REPUBLIC OF TURKEY

AKDENİZ UNIVERSITY

OPTIMAL DESIGN OF PUBLIC TRANSPORTATION LINES USING ARTIFICIAL INTELLIGENCE METHODOLOGIES FOR ANTALYA

Barış Doruk BAŞARAN

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER OF SCIENCE THESIS

JUNE 2022

ANTALYA

REPUBLIC OF TURKEY

AKDENİZ UNIVERSITY

OPTIMAL DESIGN OF PUBLIC TRANSPORTATION LINES USING ARTIFICIAL INTELLIGENCE METHODOLOGIES FOR ANTALYA

Barış Doruk BAŞARAN

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER OF SCIENCE THESIS

JUNE 2022

ANTALYA

REPUBLIC OF TURKEY
AKDENİZ UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

OPTIMAL DESIGN OF PUBLIC TRANSPORTATION LINES USING ARTIFICIAL INTELLIGENCE METHODOLOGIES FOR ANTALYA

Barış Doruk BAŞARAN

DEPARTMENT OF COMPUTER ENGINEERING

MASTER OF SCIENCE THESIS

This thesis was accepted unanimously by the jury on 17/06/2022.

Prof. Dr. Melih GÜNAY (Supervisor)
Assoc. Prof. Dr. Alper BİLGE
Asst. Prof. Dr. Kamer ÖZGÜN

ÖZET

YAPAY ZEKA YÖNTEMLERİ KULLANARAK TOPLU TAŞIMADA ANTALYA İÇİN OPTIMUM HAT TASARIMI

Barış Doruk BAŞARAN

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Melih GÜNAY

Haziran 2022; 65 sayfa

Toplu taşıma, bir şehrin en önemli hizmetlerinden biridir. Şehirler geliştikçe toplu taşımaya olan taleple beraber toplu taşımanın hizmet alanı da genişleyecektir. Talep fazlası ve genişleme ihtiyacı, toplu taşıma maliyetlerinin artmasına ve yolcu konforunun da azalmasına yol açacaktır. Bu nedenle, gelişen bir şehrin toplu taşıma sistemi olabilecek en verimli ve dengeli bir şekilde güncellenmeli ve optimize edilmelidir. Gelişen veri toplama ve büyük veri teknolojileri sayesinde, toplu taşıma planlaması için daha kapsamlı optimizasyon yöntemleri araştırmak mümkündür. Bu tezde tanıtılacak yöntem ve yaklaşımlar, Frequency Setting Problem gibi çeşitli toplu taşıma planlama konularında, karar vermede yardımcı olabilecek analizler içermektedirler. Boarding Pattern Clustering yönteminin amacı, taşınan yolcu sayısından bağımsız olarak benzer günlük taleplere sahip olan otobüs hatlarının kümelenmesidir. Böylece, çeşitli planlamalarda aynı kümeye mensup olan hatlar beraber ele alınabilecektir. Time Slot Clustering yönteminde, bir hattın benzer seviyede talebe sahip zaman aralıkları kümelenmesi amaçlanmaktadır. Her bir kümedeki talep seviyelerine göre frekans ayarlanarak yolcu konforu ve sefer maliyetleri arasında denge sağlanabilecektir. Belirli bir güzergahtaki iniş tahmini için aynı hattaki karşıt güzergahtan elde edilen veriler doğrultusunda, olasılıksal bir yaklaşım denenmiştir.

ANAHTAR KELİMELER: Akıllı Kart, Boarding Pattern, Destination Inference, İniş Tahmini, Frekans Ayarlama, Kümeleme, Otobüs Doluluğu, Toplu Taşıma, Zaman Serisi

JÜRİ: Prof.Dr. Melih GÜNAY
Doç.Dr. Alper BİLGE
Dr. Öğr. Üyesi Kamer ÖZGÜN

ABSTRACT

OPTIMAL DESIGN OF PUBLIC TRANSPORTATION LINES USING ARTIFICIAL INTELLIGENCE METHODOLOGIES FOR ANTALYA

Barış Doruk BAŞARAN

MSc Thesis in COMPUTER ENGINEERING
Supervisor: Prof. Dr. Melih GÜNAY
June 2022; 65 pages

Public transportation is one of the most important services for cities. The growth of the cities constantly increases the passenger demand and the covered area for the public transportation. Increase in demand and coverage causes discomfort and high cost of service. Thus, the public transportation system has to be updated in most efficient and balanced way. With advancing data collecting and big data technologies, it is now possible to research more comprehensive optimization methods for public transportation planning. The methods introduced in this thesis can be helpful in various public transportation planning decisions regarding mostly bus lines such as Frequency Setting Problem. The aim of Boarding Pattern Clustering is clustering the bus lines with similar demand patterns through a regular day regardless of its popularity. Bus lines in same clusters can be considered together in various plannings. In Time Slot Clustering, various time slots that similar demands are determined and each time slot can have its own frequency setting to balance cost and comfort more dynamically. Alighting counts and bus occupancies of a route are estimated with probabilistic approach by using the data belongs to its reverse route.

KEYWORDS: Alighting Estimation, Boarding Patterns, Bus Occupancy, Clustering, Destination Inference, Frequency Setting, Public Transportation, Smart Card, Time Series

COMMITTEE: Prof.Dr. Melih GUNAY
Assoc.Prof.Dr. Alper BILGE
Asst.Prof.Dr. Kamer OZGUN

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Dr. Melih GÜNAY and Asst. Prof. Dr. Kamer ÖZGÜN for their guidance and would like to thank all my teachers for their precious lessons. Most importantly, I am grateful to my family for their support.

