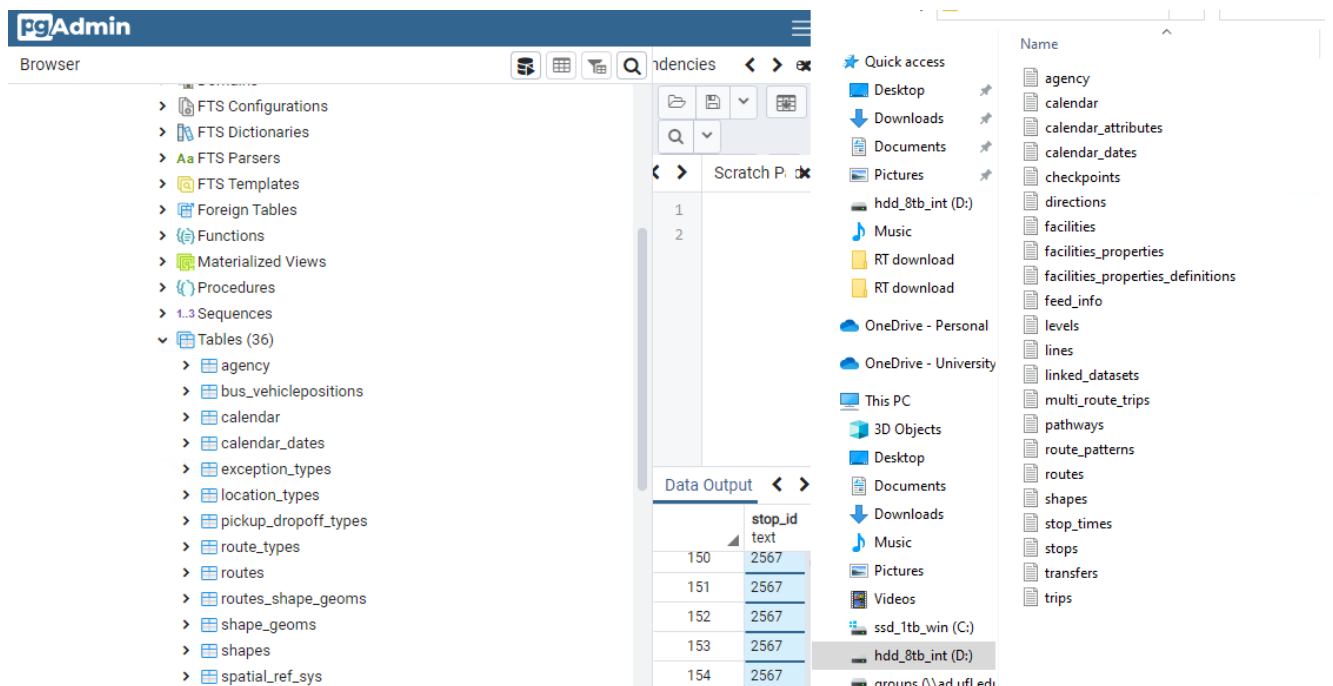


Figure 20 Static data in the PostgreSQL Database tables and zip file tables


```

CREATE TABLE stops (
  stop_id text,
  stop_code text,
  stop_name text DEFAULT NULL,
  stop_desc text DEFAULT NULL,
  platform_code text DEFAULT NULL,
  platform_name text DEFAULT NULL,
  stop_lat double precision,
  stop_lon double precision,
  zone_id text,
  stop_address text DEFAULT NULL,
  stop_url text,
  level_id text DEFAULT NULL,
  location_type integer,
  parent_station text DEFAULT NULL,
  wheelchair_boarding smallint,
  municipality text DEFAULT NULL,
  on_street text DEFAULT NULL,
  at_street text DEFAULT NULL,
  vehicle_type int DEFAULT NULL,
  stop_geom geometry('POINT', 4326),
  CONSTRAINT stops_pkey PRIMARY KEY (stop_id)
);

```

Figure 21 Creating GTFS Static Data tables in PostgreSQL

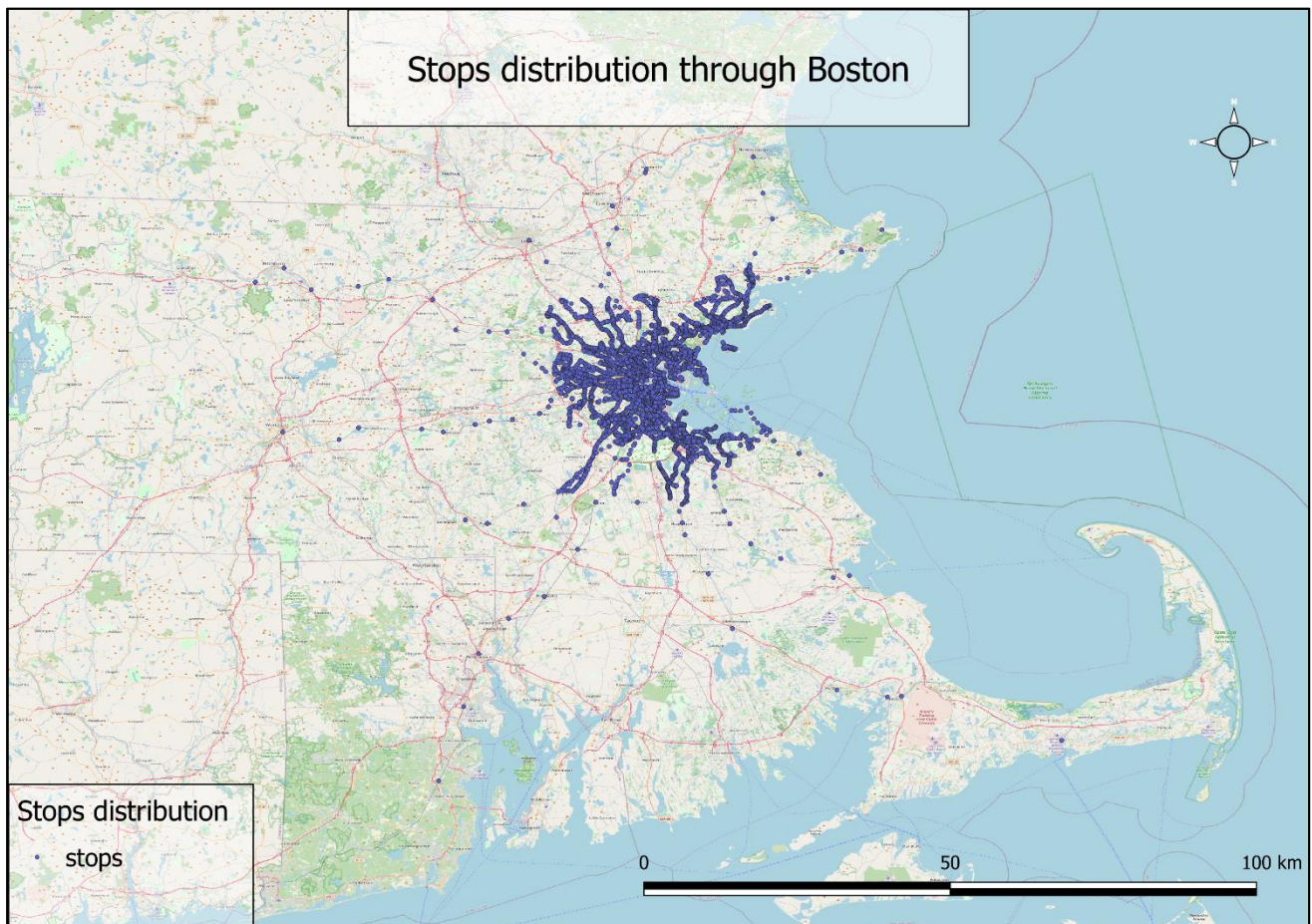


Figure 22 Stops distribution through Boston

4.2 MBTA Realtime Transit Data Completeness & Accuracy Evaluation

Evaluating Real-time transit information completeness & Accuracy to be used in trip planning is the major part of this research. As currently, many transit agencies are providing Real-time data, MBTA agency is what this research deals with as the first case. This GTFS Real-time data is accessible manually or via programming interfaces. *Python (3.5)* is used as the main programming language for this work. *IDLE* and *PyCharm* were the development platforms used to extract, manage, and import the data into the *PostgreSQL* database.

4.2.1 Realtime Data Collection

Realtime data was collected for a certain time which is a whole day starting from 5:59 am till 8:25 pm on Wednesday, June 24, 2020. As both, the *TripUpdates* and the *VehiclePositions* files are updated continuously every 10 seconds on any agency website, these files were also downloaded for this day every 10 seconds using the *IDLE* development platform. Two pieces of Python code were written to guarantee these files were downloaded locally continuously during this period specified, one for *TripUpdates* and the second for *VehiclePositions* (Figure 23).

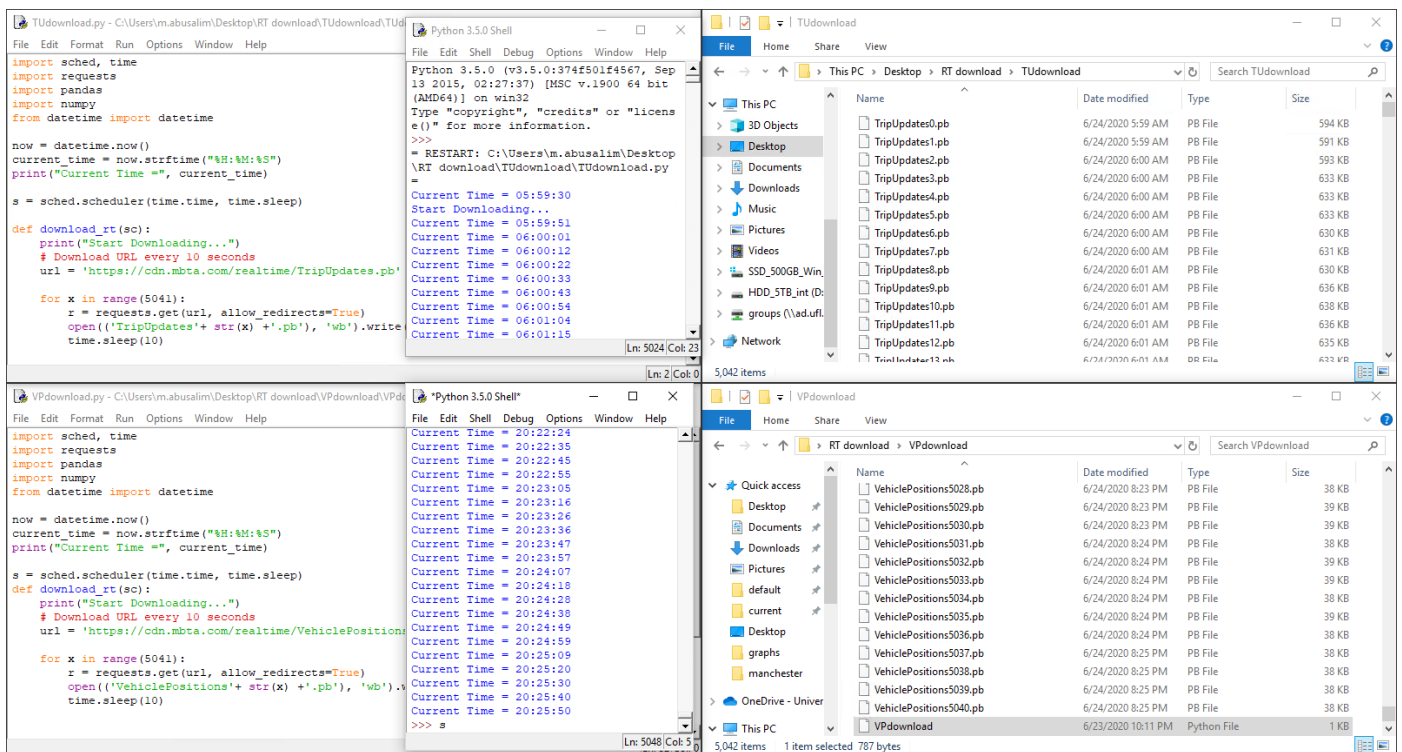


Figure 23 Python codes that used to download the *TripUpdates* and the *VehiclePositions* files

4.2.1 Realtime Data Transformation

Alongside GTFS-Static data, GTFS-Realtime data need to be stored in the database to use joined data from both of them in the analysis. The PostgreSQL database supports queries and changes, along with the linkage among tables using their unique identifiers. One of the most important benefits of importing all the feeds into a database is the computation speed as no CSV-files browser has the same ability or can join several tables.

TripUpdates and the VehiclePositions files were decoded to be readable as objects using the Python programming environment. As the compressed Realtime information is stored in the protocol buffer file, the way to process it is to use classes that store the data in the memory, then *protoc.exe*, which is a compiler from Google, use the description of *GTFS-Realtime .proto* importing *GTFS _Realtime _pb2* library from *google.transit* to create classes for the programming language used. These classes include techniques that support the encoding of data into binary files (Masina 2019). These binary files are used to import the objects into the PostgreSQL database which contains the *PostGIS* extension which is used in spatial queries. All this process of encoding has been done after creating all table's columns and specifying their types to import the data into these columns of the table in the PostgreSQL database within one Python script.

