Pontifícia Universidade Católica
do Rio de Janeiro

Kathrin Rodríguez Llanes

Bus Network Analysis and Monitoring

Tese de Doutorado

Thesis presented to the Programa de Pós-graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências - Informática.

Advisor: Prof. Marco Antonio Casanova

Rio de Janeiro
May 2017

Pontifícia Universidade Católica
do Rio de Janeiro

Kathrin Rodríguez Llanes

Bus Network Analysis and Monitoring

Thesis presented to the Programa de Pós-graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências - Informática. Approved by the undersigned Examination Committee.

Prof. Marco Antonio Casanova
Advisor
Departamento de Informática – PUC-Rio

Prof. Hélio Cortes Vieira Lopes
Departamento de Informática – PUC-Rio

Prof. Markus Endler
Departamento de Informática – PUC-Rio

Prof. José Antonio Fernandes de Macêdo
UFC

Prof. Fábio André Machado Porto
LNCC

Prof. Marcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, May 26th, 2017

Kathrin Rodríguez Llanes

Kathrin Rodríguez Llanes holds a master's degree in computer science from Informatics Science University (UCI) in Havana, Cuba, and a computer engineering degree from UCI. Kathrin worked as a system analyst, architect and software developer at the Engineering Automation Cell (CAE) of Tecgraf Institute (PUC-Rio). She possesses the assistant professor academic category. She was professor of Mathematics (2007-2009), Data Base (2009-2011) and Software Engineering (2011-2012) at UCI. His current research focuses on Traffic Modeling, Intelligent Transportation System, Big Data, Data Mining, and Data Stream Processing.

Bibliographic data

Rodríguez Llanes, Kathrin

Bus Network Analysis and Monitoring / Kathrin Rodríguez Llanes; advisor: Marco Antonio Casanova. Rio de Janeiro: PUC-Rio, Departamento de Informática – 2017.

v., 137 f.: il. ; 29,7 cm

1. Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2017.

Inclui bibliografia

1. Informática – Teses. 2. Detecção de anomalias no trânsito. 3. Estimativa do tempo de viagem. 4. Mineração de dados de trajetórias. 5. Redes de ônibus. I. Casanova, Marco Antonio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CCD: 004

I dedicate this thesis to people whom I love most in the world:

My grandparents, my parents, and my sister.

Acknowledgments

My special thanks are due to Professor Dr. Marco Antonio Casanova, who besides being my advisor and mentor, was my teacher in several subjects. His organization, dedication and his great capacity of work have represented for me a constant inspiration source during all this time. I thank him for his guidance, encouragement and for providing me excellent ideas during the development of this work. Without his guidance and persistent help, this dissertation would not have been possible. Actually, I am very pleased to have done my PhD under his mentoring.

I want to express sincere gratitude to my committee members: Prof. Hélio Cortes Vieira Lopes, Prof. José Antonio Fernandes de Macêdo, Prof. Fábio André Machado Porto, and Prof. Markus Endler, who have provided, with kindness, their insight, and suggestions, which are precious to me.

Especially, I thank Prof. Hélio Cortes Vieira Lopes for his greats ideas and his collaboration in writing papers. I am also thankful with Prof. José Antonio Fernandes de Macêdo for his collaboration in writing papers, providing valuable comments and suggestions.

In addition, a thank you to Prof. Eugenio Epprecht for numerous helpful advices and inspiring discussions about probabilistic issues, for his always cheerful spirit and his willingness to give me his help whenever I needed it. Dr. Eugenio Epprecht has directly involved with aspects of Chapter 6.

My sincere thanks go to my friend, the PhD candidate Noel Moreno, who is always available to help, discuss and collaborate, for his very helpful feedback and advice during the development of this work.

Thank you to my friend, the PhD. candidate Lívia Ruback for her insight and patience and for her good advice in times of stress.

I thank the Pontifical Catholic University of Rio de Janeiro, especially the Department of Informatics (DI) for giving me the opportunity to improve my academic, professional and scientific education and to meet colleagues from different countries, even from my own country and become us a family, where we support each other and share moments of tension and happiness.

Thank the rest of my friends, lab-mates, and the welcoming people of DI for their kindness and friendship.

Thank each and every teacher and department officials that helped me during this time.

I would also like to thank CAPES and CNPq Foundations for their financial support granted through doctoral fellowship.

I would like to express my eternal gratitude to my parents and my sister for their everlasting love and support.

To all who contributed to the achievement of this objective, I am forever grateful.

Abstract

Rodríguez Llanes, Kathrin; Casanova, Marco Antonio (Advisor). **Bus Network Analysis and Monitoring**. Rio de Janeiro, 2017. 137p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Buses, equipped with active GPS devices that continuously transmit their position, can be understood as mobile traffic sensors. Indeed, bus trajectories provide a useful data source for analyzing traffic in the bus network of a city, if the city traffic authority makes the bus trajectories available openly, timely and in a continuous way. In this context, this thesis proposes a bus GPS data-driven approach for analyzing and monitoring the bus network of a city. It combines graph algorithms, geospatial data mining techniques and statistical methods. The major contribution of this thesis is a detailed discussion of key operations and algorithms for modeling, analyzing and monitoring bus network traffic, specifically: (1) modelling, analyzing, and segmentation of the bus network; (2) mining the bus trajectory dataset to uncover traffic patterns; (3) detecting traffic anomalies, classifying them according to their severity, and estimating their impact; (4) maintaining and comparing different versions of the bus network and traffic patterns to help urban planners assess changes. Another contribution is the description of experiments conducted for the bus network of the City of Rio de Janeiro, using bus trajectories obtained from June 2014 to February 2017, which have been made available by the City Hall of Rio de Janeiro. The results obtained corroborate the usefulness of the proposed approach for analyzing and monitoring the bus network of a city, which may help traffic managers and city authorities improve traffic control and urban mobility plans.

Keywords

Bus Networks; Detection of Traffic Anomalies; Trajectory Data Mining; Travel Time Estimation

Resumo

Rodríguez Llanes, Kathrin; Casanova, Marco Antonio. **Análise e Monitoramento de Redes de Ônibus.** Rio de Janeiro, 2017. 137p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Ônibus, equipados com dispositivos GPS ativos que transmitem continuamente a sua posição, podem ser entendidos como sensores móveis de trânsito. De fato, as trajetórias dos ônibus fornecem uma fonte de dados útil para analisar o trânsito na rede de ônibus de uma cidade, dado que as autoridades de trânsito da cidade disponibilizem as trajetórias de forma aberta, oportuna e contínua. Neste contexto, esta tese propõe uma abordagem que usa os dados de GPS dos ônibus para analisar e monitorar a rede de ônibus de uma cidade. Ela combina algoritmos de grafos, técnicas de mineração de dados geoespaciais e métodos estatísticos. A principal contribuição desta tese é uma definição detalhada de operações e algoritmos para analisar e monitorar o tráfego na rede de ônibus, especificamente: (1) modelagem, análise e segmentaçãoda rede de ônibus; (2) mineração do conjunto de dados de trajetória de ônibus para descobrir padrões de tráfego; (3) detecção de anomalias de trânsito, classificação de acordo com sua gravidade, e avaliação do seu impacto; (4) manutenção e comparação de diferentes versões da rede de ônibus e dos seus padrões de tráfego para ajudar os planejadores urbanos a avaliar as mudanças. Uma segunda contribuição é a descrição de experimentos realizados para a rede de ônibus da Cidade do Rio de Janeiro, utilizando trajetórias de ônibus correspondentes ao período de junho de 2014 até fevereiro de 2017, disponibilizadas pela Prefeitura do Rio de Janeiro. Os resultados obtidos corroboram a utilidade da abordagem proposta para analisar e monitorar a rede de ônibus de uma cidade, o que pode ajudar os gestores do trânsito e as autoridades municipais a melhorar os planos de controle de trânsito e de mobilidade urbana.

Palavras-chave

Detecção de anomalias no trânsito;Estimativa do tempo de viagem;Mineração de dados de trajetórias;Redes de Ônibus.

Summary

List of Figures

List of Tables

List of Acronyms

BTN	Bus Transport Network
BRT	Bus Rapid Transit
CSV	Comma-separated Values
DBMS	Data Base Management System
GIS	Geographic Information System
GPS	Global Positioning System
GTFS	General Transit Feed Specification
NoSQL	Not only Structured Query Language
OSM	Open Street Map
PDF	Probability Density Function

1
Introduction

1.1.
Motivation

The ever-increasing concentration of urban populations has long forced city authorities to face problems associated with the growing demand for public services, such as inadequate transportation systems, inadequate city services, and increasing pollution. To address these issues, a number of *Smart City* projects have been started in the recent past (CARAGLIU et al., 2011; "The SMARTY project," 2013; ANASTASI et al., 2013).

Specifically, public transportation systems affect people in their daily routine and, for this reason, must be efficiently implemented. Buses are among the most popular public transportation systems, but obviously, have a strong interdependency with traffic conditions and, therefore, may result in a considerable waste of time by quite a large number of citizens. In this context, the bus travel time has been considered as an important quantitative measure of bus public transportation service quality. The quality of the service will deteriorate if the in-bus time or the waiting time of passengers increase (YE et al., 2015).

The fact that traffic managers can continuously monitor the travel time in the bus network helps improve traffic conditions, reduce travel times, avoid traffic congestions and act quickly when a traffic anomaly occurs. In such situations, this continuous monitoring allows city authorities not only to act quickly but alsoto take responsible actions to return the system to normal conditions.

In recent years, due to the rapid advent of wireless communication and positioning technologies, there is a tendency to use GPS data to monitor and control traffic, especially to estimate traffic behavior (ZHANG et al., 2013; ZHU; XU, 2015) and detect the occurrence of anomalies (CHEN et al., 2011; KUANG et al., 2015). One of the main advantages of GPS devices is that they represent a

promising tool for obtaining reliable data, because of their accuracy in recording time and their precision in capturing the geolocation (SHEN; STOPHER, 2014).

In major metropolitan cities, buses are equipped with GPS devices. They operate continuously for nearly 24 hours per day. Then, buses, equipped with active GPS devices that continuously transmit their position, can be understood as mobile traffic sensors. These buses generate a huge amount of data in the form of raw trajectories, that include: date, time of day, instant speed, geo-location, and so on. Thus, a *raw bus trajectory* is a continuous data stream acquired from such a GPS device. Bus trajectories provide a useful data source for analyzing traffic in the bus network, if the city traffic authority makes the bus trajectories available openly, timely and in a continuous way. Under such conditions, bus trajectories are a better data source to analyze traffic in the bus network than data generated by proprietary traffic applications that acquire the position of private cars and that depend on drivers' volunteered traffic feedback, such as Waze(WAZE MOBILE, 2013). Indeed, bus trajectories are a stable data source, in the sense that they cover the same set of streets, at predictable regular intervals, if traffic conditions permit. In fact, this is the point: if the buses in a given area are not running according to the usual schedule, then a traffic perturbation is the most probable cause. Furthermore, if stored in an adequate way, bus trajectories will provide, over time, a historical picture of how the city evolved, much in the same way as satellite imagery gives a historical picture of how an urban area grew.

On the one hand, the benefits of monitoring traffic in bus networks using bus travel time and, on the other hand, the advantage of using data from GPS installed on buses, motivate the development of intelligent control and management solutions that use a bus GPS data-driven approach for monitoring, controlling and analyzing the traffic conditions in the bus network.

Additionally, the specific case of public transportation system in the City of Rio de Janeiro is historically deficient, mainly because it is based on an old bus system. However, from a total of approximately 6.3 million people in its metropolitan area, the public transportation system based on buses carries almost 60% of all passengers transported daily. Therefore, the structure and properties of such transportation system have substantial implications for urban planning and public policies for the sustainable development of the city.

In order to eliminate this historical deficiency, the City Hall implemented some automation initiatives, such as the development of an open data project that exposes on the Web since 2014, at almost every minute, the GPS instant position of all buses operating in the city (MATHEUS; RIBEIRO, 2014). Although it is not a new technology, it is the first initiative to be developed in Rio (AMARAL, DO et al., 2016). This open data project, unleashes a set of challenges for academics, researchers and practitioners related to how to use them for enhancing public transportation service quality in the City of Rio de Janeiro.

In this context, this thesis proposes a bus GPS data-driven approach for analyzing and monitoring the bus network of a city. It combines graph algorithms, geospatial data mining techniques and statistical methods. To validate the proposal, experiments for the bus network of the City of Rio de Janeiro, using bus GPS data obtained from June 2014 to December 2016, which have been made available by the City Hall of Rio de Janeiro have been conducted.

1.2. Problem Statement

The motivation of the thesis leads to the broader research problems that arise when building an approach for monitoring, analyzing, and controlling the traffic in a bus network, based on bus GPS data.

The first challenging problem is analyzing the data from the specification of bus routes serving the city and managing the GPS data generated by buses operating on such routes. There are some problems with respect to the errors and completeness of the bus GPS observations. For instance, some samples lack of the timestamp, latitude and longitude, bus identifier, etc. The entries, in which those attributes are missing, should be discarded automatically, because such information is indispensable for monitoring purposes. Regarding the entries lacking the bus line number, they are never ruled out because they are valid even without the line for various purposes. Thus, the pre-processing procedures should identify and remove invalid GPS points, and should ensure that only consistent and non-duplicate entries are stored.

For providing an effective analysis of bus GPS data and better understand the bus travel behavior, it is important to understand the relationship between bus trajectory data and the city road network map. In that sense, new challenges associated to the modelling, analyzing, and segmentation of bus network are exhibited.

The estimation of travel time that buses take to traverse the road segments of the bus network and the discovering of travel time patterns are better related to the extraction of knowledge from the massive collection of historical bus trajectory data. Other challenges, associated with the monitoring task, are: how to detect traffic anomalies, classify them according to the severity degree, and estimate their impact.

The facts that bus network can change its structural features and that the bus routes can also be modified pose the following challenges: how to maintain and compare different versions of the bus network and travel time patterns to help city planners assess changes.

1.3. Thesis Contributions

In the context of the problems stated above, the first and the major contribution of this thesis is a detailed discussion of key operations and algorithms for analyzing and monitoring bus network traffic, specifically: (1) modelling, analyzing, and segmenting the bus network (Chapter 4); (2) mining the bus trajectory dataset to uncover traffic patterns (Chapter 5); (3) detecting traffic anomalies, classifying them according to their severity, and estimating their impact (Chapter 6); (4) maintaining and comparing different versions of the bus network and traffic patterns to help urban planners assess changes (Sections 4.4 and 5.4).

The second contribution is the description of experiments conducted for the bus network of the City of Rio de Janeiro, using bus trajectories corresponding to the period from June 2014 to February 2017, which have been made available by the City Hall of Rio de Janeiro. The experiments include the detection of traffic anomalies that affected the traffic in the city in this period and the impact evaluation, in terms of travel time, of the implementation of a set of bus network

changes mostly to facilitate the urban mobility during the Rio 2016 Olympic Games. In general, the results of the experiments corroborate the usefulness of the proposed approach for analyzing and monitoring the bus network of a city.

The results of this thesis demonstrate the value of open urban data for understanding and monitoring the traffic conditions in the bus network, evaluating the bus performance and assessing the variations in bus travel time patterns. The results also reveal that, free access to the open data enables researchers and the general public to explore sustainable solutions for enhancing the public transportation, and thus, improve passenger comfort.

The results in this thesis were published in conferences in the areas of Databases and Information Systems Integration, Sensor Data, Data Mining and Intelligent Transportation System. The investigation on the problem of the publishing of sensor data on the Web, in a standardized and enriched way so that they can be used by other applications, with the minimum of understanding the details, was published in (Llanes et al. 2015; Llanes et al. 2016b). Our work about the monitoring of traffic conditions using bus GPS data was published in (Llanes, et al. 2016a). Our research in the analysis of the impact of bus network changes on bus travel time patterns was published in (LLANES et al., 2017).

1.4. Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 discusses related work. Chapter 3gives an overview of the proposed approach. Chapter 4 deals with bus network modeling and analyzing. Chapter 5presents the strategy for mining the bus trajectory dataset to uncover travel time patterns. Chapter 6 describes methods for detecting anomalies, classifying them according to their severity, and estimating their impact. Chapter 7covers experiments with real data and discusses the results. Finally, Chapter 8 concludes the thesis and recommends future work.

2
Related Work

The analysis of traffic conditions and trajectory data management are very active research areas and, thus, several works have been developed. In this chapter, we review work which is closely related to our study, focused on four topics: (i) segmentation of raw trajectories; (ii) estimation of traffic patterns from GPS data; (iii) detection of traffic anomalies and (iv) evaluation of the impact of traffic anomalies.

The segmentation of raw trajectories section discusses different criteria to segment trajectories, which are defined as a finite set of timestamped positions collected by a mobile device. The estimation of traffic patterns from GPS data section presents the main techniques to discover traffic behavioral patterns from data recorded by GPS devices. The detection of traffic anomalies section describes methods to identifywhen the traffic patterns deviate from the typical patterns. Finally, the evaluation of the impact of traffic anomalies section reviews studies that focus on predicting the impact and measuring the impact after the anomaly occurred.

2.1.
Segmentation of raw trajectories

There are different criteria to segment raw trajectories. They range from the transportation means used (Biljecki et al. 2013), potential-transition locations (e.g. bus stops) (LIAO et al., 2006), geo-spatiotemporal information (Buchin et al. 2011), detection of similar sub-trajectories (Sankararaman et al. 2013) and movement states(Alewijnse 2013;Alewijnse et al. 2014).

Related to the transportation means segmentation criterion, the common background is a characterization of each mode of transportation, basically in terms of typical speed (e.g. walking speed is less than 5km per hour), motion continuity

(e.g., a bus makes stops while a taxi does not), and direction and route constraints (e.g., the bus network uses only some defined routes while a tram network uses only rail tracks). For instance, (LIAO et al., 2005) proposes a Gaussian mixture model to segment the trajectory based on three-speed ranges: walking, low speed, and high speed. Thus, whenever a switch in the speed range is detected, a new segment is determined, which is associated with a mode of transport which is different from the previous one. A similar approach that uses speed change rate is developed in (ZHENG et al., 2010). The authors apply a combination of techniques: supervised learning, decision tree inference, and a post-processing step to improve the accuracy of the segmentation. Other works use semantic information such as roads categories (e.g. running and bike paths) and public transport networks (e.g. bus stops and train stations) to determine whether a trajectory segment is traveled by walking, bike, car, bus, or train (YAN et al., 2011). For the same purpose, there are more complex works based on methods such as hidden Markov model (REDDY et al., 2010; ZHENG et al., 2010; WAGA et al., 2012), neural networks (GONZALEZ et al., 2010) and fuzzy logic (XU et al., 2010).

The trajectory segmentation using potential-transition locations is closed related to the previous criterion since it also helps to identify the transportation modes used in a trajectory. To segment multi-modal trajectories,(LIAO et al., 2006) analyzes the proximity to potential-transition locations such as bus stops and parking lots. However, because of the distance threshold defined to ensure the moving object is near to a potential-transition location is small, this approach does not work well in areas with dense traffic characteristics, where the distance between potential transition points for various modes may be in the range of the GPS error. Therefore, this deficiency forces the method to be used only partially in order to discern between cars, buses, and trams. Some studies (ZHENG et al., 2010; DAS; WINTER, 2016) ensure that a person often walks or stops during the transition. Biljeckiin F. in his thesis (BILJECKI, 2010) corroborates this statement and also demonstrated that the transitions usually cause GPS data interruption (i.e. signal shortage under the roof in a train station, or entering a bus). Accordingly, it proposes to use the stops and signal shortages as an additional indication for a potential-transition location (BILJECKI et al., 2013).

Related to the movement states criterion, most of the proposals segment the trajectories splitting its path into periods of time when the object is considered as stationary and periods where the object is indeed moving. These periods are denoted as *stops* and *moves*(SPACCAPIETRA et al., 2008) and represent *move episodes*(PARENT et al., 2013). Generally, during stops and move, movement parameters such as speed and direction present a totally different profile. Accordingly, it is very natural to apply these parameters to identify *move episodes.* Focus on the variation of the speed of the trajectory, (KRUMM; HORVITZ, 2006; ANDRIENKO et al., 2007) associate a stop to the GPS position which velocity is zero or near to zero during a given interval of time. The interval of time is defined depending on the applications. Thus, some approaches use a larger temporal value than others. A relatively similar, but a more refined assumption is adopted by (ZHENG et al., 2011), that identify stops when a sequence of consecutive GPS positions, such that their spatial distance is below a threshold, exceeds a temporal duration defined by another threshold. Others authors detect stops by finding the regions where the velocity is lower than the average speed of the trajectory (PALMA et al., 2008; ZIMMERMANN et al., 2009; TRAN et al., 2011). Follow this same paradigm, but also treating the noisy in trajectories, (XIANG et al., 2016) presents a sequence-oriented clustering approach for extracting stops. Other methods give attention to the repeat travel behaviors of moving objects for detecting stops. (HUANG et al., 2016) specifically, finds common sequences of stop regions where a certain number of objects visit with similar stop duration. Works based on the variation of the direction change parameter for discerning between *stops* and *moves* are presented in (LAUBE et al., 2005; ROCHA et al., 2010; GONG et al., 2015). To further understand the *move episodes* within a trajectory, semantic studies characterize (MORENO et al., 2010) and enrich (PARENT et al., 2013; YAN et al., 2013) them.

According to the available literature, the most commonly used methods to segment trajectories into similar sub-trajectories are based on clustering techniques. These techniques consider spatial (i.e., locations), temporal (i.e., timestamps), semantic and other special features for similarity measurements. For example, in (LEE et al., 2007) the authors developed a framework called

TRACLUS to discover common sub-trajectories from a trajectory dataset. It consists of two phases: partitioning and grouping. In the first phase, each trajectory is partitioned into a set of line segments at characteristic geospatial points, and then, in the second one, similar line segments in a dense region are grouped into a cluster. (Huang et al. 2011) proposes a mining algorithm to identify Longest Common Route (LCR) patterns based on turning regions (LCRTurning), which discovers a sequence of turning regions to abstract a trajectory. A relatively similar approach based on Longest Common Subsequences (LCSS) and Dynamic Time Wrapping (DTW) is presented in (VLACHOS et al., 2006). Its major contribution is that support multiple distance measures. (LEE et al., 2012) proposes a prediction framework to estimate the travel time of buses by exploiting collected bus trajectory data as follow. In order to identify a set of similar sub-trajectories effective for travel time prediction, they segment the trajectories based on travel time passed segments, days and hours by using similarity measures for time series, e.g., LP-norm, DTW, and LCSS. Other works use the duration of staying on each location of trajectories (HUANG et al., 2016), semantic aspects (LI et al., 2008; ZHENG et al., 2011) and some special behavior phenomena, like the silent durations (HUNG et al., 2015) to find similar sub-trajectories.

The segmentation of trajectories using the geo-spatiotemporal criterion fundamentally focuses on the selection of consecutive trajectory points that share a set of similar spatiotemporal properties. In this regard, several solutions also based on clustering algorithms have been developed in the last years. For example, (ETIENNE et al., 2012) performs a data mining on a huge trajectory dataset of mobile object's moving in an open space (e.g. maritime area) in order to find spatiotemporal patterns. The approach consists of six steps and one of them refers to the *spatiotemporal trajectories extraction and filtering*. In this step, the trajectories that follow the same itinerary are extracted from the spatiotemporal database and clustered using three criteria. The first one is the type of the mobile object, the second is a geographical one and the last one is time. Other spatiotemporal segmentation techniques are explained in (Kang & Yong 2010; Rao et al. 2012;).Some other approaches only consider one feature (e.g. spatial or

temporal) to segment a set of trajectories, however, a combination of them can guarantee more precise results (HUANG et al., 2016).

Another parameter, which in addition to spatiotemporal is considered in clustering algorithms is the direction of trajectories. One of the most well knows spatiotemporal-directional clustering methods is DB-SMoT (ROCHA et al., 2010). The goal of this approach is to find interesting places in trajectories, considering the variation of the direction as the main aspect. For such purpose, they segment single trajectories by finding clusters based on the direction change and spatiotemporal features.

(BAK et al., 2012) presents a method for the detection of spatiotemporal encounters. The method segments a trajectory dataset as follow. Firstly, it selects trajectory points with time difference shorter than a defined threshold ($\Delta T = 30$ seconds). Afterward, it finds pairs of trajectory points located with distance less than another threshold ($\Delta S = 10$ meters). Then, depending on the angle (e.g. 0°, 90°, 180°) between the trajectories that contain the resulting points, they classify the type of the encounter as parallel and in the same direction, perpendicular or parallel and in opposite direction. Similarly, to our approach, the authors define a temporal segmentation, but differently, they don't perform an a priori spatial segmentation, since they just spatially segment trajectories when a potential encounter is detected.

Works such as (FAN et al., 2007; DAEINABI et al., 2011; SUTAGUNDAR et al., 2016) propose clustering algorithms over a set of trajectories in Vehicular Ad Hoc Networks (VANETs). During the creation of clusters, one vehicle node per cluster is selected as the cluster head to act as the routing node. Clustering is over a set of trajectories. One disadvantage is their high re-affiliation frequency when the network changes very fast. Specifically, the high dynamic mobility of vehicles and high change of network topology reduce the stability of cluster formation.

Moreover, a few approaches include other variables in clustering algorithms. One example of a multidimensional clustering algorithm is presented in (BUCHIN et al., 2011), whose authors propose a framework for segmenting a trajectory based on spatiotemporal criteria that also takes into account heading,

speed, curvature, sinuosity, curviness, and shape of the trajectory. This framework allows not only segment the trajectories by any of these criteria, but also by any combination of them.

From all aforementioned trajectory segmentation criteria, our work is more related to the spatiotemporal one. However, differently from the above existing work, we perform the segmentation of raw trajectories as follow:

1. Spatial segmentation: using an *a priori* defined a polygonal region (geo-fence), which is associated to control points located over the bus network.
2. Temporal segmentation: considering time period (e.g. weekdays and weekends) and predefined time intervals (e.g. 8:00 AM - 9:00 AM)
3. Directional Segmentation: separating trajectories that, despite corresponding to the same space area and at the same time interval, have opposite directions

This segmentation scheme is discussed in more details in Section 4.4.

2.2. Estimation of traffic patterns from GPS data

Multiple traffic patterns can be estimated using historical trajectory data generated by GPS, such as density, occupancy, volume, flow, speed and travel time. The extraction of these traffic patterns pursues two main objectives: to explain common traffic conditions and to predict future traffic conditions. Work focused on the first objective addresses traffic monitoring (COSTANZO, 2013), detection of traffic anomalies (Kuang et al. 2015), and traffic performance analysis (SHI et al., 2008; GRASER et al., 2012).

Work focused on the second objective includes traffic state prediction (Zhang et al. 2013), urban traffic congestion forecasting (HOU et al., 2012; KONG et al., 2016), prediction of traffic anomaly duration (LI, 2015), and estimating time of arrival (Coquita et al. 2015; Kormáksson et al. 2014; Jithendra. H. K 2015; Si 2012).

Both for the purposes of understanding the current traffic conditions and for the purposes of predicting future traffic behavior, some approaches discover

certain regular patterns from the historical data collected over time, by fitting the historical data to statistical models such as Gaussian model (SUN; XU, 2011), Regression models (DUNNE; GHOSH, 2011), Bayesian network, and Markov Chains (MANLEY, 2015). Moreover, methods based on time-series analysis focus on discovering the internal relationship among historical time-series data (DU et al., 2012; REMPE et al., 2016). Other works are based on artificial intelligence techniques such as Neural Networks (SUN et al., 2012; HABTIE et al., 2016), Fuzzy Logic(STATHOPOULOS et al., 2010; YANG et al., 2015), Support Vector Machine(LIU, X. et al., 2011) and Evolutionary algorithms (HONG et al., 2011). Hybrid approaches that combine statistical and computational intelligence models have also been developed (HABTEMICHAEL; CETIN, 2015; WANG et al., 2016).

In this thesis, we focus on estimating the travel time pattern of buses on each segment of the bus network to model past and current traffic behavior. To estimate this pattern, we compute the average bus travel time derived from historical GPS data. Despite the existence of more complex pattern estimation models, there are studies that have demonstrated that, for monitoring and prediction of traffic conditions, computing patterns using just the average provides satisfactory results (LEE et al., 2012; HA; OH, 2014; WANG et al., 2014; NARAYANAN et al., 2015).

2.3. Detection of traffic anomalies

Some relevant works address the detection of traffic anomalies with GPS data (LIU, W. et al., 2011; CHAWLA et al., 2012; PANG et al., 2013; WANG et al., 2013), while others use social media data as a source of mobility data to detect anomalies (DALY et al., 2013; PAN et al., 2013; SAKAKI et al., 2013; CHEN et al., 2014; D'ANDREA et al., 2015). According to (KUANG et al., 2015), traffic anomaly detection methods can be classified into four main categories: distance-based (SETHI, 2013), cluster-based (ANBAROGLU et al., 2014), classification-based (LAN et al., 2014), and statistics-based categories (Kinoshita et al. 2015). However, according to (CHEN et al., 2010), the most popular and effective to

detect anomalies in traffic domain is the statistics-based approach. In this thesis, we focus on bus GPS data using a detection method that can be classified as statistics-based.

Statistical anomaly detection methods are based on the following key assumption: "*Normal data instances occur in high probability regions of a stochastic model, while anomalies occur in the low probability regions of the stochastic model."* (CHANDOLA et al., 2009). In accordance with this assumption, instances that have a low probability to be generated from the learned stochastic model, based on the applied test statistic, are declared as anomalies.

Among the most commonly used statistical methods to detect anomalies are: Statistical Quality Control (SQC) (MONTGOMERY, 2009), Box Plot Rule (BENJAMINI, 1988), Grubb's test (also known as Maximum Normed Residual Test) (GRUBBS, 1969), Autoregressive Integrated Moving Average (ARIMA) (HIBON; MAKRIDAKIS, 1997), Student´s *t*-test (SURACE et al., 1998), *Hotelling* t^2*-test* (HOTELLING, 1931), Histogram Based (ESKIN, 2000) and Kernel Function Based (PARZEN, 1962). Some of them examine individual variables (univariate methods), while others examine multiples variables (multivariate methods).

One of the pioneering works to apply statistical methods for detecting anomalies in traffic was (TUROCHY; SMITH, 2000). After this work, many other statistics-based solutions have been proposed such as (CAMOSSI et al., 2013; RAIYN; TOLEDO, 2014; KUANG et al., 2015; NECULA, 2015). They provide useful information for traffic management since help traffic authorities to better understand the traffic conditions and therefore to make optimal decisions.

In this thesis, we combine the Statistical Quality Control and the discretization of confidence levels proposed by (TUROCHY; SMITH, 2000) for detecting traffic anomalies and additionally for classifying them according to their severity. This combined method can be classified as univariate since we use one variable (the travel time) during the statistical analysis.

2.4. Evaluation of the impact of traffic anomalies

Assessing the impact of traffic anomalies has been the goal of a large number of research studies. Current approaches are centered on two focal points: predicting the impact and measuring the impact after the anomaly occurred.

Focusing on the first point, models have been proposed to predict the duration of traffic incidents in expressways (LI, 2015; CHUNG et al., 2015; PARK et al., 2015), to predict the occurrence of secondary incidents and delays associated with them (PARK et al., 2015), to estimate the scope of incident influence (XIE; WANG, 2015) in terms of the number of blocked lanes (CHUNG et al., 2015), and the length of the queue of cars produced by incidents on urban expressways (LV et al., 2015). Other approaches forecast the impact of traffic events in terms of cost (MILLER; GUPTA, 2012) and environmental pollution (ZHANG et al., 2014).

With a focus on the second point, a method to quantify the effect of events in the urban traffic flow from raw data emitted by sensors is proposed in (HOU et al., 2012). Hojati proposes, in his thesis(TAVASSOLI HOJATI, 2014), a general methodology to analyze the impact of traffic incidents on the reliability of travel time. This methodology offers insights to understand the relationships between travel time reliability, incident details, traffic characteristics, road network infrastructure characteristics, and weather conditions.

Case studies in different cities investigate and measure the effects of weather conditions on traffic in the road network (KOETSE; RIETVELD, 2009; COOLS et al., 2010; ARANA et al., 2014; SINGHAL et al., 2014). A study on congestion levels produced by traffic accidents between 2004 and 2010 in Cadiz, Colombia, is presented in (ARBOLEDA et al., 2012).

The work of (BARBOSA et al., 2014) about the City of Rio de Janeiro is closely related to our study. The authors developed a system, called Vistradas, for the visual analysis of bus trajectory data. One of the features of this system is that measures the impact of events on traffic, only in terms of bus speed. To do this, they calculate the difference between the average speed of traffic on bus routes in a period before and after a given event. The main shortcoming of this approach is

that it does not consider the time of day, nor the day of the week for the impact assessment. Differently, in this thesis, we evaluate the impact or traffic anomalies in term of the duration and the delay produced over the typical travel time pattern of buses, considering the period of time and the interval of the day.