LIST OF CONTENTS

TEXT OF OATH

I declare that this study "Optimal Design of Public Transportation Lines using Artificial Intelligence Methodologies for Antalya", which I present as master thesis, is in accordance with the academic rules and ethical conduct. I also declare that I cited and referenced all material and results that are not original to this work.

17/06/2022
Barış Doruk BAŞARAN

ABBREVIATIONS

CSV	: Comma Separated Values
ID	: Identifier
SQL	: Structured Query Language
BPC	: Boarding Pattern Clustering
ED	: Euclidean Distance
DTW	: Dynamic Time Warping
BD	: Band Distance
TSC	: Time Slot Clustering
STSC	: Stepped Time Slot Clustering
ATSC	: Adaptive Time Slot Clustering
BSG	: Bus Stop Group
ARR	: Aggregated Reverse Route
DRR	: Direct Reverse Route
TC	: Trip Chaining
DFM	: Distance From Mean
DFTC	: Distance From Trip Chaining
EM	: Expectation Maximization
CEM	: Conditional Expectation Maximization
OD	: Origin Destination
FS	: Fixed Schedule
GS	: Given Schedule
SS	: Stepped Schedule
AS	: Adaptive Schedule

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Public transportation is one of the most important aspects of urban cities. As the cities are constantly growing, citizens' need for public transportation increases. That creates an increasing demand which makes it mandatory to maintain the balance between passenger comfort and operating cost.

Completely redesigning a public transportation system when a city reaches a certain grow rate might be also an option. But redesigning the system may be inefficient and cause passenger discomfort. Thus, small fixations and updates may be more efficient and effective in a long run.

Data collecting systems such as Automated Fare Collection (AFC), Automatic Passenger Counter (APC), Automated Vehicle Location (AVL), and Geographical Positioning Systems (GPS) makes it possible to research efficient and effective solutions for public transportation systems much faster.

In this thesis, various data mining approaches and methods are discussed and suggested. The discussed methods can be used as tools for maintain the optimization of public transportation systems. The data obtained from The Transportation Department of Antalya is used for the development of the discussed methods.

The transportation system data that used in this thesis was previously obtained and managed. Since methods in this thesis are much more passenger centered, the additional boarding data was required. The additional boarding data has different types of columns and additional information, besides having the same source with the previous boarding data. So, the additional boarding data cannot be stored in the same table with preexisting boarding data. Thus, additional boarding data stored in two tables as boarding and trip tables.

The boarding data mainly consist of the boardings occurred in terms when the tourist activity is low. So, the boarding data can be used to represent passenger activities in regular days. The most suitable format to represent daily passenger activity is time series format. So, date field is ignored and only time field is considered in boarding time column which is represented as date-time format. The time field is grouped by 30 minutes time intervals and the count of rows are considered as boarding count or passenger count.

Time series format that consist of boarding counts in all 48 time intervals is calculated in the result of this group-by operation. The boarding counts or passenger activity is indicates passenger demand in particular time intervals.

Daily demand of a bus line is obtained by filtering the boarding data before calculating the daily demand. In general, the daily demands of different bus lines are quite different when projected on a line plot. But some of the daily demands has similar patterns. The bus lines with similar demand patterns may be considered together in various public transportation plannings. In order to extract patterns from daily demands, the daily demand of each bus line have to be normalized according to its maximum boarding count. Thus, daily demands can be represented in a percent scale. A patterns that extracted via this method is named boarding pattern. Boarding Pattern Clustering (BPC) is a clustering approach that used for clustering the boarding patterns of bus lines.

Daily demand patterns have different areas that indicate its pattern like peaks and dips or trends. These areas are named as time slots since these areas are like slots or partitions that reside in time series. These time slots can be determined programmatically with the usage of clustering approach. Time slot clustering (TSC) approaches are used to determine these areas.TSC can be powerful tool for various transportation planning decisions such as frequency setting. Since the time slots consist of the time intervals in which the similar demands are present, each time slot may have its own frequency setting. Which have potential to result much more dynamic schedules that increase efficiency and passenger comfort in the transportation system.

The sections regarding the BPC and TSC are based on "Boarding Pattern Classification With Time Series Clustering" and "Demand Profiling of Bus Lines in Public Transportation". Since i was working on these publications with my supervisors in the thesis term, i included them in this thesis.

In the transportation system of Antalya, pricing is only based on the category of passenger (student, adult or senior). Which means, passengers only tap on boarding but on alighting. Thus, alighting data, which is a valuable source of data, is unobtainable. Also, in a city like Antalya, where the passengers want to reach destination directly without transfers, it is difficult to use Trip Chaining method. Although, alighting counts in a route can be estimated via boarding counts on its reverse route because of the same reasons.

The boarding counts from the bus stops of the opposite route are converted into alighting probabilities in alighting estimation from reverse routes. An origin destination probability matrix is produced with these probabilities. The probability matrix indicates alighting probability on an alighting bus stop for a passenger that boarded from a certain bus stop. Also, a probability matrix can be used to predict alightings of a different trip or time window.

In order to verify the reliability of the alighting estimation results, video recordings of some trips are obtained from The Department of Transportation of Antalya. Because of the camera angles are determined for security causes instead of data collection, it is nearly impossible to count boarding and alighting with image processing techniques. So, counting is done by hand and in this process none of the passengers' identity is exposed.

