ANALYSIS OF TRAFFIC NETWORK NEAR BUS STOPS USING BUS GPS DATA

by

Yiğit Çetinel

B.S., Civil Engineering, Boğaziçi University, 2014

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Civil Engineering

Boğaziçi University

2017

ANALYSIS OF TRAFFIC NETWORK NEAR BUS STOPS
USING BUS GPS DATA

APPROVED BY:

Assist. Prof. Ilgın Gökaşar . . . . . .
(Thesis Supervisor)

Assist. Prof. Mustafa Gökçe Baydoğan . . . . . .

Assist. Prof. Gürkan Günay . . . . . .

DATE OF APPROVAL:

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and deepest appreciation to my thesis supervisor Assist. Prof. Ilgın Gökasar for her invaluable guidance, innovative suggestions, continued encouragement, remarkable patience and enthusiasm throughout this study and each step of my graduate education.

I wish to thank Assist. Prof. Mustafa Gökçe Baydoğan for his unconditional and valuable guidance for this thesis. I would like to thank to the member of the thesis committee, Assoc. Prof. Gürkan Günay for his valuable comments and precious suggestions.

This project is supported by the Bogaziçi University Research Fund with the project number 11660 and the project code 16A04P2. I also would like to thank Istanbul Electricity, Tramway and Tunnel General Management (İstanbul Elektrik Tramvay ve Tünel İşletmeleri Genel Müdürlüğü) and Department of Traffic in Istanbul Metropolitian Municipality (Istanbul Büyüksehir Belediyesi Ulasım Daire Baskanlıgı Trafik Müdürlügü) for providing valuable information, bus GPS data and İstanbulkart data throughout the project.

Deliverables of this thesis include two international conference paper, one national conference paper and two journal paper in review.

I am indebted to my parents for the endless support, encouragement and love they have given. I cannot forget the support of my dear mother and father.

ABSTRACT

ANALYSIS OF TRAFFIC NETWORK NEAR BUS STOPS USING BUS GPS DATA

In congested cities where the commuting time doubles during peak hours, it is crucial to identify every network problem. While it is costly to implement classic sensor technologies that are used for identification all around the network, using trajectory data of GPS (Global Positioning System) equipped bus fleet is suggested in literature. In this study, it is aimed to analyze the traffic behavior of the buses around the routes of these buses, and to express the effects of the selected parameters on the buses near bus stops numerically using the bus GPS data in Istanbul (IETT). The data consist of more than 5000 daily trajectory log files, including more than 25 million rows of location and time information during April 2016, of buses working in 12 bus routes which are selected for the most variety. The influence distance, wherein the buses affect traffic network while slowing down around the bus stops, are measured for each bus stop by using Fused Lasso method on the speed patterns of buses along the bus routes. Possible interruptions and the correlation of their distances to the bus stops with the influence distances are investigated for the surrounding network of 438 bus stops. M5P, random forest and extremely randomized trees models are created to predict the influence distances using the bus stop parameters. The models show that, although the passenger demand plays huge role on the influence distances of the bus stops, the other parameters such as change in the number of lanes and location of the traffic lights should be used to predict the influence distances. The influence distance for the bus stops varies from 36 to 174 meters, with an average value of 98 meters.

ÖZET

OTOBÜS GPS VERİLERİ İLE DURAK ÇEVRESİNDEKİ TRAFİK AĞININ ANALİZİ

Zirve saatlerdeki trafiğin, işe geliş gidiş zamanlarını iki katına çıkarabildiği, İstanbul gibi, nüfus yoğunluğu yüksek şehirlerde, ulaşım ağındaki en ufak sorunların bile tespit edilmesi büyük önem teşkil etmektedir. Bu sorunları saptamak için kullanılan sabit trafik sensörlerinin tüm ağa kurulmasının maliyeti yüksek olabilir. GPS (küresel konumlama sistemi) takılmış olan otobüs filolarının güzergâh bilgilerinin, trafik analizinde kullanımının örnekleri son zamanlarda literatürde yer almaktadır. Bu çalışmada, İstanbul'daki otobüs (İETT) GPS verilerini kullanarak, otobüslerin güzergâhlarındaki trafik davranışlarını analiz etmek ve seçilen parametrelerin otobüs durağına yakın otobüslere etkilerini sayısal olarak göstermek amaçlanmıştır. Otobüs verisi, çeşitlilik hedef alınarak seçilen 12 otobüs rotasında çalışan otobüslerin, Nisan 2016 boyunca kaydettiği 25 milyondan fazla yer ve zaman bilgisi satırı içeren 5000'den fazla günlük yörünge kaydından oluşmaktadır. Otobüslerin, durak çevrelerinde yavaşlarken trafik ağını etkilediği durak etki mesafeleri, rotaları boyunca kaydedilmiş hız verilerinin üzerine Fused Lasso yöntemi kullanılarak, her otobüs durağı için hesaplanmıştır. Muhtemel etki faktörleri ve bunların durağa olan mesafeleri ile otobüs duraklarının etki mesafelerinin korelasyonu, 438 otobüs durağının çevresindeki ağda araştırılmıştır. Otobüs durağı parametrelerini kullanarak etki mesafelerini tahmin etmek için M5P, random forest ve extremely randomized trees modelleri oluşturulmuştur. Oluşturulan modeller, yolcu talebinin otobüs duraklarının etki mesafeleri üzerinde büyük rol oynamasına rağmen, şerit sayısındaki değişiklikler ve trafik ışıklarının konumu gibi diğer parametrelerin de etki mesafelerini tahmin etmek için kullanılması gerektiğini ortaya çıkarmıştır. Otobüs duraklarının etki mesafeleri 36 metreden 174 metreye değişkenlik gösterirken, ortalaması 98 metre olarak hesaplanmıştır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

ABSTRACT iv

ÖZET v

LIST OF FIGURES viii

LIST OF TABLES xii

1. INTRODUCTION AND BACKGROUND 1

1.1 Problem Statement 3

1.2 The Goal and Objectives 3

1.3 Literature Review 4

1.4 Thesis Outline 7

2. THEORY 8

2.1 Fused Lasso 9

2.2 Hierarchical Clustering 10

2.2.1 Linkage Criteria 11

2.2.2 Similarity Metrics 12

2.3 Decision Trees 13

2.3.1 ID3 17

2.3.2 C4.5 17

2.3.3 CART 17

2.3.4 M5 model tree 18

2.3.5 M5P model tree 20

2.3.6 Random Forest 21

2.3.7 ExtraTrees 23
3. METHODOLOGY 24
3.1 Data Storing Process 26
3.2 Bus Route 32
3.3 Map Matching 37
3.4 Calculation of Speeds 38
3.5 Detection of Hotspots 41
3.6 Detection of Stopping of Buses 43
3.7 Calculation of Dwell Times 44
3.8 Preparation of Bus Stop Characteristics Data 46
3.9 Bus Stop Transit Card Data 47
3.10 Bus Stop Influence Distance Measurement 48
4. ANALYSIS AND RESULTS 53
4.1 Hotspot Detection 53
4.1.1 Bus stop effects on stopping points of buses 56
4.1.2 Number of lane change effects on stopping points of buses 57
4.2 Dwell Pattern Clustering 58
4.3 Bus Stop Clustering 63
4.4 Bus Stop Influence Distance 66
4.4.1 M5P model results 75
4.4.2 Random forest model results 77
4.4.3 Extremely randomized trees model results 80
5. CONCLUSION 85
REFERENCES 88

LIST OF FIGURES

Figure 1.1. Mode share of Istanbul public transportation and public road transportation 2

Figure 2.1. Fused lasso applied to the speed of a sample bus GPS data 10

Figure 2.2. A dendrogram example 11

Figure 2.3. An example tree structure 14

Figure 2.4. An example M5 model tree for bus stop attributes data 19

Figure 3.1. The selected bus routes 24

Figure 3.2. The flowchart of the methodology 26

Figure 3.3. Trajectory of 5000 data points 30

Figure 3.4. R code that imports data files into SQL Server with the odbc driver 30

Figure 3.5. Creating database table “Geopoint” with geography point 31

Figure 3.6. R code querying all buses working on 30A route to get all data points 31

Figure 3.7. A query asking public transportation stops within 200-meter-circle from the selected location in CitySDK API 33

Figure 3.8. A query asking public transportation lines containing “Cengiz Topel” bus stop in CitySDK API 33

Figure 3.9. A query asking public transportation lines has a short name "DT2" in CitySDK API 34

Figure 3.10. Three million observations in Kabataş district that contain high level of noisy data 37

Figure 3.11. Before and after route fixing for long roads (above) and curvy roads (below)..... 39

Figure 3.12. Example code of using "distGeo" function 41

Figure 3.13. Number of data points buses log in 1 kilometer changing with bus speed 42

Figure 3.14. Data points of a sample route and its density map..... 43

Figure 3.15. One second bus stopping simulation to calculate the threshold speed Vs 44

Figure 3.16. Stopping duration and hour and dwelling pattern 46

Figure 3.17. The first (a) and the last (b) 12 columns of the Excel file where the measurements of the bus stops are recorded 47

Figure 3.18. Data Density (nodes vs time) along the bus route 49

Figure 3.19. Every 5 node numbers (green) and the bus stops (blue) of U1 route..... 49

Figure 3.20. (a) Space time speeds and location of bus stops as vertical blue lines (b) Space time speeds after fixed with interpolation..... 50

Figure 3.21. Off-peak average speed distribution along the selected route filtered with one dimensional generalized fused lasso method 51

Figure 3.22. The split points where speed data have a dramatic change before (vertical dashed blue line) and after (vertical dashed red line) the bus stop location 52

Figure 4.1. Number of lanes, bus stops and stopping points (bottlenecks) on Barbaros and Nispetiye Boulevards 54

Figure 4.2. Stopping points on nodes of 43R route during peak, off-peak and cumulative time windows 55

Figure 4.3. Number of lanes of route lines and stopping points during peak and off-peak hours with corresponding node points 57

Figure 4.4. The selected bus route (Beşiktaş-Mecidiyeköy) and its bus stops 59

Figure 4.5. (a) Dwelling patterns of bus stops on selected route (b) Dwell patterns of outlier bus stops..... 61

Figure 4.6. Close up to dwell patterns after extracting outlier bus stops..... 62

Figure 4.7. Bus dwell patterns cluster dendrogram..... 63

Figure 4.8. Bus dwell patterns cluster and characteristic cluster comparison 65

Figure 4.9. The relationship of the number of lanes and pocket of the bus stops..... 67

Figure 4.10. Demands of the bus stops in Besiktas area 68

Figure 4.11. Boxplots of monthly passenger demand vs the number of lanes . 68

Figure 4.12. Interruptions before and after the bus stops: (a) Traffic lights, (b) Crossings, (c) Drops in the number of lanes, (d) Increase of the number of lanes, (e) Roundabouts, (f) Significant entries..... 70

Figure 4.13. Histogram of influence distances of bus stops 71

Figure 4.14. Boxplots of influence distance of bus stops grouped by (a) number of lanes and (b) pocket..... 71

Figure 4.15. Influence distance with demand counts on logarithmic scale 72

Figure 4.16. Distance to the nearest (a, b) entrances and (c, d) traffic lights, and the influence distances..... 73

Figure 4.17. Relationship of the other interrupts: (a, b) Crossing, (c, d) Roundabout, (e, f) Drops in the number of lanes, (g, h) Increase of the number of lanes..... 74

Figure 4.18. The log file of M5P model from WEKA 76

LIST OF TABLES

Table 3.1. The “Route” table in the database, which has data file information. 28

Table 3.2. The “Geography” table in the database, which has trajectory information. 29

Table 3.3. The bus stops of “DT2” bus route. 35

Table 3.4. Speed calculation example 40

Table 3.5. Stopping duration calculation. 45

Table 4.1. Bus stop attributes. 64

Table 4.2. The distribution of bus stops according to the number of lanes 66

Table 4.3. Cross-validation summary of the M5P model 75

Table 4.4. Random forest optimization table for “binary” dataset 78

Table 4.5. Random forest optimization table for “300” dataset 78

Table 4.6. Random forest optimization table for “500” dataset 79

Table 4.7. Random forest optimization table for “1000” dataset 79

Table 4.8. Top 10 variable importance measures of two Random forest models 80

Table 4.9. ExtraTrees optimization table for “300” dataset. 81

Table 4.10. ExtraTrees optimization table for “500” dataset. 81

Table 4.11. ExtraTrees optimization table for “1000” dataset. 82

Table 4.12. ExtraTrees optimization table for “binary” dataset. 83

Table 4.13. Top 10 variable importance measures of two ExtraTrees models 84

Table 4.14. Model summaries 84

1. INTRODUCTION AND BACKGROUND

Traffic congestion in a city with large population can have severe economic and social impact on both the community and the environment. A traffic network with many highly congested regions in traffic network will cause the travelers to complete their trip with a significant delay that can cause stress and boredom [1]. The issue is not experienced by private vehicle users only but also by public transportation passengers as well. A solution, for resolving the regions with dense traffic, is to increase the number of alternative modes and especially promoting public transportation.

Istanbul, the study area, is one of the cities that has one of the most congested traffic in the world [2]. With a population over 14 million [3] and nearly 4 million vehicles in traffic, especially in the peak hours, the traffic comes close to a full stop [4]. For this problem to be resolved proper transportation infrastructure must be provided and public transportation must be promoted. Identifying and analyzing problematic spots in the public transportation bus network is important for both the passengers and the city authorities.

There are three main public transportation modes in Istanbul, road, rail and sea transportation. Road transportation includes bus, bus rapid transit, minibus, private service and minibus taxi. Rail transportation composes of metro, light rail train, tram, Marmaray, funicular and cable car. Sea transportation contains ferry and private sea boat. Ride shares of public transportation are presented in Figure 1.1. Daily 13 million passengers are carried with public transportation in Istanbul in 2016 [5]. The biggest pie is related to road transport with 78%, and 37% of road transport is related to bus transportation.

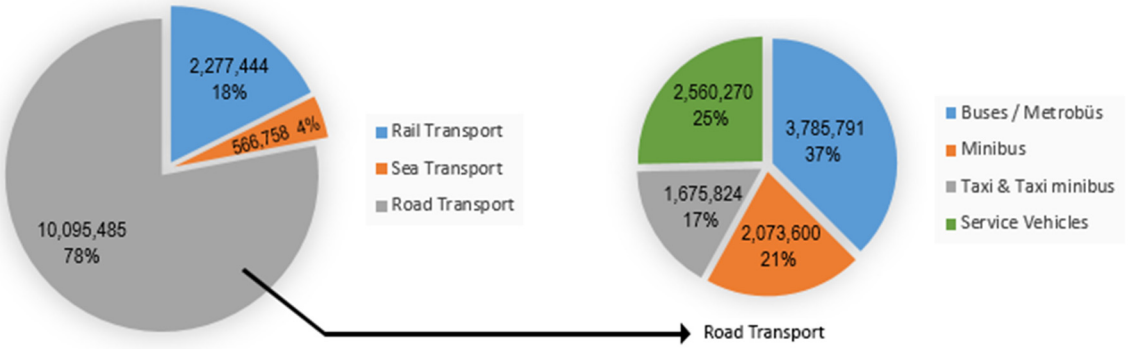

Figure 1.1. Mode share of Istanbul public transportation and public road transportation.

This study focuses on the effect of an influence distance of a bus stop on an arriving or departing bus. The influence distance lies between the points where the bus starts to decelerate while arriving at the bus stop and starts to recover its speed until it is uniform with the traffic flow. It can also be said that the influence distance is expected to be correlated with the hotspots before and after the bus stops, due to their impact on the arriving or departing busses. The detection of influence distance is helpful in terms of detecting bus stops which requires special attention. This is an important information for classifying bus stops and detecting any potential problems at any bus stop. Combining this information with the other characteristic components for evaluating the performance of the bus stop and classifying them, such as dwell time, punctuation and so on, will provide a different perspective for this matter. The detection of these uncommon bus stops makes the public transportation planners to adjust their performance criteria accordingly, such as punctuality threshold and so on. The obtained results, which are the main contribution to the literature, will benefit both the public transportation planners, policy makers and bus users in the future.

1.1. Problem Statement

Traffic congestion is a severe problem of the big cities beside harmful effects on the environment. The most important one of the modes which have a significant role in traffic jam problem is considered as private car. For this situation, one of the most recommended solutions is to direct private car users towards public transportation. However, even when public transportation modes are widely being used, problematic spots in the road network, which can be found in many cities, can still cause a significant traffic congestion. These spots in the network decreases capacity significantly; therefore, it is important to analyze them in detail.

1.2. The Goal and Objectives

Bus GPS data can be useful in detecting problems on the network in order to provide better service for public transportation users. Therefore, it is aimed to extract information regarding the problematic spots experienced by buses from the GPS data. To achieve this goal, the following research objectives are aimed:

(i) To detect the hotspots experienced by the buses along the bus routes in order to find out possible parameters that have effect on the speed distribution of buses,

(ii) To extract dwell time information of bus stops from GPS data in order to correlate them with the network information near bus stops,

(iii) To provide an influence distance measure for the bus stops in order to determine which bus stops have more influence on the speed distributions of buses near the bus stops.

(iv) To create models to predict the influence distance of the bus stops using the network information near bus stops.

1.3. Literature Review

Determining hotspots concerning the public transportation buses and dwell times at a bus stop are both important information for evaluating the transit reliability which is an important factor for travelers in making decisions, especially in urban areas [6]. Using GPS data from the buses is the most common method for achieving these objectives but can create some difficulties such as separating movement and dwelling phases of the data obtained from vehicle tracking system [7].

Even then it is also crucial to have the knowledge to maintain and manage large quantities of GPS data. An alternative method for determining dwelling times and patterns is to use the characteristics of the bus stops, which requires much less information than using a large chunk of GPS data extracted from busses [8].

ITS technologies and methods have been discovered many different issues and their causality. These methods can be crucial for the improvement of all modes of transportation in a city like Istanbul, if they are implemented properly. Studies for sensor technologies have been made in order to improve the transportation infrastructure. Over recent decades, new methods for modelling, estimating and controlling traffic by using wireless sensor inputs have been presented [9, 10, 11, 12]. This type of technology has the capability of building a cognitive network in the near future, so that information collected from various type of transportation networks can be integrated and be used to benefit the drivers, creating an autonomous environment, benefitting the entire network [13]. Currently not all types of data can be used for all kinds of purposes. Generally, traffic data obtained from private vehicles or trucks are more suitable for motorways and rural areas, while, in

case of urban traffic, taxi and bus fleets are particularly useful due to their high number and homogeneous spread in the urban area [14].

Studies where private vehicles are used as probes, gives information about the traffic state of a segment of a network. In a study, a method is proposed where even a single vehicle is sufficient to determine the state of the traffic, which is incredibly useful in the sense that the traffic environment is not affected significantly while information is collected [15]. In the same way while studying on a method for determining the expected queue length and its variance, it was discovered that last probe vehicle is sufficient to estimate these values [16]. Similar to private vehicles, there are studies for traffic estimation by using taxis as probes. An urban traffic estimation method is presented using a huge amount of GPS data from over 5000 taxis that are within the city and with it, density levels are determined and a method for automatically determining the capacity of roads is introduced [17]. While traffic states can be estimated from the data provided from taxis and private vehicles, data from public busses cannot be used for the same purpose due to the behavioral difference of the private vehicles and taxis compared to public busses. Therefore, combining this information with developed methods will provide a much broader understanding of the overall network.

Bus stops are the most important component for the reason of the behavioral difference of public busses compared to other vehicles. The bus stops are points where the bus will definitely have an interaction with the bus stop, whether it will stop completely or slow down because of the interaction. Providing accurate information for arrival time of a bus to a bus stop and minimization of dwell time at a bus stop are important performance criteria for public bus mode. All these components will influence the most important value for a public bus system, which

is the travel time. There are many different studies, which makes the public transportation mode a more attractive mode and makes the system to function even more efficiently.

Where there is a bus stop, there will be also an influence on the traffic flow near that stop. The road capacity for cars near any stop is a function of flow rates of various streams and the dwell time of the busses [18]. Bus bays can be used if there is sufficient space, in order to decrease the influence of the bus stop on the traffic flow and it is advised to provide them where a 25% drop of general traffic speed or even more is caused [19]. It is also crucial to determine the locations of these bus stops and have the optimal space between them. A value important for determining this distance is the total cost sensitivity to various parameters such as value of users' time, access speed, demand density and so on [20]. By adequate planning of stop spacing and number of required busses for a sustainable level of service the operating cost can be decreased as well [21]. An important factor for level of service is the dwell time, which can influence arrival time of a bus to a bus stop and travel time of a bus route. Detecting these value accordingly and minimizing them as well, encourages more and more citizens to use public transportation.

Important parameters for dwell time consists of passenger activity, lift operations, time of day, route type and so on [22]. As the dwell time increases the average traffic speed decreases which causes increased travel times and delays [19]. So minimizing this dwell time as much as possible is an important step towards normalizing the impact of a bus stop to the traffic flow. More details about the causes of increase of dwell time are discovered such as a friction effect, which occurs when two queues of passengers try to board the bus through a single door, crowding effect, which is caused by passengers standing inside the bus and even compared the

performance of payment methods to determine the impacts of these factors on the dwell time [23]. With increasing dwell time the delay of arriving to the next stop will increase thus the total travel time will also increase. As the number of busses increase at a given bus stop with high dwell time values, it is expected to observe hotspots close to these bus stations.

While determining these hotspots, it should be noted that the hotspots valid for public transportation busses, may not be experienced by private vehicles, which can also be possible for other way around. Nevertheless there may be an unintentional interaction due to these problematic spots for both the private vehicle and public transportation busses, which must be prevented or at least minimized. If necessary precautions are not applied, public transportation busses can experience even further delays and congestion, which will make, the most eligible solution against traffic congestion, an unattractive mode of transportation.

1.4. Thesis Outline

The remainder of this thesis is organized as follows: In the next section, the background and theory of the methods which are used in this study are discussed. Then, in Chapter 3, the methodology of every step of this study is explained with examples as well as usage of the methods starting from how to obtain and store the data to how to extract information from the data. It is followed by the data analysis, the results of the analysis, and comparison of the model performances to predict influence distances of the bus stops in Chapter 4. Finally, conclusions and recommendations are provided in Chapter 5.

2. THEORY

In this thesis, some data mining and machine learning methods such as clustering and tree algorithms, namely fused lasso, hierarchical clustering, and decision trees, are used to detect the problems which buses experience along their routes daily.

In order to determine the disparity in the speed distribution of buses near bus stops, one dimensional adaptation of fused lasso is used. The drop in speed is used to calculate influence distances of bus stops. The fused lasso automatically separates the regular speed values and decreased speed values in a speed pattern.

Then, hierarchical clustering is explained because it is used to group bus stops by their properties. Same bus stops are clustered again by their dwell times. The comparison of group members shows that both groups have similar members, which shows a correlation between bus stop characteristics and dwell times.

The background of decision trees are explained because all methods (M5P, Random Forest and ExtraTrees) are improved versions of decision trees. In M5P, a regression model is implemented on a decision tree. Random Forest and ExtraTrees models ensemble multiple trees together to improve predictions. These three methods are used to predict the influence distance values using the characteristics of the bus stops.

2.1. Fused Lasso

The fused lasso is a method for spatial or temporal data structure to extract data patterns separated with points where there is large changes in data. In a prediction problem, the fused lasso objective function is shown in Equation 2.1.

$$\min_{\beta}\left\{\frac{1}{N}\sum_{i=1}^{N}(y_i - x_i^t\beta)^2\right\}$$
$$subject\ to \sum_{j=1}^{p}|\beta_j| \le s_1 \ \ and \ \ \sum_{j=2}^{p}|\beta_j - \beta_{j-1}| \le s_2 \tag{2.1}$$

where there is N cases having outcomes y from i to N, and features x from j to p. The first constraint encourages sparsity in the coefficients; the second encourages sparsity in their differences, i.e. flatness of the coefficient profiles β_j as a function of j.

Suppose that y has a 1-dimensional data that is the coordinates of y corresponding to successive positions. The second constraint, s_2 of Equation 2.1 penalizes the absolute differences in adjacent coordinates of β, and is known as the 1-d fused lasso [24]. This gives a piecewise constant fit. It is used in settings where coordinates in the true model are closely related to their neighbors. Figure 2.1 shows an example of the 1-d fused lasso applied to the speed distribution of a sample bus GPS data. The red line represents the 1-d fused lasso result which is the flatten speed distribution. The similar value trends are grouped with the same flatten value separated at the significant changes.