The Realtime tables have the following schemas:

- VehiclePositions table structure in the PostgreSQL database:

Table 5 VehiclePositions table structure in the PostgreSQL database

Column	Type	Description
trip_id	TEXT	Trip Unique Identifier
start_time	TEXT	Trip Start Time
start_date	TEXT	Trip Start Date
route_id	TEXT	Route Unique Identifier
direction_id	INTEGER	Direction Unique Identifier
vehicle_id	TEXT	Vehicle Unique Identifier
vehicle_label	TEXT	Vehicle Label
timestamp	INTEGER	The moment at which the vehicle's position was measured. (In POSIX time)
MeasureTime	TEXT	The moment at which the vehicle's position was measured
stop_sequence	INTEGER	The stop sequence index of the current stop
stop_id	TEXT	Identifies the current stop
current_status	TEXT	The exact status of the vehicle concerning the current stop
latitude	FLOAT	Vehicle latitude
longitude	FLOAT	Vehicle longitude
pos_geom	GEOMETRY(POINT,4326)	Position of the vehicle using latitude & longitude

- TripUpdates table structure in the PostgreSQL database:

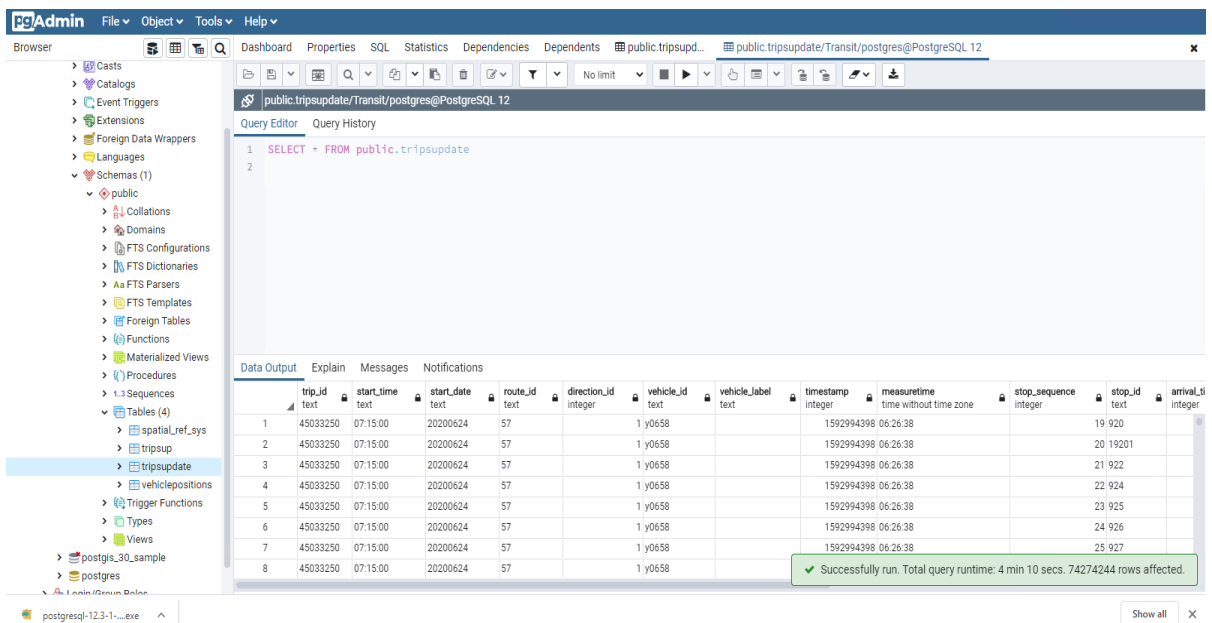
Table 6 TripUpdates table structure in the PostgreSQL database

Column	Type	Description
Trip_Id	TEXT	Trip Unique Identifier
Start_Time	TEXT	Trip Start Time
Start_Date	TEXT	Trip Start Date
Route_Id	TEXT	Route Unique Identifier
Direction_Id	INTEGER	Direction Unique Identifier
Vehicle_Id	TEXT	Vehicle Unique Identifier
Vehicle_Label	TEXT	Vehicle Label
Timestamp	INTEGER	The moment at Which the Vehicle's Position Was Measured. (In Posix Time)
Measuretime	TEXT	The moment at Which the Vehicle's Position Was Measured
Stop_Sequence	INTEGER	The Stop Sequence Index of The Current Stop
Stop_Id	TEXT	Shows the Current Stop
Arrival_Time	INTEGER	Trip Arrival Time
Arrival_Uncertainty	INTEGER	Trip Arrival Uncertainty
Arrival_Delay	INTEGER	Arrival Delay (In Seconds)
Departure_Time	INTEGER	Trip Departure Time
Departure_Uncertainty	INTEGER	Trip Arrival Uncertainty
Departure_Delay	INTEGER	Departure Delay (In Seconds)

4.2.2 Realtime Data Importing

The step after the data encoding into a readable form is importing the data into the tables that were created in the PostgreSQL database using the same Python script used before for encoding. Having these tables allows an accurate evaluation of the time and location of each vehicle and gives useful information for creating modified GTFS feed tables that contain accurate information about departure and arrival times of each trip. The following table shows the TripUpdates table after importing it into the PostgreSQL database (Table 7).

Table 7 TripUpdates table after importing it into the PostgreSQL database



The screenshot shows the pgAdmin interface with a query editor and a data output table. The query editor contains the following SQL query:

```
1 SELECT * FROM public.tripsupdate
2
```

The data output table displays the following columns and data:

trip_id	start_time	start_date	route_id	direction_id	vehicle_id	vehicle_label	timestamp	messurtime	stop_sequence	stop_id	arrival_t
1	45033250	07:15:00	20200624	57			1592994398	06:26:38		19	920
2	45033250	07:15:00	20200624	57	1	y0658	1592994398	06:26:38		20	19201
3	45033250	07:15:00	20200624	57	1	y0658	1592994398	06:26:38		21	922
4	45033250	07:15:00	20200624	57	1	y0658	1592994398	06:26:38		22	924
5	45033250	07:15:00	20200624	57	1	y0658	1592994398	06:26:38		23	925
6	45033250	07:15:00	20200624	57	1	y0658	1592994398	06:26:38		24	926
7	45033250	07:15:00	20200624	57	1	y0658	1592994398	06:26:38		25	927
8	45033250	07:15:00	20200624	57	1	y0658	1592994398	06:26:38			

A green notification box at the bottom right of the table indicates: "Successfully run. Total query runtime: 4 min 10 secs. 74274244 rows affected."

4.2.1 Realtime Data Filtering

In the data filtering phase, the goal is to filter out each trip collected information. Firstly, all duplicates records were dropped from both TripUpdates and VehiclePositions tables which minimized their sizes. The *current status* indicates the vehicle status if it is “stopped at” or “in transit to” a certain stop. When a vehicle has the status “stopped at” a specific stop the time range during being at this stop is used the modification of the original GTFS feed. The first timestamp is used as the observed arrival time and the last one is used as departure time. For example, the original GTFS feed *stop times* have an arrival and departure time at “08:00:00” at stop number 2, whilst the observed arrival time from the VehiclePositions table was at “07:57:14” and the observed departure time was at “07:59:44” Eastern Time (GMT-4), which means that this vehicle arrived 3 minutes and departed about 15 seconds before the time scheduled.

Table 8 shows an extract of a trip from the VehiclePositions feeds table. Each row represents information about a specific timestamp where the VehiclePositions feed has been collected.

Table 8 Extract of a trip from VehiclePositions table

	trip_id text	stop_id text	stop_sequence integer	current_status text	measuretime text	latitude double precision	longitude double precision
1	CR-Weekd...	TF Green Ai...	2	2	07:55:47	41.71686935424805	-71.4460220336914
2	CR-Weekd...	TF Green Ai...	2	2	07:47:14	41.59144973754883	-71.48600006103516
3	CR-Weekd...	TF Green Ai...	2	2	07:54:14	41.70140838623047	-71.45159912109375
4	CR-Weekd...	TF Green Ai...	2	2	07:47:43	41.5968017578125	-71.48329162597656
5	CR-Weekd...	TF Green Ai...	2	2	07:55:14	41.71118927001953	-71.44808197021484
6	CR-Weekd...	TF Green Ai...	2	1	07:57:14	41.72766876220703	-71.4420394897461
7	CR-Weekd...	TF Green Ai...	2	1	07:59:44	41.7277717590332	-71.4420394897461
8	CR-Weekd...	TF Green Ai...	2	2	07:52:43	41.68436813354492	-71.45069885253906
9	CR-Weekd...	TF Green Ai...	2	2	07:53:14	41.69137954711914	-71.45406341552734
10	CR-Weekd...	TF Green Ai...	2	2	07:53:43	41.69607162475586	-71.45355987548828
11	CR-Weekd...	TF Green Ai...	2	2	07:54:44	41.70621871948242	-71.4498291015625
12	CR-Weekd...	TF Green Ai...	2	2	07:56:13	41.721519470214844	-71.44432067871094
13	CR-Weekd...	TF Green Ai...	2	2	07:50:13	41.639278411865234	-71.46180725097656

Table 9 reveals an extract of the same trip from the stop times table. Each row represents information about a specific stop.

Table 9 Extract of a trip from Stop_times table

	trip_id [PK] text	stop_id text	stop_sequence [PK] integer	arrival_time interval	departure_time interval
1	CR-Weekday...	Wickford J...		1 07:45:00	07:45:00
2	CR-Weekday...	TF Green Ai...		2 08:00:00	08:00:00
3	CR-Weekday...	Providence		3 08:15:00	08:25:00
4	CR-Weekday...	South Attle...		4 08:34:00	08:34:00
5	CR-Weekday...	Attleboro		5 08:44:00	08:44:00
6	CR-Weekday...	Mansfield		6 08:54:00	08:54:00
7	CR-Weekday...	Sharon		7 09:03:00	09:03:00
8	CR-Weekday...	Canton Jun...		8 09:10:00	09:10:00
9	CR-Weekday...	Route 128		9 09:15:00	09:15:00
10	CR-Weekday...	Hyde Park		10 09:20:00	09:20:00
11	CR-Weekday...	Ruggles		11 09:29:00	09:29:00
12	CR-Weekday...	Back Bay		12 09:33:00	09:33:00
13	CR-Weekday...	South Stati...		13 09:39:00	09:39:00

The observed arrival and departure times are used to modify the original GTFS feed to be used in routes calculation. This means that the stop times table from the original feed will be modified using the information from the VehiclePositions table so a modified stop times table will be exported from the database, then this table is exported into a stop_times.txt which replace the original one in the zip file to have modified Static GTFS feed.

5 Results

In this chapter, a deep look into the collected data has been made by applying data analysis and visualization techniques to detect the effectiveness and accuracy of the data.

The whole process of Realtime data collection for this research has been done on June 24th, 2020, from 5:59 A.M. until 8:25 P.M. . *VehiclePositions* and *TripUpdates* feeds were collected every 10 seconds during the data collection period. Thus, high collection frequency has been done to guarantee the desired accuracy in the estimation of arrival and departure times and to have all trip updates. The *GTFS Static* feed along with the OSM file of Boston city was downloaded for the same period. The following sections include a detailed description of the collected information and analysis.

During the analysis of the collected data, five main analysis steps were performed. **Descriptive Analysis** of the data which is generating insightful statistics of trips, stops, and routes. Besides, calculating the percentage of *VehiclePositions* and *TripUpdates* was the first step. The second step is showing the distribution of the *VehiclePositions* and *TripUpdates* feeds on the studied region. The third step is to perform comparisons of the trips and stops between the Static and Realtime data per vehicle's type to clarify the percentage of stops that have updates in the Realtime feeds. Next, the arrival and departure delay has been analyzed for the stops and trips in total and then was broken down for each vehicle's type. Finally, a comparison was performed between the delay data with and without outliers to show the delay measurements were affected by outliers (Figure 24).