Another purpose of the alighting estimation is estimating the occupancy of busses at the certain bus stop of the route. Bus occupancy is another valuable information for public transportation and generally calculated with the use of boarding and alighting data. So, bus occupancy results also considered in verification phase of alighting estimation.

2. LITERATURE REVIEW

As being the one of the most important services for cities, public transportation services has to keep improving in order to meet citizens needs while the cities are keep expanding and get more populated. Researches and new planning strategies are helping public transportation to improve more efficient and effectively (Özgün et al. 2020). The researches are getting more faster and advanced because of the improving data collecting systems such as Automated Fare Collection (AFC), Automatic Passenger Counter (APC), Automated Vehicle Location (AVL), and Geographical Positioning Systems (GPS) (Harrison, Grant-Muller, and Hodgson 2020; Lu et al. 2020; Welch and Widita 2019).

The big data obtained from the data collecting systems can be used in researches and plannings that concern various types of parameters and performance measures such as comfort and efficiency (Zhu et al. 2019; Pelletier, Trépanier, and Morency 2011; Van Oort and Cats 2015). Big data platforms and systems offer various types of data cleansing, monitoring and visualizing options in order to make decisions by considering different parameters and performance measures for transportation planners and authorities (Berthold et al. 2007; Yinhai Wang and Zeng 2018). Performance measures can be expressed in different levels such as entire transit system, neighbourhoods, lines and routes, stations or passengers (Stewart et al. 2016).

Ibarra-Rojas et al. 2015 categorized the public transportation planning decision levels as:

i) Strategic: Plannings about coverage and network design

ii) Tactical: Plannings about trip frequency and schedules

iii) Operational: Plannings about vehicle fleet and scheduling and driver roster and scheduling

iv) Real-time Control: Control strategies in order to control an operating transportation system

Frequency Setting Problem (FSP), which is also known as service adjustment, is stays in tactical level decision category. It implies the frequency of trips of a route in a certain period of time. Being the one of the most encountered problems it can be also crucial since an unoptimized frequency setting for a route may cause high costs for insignificant levels of passenger loads or cannot meet the passenger demand in a certain period of time like AM and PM peak hours in demand patterns (Hadas and Shnaiderman 2012).

Frequency setting is also important for the further levels of planning such as fleet size and scheduling which are included in operational planning decisions. In the study of Tekin et al., frequency setting of trips are determined via Linear Goal Programming method and also fleet size that required for the bus lines that tested is determined according to the resulted frequency setting .

Redman et al. categorised performance measures that will increase the quality of public transportation services. The most relevant quality attribute with frequency setting is the passenger comfort since their access to the available seats will directly effected with the frequency decision. So it can be said that the frequency setting is directly related with the balance between cost efficiency and passenger comfort (Więcek et al. 2019). The demand characterization based on passenger load in time intervals by determining the time boundaries of sequential time intervals can be a powerful tool for optimizing this balance in FSP (Ceder 2016). Thus, the place of a peak or off peak can be identified automatically without observing the visualization of data.

Aside from FSP, demand characterization can be also used in different analyzes. For example, Faroqi and Mesbah identified the behavior of passengers with different trip purposes through the different time windows of a day. These time windows are determined via the demand characterization such as AM and PM peak and off peak time windows.

Mohamed et al. clustered the weekly passenger profiles that represented as boarding activities grouped with 1-hour time intervals. After the visualization of results with day-hour heatmaps, peak times of the weekly profiles for each cluster have been clearly identified. Another similar approach to the clustering passenger pattern based on boarding activities can be also seen in the study of Briand et al.

2.1. Time Series Clustering in Public Transportation

Time series format is used to describe a time-driven data with a predetermined time intervals (V. Vo, Luo, and B. Vo 2016). Most of the references mentioned above described the data as time series. In many occasions, distance measures that generally used to cluster time series are generally specialized for measuring the distance between time series. He, Agard, and Trépanier classified the individual passenger behaviour with time series clustering. In the study, performances of different distance measures are tested. The tested distance measures are Cross-Correlation Distance and Dynamic Time Warping which are specialized for time series. The traditional distance measures, Euclidean and Manhattan, are also mentioned but did not tested because of being not suitable for time series data. Since if one of the two time series shifted towards one time interval, it will be completely change the result.

Clustering the data with time series format will give the information about sequence of similar activities. In public transportation, time series clustering can be used to classify various types of activities. Matias et al. analyzed the bus schedules and trips with Dynamic Time Warping measure. Schedules of each bus are formatted as daily frequencies through 8 months with 1-day time intervals in order to validate the number and coverage of bus schedules.

Tupper, Matteson, Anderson, et al. proposed the Band Distance for time series. It measures the distance between two time series according to how many times they are stayed within the same band. The bands are collection of binary combinations of the other time series than these two. Band Distance is used along with Euclidean Distance in order to classify the activities of bus lines in the study of Tupper, Matteson, and Handley 2016.

Dubos-Golain, Trépanier, and Morency describes the bus lines as time series in which the daily boarding counts within 1-hour time intervals is considered. However, when classifying the use patterns of bus lines, Dubos-Golain, Trépanier, and Morency includes another variables which makes it a mixed approach rather than pure time series clustering approach. The variables are hourly boarding counts for 24 hours, statistics of the normalized hourly boarding counts such as mean, minimum, maximum and standard deviation of 24 hourly time intervals and weather variables such as mean temperature and daily snow and rain distribution.