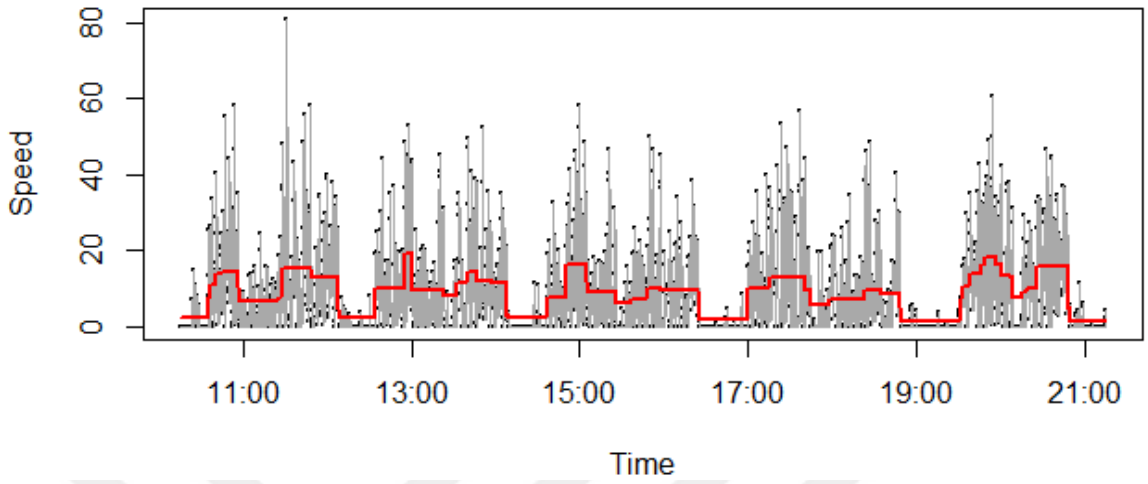

Figure 2.1. Fused lasso applied to the speed of a sample bus GPS data

2.2. Hierarchical Clustering

Clustering algorithms are used to find groups in a data set that share a common pattern. The algorithms can automatically find clusters in an unsupervised manner which means that the groups can be created without any predetermined group or human inspection. The grouping continues until the data within the same cluster are adequately similar to each other [25].

Hierarchical clustering generates a tree of clusters. The tree can be generated by splitting clusters into smaller pieces starting from the root cluster which has all the data, or by merging individual data pieces into larger clusters. A cluster becomes parent when it splits, and creates child clusters. The deeper in the cluster tree, the more similar members are in the clusters. The clusters which have same parent are more similar than the ones that have not.

The process continues until the desired numbers of clusters are generated. The farthest reached clusters for each branch are called leaves. Every cluster except

root and leaves have parent and child clusters. This generated tree is often called as dendrogram. An example dendrogram is shown in the Figure 2.2. At the top of the tree, there is a root node which consists all members. Every three edge intersection is a decision node which splits members into smaller groups.

2.2.1. Linkage Criteria

In constructing hierarchical cluster tree, the splitting approach is called divisive, the merging approach is called agglomerative. Divisive clustering starts from top to down, one cluster to multiple, agglomerative clustering is the opposite.

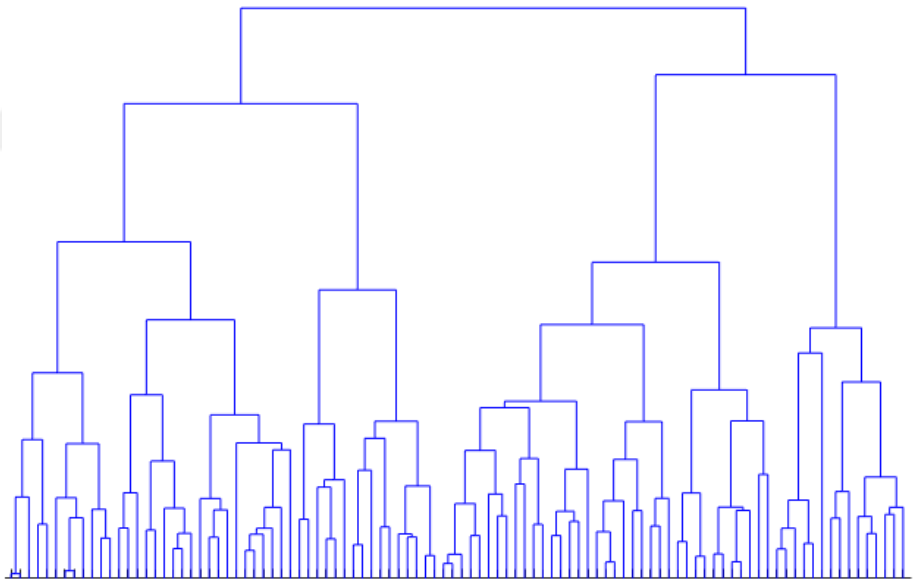

Figure 2.2. A dendrogram example

For each step of splitting or merging, similarities are calculated with linkage metrics. According to linkage metrics, the least similar data are separated the most from each other in a dendrogram. Three of most used linkage metrics are single link algorithm, complete link algorithm and group average algorithm [26].

Single Link Algorithm (SLA) measures the maximum of the pair-wise similarity from each cluster to merge a pair of clusters [27].

$$SLA(C_A, C_B) = \max_{x \in C_A, y \in C_B} \text{dist}(x, y) \tag{2.2}$$

where C_A and C_B are clusters having nodes x and y, dist is the chosen similarity metric.

Complete Link Algorithm (CLA) measures the minimum of the pair-wise similarity from each cluster [28].

$$CLA(C_A, C_B) = \min_{x \in C_A, y \in C_B} \text{dist}(x, y) \tag{2.3}$$

Group Average Algorithm (unweighted pair group method with arithmetic mean (UPGMA)) measures the average of the pair-wise similarity of the documents from each cluster [29].

$$GMA(C_A, C_B) = \frac{1}{n_i n_j} \sum_{x \in C_A, y \in C_B} \text{dist}(x, y) \tag{2.4}$$

2.2.2. Similarity Metrics

The Euclidean distance and Cosine similarity measure are the most popular similarity functions which are used to find the proximity between the clusters [30]. Euclidean distance measures the distance between two datasets or a dataset and a centroid projected in the Euclidean space. Euclidean distance is represented as D in the Equation 2.5 where N is the total number of terms in the dataset.

$$D(x,y) = \sqrt{\sum_{i=1\ldots N} (x_i - y_i)^2} \tag{2.5}$$

Cosine similarity can be used to find the similarity between two datasets, shown in Equation 2.6.

$$Cos(x,y) = \frac{x^t y}{\|x\| \, \|y\|} \tag{2.6}$$

where x and y are vectors, and $\|x\|$ indicates the magnitude of the vector x. If two datasets x and y are similar then the cosine similarity measure will be closer to 1 otherwise it will be closer to 0.

Although there are many other linkage metrics, Equation 2.2, 2.3, and 2.4 are commonly used. These methods use similarity measures like Euclidean distance or Cosine similarity matrix for their comparisons and measurements.

2.3. Decision Trees

Decision tree is a type of classifier which is used in supervised-learning tasks which means the data have already been labeled or the target data exist so that the models can be trained. Models can be built with supervised-learning when the training dataset is labeled. Classifier algorithms are used to separate data rows into some groups with labels which are called classes. Trees are special form of graph structures which are collections of points and lines called “nodes” and “edges” in mathematical terminology. In tree structure, there is no loops and circuits, and it has only one path between any two nodes.

Decision trees are starting from a root node which contains all data rows. Each node is a decision point that separates data rows with a rule. For example, a dataset of bus stops can be split into two groups with a decision rule of whether bus stops have pockets or not, shown in Figure 2.3. In decision tree structure, there cannot be more than one rule that is set for a node. Thus, a node can be separated only once and creates two new nodes under it. When new nodes are created under the parent node, the depth that allows to track how deep the tree is, increases.

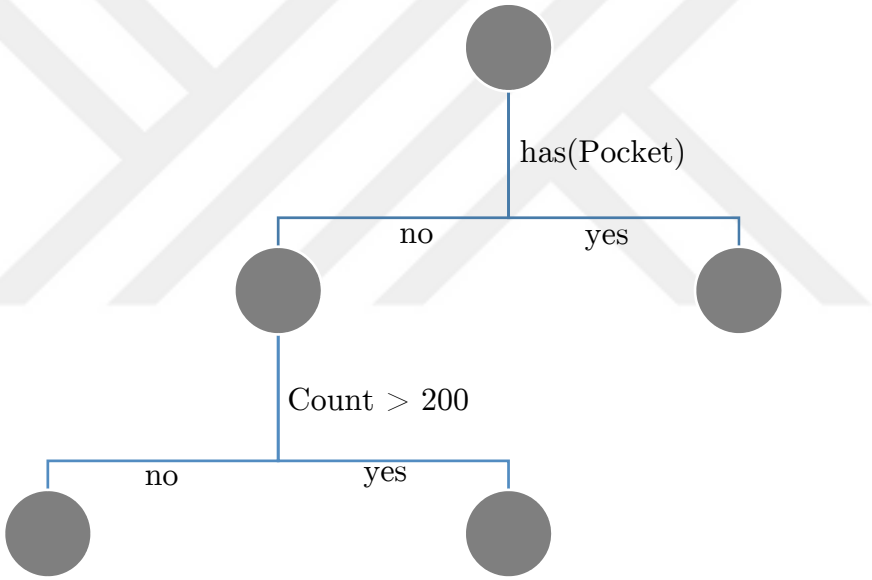

Figure 2.3. An example tree structure.

Every split of data creates a branch of tree. A branch cannot contain any loops or cannot join into another branch. Splits of data set in decision trees continue until a stopping condition like maximum depth limitation. When the split operation finishes, the terminal nodes are called as leaves.

Selection of decision rules is the key point for a good decision tree. One dataset can be represented by multiple different decision trees. In order to find the

best decision rules, some measurements can be used to optimize decision trees. The measure use given data points as training set to choose parameters of the best decision rules.

The entropy is used as a measurement in decision making on the nodes. The entropy is a measure of impurity that is the opposite of the purity that expresses how less difference or classes the data have in a node. A node is considered the purest when there is only one class of data in a node.

On the other hand, the entropy function reaches its maximum when the probability of both classes being either class1 or class2 is fifty-fifty so that it has maximum uncertainty. The probabilities of zero entropy which is the lowest value is p=1 or p=0 with complete certainty $p(X = class1) = 1$ or $p(X = class2) = 0$ respectively [31]. The definition of the entropy H(X) can be generalized for a discrete random variable X with N outcomes as

$$H(X) = -\sum_{i=1}^{N} p(x_i) \log_2 p(x_i) \tag{2.7}$$

where $\log_2 p(x_i)$ is logarithm to base 2 of the probability of x for each outcome of random variable X.

One metric which measures the degree of split is called information gain. Information gain (I) is usually used for multiclass classification problems, and it can be defined as

$$I_j = H(X_j) - \sum_{k\epsilon(L,R)} \frac{|X_j^k|}{|X|} H(X_j^k) \tag{2.8}$$

where Xj is the set of training points at node j, H(Xj) is the entropy at node j before the split, and $\left|X_j^L\right|$ and $\left|X_j^R\right|$ are the absolute values of the sets of points at the right child and left child respectively of the parent node j after the split.

Training the parameters of node j involves maximizing the information gain at the node. Each split node is associated with a binary split function that decides which child of the node each data point traverses to next in the tree.

The advantages of decision trees are basically based on their simplicity, they are easy to understand and interpret. The results of classification and regression models can be explained by binary logic. They can easily be implemented independent to the size of the data set. The decision trees can handle varying in feature value types such as numeric, categorical values.

However, the optimal decision tree building problem is NP-complete, which refers to "nondeterministic polynomial time" problem. For this reason, trees created with greedy algorithms may not produce an optimized solution from a global perspective. Another problem associated with individual decision trees is the tendency to over-fit and generalize poorly. The combination of decision trees such as random forests and extremely randomized trees or the use of different models together such as M5P alleviate the problems [32].

The three important construction algorithms of decision trees (ID3, C4.5, CART) in the background of the ensemble tree models are listed in the following sections as well as the theory of the models (M5P, Random forest, ExtraTrees).

2.3.1. ID3

Iterative Dichotomiser 3 (ID3) algorithm is one of the first decision-tree construction algorithms [33, 34, 35]. In every iteration of ID3, a previously unselected attribute having the biggest information gain is selected to split the set of data. The iterations start at the root of the tree and continues until the following stopping conditions.

- Every member of a node has the same label.
- All attributes have been previously selected.
- A subset after a split is empty.

2.3.2. C4.5

C4.5 algorithm is a decision-tree construction algorithms as improvement to ID3 [36]. The improvements made from ID3 include the ability of the algorithm to handle continuous and missing attribute values in addition to discrete attribute values in the training set. Attributes with missing values are ignored during the process of the attribute selection process that best satisfies the separation criteria. In addition, decision trees constructed with C4.5 are pruned which means that the size of the decision tree is reduced so that it prevents overfitting and improves accuracy.

2.3.3. CART

Classification and Regression Trees (CART) method [37] is another decision-tree construction algorithm similar to the C4.5 method. A decision tree is first constructed fully-grown, and then pruned sequentially back to the root. The optimal

decision tree is selected from maximum size to fully-pruned. CART trees can be optimized to perform classification or regression.

2.3.4. M5 model tree

M5 model tree is a numeric value estimator for given instances [38]. The output feature needs to be numeric for the algorithm, whereas the input features can be either discrete or continuous. Until a final leaf node is reached for a given instance, the model tree is traversed from top to bottom. At each node a decision is made based on a test condition for a feature assigned to that node. After the decision is made that path is followed. At each leaf node a linear regression model is assigned of the following form;

$$y = \beta_0 + \beta_1 x_1 + \ldots . + \beta_k x_k \tag{2.9}$$

where there is k number of regression coefficients β_i, and β_0 is the offset term. In the regression model, y is the objective value, and there is k number of explanatory variables x_i.

By using standard regression, weights $w_0, w_1, \ldots, w_k$ of input features are calculated based on some of the input features $a_1, a_2, \ldots, a_k$. The fact that each leaf node is containing a linear regression model for obtaining an estimation output is the reason why this tree is being called a model tree. When the M5 algorithm is applied on the bus stops example, a model tree with the form shown in Figure 2.4 will be generated. This sample model is split by three condition; those are passenger demand, number of lanes and distance to intersection. Then, four linear models are fitted for each created leaf.

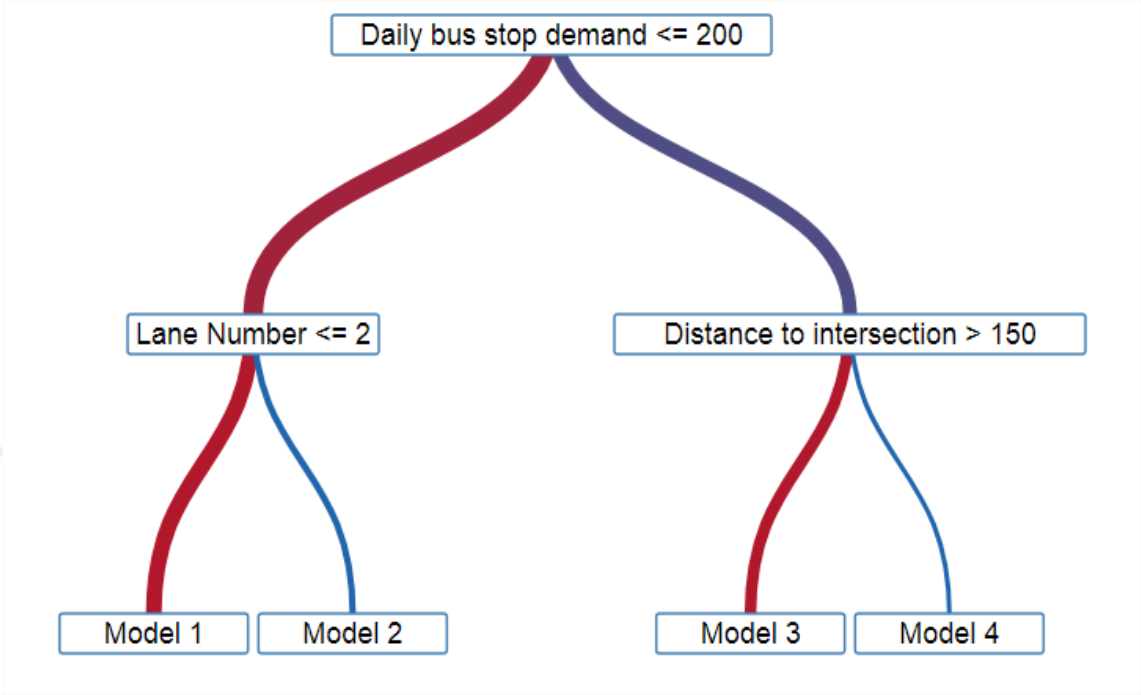

Figure 2.4. An example M5 model tree for bus stop attributes data.

A set of training instances is used initially to build the model tree using the algorithm. Divide and conquer method is used for this purpose. Starting with the root node, each node reached with an instance set is assigned as a leaf or a test condition. Subsets are formed based on the test outcome. A test is used to decide which path to follow. These tests are based on features value and there are multiple potential tests which can be used at a given node. For M5, the test that maximizes error reduction is selected, which is found using the following formula where S is the set of instance passed to the node, stdev(s) is the standard deviation;

$$\Delta error = stdev(S) - \sum_{i} \left(\frac{|S_i|}{|S|} stdev(S_i) \right) \tag{2.10}$$

where there is N number of splits, and S_i is the subset of S obtained from the split at the node with ith outcome of the test. The process of creation of new nodes is

repeated until there are very few instances to proceed any further or the variation of the obtained values in the instances that reach the node is small.

A linear model, a regression equation, is constructed at each node after the construction of the model tree has been completed. The used features in the equation are the ones that are tested or are used in linear models that are in the sub-trees below that node. The features tested above that node are not being used in the equations due to the fact that their impact of estimating the output has already been obtained at the nodes above. By eliminating more and more features in a built linear model, it is being simplified even further. The error is defined as the absolute difference between the output value estimated by the model and the actual output value observed from a given instance. The features are removed if they lead to a reduction in the overall error.

2.3.5. M5P model tree

The M5P or M5Prime algorithm is an improvement of M5 model which is a regression-based decision tree algorithm [39]. M5P is based on M5 with some additions. As the trees can become too much complex, they must be pruned in order to make it simpler without the loss of base functionality. The error for each linear model is calculated at each node starting from the bottom. If the obtained error value for the linear model at a node is less than the model sub-tree below, then the sub-tree for that node is pruned.

In case of missing values in training instances, M5P changes the expected error reduction to the formula shown in Equation 2.11,

$$\Delta error = \frac{m}{|S|} \times \beta(i) \times \left[stdev(S) - \sum_i \left(\frac{|S_i|}{|S|} stdev(S_i) \right) \right] \quad (2.11)$$

where m is the number of instances without missing values for that feature, S is the set of instances at that node, β(i) is the factor multiplied in case of discrete attributes and j takes values L and R with S_L and S_R being the sets obtained from splitting at that attribute.

2.3.6. Random Forest

Random Forest algorithm [40] is an ensemble method which performs better than individual decision trees. Random Forest can be used for both classification and regression. In Random Forest, multiple decision trees are grown independently and can be grown in parallel. Each tree is built using all the training samples which are sampled with replacement.

Random Forest can be used to optimize decision parameters of each tree. In training process, nodes can only access randomly chosen features as a feature subset. At each node, a decision feature is selected from the feature subset, which best splits the data set. None of the trees is pruned, in other words, all trees are fully grown.

In testing process of classification, all decision trees in the forest classify the test data. The most frequently selected labels are used for predictions of the final model. In testing process of regression, average values of each tree are used for predictions of the final model.

Random forests differ from other ensemble classification methods such as bagging and boosting. Bagging, an acronym for “bootstrap aggregating”, is a method

of creating multiple versions of a decision tree predictor, which are used in aggregation to form a decision by consensus [41]. Each decision tree is constructed using bootstrap replicates of the training set, that is, each training set is constructed by sampling uniformly from the original training set with replacement. The concept of boosting, introduced by Friedman [42], involves the sequential construction of trees, where the structure of a decision tree depends on the structure of previously-built trees. Random Forest algorithm can be listed using the following steps [43].

(i) Let C be a training set $\{C_1, ..., C_n\}$ with $C_i \equiv (x_i, y_i)$ which is an independent test case.

(ii) Sample the training set C with replacement to generate bootstrap resamples $B_{1...M}$.

(iii) For each resample B_m, m = 1, ..., M, grow a classification or regression tree T_m, except for the following modifications. At each split, only randomly selected predictors are considered. Let p indicate the total number of predictor variables in C. Breiman suggested to use $(p/3)$, which is the default value in the R package randomForest. Each tree is grown until all nodes contain observations no more than the maximal terminal node size, MTN, a pre-specified parameter. Unlike CART, trees in RFs are not pruned.

(iv) For predicting the test case C_0 with covariate x_0, the predicted value by the whole RF is obtained by combining the results given by individual trees. Let the RF prediction equation is

$$\begin{cases} \frac{1}{M}\sum_{m=1}^{M} \hat{f}_m^*(x_0) & \text{for regression problems} \\ argmax_g \left\{ \sum_{m=1}^{M} I\left[\hat{f}_m^*(x_0) = g\right] \right\} & \text{for classification problems} \end{cases} \tag{2.12}$$

where C_0 is the test case with covariate x_0, $\hat{f}_m^*(x_0)$ denotes the prediction of C_0 by m^{th} tree, M is the number of bootstrap resamples, I is information gain, and g is category number.

2.3.7. ExtraTrees

ExtraTrees (Extremely randomized trees) is another method for randomized ensemble of trees [44]. The method suggest that splits are selected completely at random for both predictor variable and its cut-points. Extremely randomized trees where instead of choosing the best split among a subset of variables under search for maximum information gain, a random split is chosen. This improves the prediction accuracy.

The notoriously high variance of decision trees partly finds its origins from the high dependence of the splits with the random nature of the learning set [45]. The variance of the optimal cut-point v (in the case of ordered input variables) may indeed be very high, even for large sample sizes. In particular, the cut-point variance appears to be responsible for a significant part of the generalization error of decision trees [46].

As a way to smoothen the decision boundary, it is proposed in Extremely Randomized Trees (ETs) to combine random variable selection with random discretization thresholds [44]. Extremely randomized trees can therefore be seen as a way to transfer cut point variance from the variance term due to the learning set to the (reducible) variance term that is due to random effects.

3. METHODOLOGY

12 bus lines are selected for the most variety, which are shown on map in Figure 3.1. These bus lines include ring and straight routes in congested and non-congested regions, touristic/sea-side roads, highways, multi-lane and single-lane roads. Buses of the selected bus lines are operating on both Asian and European side, and two of them are crossing the Bosphorus.

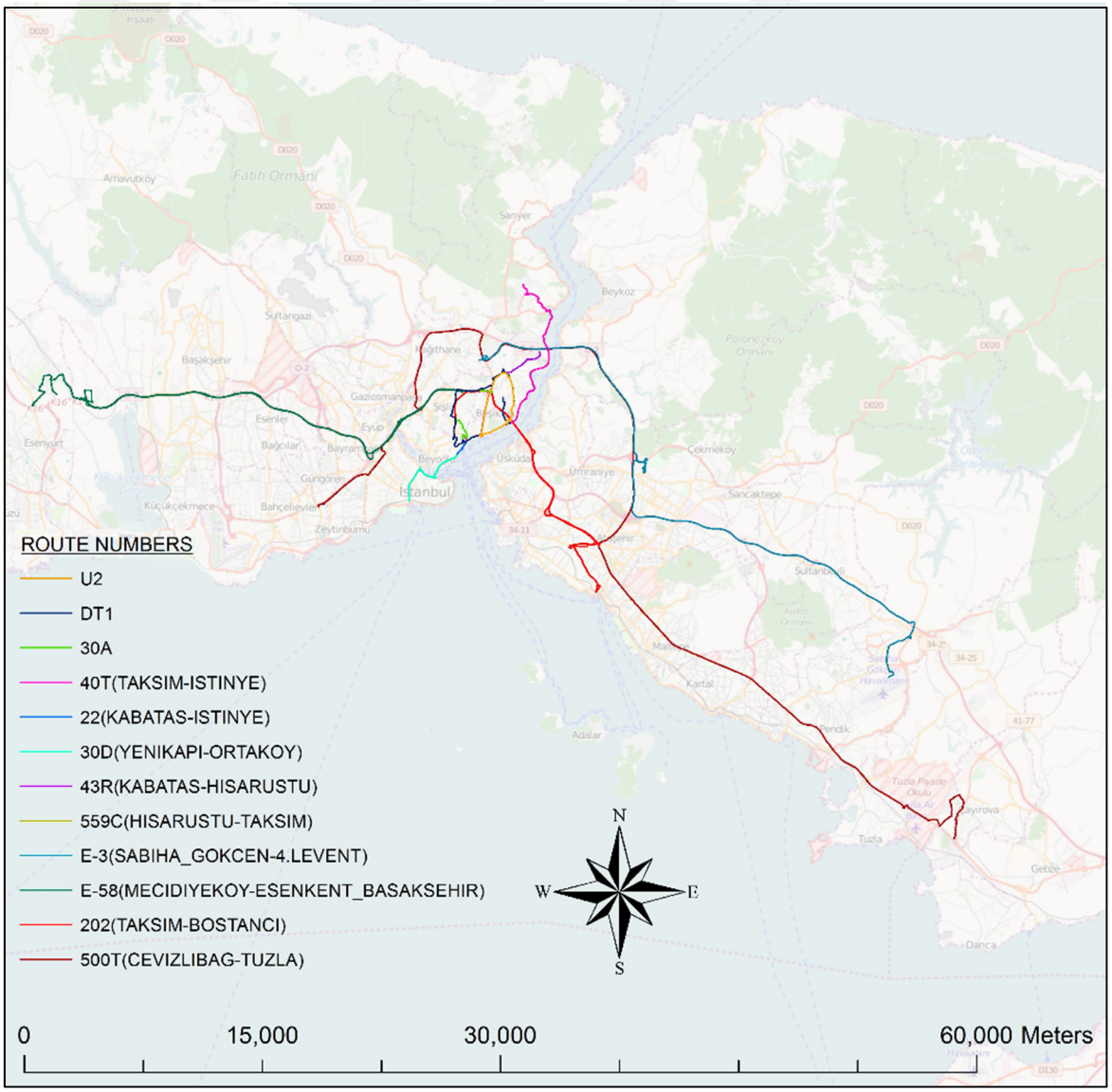

Figure 3.1. The selected bus routes

Firstly, how to obtain the data and to store it in SQL database is explained. The structure of the data is described with a sample. The processing is performed using R-stats software to import into the SQL database. Geographic features of SQL database are also explained.