Collected Data Analysis

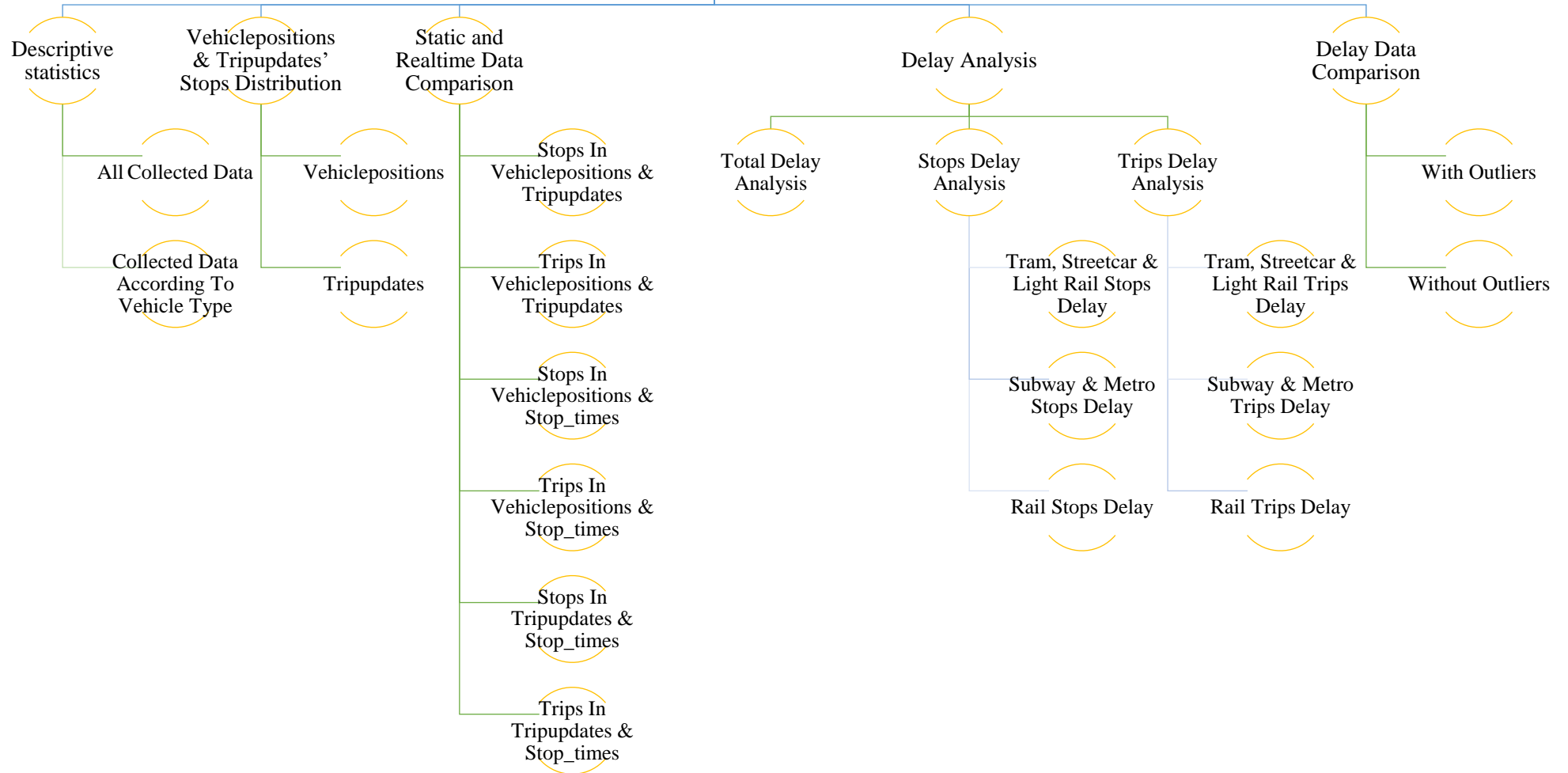


Figure 24 Collected Data Analysis Steps

5.1 Descriptive statistics of all collected data

The GTFS Static feed of Boston that was collected covers 241 routes used for 100797 trips and 9873 stops. Also, about 311051 shapes and more than 2.5 million different stop times for the arrival and departure time information at each stop for all trips. Likewise, during 14 hours of data collection, about 74 million TripUpdates entries and 1.6 million VehiclePositions were collected. The total trips involved in TripUpdates entries are 12017, while there was 11307 participating in the VehiclePositions feed, which is about 11% of all available trips in the stop times table. The descriptive statistics in the following table provides information about the complete amount of collected data (Table 10).

Table 10 Descriptive statistics of the collected data

	Entries	Routes		Stops		Trips	
		Total	Percentage	Total	Percentage	Total	Percentage
Stop_times	2528128	241	100%	7892	100%	100797	100%
VehiclePositions	1629524	161	67%	6736	85%	11307	11%
TripUpdates	74274244	157	65%	6830	87%	12017	12%

5.1.1 Descriptive statistics of the collected data according to vehicle type

On the other hand, the collected data were classified per different vehicle types as shown in the following table. When the current status = 1, that means the vehicle is currently at one of the stops along its trip, and its arrival and departure times could be calculated using the measured time and timestamp, otherwise, exact information about the timing of arrival or departure of the vehicle cannot be measured. Lastly, Ferry has no Realtime feeds at all (Table 11).

Table 11 Descriptive statistics of the collected data according to vehicle type

Code	vehicle_type	Stops	VehiclePositions	VehiclePositions current_status = '1'	TripsUpdate
0	Tram, Streetcar, Light rail	149	70237	16215	2249820
	Unique		119	118	118
1	Subway, Metro	114	51289	13004	1436158
	Unique		109	106	108
2	Rail	176	46317	16101	317724
	Unique		155	152	166
3	Bus	7487	1256589	88	70262818
	Unique		6336	84	6419
4	Ferry	6	0	0	0
	SUM	7932	1424432	45408	74266520
	Unique	7932	6719	460	6811

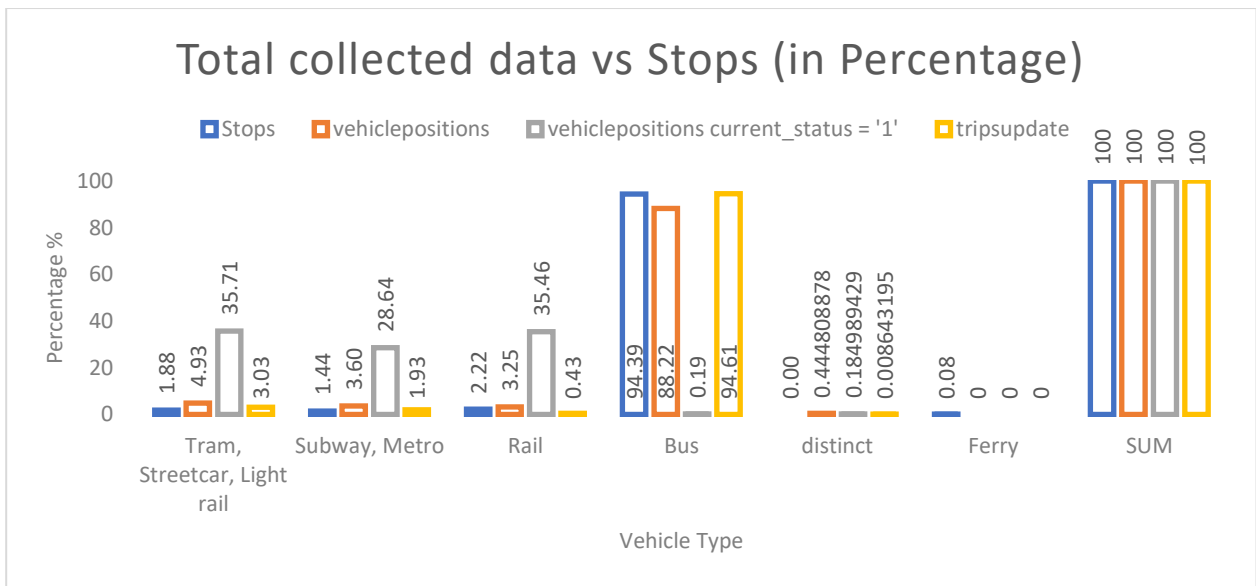


Figure 25 Total collected data vs Stops (in Percentage)

From the previous graph (Figure 25) it is clear that the information about Buses arrivals and departures are missing (0.19% of the total collected data) because the (*current status = 1*) is not provided in the VehiclePositions feeds for almost all the stops which lead to this error. Therefore, the Bus is dropped from the next analysis, and other methods should be performed to have an estimation of this missing information from VehiclePositions feeds. The following chart (Figure 26) shows how much data has the (*current status = 1*) out of all VehiclePositions feeds categorized by vehicle type, which reveals the Bus data issues. On one hand, only 88 out of 1,256,589 Bus's VehiclePositions have the (*current status = 1*), which is barely (0.01%). On the other hand, all other vehicles have around (25%) of VehiclePositions have the (*current status = 1*).

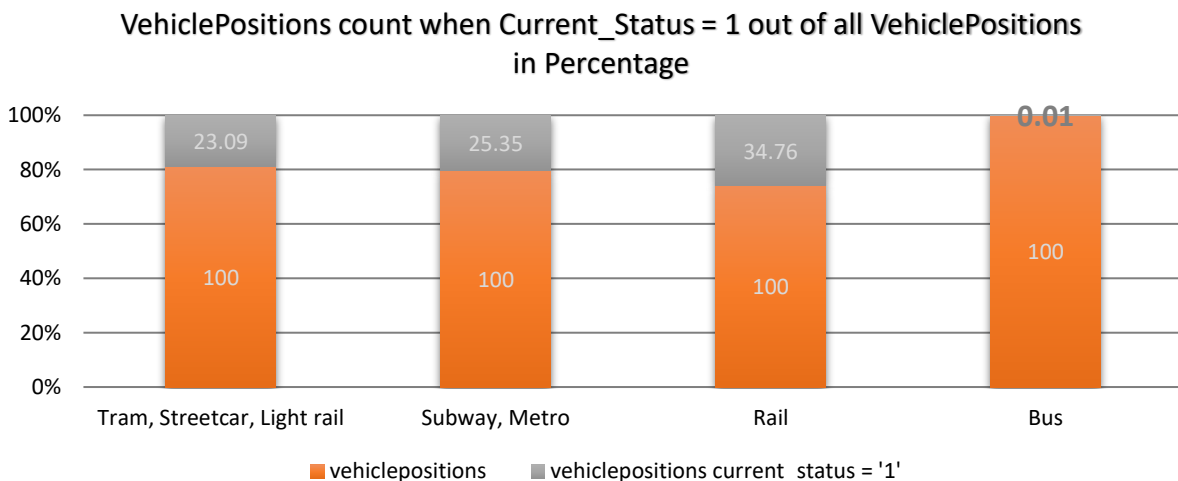


Figure 26 VehiclePositions count when Current Status = 1 out of all VehiclePositions in Percentage

5.2 VehiclePositions feeds and TripUpdates feeds' Stops distribution in Boston.

As a heatmap visualizes "hotspots" of the distribution of features on the map i.e. dense areas, it was used to show stops in Boston that are affected by Realtime updates. Each stop has a fixed number of trips running through it, the following will be shown how often every stop was mentioned in these feeds.

The following maps (Figure 27 & Figure 28) are illustrations of the distribution of the stops included in VehiclePositions and TripUpdates feeds through Boston city. It is noticeable that the central area of Boston has most of the feeds.

- VehiclePositions feed distribution per stop.

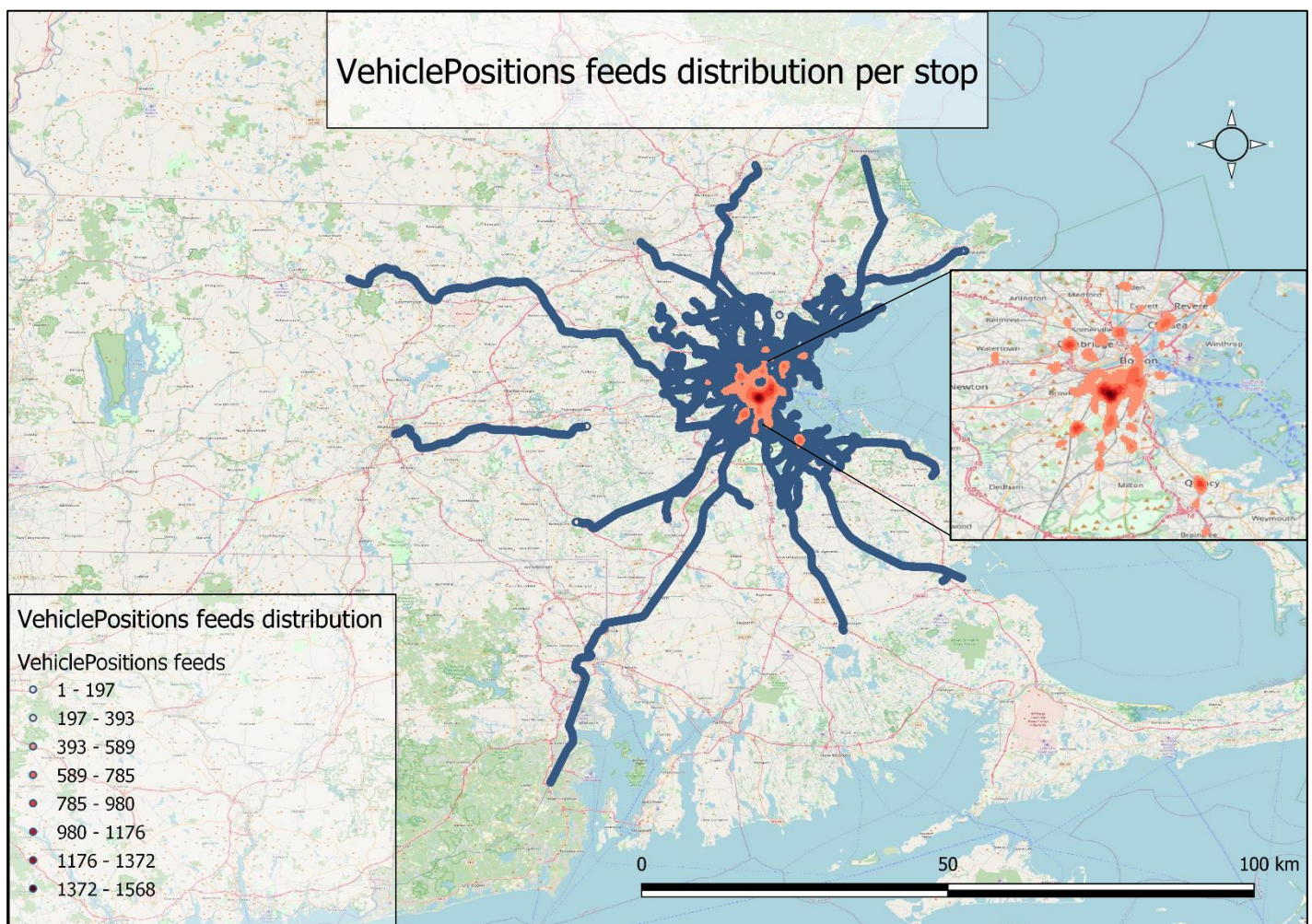


Figure 27 VehiclePositions feeds distribution per stop

- TripUpdates feeds distribution per stop.