There are also mixture model approaches exists in literature for time series clustering. Briand et al. uses Gaussian mixture model with CEM and EM algorithms in order to classify individual passenger behaviour. Mohamed et al. uses Poisson mixture model with EM algorithm to classify transit stations. Min uses Gaussian mixture model in order to convert boarding and alighting time series of metro stations to probability distributions. Then, identifies the AM and PM peak occurrence times of boarding and alighting distributions. The identification results 4 variables to use in K-Means algorithm.

2.2. Alighting Estimations

Origin Destination (OD) estimation is one of the most researched public transportation analysis in literature. It is not only used in public transportation but in general traffic topics as well such as congestion detection and traffic simulation (H. Yang and Rakha 2019). But OD estimation approaches for general traffic are not mostly available for the public transportation. Because data gathering methods and purpose of the approaches are much different than public transportation (Bera and Rao 2011). In public transportation, OD estimations can be used in various analyses such as passenger trip purposes (Lee and Hickman 2014), transfer-activity identification (Nassir, Hickman, and Z.-L. Ma 2015) and even frequency settings after the calculation of bus occupancy (Ceder 2016; W. Wang, Attanucci, and Wilson 2011).

Because of the restrictions of data collecting or storing in public transportation systems, makes it mandatory to make additional analyses prior to OD estimation such as zone to zone OD matrices and alighting estimations (Hussain, Bhaskar, and Chung 2021). The need for alighting estimation analysis is stems from the most of the public transportation systems collects fares only at entry (T. Li et al. 2018. T. Li et al., categorizes the approaches for alighting estimation in three categories:

i) Trip Chaining Method

ii) Probabilistic Methods

iii) Deep Learning Methods

In this study, Trip Chaining (TC) method and probabilistic methods are considered.

Trip chaining method is introduced by Barry, Freimer, and Slavin with two main arguments for estimation of alighting locations. First one is "most of the passenger's origin of the next trip is near the destination of their previous trip" and second one is "most of the passengers end their last trip of a day near the origin of the first trip of the day" (Barry, Freimer, and Slavin 2009).

Over the time many extensions are studied for TC method. Trépanier, Tranchant, and Chapleau used monthly data for TC and changed the last trip rule from "first trip of the day" to "first trip of next day". Additionaly, if a passenger made one transaction for a day, this trip is estimated by other similar trips of the particular passenger within the month. X. Ma et al. also characterized the single trips with the similarities between the trips of other same type of passengers.

He, Nassir, et al. made a validation for calibrating the tolerance distance between the sequential trips. Munizaga and Palma changed concept of the tolerance distance between destination and origins of sequential trips with generalized time and represent the distance as time between trips.

The aim of probabilistic methods is to calculate the alighting probability of passengers on possible destination stops. Y. Yang et al. proposed a non linear programming model to calculate a real time OD matrix. Alightings are predicted within a deep optimization framework. Ait-Ali and Eliasson proposed a model that estimates the most probable OD matrix by maximizing the system entropy with predetermined constraints. The model calculates the OD matrix entirely instead of inferring each alighting.

The seed matrix concept is used for the methods that calculates an OD matrix via a pre-existing OD matrix Cui 2006; Yuandong Wang et al. 2019. The pre-existing OD matrix can be either consist of the number of passengers or alighting probabilities.Cheng, Trépanier, and Sun used a modified latent dirichlet allocation (LDA) model in natural language processing (NLP). Alighting of passengers has been estimated by characterizing each passenger trip by a multinomial distribution. It requires a pre-estimated OD matrix or an OD matrix that estimated via real life data.

B. Li used bayesian inference for the parameters of Markov Chain Model which models the route level OD matrices. It uses for the alighting probabilities of passengers that boarded a certain bus stop. There are several extensions are exist for Markov Chain Model (Hazelton 2010; X.-l. Ma et al. 2012).

For the validation of OD matrix results, the general methods are using the real life data or survey data T. Li et al. 2018. There are several survey methods exist. For example, Huang et al. made a ticket recycling survey in which one surveyor gives tickets to the boarding passengers and the other surveyor takes tickets from the alighting passengers. Farzin used a household travel survey for validating two different OD matrices that lacks conformity.

3. MATERIAL AND METHOD

3.1. Data Management

3.1.1. Pre-Existing Public Transportation Database

The source data that contains all information about public transportation of Antalya was obtained on 31 January 2020 from Department of Transportation for the city of Antalya. The data include information about all lines, routes and bus stops, bus stop and route relationships, geo-location data of bus stops and routes and boarding data that contains boardings from three different days, 12 June 2019, 16 October 2019 and 18 December 2019.

The data was organized and stored in a PostgreSQL database by Bulut 2021. All additional data obtained after 31 January 2020 added into this database as additional tables. Because all additional data obtained are consist of incompatible data fields with existing boarding table but still compatible with the current route and bus stop tables.

3.1.2. Additional Data

The additional data that contains boarding information of four different days and two months was obtained on 18 September 2020 from the same source. The database already has the three days of these four daily data which are mentioned above. Newly added daily and monthly boarding data contain information about boardings that occurred on 1 September 2020, May 2019 and October 2019 respectively. Since one of the daily data already covered in monthly data, the additional boarding data covers 63 days in total, consisting 62 days in pre-pandemic era and 1 day in pandemic era.