The route information about the selected bus lines are obtained in the next step. For this purpose, CitySDK, open source project is investigated. The project contains digitalized version of shapes of the bus routes and bus stops. The application interface of the project allows users to download this information.

Since the GPS data are noisy, the data points are matched with the shapes of the bus routes using map matching methods. The section also gives a brief methodology about map matching and fixing the shape of the bus lines, which is necessary for smooth map matching.

After matching the data, time difference and distance of the consecutive data points are calculated using R package called “geosphere” to determine the speed information. This section includes a calculation example and R code for the package.

The next step shows the detection of hotspot using estimation of density of data points. It shows the relationship between the number of data points and the speed information, so that density of the points can be used to detect hotspots.

Then, for analyze chapter, the datasets are prepared. The dwell times are calculated with the duration of bus stoppings. The bus stop characteristics are observed and recorded using street view of the bus stops using Google Maps, Yandex Maps and Istanbul Municipality Maps. Finally, for each bus stop, using Fused Lasso

method, the influence distances for bus stops are defined and estimated. The flowchart of the methodology is shown in Figure 3.2.

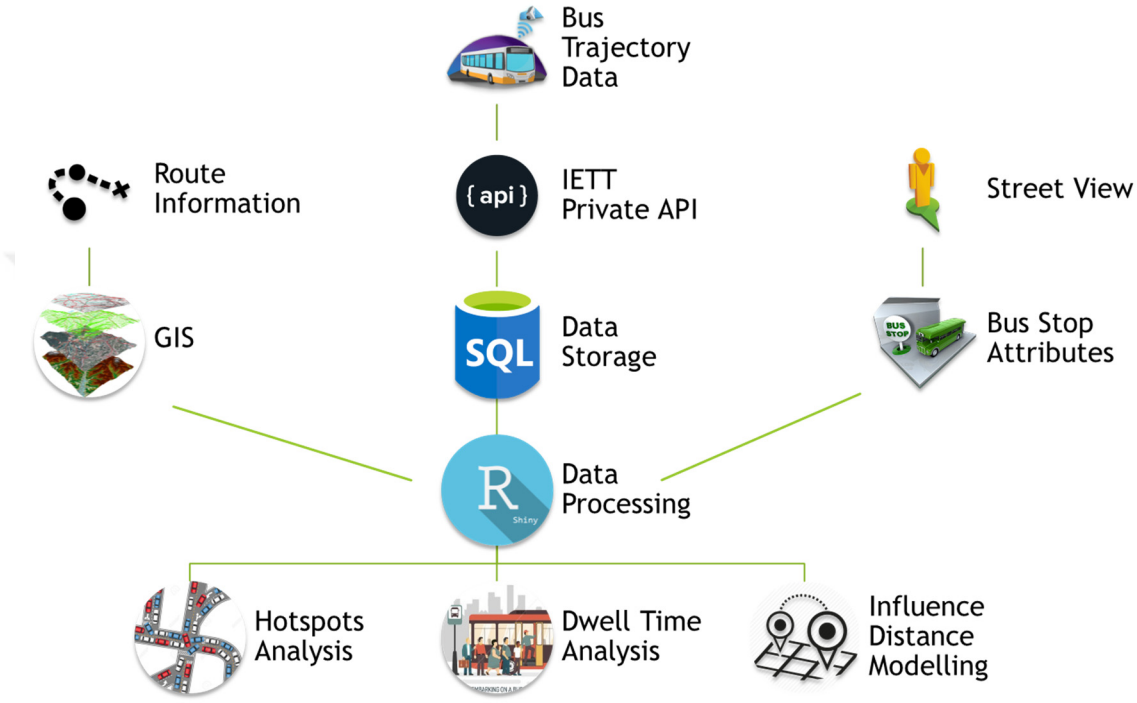

Figure 3.2. The flowchart of the methodology

3.1. Data Storing Process

Trajectory logs of each bus are separated as data files for each day. Each data file has approximately 5000 data rows with time and location information and size of 250KB. Data of one month of all selected routes have more than size of 1GB. Data files were downloaded with private API call from IETT website with the permission of the officials in IETT.

HTTP request of a tool called CURL is used to download files. CURL is open source software for transferring data in command lines or scripts. A script file is prepared for bulk download. Each line downloads one file and names it with a route

name, a vehicle number and a date. The example code, where the vehicle number is “A-001”, the route name is “43R” and the date is April 1st, 2016, is curl “http://.....&date=20160401&vehicle=A-001” -o “43R_A-001_20160401.csv”.

Private API needs two query strings, which are date of the record and the vehicle door number. The output file includes location and time data but there is no information about route name without analyzing the trajectory data. In order to simplify the analysis, the information of the bus assignments to the routes for the selected dates are gathered from IETT. In other words, the door numbers of buses working on the selected routes in each day of April are recorded by IETT and this information is used while downloading data files.

The files are in CSV (Comma-separated Values) format. CSV format is commonly used as data exchange format and widely supported by computer applications. However, the flat-file format is not suitable for data storing and fast querying the data. On the other hand, tabular format can be easily imported into relational database management systems (RDBMS). Some of the most common RDBMS systems are MySQL, Microsoft SQL Server, Oracle and DB2. Because of familiarity of users to the product, SQL Server is chosen for storing the GPS data.

Before importing all data files into database, the files are analyzed descriptively. In Table 3.1, randomly chosen 10 data files are shown. “Start” and “End” columns represent logging time interval in UNIX timestamp format. UNIX timestamps which are commonly used in computer science, are seconds since Jan 01 1970. “Id” is the primary key which is the information about that data file. The primary key will create a connection between database tables. The table is imported to the SQL server.

Table 3.1. The "Route" table in the database, which has data file information.

Id	**Route**	**Day**	**Filename**	**Start**	**End**	**Line**
4779	500T	28	500T_28.4779	1461790812	1461877191	5430
751	500T	25	500T_25.751	1461531610	1461617998	5297
1811	202	1	202_01.1811	1459458003	1459544386	5456
5156	500T	1	500T_01.5156	1459458005	1459544384	5492
4694	500T	26	500T_26.4694	1461618008	1461704388	5475
4672	30A	26	30A_26.4672	1461618002	1461704390	5488
2752	202	27	202_27.2752	1461704412	1461782945	4983
1813	43R	1	43R_01.1813	1459458000	1459544393	5202
3010	202	13	202_13.3010	1460494800	1460581185	5452
5284	500T	4	500T_04.5284	1459731519	1459803589	1026

Each data file has "Id" value that will be used to access the file information via the "Route" table. The id value needs to be implemented to data points. Data points inside the data files have 4 values: Bus vehicle door number, Date time, Latitude and Longitude. The door numbers can be replaced with the id value since the attributes of the data files have already this information.

SQL Server supports geography spatial data type. Utilizing this feature, Geographic Information Systems (GIS) functionality can be used in finding nearest point and coordinate transformation. All data files are bound together with their route id number. Geography point objects of SQL Server are created using latitude and longitude values of all data points. Time values are also transformed in international format. Finally, all data are imported to the "Geography" table in the database, which is shown in Table 3.2.

Table 3.2. The "Geography" table in the database, which has trajectory information.

Geo Id	Route Id	Time	Latitude	Longitude	Geography Point
32633	1816	2016-04-01 22:31:59	41.055000	29.023607	0xE6100000010CD7A3703D0A874440E8F4BC1B0B063D40
32634	1816	2016-04-01 22:32:15	41.055733	29.022320	0xE6100000010C6B274A4222874440F60B76C3B6053D40
32635	1816	2016-04-01 22:32:30	41.056545	29.020860	0xE6100000010C4CE0D6DD3C87444068CBB91457053D40
32636	1816	2016-04-01 22:32:46	41.057457	29.019165	0xE6100000010C90F63FC05A874440543A58FFE7043D40
32637	1816	2016-04-01 22:33:01	41.058190	29.017963	0xE6100000010C257A19C572874440BE89213999043D40
32638	1816	2016-04-01 22:33:17	41.058810	29.017235	0xE6100000010CC156091687874440B858518369043D40
32639	1816	2016-04-01 22:33:33	41.059840	29.016462	0xE6100000010C0F7F4DD6A887444073A087DA36043D40
32640	1816	2016-04-01 22:33:49	41.061115	29.016054	0xE6100000010CC425C79DD287444051F86C1D1C043D40
32641	1816	2016-04-01 22:34:04	41.062298	29.015690	0xE6100000010C984D8061F9874440CEDF844204043D40
32642	1816	2016-04-01 22:34:20	41.063960	29.015303	0xE6100000010C47205ED72F884440DB87BCE5EA033D40
32643	1816	2016-04-01 22:34:35	41.065323	29.014616	0xE6100000010C9B560A815C88444093FFC9DFBD033D40

To simplify the data structure, latitude and longitude columns can be removed but they are kept for debugging purpose. SQL Server automatically creates "GeopointId" column as the primary key of the database table. However, it may be removed since the timestamp values are already unique for each bus, which can be used as the primary key. To demonstrate spatial data functionality on SQL Server, 5000 data points are shown on coordinate system in Figure 3.3.

While importing data files into the SQL Server, R software is used and the connection is established via an odbc driver with an R package named "RODBC". The driver provides a query mechanism in R-stats. Additionally, data tables of R can be imported and exported from/to SQL Server. Data files are preprocessed as data tables in R and transferred into SQL Server with the odbc driver. The script is shown in Figure 3.4.

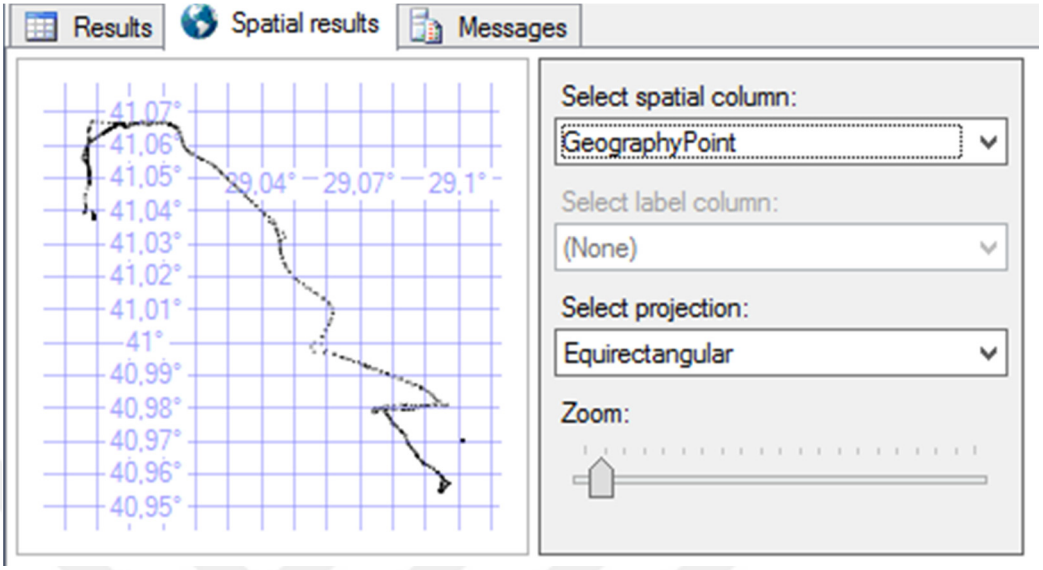

Figure 3.3. Trajectory of 5000 data points

```
1  library(RODBC)
2  myconn <-odbcConnect("BUS_GPS")
3  sqlSave(myconn, Route, "Route", rownames=F, addPK='RouteId')
4
5  apply(Route, 1, function(row){
6    dat <- read.table(row['filename'], sep="\t", skip=1, col.names = c("OTO","TIME","LON","LAT",""))[-5]
7    dat$RouteId <- row['id']
8    sql <- paste0("UPDATE Route SET DoorId = '",
9                  as.character(dat[1,'OTO']) ,
10                 "', Route = '",
11                 as.character(row['hat']) ,
12                 "' WHERE RouteId = ",
13                 as.numeric(row['id']) )
14   sqlQuery(myconn, sql )
15   sqlSave(myconn, dat[,2:5], "Geopoint", rownames=F, append = T)
16 })
17
18 close(myconn)
```

Figure 3.4. R code that imports data files into SQL Server with the odbc driver.

In this script, the route information which is prepared above is imported (line 3). For each data file, data are parsed from csv format (line 6). Bus vehicle door number is extracted and saved in the route information table (line 7-14). Only necessary columns are imported in “Geopoint” table in database (line 15). The R package does not support geography feature, so the necessary transformation is performed in SQL Server Management Studio shown in Figure 3.5. Number 4326

represents WGS 84 (World Geodetic System) which is most widely used standard in cartography, geodesy, and navigation including GPS.

```
CREATE TABLE [dbo].[Geopoint]
(
    [GeopointId] BIGINT NOT NULL PRIMARY KEY IDENTITY,
    [RouteId] INT,
    [Time] DATETIME NOT NULL,
    [Latitude] float NOT NULL,
    [Longitude] float NOT NULL,
    [GeographyPoint]  AS [geography]::Point([Latitude], [Longitude], 4326)
)
```

Figure 3.5. Creating database table “Geopoint” with geography point.

Before processing the data in R software, a subset of the data can be queried with the odbc driver. A query can use all the power of SQL language, such as aggregating, joining, ordering functionality. R code querying all buses working on 30A route and getting all data points is shown in Figure 3.6. In the first code block (line 4-8), a new connection to database is established. This retrieves route information of “30A”. The second code block is creating clusters for parallel computing. The third code block (line 14-20) is getting data of all the selected routes. This method is much simpler and efficient than dealing with plain files.

```
1  library(RODBC)
2  library(magrittr)
3  library(parallel)
4  cc <- odbcConnect("BUS_GPS")
5  route <- "30A"
6  hatlar <- route %>%
7    sprintf("SELECT * FROM Route WHERE Route = '%s'",.) %>%
8    sqlQuery(cc, .)
9
10 cl <- makeCluster(40)
11 clusterEvalQ(cl, library(magrittr))
12 clusterEvalQ(cl, library(RODBC))
13
14 data <- parLapply(cl, hatlar$RouteId, function(routeId){
15   cc <- odbcConnect("BUS_GPS")
16   routeId %>%
17     sprintf("SELECT Time, Latitude, Longitude FROM Geopoint WHERE RouteId = '%i'",.) %>%
18     sqlQuery(cc, .) %>%
19     return
20 })
```

Figure 3.6. R code querying all buses working on 30A route to get all data points

3.2. Bus Route

Processing bus GPS data requires the shape of the bus routes, the bus stops that belong to each specific bus route, and the location of each bus stop. The data provided by IETT does not include the bus route information. Therefore, necessary data should be obtained from other sources.

Firstly, publicly shared data from “citySDK” are explored. CitySDK is a 5-year-old project providing “service development kit” for cities and developers. 8 cities have worked together, namely Amsterdam (Netherlands), Barcelona (Spain), Helsinki (Finland), Istanbul (Turkey), Lamia (Greece), Lisbon (Portugal), Manchester (UK), Rome (Italy). Istanbul has been piloted for mobility component including general base layer of information and geography (OpenStreetMap), public transport and schedules (İETT, Ulaşım AŞ., Şehir Hatları AŞ.) and Istanbul's spatial data [47].

As a demonstration, a simple query is run on the citySdk API. A query asks public transportation stops within 200-meter-circle from 41.083 latitude and 29.04 longitude. From the query, 4 stops are returned (2 stops in each direction) named "CENGİZ TOPEL" and “BASIN SİTESİ”, shown in Figure 3.7. “ptstops” represents public transportation stops, “lat” is latitude and “lon” is longitude. The response is in JSON (Javascript Object Notation) format. General Transit Feed Specification is denoted using “gtfs”.

The id of the first bus stop is selected (gtfs.stop.istb.183495) and used in another query asking transportation lines containing this stop. As shown in Figure 3.8, 5 records are returned. The lines containing “Cengiz Topel” stop is 559C, 59K,

59R, 59RS, 59UÇ. Each route information includes the terminals where the bus coming from and going to, and also the ids which can be used for other queries.

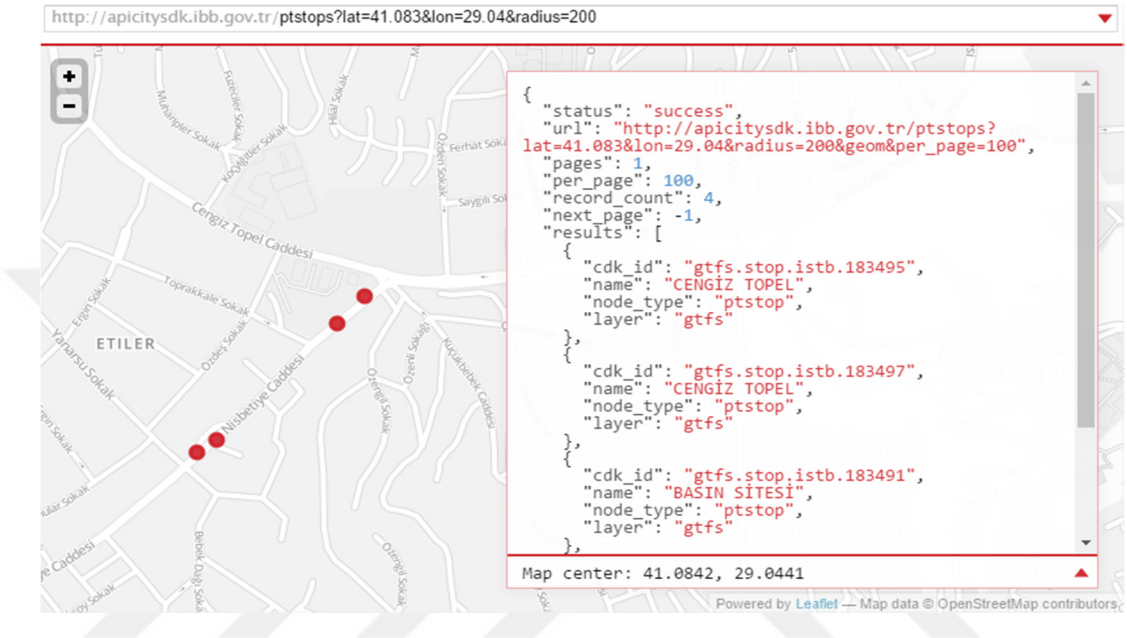

Figure 3.7. A query asking public transportation stops within 200-meter-circle from the selected location in CitySDK API.

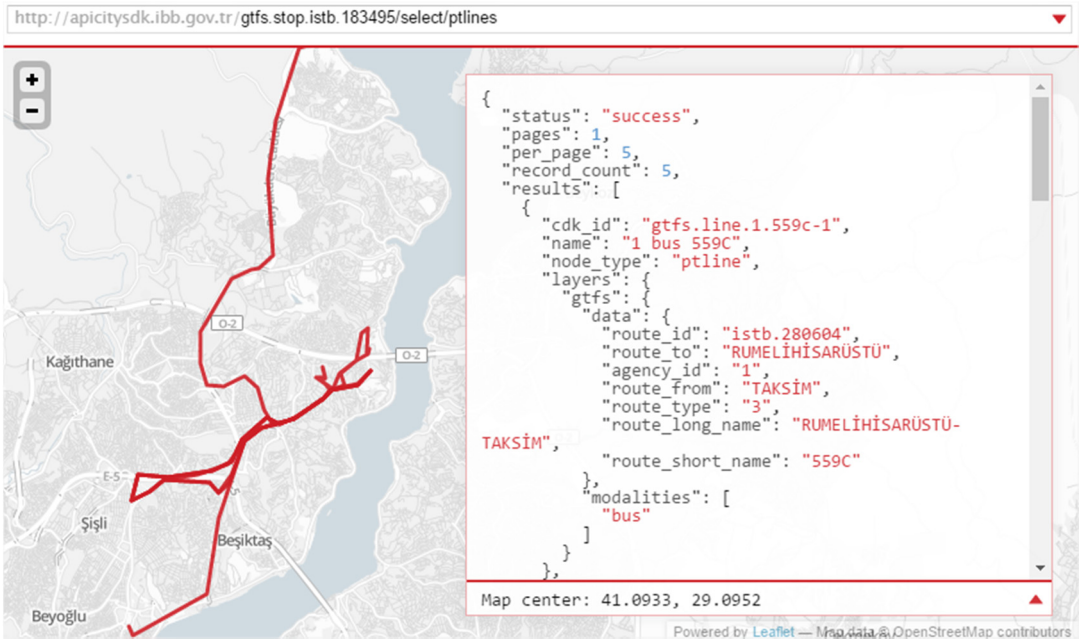

Figure 3.8. A query asking public transportation lines containing “Cengiz Topel” bus stop in CitySDK API.

Another way to select bus routes is to query it with names. In Figure 3.9, the bus route is queried from its short name “DT2”. After accessing its cdk_id, the API can be asked to return all bus stops. Additionally, “&geom” can be added into query in order to print all coordinates of polyline drawn on the screen. The coordinates of the bus route is recorded to be used for map matching.



Figure 3.9. A query asking public transportation lines has a short name “DT2” in CitySDK API.

Bus stops information of a selected route can be queried by adding “/select/ptstops?geom” at the end of the id of the route. The API will return all bus stops with id, names and coordinates in JSON format. The result can be parsed and returned into a table with the following code, “respond.results.map(s=>s.name,s.layers.gtfs.data.stop_id.concat(s.geom.coordinates))”. A sample result of “DT2” bus route is shown in Table 3.3.

Table 3.3. The bus stops of “DT2” bus route.

Bus Stop Name	**Id**	**Longitude**	**Latitude**
VADİ	istb.183403	29.02330	41.06119
GÜL	istb.183399	29.02314	41.05694
VADİ PARK	istb.183146	29.02290	41.05579
DEREBOYU	istb.183606	29.02286	41.05411
ZÜBEYDE HN.KIZ LİSES	istb.183602	29.02463	41.05105
KABATAŞ LİSESİ	istb.193246	29.02427	41.04809
GALATASARAY ÜNV.	istb.193244	29.02093	41.04652
YAHYA EFENDİ	istb.193241	29.01832	41.04550
ÇIRAĞAN	istb.193240	29.01281	41.04342
BEŞİKTAŞ B.ÜNİVERSİT	istb.183780	29.00781	41.04239
AKARETLER	istb.193238	29.00394	41.04165
İNÖNÜ STADI	istb.184410	28.99500	41.04030
D.BAHÇE GAZHANE CD.	istb.184415	28.99281	41.03995
TEKNİK ÜNİVERSİTE	istb.184168	28.99308	41.03791
GÜMÜŞSUYU	istb.184169	28.99073	41.03691
TAKSİM	istb.184160	28.98691	41.03728
ELMADAĞ	istb.184153	28.98660	41.04228
HARBİYE	istb.184151	28.98762	41.04871
PANGALTI	istb.184149	28.98752	41.05186
OSMANBEY	istb.184146	28.9871	41.05490
ŞİŞLİ ETFAL	istb.184142	28.987	41.05790

Table 3.3. The bus stops of "DT2" bus route (cont.).

ŞİŞLİ CAMİİ	istb.184139	28.98956	41.06232
ŞİŞLİ MERKEZ	istb.184136	28.99222	41.06402
MECİDİYEKÖY-VİYADÜK	istb.184114	28.99589	41.06637
GAYRETTEPE	istb.184108	29.00508	41.06752
ESENTEPE	istb.184105	29.00843	41.06816
ZİNCİRLİKUYU	istb.184103	29.01271	41.06952
LEVENT	istb.183458	29.01743	41.07416
GÜVERCİN	istb.183482	29.02193	41.07677
BELEDİYE SİTESİ	istb.183577	29.02474	41.07693
TALİM YERİ	istb.183579	29.0282	41.07331
TURİZM SİTESİ	istb.183581	29.02969	41.07028
ULUS MAHALLESİ	istb.183583	29.03032	41.06661
ÇAMLITEPE	istb.183585	29.03069	41.06490
ESENEVLER	istb.183588	29.03117	41.05942
T.R.T.	istb.183596	29.02979	41.05761
PORTAKAL YOKUŞU	istb.183598	29.02919	41.05564
KÖPRÜ AYAĞI	istb.183600	29.02867	41.05219
ZÜBEYDE HN.KIZ LİSES	istb.183604	29.02644	41.05048
ZÜBEYDE HN.KIZ LİSES	istb.183603	29.02509	41.05065
VADİ PARK	istb.183147	29.02293	41.05579
GÜL	istb.183400	29.02335	41.05696
VADİ	istb.183404	29.02405	41.06022

3.1. Map Matching

The GPS data rarely shows the actual position with high accuracy. Noises of locations can reach more than 40 meters due to weather conditions and high buildings [48]. Figure 3.10 shows 3 million observations in Kabataş district that contain high level of noise. These faulty data may corrupt the analysis and mislead some problems. In order to deal with it, incorrect data points should be either removed or matched with the actual positions.