3
Overview of the Proposed Approach

3.1.
Introduction

In this chapter, we overview the approach we propose for analyzing and monitoring the bus network of a city. The overview includes a definition of the basic concepts used throughout the thesis, a description of the proposed architecture and a list of the tools used for the implementation of the prototype that supports the proposal.

The approach depends on data generated by GPS devices installed in buses. In that sense, buses equipped with GPS devices are treated as mobile traffic sensors, which describe trajectories that cover the same set of streets, at predictable regular intervals. Therefore, our approach can be applied to cities served by a network of buses equipped with GPS devices, that continuously transmit their position.

3.2.
Basic Concepts

A *road network* is a labelled, directed graph $G = (V, E, nl, el)$, where *V* is the set of nodes, *E* the set of edges, *nl* associates a geo-referenced point (in an appropriate geographic coordinate system) with each node in *V* and *el* assigns a geo-referenced line segment (in the same geographic coordinate system) to each edge in *E*. Intuitively, the edges represent road segments and the nodes indicate the start and end points of the road segments.

A *bus network* is a labelled, directed graph $B = (V_b, E_b, nl_r, el_r, nl_b, el_b)$, where

- nl_rassociates a geo-referenced point (in an appropriate geographic coordinate system) with each node$n$ inV_b

- el_rassociates a geo-referenced line string (in the same geographic coordinate system) with each edge e inE_b
- nl_blabels each node$n$ in$V_b$with the bus routes that pass through n
- el_blabels each edge e in$E_b$with the bus routes that pass through e

Intuitively, the edges represent road segments that buses traverse and the nodes indicate the start and end points of such road segments.

A *one-way trip route* is a path of the bus network and a *round-trip route* is a loop of the bus network.

A *bus route R* is a connected subgraph of *B* composed of a set of round-trip routes and one-way trip routes such that *R* is the union of the round-trip routes and alternative simple trip routes. We acknowledge that this definition is a simplification of the bus routes observed in practice, but it suffices for the purposes of this thesis since it focuses on monitored paths and travel time patterns defined in what follows.

A *bus network version* is a triple$B_t = (B, t_i, t_f)$ where B is a bus network and $t_f$and$t_i$are timestamps that delimit the period $\Delta t = t_f - t_i$ during which the bus network maintained the same characteristics (such as structural features and the bus routes).

A *monitored bus network* is a subgraph of *B*. Intuitively, a monitored bus network consists of the nodes and edges of B that are frequently traversed by buses so that meaningful statistics can be computed.

A *monitored path* is a path $p_j$of *B*. The *control points pair* of $p_j$is the pair (c_1^j, c_2^j), where c_1^j is the start node and c_2^j is the end node of p_j. Note thatnl_r provides a geo-referencing for the control points and el_r provides a geo-referencing for the path.

A *raw bus trajectory s* is a sequence $s = \langle (p_1, t_1), (p_2, t_2), \ldots, (p_n, t_n) \rangle$ such that $p_i = (x_i, y_i)$ is a geo-referenced point and t_i is a timestamp such that $t_i < t_{i+1}$, for $i = 1, \ldots, n$. Araw bus trajectory *s* represents the position evolution of a moving bus.

A *travel time pattern* for a monitored path over a period of time isany statistical measure of the travel time of the buses that traverse the given path during the given period, represented by a function.

Given a bus network *B*, the *travel time pattern problem* for *B* refers to the problem of determining bus travel time patterns for a given set of monitored paths of *B* over a given period and a given time interval.

Given a bus network *B*, the *problem of traffic anomaly detection* refers to the problem of determining when the travel time of buses passing through a given set of monitored paths of *B*, at a given period and at a given time interval deviate from travel time pattern.

3.3. Architectural Overview

Figure 1shows the architecture proposed for a prototype tool that implements the bus network analysis and monitoring approach we propose in this thesis. As illustrated, the architecture is composed of three layers: *Data*, *Non-real Time Processing*, and *Real Time Processing*.

The *Data* layer includes: a set of General Transit Feed Specification[1](GTFS) files containing the routes that buses operating in the city should follow; an Open Street Map[2] (OSM) file in XML format with a standard specification of the geographic position of the city road network points; a database that stores the graphs corresponding to different versions of the bus network of the city; and other four datasets that store data about: (i) the historical bus trajectories, (ii) the travel time used by buses to traverse each segment (path) of the monitored bus network, (iii) the travel time patterns for each monitored segment, and (iv) the traffic anomalies detected.

The files with the bus routes, the OSM file and the dataset with the historical bus trajectories represent the inputs of the proposed approach. The

[1]https://developers.google.com/transit/gtfs/reference/

[2]http://www.openstreetmap.org

datasets that store the travel times, the travel time patterns and the traffic anomalies detected are the output of the proposed approach.

The *Non-real Time Processing* layer encompass three modules: *Bus Network*, *Travel Time Patterns*, and *Traffic Anomalies*.

The *Bus Network* module is responsible for making the map matching of bus routes. This is because the geometry of each bus route provided either by the governing bodies of the cities or by other sources may have errors or simply may not have been determined using the same cartographic system used by OpenStreetMap, which is the standard selected for this thesis. Thereby, to ensure its correctness, it is necessary to perform a route matching process to associate each route position to the correct road point. For such purpose, in addition to the file with the bus routes, the OSM standard specification file is used.

With the bus routes correctly mapped, a bus network version of the city is modeled and built. The bus network of a city encompasses multiples bus network versions. Each bus network version is composed of nodes and edges, as defined in Section 3.2. Afterward, according to an analysis of the bus routes that pass through the edges and nodes, the monitored bus network is defined. Finally, the monitored bus network is segmented into paths whose traffic will be monitored with the help of bus trajectories. The information about the graph structure of each bus network version, its corresponding monitored bus network and its segmentation is stored in the Bus Network database.

The *Travel Time Patterns* module performs a spatial-temporal-directional segmentation of the historical trajectories of the buses based on the spatial segmentation of the bus network and on a statistical analysis of the historical trajectories of the buses. According to this segmentation, the travel time that each bus spent in crossing each monitored path is computed and saved in the historical travel time dataset. Using these data, the travel time patterns to traverse each monitored path during each time period of the year and at each interval of the day are estimated. Then, the travel time patterns are stored into the respective dataset.

At this point, we note that, whenever there is a change in the bus routes or in the road network structure of the city that affects the bus network, the functionalities of the *Bus Network* and *Travel Time Patterns* modules must be re-

executed. Hence, we are able to maintain different versions of the bus network and their corresponding traffic patterns. This capability allows us to compare the versions and thereby, help traffic planners assess the impact of road network changes on the traffic behavior.

The *Traffic Anomalies* module has two components, one belongs to the *Non-real Time Processing* layer and the other to *Real Time Processing* layer. The *Traffic Anomalies* component corresponding to the *Non-real Time Processing* layer is in charge of detecting traffic anomalies that happened in the past, which are hidden in the data of the historical bus trajectories. Thus, the detection of traffic anomalies focuses on identifying when the travel time patterns of the segments deviated from typical travel time patterns. After the anomalies are detected, they are classified according to their severity, and their impact in terms of duration and delay that caused in the travel time of buses is estimated. The data resulting from the detection, classification, and estimation of the impact of traffic anomalies are saved into the traffic anomaly dataset.

The *Real-Time Processing* layer is responsible for constantly monitoring the instant position of buses to alert when, in a certain monitored path, a traffic anomaly occurs, which will be expressed through a delay in the travel time of the buses running in that region. The current layer contains two modules: *Monitoring* and *Traffic Anomalies.*

As the instant position of buses is emitted by the GPS devices, these data in the form of raw trajectories are replicated to the processing nodes. Each processing node filters these raw trajectories, keeping only those GPS observations that belong to its monitoring area. Then, these GPS observations are segmented using the delimitation of monitored paths stored in the graph database. After that, the observations are cleaned to avoid storing those containing incomplete information.

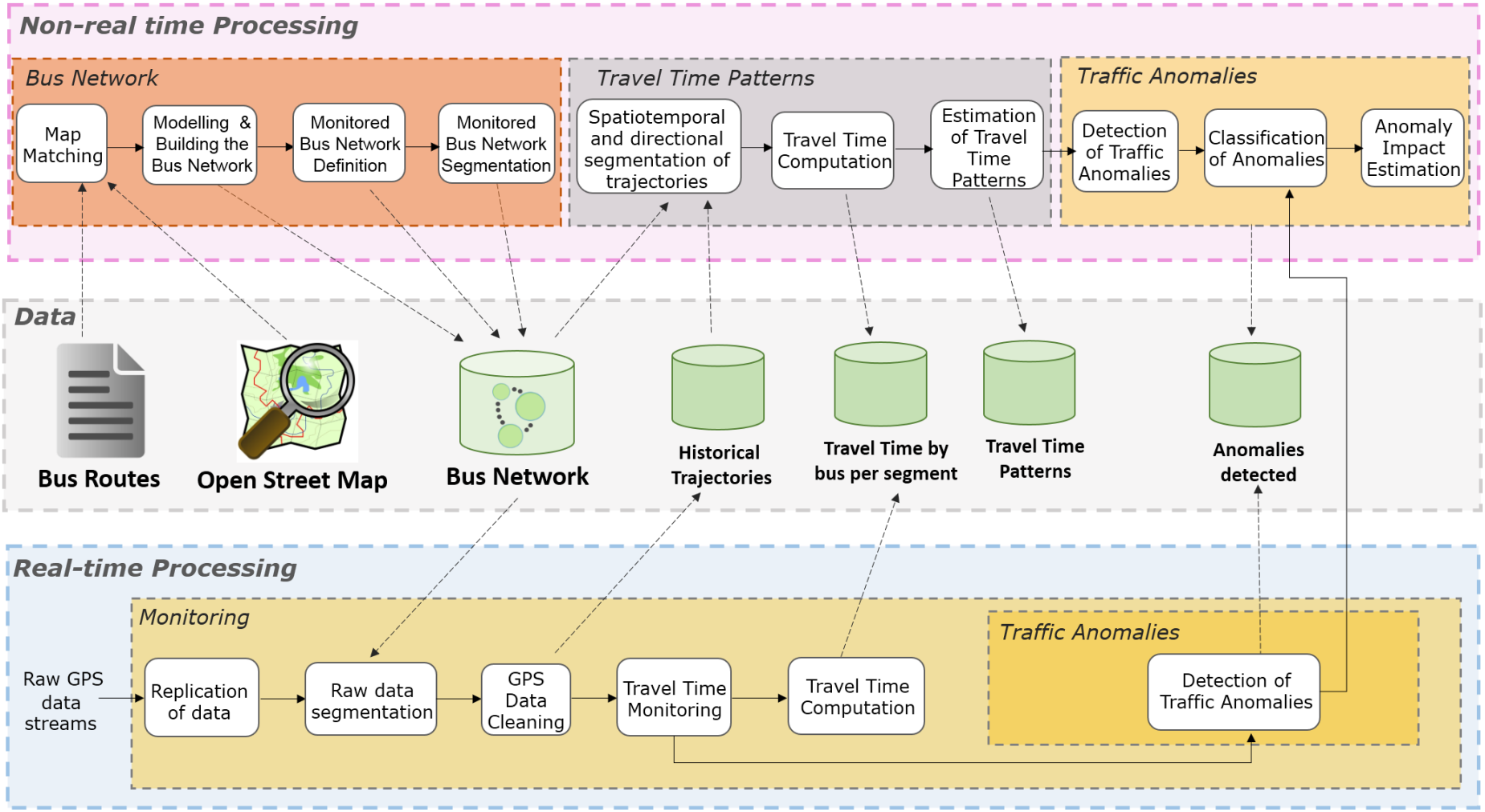


Figure 1: Overview of the architecture for Bus Network Analysis and Monitoring approach.

The processing nodes are continuously monitoring the current position of the buses and the travel time used to reach this position. If a bus reaches the end of the segment without manifesting any delay in its travel time with respect to the pattern, then the travel time that the bus wasted for traversing the road segment is stored in the travel time dataset and the *Traffic Anomalies* module is not executed. Otherwise, if a significative delay in the travel time of the buses occurs, it is perceived, by the *Traffic Anomalies* module. After its detection, the traffic anomalies are classified according to their severity; and their impact in terms of duration and delay that caused in the travel time of buses is estimated,in the same way as the *Traffic Anomalies* module of the Non-real Time Processing layer does.

The modules of the proposed architecture are described in the next three chapters in more details with technical information. Specifically, the *Bus Network* module is addressed in Chapter 4, the *Travel Time Patterns* module in Chapter 5, and *Monitoring* and *Traffic Anomalies* modules are described together in Chapter 6.

3.4. Implementation details of the Prototype Tool

The implementation of the prototype tool designed to support the proposed approach stores data mainly in MongoDB[3] and Neo4j[4].

MongoDB is used to store the historical bus trajectory data generated by GPS devices, bus travel time, travel time patterns and the detected anomalies. MongoDB is a NoSQL Database Management System (DBMS) with a notable share in GIS since it supports geodata types and provides high-performance geodata operations. This is achieved by supporting three types of data of Geospatial indexes: 2D that uses simple coordinate (longitude, latitude), 2D Sphere that allows queries of any geometries on an earth-like sphere, and Geo Haystack, used to query on very small areas. In this thesis, we use the 2D index, both to store data as points on a two-dimensional plane and to retrieve Point,

[3] https://www.mongodb.com/

[4] https://neo4j.com/

LineString, Polygon, and MultiLineString geometry types.

Another reason for choosing MongoDB is its scalability. In that sense, MongoDB automatic sharding distributes data across fleets of commodity servers, with complete application transparency. With multiple options for scaling – including range-based, hash-based and location-aware sharding – MongoDB can support thousands of nodes, petabytes of data, and hundreds of thousands of operations per second without requiring you to build custom partitioning and caching layers. This feature is very useful for the on-the-flight data analyzes.

Neo4jis used to save the structure of a bus network. Neo4j is a graph database management system with a native Graph Processing Engine that allows storing data in the form of an edge, a node, or an attribute, where each node and edge can have any number of attributes. Additionally, it offers a simple and powerful data model, which avoids having to execute complex *Joins* to retrieve connected/related data, i.e., Neo4j makes it very easy to retrieve their adjacent node or relationship details without *Joins* or *Indexes*.

The prototype tool was implemented in Python, which was selected because of the availability of packages for:

- GIS operations and statistics (Numpy[5], Pyproj[6], and Scipy[7]
- Interacting with data in MongoDB (PyMongo[8])
- Interacting with data in Neo4j, that supports queries directly in Cypher Query Language (Py2neo[9])
- Creating and managing the structure, dynamics, and functions of complex networks, with support for graphs, digraphs, multigraphs and Multidigraphs (graphs that permit multiple edges between nodes in any direction) algorithms (NetworkX[10])

[5] http://www.numpy.org/

[6] https://pypi.python.org/pypi/pyproj

[7] https://www.scipy.org/

[8] https://api.mongodb.com/python/current/

[9] http://py2neo.org/v3/

[10] https://networkx.github.io/

- Importing, storage, and querying of spatial data in the Neo4j with efficient geospatial indexing (Neo4j Spatial[11])
- Creating geospatial geometries that delimit objects (shapely.geometry[12])

The Mapbox[13] Python SDK was used for map matching of the bus routes.

Finally, we observe that all tools used are open-source. Hence, the prototype tool was completely built on free software. It is available at https://github.com/kathrinr.llanes/PhD.

[11] https://neo4j.com/blog/neo4j-spatial-part1-finding-things-close-to-other-things/

[12] http://toblerity.org/shapely/shapely.geometry.html

[13] https://www.mapbox.com/help/define-map-matching/

4

Bus Network

4.1. Introduction

The modeling and analysis of bus transport networks (BTNs) are of practical importance for urban and traffic management. They are tools to support the decision-making that, in addition to describing the topological features of bus networks and the bus routes that serve them, provide a detailed view of the operation of bus public transportation systems and facilitate other derived analysis. As examples, we have: the evaluation of bus network structure (ZHANG, 2017) and bus transportation system performance(LI et al., 2013; VAIDYA, 2014; BONA, DE et al., 2016); the optimization of the bus routes and their frequency(MAUTTONE et al., 2010; MARTINEZ et al., 2014); the extraction of BTN statistical properties(CHATTERJEE et al., 2016)and other indicators, such as the number of passengers per route and bus trip(CHU, 2009), the paths most traveled by buses (LLANES et al., 2017); the estimation of origin-destination matrices(BERA; RAO, 2011; JI et al., 2015), useful to explore urban mobility trends and bus travel demand (YU; HE, 2017); the identification of bus network expansion needs(SAHA; SHINSTINE, 2015); and the detection of urban organizational inconsistencies(HÁZNAGY et al., 2015), which have negative impacts on bus network performance and in transportation dynamics in general.

In this context, the modeling and analysis of bus networks complement the professional knowledge and experience of transport planners with real quantitative elements and thus enable an efficient organization and management of public urban transportation systems.

In order to achieve these objectives and also for the purposes of monitoring the operation of buses, our proposal includes a method, based on graph theory, for modeling a bus network, which is described in this chapter. According to this

modeling, we propose an algorithm for building the versions of bus network (Algorithm 1). Additionally, as part of the analysis that can be done over a bus network, we propose algorithms for selecting road segments whose traffic will be monitored with the help of bus trajectories. Specifically, Algorithm 2 computes the monitored bus network, Algorithm 3 selects candidates for monitored paths, and Algorithm 4 refines the candidate monitored paths points.

4.2. Bus Network Modeling and Building

Bus networks may be static, when their topological features and bus routes do not vary with time, or dynamic. Therefore, for modeling and analyzing the latter it is necessary to take into account the variation of their structure and the routes that serve them over time.

Considering that the real-world bus networks are dynamic, they should be modeled according to their time-based variations. For this reason, we propose a spatiotemporal modeling for bus network, such that a bus network is modeled as a collection of its different versions. We recall that each version corresponds to a delimited time period during which the bus network conserves the same characteristics (structural features and the bus routes).

Among existing mathematical and computational structures, graphs, in particular, has been immensely successful in modeling real world networks data in the recent times (CHATTERJEE, 2015). Accordingly, the proposed modeling of a bus network version is based on graph theory. It summarizes, as *nodes* and *edges* of a graph, three main types of data: the topological structure of the bus network, the bus routes that serve it, and statistics about the historical bus trips made on it. The proposed model provides a description of a bus network and the properties associated with it.

The relationships among the different elements used for the proposed modeling are illustrated inFigure2. A bus network, bus network version, a node, and an edge are modeled as follows in Tables 1, 2, 3, and 4respectively.

Modeling a bus network data in a spatiotemporal way and using graphs allows organizing the change of information of the bus network in an efficient way to facilitate spatiotemporal queries and spatiotemporal analysis. Thereby, it is possible to maintain and compare different versions of a bus network and their traffic patterns, which helps understand the evolution, growth, robustness and resiliency of a bus network of a city and helps assist city planners assess changes and, thereby, implement more efficient bus networks.

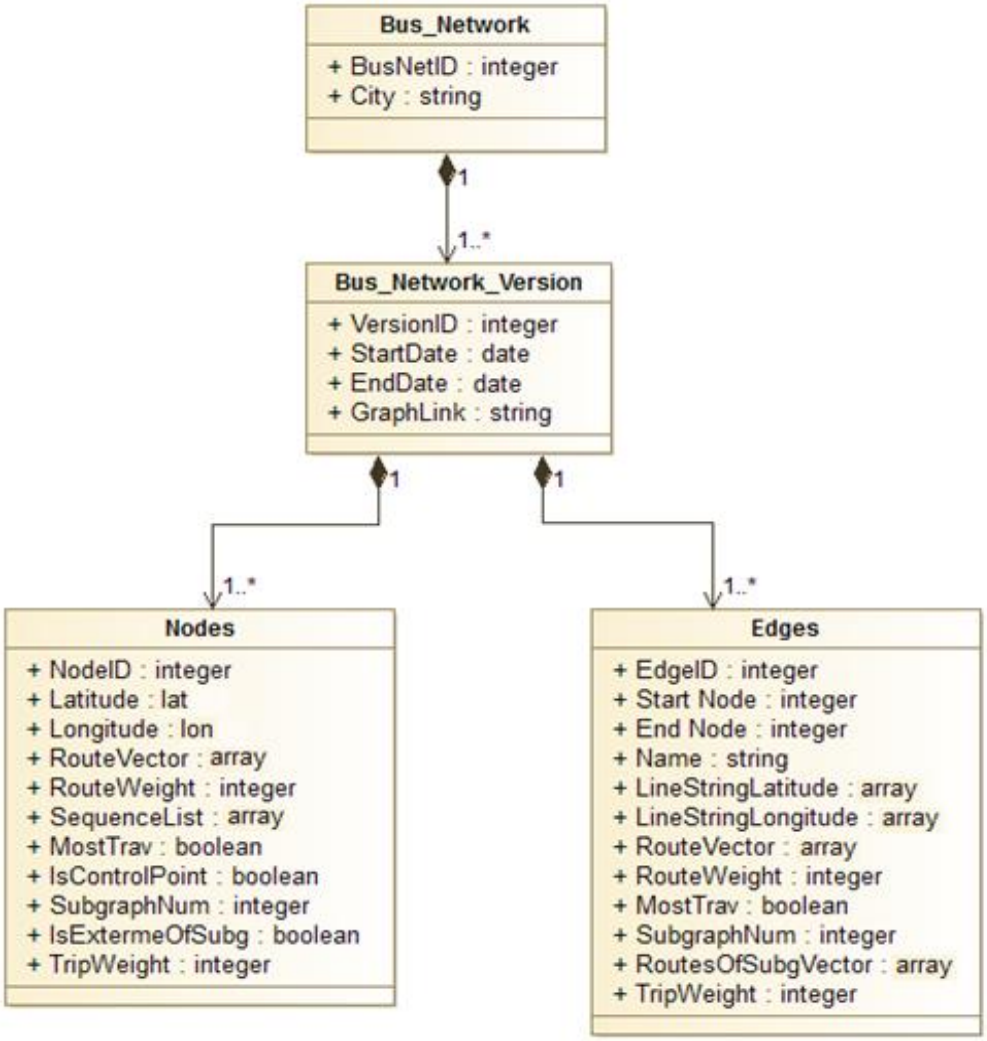


Figure 2: UML class diagram for the proposed model.

Table 1: Bus Network Entity.

Property Name	Type	Description
BusNetID	int	Unique identifier of the bus network

City	string	Name of the city to which the bus network belongs

Table 2: Bus Network Version Entity.

Property Name	Type	Description
VersionID	int	Unique identifier of the bus network version
StartDate	date	Start date of the bus network version
EndDate	date	End date of the bus network version
GraphLink	string	Link where the graph of the bus network version is stored

Table 3: Node Entity.

Property Name	Type	Description
NodeID	int	Unique identifier of the node
Latitude	lat	Geographic coordinate (latitude) of the node
Longitude	lon	Geographic coordinate (longitude) of the node.
RouteVector	string[]	List of bus routes that pass through the node
RouteWeight	int	Weight associated with the node according to the number of bus routes passing through it.
SequenceList	int[]	Listof the number of the node in each ordered sequence of nodes of the bus
MostTrav	boolean	Indicates if the node belongs to the monitored bus network.

IsControlPoint	boolean	Indicates if the node represents a monitored bus network segmentation point or not.
SubgraphNum	int	Number of the connected component of the monitored bus network to which the node belongs. For nodes do not belong to the monitored bus network (i.e. MostTrav = False), this property is set to 0. Otherwise, for nodes whose MostTrav= True, the value of this property is greater than 0 and less than or equal to the number of connected components.
IsExtermeOfSubg	boolean	Indicates if the node is an extreme node of the connected component of the monitored bus network to which it belongs.
TripWeight	int	Weight associated with the average per day of bus tripsthat pass through the node. The weight is computed based on historical bus trajectory data.

Table 4: Edge Entity.

Property Name	Type	Description
EdgeID	int	Unique identifier of the edge
Start Node	int	Start node of the edge
End Node	int	End node of the edge
Name	string	Name of the edge (CONNECTED_TO). This property is mandatory in Neo4j.

LineStringLatitude	lat[]	Listwith the geographic coordinates (latitude) corresponding to the line string that form the edge.
LineStringLongitude	lon[]	Arraywith the geographic coordinate (longitude) corresponding to the line string that form the edge.
RouteVector	string[]	Listof thebus routes that pass through the edge.
RouteWeight	int	Weight associated with the edge according to the number of bus routes passing through it.
MostTrav	boolean	Indicates if the edge belongs to the monitored bus network.
SubgraphNum	int	Number of the connected component of the monitored bus network to which the edge belongs. For edges do not belong to the monitored bus network (i.e. MostTrav = False), this property is set to 0. Otherwise, for edges whose MostTrav= True, the value of this property is greater than 0 and less than or equal to the number of connected components.
RoutesOfSubgVector	int[]	Array of 0s and 1sthat indicate which of the buses that serve the connected component to which the edge belongs, actually pass through the edge.

TripWeight	int	Weight associated with the average per day of bus tripsthat pass through the edge. The weight is computed based on historical bus trajectory data.

Algorithm 1 builds a bus network version as follows.

Route Matching. The algorithm receives as input an Open Street Map (OSM) file in XML format with a standard specification of the geographic position of the city road network points and a set of GTFS files in CSV format containing the shape of the bus routes that serve the city. According to Google specifications (GOOGLE, 2017), a General Transit Feed Specification (GTFS) bus shape file should have the following data: the *bus_route_ID* that contains an ID that uniquely identifies a route, *route_name* that contains the short name of a route; the *shape_ID* that contains an ID that uniquely identifies a shape, the *latitude* and *longitude* that respectively associates a shape point latitude and longitude with a shape ID; a *sequence*, which associates the latitude and longitude of a shape point with its sequence order along the shape. Based on this information, Line 2 verifies, in the GTFS file of each route shape, if there are consecutive nodes with the same geographical coordinates, and eliminates the duplicated nodes. Line 3 executes a route matching process to associate each route shape position to the correct road point according to the OSM data.

Line 4 describes a loop consisting of the statements in Lines 5-37. The loop continuous as long as all bus routes, correctly mapped, have not been analyzed.

Create Nodes. In Line 5, a while loop is used to continuously repeat the statements from Line 6 to 20until all nodes of the bus route shape have been inserted in the bus network graph. For each point (node) of a bus route shape, it is verified if it already exists in the graph of the bus network (Line 6).

When a node does not exist, the node is created and its properties are set (Lines 7-17). The properties *latitude*, *longitude*, *routeVector*, *routeWeight* and *SeqList*are filled based on the information extracted from the bus route shape file. Specifically, the *routeWeight*is initialized to 1, and the values corresponding to *bus_route_ID*and *sequence*are respectively added to the arrays *routeVector* and *SeqList*, that are empty by default.

Algorithm 1 Pseudocode for building the Bus Network

```
1: function BusNetwork(busRoutes, OSMfile)
2:     cleanRoutes ← delDuplicConsecNodes(busRoutes)
3:     matchedRoutes ← MapMatching(busRoutes,OSMfile)
4:     for each route in matchedRoutes do
5:         while currNode.GetNumber() < route.GetNumberOfNodes() do
6:             if graph.matchNode(node = currNode) = 0 then
7:                 graph.createNode(node = currentnode,
8:                                  lat = getLat(currNode),
9:                                  lon = getLon(currNode),
10:                                 routeVector.add(routeID),
11:                                 routeWeight = 1,
12:                                 seqList.add(getSequence(currNode)),
13:                                 mostTrav = False,
14:                                 IsControlPoint = False,
15:                                 subgraphNum = 0,
16:                                 IsExtermeofSubg = False,
17:                                 tripWeight = 0)
18:             else
19:                 UpdateNodePropert(currNode)
20:             end if
21:         end while
22:         while currEdge.GetNumber() < route.GetNumberOfEdges() do
23:             if graph.matchRel(startNode = n2, endNode = n2) = 0 then
24:                 graph.createRelation(startNode = n2, endNode = n2
25:                                  name = "CONNECTED_TO",
26:                                  lineStrLat = getLineStrLat(n1, n2),
27:                                  lineStrLon = getLineStrLon(n1, n2),
28:                                  routeVector.add(routeID),
29:                                  routeWeight = 1,
30:                                  mostTrav = False,
31:                                  subgraphNum = 0,
32:                                  routesOfSubgVector = [ ],
33:                                  tripWeight = 0)
34:             else
35:                 UpdateRelPropert(startNode = n2, endNode = n2)
36:             end if
37:         end while
38:     end for
39: end function
```

Algorithm 1: Building the Bus Network.

The remainder of the properties – *MostTrav*, *IsControlPoint*, *SubgraphNum*, *IsExtermeOfSubg*, and *TripWeight*– are initialized to False, False, 0, False and 0, respectively. These properties are subsequently modified as a result of further processing (Algorithms 2, 4 and 5).

When a node already exists, only the properties *routeVector*, *routeWeight*, and *SeqList* are modified (Line 19). Similarly to the previous case, the values corresponding to *bus_route_ID* and *sequence* are respectively added to the *routeVector* and *SeqList*arrays, and the value of the *routeWeight* property is incremented in 1.

Create Edges. In Line 22, a while loop is used to continuously repeat the statements inLines23 to 36,until all edges, which are defined in the bus route shape file by any two consecutive nodes, have been inserted in the bus network graph. For each pair of consecutive points of a bus route shape file, Line 23 verifies if an edge between the nodes that represent these consecutive points already exists in the graph of the bus network.

When an edge between two consecutive points (nodes) of the bus route shape file does not exist in the bus network graph, it is created and its properties are set (Lines 24-33). The values of the *startNode* and *endNode* properties of the new edge correspond to the identifiers that the nodes that form it have assigned in the bus network graph. The value of the *name* propertywill always be "CONNECTED_TO". The *routeVector* and *routeWeight* properties are filled based on the information extracted from the bus route shape file. Specifically, the *routeWeight*is initialized to 1, and the value corresponding to *bus_route_ID*, defined in the bus route shape file, is added to the array *routeVector*, that are empty by default. The values of *LineStringLatitude* and *LineStringLongitude* properties are filled using data from OSM. The remainder of the properties – *MostTrav*, *SubgraphNum*, *RoutesOfSubgVector*, and *TripWeight* – are initialized respectively to False, 0, empty and 0. These properties are also subsequently modified as a result of further processing (Algorithms 2, 4 and 5).

When an edge already exists, only the values of the *routeVector*and *routeWeight* properties are modified (Line 35). Similarly to the previous case, the value corresponding to *bus_route_ID* is added to the *routeVector* and the value of the *routeWeight* property is incremented by 1.

4.3. Computation of the Monitored Bus Network

It is important to note that, in order to achieve precision and efficiency in the behavioral analysis and monitoring of bus transit tasks, it is necessary to have sufficient data to allow robust statistical analysis(TRIOLA, 2004).In this sense, the streets of the bus network that are less frequently traveled by buses will not provide enough data so that meaningful statistics can be computed. For this reason, in this section, we propose an algorithm to determine the subset of the bus network that can be adequately monitored with the help of bus trajectories, which is estimated as the set of the road segments most traversed by buses.

Algorithm 2 computes the monitored bus network as follows.

Algorithm 2 Pseudocode for create the Monitored Bus Network

```
1: function MonitoredNetwork(busNetwork)
2:     sortEdgesList ← GetMostTravEdges(BusNetwork)
3:     while len(sortEdgesList) ≠ 0 do
4:         mostTravEdge ←sortEdgesList(0)
5:         initialNode ←mostTravEdge(0)
6:         finalNode ←mostTravEdge(1)
7:         RemoveEdge(mostTravEdge, busNetwork)
8:         DeleteEdge(mostTravEdge, sortEdgesList)
9:         subgraph1 ← ReverseBfs(busNetwork,initialNode)
10:        subgraph2 ← Bfs(busNetwork,finalNode)
11:        subgraph ← subgraph1 + mostTravEdge + subgraph2
12:        for each edge in subgraph1 do
13:            RemoveEdge(edge, busNetwork)
14:            DeleteEdge(edge, sortEdgesList)
15:        end for
16:        for each edge in subgraph2 do
17:            RemoveEdge(edge, busNetwork)
18:            DeleteEdge(edge, sortEdgesList)
19:        end for
20:        subgraphSet ← subgraphSet + subgraph
21:    end while
22:    monitoredNetwork ← GetConnectedComp(subgraphSet)
23:    return monitoredNetwork
24: end function
```

Algorithm 2: Computation of the Monitored Bus Network.

Select the most traversed road segments. The algorithm receives as input a version of the bus network, similar at that one in Figure 16.Line 2 ranks the edges by the number of bus routes that traverse them and returns the most traversed edges.

Find connected components. For each edge in the set of the most traversed edges, Lines 5 and 6 compute the initial and final nodes of the edge, and Line 9 performs a reverse breadth-first search (BFS) over the version of the bus network starting from the initial node of the edge. Line 10 executes a direct BFS starting from the final node of the edge.

Both modifications of BFS algorithm (reverse BFS and direct BFS) explore the neighbor edges first, before moving to the next level neighbors and they are including in the result set the edges that are served by the same set of bus routes that serve the most traversed edge under analysis. When an edge served by a different set of bus routes is encountered, the algorithms stop. Thus, the algorithms form sub-paths composed by connected edges that are served by the same bus routes. As a result of both searches, two sub-paths are obtained.