As mentioned, the additional data obtained are contains different data fields than the older boarding data. The older boarding data consist of card id, boarding time, bus id, route id, bus stop id and the additional data consist of passenger id, boarding time, bus id, route id, trip start time, trip end time, bus stop id and longitude and latitude of boarding. These data field differences makes it mandatory to store new data in different tables. The reasons are:

i) Card id and passenger id has entirely different entities which are contain 16 and 30 digit numbers respectively. The reason behind this difference is the data provider hashed the passenger id to card id when providing the first batch of data because they thought it may be cause a security breach. In the second batch they did not hash the passenger id since it won't be any security breach. This difference makes the new boarding data incompatible with the old boarding data

ii) The existence of trip start and end times. Trip information is definitely useful source of data but adding the trip information makes it mandatory to redesign the boarding data storage.

iii) The existence of location data of the boardings. Boarding locations can be used in possible future researches or for error detection.

The additional data obtained in five CSV files, one file for each time period. CSV means comma separated values and all data fields have to be separated by comma. But comma is also used as percent separator in Turkey. Because of this reason longitude and latitude values also separated with comma which translates into some rows have 9 and some others have 11 columns for CSV reading libraries and programs. I wrote a multi threaded python script in order to this problem as quick as possible. The script takes the csv file as input and iterates over rows. If a row has 11 commas, changes the 9th and 11th commas into dots. Every thread makes their own iteration and process the rows that are assigned to them. This assignment is based on the Equation 3.1 where N is the number of threads, r is the index of the row that is going to assigned to thread at index i.

$$i = r(modN) \tag{3.1}$$

After solving the separator issue, 5 files are merged into one file which has total 25976695 raw records. In the raw records, there are missing values for the columns route id, bus stop id and locations. Missing values for locations are not serious since it won't be any effect for the relation design of the current database. But missing route id and bus stop id values is crucial and additionally these data will be needed for later. To keep the logical integrity with the current database, Bus stop and route tables are fetched and ids of the both of these tables are matched with values of the bus stop id and route id columns of the new boarding data. 21820323 rows of clean data remains after the elimination of the unmatched data.

Preparing new table and matching with existing data phases solves the issues for passenger id and boarding location information respectively. But for trip data, a new trip table has to be added to the database. Definition of the trip is "a single travel through the route". So, the aspects that determine the route within the data are bus id, route id and trip start and end times. So the boarding data grouped by these four columns to define a trip table. Each trip is numbered after sorted by trip start time in order to produce trip id. Then, boarding and trip table are inner joined in order to assign each boarding a trip id. Now, there are two tables to add into the database:

i) Boarding table consist of: id, passenger id, boarding time, bus stop id, trip id, longitude, latitude

ii) Trip table consist of: id, trip start time, trip end time, bus id, route id

These two tables, "BoardingV01" and "TripV01", are added into the database in order to distinguish newly added data from the old data. Then, a new view is created based on the code below in order to get the data without writing complex queries for studies.

```sql
select
        b."PassengerId",
        b."BoardingTime",
        b."Latitude" as "BoardingLatitude",
        b."Longitude" as "BoardingLongitude",
        t."BusId",
        t."Id" as "TripId",
        t."TripStartTime",
        t."TripEndTime",
        r."LineId",
        l."LineCodeRepresentation",
        r."DirectionId" as "Direction",
        t."RouteId",
        b."BusStopId",
        bs."Latitude" as "BusStopLatitude",
        bs."Longitude" as "BusStopLongitude",
        bs."StopName" as "BusStopName"
from "public"."BoardingV01" b
left join "public"."TripV01" t on b."TripId" = t."Id"
left join "public"."BusStop" bs on b."BusStopId" = bs."Id"
left join "public"."Route" r on t."RouteId" = r."Id"
left join "public"."Line" l on r."LineId" = l."Id"
```

KNIME analytics platform (Berthold et al. 2007) and custom Python scripts are used for implementation of the following methodologies.

3.2. Boarding Pattern Clustering

Daily passenger demand of a bus line can be represented as time series where y-axis is boarding count and x-axis is time. A day has to be divided with a predetermined time intervals in order to produce a meaningful time series from the boarding data. The time interval value for daily passenger demand is selected as 30 minutes for the study. 30-minute time intervals will be sufficient to represent an average bus line demand for a city like Antalya. But it can be adjusted lower for a more crowded city.

In order to obtain an average daily demand of a bus line, average boarding count of a collection of regular days has to be extracted for each time interval. For example, for 14:00, which translates into the time interval that covers between 14:00 and 14:30, average count of boardings that occurred within time interval 14:00 of 62 days. Boarding data of 1 September 2020 did not taken into calculations because it is in pandemic era.

A regular day for a bus line starts at 6:00 and ends at 23:59. But ending time implies the start of the last trip of the day, so lines take passengers after 23:59. That means, if a day starts from 00:00 there will be inconsistencies on time series of daily demands. In order to produce healthy time series, it is assumed that the day starts at 4:00 and ends at 3:59.