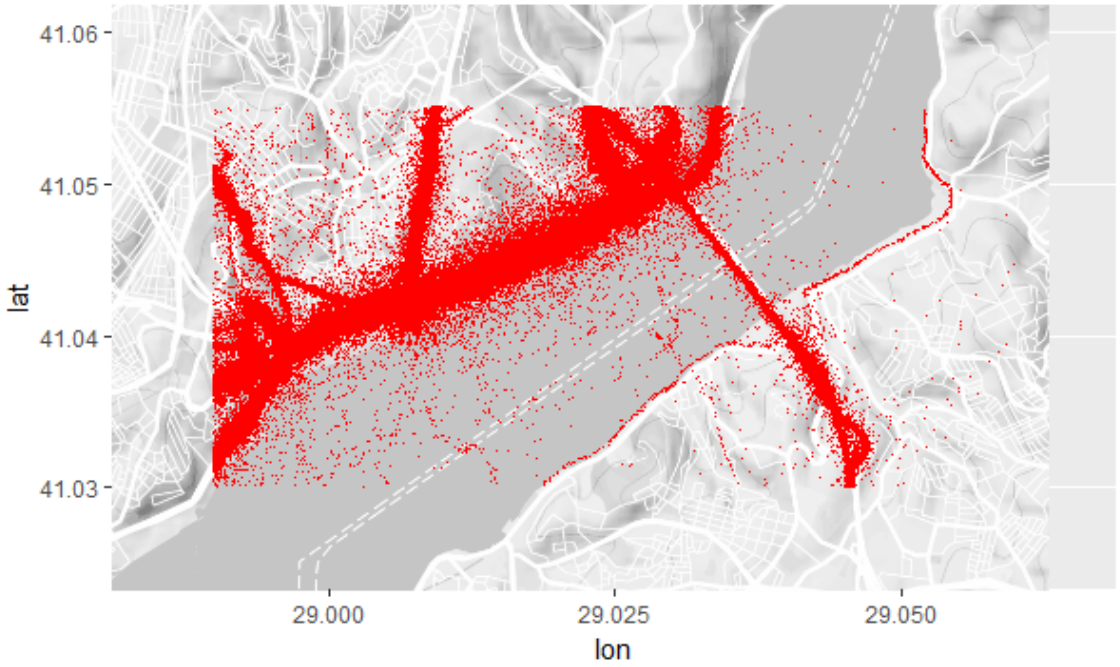

Figure 3.10. Three million observations in Kabataş district that contain high level of noisy data

Matching the data points to the most possible position on the roads is called map matching. To find the most possible position; firstly, the road network should be modeled. The roads can be represented as nodes and links. The nodes are the smallest number of connection points which construct the shape of the road smoothly.

By the nature of shape files, curvy roads contain more nodes and shorter links, straight roads contain less nodes and longer links. For the sake of consistency, the link length is fixed around 20 meters. The long edges are split into 20-meter pieces and nodes are created accordingly. If a link is less than 20 meters, it is merged with the next link and split again. An example link splitting is shown in Figure 3.11.

After creating the network by arranging all nodes and links through bus routes, the next step is to match the GPS data to the network. The most common method suggested in literature is to find the Euclidian distance of the data points to the nodes and match them with the nodes having the shortest distance. This method is working fine with non-complex networks like straight one-way routes.

Intersecting roads is decreasing the accuracy of map matching because of noise of the data points. Another method that increase the accuracy is to add orientation (bearing) information. To apply this method, orientation of the links of the network is calculated. Then, the data points are matched with the nearest nodes that have similar orientation with data points. This method allows matching a data point to the right side of the roads if the bus is using opposite side of the two-way roads.

3.2. Calculation of Speeds

Each data log has 5 thousand data points on the average with a total of 2.1 million data points. Data points are collected approximately 4 times in a minute, the median of the sample rate is 0.0625 Hz (16 second). Distance between geographic (geodetic) coordinates of successive data points is calculated based on an ellipsoid (spheroid) model of the world. WGS84 ellipsoid model is used in the calculation, which is using inverse geodesic function in “Geosphere” package of R [49].

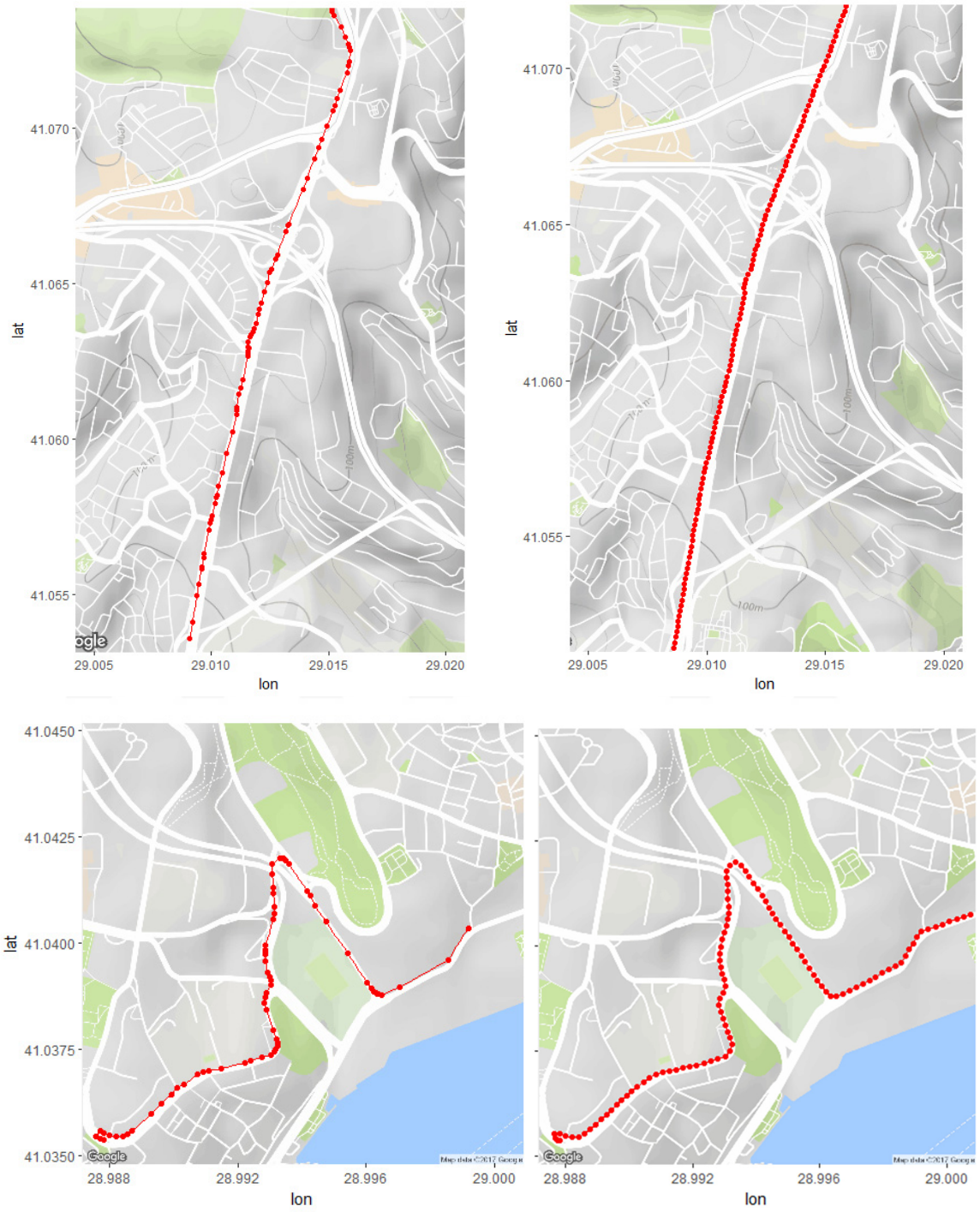

Figure 3.11. Before and after route fixing for long roads (above) and curvy roads (below).

An example data set is shown in Table 3.4. Time, longitude and latitude columns are given; time difference, distance and speed columns will be calculated. Information of each row will be calculated using values in the previous row and itself. First row will not be calculated because it does not have previous row.

Table 3.4. Speed calculation example

	Time	Time Diff	Longitude	Latitude	Distance	Speed
1	2016-04-02 14:02:16	-	29.05036	41.08808	-	-
2	2016-04-02 14:02:33	17	29.05036	41.08808	0.00	0.00
3	2016-04-02 14:02:51	18	29.05024	41.08787	25.01	4.74
4	2016-04-03 14:03:08	17	29.05000	41.08689	111.50	26.76
5	2016-04-03 14:03:24	16	29.04966	41.08659	44.19	9.94
6	2016-04-03 14:03:40	16	29.04886	41.08629	74.37	15.75
7	2016-04-03 14:03:58	18	29.04792	41.08592	89.23	17.85
8	2016-04-04 14:04:15	17	29.04736	41.08550	66.00	14.85
9	2016-04-04 14:04:31	16	29.04732	41.08550	3.61	0.81
10	2016-04-04 14:04:46	15	29.04732	41.08550	0.00	0.00
11	2016-04-05 14:05:03	17	29.04732	41.08550	0.00	0.00
12	2016-04-05 14:05:19	16	29.04717	41.08549	12.77	3.06
13	2016-04-05 14:05:34	15	29.04687	41.08538	28.16	6.76
14	2016-04-05 14:05:50	16	29.04567	41.08533	101.05	21.40

Firstly, time difference is calculated. Looking at time information of first and second row, there are 2016-04-02 14:02:16 and 2016-04-02 14:02:33 values. The time difference of them is 17 seconds. This value is written on time difference column of the second row. The same process is repeated on the other rows.

Distances between each data point are calculated using R package name "geosphere" where geodesics algorithms are implemented. "distGeo" function of the package asks 4 arguments: Longitude and latitude information of 2 points, major (equatorial) radius of the ellipsoid and ellipsoid flattening. The default values of WGS84, which is the commonly accepted ellipsoid model for Earth, are used for the last two arguments. Example usage of "distGeo" function is shown in Figure 3.12.

```
p1 = c( 29.05036, 41.08808 )

p2 = c( 29.05024, 41.08787 )

geosphere::distGeo(p1, p2, a=6378137, f=1/298.257223563)

>>> Distance: 25.40794 meter
```

Figure 3.12. Example code of using "distGeo" function

After calculation of time difference and distance, speed information can be calculated by the formula, $V_m = \Delta x / \Delta T$. Due to the fact that V_m is in meter/second unit, it needs to be multiplied by 3600 sec / 1000 meter to convert it into km/hour. Then, the final formula which is used to calculate speed using distance and time difference is $V = \Delta x \ / \ \Delta T \times 3.6$.

3.3. Detection of Hotspots

In this dataset, GPS data points are logged every 16 seconds. When a bus logs with a constant frequency, it leaves same amount of logs in a defined amount of time. That results in direct relationship between speed of buses and data points in the area. The slower buses travel, the more data points they create in 1 km. This means data points will be denser in the areas that have slower average speed.

The relationship is shown in Figure 3.13. As an example, if a bus travels 40 km/h, it will leave 5 data points in 1 kilometer. The speed can be converted to 11.1 m/s and in 16 seconds it will travel 177.8 meter. 1 kilometer over 177.8 is 5.6 times. It can be assumed that after 5 data points, the bus will leave the area.

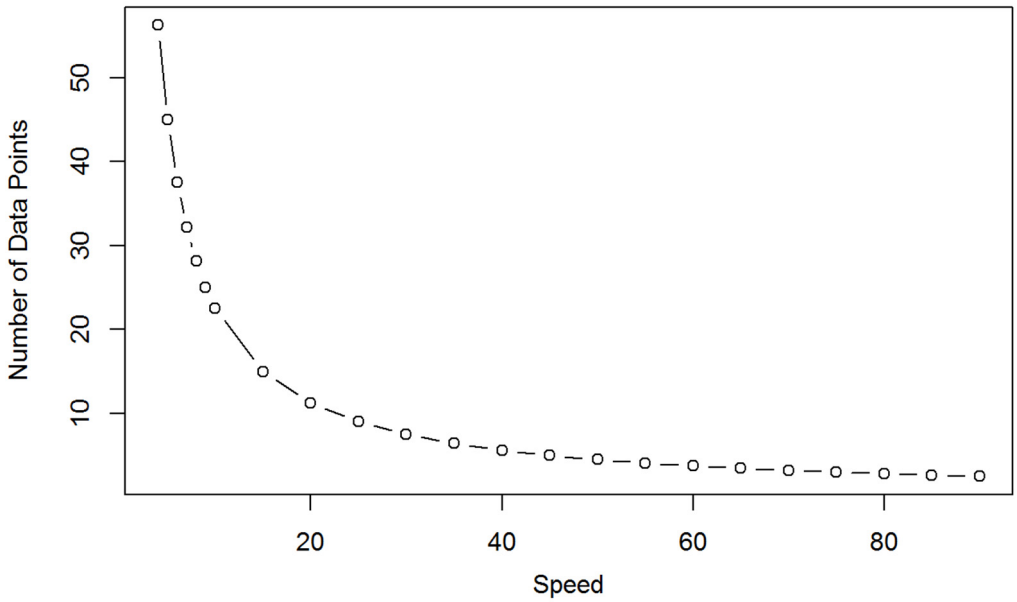

Figure 3.13. Number of data points buses log in 1 kilometer changing with bus speed

In order to identify the hot spots in the network, the data point density can be used, since the more data points indicate the slower average bus speed. Firstly, a sample bus route is selected for analysis. The selected route "40T" has mostly 1-2 lanes in one direction, it can be considered as a seaside route.

All data points (620,000) of buses working in this route are combined. The data points are plotted as dots on the left side of Figure 3.14. To create a density map two-dimensional kernel density estimation is used. The function is implemented in R named "kde2d" in "MASS" package. The density estimation is drawn on map with "ggplot" package.

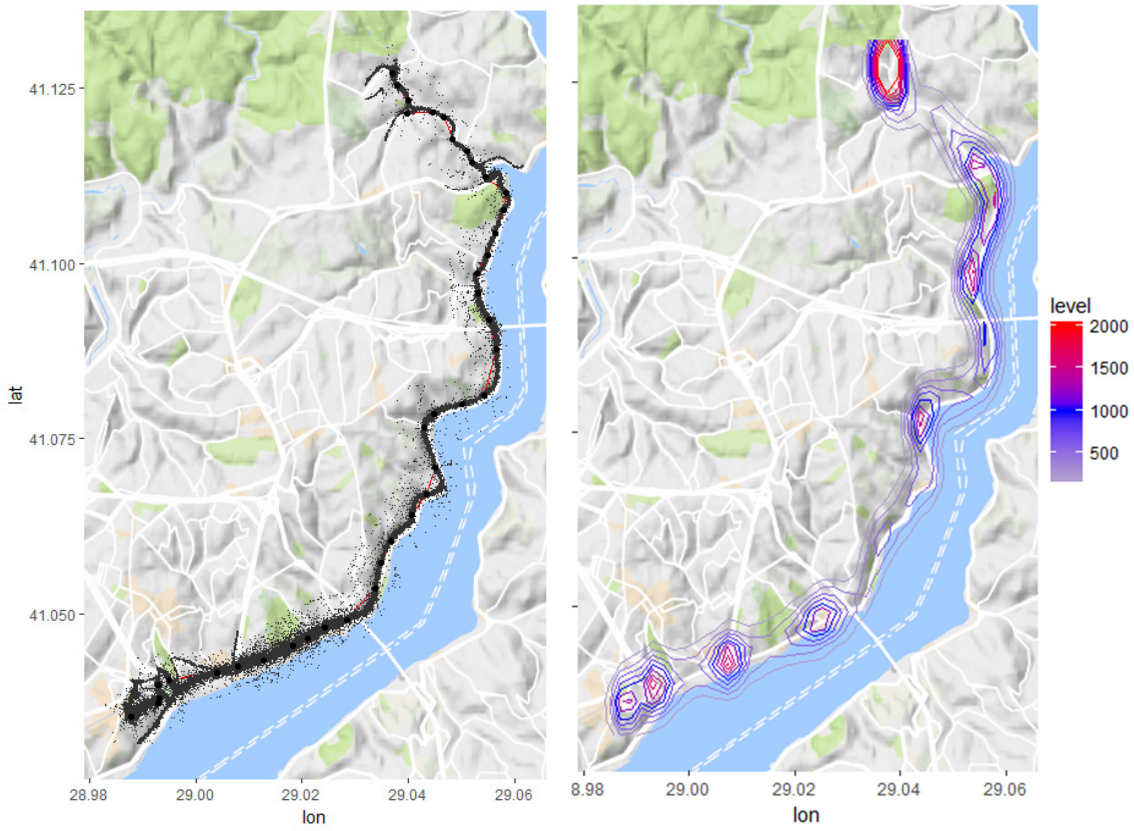

Figure 3.14. Data points of a sample route and its density map

3.4. Detection of Stopping of Buses

Threshold speed for determining whether a bus is stopping is needed, since the bus can stop without having zero speed value, because only average speed can be calculated from data. Since the median sampling time is 16 seconds, the highest average speed which includes bus stopping can be calculated.

The threshold speed V_S is calculated with simple bus stopping simulation as shown in Figure 3.15. The bus deceleration and acceleration rates are taken from the literature [50] as 0.19g and 0.15g. In the shortest stopping interval case, it is assumed that bus is stopping only 1 second. In 16 seconds period, which is median sampling period of data, the average bus speed is calculated as 5.9 km/h.

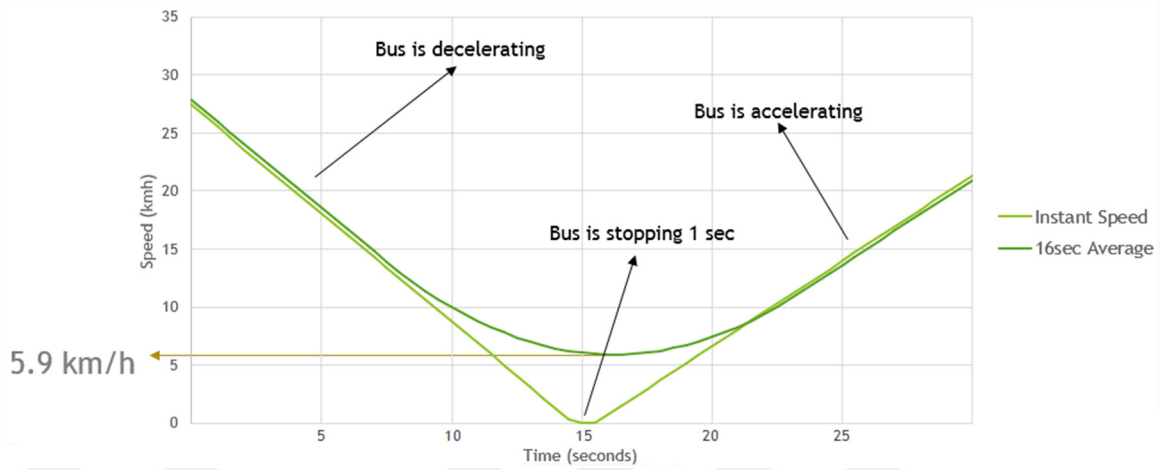

Figure 3.15. One second bus stopping simulation to calculate the threshold speed V_S.

All data points, which have less speed value than threshold speed V_S, are labelled as stopped. With longer stopping cases, average bus speeds will be decreasing. Thus, it is claimed that data with less than V_S gives stopping of the buses. Consecutive stopping point durations are summed up to determine durations of moments when buses stop.

3.5. Calculation of Dwell Times

First of step of the dwell time calculation is to calculate stopping duration. The data points are labeled as stopped if they are below the threshold speed V_S which is calculated in the "Stopping Detection" section. Duration of a stopping is calculated by summing up all time differences of successive "stopped" labeled points.

In the example Table 3.5, there are six data points. Starting from the 9th row to the 12th row, the data points are labeled as "stopped" because their speed are less than the threshold speed V_S. Due to the fact that they are consecutive, the total stopping duration is calculated by summing their time differences: 16, 15, 17 and 16. The total stopping duration of this stopping is calculated as 64 seconds.

Table 3.5. Stopping duration calculation.

	Time	Time Diff	Longitude	Latitude	Distance	Speed	Label
8	2016-04-04 14:04:15	17	29.04736	41.08550	66.00	14.85	Moving
9	2016-04-04 14:04:31	16	29.04732	41.08550	3.61	0.81	Stopped
10	2016-04-04 14:04:46	15	29.04732	41.08550	0.00	0.00	Stopped
11	2016-04-05 14:05:03	17	29.04732	41.08550	0.00	0.00	Stopped
12	2016-04-05 14:05:19	16	29.04717	41.08549	12.77	3.06	Stopped
13	2016-04-05 14:05:34	15	29.04687	41.08538	28.16	6.76	Moving

As the second step, the detected stopping points are needed to match to the bus stops. Distances between every detected stopping points and bus stops on the route are calculated. The shortest distances of them are calculated. If a stopping point is in 100-meter radius of a bus stop, it is matched that bus stop.

For a demonstration purpose, the third bus stop (Barbaros Bulvarı) is focused on. All dwell times of the stop plotted with record time on x axis in Figure 3.16. Bus operation hours starting from 6 AM and there is not much data after midnight.

The recorded times of the dwelling times is on the x axis and their duration is on the y axis. In order to find the pattern LOESS (Local Polynomial Regression Fitting) algorithm is used. This algorithm can be used with “loess” function in R.

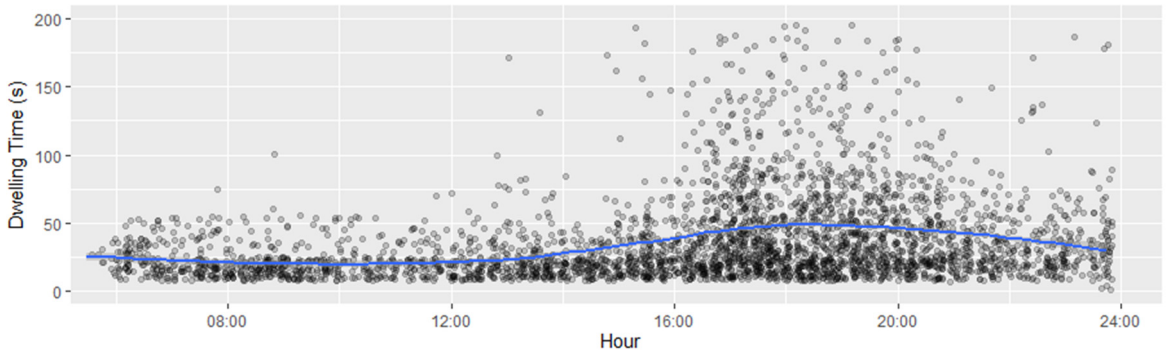

Figure 3.16. Stopping duration and hour and dwelling pattern.

3.6. Preparation of Bus Stop Characteristics Data

12 bus routes are selected for this study and they have total 451 bus stops. All bus stops are investigated using Yandex Maps, Google Maps, IBB Şehir Rehberi (Istanbul Municipality City Map). The maps have the ability to show the network around bus stops from above and their street views. With a simple Excel formula, coordinates of the bus stops are converted into 3 links of each map view.

14 information of each bus stop is recorded: Firstly, two of them are number of lanes where the bus stops locate and whether pockets of the bus stops exist. Other information are related interruptions to the network around the bus stops and how far these interruptions are from the bus stops. These interruptions are drops in the number of lanes, increase of the number of lanes, traffic light, crossroad, roundabout, and significant entry. These are separately recorded whether they are before or after the bus stops.

To measure the distance between the bus stop and the interruption, following steps are applied. First, the existence of an interruption is observed in the street view. The interruption is located in the map views. The bus stop has been already located thanks to Excel link. The maps can measure distances by clicking two or

more points on the map. The distance which is read on the measurement window are recorded in the excel file shown in Figure 3.17.