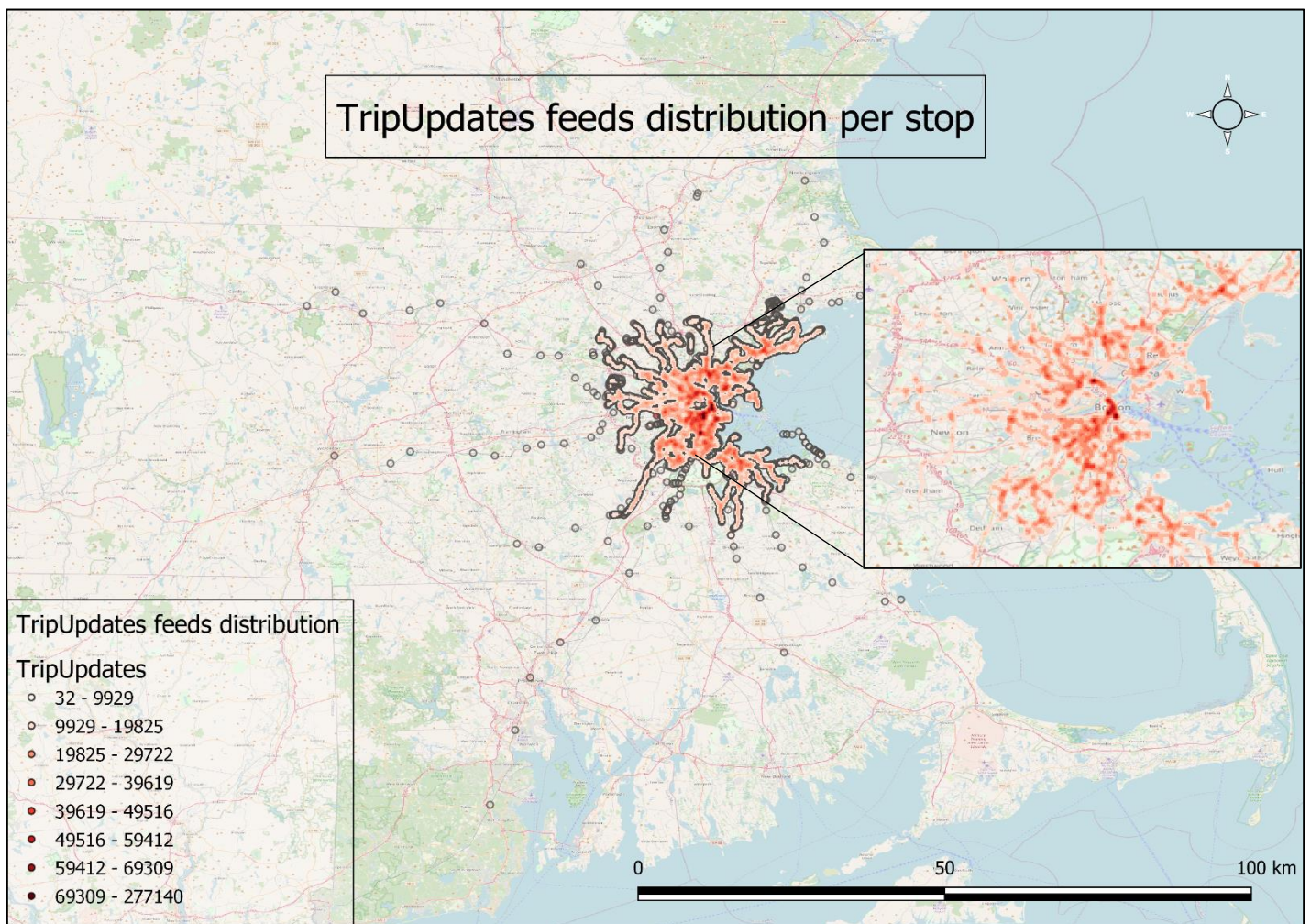


Figure 28 TripUpdates feeds distribution per stop

5.3 Static and real-time information Comparison

In this section, different comparisons of stops and trips were made between static and real-time information to give more details about the data.

- **Stops included in the VehiclePositions feeds and TripUpdates feeds.**

When comparing stops that were included in VehiclePositions feeds with stops included in TripUpdates feeds, 96 stops were mentioned in the TripUpdates feeds and there was no information at all about them in the VehiclePositions feeds. 12 stops of these were Rail stops while the rest were Bus stops. On the other hand, only 2 stops are having VehiclePositions feeds without any information in the TripUpdates feeds (Table 12).

Table 12 Comparison of stops included in the VehiclePositions and TripUpdates feeds

<i>Vehicle Type</i>	TripUpdates	VehiclePositions
<i>Tram, Streetcar, Light rail</i>	-	1
<i>Subway, Metro</i>	-	1
<i>Rail</i>	12	-
<i>Bus</i>	84	-
<i>Total</i>	96	2

- **Trips included in the VehiclePositions feeds and TripUpdates feeds.**

For the trips' updates, 75 trips are having VehiclePositions feeds without any information in the TripUpdates feeds. Only 2 trips are using Subway and Metro while the rest are using the bus. In comparison, there were 984 trips included in the TripUpdates feeds only without any VehiclePositions updates (Table 13).

Table 13 Comparison of trips included in the VehiclePositions and TripUpdates feeds

<i>Vehicle Type</i>	TripUpdates	VehiclePositions
<i>Tram, Streetcar, Light rail</i>	281	-
<i>Subway, Metro</i>	83	2
<i>Rail</i>	3	-
<i>Bus</i>	617	73
<i>Total</i>	984	75

- **Stops included in the VehiclePositions feeds and Stop_times.**

Stop_times table has 1206 stops which are not indicated in the VehiclePositions feeds. In contrast, 31 stops have VehiclePositions updates, but they are not in the Stop_times table. These stops are divided into 24 Rail stops, 6 Subway & Metro stop, and 1 Tram, Streetcar, or Light rail stops.

- **Trips included in the VehiclePositions feeds and Stop_times.**

Although the total amount of trips in the Stop_times table is 100797 trips and 11307 in the VehiclePositions feeds which means the difference between them is 89490 trips, that is not the case here. There are 91240 trips in the Stop_times table not in the VehiclePositions feeds. Surprisingly, 994 trips have VehiclePositions updates, but they are not in the Stop_times table. When these trips were investigated, they seemed like added trips during the day as all these trips have either “ADDED-“ before the trip name for the Tram, Streetcar, Light rail, Subway and Metro trips, for example (ADDED-1580448294), or the “OL” at the end of the bus trips for instance (45043214-OL1).

- **Stops included in the TripUpdates feeds with Stop_times.**

Stop_times table has 1113 stops that are not shown in the TripUpdates feeds. In contrast, 32 stops have TripUpdates updates, but they are not mentioned in the Stop_times table. These stops are divided into 25 Rail stops, 6 Subway & Metro stop, and 1 Tram, Streetcar, or Light rail stops. These stops that are included in the TripUpdates are the same stops that were mentioned in the VehiclePositions feeds.

- **Trips included in the TripUpdates feeds with Stop_times.**

While the total amount of trips in the Stop_times table is 100797 trips and 12017 in the TripUpdates feeds which means the difference between them is 88780 trips, there are 90517 trips in the Stop_times table not in the TripUpdates feeds. Additionally, strangely, 981 trips have TripUpdates updates, without information in the Stop_times table. When these trips were examined, they appeared the same trips in the VehiclePositions feeds, with some kind of names having either “ADDED-“ before the trip name for the Tram, Streetcar, Light rail, Subway, and Metro trips, for example (ADDED-1580448294), or the “OL” at the end of the bus trips for instance (45043214-OL1).

5.4 Delay analysis

In this stage, the delay information was calculated for all data per stop and per trip. To do so, the difference between observed times and schedules times (static data) was computed. Massive outliers were detected during that. Firstly, delay analysis was performed in total. Then, the data were classified per vehicle type for stops and trips.

5.4.1 Total Delay Analysis

Delays for all arrival and departure times were analyzed using a box plot to show the distribution of the delays. Box plot can outline the outliers, the max, the median, and the min of the data which gives a clear picture of the data. Following are box plots of the arrival and departure delays showing similar delay distribution (Figure 29 & Figure 30).

Table 14 shows a comparison of the information collected from these box plots. It shows, both arrival and departure delays have the same maximum delay with 8874 seconds, and almost the same minimum with only a 30-second difference. For the first and third quartiles, the difference between arrival and departure delays less than 12 seconds, which means they are similar. This means that 50% of the delays are between -3 and +1.5 minutes, which is far from the minimum and the maximum delays leading to huge outliers. Finally, the mean and the median for arrival and departure delays are less than 11-second different.

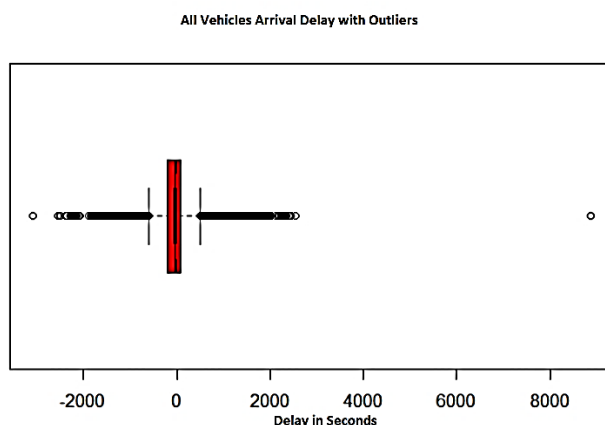


Figure 29 All vehicles arrival delay with outliers

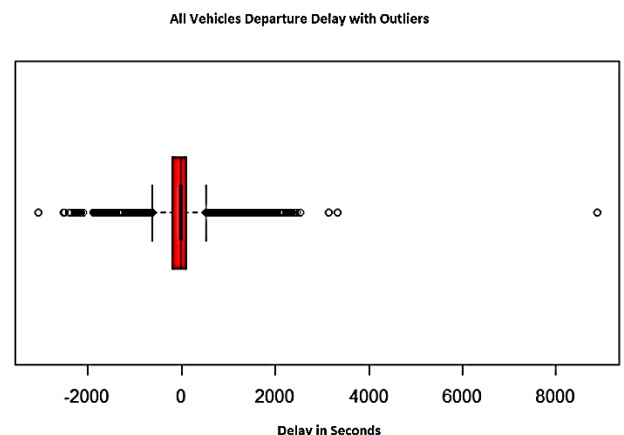


Figure 30 All vehicles departure delay with outliers

Table 14 Comparison of box plots information of arrival and departure delay

All Vehicles		
	Arrival	Departure
Min.	-3100	-3070
1st Qu.	-198	-196
Median	-30	-16
Mean	-20.28	-10.56
3rd Qu.	77	88
Max.	8874	8874

5.4.2 Stops Delay Analysis

Delay average was calculated for all stops during the day, then these stops were classified according to the vehicle's number which goes through these stops to observe the relationship between them. The following graphs demonstrate average vs median delays per stop for arrival and departure times. These graphs in general indicate that vehicles tend to arrive and depart early before the time scheduled from stops that have less than 80 vehicles went through. In contrast, when the stop has more than 80 vehicles went through, these vehicles arrived and departed late after their schedule. Additionally, stops that have similar average and median have similar delays in general and vice versa (Figure 31). Figure 32 shows up positive and negative delay distribution of the arrival and departure.