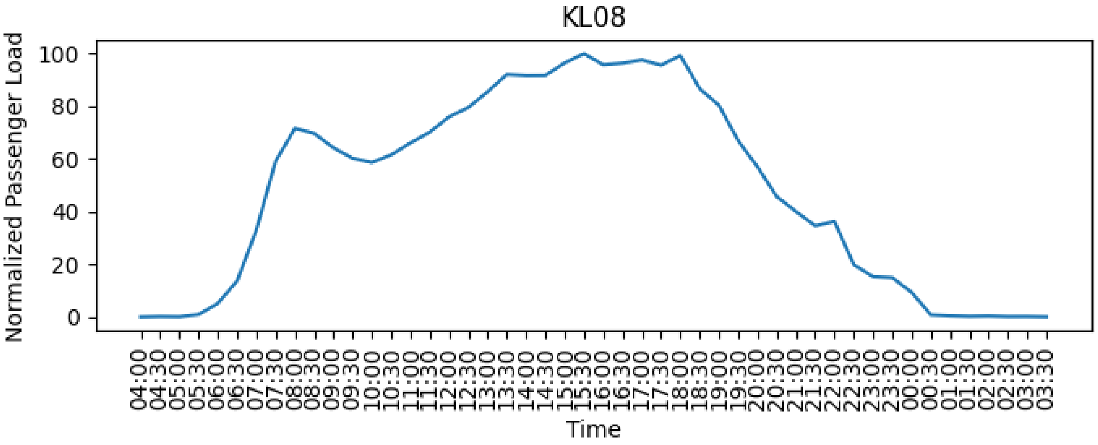

Figure 3.1. Boarding Pattern of KL08

After these adjustments, the daily passenger demand of a bus line can be seen in Figure 3.1.

Clustering the bus lines according to their daily passenger demands directly in current state will not give any fine result. Because it will give same results with clustering bus lines according to solely by their total passenger count. In order to prevent this problem, each daily demand has to be normalized by its own maximum value (Equation 3.2 where the boarding count on i^{th} time interval is t_i, maximum boarding count of the bus line in an average day is t_{max} and the normalized value of i^{th} time interval is T_i).

$$T_i = t_i/t_{max} \tag{3.2}$$

The boarding pattern of each bus line is calculated by normalizing its daily demand. By clustering the boarding patterns of bus line, it will be possible to detect the bus lines that shows similar daily demand characteristics regardless of their difference in popularity.

Boarding patterns of all bus lines can be seen on Figure 3.2. There are some of the outlier boarding demands can be observed in some of the bus lines. In order to prevent these outlier values to create inconsistencies in clustering results, All of the bus lines are smoothed via moving average method. Window size of the moving average choosen as 5 time intervals. Starting time intervals where the window size is less than 5 time intervals are not excluded in calculations.

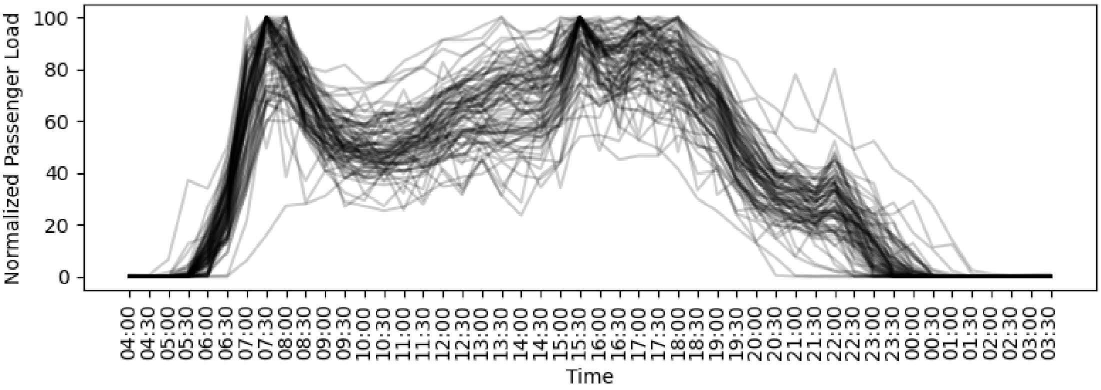

Figure 3.2. Boarding Patterns of All Bus Lines

Clustering calculations for boarding patterns can be done with following distance metrics.

3.2.1. Euclidean Distance

Theoretically, Euclidean Distance (ED) metric is not meant for the calculating the distances between time series. Distance calculation with ED will be resulted as if the boarding count that occurred in each time interval as data fields. But ED still can be tried since the boarding pattern data is represented the same way.

Sklearn library of Python is used for the clustering calculation with ED metric. K-means algorithm is selected to be the clustering method for the calculations.

3.2.2. Dynamic Time Warping

Dynamic Time Warping (DTW) is a well known distance metric that can be used on time series. The idea is measuring the distance between time series based on their shape regardless of length or time difference between occurrences. But even so, time difference between occurrences may be crucial for boarding patterns. Because, if time difference between occurrences is too high, it has to be a distant boarding pattern. For instance, if two lines have their morning demand peaks at 8:30 and 10:00 with same curves, it will be a problem for the transportation planners because they have to adjust timing addition to the scales when planning based on the clusters.

In order to solve this issue, lengths of all boarding patterns adjusted to 48 time intervals even the line is not active at some time intervals in a day. This may also create an advantage to the clustering algorithms to separate bus lines operate regularly and nightly.

Tslearn library of Python is used for the clustering calculation with DTW metric. Time series k-means algorithm is selected to be the clustering method for the calculations.

3.2.3. Band Distance

Band Distance (BD) is a distance metric proposed by Tupper, Matteson, Anderson, et al. It is used to measure distance between any two time series in a collection of time series. The idea is measuring the distance between two time series based on how much they are both included in the areas inside bands which are the binary combination of the time series other than their BD calculated. So, increasing the collection size may result more accurately.

The process of BD, starts with choosing two time series. Time series that other than the chosen two are used for creating bands. Both of the time series are checked in which time intervals they are stayed in the area inside of the band boundaries and this will result with two time interval set. After checking, Jaccard distance of these two sets is calculated for each band. Mean of all Jaccard distances will give the BD of the chosen two time series.