					Yandex	İBB	Google	Pocket?	Lane num	Lane drop (m)	
Id	Lat	Lng	Direction	Name				(1-0)	(one way)	Before	After
ist_225151	41	29	Bostancı - Taksim	Öğretmen Hayrullah	Street	Sokak	Harita	1	1	0	0
ist_225161	41	29	Bostancı - Taksim	Şenesenevler	Street	Sokak	Harita	0	2	0	25
ist_225181	41	29	Bostancı - Taksim	Bostancı	Street	Sokak	Harita	0	2	0	0
ist_225621	41	29	Bostancı - Taksim	Göztepe Köprüsü	Street	Sokak	Harita	1	1	100	0
ist_225631	41	29	Bostancı - Taksim	Yenisahra	Street	Sokak	Harita	0	2	0	0
ist_226251	41	29	Bostancı - Taksim	Siteler	Street	Sokak	Harita	1	2	0	0
ist_226261	41	29	Bostancı - Taksim	Üsküdar Caddesi	Street	Sokak	Harita	1	2	0	0
ist_226302	41	29	Bostancı - Taksim	Yelkenli Değirmen	Street	Sokak	Harita	0	2	0	0
ist_229681	41	29	Bostancı - Taksim	İst.Medeniyet Ünv.	Street	Sokak	Harita	1	4	0	0
ist_303361	41	29	Bostancı - Taksim	Mecidiyeköy Viyadük	Street	Sokak	Harita	1	4	0	120
ist_305421	41	29	Bostancı - Taksim	Gümüşsuyu Peron	Street	Sokak	Harita	1	2	0	0
ist_401181	41	29	Bostancı - Taksim	Üstbostancı	Street	Sokak	Harita	1	1	0	0
ist_105561	41	29	Cevizlibağ - Tuzla Şifa Mah	Panorama 1453	Street	Sokak	Harita	1	3	0	92
ist_106071	41	29	Cevizlibağ - Tuzla Şifa Mah	Topkapı Alt Geçit	Street	Sokak	Harita	1	4	0	98
ist_107402	41	29	Cevizlibağ - Tuzla Şifa Mah	Edirnekapı Kaleboyu	Street	Sokak	Harita	1	3	0	117

(a)

Lane drop (m)		Lane increase (m)		Traffic light (m)		Crossroads (m)		Roundabout (m)		Significant Entry (m)	
Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
0	0	0	0	105	5	0	0	0	0	0	0
0	25	0	0	50	0	0	0	0	0	0	0
0	0	10	0	0	0	0	0	0	0	0	0
100	0	0	70	0	0	0	0	0	0	0	70
0	0	0	0	0	0	0	0	0	0	0	50
0	0	0	0	0	60	0	0	0	0	0	0
0	0	0	0	95	0	0	0	0	0	0	0
0	0	0	100	45	0	0	0	0	0	0	5
0	0	0	0	0	0	0	0	0	0	0	0
0	120	0	0	0	60	0	0	0	0	130	0
0	0	0	0	0	0	0	0	0	0	0	30
0	0	0	0	0	0	0	10	0	0	0	0
0	92	0	0	0	0	0	0	0	0	98	0
0	98	0	0	0	0	0	0	0	0	0	0
0	117	20	0	77	0	0	0	0	0	0	0

(b)

Figure 3.17. The first (a) and the last (b) 12 columns of the Excel file where the measurements of the bus stops are recorded

3.7. Bus Stop Transit Card Data

Passenger demands of the bus stops are obtained as bus stops characteristics. The demand is calculated using transportation smart card data (İstanbulkart) that is used to get in buses all around Istanbul. The data consists of passengers boarding

public transportation. A timestamp, user id and bus stop id are recorded when a passenger uses the card while boarding a bus. It allows calculating demand for each bus stop monthly, daily or hourly.

Transportation smart card data of April 2016 for the selected route is obtained from IETT. The data aggregated by bus stop id and day. After aggregation, a data table is generated with rows of bus stop ids, columns of dates, and values of daily passenger demand of each bus stop on a determined day. All bus stop demands of April 2016 are summed up to calculate total monthly bus use for each bus stop.

3.8. Bus Stop Influence Distance Measurement

Speed patterns around bus stops are examined to find out influence distance of a bus stop. The purpose is to measure the distance between the position where bus speed decreases significantly and the position where bus speed increases significantly around bus stops.

Firstly, nodes are created along the bus route as in the map matching methodology. Then, density measurements from hotspot detection method are applied to find out any possible locations where speed decreases significantly. This methodology is applied for each 15-minute time interval. An example route (U1 Besiktas-Ulus-Zincirlikuyu ring route), is shown in Figure 3.18.

In this example route, data density increase around nodes number 10, 125, 350 and 425. The first (upper left) black zone shows evening traffic congestion in the center of Besiktas. The other hotspot around the node number 125 that is in Zincirlikuyu district is denser at the morning peak. The node number 350 is in Ortakoy where the number of lanes decreases into one due to the historical buildings.

The last bottleneck is also located in the center of Besiktas. Every 5 node numbers and the bus stops of the selected route are represented on map in Figure 3.19.

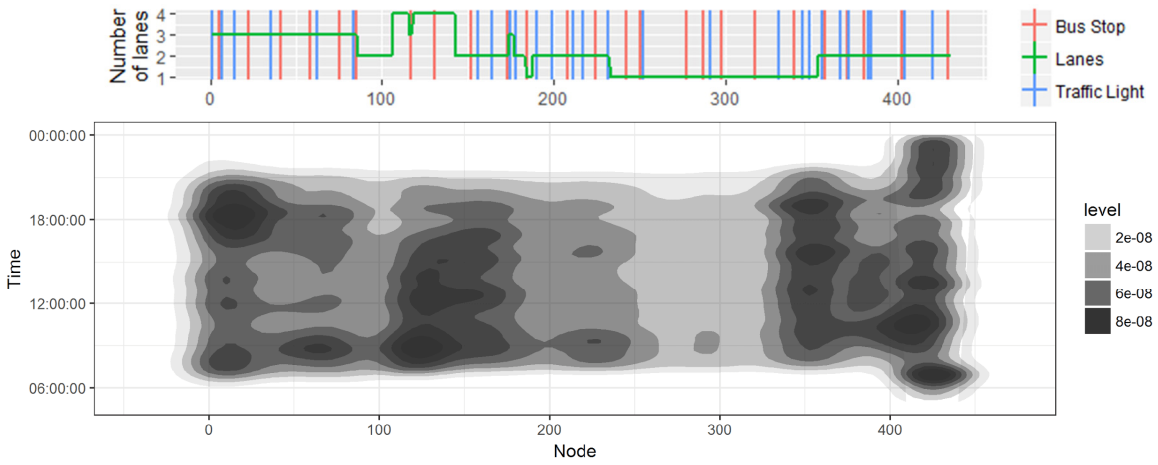

Figure 3.18. Data Density (nodes vs time) along the bus route.

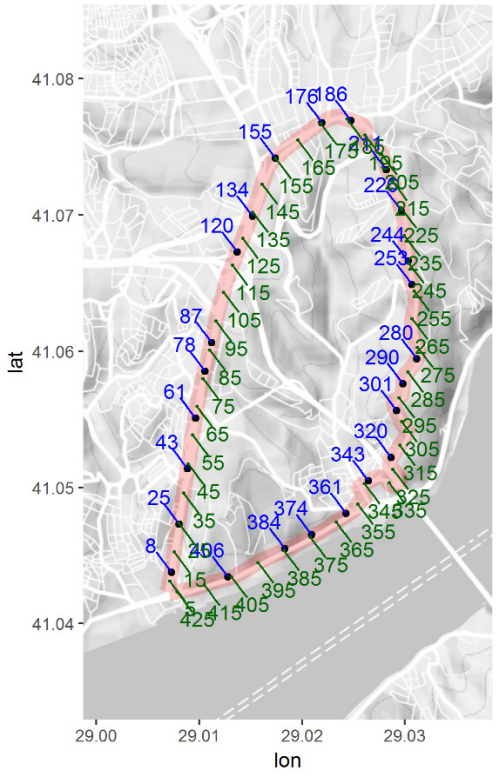

Figure 3.19. Every 5 node numbers (green) and the bus stops (blue) of U1 route

Speed of buses are calculated for each node and time interval. Aggregated values are plotted in Figure 3.20 (a). Locations of the bus stops are drawn as vertical blue lines. Because of intervals of the measurements are too detailed, there are some spaces seen as white squares. Missing data are interpolated and the final version is shown in Figure 3.20 (b).

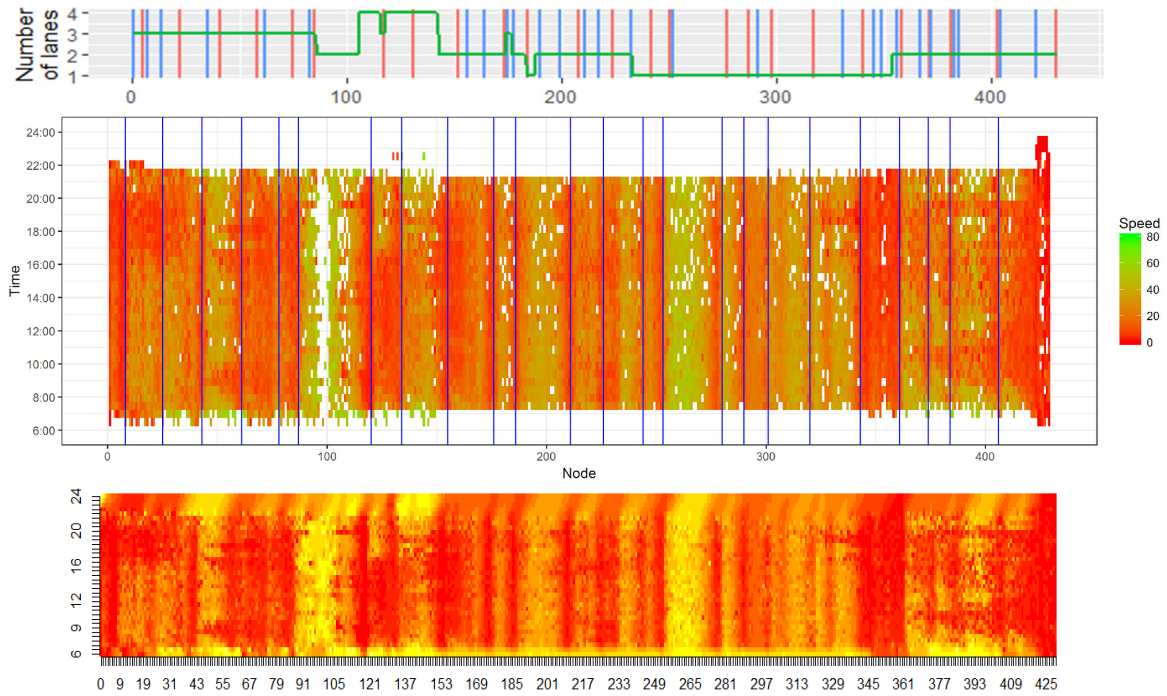

Figure 3.20. (a) Space time speeds and location of bus stops as vertical blue lines (b) Space time speeds after fixed with interpolation

Red areas in the figures represent slow areas, and slow areas near vertical blue lines are influence distances of bus stops. The vertical red patterns on space time speeds graph show that the influence distances are not time specific, although there are some irregularities in morning and evening peaks.

The next step is to measure the width of these influence distances for each bus stop. For example, an off-peak hour of data is selected, and it is filtered with

one dimensional generalized fused lasso method, shown in Figure 3.21. A trend filtering model is chose by perform k-fold cross-validation.

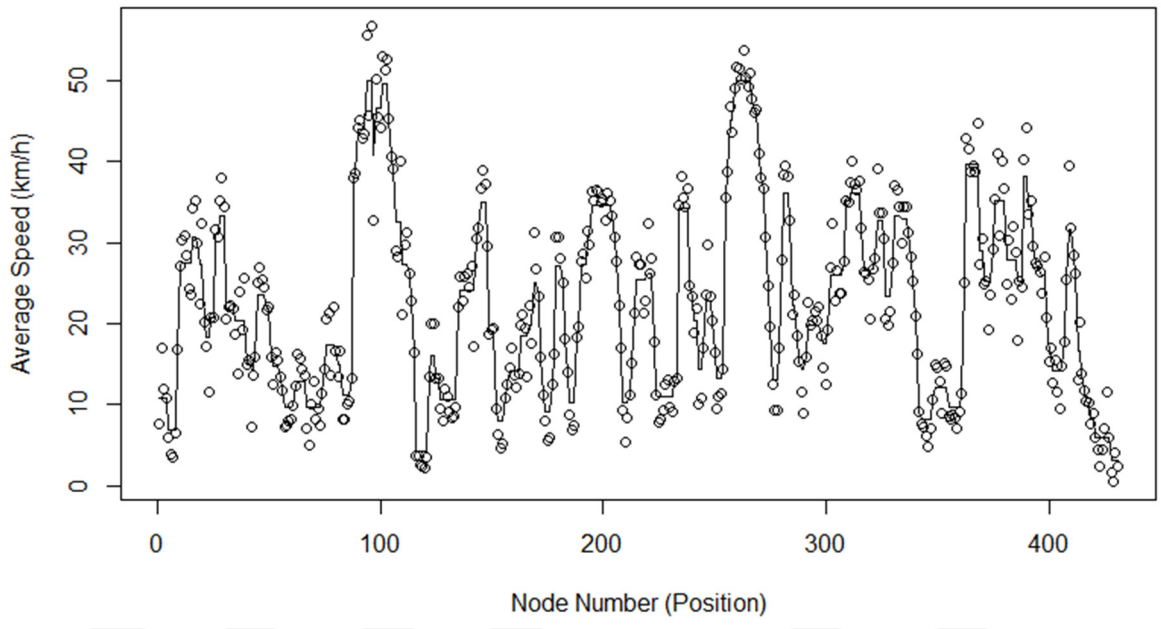

Figure 3.21. Off-peak average speed distribution along the selected route filtered with one dimensional generalized fused lasso method

After filtering the speed data, the flatten data are split at the points where speed data have a dramatic change. The split points before the bus stop location are represented with vertical dashed red line, and the points after the bus stop locations are with vertical dashed blue line. The process of the example route is shown in Figure 3.22 with bus stop order numbers. Then, the differences between points represented with red and blue lines are calculated and assigned to the near bus stops as the influence distance.

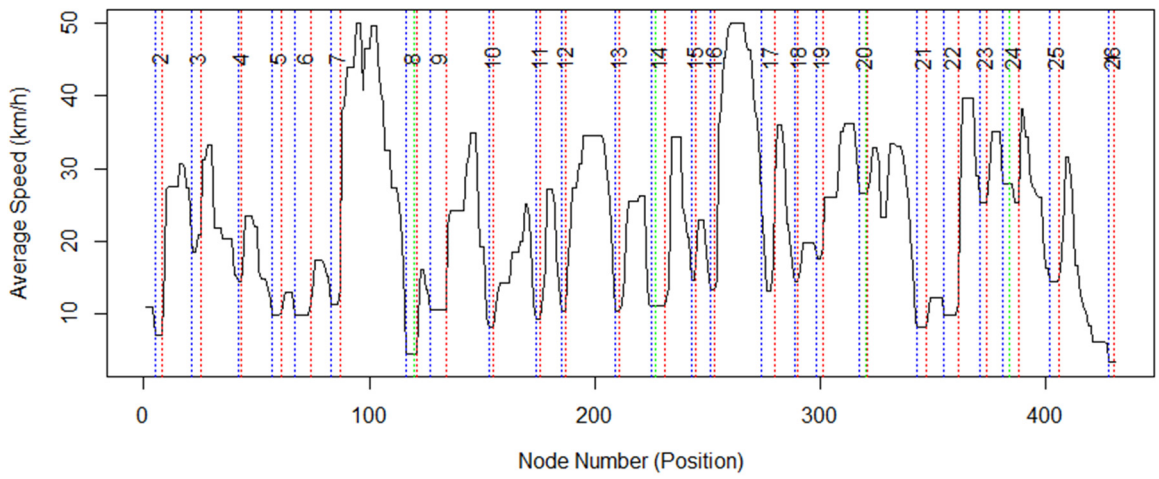

Figure 3.22. The split points where speed data have a dramatic change before (vertical dashed blue line) and after (vertical dashed red line) the bus stop location

4. ANALYSIS AND RESULTS

In this chapter, firstly, it is shown that the bus stops and the network properties such as decrease in number of lanes affects hotspot of buses. Then, the network properties such as pocket existence and number of lanes as well as passenger demand of bus stops are correlated with waiting times in the bus stops. In light of this information, these properties of the bus stops and of the network around the bus stops are used to predict the influence distance of the bus stops using M5P, Random Forest and ExtraTrees methods.

4.1. Hotspot Detection

GPS trajectory data of buses that are in service during one month (April 2016) are obtained from Istanbul Electric Tram and Tunnel Company (IETT) in comma separated vector (CSV) format. A bus route, which has a variety of traffic density and number of lanes, is selected. The selected bus route (43R) is analyzed to identify single trips between stations for separating the data in time windows using R-Stats software.

The raw input data consists of vehicle locations measured by GPS. Each measured point consists of a time-stamped latitude/longitude pair. The roads are also represented in a conventional manner using a graph of nodes and links. The nodes are at intersections, dead ends, and changes in road names. The links represent road segments between the nodes.

On the average, 12 different buses are working on selected route every day, most of which are making around eight trips, and they generate more than 1.5

million data points in one month. One weekday of a month is selected for a detailed investigation in peak and off peak hours, shown in Figure 4.1.

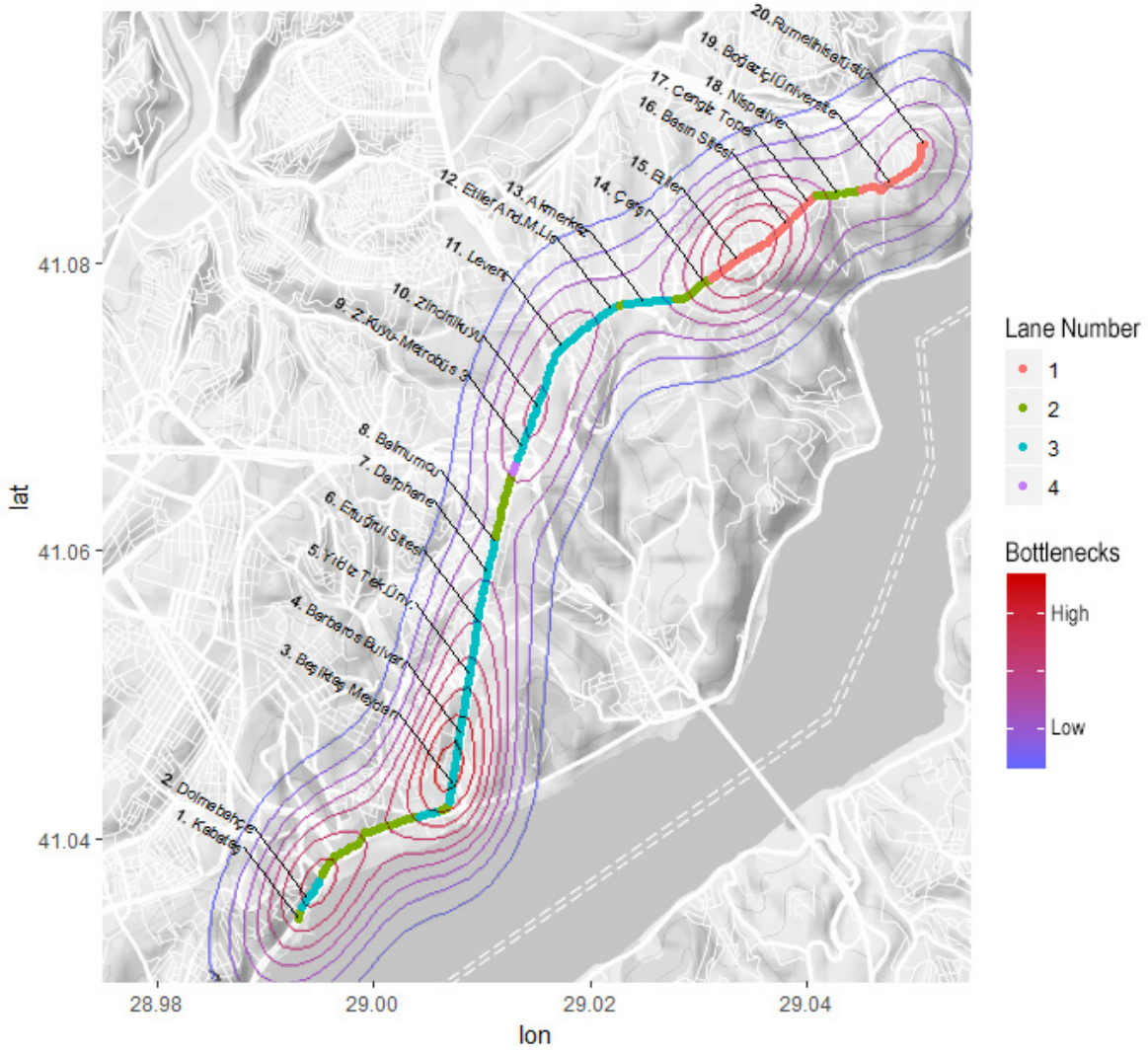

Figure 4.1. Number of lanes, bus stops and stopping points (bottlenecks) on Barbaros and Nispetiye Boulevards.

The number of lanes on 43R line are indicated in Figure 4.1. The inscriptions are pointed out by the black arrows show the location and the names of the stops. The hotspots are drawn where the bus frequently stops. 5 hotspots can be observed. The first one is after Dolmabahçe stop, on the spot where the road is not 3 lanes

anymore, but 2. The second one starts behind Beşiktaş Meydan, and goes up to Yıldız Teknik Üniversitesi. The third one is around Zincirlikuyu stop. The fourth starts with the drops in the number of lanes around the Çarşı stop in Etiler area and continues until Basın Sitesi stop. The least intensity is observed in Rumelihisarüstü area.

The nodes where buses are stopping the most, according to labelled data, are plotted in Figure 4.2. The first row shows only off-peak hours, the second one shows peak hours and the last row shows the cumulative. Additionally, locations of the bus stops are plotted on each row in order to both differentiate the reason of stopping and compare the rows easily.

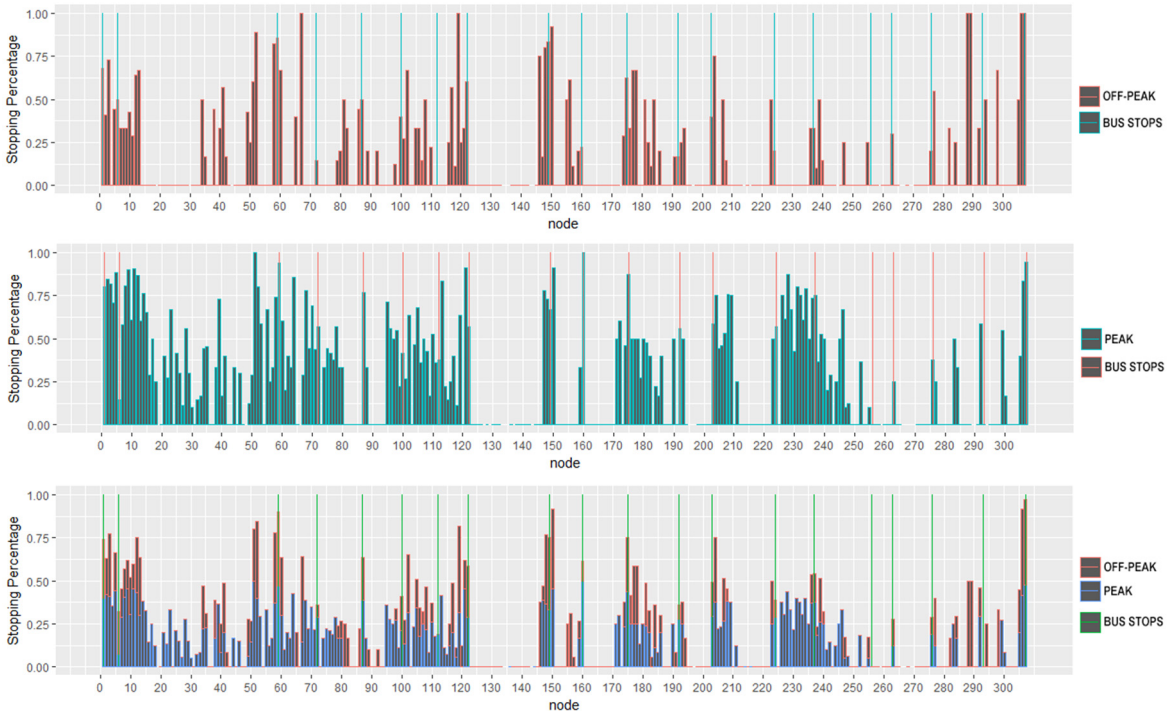

Figure 4.2. Stopping points on nodes of 43R route during peak, off-peak and cumulative time windows

Public transportation vehicles are expected to behave differently than private vehicles; hence, their stopping behaviors are different. However, a detection of an unexpected behavior of a bus outside bus stop zones may indicate a traffic problem concerning all vehicles. For reaching a general conclusion for the network traffic, the underlying reasons of the stopping behavior should be identified. For the sake of brevity of this study, effects of bus stops and number of lane changes will be investigated.

4.1.1. Bus stop effects on stopping points of buses

With more than 300 nodes, 43R's route is divided into pieces whose link size is 50m at maximum. The connection nodes between those pieces is provided in one axis of Figure 4.2. The node which is close to all the locations where buses stop is determined and shown as a bar chart. The number of stops made on a "node" is divided by "n, that is the number of journeys on that node" shows the height of the bar chart. The journeys made during off-peak hours (11 a.m.-12 p.m.), peak hours (5-6 p.m.) and their cumulative is shown in 3 different graphics. The thin vertical lines in different colors indicate the bus stops.