Line 11 combines both sub-paths and the edge under analysis to compose a subgraph. As new edges are found by the direct and reverse BFS, they are removed from bus network and from the list of most traversed edges to avoid infinite loops. Lines 12 to 19 then gradually reduce the bus network and the list of the most traversed edges until they are empty. Line 20 adds each subgraph, generated by each of the most traversed edges, to a set of subgraphs. The same process (Line 4 - 20) is repeated until all edges in the most traversed set are analyzed.

Line 22 calls a function to find, within the set of subgraphs, those that have a common node and joins them in a single connected component. Thus, a set of disjoint subgraphs is obtained, which is the monitored bus network. Finally, Line 23 returns the monitored bus network, represented by its connected components, as illustrated in Figure 17.

.

4.4. Segmentation of the Monitored Bus Network

To segment the monitored bus network, we use the concept of control points. Then, monitored paths composed of a sequence of connected road segments are obtained, which are the minimal unit for monitoring the behavior of buses.

Algorithm 3 determines control points in the monitored bus network as follows.

Cluster edges by bus routes. The algorithm receives as input the set of connected components that form the monitored bus network. Line 4 applies a clustering function to each connected component that groups edges traversed by the same bus routes.

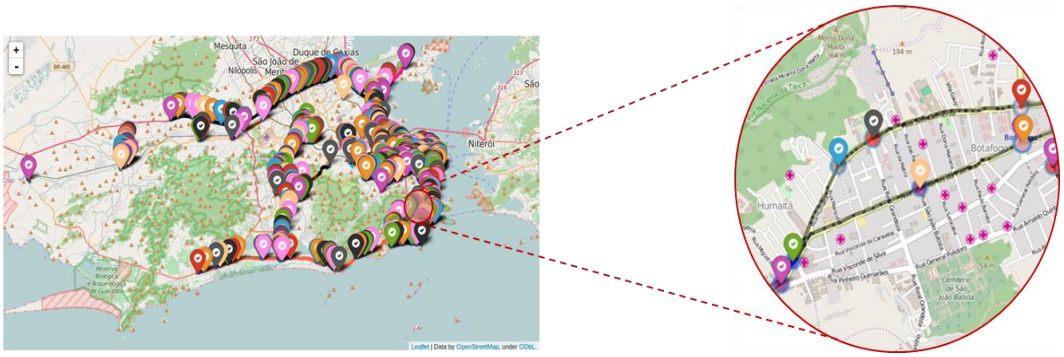
Find disjoint paths between the same cluster. Segments that correspond to the same cluster may be consecutive or not. If they are consecutive, they form longer paths served by the same bus routes. Line 6 combines all such paths into the same group.

Determine the initial and final nodes of disjoint paths. Lines 8-9 compute the initial and final nodes. For each path in the set *disjPaths.* Line 10 adds these pairs of initial and final nodes to the list of candidate control points and the monitored path between them. Finally, Line 14 returns a list of candidate control points. See

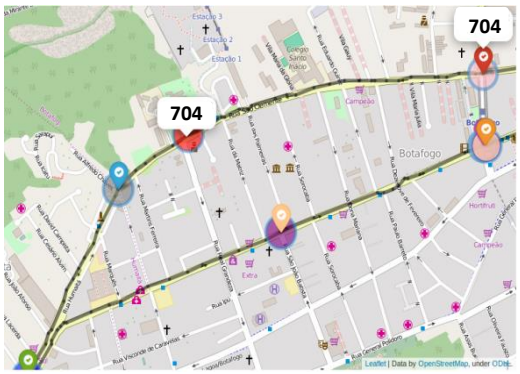
Algorithm 3 Pseudocode to define the control points of Monitored Bus Network

```
1:  function ControlPoints(MonitoredNetwork)
2:      contPointsCandidates ← [ ]
3:      for each component in monitoredNetwork do
4:          clusters ← ClusteringByBuses(component)
5:          for each cluster in cluters do
6:              disjPaths ← DisjointPaths(cluster)
7:              for each path in disjPaths do
8:                  initialNode ← GetInitialNode(path)
9:                  finalNode ← GetFinalNode(path)
10:                 contPointsCandidates.append((initialNode, finalNode))
11:             end for
12:         end for
13:     end for
14:     return contPointsCandidates
15: end function
```

Algorithm 3: Computation of the candidate control points.

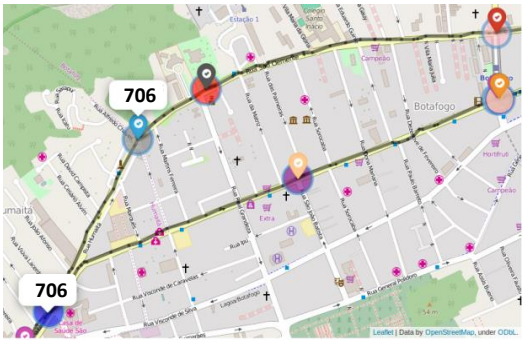


(a) Pairs of Control Points Candidates

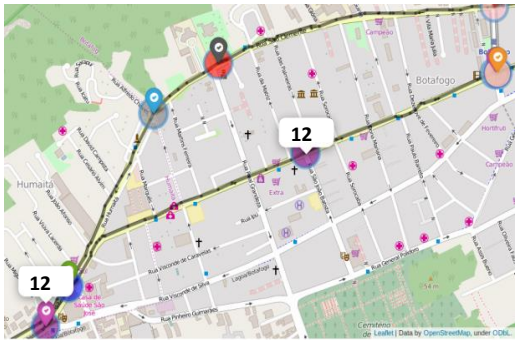


(b) Pair 704

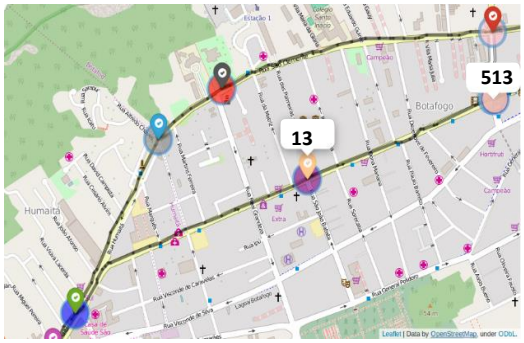
(c) Pairs 705



(d) Pairs 706



(e) Pairs 512



(f) Pairs 513

Figure 3: Candidates Control Points.

However, according to (ALEWIJNSE, S. P. A. et al., 2014), an optimal criteria-based segmentation is that with a minimal number of segments. On that basis, we analyze the candidate control points and identify thatnot all control point candidates have the same level of relevance in terms of traffic monitoring. For instance, one may discard intermediate nodes connecting two consecutive paths of the monitored road network that belong to the same street. In such cases, there is no significant difference in the bus routes serving each path. For this reason, both paths can be combined.

To address this issue and improve the quality of the segmentation process of the monitored bus network, Algorithm 4 refines the set of candidates for control points, using data provided by the road network map, as follows.

Intermediate Nodes. Line 2 assigns to the *ctrlPts* variable the list of candidate control points, passed as input. Line 3 computes the intermediate nodes between the list of candidates for control points. Given two pairs of candidate control points, an *intermediate node* is a node that is the end node of one of the control points and the initial node on the other.

Discard non-relevant intermediate nodes. For each node in the intermediate node list, Lines 5 and 6 extract, according to the direction and sense of the street,

Algorithm 4 Pseudocode to refine the list of control points

```
1:  function RefineControlPoints(contPointsCandidates)
2:      controlPoints ← contPointsCandidates
3:      intermNodes ← GetIntermediateNodes()
4:      for each node in intermNodes do
5:          prevNode ← GetPreviousNode(node)
6:          followNode ← GetFollowNode(node)
7:          streetName1 ← GetStreetName(prevNode,node)
8:          streetName2 ← GetStreetName(node,followNode)
9:          if streetName1 = streetName2 then
10:             controlPoints ← RemoveIntNode(node,controlPoints)
11:         end if
12:     end for
13:     return controlPoints
14: end function
```

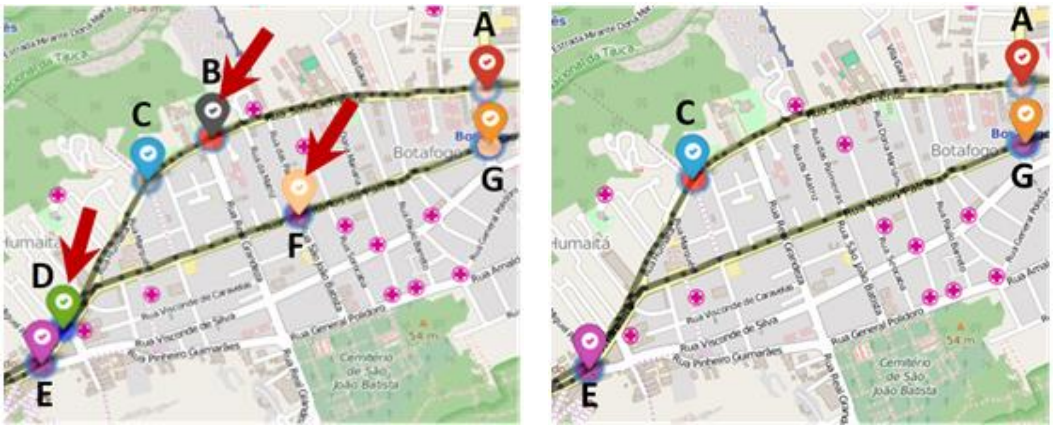
Algorithm 4: Refine the list of candidates for control points.

its previous and subsequent nodes in the monitored bus network. Neither the previous node nor the subsequent node must necessarily be candidates for control points. It occurs just when the path delimited by a pair of control points, where one of the points is an intermediate node, encompasses only one street segment.

Using data from the network map, a plausible name of the street that connects the previous node with the intermediate node can be obtained, as well as the name of the street that connects the intermediate node with the subsequent node. For this purpose, Line 7 and 8 call a crawler-function that processes machine-readable road tags and the semantic relations between them, to extract the name of the street in question, to which two given coordinate points belong.

Once the street names are found, Line 9 compares them. If the names are the same, both street segments belong to the same street, and it means that there is no change of street around the intermediate node. Therefore, both paths, where the intermediate node belongs can be joined into one to be monitored.

Line 10 removes the intermediate node from the list of candidate control points. This process is repeated until all nodes of the intermediate nodes list have been analyzed and the list of control points has been fully refined. Line 12 returns the list of control points as output, which define the monitored paths (See Figure 4).In this figure, note that a monitored path starts with a balloon that has a specific color and ends with a circle that has the same color.



(a) Intermediate nodes around which, there is no change of street

(b) Intermediate nodes were removed

Figure 4: Refining the set of Control Points.

4.5. Conclusions

In this chapter, in order to conduct a more comprehensive analysis of a bus network, we proposed an evolutionary model of the bus transport network using versions, a method to compute the monitored network associated to each version, and a method to segment the monitored network into monitorable paths.

The model includes topological and operational features of the bus network. Accordingly, a bus transport network consists of several versions of a bus network, where each version contains information about its structural characteristics, the bus routes that serve it, and statistics about the historical bus trips of such bus routes. This modeling permits analyzing the bus transport network as it evolves over time.

The two methods are based on the bus routes that serve the city and allows selecting the streets whose traffic can be monitored with the help of bus trajectories.

5
Travel Time Patterns

5.1.
Introduction

Travel time variability is an important quantitative measure to evaluate the behavior of the bus transportation system(YE et al., 2015) and the performance of traffic conditions(CHEN et al., 2010). Therefore, reliable data that allows to correctly estimate the bus travel time and infer if a bus travel time corresponds to a typical behavior or not is very useful for these assessment analyses.

In major metropolitan cities, buses are equipped with GPS devices. They operate for almost 24 hours per day, following regular itineraries and continuously transmitting their positions. Thereby, these buses generate a huge amount of data in the form of raw trajectories. Historical trajectory data generated by buses contain relevant, but hidden information about their operation. Data mining techniques provide us the opportunity to discover valuable knowledge, such as frequent and anomalous behaviors, which is useful for describing both past and current traffic behavior and predicting future traffic behavior. Especially, spatio-temporal pattern mining is the most intuitive and attractive approach to extract frequent behaviors in trajectory data (KANG; YONG, 2010). A common method for mining spatio-temporal patterns consists first in discretizing the space to identify the regions of interest within the trajectories and then applying temporal mining on the trajectory segment data corresponding to these regions (MAZIMPAKA; TIMPF, 2016).

In this chapter, we address the problem of discovering frequent travel time patterns of buses from historical GPS dataset of bus trajectories. For this purpose, we adopt a spatio-temporal pattern mining approach and also include the directional component of bus movement. Then, to extract frequent bus travel time patterns, we explore the historical bus data by segmenting the trajectories by their

spatio-temporal-directional characteristics and, after that, over the trajectory data resulting from the segmentation, we compute the average travel time.

Specifically, we implemented two algorithms to execute these operations. The first algorithm spatially and directionally segments the bus trajectories and computes the travel time that buses take to traverse each path of the monitored bus network version. The second algorithm temporally segments the trajectories that were previously segmented according to their spatial-directional characteristics and then computes the average of their travel times.

5.2. Estimating Travel Time

Once a Monitored Bus Network version is defined and segmented, it is possible to estimate the travel time that buses take to traverse each of its paths. In this section, we propose an algorithm (Algorithm 5) to execute this operation. It works as follow.

Algorithm 5 Pseudocode to estimate the travel time

```
1:  function TravelTime(day, monitPaths)
2:      for each path in monitPaths do
3:          linestring ← GetLineString(path)
4:          bufferRegion ← GetBufferRegion(linestring, distance)
5:          obsInsideBuffer ← GetObservations(day, bufferRegion)
6:          busList ← GetDifferentBuses(obsInsideBuffer)
7:          for each busLine, busId in busList do
8:              rightDirObserv ← GetRightObserv(busLine, busId)
9:              trips ← GetTrips(rightDirObserv)            ▷ interpolation
10:             for each trip in trips do
11:                 travTime ← ComputeTravelTime(trip)
12:                 saveTravTime(busId, busLine, path, T_i, T_f, travTime)
13:             end for
14:         end for
15:     end for
16: end function
```

Algorithm 5: Estimation of travel time.

Figure 5: Buffer Zone around the Voluntários da Pátria Street.

Buffer zone definition. The algorithm receives as input the monitored paths, and a period covered by the monitored bus network version validity period. For each monitored path, Line 3 extracts the LineString that joins the consecutive geographical positions forming the path.

Line 4 creates a buffer zone around the LineString, with a specific width. Note that the width value is computed as the sum of the width of the street under analysis and the GPS measurement error, which typically ranges from 5 to 10 meters. As a result, the buffer zone is a polygon, used to spatially delimit the raw bus GPS observations transmitted between a pair of control points. An example of the content of the GPS observation data is described in Table 5.

Spatial-directional segmentation of raw trajectory data. Line 5 executes a geospatial-temporal query to retrieve all GPS observations inside the defined buffer zone for the specified period. As illustrated in Figure 5, this query allows selecting GPS points that may not exactly fit road geometries, without having to execute (expensive) map-matching operations.

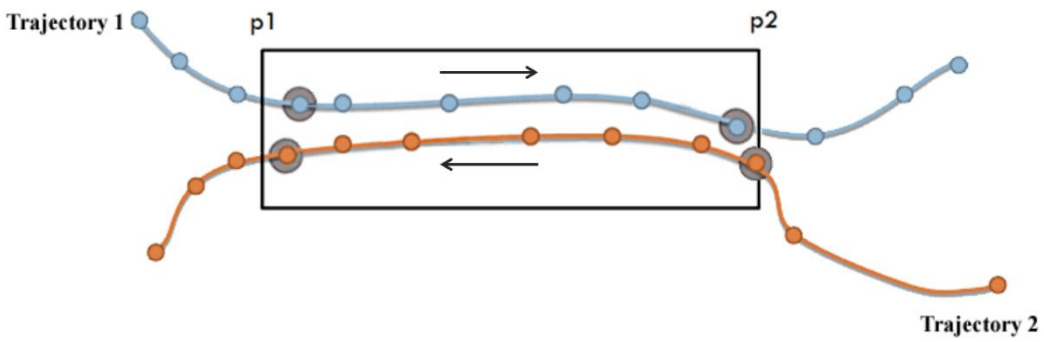
Line 6 finds all distinct buses (*busLine, busId*) that transmitted their positions within the buffer zone. Line 7 repeats the loop to read each bus found. Line 8 extracts only the observations that correspond to trips that go in the direction from the start to the end nodes of the monitored segment. Line 9 computes the trips.

Table 5: Example of GPS observations data.

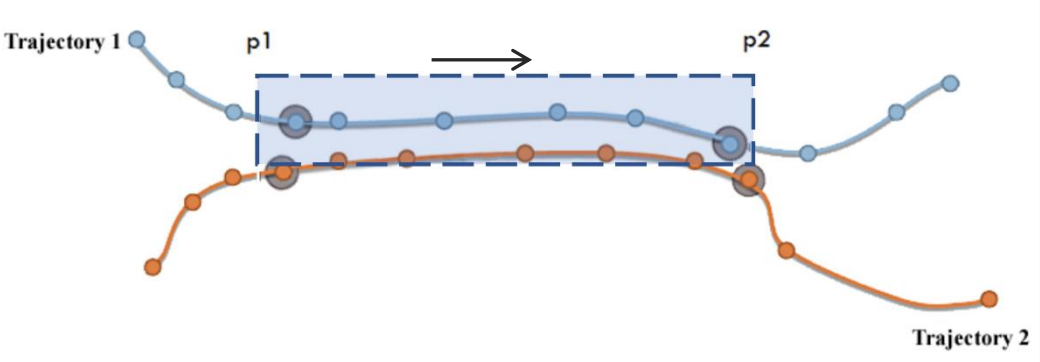
Timestamp	Bus_id	Line	Latitude	Longitude	Speed
13-09-2015 00:00:01	C27109	940	-22.827141	-43.294739	32.0
13-09-2015 00:00:01	C41401	303	-22.857653	-43.245167	0.7
13-09-2015 00:00:01	C50112	301	-22.929371	-43.253754	0.0
13-09-2015 00:00:01	C51512	738	-22.877365	-43.368198	0.0
13-09-2015 00:00:01	C72081	805	-22.889046	-43.292263	0.2
13-09-2015 00:00:01	C82596	363	-22.858412	-43.371071	0.9
13-09-2015 00:00:01	D58684	840	-22.841305	-43.371494	2.8
13-09-2015 00:00:01	C72081	805	-22.889046	-43.292263	0.2
13-09-2015 00:00:01	C82596	363	-22.858412	-43.371071	0.9
13-09-2015 00:00:01	D58684	840	-22.841305	-43.371494	2.8

Directional filtering is necessary since, for instance, for two-way streets, the buffer region around a one way may overlap the buffer region around the opposite way, as shown in Figure 6, and therefore, when performing a spatial query to extract GPS observations from one buffer region, observations from the other buffer region can also be captured.

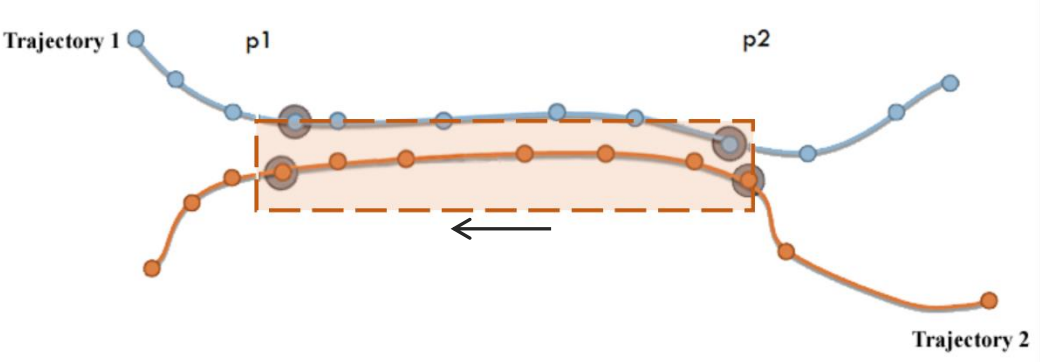
Travel time computation. For those trips for which the first or the last observation do not match the position of the start and the end nodes of the monitored segment, a linear interpolation is used to discover the timestamps when the bus passed through these end points (see Figure 7). Lines 10-13 compute and save the travel time for each trip. As a result of the algorithm, a travel time table, such as Table 6, is obtained. From each bus trip computed, the *MonitoredPath*traversed, the *Line* and the *ID* of the bus, the timestamp when the bus arrived in the monitored path (*Arrive_Hour*), and the timestamp when the bus left the monitored path (*Departure_Hour*)are registered in the table of trips.



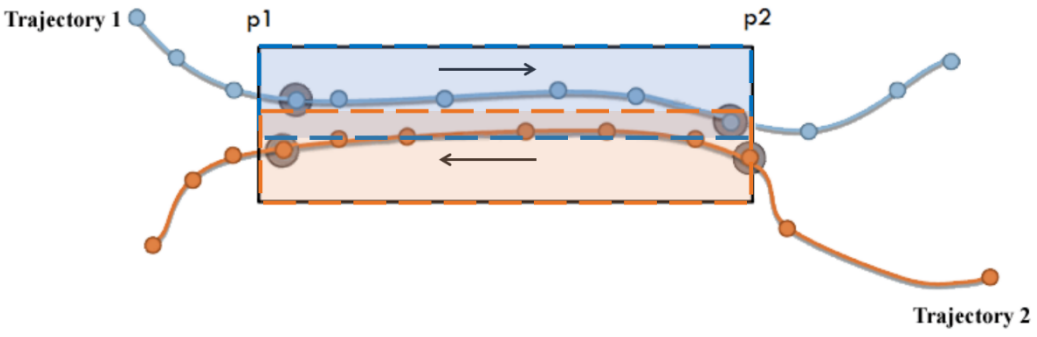
(a) Opposite trajectories of a two-way street



(b) Buffer region for the street from P_1 to P_2



(c) Buffer region for the street from P_2 to P_1



(d) Overlap of buffer regions

Figure 6: Overlap of buffer regions.

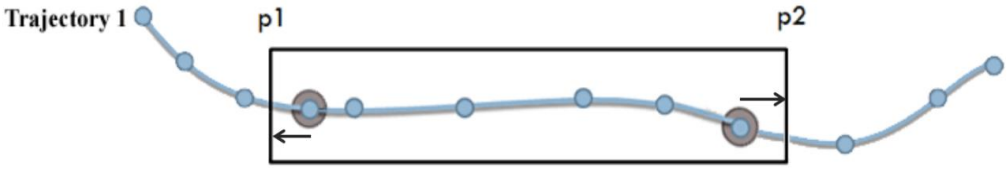


Figure 7: Interpolation of GPS observations inside the buffer region and nearby control points.

It is worth mentioning that Algorithm 5 computes the travel time of trips made in the period defined by the input parameter *day*, whose value may be set to be one day or multiple days belonging to the period of each monitored bus network version stored in the dataset. To repeat the computation for several days, one only has to pass a set of days as the value of the input parameter.

This algorithm was also implemented using two other variants. The first variant tracks all GPS positions of buses to know when they passed through the control points, and, thus, estimates the travel time. This implementation requires a map-matching process for all bus GPS positions.

The second variant creates a circle buffer zone around the control points of the monitored paths. It then captures the observations within these circles to determine when the buses passed by the control points and thus calculate their travel time. Considering that the occurrence of GPS observations within the circles depends on the speed of buses, then, when the buses run fast, many observations are lost. This fact causes the sample, to compute the travel time, not to be statistically significant. On the other hand, for the cases where there are multiple monitored paths, which start or end at the same control point, this variant has an additional shortcoming. It does not allow to identify from which of those monitored paths the buses came, which may result in erroneous estimations of travel time.

For these reasons, both implementations were discarded and a third variant, represented in Algorithm 5, was adopted.

Table 6: Example of trip table

Line	Bus_id	MonitoredPath	Arrive-Hour	Departure_Hour	Travel Time
309	C41407	MonitPath12	27/07/2015 7:03:47	27/07/2015 7:07:48	4,0166
177	A63543	MonitPath12	27/07/2015 7:05:15	27/07/2015 7:09:42	4,4500
172	A55157	MonitPath12	27/07/2015 7:10:27	27/07/2015 7:14:02	3,5830
177	A63508	MonitPath12	27/07/2015 7:12:12	27/07/2015 7:16:42	4,5000
172	A41319	MonitPath12	27/07/2015 7:16:48	27/07/2015 7:20:08	3,3300
177	A63501	MonitPath12	27/07/2015 7:19:02	27/07/2015 7:23:32	4,5000
316	C41424	MonitPath12	27/07/2015 7:23:04	27/07/2015 7:26:47	3,7166
178	A41047	MonitPath12	27/07/2015 7:30:51	27/07/2015 7:34:57	4,1000
177	A63528	MonitPath12	27/07/2015 7:39:56	27/07/2015 7:44:56	5,0000
172	A41213	MonitPath12	27/07/2015 7:44:49	27/07/2015 7:49:04	4,2500

5.3. Computing Travel Time Patterns

Algorithm 6 computes the travel time pattern of each path at different time period and time interval as follow.

Temporal segmentation of trajectory data. The algorithm receives as input a monitored bus network version and a dataset that contains all trips made during the validity period of a monitored bus network version. Line 2 retrieves the different paths/segments in which the monitored bus network version was previously segmented using the algorithm 4 described in the Section 4.4. Line 3 uses a loop to analyze each of these segments.

For each segment, based on an analysis of the travel time of buses that transit through it, the monitored bus network version validity period is partitioned

Algorithm 6 Pseudocode to compute the travel time pattern

```
1: function TravelTimePattern(MonitNetwork, Trips)
2:     monPaths ← GetMonPaths(MonitNetwork)
3:     for each path in monPaths do
4:         tempSegmentation ← GetTempSegmentation(path, Trips)
5:         periods ← GetPeriods(tempSegmentation)
6:         intervals ← GetIntv(tempSegmentation)
7:         for each P in periods do
8:             t ← 0
9:             while t < numberOfInterv do        ▷ e.g. 24 interv of one hour
10:                tripSet ← GetTrips(t, P, path, Trips)
11:                n ← tripSet.count()                          ▷ Number of trips
12:                travTimePattern ← (1/n) Σ_{i=1}^{n} (tripSet(i).GetTravTime())
13:                saveTravTimePattern(path, P, t, travTimePattern)
14:                t ← t + 1
15:            end while
16:        end for
17:    end for
18: end function
```

Algorithm 6: Computation of the travel time pattern for all paths of a Monitored Bus Network version at different periods of time and time intervals.

(Line 4). It is important to note that the monitored bus network version validity period can be partitioned differently for each monitored segment, since each segment has its own behavior regarding the travel time of the buses.

It is worth mentioning that the implementation of this temporal partitioning function is not objective of this thesis. We assume that it is already implemented, and we only use it. On the basis of this assumption, for instance, for a particular segment *S*, a temporal partitioning can be as follows: school classes period and school vacations, both divided into days of the week, with the weekdays separated in holidays and working days. The temporal segmentation made up to this level of granularity defines time periods denoted as *P* (Line 5).An example of a time period *P* would be “all Mondays during a school classes period that are working days”.

Also, taking into account that the traffic has a different behavior during the day, since it is in correspondence with the activities of citizens, then, each period is divided by intervals (e.g. 24 fixed time intervals of one-hour each). The

temporal segmentation made up to granularity level of an interval of the day defines *T* (Line 6). An example of a time interval *T* would be "Monday, August 10th, from 8:00 AM to 9:00 AM". We also say that a time interval *U*, such as "Monday, August 17th from 8:00 AM to 9:00 AM", is *consistent withT* w.r.t. time period *P* (see Figure 8). This temporal segmentation can be as fine grained as it may be considered.

The basic idea behind using the spatial-directional and temporal segmentation of bus trips is to select a collection of "similar" historical trajectories (i.e., that share similar spatial, directional and temporal characteristics) to compute the bus travel time patterns.

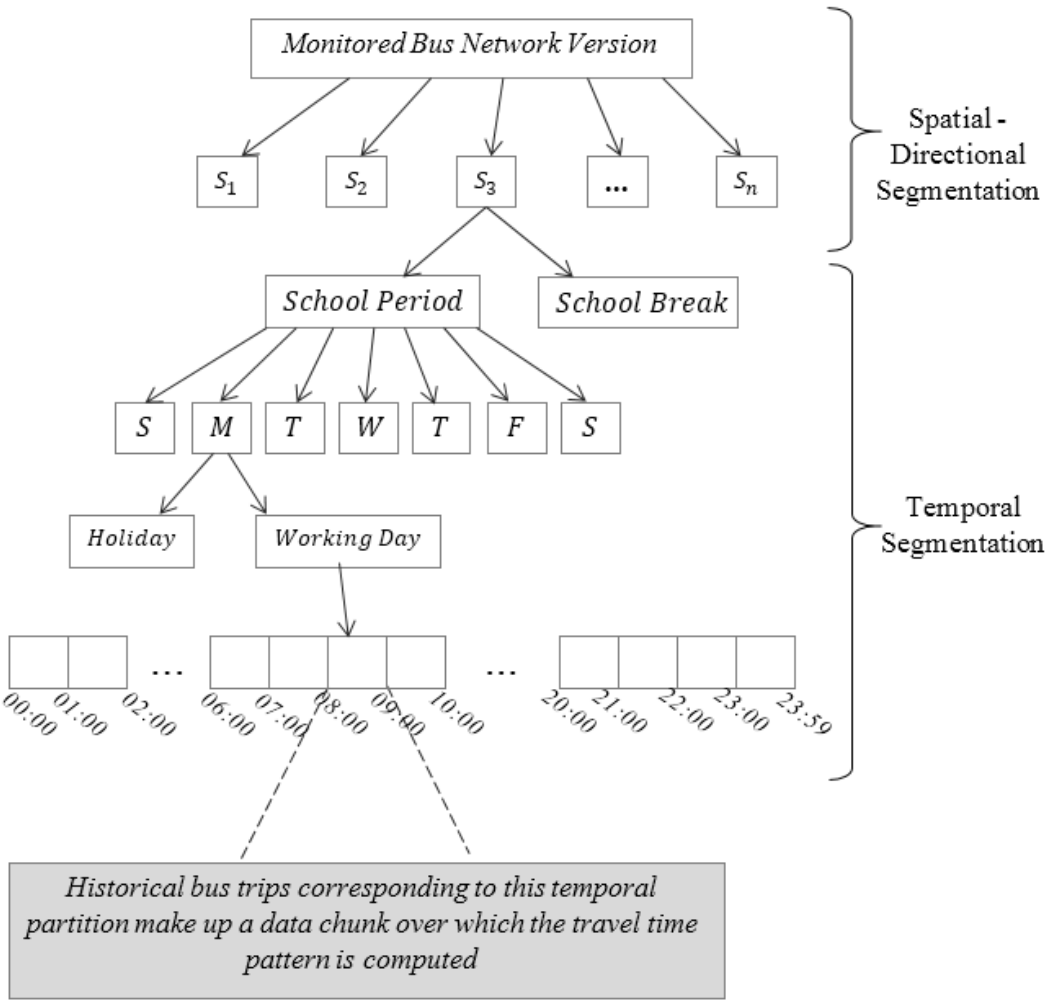


Figure 8: Example of temporal segmentation of a Bus Network Version validity period for particular path (segment).

Travel Time Pattern Computation. Lines 7 and 9 use loops to respectively step through each period of time in *P* and each time interval in *T*. According to this temporal partitioning, Line 10 recovers trips from the trip dataset that corresponds to the monitored path *p*, and whose time of entry into the path belongs to the time period and time interval being analyzed. Line 11 counts the number of these trips. The travel time pattern for this particular time period and this specific interval is computed in Line 12 as the mean travel time for the same path during the referred time period and at the referred time interval. Line 13 associates to each monitored path for a given period of time and a given time interval, a specific travel time pattern.

The statements between Lines 10 and 14 are repeated until all paths, time periods and time intervals of the day have been examined.

Since Algorithm 6 computes the travel time patterns of all segments of a monitored bus network version, in order to maintain a versioning of the travel time patterns of a monitored bus network, the same process should be repeated for all versions of the monitored bus network.

5.4. Conclusions

In this chapter, we proposed a strategy for mining frequent travel time patterns in historical bus trajectory data generated by GPS. The strategy is based on the spatio-temporal-directional segmentation of bus trajectories and on the average computation of the travel time of the trajectory segments resulting from this three-dimensional partition.

Frequent travel time patterns based on spatio-temporal-directional features of bus trajectories help us interpret the bus traffic behavior and identify anomalous traffic patterns from large amounts of multidimensional bus trajectory data. It also allows to understand, in more details, the behavior of the traffic of a city, determining the travel time variability at different times of the day, days of the week, and periods of the year and identify the peak traffic hours, which correspond to the daily activity of the citizens.