As it can be said that the time complexity of BD is too expensive. If n is the number of lines and t is the number of time intervals, time complexities are:

i) ED: $O(n^2 * t)$

ii) DTW: $O(n^2 * t^2)$

iii) BD: $O(n^4 * t)$

Since BD is only for specific purposes, it is not included in any Python library. So, BD is implemented with dynamic programming principles via storing the calculations of the inclusion of each bus line in every possible band combination in a separate data structure. Thus the time complexity of the BD reduced into $O((n^2 * t) + (n^4)) = O(n^4)$. It is still expensive since searching through the band combinations and calculations of

Jaccard distance are still has to be performed in the main loop.

After implementing the BD as Python script, K-Medoids algorithm of Sklearn library is used for the clustering calculations.

3.3. Time Slot Clustering

In Boarding Pattern Clustering, clusters will be expected to consist of lines with similar demand patterns like having peaks, dips or trends at close times with similar demand levels. So it can be said that the peak and off-peak areas can be used to profile the daily demand of a single bus line.

In order to profile daily demands, clustering approach is preferred. Clustering approach starts from the first time interval and checks the demands of time intervals sequentially. If a time interval tends to show a different characteristic, which is identified by predetermined boundaries that changes with methods, then the algorithm starts a new cluster from that time interval. After the clustering algorithm ends, clusters will be consist of the sequential time intervals that shows similar characteristics within boundaries that determined by the cluster method. These clusters are named as time slots.

Many different types of time slot boundaries can be determined for various types of purposes. Two of them are identifying the peak and off-peak time slots and identifying the demand trends within a day. The proposed Time Slot Clustering (TSC) methods for these purposes are Stepped Time Slot Clustering (STSC) and Adaptive Time Slot Clustering (ATSC) respectively.

The same data that used in previous section is used for this section as well since the topic is the daily demands of bus lines. Normalization and smoothing is not used for TSC calculations. Because of TSC can be applied to the daily demands without normalization and smoothing.

3.3.1. Stepped Time Slot Clustering (STSC)

As mentioned, the purpose of STSC is identifying the time slots with peak and off-peak demands. So, determining the clusters according to their average demand level will most likely to give peak and off-peak time slots.

The algorithm, starts with the earliest time interval and iterate through the day.

Thus, a predetermined threshold is needed in order to decide where the time slots start and end. The threshold value is used for limiting the change of the average boarding count of the time slot that currently calculated by the algorithm when adding a new time interval.

Elbow method like approach is used in order to determine a threshold value when using STSC on a bus line. STSC is repeatedly executed with different threshold inputs and sum of squared errors (SSE) of every execution is considered in order to find a fine threshold for the bus line. In this process, threshold is increased by a set value on every iteration. Selection of the threshold is based on the deepest decrease occurred at SSE values. That's because, if threshold value is selected too high or too low there will be only one cluster or 24 clusters respectively. Thus, decision of the threshold value has to be made based on an optimized SSE value.

After determining the threshold for the bus line, STSC is used with these steps:

i) Load boarding dataset

ii) Filter by Bus Line

iii) Group by 30-minute time intervals

iv) Sort time intervals ascending

v) Create a new time slot

vi) Iterate over time intervals

vii) Set $CurrentMean$ equal to the mean of the boarding counts in the time slot

viii) Set $NewMean$ equal to mean boarding counts for time slot and current time interval

ix) If $\mid NewMean - CurrentMean \mid > Threshold$ then create a new time slot

x) Add current time interval to the time slot

xi) Return to 7 if there is no time interval remaining

3.3.2. Adaptive Time Slot Clustering (ATSC)

The purpose of ATSC is identifying the demand trends within a day as mentioned. Demand trend is stands for the frequent increase, decrease or staying same demand level for several time intervals. It is likely to draw a linear regression with boarding counts throughout the day. Thus, the main idea of ATSC is using linear regression for time slot calculation.

In ATSC, time intervals in a time slot have to stay in same linear regression. If the linear regression slope of the time slot changes more than a predetermined tolerance angle, the time slot ends which translates into the end of the demand trend.

The end of a demand trend does not always turning to the reverse of the current state. For example, an increasing demand trend breaks into a more aggressively increasing demand trend if the tolerance angle is set low enough.

The biggest issue for the tolerance angle and slope based clustering is the axes are not assigned with same units. In calculations, units of the time and boarding axis are hours and passengers respectively. Which means, the difference between time intervals are 0.5 hours while differences between time intervals as boarding counts are generally larger than 1000 passengers for 62-days data. Thus, the slope will be unrecognizable in degrees. In order to solve this issue, the unit of time axis is temporarily changed into minutes from hours and multiplied by 60 for calculating 62 days of data. Thus, the slope between time intervals can be expressed by degrees.

3.4. Alighting Estimation and Occupancy via Reverse Route

In a weekday routine, passengers tend to prefer the reverse routes of a line when traveling between home and work or school in general (Alsger et al. 2018; Faroqi and Mesbah 2021). This can be easily said when observing the boarding patterns of bus lines where the public transportation usage is high between 07:00 and 08:30 in morning and between 15:00 and 18:30 in afternoon (Figure 3.2). Therefore, it may be possible to predict alightings in afternoon of a route with boardings occurred in morning of its reverse route.

In order to test the reverse routes hypothesis, it has to be known that which bus stops of reverse routes are opposing to each other. But unlike metro stops, bus stops are considered only have one direction. Additionally, number of bus stops are assigned to reverse routes are not same for most of the cases. Therefore, opposing bus stops have to be estimated before estimating alightings with reverse routes.