Bus stops are obvious reasons of stopping behavior of public transportation vehicles. It is a primary discrepancy from private vehicles data, which needs to be investigated in detail. As seen in Figure 4.2, there is a significant likelihood in the data; a lot of stopping point peaks on bus stop indicators. However, there are stopping point peaks that are located on places other than bus stops, even in off-peak periods.

4.1.2. Number of lane change effects on stopping points of buses

The reduction in the number of lanes causes bottleneck in the traffic. Public transportation vehicles also experience the bottleneck conditions if they are not separated from the road.

In Figure 4.3, peak, off-peak and cumulative stopping nodes' distributions are plotted if the occurrence is more than 50%. In other words, if a node is experiencing any stopping frequently, a vertical line is drawn on the plot. The stopping data are filtered by 0.5, and if both peak and off-peak data intersect on a node, it is labelled as "Both". Black dots represent the bus stops.

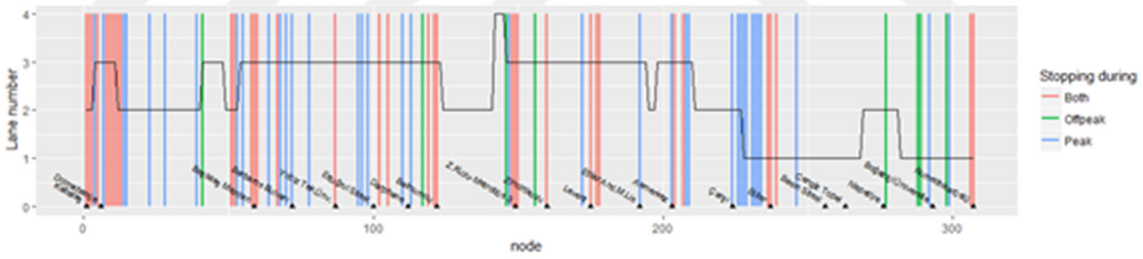

Figure 4.3. Number of lanes of route lines and stopping points during peak and off-peak hours with corresponding node points.

The areas where the buses stop the most are shown with the bus stops and the number of lanes. If there were frequent stops both in peak and off-peak hours, they were shown in red; if there were frequent stops only in off-peak hours, with green; if only in peak hours, with blue. Frequent stops are taken from the graphics in Figure 4.3, and the frequency of the stops in nodes are filtered with a minimum of 50%.

The number of lanes decrease 8 times on the route. Hotspots, namely stopping points of the buses, are mostly located near drops in the number of lanes as seen in Figure 5. The first congestion area started at the beginning of the route is ended just after the location where the number of lane decreases from 3 lanes to 2 lanes, likewise hotspots near the third such place near Balmumcu stop and sixth hotspot after Akmerkez. It is reasonable to conclude that the traffic slows down due to the number of lane reductions near these areas and the traffic lights.

On the other hand, bus behavior is different on other locations where the number of lane is reduced in the route. For example, the second such place before Beşiktaş Meydan stop and the fourth one before Zincirlikuyu do not seem to cause any hotspots before that. However, it can be seen that there are stopping points near the bus stops that are some of the most congested bus stops on the bus route and near the traffic lights.

The hotspot area between Çarşı and Etiler is in a congested region where the hotspots are at the peak hours only. In this location, the reduction in the number of lane has no adverse effect on the traffic condition except during peak hours due to lack of capacity. Clearly, 1-lane road is not sufficient for that specific traffic demand on peak hours. That is why a new subway system is recently built for this area.

4.2. Dwell Pattern Clustering

The bus route 30A is selected for this analysis because it consists of a variety of traffic density and the number of lanes. It is a ring route between two densely populated areas, namely Mecidiyeköy and Beşiktaş, passing through a network,

consisting of 3-lane arterial, 1- and 2-lane local roads. The selected ring route and its bus stops are shown on Figure 4.4.

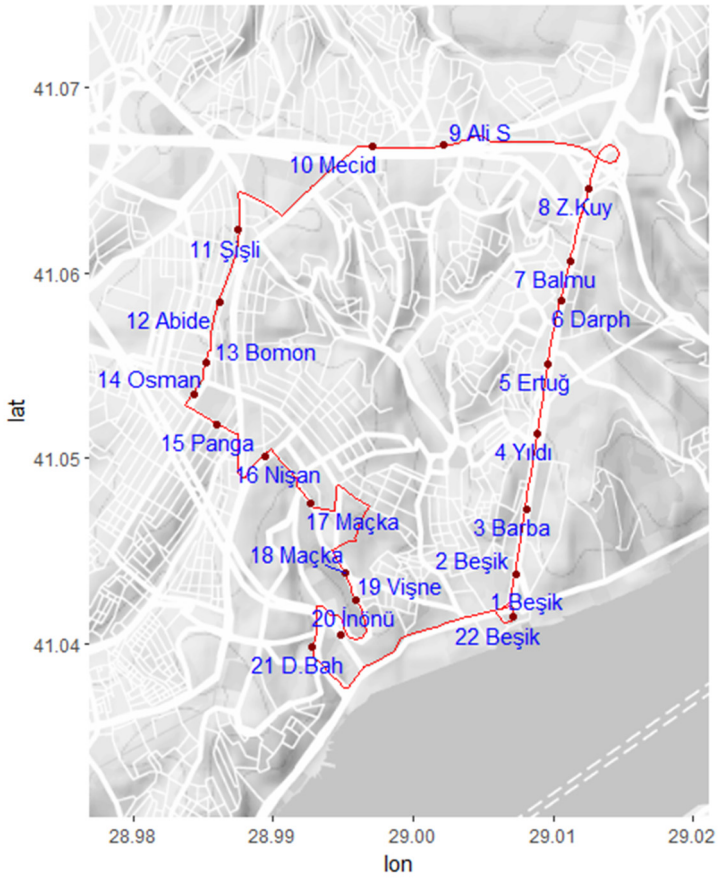

Figure 4.4. The selected bus route (Beşiktaş-Mecidiyeköy) and its bus stops.

The selected bus route is analyzed using R-Stats software. 421 data files of single daily bus logs are recorded in 30 days. On that specific route, mostly 14 different buses are operating each day. Each data log has five thousand data points on average, and in total, there are 2.1 million data points. Data points are collected approximately 4 times in a minute, the median of the sample rate is 0.0625 Hz (16 second).

Distance between geographic (geodetic) coordinates of successive data points is calculated based on an ellipsoid (spheroid) model of the world. WGS84 ellipsoid

model is used in the calculation, which is using inverse geodesic function in "Geosphere" package of R. Then, using the time and position differences, average speed values are calculated for each data point.

All data points, which have less speed value than threshold speed V_S that was calculated as 5.9 km/h, are labelled as stopped. With longer stopping cases, average bus speeds will be decreasing. Thus, it is claimed that data with less than V_S gives stopping of the buses. Consecutive stopping point durations are summed up to determine durations of moments when buses stop.

The explained analysis are applied to all bus trips. Then, the stoppings are matched to bus stops to find dwelling patterns. The data of each bus stop is smoothed with LOESS (Local Polynomial Regression Fitting) model. Polynomial surfaces are fit using local fitting to find out the dwelling patterns of each bus stops.

All dwelling patterns of bus stops on selected route are plotted starting from 6AM to midnight in Figure 4.5. The first plot (a) shows dwell patterns of all bus stops on the selected route. The bus stops are selected as outliers when their dwell time is exceeding one minute at any time (b). Figure 4.6 shows regular bus stop patterns except outlier ones.

Most of the stops have increased dwell times in the evening peak starting from 4PM to 7PM, and they also have peak dwell times around 6PM. When some of them have saddle point of dwell times around 1PM, some others reaches their peak points. Although a significant increase of dwell pattern is observed on one bus stop at the end of the day, the rest of them have decreasing pattern around 11PM.

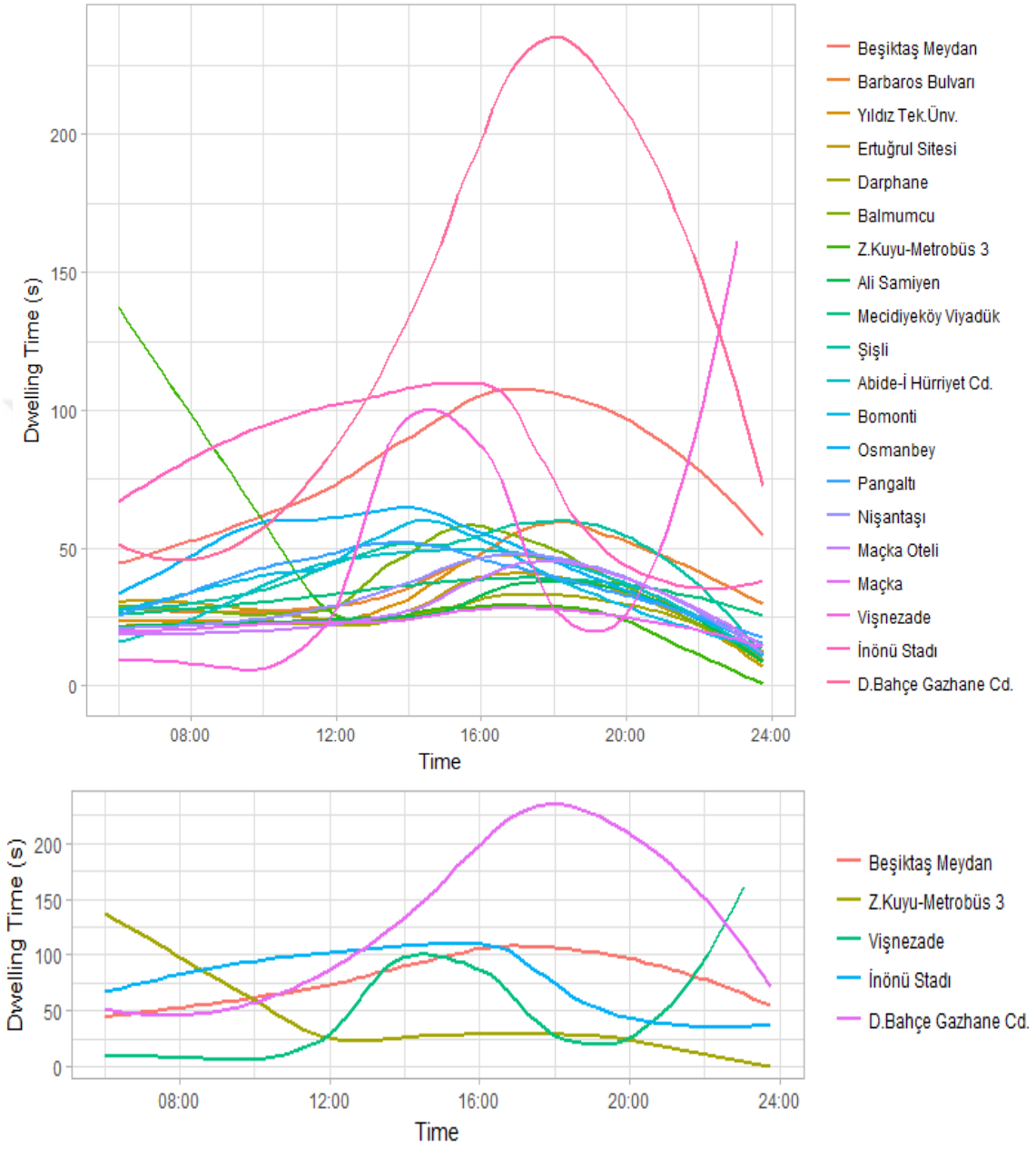

Figure 4.5. (a) Dwelling patterns of bus stops on selected route (b) Dwell patterns of outlier bus stops

Dwelling pattern similarities are determined using hierarchical clustering method. Euclidian distances shown in Equation 4.1 among bus stops´ patterns are calculated for each moment. The patterns are clustered using maximum of the calculated distances, shown in Equation 4.2.

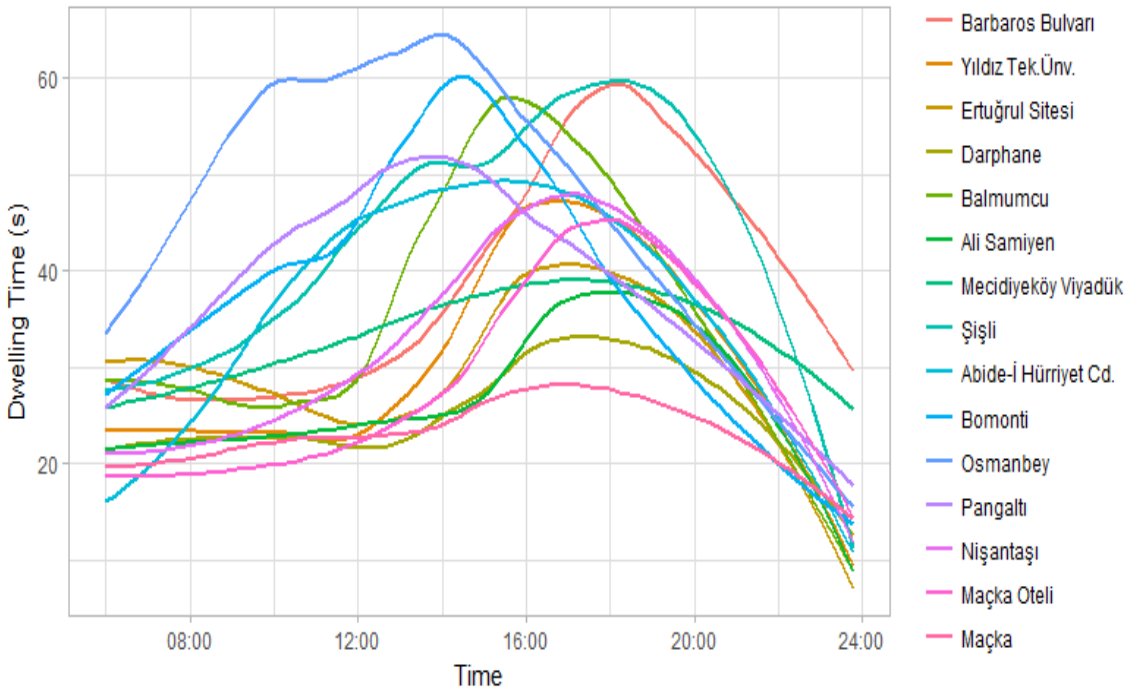

Figure 4.6. Close up to dwell patterns after extracting outlier bus stops

$$\|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2} \tag{4.1}$$

$$\max\{d(a,b): a \in A, b \in B\} \tag{4.2}$$

The dendrogram of the clusters is shown in Figure 4.7. When using 7 clusters, all outliers are separated from regular bus stops. While the cluster number 1, 4, 5, 6 and 7 are outlier clusters, each with 1 member, the cluster number 2 and 3 includes bus stops with regular patterns.

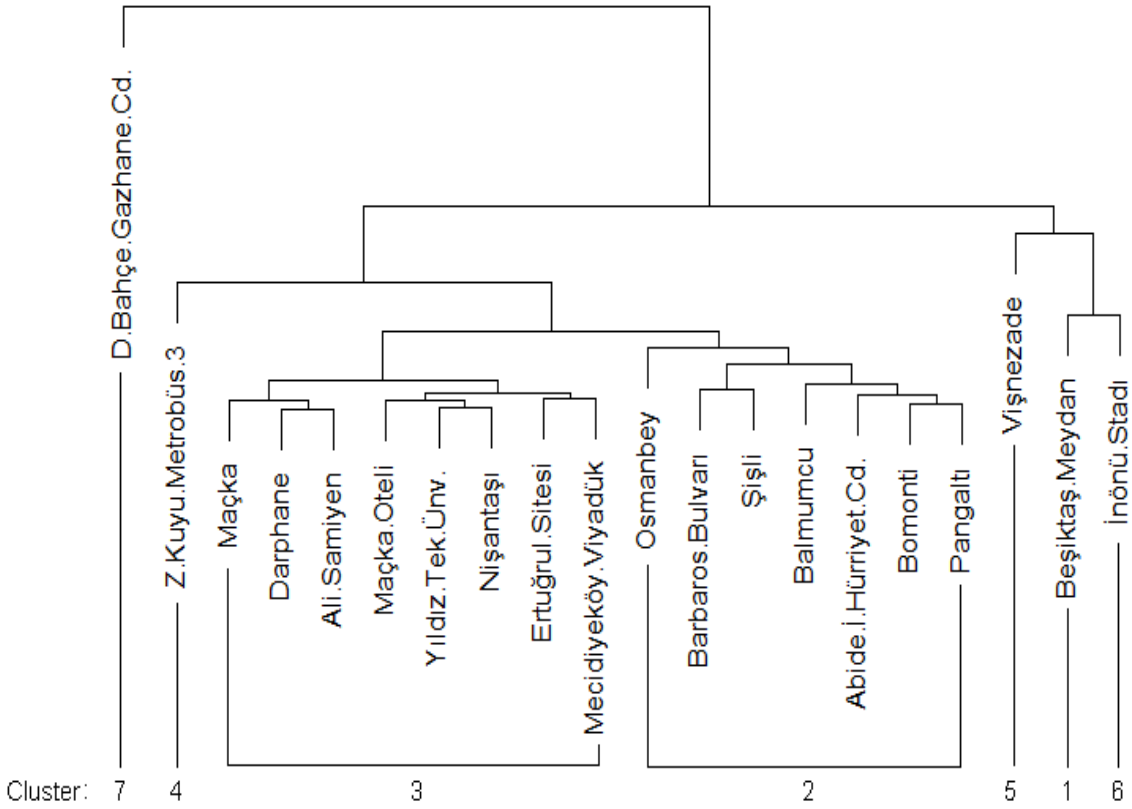

Figure 4.7. Bus dwell patterns cluster dendrogram.

4.3. Bus Stop Clustering

All buses in the selected bus route are almost identical. They have one door for entrance which allows only one person to board and two exit doors which allow two or three people to alight. Thus, only the network information around the bus stops and the demand are considered as attributes.

The first stop (number 1 and 22) is the terminal station of the selected ring route. The terminal station has long waiting times in order to fit on schedules of the route. Since these waiting times are different from dwelling times, the first stop is eliminated from the analysis.

Table 4.1. Bus stop attributes.

No	**Name**	**Number of Lanes**	**Pocket**	**Passenger Demand**[1]
2	Beşiktaş Meydan	3	1	858
3	Barbaros Bulvarı	3	1	695
4	Yıldız Tek.Ünv.	3	1	801
5	Ertuğrul Sitesi	3	0	708
6	Darphane	3	0	690
7	Balmumcu	3	0	729
8	Z.Kuyu-Metrobüs 3	3	1	334
9	Ali Samiyen	1	0	757
10	Mecidiyeköy Viyadük	4	1	868
11	Şişli	2	0	853
12	Abide-İ Hürriyet Cd.	1	0	676
13	Bomonti	1	0	569
14	Osmanbey	2	1	635
15	Pangaltı	1	0	850
16	Nişantaşı	2	1	665
17	Maçka Oteli	1	0	567
18	Maçka	1	0	494
19	Vişnezade	1	0	209
20	İnönü Stadı	3	0	160
21	D.Bahçe Gazhane Cd.	3	1	331

[1] Transportation card use of the selected bus route in April 2016

After a bus finishes a trip, it waits in the terminal until its next schedule. Near the scheduled time, 5-10 mins before the time, it opens the door to allow passengers to board. The time usually much more than enough for all passengers to board. If the bus driver sees an extraordinary demand in the terminal, it opens the door even earlier so that there will be no delay in schedule.

In the selected route, although majority of the bus stops locate in 3-lane road, 35% of them are in one lane road. 40% of bus stops have separated pocket. More than 13.3 thousand people, and 622 people per bus stop, are using this bus route monthly when including the terminal station (bus stops number 1) which is outside the scope of this analysis. Dwell pattern clusters and characteristic clusters are compared in Figure 4.8.

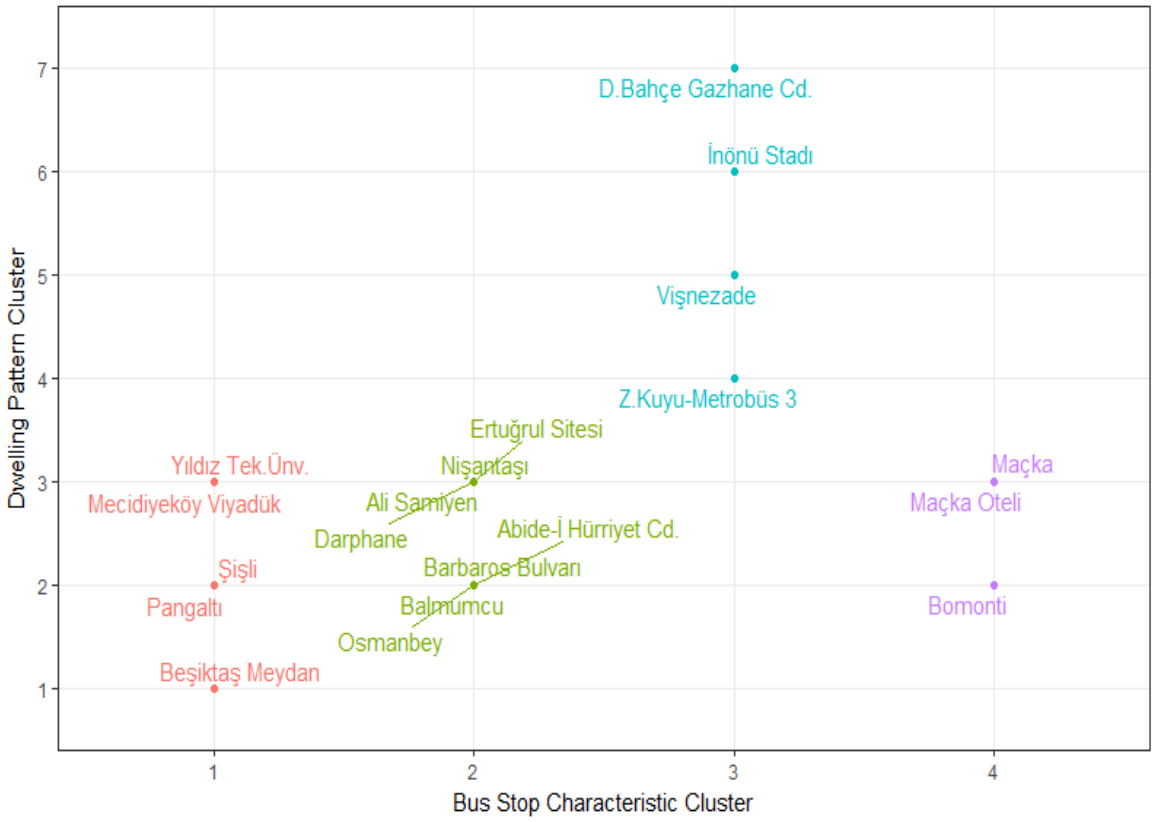

Figure 4.8. Bus dwell patterns cluster and characteristic cluster comparison.

Bus stops characteristic are clustered using hierarchical clustering method. When using 4 clusters, all outliers are separated from regular bus stops except the first stop and the outliers are clustered together in characteristic cluster number 3.

4.4. Bus Stop Influence Distance

Matrix of bus stop attributes of 12 bus lines in Istanbul are prepared with the help of online street view services, as explained in the section 3.6. Along these lines, there are 451 different bus stops. After eliminating terminal stops, the data set is prepared for 438 bus stops. 15 characteristic information of each bus stop is recorded. These are number of lanes where the bus stop locates, whether the bus stop has a pocket, passenger demand on the bus stop and before and after distances from the bus stop to the nearest drops in the number of lanes and increase of the number of lanes, traffic light, crossroad, roundabout, and entrances.

Matrix of bus stop attributes of 12 bus lines in Istanbul which are selected for the most variety are prepared with the help of online street view services. 14 information of each bus stop is recorded. Firstly, the bus stop pockets are analyzed. 217 of the bus stops (49.54%) do not have any separated pocket, the remaining (221) (50.46%) has a pocket. Majority of the bus stops are located near roads having two or less lanes (73.97%). The distribution of bus stops according to number of lanes are shown in Table 4.2.

Table 4.2. The distribution of bus stops according to the number of lanes

Number of Lanes	**1**	**2**	**3**	**4**
Count	149	175	91	23
Percentage	34.02%	39.95%	20.78%	5.25%

The bus stops with and without pockets are equally distributed on one lane roads. Almost 60 percent of the bus stops do not have a pocket on 2-lane roads. On the other hand, the percentage is reversed on 3-lane roads, 65 percent of bus stops have a pocket. For 4-lane roads, this percentage is increased to 87 percent. The relationships of number of lanes and pocket of bus stops are shown in Figure 4.9.