Despite the fact that, in order to compute the travel time patterns, Algorithm 6 computes the average, and not a generic time travel statistic, we note that this algorithm can be easily modified to account for more general settings.

6
Traffic Anomalies

6.1.
Introduction

Traffic anomalies represent unusual and significant changes in the road network traffic levels, that can often span multiple adjacent street segments. They can be produced by the influence of planned or unplanned disturbances, such as special events (i.e. sporting events, concerts, fairs, and conventions), road works, work zone closures, and traffic accidents and weather respectively. Traffic anomalies can have a negative impact on the transportation system of the cities, affecting the normal flow of traffic and causing high levels of congestion, which may result in a considerable waste of time by quite a large number of citizens (drivers and passengers), increase air pollution and jeopardize safety.

The detection of traffic anomalies and understanding their nature are important tasks since they allow for proactive planning and rapid responsive actions to be taken by the city authorities and traffic managers to mitigate them in the shortest possible time, and thereby return the transportation system to normal conditions. But, at the same time, the detection of traffic anomalies on non-recurrent congestions is one of the major challenges for traffic managers (VUCHIC, 2005). There are several indicators that can be used to identify the occurrence of a traffic anomaly. They include travel time, average speed and traffic flow of vehicles on a particular street. For the purpose of this thesis, we focus on bus travel time as an indicator to detect traffic anomalies.

Once a traffic anomaly took place, besides detecting its occurrence, another significant aspect is to determine how much it impacted the traffic conditions and citizens. This impact can be estimated using as metrics the travel time delay, the incident duration, the incident propagation and the number of people affected. In

this work, we focus on travel time delay and incident duration to evaluate the impact of traffic anomaly.

The remainder of this chapter is structured as follows. Section 6.1 describes the real-time and non-real-time strategies used to detect traffic anomalies. Section 6.2 presents the technique to estimate the severity of a traffic anomaly. Section 6.3 explains how to estimate the impact of a traffic anomaly in terms of incident duration and travel time delay. Finally, Section 6.4 concludes the chapter.

6.2. Detection of traffic anomalies

Traffic anomalies are deviations from the typical traffic behavior patterns. The traffic state presents two different behaviors: on one hand, a stable and predictable behavior due to usual traffic patterns (e.g. daily travel time to traverse the street segments of the road network); on the other hand, an abrupt and uncommon behavior due to unusual incidents.

The observations generated by the GPS of buses during their operation through a monitored bus network represent a very useful data source for the detection of traffic anomalies. Specifically, by analyzing both the historical and the real-time streams of bus trajectory data, and computing the travel time of bus trips on each segment (without preferential bus lanes) of the monitored bus network, we can detect traffic anomalies and classify them based on their severity degree.

If buses in a given area are not running according to the usual schedule, then a traffic perturbation is the most probable cause. In that sense, if the travel time spent by a set of buses to traverse any street segment of the bus network, during certain periods of time, deviate from the typical patterns, then we can say a traffic anomaly seemingly occurred.

The analysis of these data can also help: delimit the location of the incident; compute the number of lanes blocked and the total length of the road segments affected by the traffic incident; and estimate the incident severity.

6.2.1.
Non-Real-time traffic anomaly detection strategy

The non-real-time traffic anomaly detection problem is defined as follows: “Given a bus network version B_t and a set of bus trips β over each monitored segment S of B_t, detect if the travel time to traverse S of a subset of trips deviated significantly from the pattern, and when the deviation occurred”.

To address this problem, in this section, we propose an algorithm that uses an univariate Statistical Quality Control technique (SQC)(MONTGOMERY,

Algorithm 7 Pseudocode to detect traffic anomalies in non-real time

```
1: function AnomalyDetection(MonBusNetVer, Trips)
2:     monPaths ← GetSpatioDirectSegmentat(MonBusNetVer)
3:     tempPartitions ← GetTemporalSegmentat(MonBusNetVer)
4:     for each path in monPaths do
5:         for each Period, Interv in tempPartitions do
6:             tripSet ← GetTrips(Trips, path, (Period, Interv))
7:             travTimeVector ← getTravTime(tripSet)
8:             n ← travTimeVector.size()
9:             t̄ ← (1/n) Σ_{i=1}^{n} (t_i)          ▷ Average Travel Time for all trips
10:            MR̄ ← (1/(n−1)) Σ_{i=2}^{n} (|t_i − t_{i−1}|)
11:            σ ← MR̄ / d_2
12:            UCL ← UpperControlLimit(t̄, σ)
13:            LCL ← LowerControlLimit(t̄, σ)
14:            slowAnomalousTrips ← GetAnomalyTrips(tripSet, UCL)
15:            fastAnomalousTrips ← GetAnomalyTrips(tripSet, LCL)
16:            probableTrafAnom ← ConsecSlowAnomTrips()
17:            noiseAnomSlowTrips ← IsolatSlowAnomTrips()
18:            traffAnomTrips ← NotPeriodBehavior(probableTrafAnom)
19:            saveTraffAnom(path, Period, Interv, t̄, UCL, traffAnomTrips)
20:            noiseAnomFastTrips ← IsolatFastAnomTrips()
21:            probTrafAnomFastTrips ← ConsecFastAnomTrips()
22:            traffAnomFastTrips ← NotPeriodBehavior(probTrafAnomFastTrips)
23:            sampleRefinement(tripSet, noiseTrips, traffAnomTrips)
24:            updateTravTimePattern(refinedSample)
25:            updateUCL(refinedSample)
26:            updateLCL(refinedSample)
27:            ReanalyzeTrafAnom()
28:        end for
29:    end for
30: end function
```

Algorithm 7: Detection of Traffic Anomalies in non-real-time.

2009) to analyze a historical dataset of bus trips that share the same spatial, directional, and temporal characteristics; and, from this analysis, determine control limits for the travel time of buses. Then, based on these limits, the algorithm identifies trips that correspond to a traffic anomaly. Algorithm 7 shows the pseudo-code, whose main steps are described as follows.

Determining the subset of data to be statistically analyzed. The algorithm receives as input a monitored bus network version and a dataset that contains all trips made in the network. From each bus trip in the dataset, the bus line, the bus ID, the traversed segment, the date and time at which the bus arrived at the segment (trip start time), the date and time at which the bus left the segment, and the travel time used to traverse the segment is registered, as illustrated in Table 6.

The monitored bus network version was previously segmented, spatially and directionally, using Algorithm 4 described in Section 4.4. In Line 2, the monitored paths resulting from this segmentation are retrieved. The monitored bus network version validity period was previously partitioned for each monitored path using Algorithm 6 described in Section 5.3. The result of this temporal partitioning is recovered in Line 3.

Line 4 describes a loop that continuously executes the statements contained in Lines 5-27 until all monitored paths of the network have been analyzed. Line 5 uses a loop to step through each temporal partition, which is composed of a pair of a period of time and a time interval. For the sake of simplicity, we will briefly call a combination of a time period and a time interval as a temporal partition. For each monitored path, the trips that were carried out within the temporal partition under analysis are retrieved from the dataset sorted ascendingly by their start time (Line 6).

Statistical Quality Control (SQC). Line 7 builds a vector $\boldsymbol{v}$ with the travel time of the trips obtained in the previous step. Note that this vector is already sorted by date and start time of those trips. Over this vector, the univariate Statistical Quality Control (SQC) technique is applied as explained below. SQC is used for detecting shifts with respect to the mean and to the standard deviation of the data. In this thesis, we use the difference between each observation with respect to the mean as a measure to individually analyze how the travel time of each bus trip

deviated from the typical pattern defined by the historical data. This measurement can provide for each bus trip an assessment of whether it is anomalous or not.

At this point, it is important to note that after several tests of normality to the periodically collected data concerning the travel times of the bus trips, which share the same spatial, directional and temporal characteristics, we conclude that under these conditions, the travel time variable follows a normal distribution, as shown in Figure 22, 23, and 24.

Lines 8 and 9 respectively compute the size (n) and the average ($\bar{\tau}$) of $\boldsymbol{v}$. Note that the computation of the average was actually executed in Algorithm 6 described in Section 5.3. However, we describe it again here to facilitate the explanation of the technique. A quadruple $\bar{\tau}[S, \beta, T, P]$represents the average travel time to traverse S, observed in a set β of bus trips, for time intervals consistent with T, over a period of time P.

Taking into account that the vector $\boldsymbol{v}$ contains observations (travel time) of individual trips periodically collected, a *control chart for individuals* is useful (MONTGOMERY; RUNGER, 2010). Control charts are the key tools in SQC. They are also known as *Shewhart charts* or *process-behavior charts*(SHEWHART; DEMING, 1939). A control chart for individuals uses the moving range of two successive observations to estimate the dispersion in the data over time. The moving range is defined as:

$$MR_i = |t_i - t_{i-1}| \quad (1)$$

where t_i is the travel time value for trip i. $\overline{MR}$ is the average of these ranges, which is computed in Line 10 as:

$$\overline{MR} = \frac{1}{n-1}\sum_{i=2}^{n}(|t_i - t_{i-1}|) \quad (2)$$

Then, Line 11estimates the standard deviation as:

$$\sigma = \frac{\overline{MR}}{d_2(m)} \quad (3)$$

where m= 2, because two consecutive observations are used to calculate a moving range. Therefore,$d_2(m)$ = $d_2(2)$ = 1.128, which is a predefined value registered in Table 9 on Appendix A.

Based on $\bar{\tau}$ and σ, the *Upper Control Limit* (UCL), and the *Lower Control Limit* (LCL) are calculated in Lines 12 and 13, respectively, as follows:

$$UCL = \bar{\tau} + \lambda * \sigma (4)$$

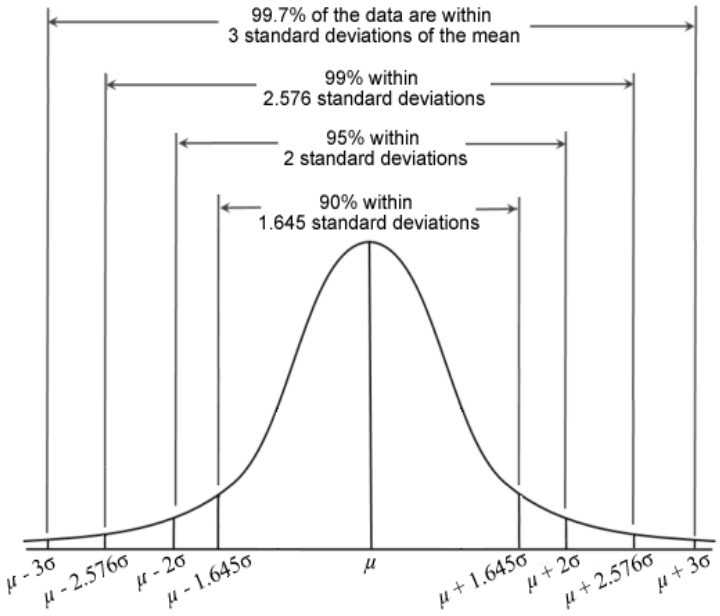
$$LCL = \bar{\tau} - \lambda * \sigma (5)$$

where the factor λ determines the confidence interval. For this case, we apply the confidence level defined by (TUROCHY; SMITH, 2000) for discerning between normal and anomalous observations of traffic domain variables. They define that normal observations are represented by about the 90 percent of the area under the curve of the probability density function, which means that λ = 1,645, as illustrated in Figure 9. On that basis, the control limits are formalized as:

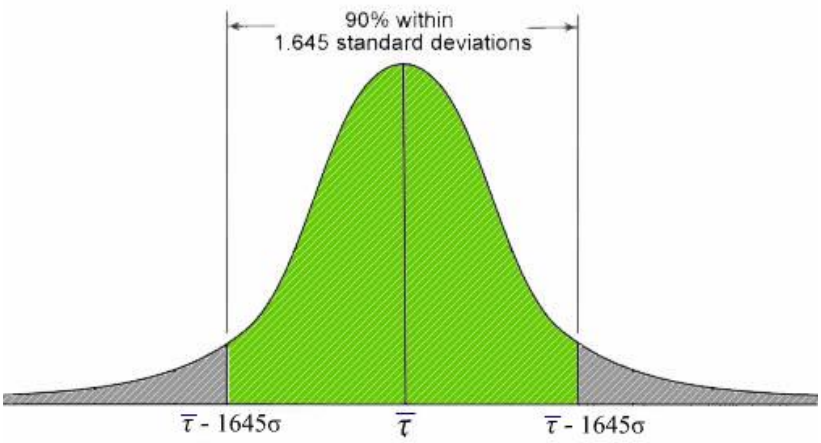
$$UCL = \bar{\tau} + 1{,}645 * \sigma \quad (6)$$

$$LCL = \bar{\tau} - 1{,}645 * \sigma \quad (7)$$

Note that LCL cannot be less than 0 because no trip has a travel time less than zero. The upper and lower control limits are symmetrically chosen about the average value and represent the maximum and minimum of expected bus travel time for a particular road segment S, in the time period P, at time interval T. They are denoted as UCL [S, β, T, P, $\bar{\tau}$,σ] and LCL [S, β, T, P, $\bar{\tau}$, σ]. Therefore, we use



(a) Probability Distribution around the mean in a Normal Distribution (percentile).



(b) Confidence interval of 90% represents normal trips (green region) and the remainder anomalous (gray region).

Figure 9: Control Limits to classify the bus trips in accordance to the Probability Distribution.

the control limits to determine whether trips are normal or anomalous. Bus trips whose travel time fall within the tolerance region, defined between the LCL and the UCL, are considered normal, otherwise, are anomalous.

The control chart in Figure 10plots the travel time of bus trips on specific segment versus the number of trips, together with its corresponding control limits. The normal trips are marked as blue circles, while the *anomalous* trips are marked as red circles on the chart. An *anomalous delayed trip* is a bus trip whose travel time exceeds the UCL value and indicates that the bus ran late with respect to the pattern (Line 14). On the other hand, an *anomalous fast trip* is a bus trip whose travel time is less than the LCL value and indicates that the bus ran faster with respect to the pattern (Line 15). Both are classified as *anomalous trips*. This classification is registered into the database. However, for the purposes of detecting travel time anomalies that indicate traffic congestion, we focus on anomalous delayed trips.

Traffic Anomaly Detection. After the individual classification of bus trips in anomalous or not and the differentiation between the anomalous fast trips and anomalous delayed trips, the next step is to determine which of the latter actually correspond to anomalous traffic conditions. For such purpose, we analyze the environment of detected anomalous delayed trips, specifically the neighbors of them. In this regard, we define that the existence of more than one consecutive

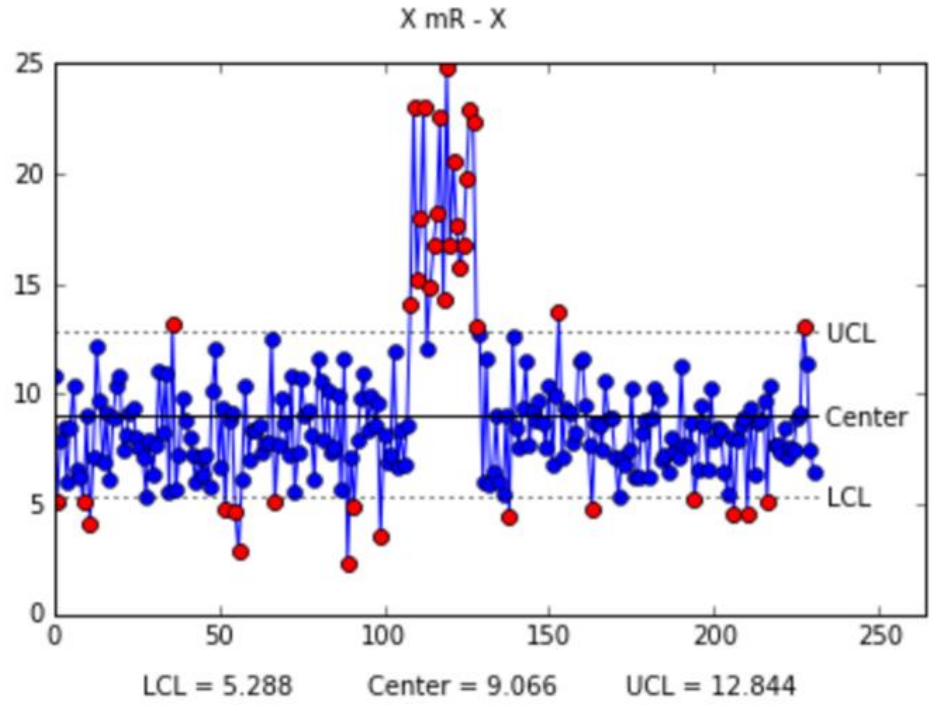


Figure 10: Statistical Quality Control chart whit control limits:
$UCL = \bar{\tau} + 1{,}645 * \sigma$ and $LCL = \bar{\tau} - 1{,}645 * \sigma$.

anomalous delayed trip in the same group of trips, states that a traffic anomaly probably happened (Line 16).

The fact that we indicate that a probable traffic anomaly occurs when there is more than one consecutive delayed bus trip is justified since the anomalous travel time of a bus trip has two main meanings: particular bus situation, which represents an outlier (noise) or generalized delayed travel time due to anomalous traffic conditions. Therefore, one anomalous delayed trip by itself is not sufficient to ensure the occurrence of an anomaly in the transit to traverse *S* at time interval *t*. Accordingly, and as shown in Figure 11, this method considers temporarily isolated anomalous delayed trips as noise (Line 17) and consecutive anomalous delayed trips as probable traffic anomalies.

Intuitively, in the same way that a single anomalous delayed trip in the midst of a chronological sequence of non-anomalous trips does not represent a traffic anomaly, nor does a single non-anomalous trip in the midst of a chronological sequence of anomalous trips mean the cessation of a traffic anomaly at that time.

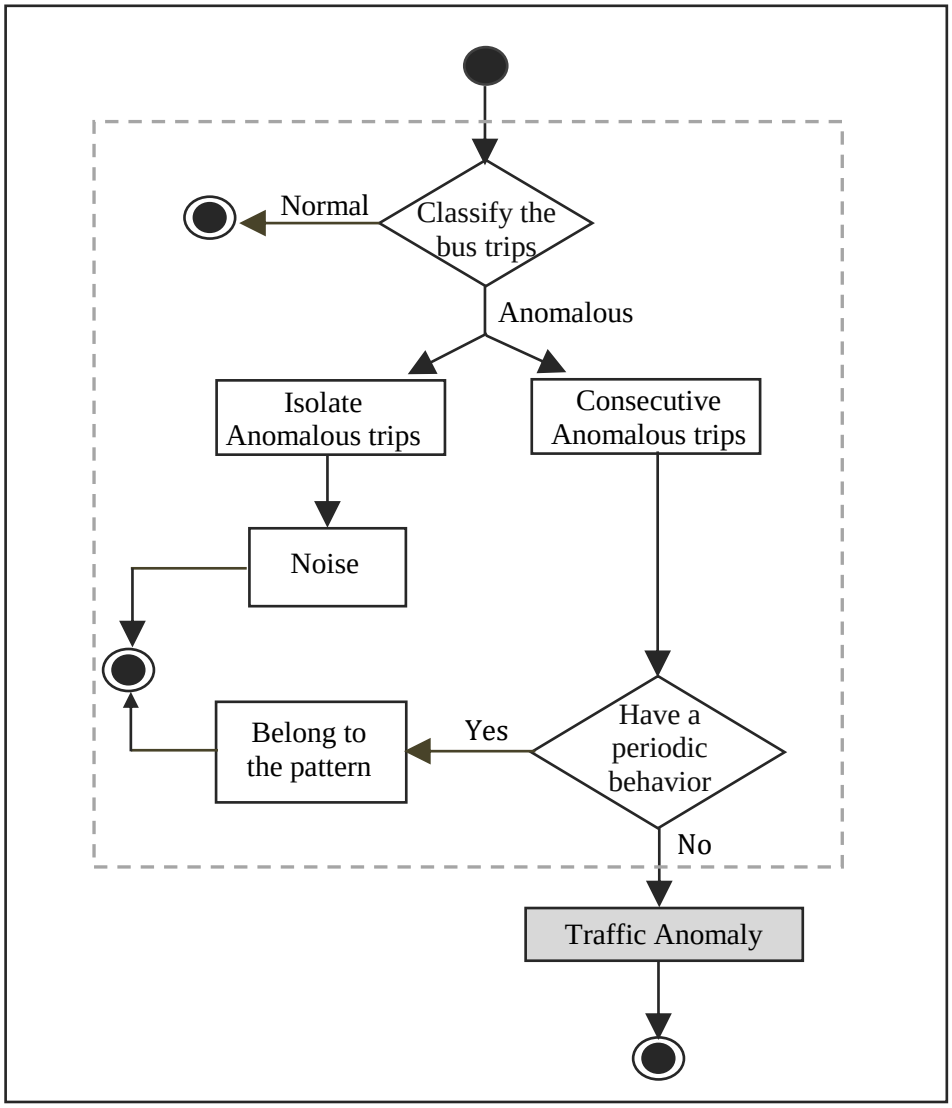


Figure 11: Scheme for traffic anomaly detection.

Then, the consecutive anomalous delayed trips that probably represent a traffic anomaly are analyzed with the objective of determining if they respond to periodic behavior pattern in Line 18. This verification is necessary because there may be segments where a traffic anomaly has never occurred, however making use of the SQC technique, even without traffic anomalies, there will always be values that are above the control limits (above 90% of the probability function for a Normal Distribution). As illustrated in Figure 11, if some consecutive anomalous trips are periodic, it means that they are part of the pattern, otherwise they actually represent a traffic anomaly, denoted $\alpha[\delta, S, P, T, \bar{\tau}, UCL]$, such that δ represents a set of anomalous delayed trips with respect to the average travel time $\bar{\tau}$ and upper control limit UCL, to traverse the segment S, during the period P, at

interval T. Line 19 saves the information about the detected traffic anomalies in the database.

Travel Time Pattern Refinement. The same analysis done for the anomalous delayed trips to identify if they are isolated anomalous trips, or when they are consecutive to verify if they are repeated periodically, is also done for the anomalous fast trips in the Lines 20-22, with the aim of identifying those anomalous fast trips that do not correspond to periodic behavior. Then, both, anomalous fast trips and anomalous delayed trips that do not correspond to a bus travel time periodic behavior for S, during P, at T are removed from the sample in Line 23. Then, Lines 7-23 are repeated until this sample is completely clear of noise and trips belong to traffic anomalies.

Finally, for the cleaned sample, the travel time pattern$\bar{\tau}$, σ, UCL, and LCL are recomputed and stored in the database (Lines 24-26) for further analysis, specifically, for real-time automatic traffic anomaly detection. It is important to note that these values must be recalculated with a certain frequency to keep them updated in accordance with the evolution of traffic conditions. Then, the anomalous delayed trips found, corresponding to traffic anomalies, are again analyzed with respect to the resulting pattern to estimate how much their travel time deviated (Line 27). This iterative refinement of the sample in non-real-time allows making a better real-time automatic traffic anomaly detection and classification.

In this way, all segments of the monitored bus network version are analyzed in order to detect the occurrence of traffic anomalies in them. Considering that the volume of historical data to process is very large, and aiming to accelerate the method speed, then a distributed scheme of this algorithm was designed, where each processing node receives a segment *S*, over which the tasks described in Lines 5-27 are executed.

6.2.2. Real-time traffic anomaly detection strategy

In this section, we address the real-time traffic anomaly automatic detection problem, which is defined as follows: “Given a set of monitored segment

S^+belonging to the bus network version B_t , a time period *P*, a time interval *T*, and bus GPS data stream, detect in real-time if the travel time of buses to traverse each *S* in $S^+$is deviating significantly from the pattern."

To solve this problem, we propose a strategy, which is relatively similar to that discussed in the previous section, but with the difference that the non-real-time strategy focuses on identifying anomalies from a historical dataset, whereas the current strategy constantly analyzes the arriving GPS data stream generated by buses moving through the monitored bus network to detect the occurrence of traffic anomalies in real-time. However, both strategies use the results of offline processing of historical data for the detection, specifically the control limits for travel time.

The real-time strategy includes an algorithm that uses the geofencing technology, and the travel time limit defined for each segment, to monitor all segments of the monitored bus network.

Geofencing is a location-based technology that is commonly used for monitoring purposes (Noei et al. 2014) and geospatial information alerting (AYOB, 2015). A *geofence* is a virtual perimeter for a real-world geographic area (STATLER, 2016) that represents a location of interest. A geofence may be created in a variety of configurations, such as a circular area defined by a radius around the location, or a polygonal area defined by a set of latitude and longitude coordinates (CHEUNG, 2016).

To apply the geofencing technology in real-time, two off-line pre-processing steps must have been completed: (1) definition of geofences, and (2) specification of geofence observers.

Definition of geofences. For tracking and monitoring the movement of buses through the monitored bus network in real-time, we use the buffer regions defined in Section 5.2 to delimit each monitored segment of a bus network version. Each buffer region is understood as a geofence. Thus, each monitored segment *S* of the bus network version B_t has a geofence boundary area g associated with it. These geofences describe polygonal areas around the segments with predefined and static geographical boundaries represented by coordinate points (latitude, longitude), as can be seen in Figure 12.

Figure 12: Geo-fence defined for the monitored segment that belong to Voluntários da Pátria Street.

We recall that the geofences do not overlap in space.

Specification of geofence observers. For each geofence, there is an observer that is continuously monitoring the buses in that area. At any time, abus network version has as many active geofence observers as monitored segments. For each geofence observer, a set of specifications is defined. According to (CHEUNG, 2016), a geofence observer may be specified using (1) Boundary Definition, (2) Criteria, and (3) Basic Functions. Conforming to this model a geofence observer is specified as follows:

1. Boundary Definition: The geofence, where the observer is responsible for monitoring.
2. Criteria: The geographic coordinates (latitude and longitude) of the position of all buses operating inside the geofence. They represent the attributes that will be checked. Other criteria, for instance, could be the geographic coordinates of buses from a specific line, if we want to monitor a single bus line; or the geographic coordinates of a bus with a particular identifier, if we want to monitor a specific bus.
3. Basic Functions: (1) Identify if a bus entered the geofence; (2) moved inside the geofence; or (3) exited the geofence (i.e. the bus is inside a neighboring/external geofence), see Figure 13. Each basic function, based on the defined criteria, initiates the execution of other specific

Figure 13: Monitoring buses using Geo-fence.

functions as explained below.

The specification of each geofence is saved in the persistent database.

Relying on this basis, Algorithm 8 detects traffic anomalies in real-time as follows:

Live storing and replication of the observations. The algorithm receives as input the geofences of amonitored bus network version and a continuous data streams of the instant positions of buses operating in this network. So, it is important to recall that, to perform the detection of anomalous trips in real-time, the result of the live geolocation data processing is compared with the travel time pattern extracted from historical data.

For high-speed processing purposes, we propose to use a buffer, which represents a primary memory, in addition to the historical database, which represents the secondary memory. The buffer will temporally store the bus GPS data streams that continuously arrive. When the GPS observations arrive, they are replicated by a gateway to each geofence observer simultaneously (Line 3). Specifically, the gateway moves the observations from the main buffer to the buffer of each observer as illustrated in Figure 14. This scheme allows to reduce the processing load over the main buffer and to avoid that new observations that arrive are lost.

Algorithm 8 Pseudocode to detect traffic anomalies in real-time

```
1:  function RealTimeAnomalyDetection(geofences, streams)
2:      for each g in geofences do
3:          ReplicateToObserver(streams)
4:      end for
5:      Each node v performs the following actions concurrently with all other
    nodes
6:      if obsevation ∈ geofence then
7:          if aBusEnteredInGeofence() then
8:              onBusEntered(busId)
9:          else if aBusMovingInGeofence() then
10:             onBusMoved(busId)
11:         else if aBusExitedGeofence() then
12:             onBusExited(busId)
13:         end if
14:     else
15:         RemoveFromObserTable(observation)
16:     end if
17:     return trafficAnomalies
18: end function
```

Algorithm 8: Detection of Traffic Anomalies in real-time.

Monitoring and tracking of buses by the geofence observer. Once the GPS observations are in the observer's buffer, the observer selects two groups of observations for the processing (Line 6). The first group includes the observations emitted within the limits of its monitored area; and the second group includes the observations corresponding to the first GPS readings of tracked buses outside the geofence. Line 15 removes all other observations from the observer's buffer.

Note, that the observer's buffer could have a spatial index to speed up the response time of these geospatial queries. After that, the observer verifies for each resulting observation, if it corresponds to a bus that entered, moved or exited the geofence (Line 7, 9, and 11).Depending on the event, one of the following location-based functions is executed: *onBusEntered*, *onBusMoved* and *onBusExited* in Line 8, 10 and 12 respectively.

The *onBusEntered* function, according to the timestamp and the geo-position of a bus that entered in the geofence, invokes an interpolation function to compute the time when the bus actually passed by the initial control point of the segment. Then, the time when the bus *busId*, that serves to the bus line *busLine*, entered in the geofence, is saved into the database.

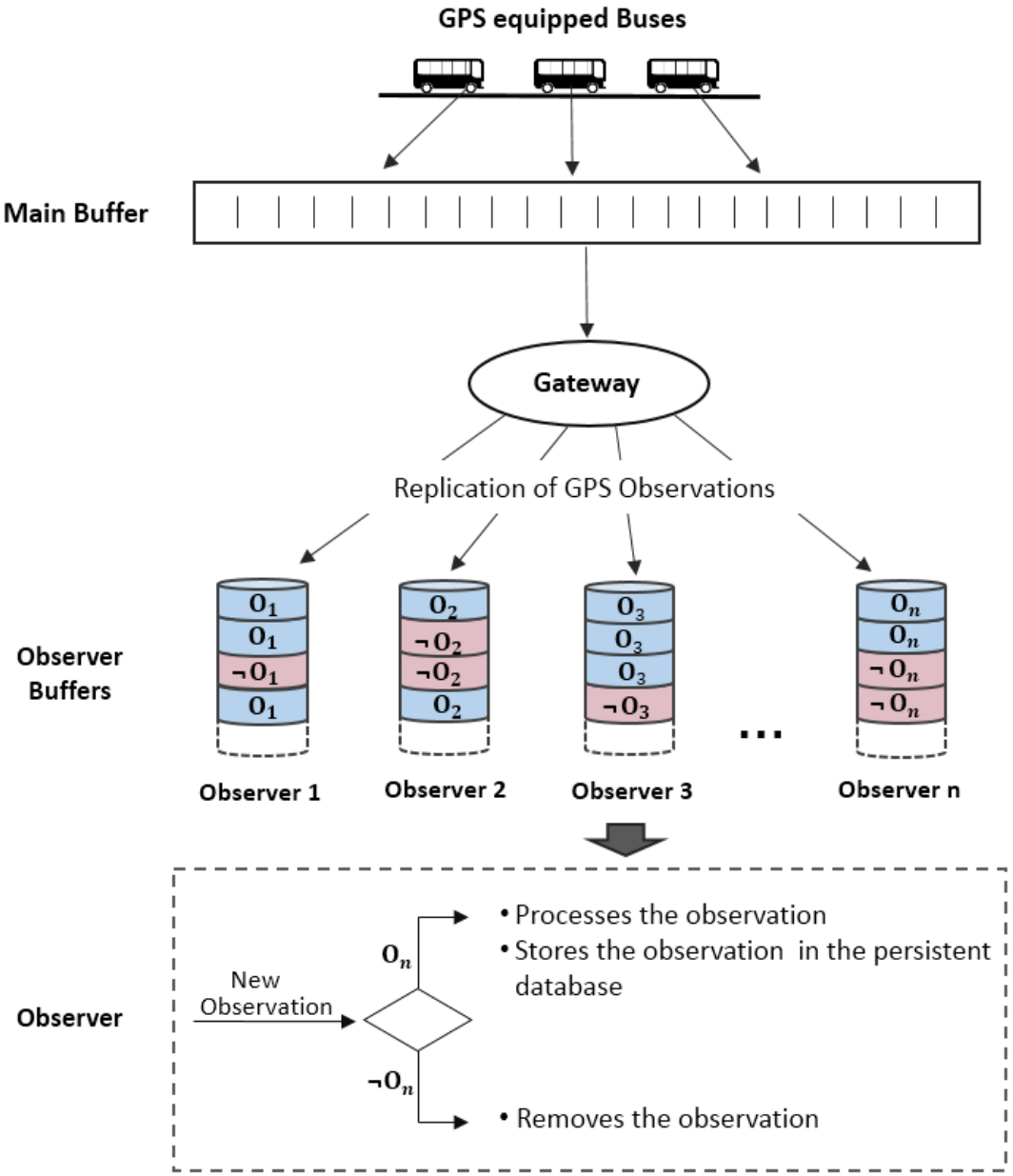


Figure 14: Real-Time bus GPS data stream processing.

The *onBusMoved* function, for each new bus observation emitted within the current geofence, the travel time wasted by bus to go from the beginning (initial control point n_i) of the monitored segment to the current location is computed. If this travel time exceeds the UCL[S, β, T, P, $\bar{\tau}$, σ] previously obtained in the non-real-time strategy, then the bus is marked as running late with respect to the travel time pattern to reach the final control point of S (n_{i+1}).