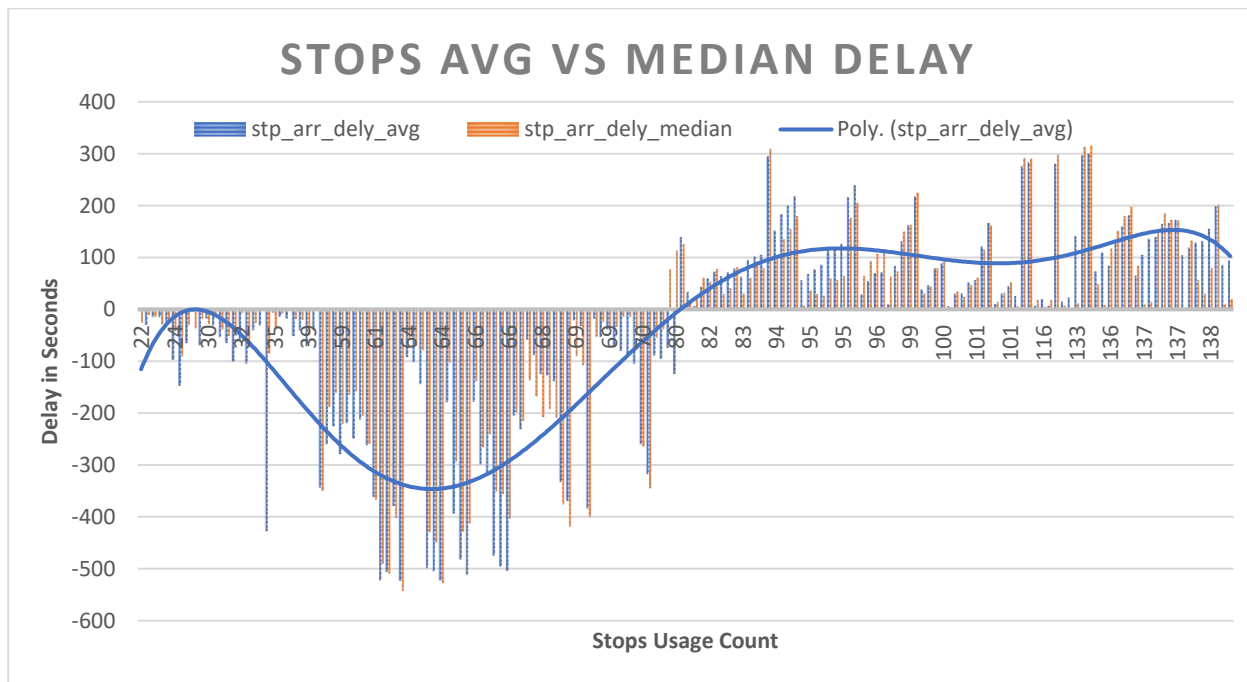


Figure 31 all stops delay average vs median

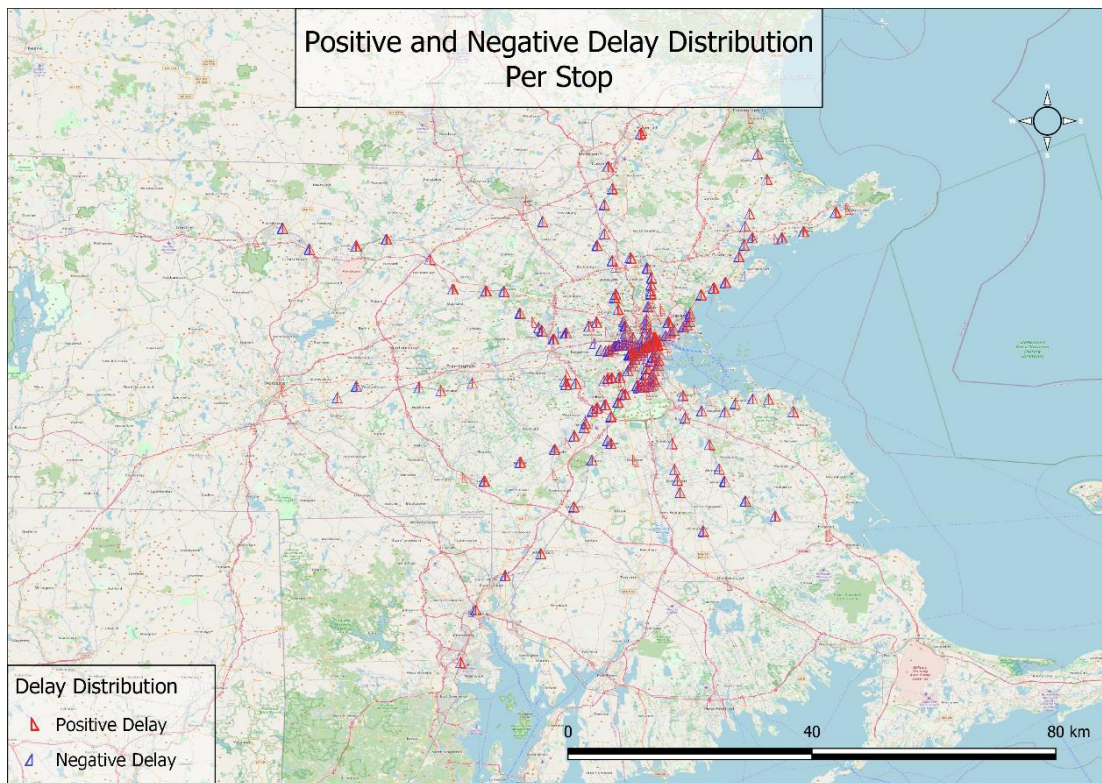


Figure 32 Positive and Negative Delay Distribution Per Stop

5.4.3 Tram, Streetcar & Light rail Stops Delay Analysis

Starting with Tram, Streetcar & Light rail, graphs show the same trends as all stops. Stops that have less than 70 vehicles went through are more likely to have vehicles arrive and leave before schedule while stops with more than 70 vehicles have delayed (Figure 33).

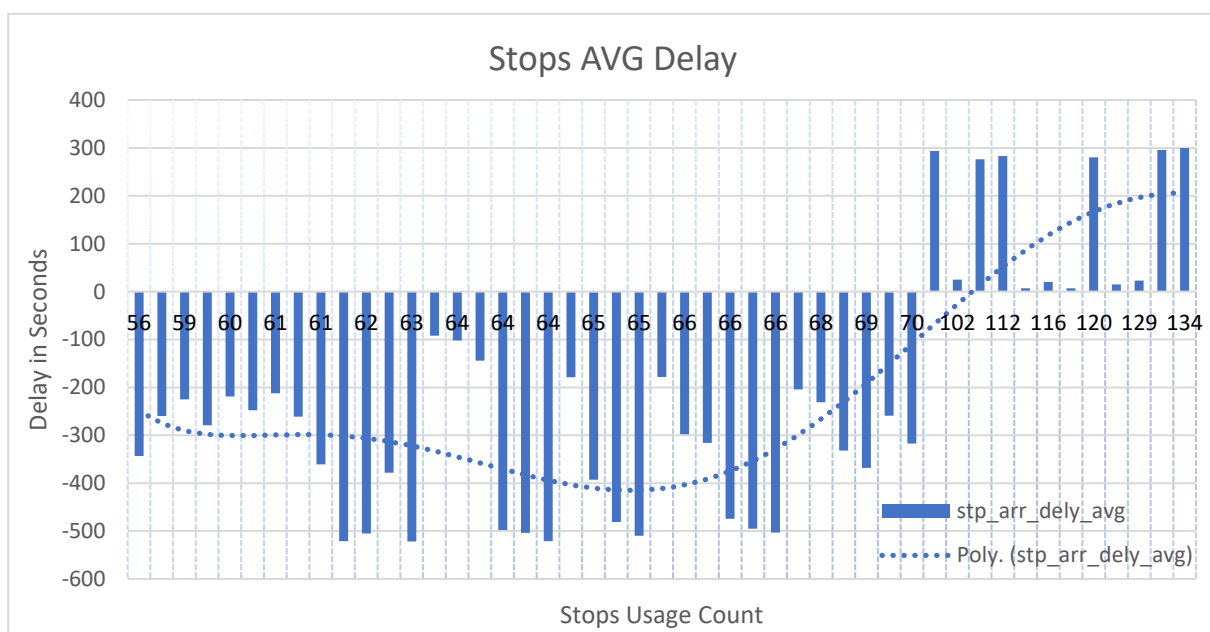
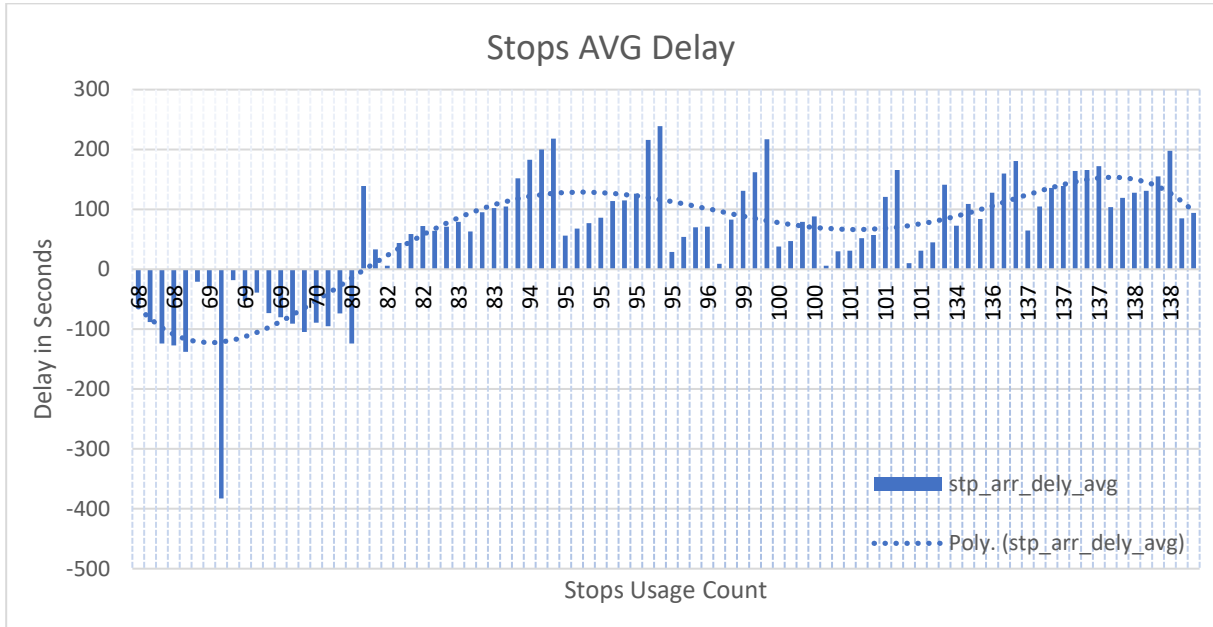


Figure 33 Tram, Streetcar & Light rail stops average delay

5.4.4 Subway & Metro Stops Delay Analysis

Subway & Metro have the highest vehicle number per stop. The following graphs indicate the same delay trends as before. Stops with less than 80 vehicles have vehicles arrived earlier than scheduled, whereas stops with more than 80 vehicles have delayed vehicles (Figure 34).



5.4.6 Trips Delay Analysis

In this part, the delay information average was calculated for each trip. Then, these trips were categorized per vehicle type showing delays trend in general for each type.

5.4.7 Tram, Streetcar & Light rail Trips Delay Analysis

Trips that used Tram, Streetcar & Light rail were delayed at three periods of the day, the beginning of the day around 6:00 am, the middle of the day around noon, and at the end of the working day around 7:00 pm. These periods are more likely related to the working hours, beginning and end of work times, and the break in the middle of the day (Figure 36).

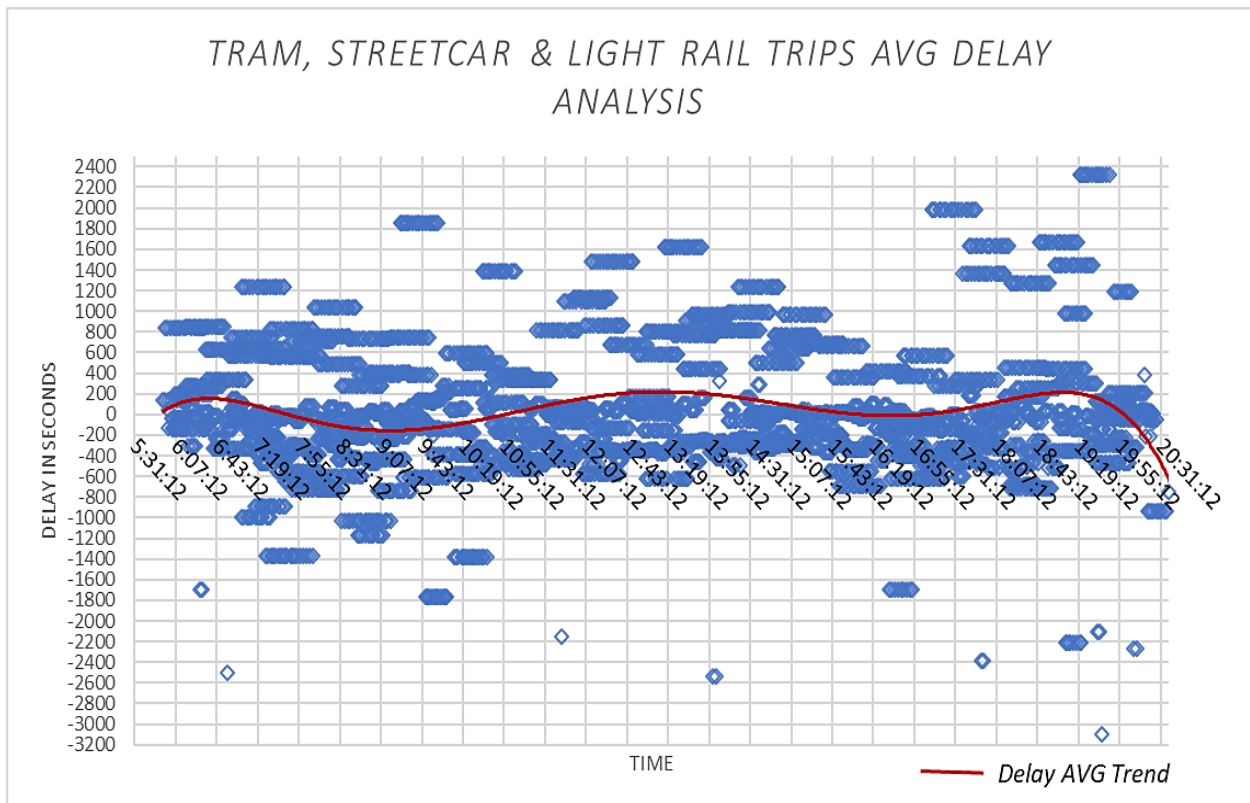


Figure 36 Tram, Streetcar & Light Rail Trips Delay Average

5.4.8 Subway & Metro Trips Delay Analysis

For Subway & Metro Trips, most of the delays for both arrival and departure times are at the beginning of the day, then the delays are distributed on all the day with less delay around 7:00 pm (Figure 37).