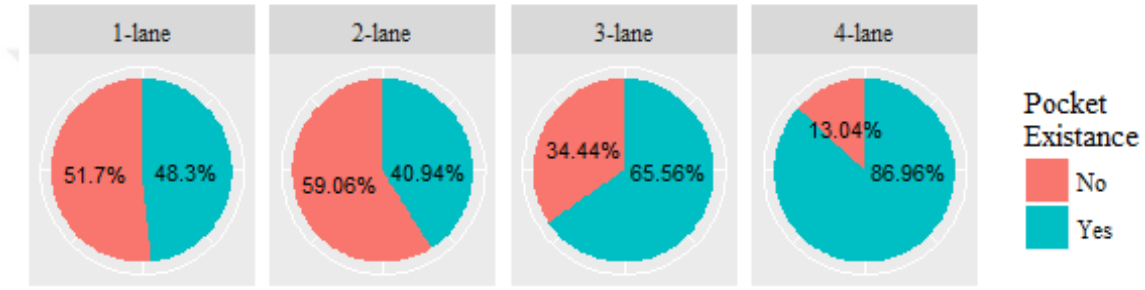

Figure 4.9. The relationship of the number of lanes and pocket of the bus stops

The passenger demands of the bus stops are determined using the transportation smart card (İstanbulkart) data. Monthly data of the selected bus routes are aggregated to determine monthly passenger demand for each bus stop, the details are explained in the section 3.7. Demands of the bus stops in Besiktas area are shown on map in Figure 4.10.

Passenger demands of the bus stops are affected by the area properties around the bus stops. The demands are higher on the terminal stops, and the bus stops near arterial streets and dense residential and commercial areas. Touristic places like near Bosphorus and Historical Peninsula have more public transportation demand.

Average demand of the selected bus stops is 347 people per bus stops, with a maximum value of 2105 people. Boxplots of the passenger demands separated with

number of lanes are shown in Figure 4.11. Most of the data set is between the ranges of 100-500 people although there are outliers reaching above 1000 people.

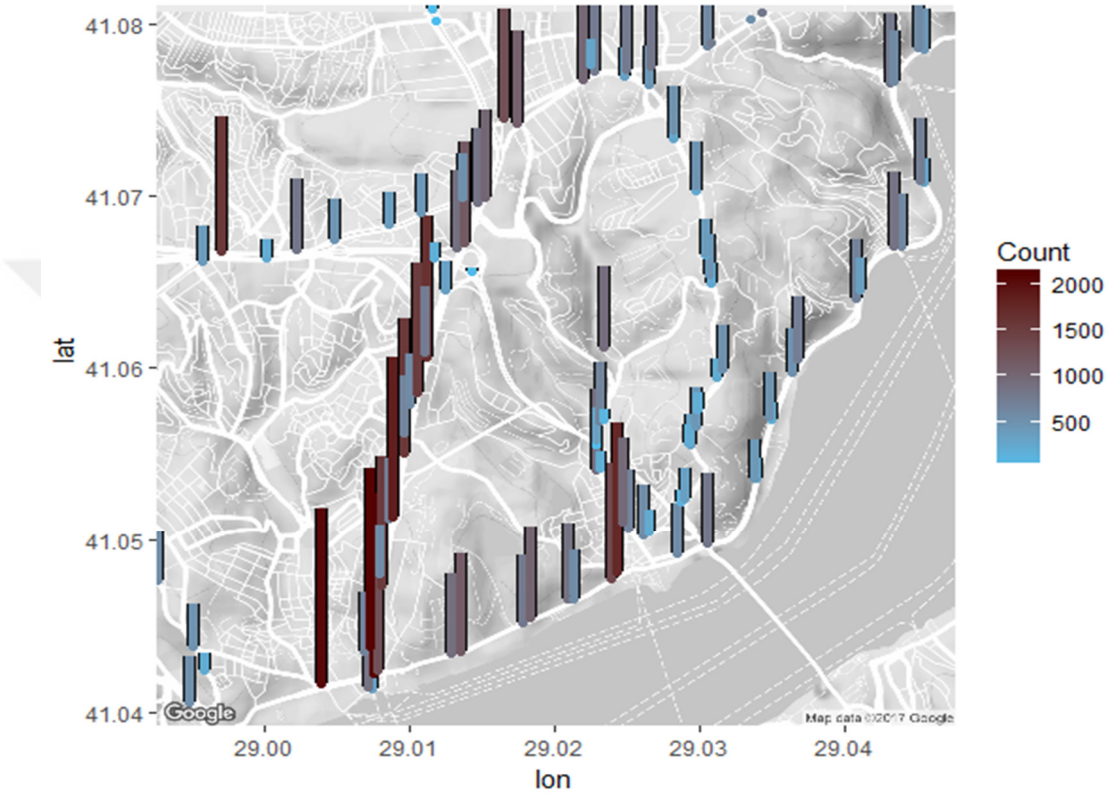

Figure 4.10. Demands of the bus stops in Besiktas area

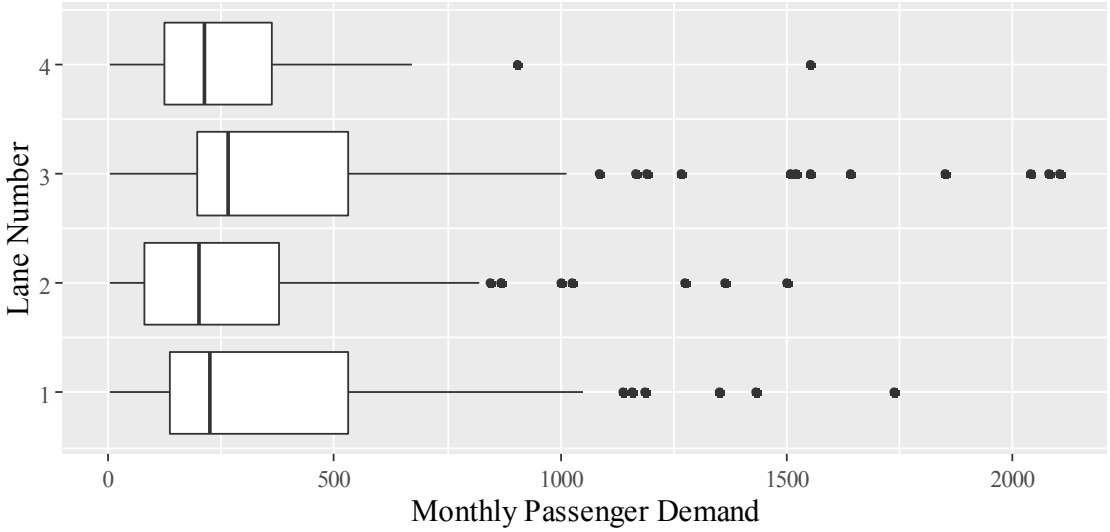

Figure 4.11. Boxplots of monthly passenger demand vs the number of lanes.

Although there is no direct relationship between the passenger demands and the number of lanes, one and three lane roads have slightly more demand. Demand increases around three-lane roads can be due to the fact that most of the road between commuting areas are upgraded to have three-lane roads due to high congestion. On the other hand, the old structure of historical areas prevents roads to develop, despite its touristic demand.

Fourthly, the interruptions before and after the bus stops are separately investigated. Distance from the interruptions to the bus stops are recorded. For demonstration purpose, the records are separated into 3 groups; 0-75 meter, 75-150 meter, and 150-225 meter. Number of each interruption in the groups are shown in Figure 4.12. Before the bus stops are represented with negative values on the left sides of the charts.

GPS measurements of buses are used to calculate the speed distribution of along the bus routes. The detail process is explained in the section 3.2. The speed distribution of buses has a decrease around the bus stops due to waiting passengers. Although the speed of buses decreases near each bus stop, the effect varies. The drop causes a sink like pattern on the speed distribution.

The speed values start decreasing at a point near and before the bus stops. After passing the bus stop, the values are increasing until a point. The starting point where the pattern starts decreasing and the ending point where the pattern ends increasing are the important points of bus stops. With using these points, the influence of the bus stops can be demonstrated.

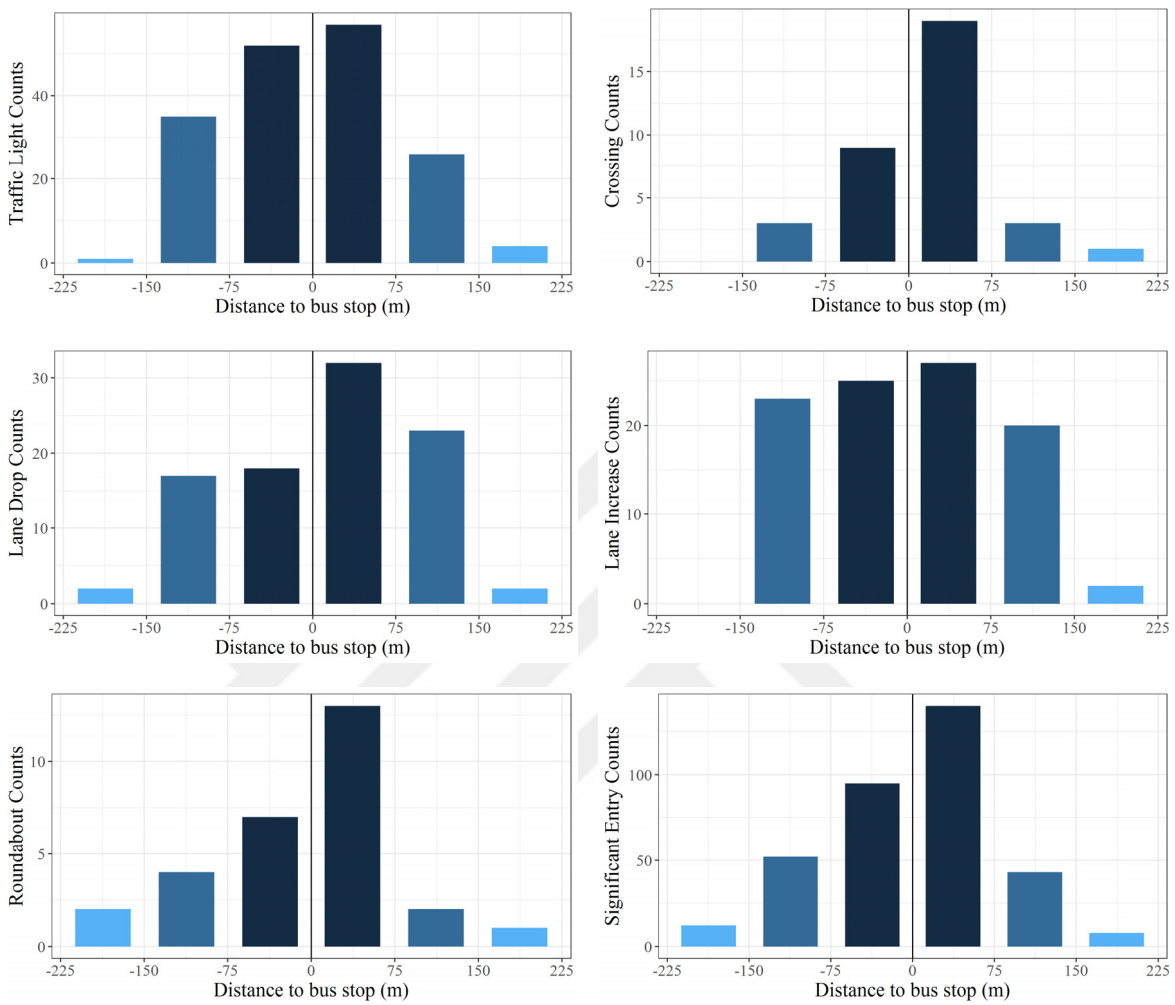

Figure 4.12. Interruptions before and after the bus stops: (a) Traffic lights, (b) Crossings, (c) Drops in the number of lanes, (d) Increase of the number of lanes, (e) Roundabouts, (f) Significant entries

The starting and ending point for each bus stop are determined using Fused Lasso method. The distance between these points are calculated and named as "Influence Distance" of bus stops. The process details are explained in the section 3.8. After calculating influence distances, the histogram of the influence distances of the bus stops are shown in Figure 4.13.

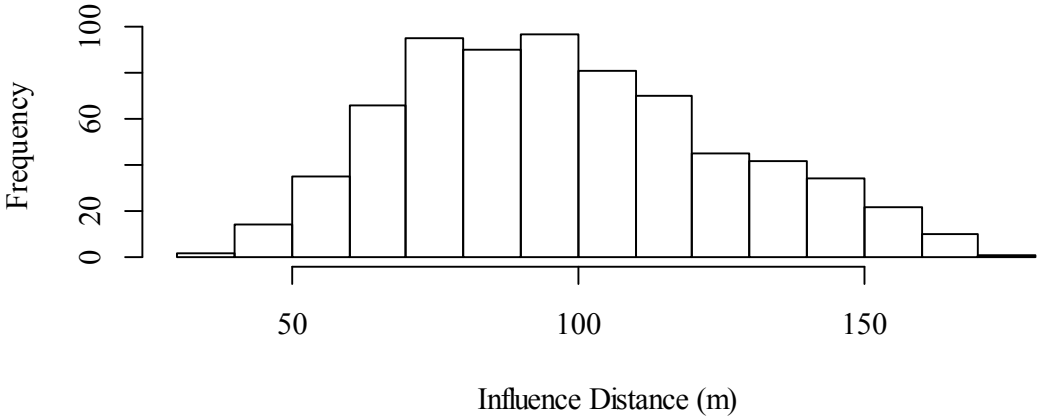

Figure 4.13. Histogram of influence distances of bus stops

Beside bus stops effect on the speed distribution of buses, other parameters may affect influence distance of bus stops. Firstly, the relationship between influence distances and number of lanes of the road is investigated, in addition to pocket existence. For each number of lanes and whether it has pocket, a boxplot of influence distance are drawn, in Figure 4.14. Median values and third quantile values of influence distance have a positive correlation with number of lanes.

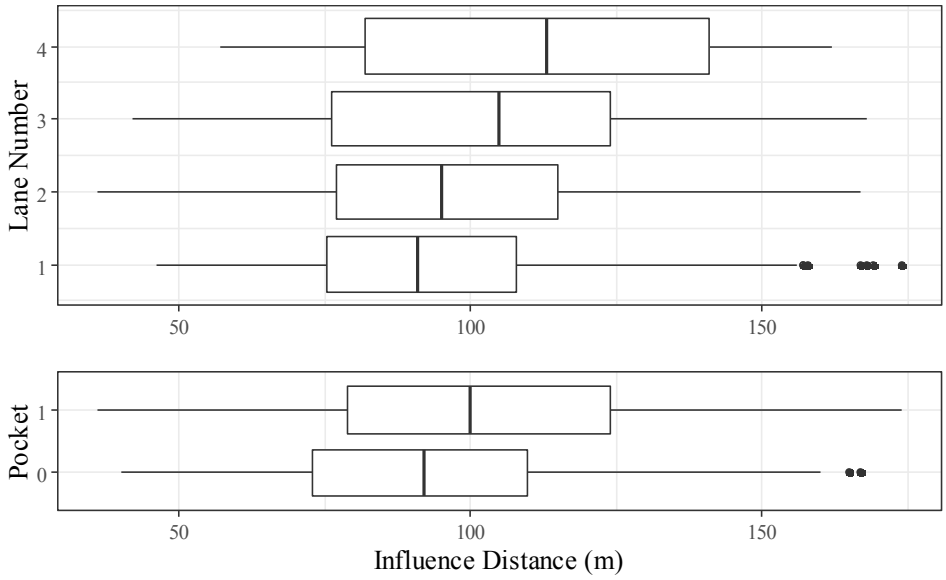

Figure 4.14. Boxplots of influence distance of bus stops grouped by (a) number of lanes and (b) pocket

This relationship can be explained with speed increase in multilane roads that causes buses to start slowing down earlier. Another explanation can be made related with driver behavior of lane changing. In right-hand traffic, the bus stops are usually at the right side of the road. It results in lane changes to reach the bus stops if it is not driven at the right lane. The bus can slow down earlier if it needs to change more lane.

As a next step, the relationship between public transportation demand and influence distance is investigated. There can be seen a slightly bolded area when the demand values are plotted on logarithmic scale with influence distance values on x scale, in Figure 4.15. Although a simple linear distribution fitted on data points, it cannot be said that it is a good model, it has low adjusted r-squared value (3.5%). However, it can be claimed that there is an increasing trend on the influence distances with the demand counts.

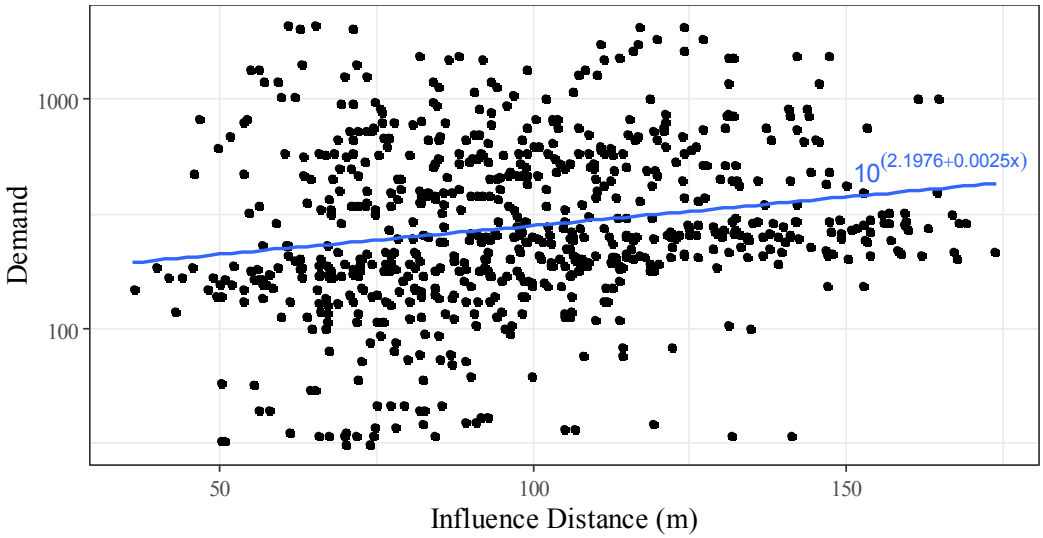

Figure 4.15. Influence distance with demand counts on logarithmic scale

The relationship of distances to the nearest traffic lights and entrances with the influence distances are displayed in Figure 4.16. It cannot be seen any significant relationship among them when only two variables are considered. These interruptions are the most common ones among the interruptions whose distances is recorded.

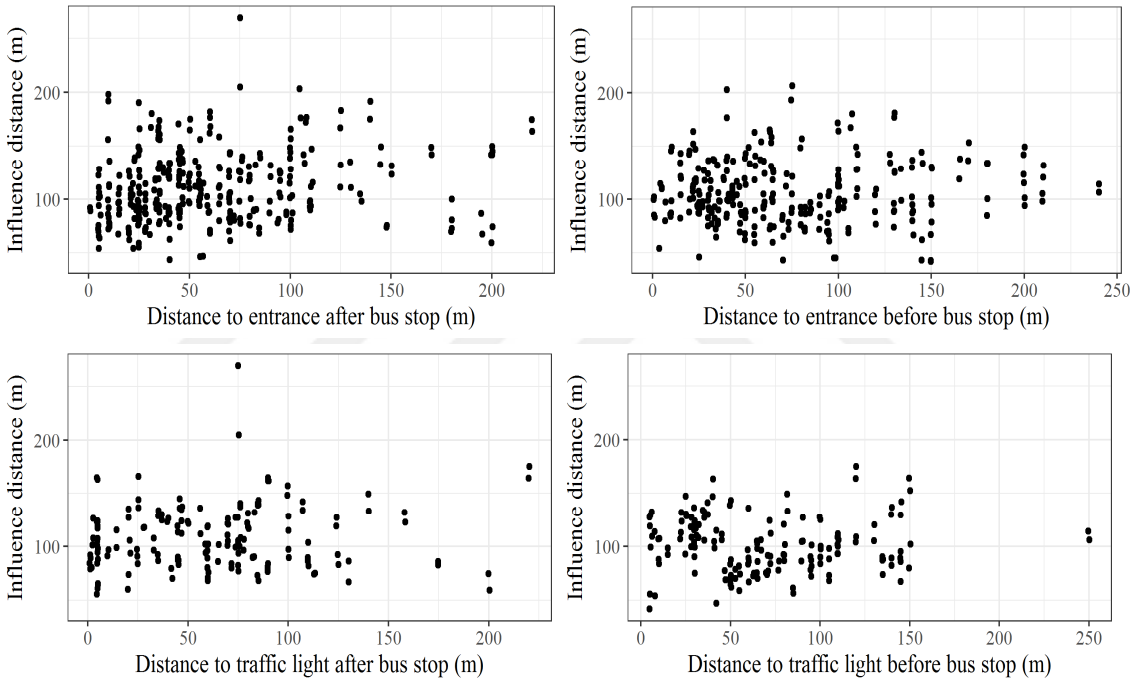

Figure 4.16. Distance to the nearest (a, b) entrances and (c, d) traffic lights, and the influence distances

The other interruptions, listed as crossing, roundabouts, drops in the number of lanes and increase of the number of lanes, are relatively less common than traffic lights and entrances. Therefore, there are not enough data to generalize the relationship among them. However, they are plotted in Figure 4.17.

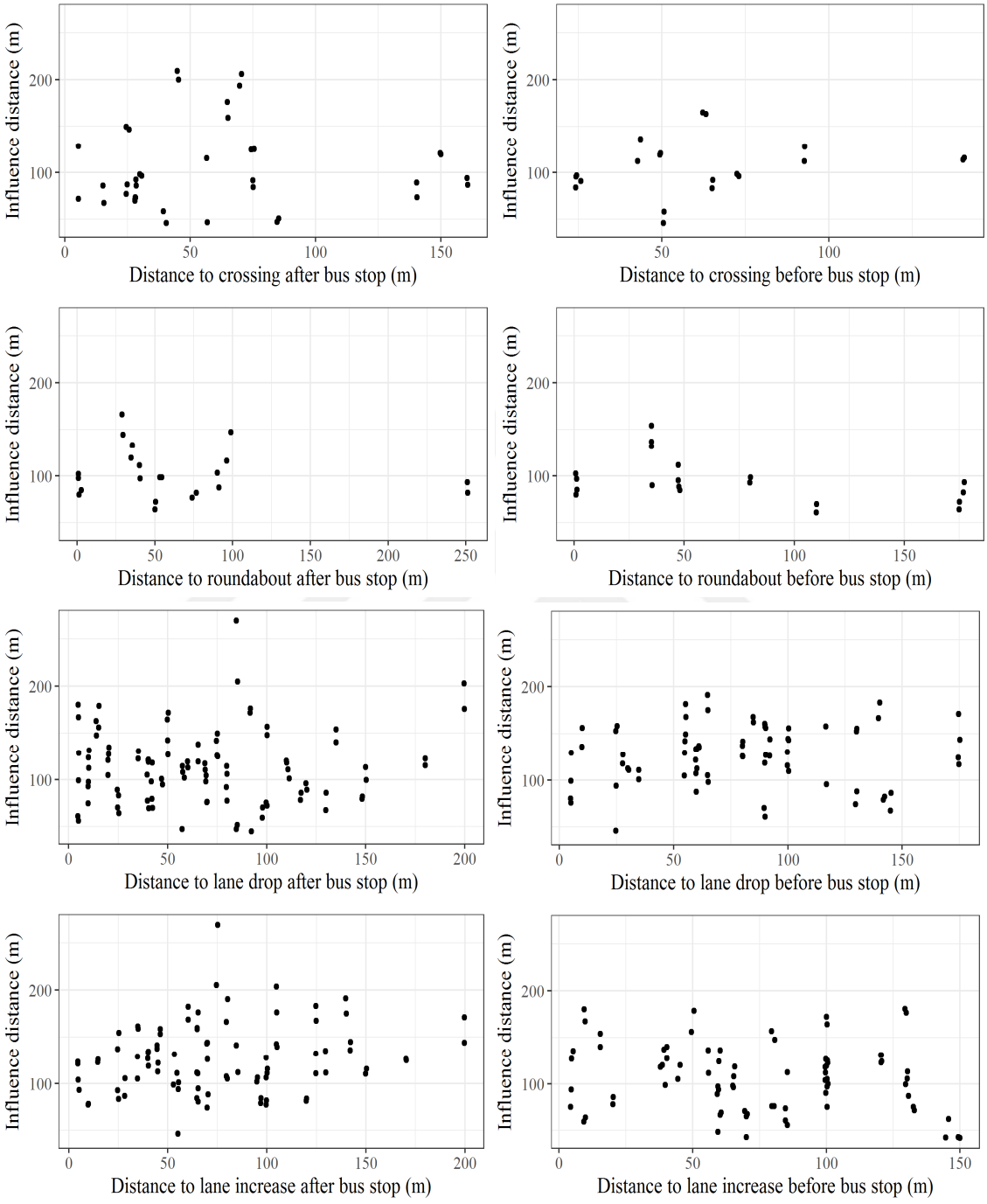

Figure 4.17. Relationship of the other interrupts: (a, b) Crossing, (c, d) Roundabout, (e, f) Drops in the number of lanes, (g, h) Increase of the number of lanes.