If more than one bus that entered in the geofence chronologically consecutive to this bus is also running late, then an alert is released to indicate the

possible existence of a traffic anomaly. Afterward, the trips marked as delayed are again analyzed to rule out that their delay corresponds to a periodic behavior (pattern). If their behavior corresponds to the pattern, these anomalous trips are discarded (Figure11); otherwise, an alarm is raised to indicate that a traffic anomaly is occurring. Then, the traffic anomaly α [δ, S, P, T,$\bar{\tau}$, UCL] is saved in the persistent database.

The *onBusExited* function extracts from observer's buffer the observations corresponding to the first GPS readings of tracked buses outside the geofence. Then, for each of those buses, the function performs an interpolation to estimate the specific time when it left the monitored segment (i.e. the bus passed through the final control point of the monitored segment). With this information, the total travel time wasted by this specific bus to traverse segment *S*, in time period *P*, at interval *T* is computed and saved in the database. Then, automatically, this bus is removed from observer's buffer of the segment S.

Note that, with a certain frequency, a garbage collector must be executed over the buffer of the observer to reclaim memory occupied by observations that are no longer in use.

In order to simultaneously monitor all geofences of the bus network, and with the aim of detecting traffic anomalies as nearly as possible to real-time, the job can be distributed across multiple processors and machines to achieve incremental scalability, so that each processing node is responsible for the functions of one geofence observer. Thereby, all nodes are concurrently analyzing the instant position of buses inside their respective geofence to detect traffic anomalies. Such distribution must be automatic and transparent. The distributed processing, according to (STONEBRAKER et al., 2005), represents one of the most important requirements to process high volumes of streaming data in real-time.

6.3. Estimating the severity of traffic anomalies

When traffic anomalies occur, besides detecting them, another important task that provides useful information for traffic managers and city authorities is to

determine their degree of severity. In this section, we address this problem and propose a method to classify the severity of a traffic anomaly detected in a road segment *S*, during a particular time period *P*, on a specific day *d*, at a certain time interval *T*.

The method is based on an analysis of anomalous delayed trips that correspond to a detected traffic anomaly. It has two main steps: (1) Individual classification of those anomalous delayed trips, according to their severity degree; and (2) Estimation of the severity of the detected traffic anomaly as a whole (i.e. as a set of multiples anomalous delayed trips) for a specific interval.

Individual classification of anomalous delayed trips, according to their severity degree.

A traffic anomaly is represented by a set of chronologically consecutive anomalous delayed trips belonging to *S*, during *P*, on *d*, at*T*. Rather than classifying trips into one of two categories – normal or anomalous – it is possible to evaluate the severity of each anomalous delayed trip, based on its travel time. For such purpose, the concept of control limits used in statistical quality control can be extended to provide an assessment of anomalous trips across a range of many regions, rather than a binary interpretation. In SQC, this is analogous to multiple tolerance regions, all centered on the point representing the mean of a sample.

To define these tolerance regions, we use several confidence levels (i.e. 90%, 95%, 99% and 99,7%), introduced in (TUROCHY; SMITH, 2000), that associate discrete values of normality degree to traffic monitoring variables. On this basis, we have stated the control limits for the travel time of buses(τ), which is our monitoring variable. The control limits for $\tau$are shown inFigure 15.

In Table 7, each tolerance region is made to correspond to its respective level of normality and a score between 1 and 4.In accordance, the travel time of trips considered as normal belongs to the green tolerance region inFigure 15,whereasthe travel time of slightly, moderately, severely and extremely anomalous trips corresponds to the tolerance regions colored in beige, yellow, orange and red respectively, which are located on both sides of$\bar{\tau}$.

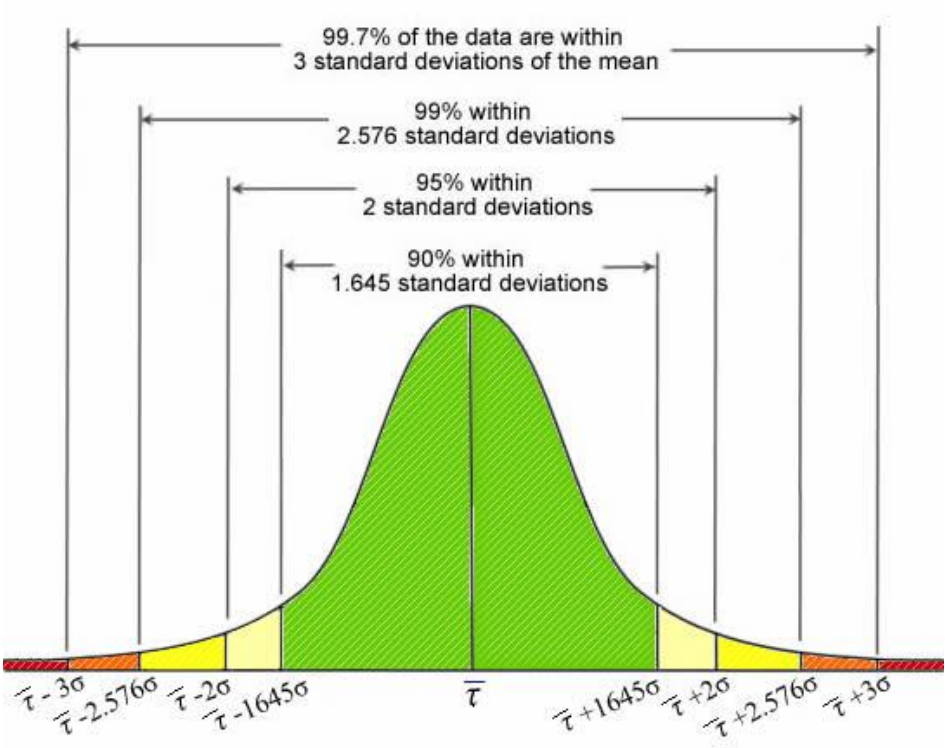


Figure 15: Control limits delimiting tolerance regions for the travel time.

Overall, we apply these control limits, to individually classify any bus trip according to their degree of normality. The classification in normal and anomalous was already discussed in Section 6.2.1. Therefore, as our focus in this section is to classify the anomalous delayed trips, we only use the control limits that allow to separate them according to their severity, specifically, in slightly, moderately, severely and extremely anomalous.

Since the travel time of the anomalous delayed trips exceeds the value of $\bar{\tau} + 1{,}645\sigma$, then, they strictly belong to one of the beige, orange or red regions located in the right tail of the Gaussian function represented in Figure 15.The greater the difference in between the values of the travel time of a bus trip and μ, the greater would be the level of the anomaly of the trip.

Table 7: Degrees of normality of bus trips according to the value of τ.

Tolerance Region	**Level of Normality**	**Score**

Probability Distribution (τ)	Value of Travel Time (τ)		
$\tau \leq \tau$ (90%)	$\bar{\tau} - 1{,}645\sigma \leq \tau \leq \bar{\tau} + 1{,}645\sigma$	Normal	0
τ (90%) $< \tau \leq \tau$ (95%)	$\bar{\tau} + 1{,}645\sigma < \tau \leq \bar{\tau} + 2\sigma$	Slightly anomalous	1
	$\bar{\tau} - 2\sigma \leq \tau < \bar{\tau} - 1{,}645\sigma$		
τ (95%) $< \tau \leq \tau$ (99%)	$\bar{\tau} + 2\sigma < \tau \leq \bar{\tau} + 2{,}576\sigma$	Moderately anomalous	2
	$\bar{\tau} - 2{,}576\sigma \leq \tau < \bar{\tau} - 2\sigma$		
τ (99%) $< \tau \leq \tau$ (99,7%)	$\bar{\tau} + 2{,}576\sigma < \tau \leq \bar{\tau} + 3\sigma$	Severely anomalous	3
	$\bar{\tau} - 3\sigma \leq \tau < \bar{\tau} - 2{,}576\sigma$		
$\tau > \tau$ (99,7%)	$\tau > \bar{\tau} + 3\sigma$	Extremely anomalous	4
	$\tau > \bar{\tau} - 3\sigma$		

Estimation of the severity of a traffic anomaly for a specific interval

We emphasize that even when a traffic anomaly extends for more than one consecutive time interval (e.g. from 9:00 AM to 10:00 AM, and from 10:00 AM to 11:00 AM), we separately estimate its severity for each interval, and not as a whole. This is justified since each interval *T* has its own control limits to define the levels of normality. Therefore, a travel time value that is considered severe in the interval *T* may be considered extreme in the interval *T+1*, or vice versa.

To estimate the severity of a traffic anomaly detected in a particular road segment *S*, during a particular time period *P*, on a specific day *d*, at a certain time interval *T*, we compute the average of the score assigned to each one of its anomalous delayed trips. Finally, according to this resulting score, the traffic anomaly is classified using the corresponding level of normality from the Table 8.

6.4. Impact of traffic anomalies

Determine the traffic anomaly duration, the travel time delay, the traffic anomaly spread and the number of people affected by traffic anomalies also represent key tasks for evaluating the impact of traffic incidents. In this section, we address the delimitation of the duration of a traffic anomaly and the estimation of the delay caused in the travel time.

6.4.1. Delimitation of traffic anomaly duration

The *delimitation of traffic anomaly duration problem* is defined as follows: "Given anomalous delayed trip sets δ_k, for k=1, 2, ..., n, associated with a traffic anomaly, which occurred on a road segment *S* during a time period *P* and extended across the intervals k, such that $\alpha_k[\ \delta_k$, S, P,T_k , $\bar{\tau}_k$, UCL_k]was detected using average travel time $\bar{\tau}_k$ and upper control limit UCL_k, determine the time when the traffic anomaly started and the time when it ended".

A *delimitation of traffic anomaly duration* strategy would go as follows:

Note that the anomalous delayed trips in$\delta_k$are chronologically ordered. Let γ_k be an anomalous delayed trip in $\delta_k$and let γ_{k_1}be the first anomalous delayed trip in δ_k. Let γ_{k_m}be the last anomalous delayed trip in δ_k.

1. Recover the start time of γ_{1_1} and recover the end time of γ_{n_m}.
2. Compute the difference between them to obtain the duration of the anomaly.

6.4.2. Estimation of travel time delays

The *travel time delay estimation problem* is defined as follows: "Given anomalous delayed trip sets δ_k, for k=1, 2, ..., n, associated with a traffic anomaly that extended across the intervals k, such that $\alpha_k[\ \delta_k$, S, P,T_k, $\bar{\tau}_k$, UCL_k], estimate the travel time delay that the anomaly caused for each interval".

A quite simple travel time delay estimation strategy would go as follows:

We recall that let γ_k be an anomalous delayed trip in δ_k .

1. Recover the travel time of each γ_k and compute $\bar{\tau}_{\delta_k}[S, \delta_k, T_k, P]$ that represents the average of the travel time of anomalous delayed trips δ_k, that traversed S, during the period P, at interval T_k .
2. Compare $\bar{\tau}_k[S, \beta_k, T_k, P]$and $\bar{\tau}_{\delta_k}[S, \delta_k, T_k, P]$. Then, the difference between them represents the travel time delay that the anomaly caused in the interval T_k .

Note that the travel time delay that an anomaly caused is analyzed for each interval separately and not for the anomaly as a whole because each interval has its own values $\bar{\tau}$.Chapter 7 provides examples of travel time delay estimations.

Finally, using travel time delay estimations, it would also be possible to estimate the number of bus passengers affected, or the total loss of time (incurred by bus passengers), if bus passenger data were available.

6.5. Conclusions

By analyzing the behavior of travel time of buses that serve the road network of urban areas, it is possible to detect traffic anomalies, understand their characteristics, and evaluate their impact. This chapter proposed two methods for traffic anomaly detection, for non-real-time and for real-time. Both are based on Statistical Quality Control and use the travel time of bus trips as the data source. Also, in order to extract more detailed insights from the bus data, in addition to making a binary interpretation of whether an anomaly occurred or not, a method to classify the anomalies according to their severity was proposed. Finally, to evaluate the impact of traffic anomalies, a strategy that includes the estimation of traffic anomaly duration and the delays caused in travel time was presented.

Detecting and understanding the characteristics of traffic anomalies, as well as estimating their impacts in terms of incident duration and travel time delays, will help traffic decision-makers to react, as soon as possible, to the abrupt changes in traffic conditions, and to select better operational strategies. Thereby, the negative impact of traffic anomalies on the emotional, physical and economic well-being of citizens can be reduced.

7
Experiments

7.1.Introduction

In this chapter, we apply and evaluate the proposed approach over data collected in the real bus network of the City of Rio de Janeiro, Brazil since June 12th, 2014 until February 28th, 2017. For such purposes, we conducted experiments to test the functionalities of the prototype that supports the proposal described in previous chapters and discuss the results.

The experiments cover the following scenarios: Section 7.2 presents an analysis of the bus network of the City of Rio de Janeiro, Brazil; Section 7.3 provides some examples of travel time patterns computed for the monitored paths of the bus network; Section 7.4 applies a normality test over the bus travel times that correspond to the patterns to validate the assumption used in algorithm 7 of Section 6.2.1; Section 7.5 shows how a set of traffic anomalies that affected the traffic conditions of the city were detected, as well as an estimation of their duration and the delays caused in the travel time of buses; Section 7.6 evaluates how bus travel time patterns in the city were affected by a traffic change implemented mostly for the Rio 2016.

All algorithms of the prototype were implemented in Python 3.7 and were ran on a computer with 3.3GHzof Intel Core i7-5820K processor and 64GB of memory, with Ubuntu 14.04 operating system.

7.2.
Analyzing the bus network of the City of Rio de Janeiro, Brazil

The public transportation system of the City of Rio de Janeiro is largely based on buses. The statistics published in the mobility transparency portal of the City Hall (Prefeitura da Cidade do Rio de Janeiro 2015, 2016, 2017) and in (Dal Piva & Estarque 2017) reveals that buses accounted for nearly 60% of all passengers

transported over the past three years. Precisely the complexity of the bus network of Rio de Janeiro is one of the main reasons that led us to select it to evaluate our proposal.

To analyze this bus network, as it mentioned above in Chapter 3, the prototype requires as input three types of data. The first one is related to the topological structure of the road network of the city obtained from the Open Street Map, the second one refers to the routes that buses operating in this city should follow, and the third one is the historical trajectory dataset generated by GPS devices installed in buses.

The road network of the City of Rio de Janeiro is delimited by the bounding box NE (-20.76347, -40.953869), SW (-23.37085, -44.888519) and the data about it topological structure were extracted from the Open Street Map, available at http://www.openstreetmap.org/#map=11/-22.9404/-43.3727&layers=T.

The General Transit Feed Specification files containing the bus routes that serve the city were retrieved from the open data portal of urban mobility of Rio de Janeiro, available at http://data.rio/dataset/pontos-dos-percursos-de-onibus, which is provided by the City Hall.

Data from GPS of buses that operate in the city are continuously captured, in about every 1min30sec, by the City Hall (DADOS RIO, 2015). Each entry contains a timestamp, the bus identifier, the line number, the position (as latitude and longitude) and the speed, as illustrated in Table 5. However, the public data portal of City Hall only provides the instantaneous data, i.e. no historical data is available. For this reason, Guberfain (GUBERFAIN; CÔRTES VIEIRA, 2015) developed a service that queries these data periodically and stores the entries as compressed text files at the URL http://www.busesinrio.com/files.php for future urban studies. This historical dataset currently contains more than 3 billion samples in CSV format, captured since June 12th, 2014 until today. To support the experiments, we use the samples collected from June 12th, 2014 until February 28th, 2017, which represent almost 3 years of bus trajectory data.

As a result of an analysis of the main structural (topological features) and operational (bus routes) variations that took place in the road network of the City of Rio de Janeiro, from June 12th, 2014 to February 28th, 2017, we identified two

significant traffic changes which may affect the performance of the bus network. The traffic changes we identified were: the itinerary change of 180 bus lines, which serve the city, from May 21st, 2016 (AGÊNCIA O DIA, 2016) and the construction of the New Joá Elevated Road and the new roads to access it, which were jointly inaugurated on May 28th, 2016(G1 GLOBO, 2016).An example of how those changes impacted on the performance of the bus network of the city is provided in Section 7.5.

Based on the referred changes, two bus network versions, B_1 and B_2, were defined, whose validity periods cover from June 12th, 2014 to May 20th, 2016 and from May 21st, 2016 to February 28th, 2017, respectively. Some statistics of both bus network versions are shown in Table 8.

Table 8: Statistics of bus network versions $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$.

Statistics	**Bus Network Version $\boldsymbol{B_1}$**	**Bus Network Version $\boldsymbol{B_2}$**
Duration Time (Months)	23	9 (current version)
Bus lines	716	441
Number of buses	8,916	8,342
Average no. number of round trips per month	1.54 million	1.2 million
Average no. number of passengers transported per month	105 million	84 million
Average no. of kilometers travelled per month	63.3 million	55.6 million
Number of companies	44	42
Number of employees	41,375	38,229
Average bus age	4.06 years	4.42 years
Average no. of passengers per kilometer	1.65	1.51
Average no. of kilometers travelled per	7,099	6,665

bus per month		

For each bus network version, the respective monitored bus network was computed using algorithm 2, specified in Section 4.3. After that, each monitored network was segmented into monitored paths, using algorithm 3, described in Section 4.4. A sequence of how a bus network version is being processed by applying these algorithms can be appreciated in Figure 16, 17, and 18for the case of B_1. Specifically,Figure 16showsthe bus network version B_1.Its corresponding monitored network is represented in Figure 17, and a sample of some of its monitored paths is depicted in Figure 18. These monitored paths, which are identified in the figure in blue, green and red, correspond to the Zuzu Angel Tunnel, the Jardim Botânico Street, and the Bartolomeu Mitre Avenue, respectively, and represent important transport arteries of the city. They approximately have a distance of 3.2 km, 3 km, and 850 m respectively. It is important to take into account the direction of the monitored paths for the analysis.

Once a monitored bus network is segmented into monitored paths, it is possible to carry out other analysis, such as finding the travel time patterns on each path and detecting traffic anomalies. Examples of these kinds of analysis are described in Sections 7.3 and 7.4 respectively.

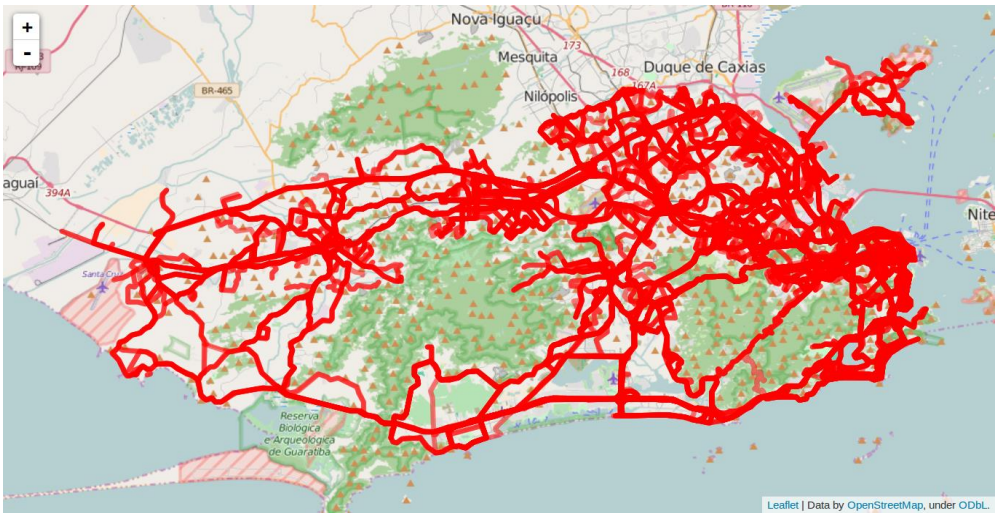
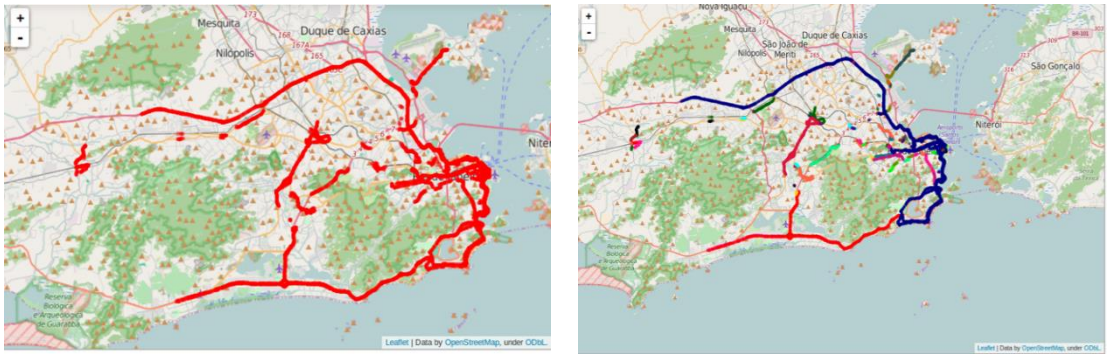


Figure 16: Bus network version of the City of Rio de Janeiro during the period from June 12th, 2014 - May 20th, 2016 ($\boldsymbol{B_1}$).



(a) Monitored Network as a single graph

(b) Monitored Network divided in Connected Components

Figure 17: Monitored bus network of $\boldsymbol{B_1}$.

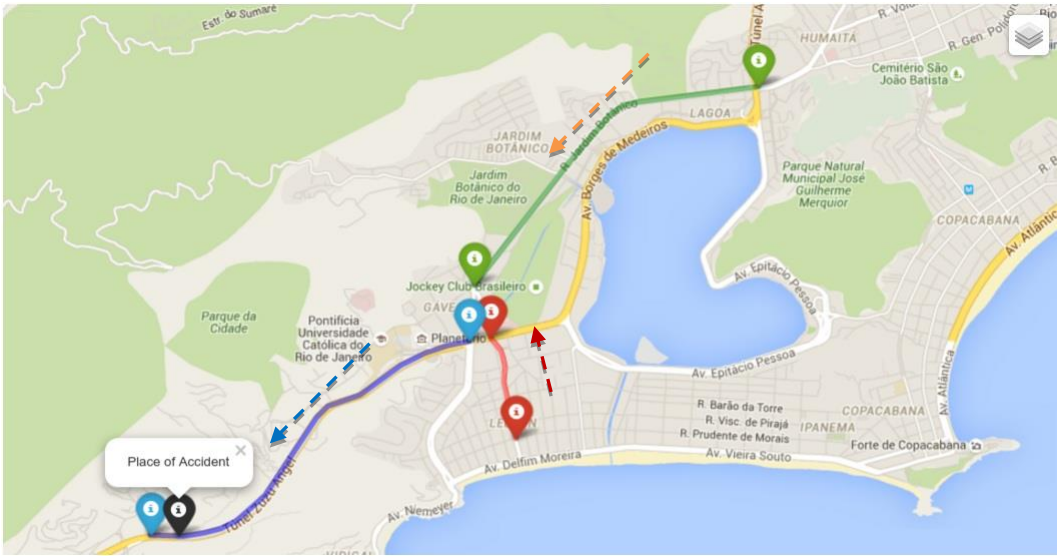


Figure 18: Example of monitored paths belong to the version $\boldsymbol{B_1}$ of the bus network of the City of Rio de Janeiro.

7.3. Travel time patterns for monitored paths

In a large city such as Rio de Janeiro, the traffic conditions during rush hours are usually worse than in other periods. Similarly, the travel times in the same segment may be different during working days and holidays, and during weekdays and weekends. Therefore, to compute the travel time patterns for the monitored paths of each bus network version of the City of Rio de Janeiro, we use the same temporal partitioning that was exemplified in Section 5.3,Figure 8. According to this partitioning, a time period P is defined as a day of the week that

belongs either to the school classes period or the school vacations, and which is either a holiday or a working day; and a time interval T represents one of the 24 fixed time intervals of 1 hour each that a day has.

To illustrate the behavior of travel time patterns in the bus network of the city, the road segments analyzed were: Zuzu Angel Tunnel, Jardim Botânico Street, and Bartolomeu Mitre Avenue, which are monitored paths of the bus network version B_1 as mentioned in the previous section.

For these monitored segments, the bus travel time behavior by hours of the day, extracted from the historical data collected within the validity period (almost 2 years) of the bus network version B_1, for days corresponding to the time period P that covers "all Mondays during a school classes period that are working days", is depicted in the boxplot graphs of Figure 19, 20, and 21.

The travel times included in each box correspond to the set of bus trips resulting from the travel time pattern refinement process, which was explained in detail in Section 6.2.1. Hence, the trips that are part of a traffic anomaly or those that denote outliers are not included in this representation. The dashed line represents the travel time pattern (sample mean).

Figures 19 and 21indicate that on Mondays during a school classes period that are working days, travel times to traverse the Zuzu Angel Tunnel and Bartolomeu Mitre Avenue in the rush hours (5:00 PM – 9:00 PM) are higher than in other hours. Specifically, for the Zuzu Angel Tunnel, it was observed a travel time peak from 5:00 PM to 9:00 PM, when the buses took up to 17 minutes, on the average, to traverse it. However, in times of less traffic flow, it is possible that a bus crosses the segment in about 5 minutes (e.g. from 11:00 PM to 12:00 AM). At the Bartolomeu Mitre Avenue, under normal traffic conditions, the maximum value of travel time is about 6 minutes, on the average, reached at the interval from 5:00 PM to 9:00 PM; and the minimum value perceived is about 3 minutes, on the average (e.g. from 6:00 AM to 7:00 AM).

It can be concluded that buses running in these segments, during rush hours exhibit a similar behavior and those running in other (nonpeak) hours are likely to be similar; although the travel time in the monitored segment of Bartolomeu Mitre

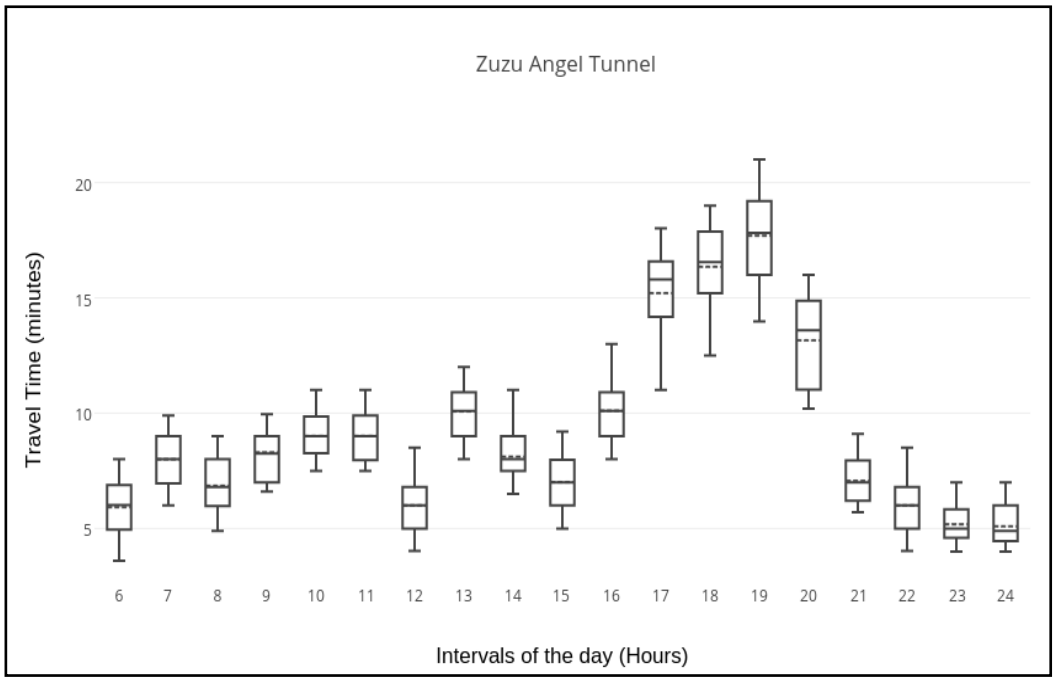


Figure 19: Travel Time of buses by hours on Mondays during a school classes period that are working days - Zuzu Angel Tunnel.

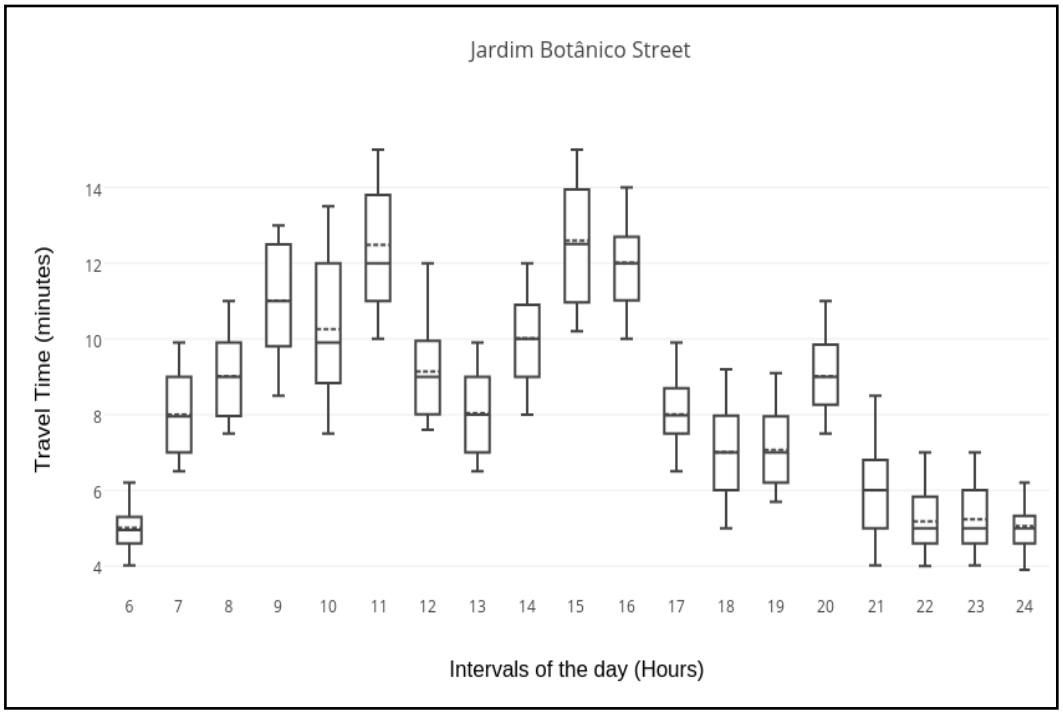


Figure 20: Travel Time of buses by hours on Mondays during a school classes period that are working days - Jardim Botânico Street.

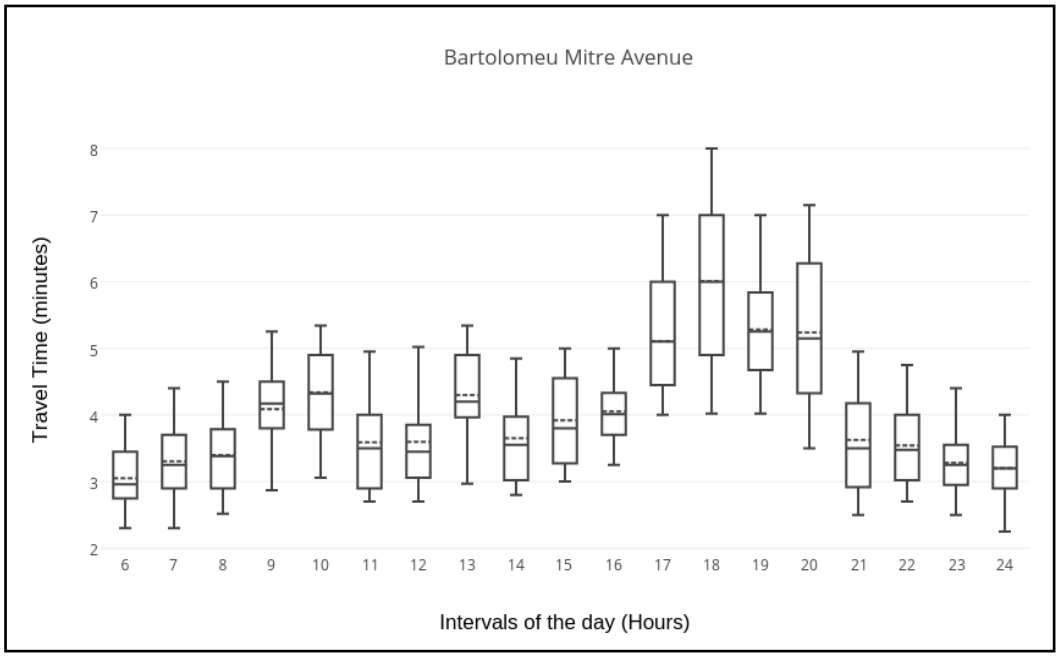


Figure 21: Travel Time of buses by hours on Mondays during a school classes period that are working days - Bartolomeu Mitre Avenue.

Avenue is always lower than in the Zuzu Angel Tunnel because the distance of the former is also considerably smaller.