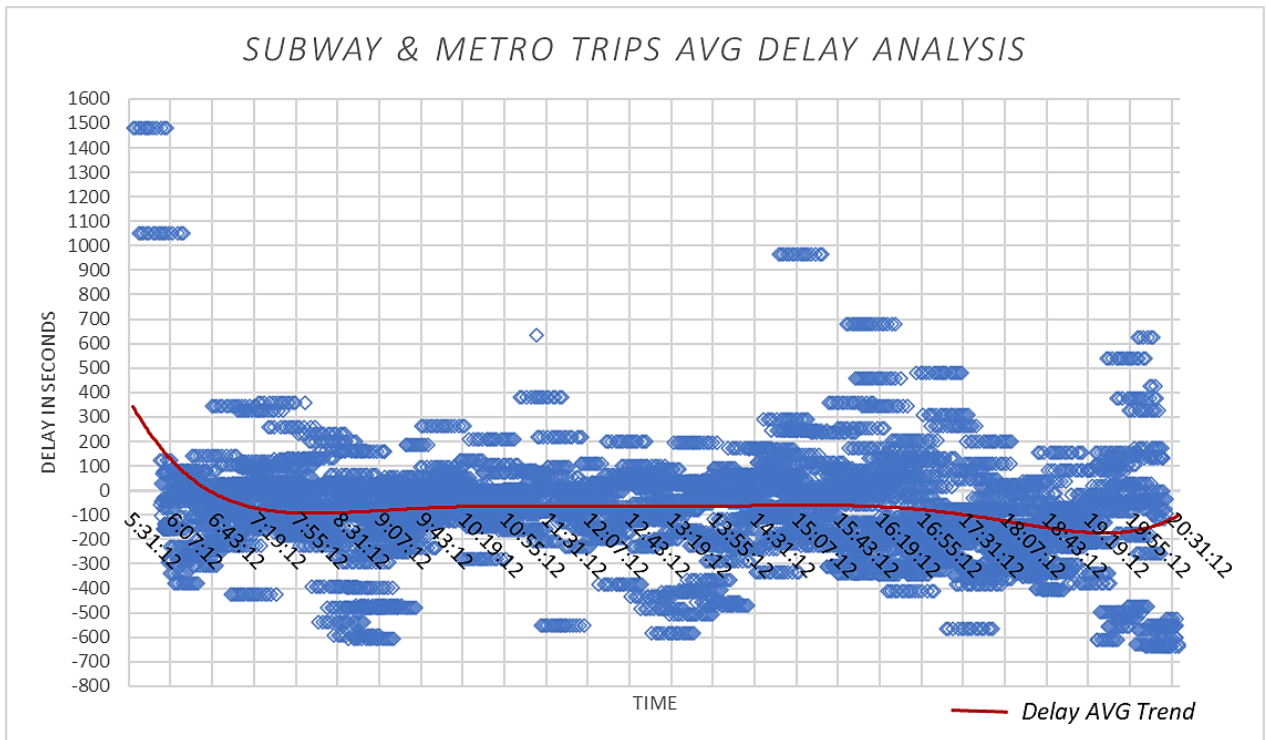


Figure 37 Subway & Metro Trips Arrival Delay Average

5.4.9 Rail Trips Delay Analysis

Most of the Rail trips are having delays between 200 and -200 seconds (Figure 38).

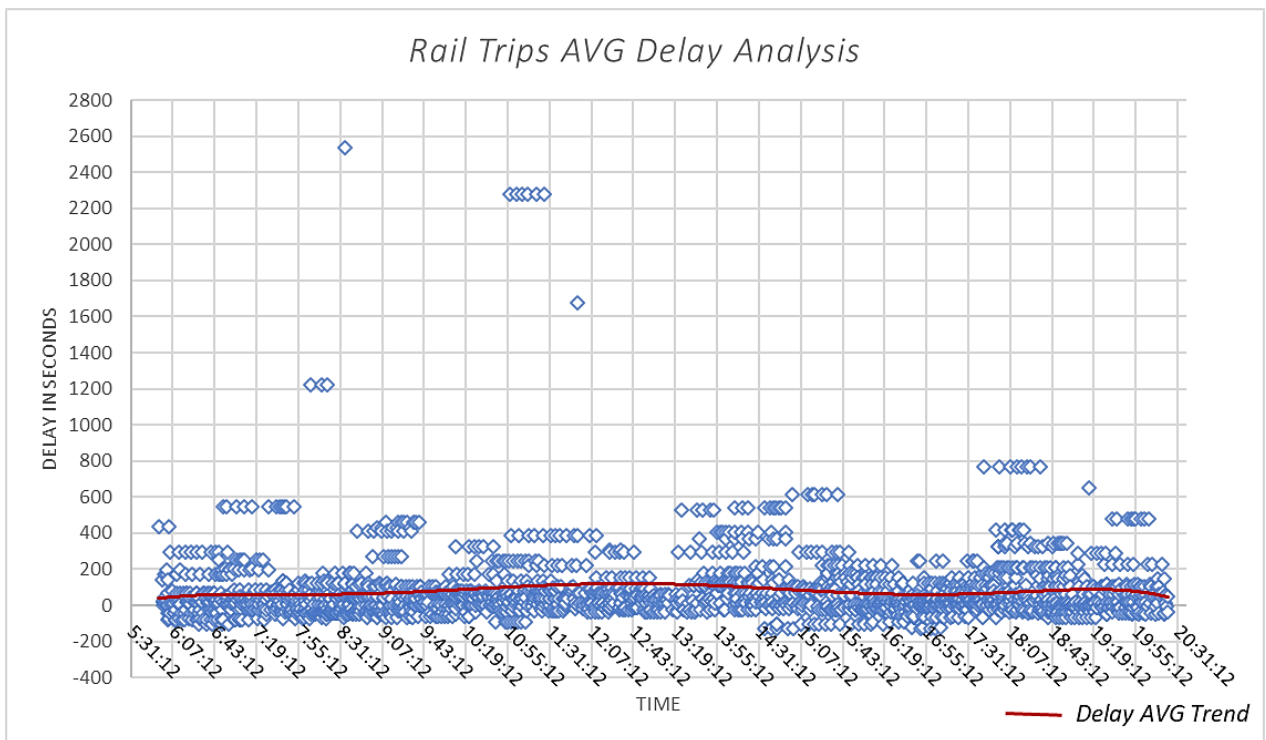


Figure 38 Rail Trips arrival delay average

5.5 Delay Data Comparison

As the delay data has a lot of outliers, further analysis has been made to compare these delays with and without outliers. These outliers are 10% of the total delay data. Next box plots and tables show a comparison of arrival and departure delays with and without outliers. Removing outliers from the data changed its distribution (-200 to 50 seconds). These box plots indicate that more than 75% of the trips have been arrived and departed earlier than their schedule (Figure 39 to Figure 42).

All Vehicles Arrival Delay with Outliers

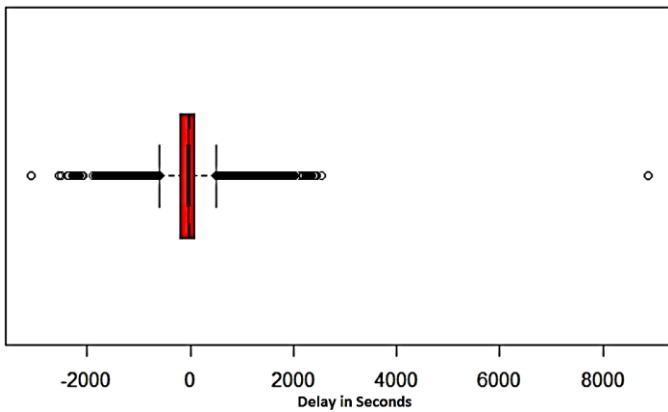


Figure 39 All vehicles arrival delay with outliers

All Vehicles Arrival Delay without Outliers

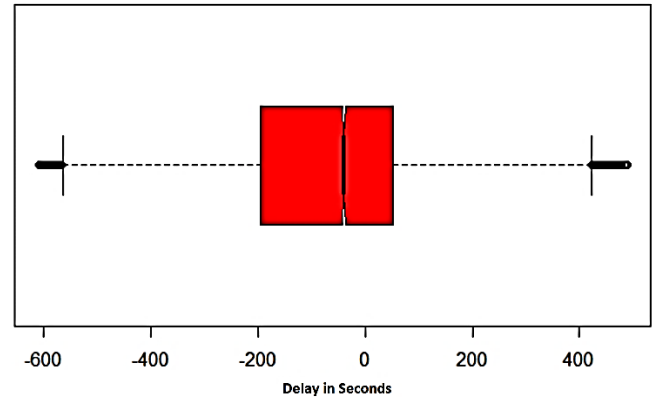


Figure 40 All vehicles arrival delay without outliers

All Vehicles Departure Delay with Outliers

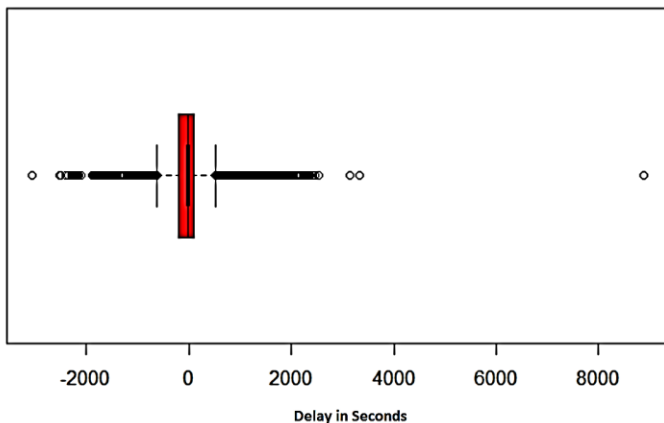


Figure 41 All vehicles departure delay with outliers

All Vehicles Departure Delay without Outliers

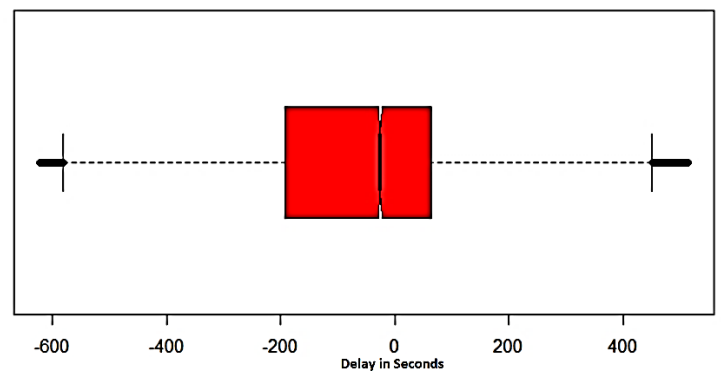


Figure 42 All vehicles departure delay without outliers

The next four box plots (Figure 43 to Figure 46) are showing the same data categorized according to vehicle type. It is noticeable, that Tram, Streetcar, Light rail & Subway, Metro have been arrived and left earlier than their schedule. Tram, Streetcar, Light rail trips delay is distributed more than Subway, Metro, and Rails trips delay.