4.4.1. M5P model results

The data set of bus stops has lots of missing values for those variables mentioned in previous paragraph. Additionally, the other observed relationships does not seem linear. For these reasons, tree based models can be claimed to work better than linear models. Firstly, M5P model is performed on the data set, which fits linear models on data split with decision tree. The model is applied 10 fold cross-validation to prevent algorithm from overfitting. The cross-validation summary is shown in Table 4.3.

Table 4.3. Cross-validation summary of the M5P model

Correlation coefficient	0.539
Mean absolute error	18.8906
Root mean squared error	23.7013
Relative absolute error	81.75%
Root relative squared error	84.22%

The M5P model splits the data with demand count values (Count) as shown in the log file of M5P model from WEKA in Figure 4.18. Decision rule of the first split is Count>202. For data points which have "Count" values less than and equal to 202, a linear model named "LM1" is set. The remaining data are split again with Count>326 rule. For remaining data points which have "Count" values less than and equal to 326, a linear model named "LM2" is set. For the others, a linear model named "LM3" is set.

```
=== Classifier model (full training set) ===
M5 pruned model tree: (using smoothed linear models)

Count <= 0.096 (Count <= 202) : LM1 (244/71.414%)
Count >  0.096 (Count >  202) :
|   Count <= 0.155 (Count <= 326) : LM2 (184/82.62%)
|   Count >  0.155 (Count >  326) : LM3 (276/86.832%)
```

```
LM num: 1
radius =
         0.1877 * Peak
         + 0.3459 * Pocket
         + 1.5328 * Lane
         + 1.4748 * Count
         - 19.5974 * Lanedrop.before.in.75m
         + 23.3402 * Lanedrop.after.in.75m
         - 0.4558 * Laneinc.before.in.75m
         - 0.7539 * Light.before.in.75m
         - 4.4004 * Entry.after.in.75m
         + 1.2478 * Lanedrop.before.in.150m
         + 0.6035 * Laneinc.after.in.150m
         - 0.8175 * Light.before.in.150m
         - 0.4173 * Light.after.in.150m
         - 0.4093 * Entry.before.in.150m
         + 0.3387 * Entry.after.in.150m
         - 14.2456 * Entry.before.in.225m
         + 83.5638
```

```
LM num: 2
radius =
         0.362 * Peak
         + 11.3374 * Pocket
         + 30.3777 * Lane
         + 234.4514 * Count
         + 14.9477 * Lanedrop.before.in.75m
         - 0.2485 * Laneinc.before.in.75m
         - 1.7623 * Light.before.in.75m
         - 0.5766 * Entry.after.in.75m
         + 17.3342 * Lanedrop.before.in.150m
         + 19.7705 * Laneinc.after.in.150m
         - 32.3401 * Light.before.in.150m
         - 0.2275 * Light.after.in.150m
         - 1.152 * Entry.before.in.150m
         + 0.1847 * Entry.after.in.150m
         + 65.6047
```

```
LM num: 3
radius =
         0.2799 * Peak
         + 0.4886 * Pocket
         + 2.0803 * Lane
         + 0.2924 * Count
         + 1.6812 * Lanedrop.before.in.75m
         - 0.2485 * Laneinc.before.in.75m
         - 9.4878 * Light.before.in.75m
         - 13.0439 * Cross.after.in.75m
         - 0.4326 * Entry.after.in.75m
         + 1.3916 * Lanedrop.before.in.150m
         + 0.9572 * Laneinc.after.in.150m
         - 1.26 * Light.before.in.150m
         - 0.2275 * Light.after.in.150m
         - 15.8207 * Entry.before.in.150m
         + 0.1847 * Entry.after.in.150m
         + 102.6635
```

Figure 4.18. The log file of M5P model from WEKA.

The model is automatically created with WEKA software after applying 10 fold cross-validation. RMSE value of the model is 23.7 meter. The correlation coefficient (r value) is 0.539, which means r-squared is 29.1%. It cannot be claimed as a good model either, although it has better values than a linear model would have.

4.4.2. Random forest model results

As a next method, random forest is applied on data set of the bus stop attributes to predict the influence distances. Random forest cannot work with missing values, so the distance values of interruptions are grouped into three categories, as indicated in Figure 4.12. Three variables for each interruption (and for before and after cases) are added into the dataset. This dataset is named "binary". After changing the variables of interruptions, the number of variables indicating the interrupters increased from 12 to 36, and the total number of variables increases 16 to 40.

Replacing missing values with high number is carried out as another method. 300, 500 and 1000 meters are tried as replacement value. The dataset replaced with 300 meters is named "300". The others are named "500" and "1000".

Random forest algorithm is applied on these four datasets. "mtry" parameter is tried to optimized, which is number of variables randomly sampled as candidates at each split. The model is applied 10 fold cross-validation to prevent algorithm from overfitting. Results are shown as summaries in Table 4.4 for "binary" dataset, Table 4.5 for "300" dataset, Table 4.6 for "500" dataset, and Table 4.7 for "1000" dataset. The optimum values are indicated with a star.

Table 4.4. Random forest optimization table for “binary” dataset

mtry	**RMSE**	**R-squared**
2	26.3827	0.3332
4	23.6620	0.4251
6	21.9984	0.4842
8	21.1060	0.5158
10	20.4858	0.5391
12	19.9648	0.5591
14	19.7514	0.5649
16	19.4393	0.5779
18	19.2661	0.5848
20	19.1736	0.5876
22	19.0157	0.5942
24	19.0303	0.5926
26	18.9025	0.5984
28	18.8044	0.6017
30	18.7753	0.6026
*32	18.6948	0.6066
34	18.7169	0.6048
36	18.7789	0.6016
38	18.7665	0.6017
40	18.7867	0.6008

Table 4.5. Random forest optimization table for “300” dataset

mtry	**RMSE**	**R-squared**
2	25.4406	0.4232
4	22.3807	0.5421
6	21.4837	0.5719
8	20.8463	0.5954
10	20.6452	0.6017
12	20.4194	0.6102
14	20.3862	0.6110
*16	20.3585	0.6116

Table 4.6. Random forest optimization table for “500” dataset

mtry	RMSE	R-squared
2	25.4009	0.4297
4	22.1755	0.5599
6	20.8924	0.6031
8	20.2730	0.6255
10	20.0954	0.6288
12	19.8713	0.6374
*14	19.7870	0.6382
16	19.8383	0.6352

Table 4.7. Random forest optimization table for “1000” dataset

mtry	RMSE	R-squared
2	25.6025	0.4212
4	22.5102	0.5424
6	21.4047	0.5817
8	21.0480	0.5906
10	20.9303	0.5957
*12	20.7582	0.6007
14	20.7595	0.5988
16	20.8347	0.5955

One of the best values of Random forest algorithm has the lowest RMSE value as 18.7 meter and r-squared values as 60.66% from “binary” dataset with mtry value as 32 variables. Another good value from “500” dataset has the highest r-squared value as 63.83% with RMSE value as 19.8 meter. Variable importance measures of two Random forest models optimized with these values are shown in Table 4.8.

Table 4.8. Top 10 variable importance measures of two Random forest models

The "500 m" dataset	
Variable Importance Measure	
Count	232745.80
Entry.after	74113.92
Light.before	57854.14
Entry.before	53305.62
Lane	41427.75
Lanedrop.after	37870.77
Cross.after	35432.78
Laneinc.after	34192.83
Light.after	30169.21
Pocket	26044.89

The "binary" dataset	
Variable Importance Measure	
Count	250649.50
Lane	48684.09
Cross.after.75	32193.15
Pocket	19317.40
Light.before.150	18173.33
Entry.before.150	15817.80
Entry.after.75	14325.48
Light.before.75	13952.29
Lanedrop.after.75	11836.67
Lanedrop.after.150	11786.37

4.4.3. Extremely randomized trees model results

ExtraTrees algorithm is applied on these four datasets. "mtry" and "numRandomCuts" parameters are tried to optimized. "numRandomCuts" represents the number of random cuts for each randomly chosen feature. The model is applied 10 fold cross-validation to prevent algorithm from overfitting. Results are shown as summaries in Table 4.9 for "300" dataset, Table 4.10 for "500" dataset, Table 4.11 for "1000" dataset, and Table 4.12 for "binary" dataset. The optimum values are indicated with a star.

Both of the best values of ExtraTrees algorithm is from "binary" dataset. It has the lowest RMSE value as 15.96 meter and r-squared values as 70.45% from "binary" dataset with mtry value as 19 variables and 3 random cuts. Variable importance measures of two Random forest models optimized with these values are shown in Table 4.13.

Table 4.9. ExtraTrees optimization table for “300” dataset.

mtry	**Number of random cuts**	**RMSE**	**R-squared**
4	1	21.8632	0.5380
4	2	20.3565	0.5966
4	3	19.6903	0.6190
6	1	20.6362	0.5813
6	2	19.3178	0.6299
6	3	18.7399	0.6416
8	1	19.9424	0.6067
8	2	18.8506	0.6556
8	3	18.5215	0.6593
10	1	19.6916	0.6155
10	2	18.7840	0.6499
10	3	18.3401	0.6659
12	1	19.5243	0.6221
12	2	18.6894	0.6533
12	3	18.3370	0.6664
14	1	19.5541	0.6212
14	2	18.7089	0.6531
* 14	3	18.3122	0.6675
16	1	19.5412	0.6218
16	2	18.8073	0.6500
16	3	18.3330	0.6675

Table 4.10. ExtraTrees optimization table for “500” dataset.

mtry	**Number of random cuts**	**RMSE**	**R-squared**
4	1	20.7672	0.5843
4	2	19.5786	0.6401
4	3	18.9478	0.6391
6	1	19.5221	0.6309
6	2	19.5817	0.6110
6	3	18.2549	0.6628
8	1	18.9247	0.6546
8	2	18.6583	0.6491
* 8	3	18.2598	0.6630

Table 4.10. ExtraTrees optimization table for “500” dataset (cont.)

10	1	19.0266	0.6525
10	2	18.6299	0.6502
10	3	18.1722	0.6506
12	1	19.8946	0.6122
12	2	18.7202	0.6485
12	3	18.2536	0.6473
14	1	20.5591	0.6183
14	2	18.7123	0.6503
* 14	3	18.1244	0.6522
16	1	20.7884	0.6124
16	2	18.7916	0.6480
16	3	18.3536	0.6437

Table 4.11. ExtraTrees optimization table for “1000” dataset.

mtry	**Number of random cuts**	**RMSE**	**R-squared**
4	1	22.0553	0.5189
4	2	20.6312	0.5752
4	3	19.8362	0.6092
6	1	20.6583	0.5582
6	2	19.5077	0.6174
6	3	18.8477	0.6490
8	1	19.9180	0.6017
8	2	19.3726	0.6346
8	3	18.8952	0.6426
10	1	20.0239	0.5990
10	2	19.1790	0.6316
* 10	3	18.7968	0.6469
12	1	19.8822	0.5992
12	2	19.4689	0.6185
12	3	19.0338	0.6427
14	1	20.1065	0.5951
14	2	19.1982	0.6321
14	3	18.9661	0.6407
16	1	20.2606	0.5962
* 16	2	19.0692	0.6504
16	3	19.0515	0.6299

Table 4.12. ExtraTrees optimization table for “binary” dataset.

mtry	**Number of random cuts**	**RMSE**	**R-squared**
4	1	22.8551	0.3600
4	2	20.3059	0.4947
4	3	18.1587	0.6121
7	1	22.4359	0.3931
7	2	20.0142	0.5215
7	3	17.5286	0.6373
10	1	22.3388	0.4041
10	2	19.7956	0.5403
10	3	17.2005	0.6481
13	1	22.3738	0.4019
13	2	18.6994	0.5927
13	3	16.1721	0.6979
16	1	21.9758	0.4223
16	2	18.5664	0.5973
16	3	16.2228	0.6953
19	1	21.8517	0.4287
19	2	18.6523	0.5946
* 19	3	15.9566	0.7045
22	1	21.5986	0.4409
22	2	18.7800	0.5915
22	3	15.9966	0.7019
25	1	21.5512	0.4421
25	2	17.9613	0.6171
25	3	15.9795	0.7026
28	1	21.6959	0.4369
28	2	18.0100	0.6159
28	3	17.1047	0.6927
31	1	21.4081	0.4488
31	2	17.9862	0.6253
31	3	18.4122	0.5957
34	1	21.1328	0.4619
34	2	18.0851	0.6228
34	3	17.1725	0.6259
37	1	21.5006	0.4469
37	2	18.1592	0.6199
37	3	17.3394	0.6194
40	1	21.2319	0.5030
40	2	18.1751	0.6203
40	3	17.2087	0.6244

Table 4.13. Top 10 variable importance measures of two ExtraTrees models

The "binary" dataset	
Variable Importance Measure	
Count	100
Lane	20.935
Lanedrop.before.in.75	20.255
Pocket	19.997
Lanedrop.before.in.150	17.195
Light.before.in.150	15.299
Light.before.in.75	14.165
Laneinc.after.in.150	13.386
Laneinc.after.in.75	12.866
Lanedrop.after.in.225	12.727

The "500 m" dataset	
Variable Importance Measure	
Count	100
Laneinc.after	52.5564
Lanedrop.before	41.4218
Light.before	31.4112
Pocket	30.4023
Lane	23.4394
Entry.before	12.0775
Peak	7.7064
Roundabout.before	6.8708
Roundabout.after	1.5619

To sum up all models, Extremely Randomized Trees model performs better than Random Forest model by 17.11%, and M5P model by 48.5%, shown in Table 4.14. Although M5P model is seen as the worst model among the ones been tried, it shows that mixture of linear models and tree based models can perform better than those themselves.

Table 4.14. Model summaries

Model	**R-squared**	**RMSE**	**ΔRMSE**
M5P	29.05%	23.70 m	+48.50%
Random Forest	60.66%	18.69 m	+17.11%
Extra Trees	70.45%	15.96 m	-

5. CONCLUSION

In this work, bus trajectory data are analyzed to investigate the possible problems that buses are frequently experiencing along their routes. 12 distinct bus routes are selected, and daily GPS data of more than 5000 bus vehicles working on the selected bus routes in April 2016 are mined. Then, the processed data are used for several analysis.

Firstly, the speed distribution of buses along bus routes are calculated. The hotspots where buses experience recurrent slow speeds are identified. The possible reasons that creates bottlenecks for buses are argued. The areas where there is a decrease in the number of lanes before are correlated with hotspot areas.

Another dataset is created for each bus stop along the selected routes. The dataset consists of the surrounding network attributes of the bus stops. It is created using street views of the bus stops and transportation card data. Transportation card data provides the passenger boarding counts on the bus stops, which lead to the passenger public transportation demands. In street views, the number of lanes, the pocket existence and the distances to the interrupters like traffic lights or change in number of lanes are recorded.

Thirdly, the dwell times of the bus stops are calculated with using the speed distribution of buses. The dwell times are aggregated according to operating hour. The aggregated average values create patterns for each bus stops. The patterns are clustered using hierarchical clustering. Another clustering is applied for same bus stops with using the data matrix of bus stop attributes. The analysis shows that

clusters are matching, it results in correlation with dwell times and attributes of the bus stops.

The speed distribution of buses are used to determine the effects of each bus stop. The starting and ending points of the decreased speed values around the bus stops are found with using Fused Lasso method. The distance between the points for each bus stops named as the influence distance. In the bus stops where buses start to slow down earlier, the influence distance values get higher. The influence distances for all the bus stops are calculated at peak and off-peak hours.

The influence distances of all bus stops of the selected bus routes are statistically modelled with three different models. First model is chosen as M5P which implements regression models on leaves of the created decision tree. After that, the random forest and extremely random trees models are tried in order to predict the influence distance values with only using the data matrix of the bus stop attributes.

The data matrix is preprocessed into four different data matrix in order to overcome missing value problems. On these models, the binary data representation have less root mean square error than pseudo high number for missing values. If a dataset includes continues values which can be missing for some data rows, for example, there is not traffic light near every bus stop, the class binary representation for continues values is more suitable.

This thesis shows that with using network attributes near a bus stop or possible position of a bus stop, the influence distances of buses slowing down and speeding up around bus stops can be calculated. It also shows the intensity of the

possible effects to current traffic network around the bus stops due to buses. Another contribution of this thesis that can help public transportation planners is to be able track in real-time the dwell time of bus stops from processed bus GPS data which is simply explained in methodology. Besides, the problematic areas where buses frequently encounter slow speeds can be investigated, alternative routes of possible improvements on network can be suggested.

This study is limited with 12 bus lines and 450 bus stops which are along the selected bus lines. In specific to Istanbul case, they are around 10% of total bus stops. Further effects can be found when number of investigated bus lines is increased. Another point is that the data are obtained for April 2016. It cannot be worked on any seasonal effects or any interesting feature is specific to April.

For future studies, the data size should be increased so that it makes possible to split and keep another test dataset in order to compare different models on it. The more interruptions should be consider to investigate different effects combinations. The sensitivity of distance of interruptions to the bus stops should be analyzed. In order to check transferability of bus vehicle hotspot results to other vehicles' hotspots, on-site measurements should be compared with the results from bus GPS data.

REFERENCES

1. Birgitta Gatersleben, D. U., "Affective Appraisals of the Daily Commute: Comparing Perceptions of Drivers, Cyclists, Walkers, and Users of Public Transport", *Environment and Behavior*, vol. 39, no. 3, pp. 416-431, 2007.

2. Tomtom, "Traffic Index Measuring Congestion Worldwide", https://www.tomtom.com/trafficindex/, accessed at June 2017.

3. İstanbul İl Nüfus ve Vatandaşlık Müdürlüğü, "Sayılarla İstanbul", http://www.istanbulnvi.gov.tr/sayilarla-istanbul, accessed at June 2017.

4. Türkiye İstatistik Kurumu, "Motorlu Kara Taşıtları", May 2017, http://www.tuik.gov.tr/PreHaberBultenleri.do?id=24600, accessed at June 2017.

5. İETT, "İstanbul'da Toplu Ulaşım", http://www.iett.istanbul/tr/main/pages /istanbulda-toplu-ulasim/95, accessed at June 2017.

6. Arhin, S. A., E. C. Noel and O. Dairo, "Bus Stop On-Time Arrival Performance and Criteria in a Dense Urban Area", *International Journal of Traffic and Transportation Engineering*, pp. 233-238, 2014.

7. Sándor, C. and Z. Csiszár, "Method for Analysis and Prediction of Dwell Times at Stops in Local Bus Transportation", *Transport*, pp. 1-12, 2016.

8. Gökaşar, I. and Y. Çetinel, "Evaluation of Bus Dwelling Patterns Using Bus GPS Data", *Models and Technologies for Intelligent Transportation Systems*, Napoli, 2017.

9. Dimitrakopoulos, G., P. Demestichas and V. Koutra, "Intelligent Management Functionality for Improving Transportation Efficiency ny Means of The Car Pooling Concept", *IEEE Transactions on Intelligent Transportation Systems*, pp. 424-436, 2012.

10. Knaian, A. N., *A Wireless Sensor Network for Smart Roadbeds and Intelligent Transportation Systems*, Ph.D. Thesis, Massachusetts Institute of Technology, 2000.

11. Tubaishat, M., P. Zhuang, Q. Qi and Y. Shang, "Wireless Sensor Networks in Intelligent Transportation Systems", *Wireless Communications and Mobile Computing*, pp. 287-302, 2009.

12. Chen, W., L. Chen, Z. Chen and S. Tu, "WITS: A Wireless Sensor Network for Intelligent Transportation System", *Computer and Computational Sciences*, pp. 635-641, 2006.

13. Tacconi, D., D. Miorandi, I. Carreras, F. Chiti and R. Fantacci, "Using Wireless Sensor Networks to Support Intelligent Transportation Systems", *Ad Hoc Networks*, pp. 462-473, 2010.

14. Leduc, G., "Road Traffic Data: Collection Methods and Applications", *Working Papers on Energy, Transport and Climate Change*, 2008.

15. Wang, C. and H.-M. Tsai, "Detecting Urban Traffic Congestion with Single Vehicle", *International Conference Connected Vehicles and Expo*, Las Vegas, NV, USA, 2013.

16. Comert, G. and M. Cetin, "Queue Length Estimation from Probe Vehicle Location and the Impacts of Sample Size", *European Journal of Operational Research*, pp. 196-202, 2009.

17. Castro, P., D. Zhang and S. Li, "Urban Traffic Modelling and Prediction Using Large Scale Taxi GPS Traces", *Pervasive Computing*, pp. 57-72, 2012.

18. Yang, X., Z. Gao, B. Si and L. Gao, "Car Capacity Near Bus Stops with Mixed Traffic Derived By Additive-Conflict-Flows Procedure", *Science China Technological Sciences*, pp. 733-740, 2011.

19. Koshy, R. Z. and V. T. Arasan, "Influence of Bus Stops on Flow Characteristics of Mixed Traffic", *Journal of Transportation Engineering*, pp. 640-643, 2005.

20. Chien, S. I. and Z. Qin, "Optimization of Bus Stop Locations for Improving Transit Accessibility", *Transportation Planning and Technology*, pp. 211-227, 2004.

21. Saka, A. A., "Model for Determining Optimum Bus-Stop Spacing in Urban Areas", *Journal of Transportation Engineering*, pp. 195-199, 2001.

22. Dueker, K. J., T. J. Kimpel, J. G. Strathman and S. Callas, "Determinants of Bus Dwell Time", *Journal of Public Transportation*, pp. 21-40, 2004.

23. Tirachini, A., "Bus Dwell Time: The Effect of Different Fare Collection Systems, Bus Floor Level and Age of Passengers", *Transportmetrica A: Transport Science*, pp. 28-49, 2013.

24. Tibshirani, R., M. Saunders, S. Rosset, J. Zhu and K. Knight, "Sparsity and Smoothness via the Fused Lasso", *Journal of the Royal Statistics Society: Series B*, pp. 91-108, 2005.

25. Rokach, L. and O. Maimon, "Clustering Methods", *Data Mining and Knowledge Discovery Handbook*, pp. 321-352, 2005.

26. Manning, C. D., P. Raghavan and H. Schütze, "Ch 17 Hierarchical Clustering", *Introduction to Information Retrieval,* New York, NY, USA, Cambridge University Press, 2008.

27. Sneath, P. H. A. and R. R. Sokal, "Numerical Taxonomy: The Principles and Practice of Numerical Classification", *A Series of Books in Biology*, Freeman, 1973.

28. King, B., "Step-wise Clustering Procedures", *Journal of the American*, pp. 86-101, 1967.

29. Jain, A. K. and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988.

30. Xu, R. and D. Wunsch, "Survey of Clustering Algorithms", *IEEE Transactions on Neural Networks*, pp. 645-678, 2005.

31. MacKay, D. J., *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, 2003.

32. Schroff, F., A. Criminisi and A. Zisserman, "Object class segmentation using random forests", *Proceedings of the British Machine Vision Conference*, 2008.

33. Quinlan, J. R., "Discovering rules by induction from large collections of example", *Expert Systems in the Micro Electronics Age*, 1979.

34. Quinlan, J. R., "Learning efficient classification procedures and their application to chess end games", *Machine Learning*, Springer, 1983, pp. 463-482.

35. Quinlan, J. R., "Induction of Decision Trees", Machine Learning, pp. 81-106, 1986.

36. Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

37. Breiman, L., J. Friedman, C. J. Stone and R. A. Olshen, *Classification and Regression Trees*, CRC press, 1984.

38. Quinlan, J. R., "Learning with Continuous Classes", *5th Australian Joint Conference on Artificial Intelligence*, Singapore, 1992.

39. Wang, Y. and I. H. Witten, "Induction of Model Trees for Predicting Continuous Classes", *Proceedings of the Poster Papers of the European Conference on Machine Learning University of Economics Faculty of Informatics and Statistics*, Prague, 1997.

40. Breiman, L., "Random Forests", *Machine Learning*, pp. 5-32, 2001.

41. Breiman, L., "Bagging Predictors", *Machine Learning*, pp. 123-140, 1996.

42. Friedman, J. H., "Greedy Function Approximation: A Gradient Boosting Machine", *Annals of Statistics,* pp. 1189-1232, 2001.

43. Xu, R., *Improvements to random forest methodology*, Ph.D. Thesis, Iowa State University, 2013.

44. Geurts, P., D. Ernst and L. Wehenkel, "Extremely Randomized Trees", *Machine Learning*, pp. 3-42, 2006.

45. Wehenkel, L., "Machine Learning Approaches to Power-System Security Assessment", *IEEE Expert*, pp. 60-72, 1997.

46. Geurts, P., *Contributions to Decision Tree Induction: Bias/Variance Tradeoff and Time Series Classification*, Ph.D. Thesis, University of Liège, Belgium, 2002.

47. IBB; Waag Society, "CitySDK Istanbul Endpoint", http://devcitysdk.ibb.gov.tr/, accessed at June 2017.

48. Quddus, M. A., W. Y. Ochieng and R. B. Noland, "Current Map-Matching Algorithms for Transport Applications: State-Of-The Art and Future Research Directions", *Transportation Research Part C: Emerging Technologies*, pp. 312-328, 2007.

49. Karney, C. F., "Algorithms for Geodesics", *Journal of Geodesy*, pp. 1-13, 2013.

50. Kirchner, M., P. Schubert and C. T. Haas, "Characterisation of Real-World Bus Acceleration and Deceleration Signals", *Journal of Signal and Information Processing*, p. 8–13, 2014.