The relatively similar travel time behavior during all-day of these two monitored segments, revealed in Figures 19 and 21,is coherent since these segments belong to contiguous streets such that the traffic flow that enters the Zuzu Angel Tunnel mainly comes from two streets, and one of them is the Bartolomeu Mitre Avenue.

In the case of Jardim Botânico Street, as illustrated in Figure 20, it exhibits a different travel time pattern from the other two monitored paths previously analyzed. In the time period under analysis, to cross this segment, buses take more time at the intervals from 7:00 AM to 12:00 AM and from 2:00 PM to 5:00 PM than in other hours. At these peak hours, under normal traffic conditions, the travel time of buses reaches values of up to 16 minutes, whereas at not peak hours the segment can be traversed in as little as 5 minutes, on the average.

To conclude, this example illustrates the ability of the prototype to mine a trajectory dataset to uncover typical patterns for selected monitored paths, time periods and intervals.

7.4. Normality test for the bus travel time that correspond to the pattern

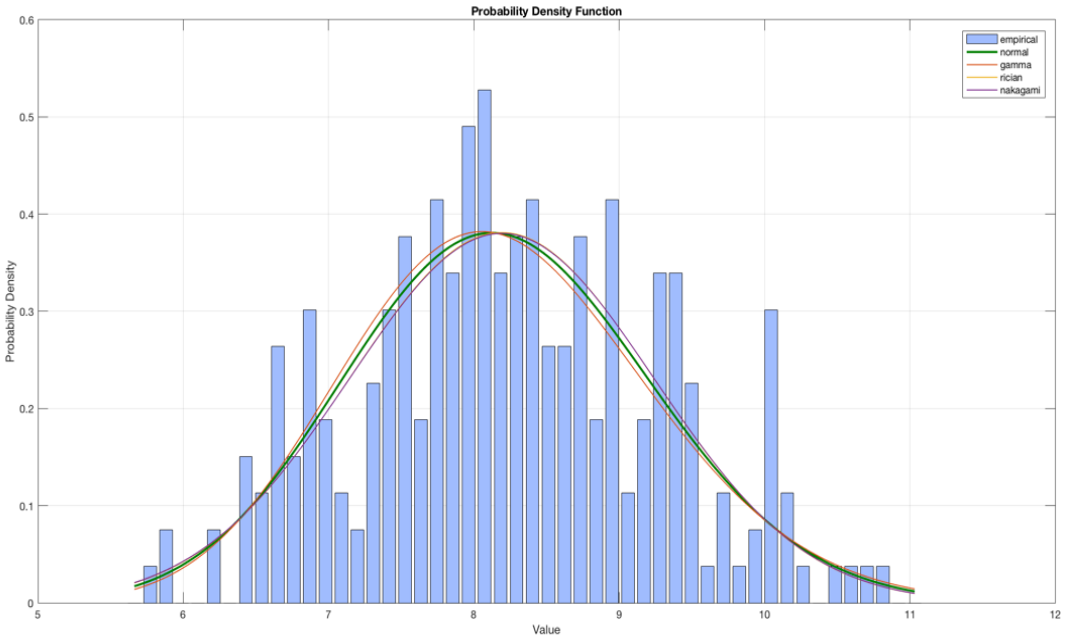
As mentioned in Chapter 6, the algorithms to detect traffic anomalies we propose presuppose that the travel time of buses, in normal traffic conditions, for each temporal partition, follows a Gaussian distribution. In order to validate this assumption, for a given monitored path, we performed normality tests over the travel time taken by buses to traverse it, at a given time period and a given time interval in normal traffic conditions (i.e. without traffic anomalies). Therefore, the normality test was applied over the set of bus travel times that correspond to the pattern.

The normality tests used Matlab[14], with two different functions. The first one $h = jbtest(\mathrm{x})$, called Jarque-Bera test(JARQUE; BERA, 1980), returns a test decision for the null hypothesis, which indicates that data in vector x come from a normal distribution with an unknown mean and variance. The alternative hypothesis indicates that data do not come from such a distribution. The result of h is 1, if the test rejects the null hypothesis at the 5% significance level, and 0 otherwise.

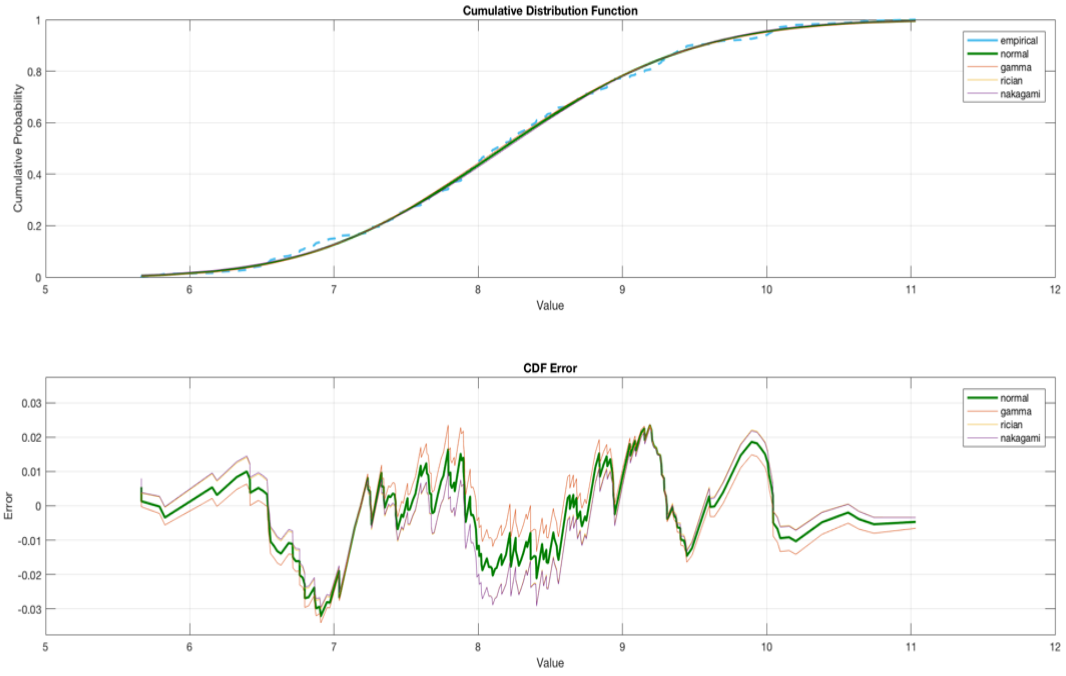
The second Matlab function, $allfitdist(data, sortby)$, was used to fit the data to the best probability distribution. The result of this fit could be sorted by four different criteria: (i) NLogL - Negative of the log likelihood, (ii) BIC - Bayesian information criterion (default), (iii) AIC - Akaike information criterion, and (iv) AICc - AIC with a correction for finite sample sizes. In this case, we used Bayesian information criterion.

The normality tests were applied to multiple travel time datasets that correspond to different paths of the monitored bus network and to different periods and intervals. As a result of the application of the Jarque-Bera test, it was obtained that 85% of cases accepted the null hypothesis at the 5 % significance level. It means that the assumption about the normal nature of the travel time is correct.

[14] https://www.mathworks.com/

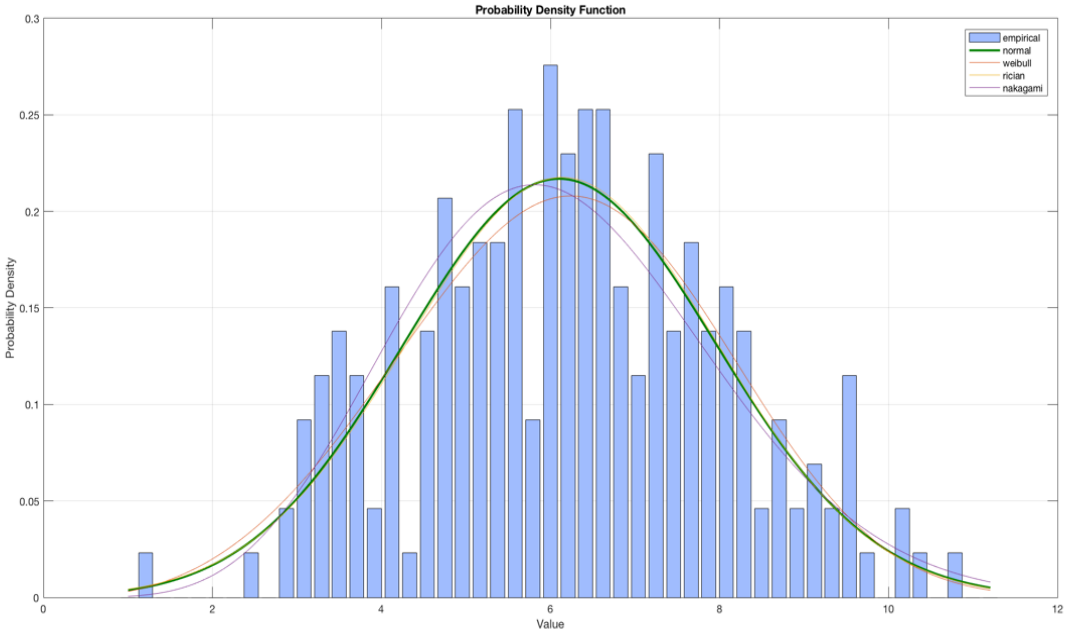


(a) Probability Density Function that fits the data.

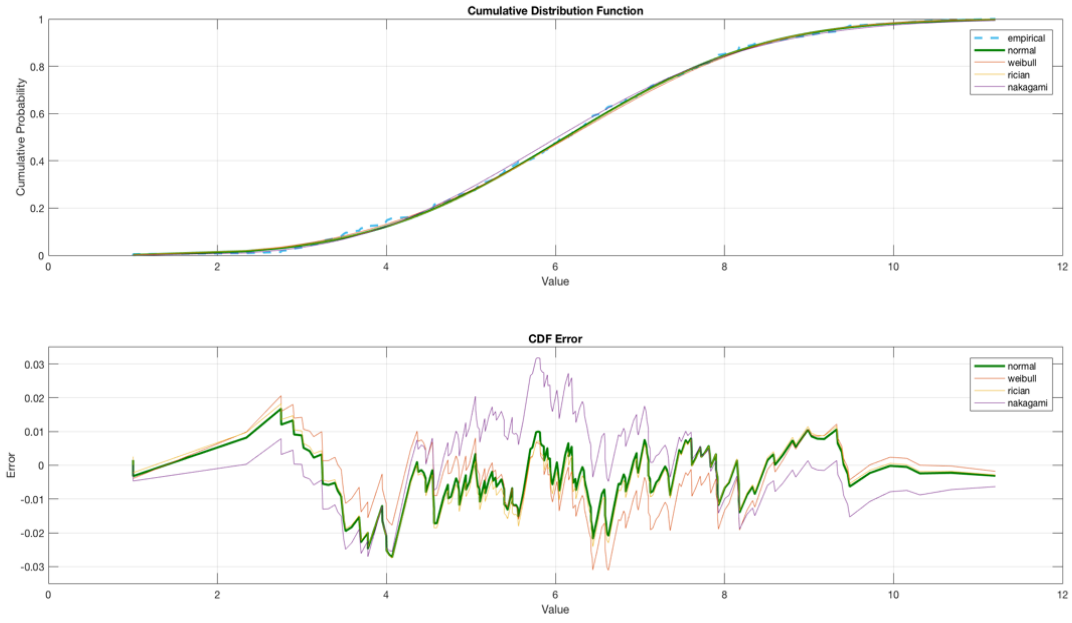


(b) Cumulative Density Function and error of applying the normality test for the data.

Figure 22: Normality test for data corresponding to travel time travel time data of bus trips that traversed Zuzu Angel from 8:00 AM to 9:00 AM on -Mondays during a school classes that are working days- and belong to $\boldsymbol{B_1}$.

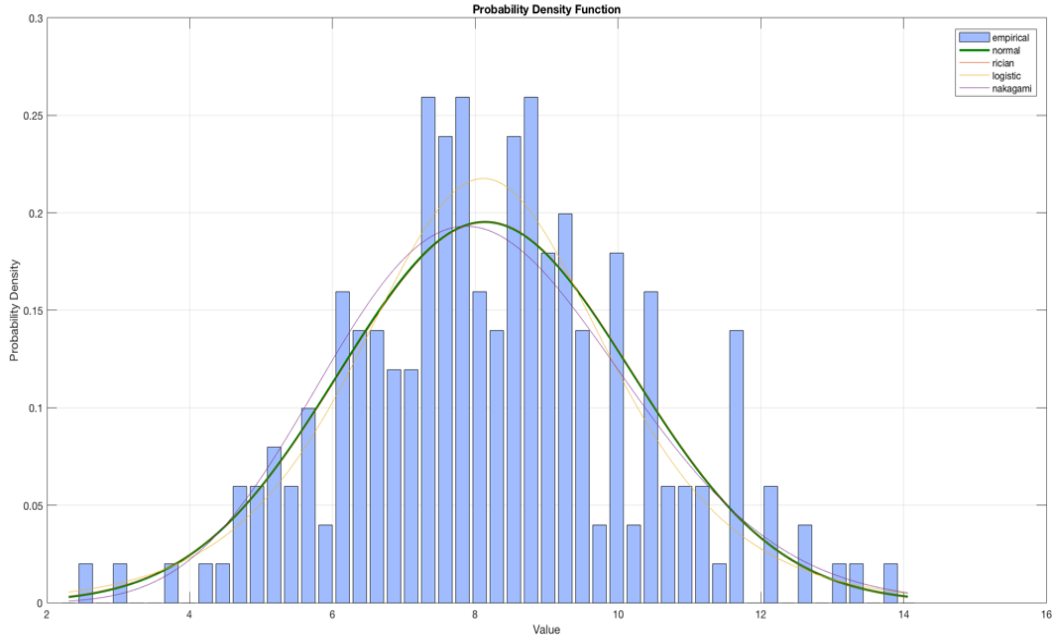


(a) Probability Density Function that fits the data.

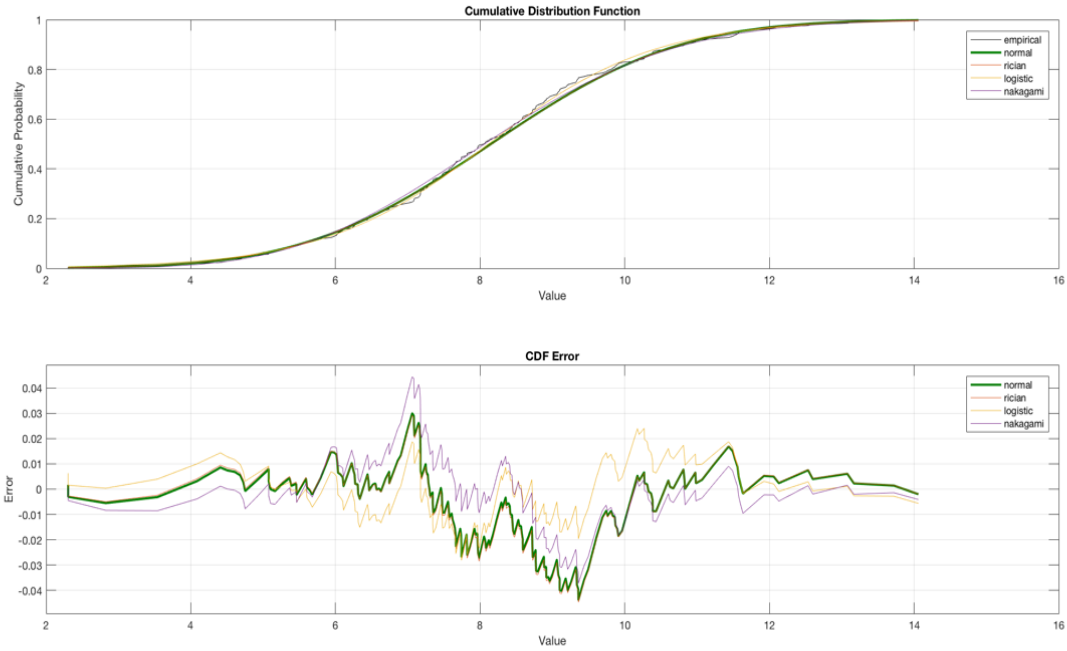


(b) Cumulative Density Function and error of applying the normality test for the data.

Figure 23: Normality test for data corresponding to travel time data of bus trips that traversed Zuzu Angel from 8:00 AM to 9:00 AM on -Mondays during a school classes that are working days- and belong to $\boldsymbol{B_1}$.



(a) Probability Density Function that fits the data.



(b) Cumulative Density Function and error of applying the normality test for the data.

Figure 24: Normality test for data corresponding to travel time of bus trips that traversed Zuzu Angel from 9:00AM to 10:00 AM on -Mondays during a school classes period that are working days- and belong to $\boldsymbol{B_1}$.

To corroborate the results, the second function (*allfitdist*) was executed. As output, we obtained the probability density function (PDF), the cumulative density function (CDF) and the respective error for the sets of travel time data corresponding to the Zuzu Angel Tunnel, in the period belonging to - Mondays during a school classes period that are working days - of B_1 at the intervals 7:00 AM - 8:00 AM, 8:00 AM - 9:00 AM, and 9:00 AM - 10:00 AM, as shown in Figure22, 23, and 24, respectively.

As can be seen in the figures, the distribution that best fits the datasets of the experiments is the normal distribution. Then, as the phenomenon under study is the same in all of the paths of the monitored network, we may suppose that the travel time pattern under normal conditions follows a normal distribution.

7.5. Examples of Traffic Anomalies Detected

After applying Algorithm 7, explained in Section 6.2.1,to the travel time data of buses that operated in the monitored paths belonging to the bus network versions$B_1$and$B_2$to detect traffic anomalies, we identified 279 traffic anomalies that happened during the validity period of B_1, and 98 that happened during the validity period of B_2. Many of the detected anomalies could be associated with real events that affected the traffic in the city, which were published in newspapers, news channels, tweets distributed by the City Hall (e.g. @OperacoesRio, @TransitoRioRJ, @CETRIO_ONLINE), and other mass media. Some of these events, which occurred on different days of the week, are presented below:

1. On Sunday, March 1st, 2015 the ceremony for the 450th anniversary of the City of Rio de Janeiro, held at the Palácio da Cidade in Botafogo (G1 RIO, 2015a). This event caused significant travel time delays in traversing the monitored paths corresponding to the São Clemente and Humaitá Streets, from 4:00 PM to 7:00 PM approximately.
2. On Monday, August 17 th, 2015, a fatal collision, that caused the death of a motorcyclist, took place at the exit of the Zuzu Angel Tunnel in the direction to the west zone, near the Rocinha community in the São

Conrado area (seeFigure 18) (G1 RIO, 2015b). The referred tunnel is part of an expressway that connects the south and the west zones of Rio. The accident caused a considerable delay in the travel time of buses to traverse the monitored paths belonging to Zuzu Angel Tunnel, Jardim Botânico Street and Bartolomeu Mitre Avenue, from 7:00 AM to 12:00 AM approximately.

3. On Saturday, September 5th, 2015, the Book Bienal took place at the Riocentro Convention Center, in the west zone of Rio (G1 RIO, 2015c). This event considerably increased the travel time of buses to traversed the monitored path corresponding to Americas Avenue, in the west direction, and Ayrton Senna Avenue, from 3:00 PM to 4:00 PM.

4. On Friday, April 1st, 2016, a taxi drivers strike, protesting against the UberCompany, blocked two ways of the Francisco Bicalho Avenue and the Aterro do Flamengo Avenue(RESENDE; PAULA, DE, 2016). This incident caused a disorder in the travel time of monitored paths belonging to Francisco Bicalho Avenue, Brazil Avenue (one of the most important expressways in Rio de Janeiro) towards downtown, the Galeão Highway, and the Gasometer Viaduct in the early hours of the morning, specifically, from 5:00 AM to 8:00 AM approximately.

5. On Thursday, July 7th, 2016, on the occasion of a delay of two hours for the removal of obstacles of constructive works for the bus rapid transit (BRT), being executed in the Brazil Avenue between the neighborhoods of Cajú and Bonsucesso, the central runway of this avenue towards the west zone was blocked, provoking a large traffic congestion(G1 RIO, 2016a). As a consequence of this incident, the travel time of buses to traverse the monitored paths belonging to the Brazil Avenue, in direction of the west zone, and the Gasometer Elevated Road, in direction to Brazil Avenue, exceeded the average travel time, from 5:00 AM to 7:00 AM.

6. On Wednesday, August 3rd, 2016, the blockading of several downtown major streets (Rio Branco, Almirante Barroso, Presidente Antônio Carlos, Primeiro de Março, Visconde de Inhaúma, Acre, Sacadura Cabral, Livramento, and Rivadavia Correa) due to the passage of the Olympic torch caused traffic congestion in nearby streets. As detected, the buses

took much longer than normal travel time to cross the monitored paths corresponding to República do Chile, República do Paraguai and Presidente Vargas Avenues, from 8:00 AM to 12:00 PM(G1 RIO, 2016b; SEGOV, 2016).

7. On Friday, November 4th, 2016, between dawn and the early morning (7:30 AM) nine accidents were recorded in the Brazil Avenue. The accident that had the worst impacted on the transit was a truck going toward the west zone that capsized and scattered all cargo along the runway(G1 RIO, 2016c). The accidents caused a substantial increment in the travel time of buses to cross monitored paths (one in the direction to the west zone and the other to downtown) belonging to Brazil Avenue from 6:00 AM to 9:00 AM.

To illustrate, step by step, how the traffic anomaly detection strategy works with a concrete example, we analyze the trajectory data associated with the second of the traffic anomalies listed above, which refers to the accident that involved a motorcyclist at the exit of the Zuzu Angel Tunnel at, approximately, latitude -22.992342 and longitude -43.249278, in the São Conrado area, on Monday, August 17th, 2015.

As mentioned above, this accident strongly affected the monitored paths depicted inFigure 18 (Zuzu Angel Tunnel, Jardim Botânico Street, and Bartolomeu Mitre Avenue), which have been intentionally used in previous sections to achieve a better understanding. A detailed explanation is provided just for the Zuzu Angel Tunnel, because the processing is the same for all paths.

As the date indicates, the accident occurred during the validity period of the bus network version B_1; and according to the temporal partitioning used for the experiments, it is consistent with the time period P that covers all Mondays during a school classes period that are working days.

Then, for the current analysis, we used the travel time of all bus trips made at the Zuzu Angel Tunnel on - every Monday during a school classes period that are working days - between the dates of June 12th, 2014 and May 20th, 2016, which is the validity period of B_1.

These trips were separated in 24 subsets, such that each subset included the

travel time of all bus trips that arrive at the segment during a one-hour interval (e.g. 8:00 AM – 9:00 AM). Over each subset, the Statistical Quality Control technique was applied. As a result, we observed that, in some of them, specifically, those corresponding to the intervals 7:00 AM – 8:00AM, 8:00 AM – 9:00 AM, and 9:00 AM – 10:00 AM, a set of consecutive bus trips exceeded the Upper Control Limit for travel time defined for each interval as illustrated in Figure25a, 26a, and 27a respectively. It means that a traffic anomaly occurred.

Obviously, the trips that are part of the traffic anomaly introduce a noise in the mean of the sample (travel time pattern). For this reason, we remove them together with other trips that can be also part of other anomalies, and the outliers from the sample to recalculate the pattern, as was explained in Algorithm 7 of the Section 6.2.1. As a result, for each subset of analysis, we obtain a clean pattern, as shown in Figure 25b, 26b, and 27b.

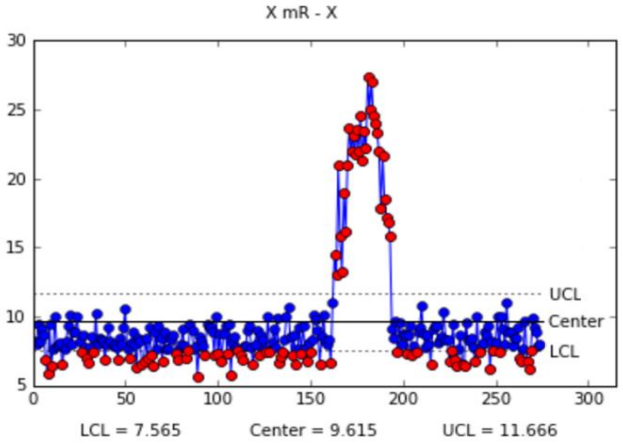
Against the clean pattern, the trips belonging to the anomaly are analyzed again, in order to really determine when the traffic anomaly started and ended within the interval. This is because, as we mention above, the anomaly introduces a noise in the mean of the sample, and some anomaly trips may have been overlapped within the noisy pattern, as observed in Figure 25c, 26c, and 27c.

However, since this analysis is individually made for each interval, to estimate the duration of the traffic anomaly as a whole for a given monitored path, the next step is to find if there are consecutive intervals affected by the same traffic anomaly as explained in Section 6.4.1. For the Zuzu Angel Tunnel, it was estimated that the anomaly started nearly of 7:00 AM and persisted until nearly 11:00 AM.

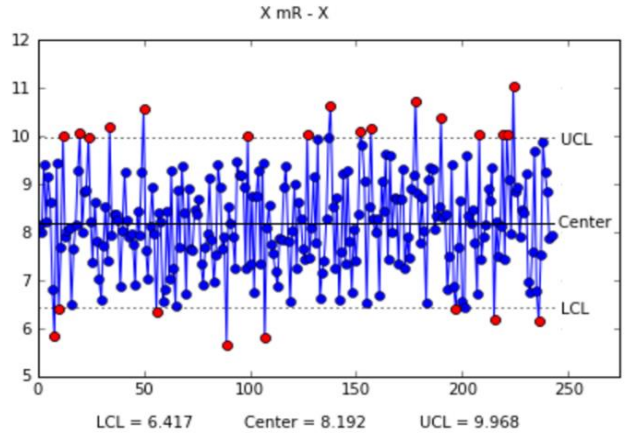
The same procedure explained so far for the monitored path Zuzu Angel Tunnel was also applied for the Jardim Botânico Street and the Bartolomeu Mitre Avenue, in which it was identified that the traffic anomaly approximately lasted from 8:00 AM to 1:00 PM and from 7:00 AM to 12:00 AM, respectively.

To evaluate the impact of this event in terms of travel time delays on the three monitored paths, we compared the travel time spent to traverse these paths on the day of the accident versus the typical travel time pattern (for the whole day).Figure 28, 29, and 30show typical patterns in green, or light gray, and

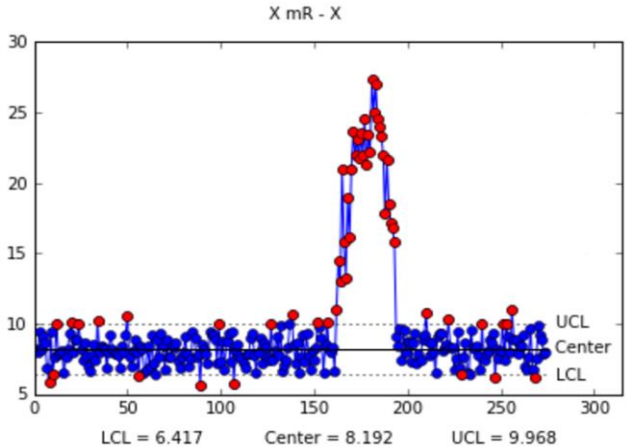
abnormal patterns in red, or dark gray.



(a) Travel time raw data containing the anomaly.

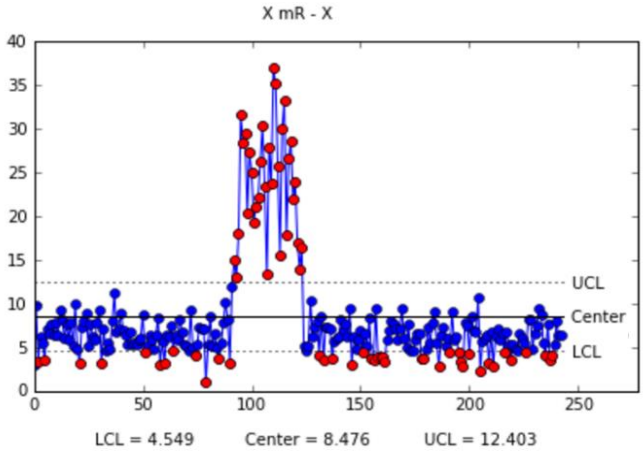


(b) Clean travel time pattern without anomaly and outliers.

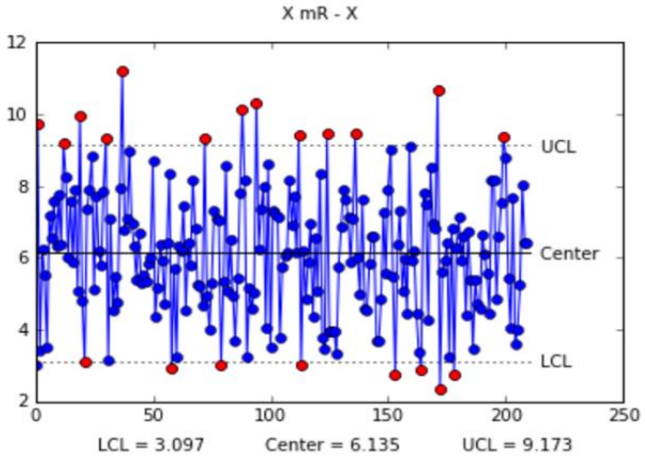


(c) Traffic Anomaly vs Clean travel time pattern.

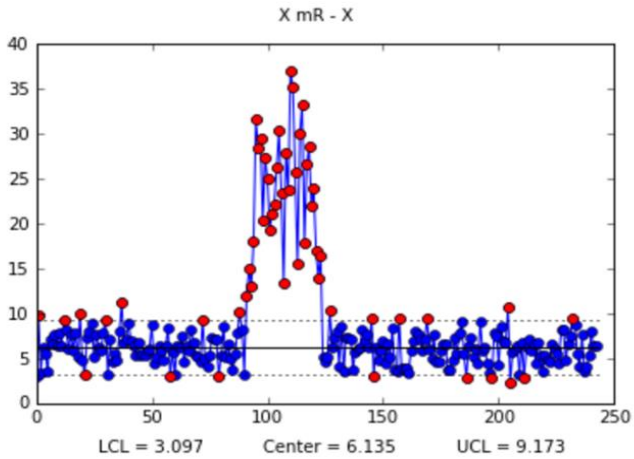
Figure 25: Control Chart for Travel time of buses on – every Monday during a school classes period that are working days-, between the dates of June 12th, 2014 and May 20th, 2016 ($\boldsymbol{B_1}$)from 7:00 AM – 8:00 AM.



(a) Travel time raw data containing the anomaly.

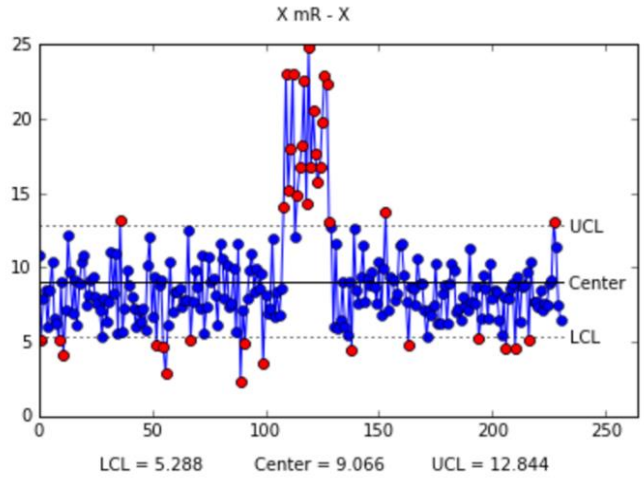


(b) Clean travel time pattern without anomaly and outliers.

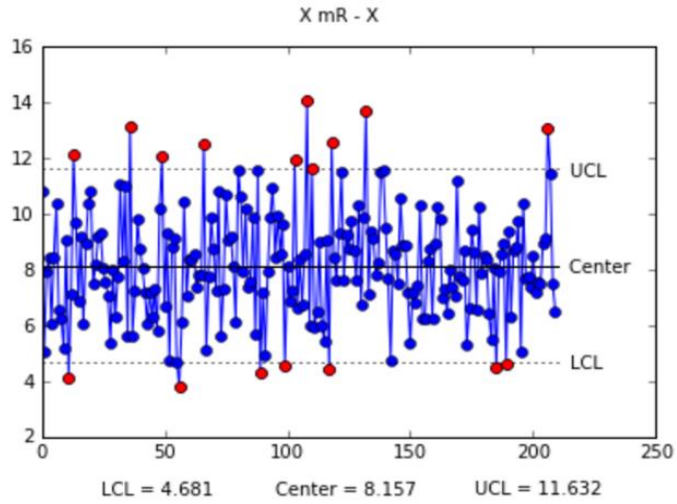


(c) Traffic Anomaly vs Clean travel time pattern.