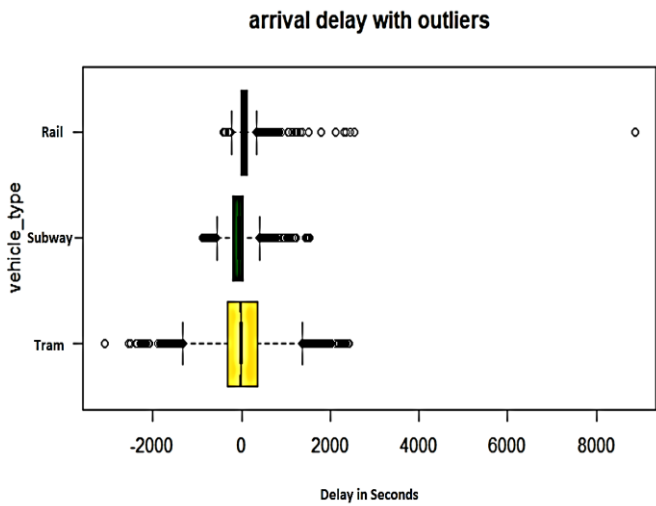


Figure 43 Arrival delay with outliers per vehicle type

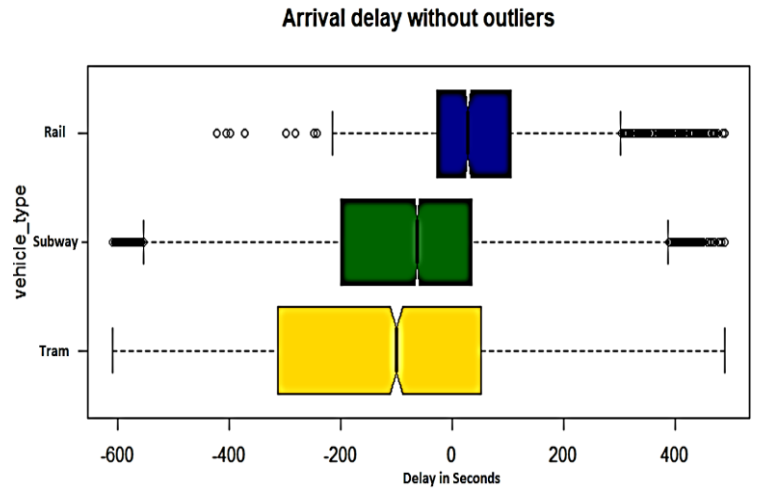


Figure 44 Arrival delay without outliers per vehicle type

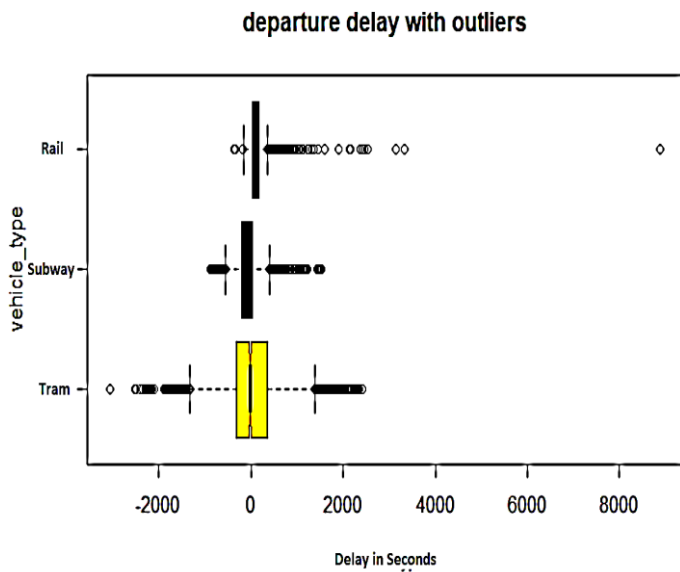


Figure 45 Departure delay with outliers per vehicle type

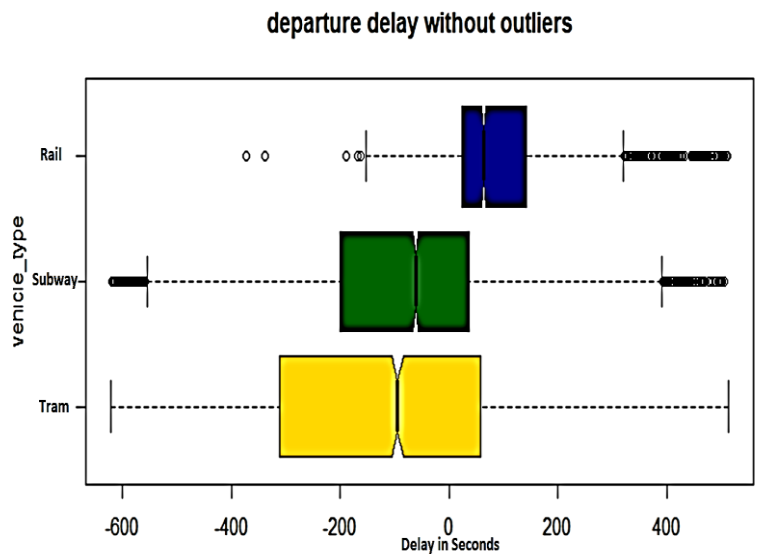


Figure 46 Departure delay without outliers per vehicle type

The following tables (Table 15 & Table 16) give detailed information about previous box plots (Figure 39 to Figure 46). In general arrival and departure, delays are almost typical before removing the outliers. And the same case after removing the outliers.

Table 15 Comparison of arrival delay data with and without outliers

Arrival							
with outliers							
All Vehicles		v0		v1		v2	
Min	-3100	Min	-3100	Min	-876	Min	-422
1st Qu.	-198	1st Qu.	-316	1st Qu.	-205	1st Qu.	-26
Median	-30	Median	-22	Median	-64	Median	32
Mean	-20.28	Mean	48.06	Mean	-78.26	Mean	81.77
3rd Qu.	77	3rd Qu.	360	3rd Qu.	37	3rd Qu.	116
Max	8874	Max	2420	Max	1529	Max	8874
without outliers							
All Vehicles		v0		v1		v2	
Min	-610	Min	-610	Min	-610	Min	-422
1st Qu.	-195	1st Qu.	-313	1st Qu.	-201	1st Qu.	-28
Median	-40	Median	-101	Median	-64	Median	27.5
Mean	-65.95	Mean	-108.1	Mean	-81.72	Mean	49.3
3rd Qu.	52	3rd Qu.	51.5	3rd Qu.	34.75	3rd Qu.	104
Max	489	Max	489	Max	488	Max	488

Table 16 Comparison of departure delay data with and without outliers

Departure							
with outliers							
All Vehicles		v0		v1		v2	
Min	-3070	Min	-3070	Min	-876	Min	-373
1st Qu.	-196	1st Qu.	-315	1st Qu.	-203	1st Qu.	24
Median	-16	Median	-21	Median	-62	Median	69
Mean	-10.56	Mean	51.75	Mean	-76.72	Mean	133.4
3rd Qu.	88	3rd Qu.	365	3rd Qu.	38.25	3rd Qu.	161
Max	8874	Max	2420	Max	1529	Max	8874
without outliers							
All Vehicles		v0		v1		v2	
Min	-622	Min	-622	Min	-620	Min	-373
1st Qu.	-194	1st Qu.	-311	1st Qu.	-201	1st Qu.	23
Median	-26	Median	-95	Median	-62	Median	64
Mean	-56.98	Mean	-101	Mean	-80.86	Mean	95.12
3rd Qu.	64	3rd Qu.	57.75	3rd Qu.	36	3rd Qu.	142
Max	514	Max	514	Max	506	Max	511

6 Summary

6.1 Discussion

Evaluating real-time transit information completeness & Accuracy to be used in trip planning is a major part of this research. As currently, many transit agencies are providing real-time data, Boston Metropolitan Area (MBTA), and Broward County were suggested as study areas. MBTA was chosen since it is bigger and has various transport modes from a single transit company (Tram, Subway, Rail, Bus & Ferry), whilst Broward County has only bus and train. Also, MBTA can be handled easier since its GTFS Realtime data is available as a protocol buffer in contrast with Broward. The whole process of real-time data collection for this research has been done on June 24th, 2020, from 5:59 A.M. until 8:25 P.M. The GTFS Static feed along with the OSM file of Boston city, were downloaded for the same period. The GTFS Static Data is updated every three months on the agency's website and can be downloaded as a zip file contains several text files. To import the GTFS Static data into the PostgreSQL database, several tables were made within the PostgreSQL database. These tables have the same names and column's data type of the text files in Static data zip file, so the Static data can be imported to these tables. Several attempts to import the Realtime data directly from the transit agency into the PostgreSQL database failed, as the Realtime data needs to be decoded first to extract information from it. This process needs more than 10 seconds per file which leads to losing files in between. Therefore, all VehiclePositions and TripUpdates feeds were downloaded for the whole period. Realtime Data (VehiclePositions and TripUpdates feed) was collected using a python script every 10 seconds during the data collection period. Thus, high collection frequency has been done to guarantee the desired accuracy in the calculation of observed arrival and departure times and to have all trip updates. After downloading these feeds, another obstacle was encountered while transforming VehiclePositions and TripUpdates feeds from binary into tabular data that can be used by the PostgreSQL database, as python (2.7) was the official version that can be used for decoding the data but it is not supported anymore. It required several tests on different python versions to finally find the only capable version, python (3.5), which could be used to decode the data. Therefore, python (3.5) was used for the entire process from connecting to the PostgreSQL database, creating the VehiclePositions and TripUpdates tables, specifying all table's columns types, encoding these feeds to extract all needed information, and finally importing this data into their exact tables' columns in the PostgreSQL database. This last process of real-time data transformation was the most time-consuming part of this project,

as the information about it was either outdated or unavailable. PostgreSQL database was used to store all the data as it supports table queries and changes, along with the linkage among tables using their unique identifiers. One of the most important benefits of importing all the feeds into a database is the computation speed, as no CSV-files browser has the same ability or can join several tables. Following the data importing phase was the *Realtime Data Filtering* phase, in which all duplicates records were dropped from both TripUpdates and VehiclePositions tables since each file has duplicated records from previous and next files. The column *current status* from the VehiclePositions table was used to indicate the vehicle status, if it is “stopped at” or “in transit to” a certain stop. When a vehicle has the status “stopped at” a specific stop the time range during being at this stop is used for calculating the observed arrival and departure times. The first timestamp was used as the observed arrival time and the last one as departure time. The observed arrival and departure times were then used to modify the original GTFS feed to be used in routes calculation. During the analysis of the collected data, five main analysis steps were performed. Starting with Descriptive Analysis of the data which is generating insightful statistics about trips, stops, and routes, also calculating the percentage of VehiclePositions and TripUpdates. There was VehiclePositions feed for 11% of trips, 85% of stops, and 67% of routes in the timetable (*stop times* table). In contrast, 12% of trips, 87% of stops, and 65% of routes had TripUpdates feeds. That means still most of the trips have no Realtime updates. Bus information of observed arrival and departure times is missing because the (current status = 1) is not provided in the VehiclePositions feeds for almost all the stops (0.19% of the total collected data) which leads to this error. On one hand, only 88 out of 1,256,589 Bus’s VehiclePositions have the (current status = 1), which is barely (0.01%). On the other hand, all other vehicles have around (25%) of VehiclePositions have the (current status = 1). Consequently, the Bus is dropped from the next analysis. The second step was showing the distribution of the VehiclePositions and TripUpdates feeds on the studied region, which exposed that most of the feeds were focusing on the center of Boston city. The third step was performing comparisons of the trips and stops between the Static and Realtime data per vehicle’s type to clarify the percentage of stops that have updates in the Realtime feeds. Surprisingly, 994 trips have VehiclePositions updates, but they are not in the Stop_times table. When these trips were investigated, they seemed like added trips during the day as all these trips have either “ADDED-“ before the trip name for the Tram, Streetcar, Light rail, Subway and Metro trips, for example (ADDED-1580448294), or the “OL” at the end of the bus trips for

instance (45043214-OL1). Next, the arrival and departure delay has been analyzed for the stops and trips in total and then was broken down for each vehicle's type. Arrival and departure delays showed similar delay distribution. Half of the delays data is between -3 and +1.5 minutes, which is far from the minimum and the maximum delays leading to huge outliers. Vehicles, in general, tend to arrive and depart early before the time scheduled from stops that have less than 80 vehicles went through. In contrast, when the stop has more than 80 vehicles went through, these vehicles arrived and departed late after their schedule. Tram, Streetcar & Light rail stops that have less than 70 vehicles went through are more likely to have vehicles arrive and leave before schedule while stops with more than 70 vehicles have delayed vehicles. Subway & Metro stops with less than 80 vehicles have vehicles arrived earlier than scheduled, whereas stops with more than 80 vehicles have delayed vehicles. Rails stops have earlier arrival and departure times than scheduled times. Delay information average was calculated for each trip. Then, these trips were categorized per vehicle type. Trips that used Tram, Streetcar & Light rail were delayed at three periods of the day, the beginning of the day around 6:00 am, the middle of the day around noon, and at the end of the working day around 7:00 pm. These periods are more likely related to the working hours, beginning and end of work times, and the break in the middle of the day. Most of the Rail trips are having delays between 200 and -200 seconds. Finally, a comparison was performed between the delay data with and without outliers to show the delay measurements were affected by outliers. As the delay data has a lot of outliers, further analysis has been made to compare these delays with and without outliers. These outliers are 10% of the total delay data. Removing outliers from the data changed its distribution to be limited between -200 and 50 seconds.

6.2 Conclusion

Analyzing the accuracy and effectiveness of GTFS transit feeds was the main goal of this research. GTFS has Static and Realtime transit feeds. GTFS Static data is the timetable of scheduled trips, while GTFS Realtime feeds are divided into *VehiclePositions* and *TripUpdates*. *VehiclePositions* feeds consist of the current positions of each vehicle as well as information about the trip it is currently on, and the stop it is going to. While, *TripUpdates* feeds have predictions about future delays, which are used by passengers for trip planning. Currently, many transit agencies are providing real-time data. Boston Metropolitan Area (MBTA) and Broward

County were suggested as study areas. MBTA was chosen since it covers a bigger area and has various transport modes from a single transit company (Tram, Subway, Rail, Bus & Ferry), whilst Broward County has only bus and train.