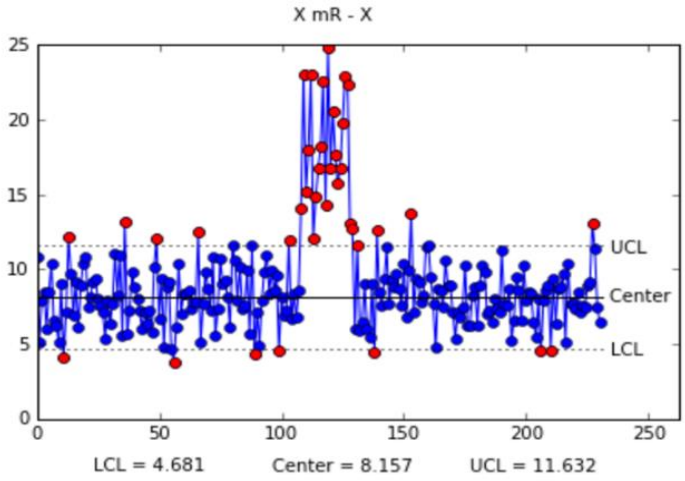
Figure 26: Control Chart for Travel time of buses on – every Monday during a school classes period that are working days-, between the dates of June 12th, 2014 and May 20th, 2016 ($\boldsymbol{B_1}$) from 8:00 AM – 9:00 AM.



(a) Travel time raw data containing the anomaly.



(b) Clean travel time pattern without anomaly and outliers.



(c) Traffic Anomaly vs Clean travel time pattern.

Figure 27: Control Chart for Travel time of buses on – every Monday during a school classes period that are working days-, between the dates of June 12th, 2014 and May 20th, 2016 ($\boldsymbol{B_1}$) from 9:00 AM – 10:00 AM.

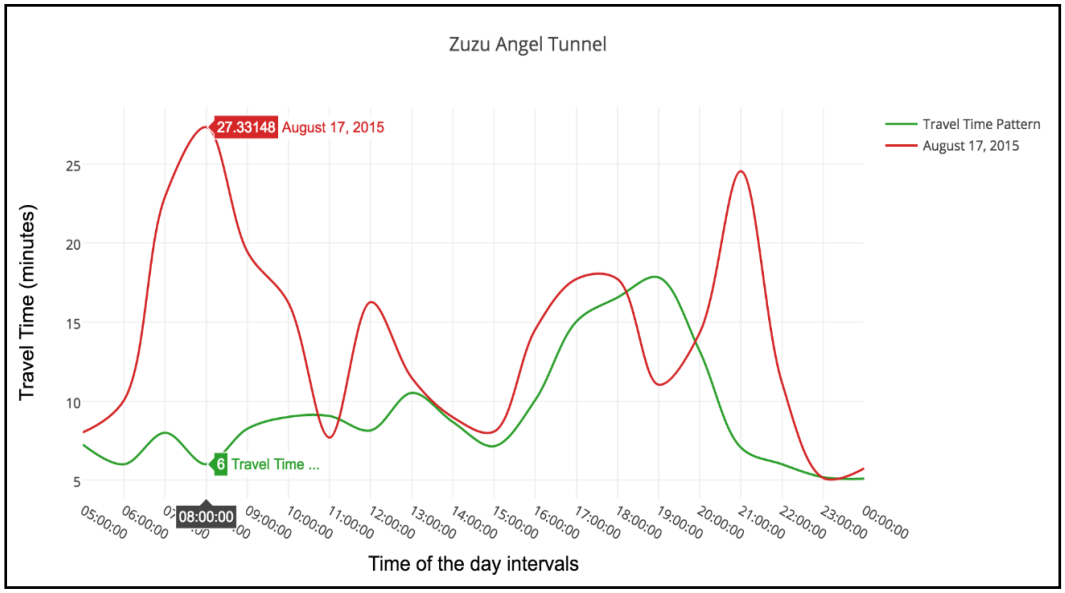


Figure 28: Travel Time Pattern vs Travel Time at the day of accident – Zuzu Angel Tunnel.

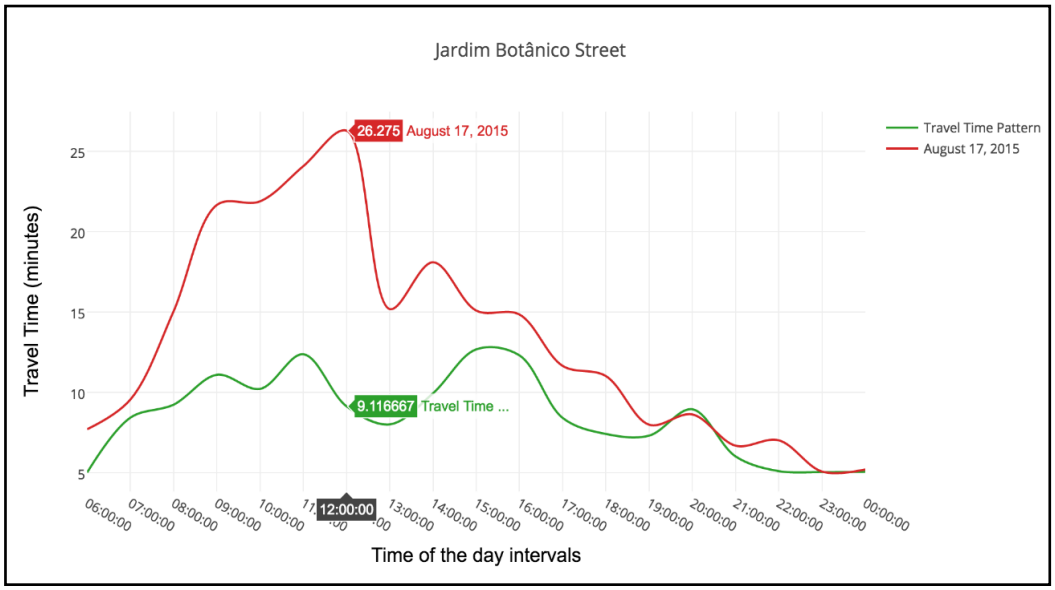


Figure 29: Travel Time Pattern vs Travel Time at the day of accident – Jardim Botânico Street.

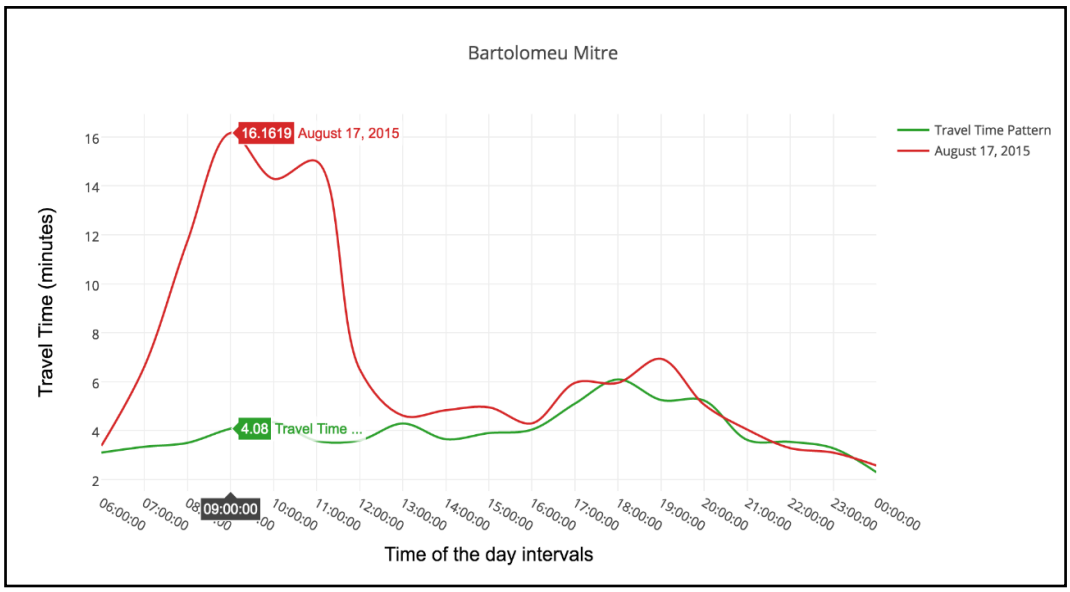


Figure 30: Travel Time Pattern vs Travel Time at the day of accident – Bartolomeu Mitre Avenue.

As the figures reveal, this event caused considerable travel time delays for a crucial period of the day. In the Zuzu Angel Tunnel, the travel time delays reached a peak of nearly 30 minutes at 8:00 AM and were observed for nearly four hours. In the Jardim Botânico Street, the travel time delays reached a peak of nearly 30 minutes at 12:00 PM and were observed for nearly five hours. In the Bartolomeu Mitre Avenue, the travel time delays reached a peak of nearly 20 minutes at 9:00 AM and were observed for nearly four hours.

Furthermore, as analyzed, travel time delays were observed throughout the Jardim Botânico Street up to the Rebouças Tunnel (indicated by the topmost dot in Figure 18), located 10 km from the accident site. This fact provides a measure of the spread that the traffic anomaly had across the road network of the city.

To conclude, this example illustrates the ability of the prototype to identify traffic anomalies for selected monitored paths and time periods, and intervals; and to compare these abnormal traffic patterns with typical patterns to assess the impact of traffic anomalies on travel time delays.

7.6. Evaluate the impact on travel time of bus network changes

As one of the main objectives of this thesis is to maintain versions of the bus network of a city that reflect significant structural and operational changes, an evaluation of the impact of such changes should also be investigated.

One of the changes that led to the creation of two versions, B_1 and B_2, of the bus network of the City of Rio de Janeiro was the construction of the New Joá Elevated Road, implemented mostly to improve the traffic flow during the Rio 2016 Olympic Games. In order to illustrate the impact of this traffic change on the bus network of the city, the experiments evaluated how bus travel time patterns were affected.

The New Joá Elevated Road has 5 km of extension and 2 lanes, whereas the Old Joá Elevated Road – still in operation – has 4 lanes. They both connect the south zone of Rio and Barra da Tijuca (a neighborhood in the west of Rio where the Olympic Games took place). We then have two scenarios, which we call *old* and *new*, defined as follows:

- Old scenario: just the Old Joá Elevated Road, with 2 traffic lanes in each direction, except during the morning traffic peak hours, when 3 lanes were used for traffic flowing from Barra da Tijuca to the south zone;
- New scenario: the Old and New Joá Elevated Roads, which in combination offer 3 traffic lanes in each direction, all day long; in each direction, one of the lanes is reserved for cars. There is no use of a reverse lane in the morning.

The bus routes connecting the south zone and Barra da Tijuca greatly benefited from this new traffic scenario. Our experiments focused on the bus traffic from Barra da Tijuca to the south zone, with emphasis on the morning peak hours.

The construction of the New Joá Elevated Road started at the end of June 2014, and the new road was inaugurated on May 28th, 2016. In our evaluation, we considered two periods: from June 12th, 2014 to May 27th, 2015; and from May 28th, 2016 to November 30th, 2016. All trajectories in the period from May 27th, 2015 to May 28th, 2016 – the peak of the construction of the new road – were

eliminated from the sample to avoid introducing noise in the computation of travel time.

To execute the evaluation, we selected a path of the monitored bus network that goes from the Ministro Ivan Lins Avenue to the Gávea Road (in the direction from Barra de Tijuca to the south zone). This path was heavily affected by the construction of the new elevated road.

For the old scenario, we analyzed a total of 24,846 trajectories, generated by 1,011 buses, serving 70 routes daily, that cover the path under study. Corresponding to the new scenario, we analyzed a total of 8,310 trajectories, generated by 115 buses serving 66 routes daily.

Since the travel times in weekdays differ dramatically from weekends, within the same scenario, we analyzed these periods separately. Figure 31 shows the travel time patterns for the weekdays belonging to the old scenario (v1) versus the new scenario (v2), while Figure 32depicts the travel time patterns for the weekends.

To estimate the difference between the patterns, we computed the area between the two curves during the morning peak hours (from 6 to 10 o'clock). The result was 15.00. This means an average reduction of the travel time in the morning peak hours by approximately 4 minutes.

As the graphs in Figure 31 corroborate, there are significant variations in travel time from one pattern to the other, specifically at the peak hours in the morning (from 6 to 10 o'clock), when the flow of vehicles in the direction Barra de Tijuca - south zone is larger than during the rest of the day. The results of the experiments then demonstrate that the commissioning of the New Joá Elevated Road produced a significant reduction of bus travel time from Barra da Tijuca to the south zone. Since such traffic change caused significant variations in the travel time patterns of buses.

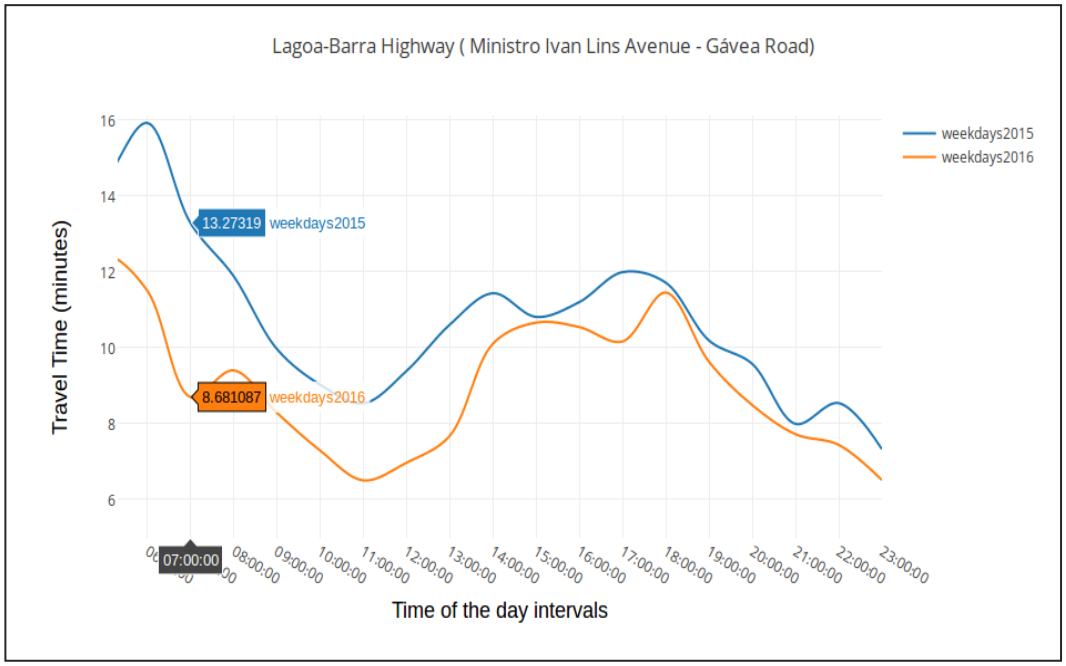


Figure 31: Travel Time Patterns for weekdays of v1 vs v2 Lagoa - Barra Highway.

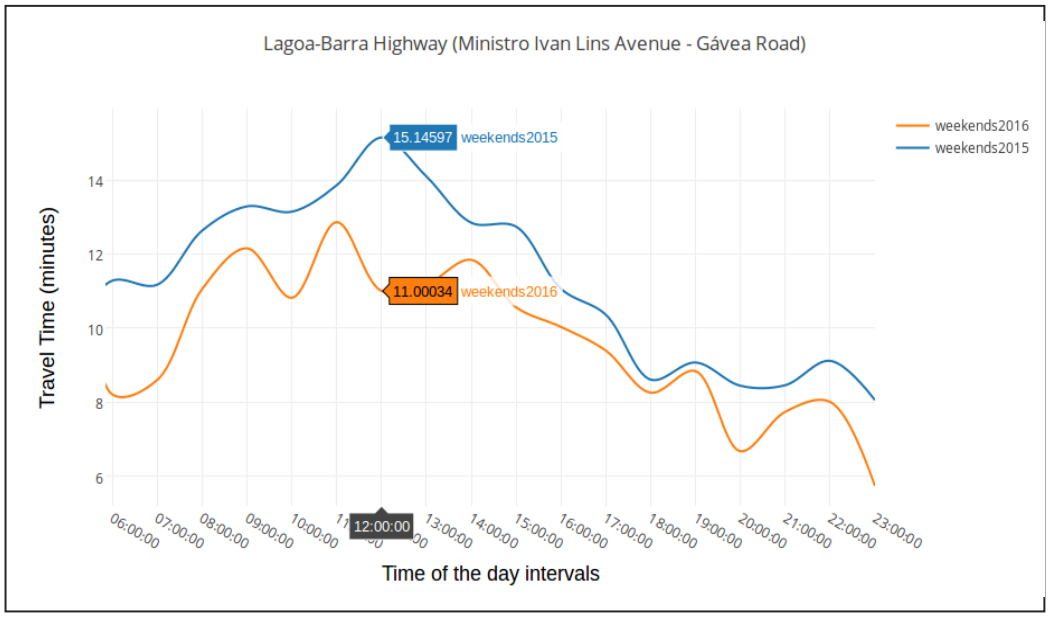


Figure 32: Travel Time Patterns for weekends v1 vs v2 Lagoa - Barra Highway.

7.7. Conclusions

In this chapter, we validated the functionalities of the prototype that supports the proposed approach. For the validation, a GPS dataset was used that was generated by more than 8,000 buses over a period of almost three years from June 12th, 2014 until February 28th, 2017 in Rio de Janeiro, Brazil.

The bus network of the City of Rio de Janeiro, Brazil was modeled and versioned. For each version, its corresponding monitored network was computed and segmented in monitored paths. Then, travel time patterns to traverse the monitored paths were discovered.

On the travel time data that correspond to the patterns, a normality test and a fit function were applied. The results corroborated that, among several probability distribution functions, the most appropriate for describing the travel time of buses under normal traffic conditions is a Gaussian distribution.

The results of the experiments about the traffic anomaly detection demonstrate the effectiveness of the proposed method since it allows identifying traffic anomalies that occurred in the metropolitan area of the City of Rio de Janeiro, which were corroborated against real-life events reported in newspapers, news channels, and tweets distributed by government agencies. The prototype was also able to estimate the duration of anomalies and quantitatively assess their impact in terms of travel time.

As demonstrated with the analysis of the New Joá Elevated Road, the prototype can also be used to evaluate how bus travel time patterns in the city are affected by structural and operational traffic changes. This functionality allows delimiting and comparing different versions of the bus network and assessing the evolution of the traffic conditions of the city.

Based on our experiments, we may conclude that the proposal helps analyzing and monitoring the bus network of a city. The experiments presented in this chapter can be reproduced using the prototype and the data available at https://github.com/kathrinr.llanes/PhD.

8
Conclusions and Directions for Future Work

In this thesis, we developed an approach for analyzing and monitoring the bus network of a city, which is based on the trajectory data continuously generated by the GPS installed on buses. In order to show the potential usefulness of the research, we implemented a prototype including the key operations and requirements that support the approach and tested it for the bus network of the City of Rio de Janeiro, Brazil. The tests use bus trajectories data collected from June 12th, 2014 until February 28th, 2017, which have been made available by the City Hall of the city.

In Chapter 3, we gave an overview of the approach, which involves a definition of the basic concepts used, a general description of the functionalities of the prototype and of its architecture, and a list of the tools used for its implementation.

In Chapter 4, we dealt with the modelling and analysis of a bus network. An evolutionary model of a bus network using versions was proposed. According to this modelling, a bus network consists of several versions, where each version maintains the same structural and operational features. The transition from one version to another is given by structural or operational changes of the bus network that significantly affect the travel time of buses. Additionally, for each bus network version, its respective monitored network is computed and segmented into paths, whose traffic will be monitored with the help of bus trajectories.

For the specific case of the bus network of Rio, during the period from June 12th, 2014 until February 28th, 2017, two versions were defined, one from June 12th, 2014 to May 20th, 2016, and the other from May 21st, 2016 to February 28th, 2017. Such versioning was motivated by two significant changes: (i) the itinerary modification of 180 bus lines in May 21st, 2016, that caused a reduction from 716 bus lines to 441; and (ii) the commissioning of the New Joá Elevated Road and the new roads to access it on May 28th, 2016, to facilitate the traffic flow between

the south zone of Rio to Barra de Tijuca. These changes strongly affected the travel time of buses, as the experiments reveal.

In Chapter 5, we addressed the problem of discovering frequent travel time patterns of buses from historical GPS dataset of bus trajectories. For this purpose, we adopted a spatio-temporal pattern mining approach and also included the directional component of bus movement. Based on it, the travel time that buses wasted to traverse each monitored path of the bus network version is calculated. Then, depending on a given temporal partition, the travel time pattern is computed as the average travel time of all buses that crossed each monitored path at a specific time interval.

In order to compute the travel time patterns for the two versions of the bus network of Rio, we used the following temporal partition: the validity period of a bus network version is divided into school classes period and school vacations, both divided into days of the week, with the weekdays separated in holidays and working days. The temporal segmentation at this level of granularity defines time periods. Then, each time period is divided into 24 fixed time intervals of one hour each. By using this temporal partition, we computed the travel time pattern for each monitored path of each bus network version. To validate the results, we selected some main roads of the city, and determined their travel time patterns. Finally, we verified that the bus travel time pattern obtained by the proposed algorithm is in accordance with frequent measurements of travel time made by the author using applications such as Wase and google maps[15].

The fact that we can store a version of a bus network with its respective structural and operational features, together with the travel time patterns of its monitored paths, allows us to better understand how the bus network evolves over time, which helps city planners assess changes.

In Chapter 6, we proposed a non-real-time and a real-time strategy for traffic anomaly detection, which use the travel time of the buses as an indicator to evaluate the behavior of traffic conditions. We also presented a technique to

[15]https://www.google.es/maps/

estimate the severity of a traffic anomaly and explained how to evaluate its impact in terms of incident duration and travel time delay.

In the non-real time strategy, a Statistical Quality Control technique was applied to detect traffic anomalies. As anomalies are detected, together with outliers, they are removed from the sample and the travel time pattern is recomputed. This is an iterative process that allows obtaining a refined pattern and actually determining all bus trips corresponding to a traffic anomaly. Therefore, the main results of the non-real time strategy are a set of anomalies that happened in past and a clean travel time pattern, which is used in the real-time strategy.

The real-time strategy combines the geofencing technology and the Statistical Quality Control technique to monitor buses operating in the monitored bus network and, thus, to detect traffic anomalies. The use of geofences allows to partition and process the GPS data stream that is continually arriving. A distributed architecture was proposed to simultaneously monitor all geofences of the bus network for detecting traffic anomalies as nearly as possible to real-time and to achieve incremental scalability.

The main limitation of our proposal is that the strategies for the detection of traffic anomalies require that the bus travel times, for each monitored path, during a given period of time, be available, and that the time interval follow a normal distribution.

In addition, we proposed a method to classify a traffic anomaly according to their severity into slight, moderate, severe and extreme; rather than the typical binary evaluation of whether an anomaly occurs or not, provided by traditional incident detection algorithms. Furthermore, we evaluated the impact of a traffic anomaly taking into consideration its duration and the travel time delays that it caused.

After applying the traffic anomaly detection strategy to two versions of the bus network of the City of Rio de Janeiro, 279 traffic anomalies were identified, that happened during the validity period of B_1 and 98 anomalies that happened during the validity period of B_2. Many of them were associated with identified events that affected traffic conditions in the city and that were published in

newspapers, news channels, tweets distributed by the City Hall (e.g. @OperacoesRio, @TransitoRioRJ, @CETRIO_ONLINE), and other mass media.

The experimental results demonstrated the effectiveness of our proposal in identifying a wide range of anomalies, estimating their severity and evaluating their impact. The assessed technical details are generic and, therefore, we expect that the derived insights will be useful for similar future research efforts.

Finally, we can conclude that the proposed approach is useful for analyzing and monitoring the bus network of a city, which may significantly help traffic managers and city authorities improve traffic control and mobility plans.

In the future, we intend to extend our work in the following three directions:

1. Develop a traffic observatory application including the algorithms we implemented in this thesis. This observatory would include a status map with the road monitored bus network segments colored according to the state of the traffic. In this context, green represents normal traffic state, beige slight anomaly, yellow moderate traffic anomaly, orange severe anomaly and red extreme traffic anomaly. Thus, the anomalous events can be easily located, visualized, and analyzed by urban planners. Furthermore, an alarm should be triggered at the moment an anomaly is identified, allowing traffic operators to act as soon as possible.
2. During the refinement of control points, which was addressed in algorithm 4 in Section 4.4, is recommended to use as a criterion to identify not significant intermediate nodes, the variation of the vector of bus routes between two consecutive street segments.
3. Extend both strategies for traffic anomaly detection (in real and in non-real time)in order to they can be able to work even when the travel times on a monitored path at a given period of time and time interval do not follow a normal distribution.
4. Propose a method to explain travel time anomalies by using a combination of change-detection analytics and auxiliary information from human-sensors (e.g., through Twitter).
5. Determine the impact of traffic anomalies in terms of congestion propagation. This problem can be solved via either a breadth- or a depth-first-search until there is no more anomalies in the vicinity, in time and space.

Bibliography

AGÊNCIA O DIA. Neste sábado, 180 linhas de ônibus sofrem mudança de itinerário no Centro. 13. May. 2016. Rio de Janeiro. Disponível em: <http://odia.ig.com.br/rio-de-janeiro/odia-no-coletivo/2016-05-13/neste-sabado-180-linhas-de-onibus-sofrem-mudanca-de-itinerario-no-centro.html>. .

ALEWIJNSE, S.; BUCHIN, K.; BUCHIN, M.; et al. A framework for trajectory segmentation by stable criteria. **Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.** p.351–360, 2014.

ALEWIJNSE, S. P. A.; BUCHIN, K.; BUCHIN, M.; SIJBEN, S.; WESTENBERG, M. A. Model-based segmentation and classification of trajectories.: **Proceedings of the 30th European Workshop on Computational Geometry**. Dead Sea, Israel,p.3–5, 2014.

AMARAL, B. G. DO; NASSER, R.; CASANOVA, M. A.; LOPES, H. BusesinRio: Buses as Mobile Traffic Sensors: Managing the Bus GPS Data in the City of Rio de Janeiro. **17th IEEE International Conference on Mobile Data Management (MDM).**v. 1, p.369–372, 2016.

ANASTASI, G.; ANTONELLI, M.; BECHINI, A.; et al. Urban and social sensing for sustainable mobility in smart cities. **Sustainable Internet and ICT for Sustainability (SustainIT).**p.1–4, 2013. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6685198>.

ANBAROGLU, B.; HEYDECKER, B.; CHENG, T. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. **Transportation Research Part C: Emerging Technologies**, v. 48, p. 47–65, 2014. Elsevier.

ANDRIENKO, G.; ANDRIENKO, N.; WROBEL, S. Visual analytics tools for analysis of movement data. **ACM SIGKDD Explorations Newsletter**, v. 9, n. 2,

p. 38–46, 2007.

ARANA, P.; CABEZUDO, S.; PEÑALBA, M. Influence of weather conditions on transit ridership: A statistical study using data from Smartcards. **Transportation Research Part A: Policy and Practice**, v. 59, p. 1–12, 2014. article, . Disponível em: <http://www.sciencedirect.com/science/article/pii/S0965856413002073>. .

ARBOLEDA, M. A.; PARRA, I. F.; ARISTIZÁBAL, I.; SABOGAL, H. Estudio dinámico de la movilidad en la ciudad de Santiago de Cali - Colombia. **X Congreso Latinoamericano de Dinámica de Sistemas.**2012. Buenos Aires, Argentina. Disponível em: <http://www.dinamica-de-sistemas.com/revista/dinamica-de-sistemas-17.pdf>. .

AYOB, M. A. **Developing and implementing geofencing in destination alarm mobile application**, 2015. UNIVERSITI TEKNOLOGI MARA. Disponível em: <http://ir.uitm.edu.my/15653/1/TM_MOHD ASRAF AYOB AP 15_5.pdf>. .

BAK, P.; MARDER, M.; HARARY, S.; YAELI, A.; SHIP, H. J. Scalable detection of spatiotemporal encounters in historical movement data. **Computer Graphics Forum.**v. 31, p.915–924, 2012.

BARBOSA, L.; KORMÁKSSON, M.; VIEIRA, M. R.; TAVARES, R. L.; ZADROZNY, B. Vistradas: Visual Analytics for Urban Trajectory Data. GEOINFO. **XV Brazilian Symposium on GeoInformatics.**p.11, 2014. article, São Paulo, Brazil. Disponível em: <http://www.geoinfo.info/proceedings_geoinfo2014.split/Paper13-S-p11.pdf>. .

BENJAMINI, Y. Opening the Box of a Boxplot. **The American Statistician**, v. 42, n. 4, p. 257–262, 1988.

BERA, S.; RAO, K. V. Estimation of origin-destination matrix from traffic counts: the state of the art. , 2011. EUT Edizioni Università di Trieste.

BILJECKI, F. Automatic segmentation and classification of movement trajectories for transportation modes. 2010. **phdthesis**, TU Delft, Delft University of Technology. Disponível em: <http://repository.tudelft.nl/islandora/object/uuid:654587d2-6e93-4619-ab9a-

29d95f843f35?collection=education>. .

BILJECKI, F.; LEDOUX, H.; OOSTEROM, P. VAN. Transportation mode-based segmentation and classification of movement trajectories. **International Journal of Geographical Information Science**, v. 27, n. 2, p. 385–407, 2013.

BONA, A. A. DE; FONSECA, K. V. O.; ROSA, M. O.; LÜDERS, R.; DELGADO, M. Analysis of Public Bus Transportation of a Brazilian City Based on the Theory of Complex Networks Using the P-Space. **Mathematical Problems in Engineering**, v. 2016, 2016.

BUCHIN, M.; DRIEMEL, A.; KREVELD, M. VAN; SACRISTÁN, V. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. **Journal of Spatial Information Science**, v. 2011, n. 3, p. 33–63, 2011.

BUCHIN, M.; KRUCKENBERG, H.; KÖLZSCH, A. Segmenting trajectories based on movement states. **Proc. 15th Internat. Sympos. Spatial Data Handling (SDH)**, p. 15–25, 2012.

CAMOSSI, E.; VILLA, P.; MAZZOLA, L. Semantic-based anomalous pattern discovery in moving object trajectories. arXiv preprint arXiv:1305.1946, 2013.

CARAGLIU, A.; BO, C. DEL; NIJKAMP, P. Smart cities in Europe. **Journal of urban technology**, v. 18, n. 2, p. 65–82, 2011.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM computing surveys (CSUR)**, v. 41, n. 3, p. 15, 2009.

CHATTERJEE, A. Studies on the structure and dynamics of urban bus networks in Indian cities. **arXiv preprint arXiv:1512.05909**, 2015.

CHATTERJEE, A.; MANOHAR, M.; RAMADURAI, G. Statistical analysis of bus networks in India. **PloS one**, v. 11, n. 12, 2016.

CHAWLA, S.; ZHENG, Y.; HU, J. Inferring the root cause in road traffic anomalies., 2012 **IEEE 12th International Conference on Data Mining (ICDM).** p.141–150, 2012.

CHEN, C.; ZHANG, D.; CASTRO, P. S.; et al. Real-time detection of anomalous

taxi trajectories from GPS traces. **International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services.**p.63–74, 2011.

CHEN, P.-T.; CHEN, F.; QIAN, Z. Road traffic congestion monitoring in social media with hinge-loss Markov random fields. **IEEE International Conference on Data Mining (ICDM).**p.80–89, 2014.

CHEN, S.; WANG, W.; ZUYLEN, H. VAN. A comparison of outlier detection algorithms for ITS data. **Expert Systems with Applications**, v. 37, n. 2, p. 1169–1178, 2010.

CHEUNG, D. Methods, systems, and apparatus for a geo-fence system. 2016. United States. Disponível em: <https://www.google.com/patents/US20160198298>. Acesso em: 8/2/2017.

CHU, X. The Efficiency of Sampling Techniques for NTD Reporting. **Journal of Public Transportation**, v. 12, n. 4, p. 1, 2009. article, .

CHUNG, Y.-S.; CHIOU, Y.-C.; LIN, C.-H. Simultaneous equation modeling of freeway accident duration and lanes blocked. **Analytic Methods in Accident Research**, v. 7, p. 16–28, 2015. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2213665715000275>. .

COOLS, M.; MOONS, E.; WETS, G. Assessing the Impact of Weather on Traffic Intensity. **Weather, Climate, and Society**, v. 2, n. 1, p. 60–68, 2010. Disponível em: <http://journals.ametsoc.org/doi/abs/10.1175/2009WCAS1014.1>. Acesso em: 22/11/2015.

COQUITA, K.; RISTAR, A.; OLIVEIRA, A. DE; TEDESCO, P. Prediction System of Bus Arrival Time Based on Historical Data Using Regression Models. **Proceedings of the XI Brazilian Symposium on Information Systems (SBSI 2015)**, 2015. Disponível em: <http://aisel.aisnet.org/sbis2015/84>. Acesso em: 18/8/2015.

COSTANZO, A. Using GPS data to monitor road traffic flows in a metropolitan area: methodology and case study. **Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering.**2013.

D'ANDREA, E.; DUCANGE, P.; LAZZERINI, B.; MARCELLONI, F. Real-Time Detection of Traffic From Twitter Stream Analysis. **IEEE Transactions on Intelligent Transportation Systems**, v. 16, n. 4, p. 2269–2283, 2015. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7057672>. Acesso em: 17/8/2015.

DADOS RIO. Prefeitura da Cidade do Rio de Janeiro. Disponível em: <http://data.rio/dataset/gps-de-onibus>. .

DAEINABI, A.; RAHBAR, A. G. P.; KHADEMZADEH, A. VWCA: An efficient clustering algorithm in vehicular ad hoc networks. **Journal of Network and Computer Applications**, v. 34, n. 1, p. 207–222, 2011.

DAL PIVA, J.; ESTARQUE, M. **Rio Ônibus divulga dados truncados para justificar aumento da tarifa**. 2017.

DALY, E. M.; LECUE, F.; BICER, V. Westland row why so slow?: fusing social media and linked data sources for understanding real-time traffic conditions. **Proceedings of the 2013 international conference on Intelligent user interfaces.**p.203–212, 2013.

DAS, R. D.; WINTER, S. Automated Urban Travel Interpretation: A Bottom-up Approach for Trajectory Segmentation. **Sensors**, v. 16, n. 11, p. 1962, 2016.

DU, L.; PEETA, S.; KIM, Y. H. An adaptive information fusion model to predict the short-term link travel time distribution in dynamic traffic networks. **Transportation Research Part B: Methodological**, v. 46, n. 1, p. 235–252, 2012.

DUNNE, S.; GHOSH, B. Regime-based short-term multivariate traffic condition forecasting algorithm. **Journal of Transportation Engineering**, v. 138, n. 4, p. 455–466, 2011.

ESKIN, E. Anomaly detection over noisy data using learned probability distributions. **Proceedings of the International Conference on Machine Learning.** 2000.

ETIENNE, L.; DEVOGELE, T.; BOUJU, A. Spatio-temporal trajectory analysis

of mobile objects following the same itinerary. **Advances in Geo-Spatial Information Science**, v. 10, p. 47–57, 2012.

FAN, P.; MOHAMMADIAN, A.; NELSON, P. C.; HARAN, J.; DILLENBURG, J. A novel direction-based clustering algorithm in vehicular ad hoc networks. **Transportation research board 86th annual meeting.**2007.

G1 GLOBO. Conheça o esquema de tráfego com a inauguração do Novo Joá, no Rio. **G1 RJ**, 27. May. 2016. Rio de Janeiro. Disponível em: <http://g1.globo.com/rio-de-janeiro/noticia/2016/05/conheca-o-esquema-de-trafego-com-inauguracao-do-novo-joa-no-rio.html>. .

G1 RIO. Comemorações marcam o aniversário de 450 anos do Rio. , 2015a. Rio de Janeiro. Disponível em: <http://g1.globo.com/rio-de-janeiro/rio-450-anos/noticia/2015/03/comemoracoes-marcam-o-aniversario-de-450-anos-do-rio.html>. .

G1 RIO. Zona Sul do Rio tem acidente com morto e manhã de trânsito confuso. , 2015b. Rio de Janeiro. Disponível em: <http://g1.globo.com/rio-de-janeiro/transito/noticia/2015/08/acidente-deixa-morto-e-atrapalha-transito-na-zona-sul-do-rio.html>. .

G1 RIO. Motoristas enfrentam engarrafamento para chegar à Bienal do Livro. , 2015c. Rio de Janeiro. Disponível em: <http://g1.globo.com/rio-de-janeiro/noticia/2015/09/motoristas-enfrentam-engarrafamento-para-chegar-bienal-do-livro.html>. .

G1 RIO. Bloqueio para obras trava Av. Brasil e Ponte Rio-Niterói. , 2016a. Rio de Janeiro. Disponível em: <http://g1.globo.com/rio-de-janeiro/noticia/2016/07/bloqueio-para-obras-trava-av-brasil-e-ponte-rio-niteroi.html?utm_source=facebook&utm_medium=social&utm_campaign=rjtv>. .

G1 RIO. Veja interdições para a chegada da tocha olímpica no Rio. , 2016b. Rio de Janeiro. Disponível em: <http://g1.globo.com/rio-de-janeiro/olimpiadas/rio2016/noticia/2016/08/veja-interdicoes-para-chegada-da-tocha-olimpica-no-rio.html>. .

G1 RIO. Acidentes deixam trânsito caótico na Av. Brasil nesta sexta-feira no Rio.

, 2016c. Rio de Janeiro. Disponível em: <http://g1.globo.com/rio-de-janeiro/noticia/2016/11/caminhao-tomba-e-interdita-pista-lateral-da-av-brasil-no-rio.html>. .

GONG, L.; SATO, H.; YAMAMOTO, T.; MIWA, T.; MORIKAWA, T. Identification of activity stop locations in GPS trajectories by density-based clustering method combined with support vector machines. **Journal of Modern Transportation**, v. 23, n. 3, p. 202–213, 2015.

GONZALEZ, P. A.; WEINSTEIN, J. S.; BARBEAU, S. J.; et al. Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. **IET Intelligent Transport Systems**, v. 4, n. 1, p. 37–49, 2010.

GOOGLE. GTFS Specification. Disponível em: <https://developers.google.com/transit/gtfs/reference/>. Acesso em: 22/3/2017.

GRASER, A.; PONWEISER, W.; DRAGASCHNIG, M.; BRÄNDLE, N.; WIDHALM, P. Assessing traffic performance using position density of sparse FCD. **15th International IEEE Conference on Intelligent Transportation Systems (ITSC).** p.1001–1005, 2012.

GRUBBS, F. E. Procedures for detecting outlying observations in samples. **Technometrics**, v. 11, n. 1, p. 1–21, 1969.

GUBERFAIN, B.; CÔRTES VIEIRA, H. **A visual analysis of bus GPS data in Rio**, 2015. Pontifícia Universidade Católica do Rio de Janeiro. Disponível em: <http://www.inf.puc-rio.br/?p=3526>. .

HA, J.-A.; OH, J.-S. Estimating Annual Average Daily traffic using Daily Adjustment Factor. **Journal of emerging trends in Computing and Information Sciences**, p. 580, 2014.

HABTEMICHAEL, F. G.; CETIN, M. Short-term traffic flow rate forecasting based on identifying similar traffic patterns. **Transportation Research Part C: Emerging Technologies**, 2015.

HABTIE, A. B.; ABRAHAM, A.; MIDEKSO, D. A Neural Network Model for

Road Traffic Flow Estimation. **Advances in Nature and Biologically Inspired Computing**. p.305–314, 2016.

HÁZNAGY, A.; FI, I.; LONDON, A.; NÉMETH, T. Complex network analysis of public transportation networks: a comprehensive study. **International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)** p.371–378, 2015.

HIBON, M.; MAKRIDAKIS, S. ARMA models and the Box--Jenkins methodology. , 1997. John Wiley & Sons, Ltd.

HONG, W.-C.; DONG, Y.; ZHENG, F.; WEI, S. Y. Hybrid evolutionary algorithms in a SVR traffic flow forecasting model. **Applied Mathematics and Computation**, v. 217, n. 15, p. 6733–6747, 2011.

HOTELLING, H. The Generalization of Student's Ratio. **Ann. Math. Statist.**, v. 2, n. 3, p. 360–378, 1931. The Institute of Mathematical Statistics. Disponível em: <http://dx.doi.org/10.1214/aoms/1177732979>.

HOU, L.; ZHANG, Z.; LU, B.; XU, R.; ZHANG, Y. Estimation of incident-induced congestion on signalized arteries using traffic sensor data. **Tsinghua Science and Technology**, 2012.

HUANG, G.; HE, J.; ZHOU, W.; et al. Discovery of stop regions for understanding repeat travel behaviors of moving objects. **Journal of Computer and System Sciences**, v. 82, n. 4, p. 582–593, 2016. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0022000015001348>. Acesso em: 4/3/2017.

HUANG, G.; ZHANG, Y.; HE, J.; DING, Z. Efficiently retrieving longest common route patterns of moving objects by summarizing turning regions. **Pacific-Asia Conference on Knowledge Discovery and Data Mining.**p.375–386, 2011.

HUNG, C.-C.; PENG, W.-C.; LEE, W.-C. Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. **The VLDB JournalThe International Journal on Very Large Data Bases**, v. 24, n. 2, p. 169–192, 2015. Springer-Verlag New York, Inc.

JARQUE, C. M.; BERA, A. K. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. **Economics letters**, v. 6, n. 3, p. 255–259, 1980.

JI, Y.; MISHALANI, R. G.; MCCORD, M. R. Transit passenger origin--destination flow estimation: Efficiently combining onboard survey and large automatic passenger count datasets. **Transportation Research Part C: Emerging Technologies**, v. 58, p. 178–192, 2015.

JITHENDRA. H. K, N. R. K. D. Predicting Bus Arrival Time based on Traffic Modelling and Real-time Delay. **International Journal of Engineering Research & Technology**, v. Volume. 4, n. Volume. 4-Issue. 06, June-2015, 2015. ESRSA Publications. Disponível em: <http://www.ijert.org/view-pdf/13439/predicting-bus-arrival-time-based-on-traffic-modelling-and-real-time-delay>. Acesso em: 15/8/2015.

JOGLEKAR, A. M. **Statistical methods for six sigma: in R&D and manufacturing**. book, John Wiley & Sons, 2003.

KANG, J.-Y.; YONG, H.-S. Mining spatio-temporal patterns in trajectory data. **Journal of Information Processing Systems**, v. 6, n. 4, p. 521–536, 2010. Korea Information Processing Society.

KINOSHITA, A.; TAKASU, A.; ADACHI, J. Real-time traffic incident detection using a probabilistic topic model. **Information Systems**, v. 54, p. 169–188, 2015.

KOETSE, M. J.; RIETVELD, P. The impact of climate change and weather on transport: An overview of empirical findings. **Transportation Research Part D: Transport and Environment**, v. 14, n. 3, p. 205–221, 2009. Disponível em: <http://www.sciencedirect.com/science/article/pii/S136192090800165X>.

KONG, X.; XU, Z.; SHEN, G.; et al. Urban traffic congestion estimation and prediction based on floating car trajectory data. **Future Generation Computer Systems**, v. 61, p. 97–107, 2016.

KORMÁKSSON, M.; BARBOSA, L.; VIEIRA, M. R.; ZADROZNY, B. Bus travel time predictions using additive models. **IEEE International Conference on Data Mining.**p.875–880, 2014.

KRINGS, G.; CALABRESE, F.; RATTI, C.; BLONDEL, V. D. Urban gravity: a model for inter-city telecommunication flows. **Journal of Statistical Mechanics: Theory and Experiment**, v. 2009, n. 7, p. L07003, 2009. IOP Publishing.

KRUMM, J.; HORVITZ, E. Predestination: Inferring destinations from partial trajectories. **International Conference on Ubiquitous Computing.**p.243–260, 2006.

KUANG, W.; AN, S.; JIANG, H. Detecting traffic anomalies in urban areas using taxi GPS data. **Mathematical Problems in Engineering**, v. 2015, 2015. Hindawi Publishing Corporation.

LAN, J.; LONG, C.; WONG, R. C.-W.; et al. A new framework for traffic anomaly detection. **Proceedings of the 2014 SIAM International Conference on Data Mining.**p.875–883, 2014.

LAUBE, P.; IMFELD, S.; WEIBEL, R. Discovering relative motion patterns in groups of moving point objects. **International Journal of Geographical Information Science**, v. 19, n. 6, p. 639–668, 2005.

LEE, J.-G.; HAN, J.; WHANG, K.-Y. Trajectory clustering: a partition-and-group framework. **Proceedings of the 2007 ACM SIGMOD international conference on Management of data.**p.593–604, 2007.

LEE, W.-C.; SI, W.; CHEN, L.-J.; CHEN, M. C. HTTP: a new framework for bus travel time prediction based on historical trajectories. **Proceedings of the 20th International Conference on Advances in Geographic Information Systems.** p.279–288, 2012.

LI, J.; CHEN, X.; LI, X.; GUO, X. Evaluation of public transportation operation based on data envelopment analysis. **Procedia-Social and Behavioral Sciences**, v. 96, p. 148–155, 2013.

LI, Q.; ZHENG, Y.; XIE, X.; et al. Mining user similarity based on location history. **Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems.**p.34, 2008.

LI, R. Traffic incident duration analysis and prediction models based on the

survival analysis approach. **IET Intelligent Transport Systems**, v. 9, n. 4, p. 351–358, 2015. IET Digital Library. Disponível em: <http://digital-library.theiet.org/content/journals/10.1049/iet-its.2014.0036>. Acesso em: 5/11/2015.

LIAO, L.; FOX, D.; KAUTZ, H. BLocation-based activity recognition using relational Markov networks. **Proceedings of the 19th International Joint Conference on Artificial Intelligence.** Edinburgh, Scotland. p.773–778, 2005.

LIAO, L.; PATTERSON, D. J.; FOX, D.; KAUTZ, H. Building personal maps from GPS data. **Annals of the New York Academy of Sciences**, v. 1093, n. 1, p. 249–265, 2006.

LIU, W.; ZHENG, Y.; CHAWLA, S.; YUAN, J.; XING, X. Discovering spatio-temporal causal interactions in traffic data streams. **Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.**p.1010–1018, 2011.

LIU, X.; FANG, X.; QIN, Z.; YE, C.; XIE, M. A Short-term forecasting algorithm for network traffic based on chaos theory and SVM. **Journal of network and systems management**, v. 19, n. 4, p. 427–447, 2011.

LLANES, K. R.; CASANOVA, M. A.; LEME, L. A. P. P.; et al. On the Design of a Traffic Observatory Application based on Bus Trajectories. **Proceedings of the International Conference of Enterprise Information Systems (ICEIS)**2016.

LLANES, K. R.; CASANOVA, M. A.; LEMUS, N. M. From the sensor data streams to linked streaming data. **3rd Symposium on Knowledge Discovery, Mining and Learning (KDMiLe).** 2015.

LLANES, K. R.; CASANOVA, M. A.; LEMUS, N. M. From Sensor Data Streams to Linked Streaming Data: a survey of main approaches. **Journal of Information and Data Management - JIDM**, v. 7, p. 130–140, 2016.

LLANES, K. R.; CASANOVA, M. A.; LOPES, H.; MACEDO, J. A. F. DE. An approach to evaluate the imapct on travel time of bus network changes. **Proceedings of the International Conference of Enterprise Information Systems (ICEIS)** 2017.

LV, Y.; DUAN, Y.; KANG, W.; LI, Z.; WANG, F.-Y. Traffic flow prediction with big data: a deep learning approach. **IEEE Transactions on Intelligent Transportation Systems**, v. 16, n. 2, p. 865–873, 2015.

MANLEY, E. Estimating urban traffic patterns through probabilistic interconnectivity of road network junctions. **Public Library of Science one**, v. 10, n. 5, p. e0127095, 2015.

MARTINEZ, H.; MAUTTONE, A.; URQUHART, M. E. Frequency optimization in public transportation systems: Formulation and metaheuristic approach. **European Journal of Operational Research**, v. 236, n. 1, p. 27–36, 2014.

MATHEUS, R.; RIBEIRO, M. M. **Case Study Open Government Data in Rio de Janeiro City**. Rio de Janeiro, 2014.

MAUTTONE, A.; CANCELA, H.; URQUHART, M. Diseño y optimización de rutas y frecuencias en el transporte colectivo urbano, modelos y algoritmos. Universidad de la República Facultad de Ingeniería, Uruguay, 2010.

MAZIMPAKA, J. D.; TIMPF, S. Trajectory data mining: A review of methods and applications. **Journal of Spatial Information Science**, v. 2016, n. 13, p. 61–99, 2016. article, .

MILLER, M.; GUPTA, C. Mining traffic incidents to forecast impact. **Proceedings of the ACM SIGKDD International Workshop on Urban Computing.**p.33–40, 2012. Disponível em: <http://dl.acm.org/citation.cfm?id=2346502>. .

MONTGOMERY, D. C. **Statistical quality control**. book, Wiley New York, 2009.

MONTGOMERY, D. C.; RUNGER, G. C. **Applied statistics and probability for engineers**. book, John Wiley & Sons, 2010.

MORENO, B.; TIMES, V. C.; RENSO, C.; BOGORNY, V. Looking Inside the Stops of Trajectories of Moving Objects. Geoinfo. p.9–20, 2010.

NARAYANAN, A.; MITROVIC, N.; ASIF, M. T.; DAUWELS, J.; JAILLET, P. Travel time estimation using speed predictions. **IEEE 18th International**

Conference on Intelligent Transportation Systems (ITSC). p.2256–2261, 2015.

NECULA, E. Analyzing Traffic Patterns on Street Segments Based on GPS Data Using R. **Transportation Research Procedia**, v. 10, p. 276–285, 2015.

NOEI, S.; SANTANA, H.; SARGOLZAEI, A.; NOEI, M. Reducing Traffic Congestion Using Geo-fence Technology. **Proceedings of the 1st International Workshop on Emerging Multimedia Applications and Services for Smart Cities - EMASC '14.**p.15–20, 2014. New York, New York, USA: ACM Press. Disponível em: <http://dl.acm.org/citation.cfm?doid=2661704.2661709>. Acesso em: 8/2/2017.

PALMA, A. T.; BOGORNY, V.; KUIJPERS, B.; ALVARES, L. O. A clustering-based approach for discovering interesting places in trajectories. **Proceedings of the 2008 ACM symposium on Applied computing.**p.863–868, 2008.

PAN, B.; ZHENG, Y.; WILKIE, D.; SHAHABI, C. Crowd sensing of traffic anomalies based on human mobility and social media. **Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.**p.344–353, 2013.

PANG, L. X.; CHAWLA, S.; LIU, W.; ZHENG, Y. On detection of emerging anomalous traffic patterns using GPS data. **Data & Knowledge Engineering**, v. 87, p. 357–373, 2013.

PARENT, C.; SPACCAPIETRA, S.; RENSO, C.; et al. Semantic trajectories modeling and analysis. **ACM Computing Surveys (CSUR)**, v. 45, n. 4, p. 42, 2013.

PARK, H.; HAGHANI, A.; ZHANG, X. Interpretation of bayesian neural networks for predicting the duration of detected incidents. **Journal of Intelligent Transportation Systems**, p. 1–16, 2015. Taylor & Francis. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/15472450.2015.1082428>. Acesso em: 21/11/2015.

PARZEN, E. On estimation of a probability density function and mode. **The annals of mathematical statistics**, v. 33, n. 3, p. 1065–1076, 1962.

PREFEITURA DA CIDADE DO RIO DE JANEIRO. Transparência da Mobilidade. Disponível em: <http://www.rio.rj.gov.br/web/transparenciadamobilidade/>. Acesso em: 22/1/2015.

PREFEITURA DA CIDADE DO RIO DE JANEIRO. **Transparência da Mobilidade.** Disponível em: <http://www.rio.rj.gov.br/c/document_library/get_file?uuid=31e0bc80-4f91-4062-9900-aa49ece0938f&groupId=4800437>. .

RAIYN, J.; TOLEDO, T. Real-time road traffic anomaly detection. **Journal of Transportation Technologies**, v. 4, n. 3, p. 256, 2014.

RAO, K. V.; GOVARDHAN, A.; RAO, K. V. C. Spatiotemporal data mining: Issues, tasks and applications. **International Journal of Computer Science and Engineering Survey**, v. 3, n. 1, p. 39, 2012.

REDDY, S.; MUN, M.; BURKE, J.; et al. Using mobile phones to determine transportation modes. **ACM Transactions on Sensor Networks (TOSN)**, v. 6, n. 2, p. 13, 2010.

REMPE, F.; HUBER, G.; BOGENBERGER, K. Spatio-Temporal Congestion Patterns in Urban Traffic Networks. **Transportation Research Procedia**, v. 15, p. 513–524, 2016. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S2352146516305750>. Acesso em: 11/3/2017.

RESENDE, D.; PAULA, D. DE. Manifestação de taxistas contra Uber provoca caos no trânsito do Rio. **O Globo**, 2016. Rio de Janeiro. Disponível em: <http://oglobo.globo.com/rio/manifestacao-de-taxistas-contra-uber-provoca-caos-no-transito-do-rio-18996126>. .

ROCHA, J. A. M. R.; TIMES, V. C.; OLIVEIRA, G.; ALVARES, L. O.; BOGORNY, V. DB-SMoT: A direction-based spatio-temporal clustering method. **5th IEEE International Conference of Intelligent Systems**. p.114–119, 2010.

SAHA, P.; SHINSTINE, D. Analysis of expansions of a bus transit network considering the needs identified by the community: case study. **Journal of**

Transport Literature, v. 9, n. 2, p. 35–39, 2015.

SAKAKI, T.; OKAZAKI, M.; MATSUO, Y. Tweet analysis for real-time event detection and earthquake reporting system development. **IEEE Transactions on Knowledge and Data Engineering**, v. 25, n. 4, p. 919–931, 2013.

SANKARARAMAN, S.; AGARWAL, P. K.; MØLHAVE, T.; PAN, J.; BOEDIHARDJO, A. P. Model-driven matching and segmentation of trajectories. **Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.**p.234–243, 2013.

SEGOV. Operações especiais antecedem os Jogos Rio 2016 e preparam a cidade para o evento. Disponível em: <http://www.rio.rj.gov.br/web/segov/exibeconteudo?id=6311382>. Acesso em: 24/10/2016.

SETHI, J. R. Study of Distance-Based Outlier Detection Methods, 2013. **phdthesis, NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA**.

SHEN, L.; STOPHER, P. R. Review of GPS Travel Survey and GPS Data-Processing Methods. **Transport Reviews**, v. 34, n. 3, p. 316–334, 2014. Routledge. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/01441647.2014.903530>. Acesso em: 23/2/2017.

SHEWHART, W. A.; DEMING, W. E. **Statistical method from the viewpoint of quality control**. 1939.

SHI, W.; KONG, Q.-J.; LIU, Y. A GPS/GIS integrated system for urban traffic flow analysis. 2008 **11th International IEEE Conference on Intelligent Transportation Systems.**p.844–849, 2008.

SINGHAL, A.; KAMGA, C.; YAZICI, A. Impact of weather on urban transit ridership. **Transportation Research Part A: Policy and Practice**, v. 69, p. 379–391, 2014. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0965856414002195>. Acesso em: 22/11/2015.

SPACCAPIETRA, S.; PARENT, C.; DAMIANI, M. L.; et al. A conceptual view on trajectories. **Data & knowledge engineering**, v. 65, n. 1, p. 126–146, 2008.

STATHOPOULOS, A.; KARLAFTIS, M.; DIMITRIOU, L. Fuzzy rule-based system approach to combining traffic count forecasts. **Transportation Research Record: Journal of the Transportation Research Board**, , n. 2183, p. 120–128, 2010.

STATLER, S. Geofencing: Everything You Need to Know. **Beacon Technologies**. p.307–316, 2016.

STONEBRAKER, M.; ÇETINTEMEL, U.; ZDONIK, S. The 8 requirements of real-time stream processing. **ACM SIGMOD Record**, v. 34, n. 4, p. 42–47, 2005.

SUN, S.; HUANG, R.; GAO, Y. Network-scale traffic modeling and forecasting with graphical lasso and neural networks. **Journal of Transportation Engineering**, v. 138, n. 11, p. 1358–1367, 2012. American Society of Civil Engineers.

SUN, S.; XU, X. Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction. **IEEE Transactions on Intelligent Transportation Systems**, v. 12, n. 2, p. 466–475, 2011.

SURACE, C.; WORDEN, K.; OTHERS. A novelty detection method to diagnose damage in structures: an application to an offshore platform. **The 8th International Offshore and Polar Engineering Conference.** 1998.

SUTAGUNDAR, A. V; HUBBALLI, P.; BELAGALI, R. Stability Oriented Cluster Dynamism in VANET (SOCDV). **International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS),**. p.117–120, 2016.

TAVASSOLI HOJATI, A. Modelling the impact of traffic incidents on travel time reliability, Oct. 2014. **phdthesis, The University of Queensland.**Disponível em: <http://espace.library.uq.edu.au/view/UQ:344024>. .

The SMARTY project. .Disponível em: <http://www.smarty.toscana.it/>. .

TOOR, M. L.; NEWMAN, S. H.; TAKEKAWA, J. Y.; WEGMANN, M.; SAFI,

K. Temporal segmentation of animal trajectories informed by habitat use. **Ecosphere**, v. 7, n. 10, 2016.

TRAN, L. H.; NGUYEN, Q. V. H.; DO, N. H.; YAN, Z. **Robust and hierarchical stop discovery in sparse and diverse trajectories**. techreport, 2011.

TRIOLA, M. F. **Estadística**. Pearson education, 2004.

TUROCHY, R. E.; SMITH, B. L. Applying quality control to traffic condition monitoring. **Proceedings of International Conference of Intelligent Transportation Systems.** p.15–20, 2000.

VAIDYA, O. S. Evaluating the Performance of Public Urban Transportation Systems in India. **Journal of Public Transportation**, v. 17, n. 4, p. 11, 2014.

VLACHOS, M.; HADJIELEFTHERIOU, M.; GUNOPULOS, D.; KEOGH, E. Indexing multidimensional time-series. **The VLDB JournalThe International Journal on Very Large Data Bases**, v. 15, n. 1, p. 1–20, 2006.

VUCHIC, V. R. **Urban transit: operations, planning, and economics**. 2005.

WAGA, K.; TABARCEA, A.; CHEN, M.; FRANTI, P. Detecting movement type by route segmentation and classification. **8th International Conference on Collaborative computing: networking, applications and worksharing (CollaborateCom).** 2012 p.508–513, 2012.

WANG, Y.; CHEN, Y.; LAI, J. Fuzzy Prediction for Traffic Flow Based on Delta Test. **Mathematical Problems in Engineering**, v. 2016, 2016.

WANG, Y.; ZHENG, Y.; XUE, Y. Travel time estimation of a path using sparse trajectories. **Proceedings of the 20th ACM SIGKDD International Conference on Knowledge discovery and Data Mining.**p.25–34, 2014.

WANG, Z.; LU, M.; YUAN, X.; ZHANG, J.; WETERING, H. VAN DE. Visual traffic jam analysis based on trajectory data. **IEEE Transactions on Visualization and Computer Graphics**, v. 19, n. 12, p. 2159–2168, 2013.

WAZE MOBILE. **Wase Application.** Disponível em: <https://www.waze.com/>.

XIANG, L.; GAO, M.; WU, T. Extracting Stops from Noisy Trajectories: A Sequence Oriented Clustering Approach. **ISPRS International Journal of Geo-Information**, v. 5, n. 3, p. 29, 2016.

XIE, W.; WANG, J. Modelling the impact scope of urban express way incident. **International Conference on Transportation Information and Safety (ICTIS).**p.121–125, 2015. IEEE. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7232052>. Acesso em: 21/11/2015.

XU, C.; JI, M.; CHEN, W.; ZHANG, Z. Identifying travel mode from GPS trajectories through fuzzy pattern recognition. **7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD).** . v. 2, p.889–893, 2010.

YAN, Z.; CHAKRABORTY, D.; PARENT, C.; SPACCAPIETRA, S.; ABERER, K. SeMiTri: a framework for semantic annotation of heterogeneous trajectories. **Proceedings of the 14th International Conference on Extending Database Technology.** p.259–270, 2011.

YAN, Z.; CHAKRABORTY, D.; PARENT, C.; SPACCAPIETRA, S.; ABERER, K. Semantic trajectories: Mobility data computation and annotation. **ACM Transactions on Intelligent Systems and Technology (TIST)**, v. 4, n. 3, p. 49, 2013.

YANG, Q.; WANG, J.; SONG, X.; et al. Urban traffic congestion prediction using floating Car trajectory data. **International Conference on Algorithms and Architectures for Parallel Processing.** p.18–30, 2015.

YE, P.; CHEN, Z.; XU, L. Analyzing Travel Time Variability on Transit Route Using GPS Data. **ICTE 2015.**p.448–456, 2015. Reston, VA: American Society of Civil Engineers. Disponível em: <http://ascelibrary.org/doi/10.1061/9780784479384.058>. Acesso em: 23/2/2017.

YOON, H.; SHAHABI, C. Robust time-referenced segmentation of moving object trajectories. **8th IEEE International Conference on Data Mining.**p.1121–1126, 2008

YU, C.; HE, Z.-C. Analysing the spatial-temporal characteristics of bus travel

demand using the heat map. **Journal of Transport Geography**, v. 58, p. 247–255, 2017.

ZHANG, H. Structural Analysis of Bus Networks Using Indicators of Graph Theory and Complex Network Theory. **The Open Civil Engineering Journal**, v. 11, n. 1, 2017.

ZHANG, J.-D.; XU, J.; LIAO, S. S. Aggregating and sampling methods for processing GPS data streams for traffic state estimation. **IEEE Transactions on Intelligent Transportation Systems**, v. 14, n. 4, p. 1629–1641, 2013.

ZHANG, X.; WU, Y.; SHEN, L.; SKITMORE, M. A prototype system dynamic model for assessing the sustainability of construction projects. **International Journal of Project Management**, v. 32, n. 1, p. 66–76, 2014.

ZHENG, Y.; CHEN, Y.; LI, Q.; XIE, X.; MA, W.-Y. Understanding transportation modes based on GPS data for web applications. **ACM Transactions on the Web (TWEB)**, v. 4, n. 1, p. 1, 2010.

ZHENG, Y.; ZHANG, L.; MA, Z.; XIE, X.; MA, W.-Y. Recommending friends and locations based on individual location history. **ACM Transactions on the Web (TWEB)**, v. 5, n. 1, p. 5, 2011.

ZHU, B.; XU, X. Urban Principal Traffic Flow Analysis Based on Taxi Trajectories Mining. **International Conference in Swarm Intelligence.**p.172–181, 2015.

ZHU, T.; MA, F.; MA, T.; LI, C. The prediction of bus arrival time using global positioning system data and dynamic traffic information. **Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP.**p.1–5, 2011.

ZIMMERMANN, M.; KIRSTE, T.; SPILIOPOULOU, M. Finding stops in error-prone trajectories of moving objects with time-based clustering. **Intelligent interactive assistance and mobile multimedia computing**. p.275–286, 2009.

Appendix A

Table 9: Control Chart Constants.

				$\overline{X}$ and R Charts			$\overline{X}$ and S Charts		
n	d_2	d_3	C_4	A_2	D_3	D_4	A_3	B_3	B_4
2	1.128	0.8525	0.7979	1.880	—	3.267	2.659	—	3.267
3	1.693	0.8884	0.8862	1.023	—	2.574	1.954	—	2.568
4	2.059	0.8798	0.9213	0.729	—	2.282	1.628	—	2.266
5	2.326	0.8798	0.9400	0.577	—	2.114	1.427	—	2.089
6	2.534	0.8480	0.9515	0.483	—	2.004	1.287	0.030	1.970
7	2.704	0.8332	0.9594	0.419	0.076	1.924	1.182	0.118	1.882
8	2.847	0.8198	0.9650	0.373	0.136	1.864	1.099	0.185	1.815
9	2.970	0.8078	0.9693	0.337	0.184	1.816	1.032	0.239	1.761
10	3.078	0.7971	0.9727	0.308	0.223	1.777	0.975	0.284	1.716
11	3.173	0.7873	0.9754	0.285	0.256	1.744	0.927	0.321	1.679
12	3.258	0.7785	0.9776	0.266	0.283	1.717	0.886	0.354	1.646
13	3.336	0.7704	0.9794	0.249	0.307	1.693	0.850	0.382	1.618
14	3.407	0.7630	0.9810	0.235	0.328	1.672	0.817	0.406	1.594
15	3.472	0.7562	0.9823	0.223	0.347	1.653	0.789	0.428	1.572
16	3.532	0.7499	0.9835	0.212	0.363	1.637	0.763	0.448	1.552
17	3.588	0.7441	0.9845	0.203	0.378	1.662	0.739	0.466	1.534
18	3.640	0.7386	0.9854	0.194	0.391	1.607	0.718	0.482	1.518
19	3.689	0.7335	0.9862	0.187	0.403	1.597	0.698	0.497	1.503
20	3.735	0.7287	0.9869	0.180	0.415	1.585	0.680	0.510	1.490
21	3.778	0.7272	0.9876	0.173	0.425	1.575	0.663	0.523	1.477
22	3.819	0.7199	0.9882	0.167	0.434	1.566	0.647	0.534	1.466
23	3.858	0.1759	0.9887	0.162	0.443	1.557	0.633	0.545	1.455
24	3.895	0.7121	0.9892	0.157	0.451	1.548	0.619	0.555	1.445
25	3.931	0.7084	0.9896	0.153	0.459	1.541	0.606	0.565	1.435

Note: Reprinted from Statistical Methods for Six Sigma (Appendix G, pp. 311), by Anand M. Joglekar (JOGLEKAR, 2003). Retrieved from http://www.bessegato.com.br/UFJF/resources/table_of_control_chart_constants.pdf