The whole process of Realtime data collection for this research has been done on June 24th, 2020, from 5:59 A.M. until 8:25 P.M. The GTFS Static feed along with the OSM file of Boston city, were downloaded for the same period. Importing the Realtime data directly from the transit agency into the PostgreSQL database failed, as the Realtime data needs to be decoded first to extract information from it, and this process needs more than 10 seconds per file which leads to losing files in between since VehiclePositions and TripUpdates are updated every 10 seconds. Therefore, all VehiclePositions and TripUpdates feeds were downloaded every 10 seconds during the data collection period. Thus, high collection frequency has been done to guarantee the desired accuracy in the calculation of observed arrival and departure times and to have all trip updates. The only capable python version which could be used to decode the data is (3.5).

To evaluate the quality and effectiveness of Realtime data, both Static and Realtime feeds are stored in a local database. PostgreSQL database was used to store all the data as it supports table queries and changes, along with the linkage among tables using their unique identifiers. One of the most important benefits of importing all the feeds into a database is the computation speed, as no CSV-files browser has the same ability or can join several tables. After the *Data Importing* phase was the *Realtime Data Filtering* phase, in which all duplicates records were dropped from both TripUpdates and VehiclePositions tables since each file in these feeds has duplicated records from previous and next files. The column *current status* from the VehiclePositions table was used to indicate the vehicle status, if it is “stopped at” or “in transit to” a certain stop. When a vehicle has the status “stopped at” a specific stop the time range during being at this stop is used for calculating the observed arrival and departure times. The first timestamp was used as the observed arrival time and the last one as departure time.

During the analysis of the collected data, five main analysis steps were performed. Starting with Descriptive Analysis of the data which is generating insightful statistics about trips, stops, and routes, also calculating the percentage of VehiclePositions and TripUpdates. There was VehiclePositions feed for 11% of trips, 85% of stops, and 67% of routes in the timetable (stop times table). In contrast, 12% of trips, 87% of stops, and 65% of routes had TripUpdates feeds. That means still most of the trips have no Realtime updates which make using the Realtime data

for trip planning still not possible. Also, Bus information of observed arrival and departure times is missing, as the (current status = 1) is not provided in the VehiclePositions feeds for almost all the stops (0.19% of the total collected data) which lead to this error. On one hand, only 88 out of 1,256,589 Bus's VehiclePositions have the (current status = 1), which is barely (0.01%). On the other hand, all other vehicles have around (25%) of VehiclePositions have the (current status = 1). Consequently, the Bus which has 94% of the data has been dropped from the analysis and made the trip planning using the Realtime feeds impossible. The second step was showing the distribution of the VehiclePositions and TripUpdates feeds on the studied region, which exposed that most of the feeds were focusing on the center of Boston city. The third step was performing comparisons of the trips and stops between the Static and Realtime data per vehicle's type to clarify the percentage of stops that have updates in the Realtime feeds. Surprisingly, 994 trips have VehiclePositions updates, but they are not in the Stop_times table. When these trips were investigated, they seemed like added trips during the day as all these trips have either "ADDED-" before the trip name for the Tram, Streetcar, Light rail, Subway and Metro trips, for example (ADDED-1580448294), or the "OL" at the end of the bus trips for instance (45043214-OL1). Next, the arrival and departure delay has been analyzed for the stops and trips in total and then was broken down for each vehicle's type. Arrival and departure delays showed similar delay distribution. Half of the delays data is between -3 and +1.5 minutes, which is far from the minimum and the maximum delays leading to huge outliers. Vehicles, in general, tend to arrive and depart early before the time scheduled from stops that have less than 80 vehicles went through. In contrast, when the stop has more than 80 vehicles went through, these vehicles arrived and departed late after their schedule. Stops that have less than 70 vehicles went through are more likely to have vehicles arrive and leave before the schedule while stops with more than 70 vehicles have delayed vehicles. Subway & Metro stops with less than 80 vehicles have vehicles arrived earlier than scheduled, whereas stops with more than 80 vehicles have delayed vehicles. Rails stops have earlier arrival and departure times than scheduled times. Delay information average was calculated for each trip. Then, these trips were categorized per vehicle type. Trips that used Tram, Streetcar & Light rail were delayed at three periods of the day, the beginning of the day around 6:00 am, the middle of the day around noon, and at the end of the working day around 7:00 pm. These periods are more likely related to the working hours, beginning and end of work times, and the break in the middle of the day. Most of the Rail trips are having delays between 200 and -200 seconds. Finally, a comparison was

performed between the delay data with and without outliers to show the delay measurements were affected by outliers. As the delay data has a lot of outliers, further analysis has been made to compare these delays with and without outliers. These outliers are 10% of the total delay data. Removing outliers from the data changed its distribution to be limited between -200 and 50 seconds.

To conclude, GTFS Realtime feeds still in a development phase and cannot be used for trip planning in this stage as most of the trips do not have Realtime feeds (89%). Additionally, on trips that have Realtime updates either there is missing information of stops during these trips or the feeds cannot be used according to missing critical information (Bus case). As the delay data has a lot of outliers, further analysis has been made to compare these delays with and without outliers. These outliers are 10% of the total delay data. Removing outliers from the data changed its distribution to be limited between -200 and 50 seconds. It is noticeable, that Tram, Streetcar, Light rail (v 0) & Subway, Metro (v1) have been arrived and left earlier than their schedule in general. Whilst, Tram, Streetcar, Light rail trips delay is distributed more than Subway, Metro, and Rails (v2) trips delay.

6.3 Future work

Although GTFS Static and Realtime feeds need expensive infrastructure, they are published by many agencies around the world nowadays. GTFS Realtime feeds would offer a great opportunity for developers to create trip planning and tourism applications if it was established correctly. The collected Realtime data shows quality issues and missing critical information, where vehicles are not following the correct stop sequence, not having the observed arrival and departure time, or having them away from the stop location. Many trips are having Realtime data without any Static data to be compared with. Even though GTFS Realtime feeds are still having these issues, it is developing and could be more accurate and complete in the future. VehiclePositions feeds could be used to monitor the transportation flow and develop the entire transportation system, the timetable, and even the infrastructure using the densest areas used by vehicles. GTFS Realtime feeds also could be used for delay analysis and environmental pollution control. In the end, it is a great open way to offer Realtime information which needs more development to be useful.

7 References

- Antrim, A., Barbeau, S.J. (2013) 'THE MANY USES OF GTFS DATA – OPENING THE DOOR TO TRANSIT AND MULTIMODAL APPLICATIONS', 24.
- Basic Tutorial - OpenTripPlanner [online] (2020) available: <http://docs.opentripplanner.org/en/latest/Basic-Tutorial/> [accessed 9 Feb 2020].
- Broward County Transit [online] (2020) available: <https://www.broward.org/BCT/Pages/default.aspx> [accessed 22 Jan 2020].
- Chien, S.I.J., Ding, Y., Wei, C. (2002) 'Dynamic Bus Arrival Time Prediction with Artificial Neural Networks', *Journal of Transportation Engineering*, 128(5), 429–438.
- Developer Guide | Protocol Buffers [online] (2020) *Google Developers*, available: <https://developers.google.com/protocol-buffers/docs/overview> [accessed 9 Feb 2020].
- Dictionary.Com [online] (2020) *www.dictionary.com*, available: <https://www.dictionary.com/browse/timetable> [accessed 22 Jan 2020].
- GraphBuilder [online] (2014) *GitHub*, available: <https://github.com/opentripplanner/OpenTripPlanner> [accessed 9 Feb 2020].
- GraphStructure [online] (2015) *GitHub*, available: <https://github.com/opentripplanner/OpenTripPlanner> [accessed 8 Feb 2020].
- GTFS Realtime Overview | Realtime Transit [online] (2020) *Google Developers*, available: <https://developers.google.com/transit/gtfs-realtime> [accessed 25 Jan 2020].
- Hickman, M.D., Wilson, N.H.M. (1995) 'Passenger travel time and path choice implications of real-time transit information', *Transportation Research Part C: Emerging Technologies*, 3(4), 211–226.
- Hillsman, E., Barbeau, S.J. (2011) *Enabling Cost-Effective Multimodal Trip Planners through Open Transit Data*, University of South Florida, Tampa, FL, available: https://scholarcommons.usf.edu/cutr_nctr/130 [accessed 24 Jan 2020].
- Introduction to Tidytransit [online] (2019) available: <https://cran.r-project.org/web/packages/tidytransit/vignettes/introduction.html> [accessed 26 Jan 2020].
- Jariyasunant, J., Work, D.B., Kerkez, B., Sengupt, R., Glaser, S., Bayen, A. (2010) 'Mobile Transit Trip Planning with Real-Time Data', *the Annual Meeting*, 17.
- Kaufman, S.M. (2012) 'Getting Started with Open Data: A Guide for Transportation Agencies', available: <https://trid.trb.org/view/1143018> [accessed 23 Jan 2020].

- Kroon, L.G., Schöbel, A., Wagner, D. (2016) ‘Algorithmic Methods for Optimization in Public Transport’, 22.
- Lin, Y., Yang, X., Zou, N., Jia, L. (2013) ‘Real-Time Bus Arrival Time Prediction: Case Study for Jinan, China’, *Journal of Transportation Engineering*, 139, 1133–1140.
- Lyoen, C., van de Ven, T., Karin, K.L. (2010) ‘Study regarding guaranteed access to traffic and travel data and free provision of universal traffic information’, *EUROPEAN COMMISSION Directorate-General Mobility and Transport*, 80.
- MARTA Developer Resources [online] (2019) available: <http://www.joederose.us/MARTA/Data/> [accessed 26 Jan 2020].
- Masina, R. (2019) Protocol Buffers 101 [online], available: http://ruthwik.github.io/other/2019-03-22-protocol_buffers/ [accessed 8 Feb 2020].
- Mass Transit [online] (2017) *Encyclopedia Britannica*, available: <https://www.britannica.com/topic/mass-transit> [accessed 22 Jan 2020].
- ‘MBTA: App judgment’ (2020) *Boston.com*, available: http://archive.boston.com/bostonglobe/editorial_opinion/editorials/articles/2009/09/05/mbta_app_judgment/ [accessed 6 Feb 2020].
- MBTA [online] (2020) available: <https://www.mbta.com/mbta-at-a-glance> [accessed 10 Feb 2020].
- Open Data Handbook [online] (2019) *Open Data Handbook*, available: <https://opendatahandbook.org/guide/en/what-is-open-data/> [accessed 23 Jan 2020].
- OpenMobilityData - Public Transit Feeds from around the World [online] (2020) available: <https://transitfeeds.com/> [accessed 19 Oct 2020].
- OpenTripPlanner [online] (2020) available: <http://docs.opentripplanner.org/en/latest/> [accessed 8 Feb 2020].
- Osmconvert - OpenStreetMap Wiki [online] (2020) available: <https://wiki.openstreetmap.org/wiki/Osmconvert> [accessed 24 Mar 2020].
- Oxford Advanced Learner’s Dictionary [online] (2020) available: <https://www.oxfordlearnersdictionaries.com/definition/english/public-transport> [accessed 22 Jan 2020].
- Soares, I., Martins, P.M. (2013) ‘PUBLIC TRANSPORT STANDARDIZATION’, available: http://labs.integra-travel.eu/files/paper_standards_WCTR.pdf.

- South Tyrol Free Software Conference (2019a) *SFScon19 - Rafael Aguilar - Digitransit & Open Trip Planner* [online], available: <https://www.slideshare.net/SFScon/sfscon19-rafael-aguilar-digitransit-open-trip-planner> [accessed 19 Oct 2020].
- South Tyrol Free Software Conference (2019b) *Digitransit & Open Trip Planner* [online], available: <https://www.slideshare.net/SFScon/sfscon19-rafael-aguilar-digitransit-open-trip-planner> [accessed 26 Jan 2020].
- Steiner, D. (2014) ‘Evaluating the effectiveness of realtime information in multimodal public transport trip planning’.
- Steiner, D., Hochmair, H.H., Paulus, G. (2015) ‘Quality Assessment of Open Realtime Data for Public Transportation in the Netherlands’.
- Subway | Schedules & Maps | MBTA [online] (2020) available: <https://www.mbtta.com/schedules/subway> [accessed 10 Feb 2020].
- Sun, F., Pan, Y., White, J., Dubey, A. (2016) ‘Real-time and predictive analytics for smart public transportation decision support system’, in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, IEEE, 1–8.
- Transit Cooperative Research Program, Transportation Research Board, National Academies of Sciences, Engineering, and Medicine (2011) *Use and Deployment of Mobile Device Technology for Real-Time Transit Information* [online], Transportation Research Board: Washington, D.C., available: <http://www.nap.edu/catalog/13323> [accessed 6 Feb 2020].
- Weyrer, T.N., Hochmair, H.H., Paulus, G. (2013) ‘Intermodal Door-to-Door Routing for People with Physical Impairments in a Web-based Open Source Platform’.