The data needed for the methods are boarding data, bus stop of route data and map point data. Boarding data used for calculations consist of only 18 December 2019 boardings. Only one day selected for the alighting estimation since it is mentioned that estimations can be done with boardings occurred in a single day. Also, boarding data of 01 September 2020 is not used because of it is in pandemic era and bus stop of route data is more compatible with 18 December 2019 data.

Bus stop of route data is a relation table between routes and bus stops and it includes the sequence of bus stops within a particular route. It does not specifies the distance between bus stop and starting point of the route or in other words the location of a bus stop on a road. But it can be estimated with the use of the map point data. Even though, the only purpose of the map point data is to be used for drawing the routes which only consist of the turning point locations within the route with a sequence.

3.4.1. Bus Stop Location on a Route

Even though bus stop locations are clearly known via their coordination, in order to know how they are close on a particular route. Some routes like KL08 forward and DC79 forward have sharp twists and some others like DC79 backward even traverses an area. This characteristics of routes makes it mandatory to know the bus stop location on a route rather than only its global coordination for some calculations.

With the use of map point and bus stop of route data it is possible to estimate the Bus stop location on a route. It can be done in a $O(b + m)$ time if number of bus stops of the route is b and number of map point of the route is m. The basic idea of the algorithm is simulating the movements of a bus that travels on the route. This is done by defining an imaginary cursor that iterates over the map point data and comparing its Haversine distance with next map point and next bus stop. Further calculations are done by comparing these two distances.

If index of map points are denoted by i and index of bus stops are denoted by j, then the calculation steps are:

i) Bus Stop of Route and Map Point tables are both filtered by the $RouteId$ of the desired route that its bus stop locations are estimated.

ii) Both of these tables are sorted by $Sequence$.

iii) Define a $Cursor$ that have the value of the first map point.

iv) Store indices of next map point and next bus stop as $i = 2$ and $j = 1$ respectively.

v) Current location on route is stored as $location = 0$.

vi) Start an infinite $while$ loop.

vii) $Cursor$'s Haversine distance between next map point (D_{CMi}) and its Haversine distance between next bus stop (D_{CBj}) are calculated.

viii) If $D_{CBj} \leq D_{CMi}$, then record D_{CBj} and $location$ values to the output table and $j = j + 1$.

ix) If $D_{CMi} \leq D_{CBj}$, then change $Cursor$ to the value of next map point and $i = i+1$ and $location = location + D_{CMi}$.

x) Stop the loop when there is no Bus Stop or Map Point left to iterate.

The algorithm seems fine in paper but fragile. Some problems experienced when developing the algorithm:

i) Routes with sharp twists and traverse may cause inconsistencies.

ii) Map points assigned over frequent that it prevents the cursor detect the bus stop.

iii) Sequence field in bus stop of route table may have errors.

iv) More than one assignment of a bus stop to a route

v) Sequence of map points reversed for a route.

vi) Map point sequences of routes that belong to a line may exchanged.

On coding phase first two of these problems can be solved in script like adding additional rules like location of a bus stop cannot be lower than its previous one and in every k^{th} map point execute a checkpoint process that checks if any of the bus stops are missed.

Reverse map point sequences can be detected by comparing the distance of first and last map points to the first bus stop. If last map point is closer then the map point sequence has to be reversed before the algorithm.

Some routes have repeated bus stops according to the data. Theoretically, it is not impossible. But, Antalya's current transportation system, which operates from 2011, does not allow routes to visit a bus stop more than once. Therefore, repeated bus stops for a route can be ignored.

Unfortunately wrong sequences and exchanged map point sequences cannot be solved automatically. These errors have to be examined and solved by hand. Routes that have these types of errors are either inactive or unpopular.

3.4.2. Bus Stop Aggregation

Opposing bus stops have to be estimated in order to estimate alightings with reverse routes. The starting point of the idea is using same bus line between origin and destination. In public transportation, origins and destinations are represented by bus stops. But in reality, the passenger's destination stands within a walking distance from the destination bus stop. The origin bus stop of the returning trip of the passenger will be still be in the walking distance boundary. So it can be estimated that these two bus stops are the opposing bus stops.

In many occasion, there will be more than one opposing bus stops or even a group of bus stops are opposing an another group of bus stops. In a city like Antalya where the walking distance may contain more than 1 or 2 bus stops for a single direction. These opposing bus stops that grouped with an index that represents an area within a bus line. This area is determined by a predetermined walking distance. The bus stops that grouped with this way are named Bus Stop Groups (BSG).

The area of BSGs are determined within a predetermined walking distance. But, dividing the route lengths directly to the walking distance directly will cause inconsistencies within BSG areas. Because of the remainder of these divisions eventually assigned to the last BSG of the route which will be coupled with the first BSG of the reverse route. In order to prevent this, number of BSGs have to be calculated with Equation 3.3 where N_{BSG} is number of BSG, $L_{shorter}$ is length of shorter route of the line and D_w is predetermined walking distance.

$$N_{BSG} = \lceil L_{shorter}/D_w \rceil \tag{3.3}$$

$$D_{Gi} = L_i/N_{BSG} \tag{3.4}$$

After finding the number of BSGs, the distance covered with BSG areas, which is named as grouping distance, have to be calculated with Equation 3.4 where D_{Gi} is grouping distance of route with direction i, L_i is length of the route. As a reminder, a bus line consist of forward and backward routes and in database it is encoded as directions 0

and 1 respectively. The index i indicates the direction of a route.

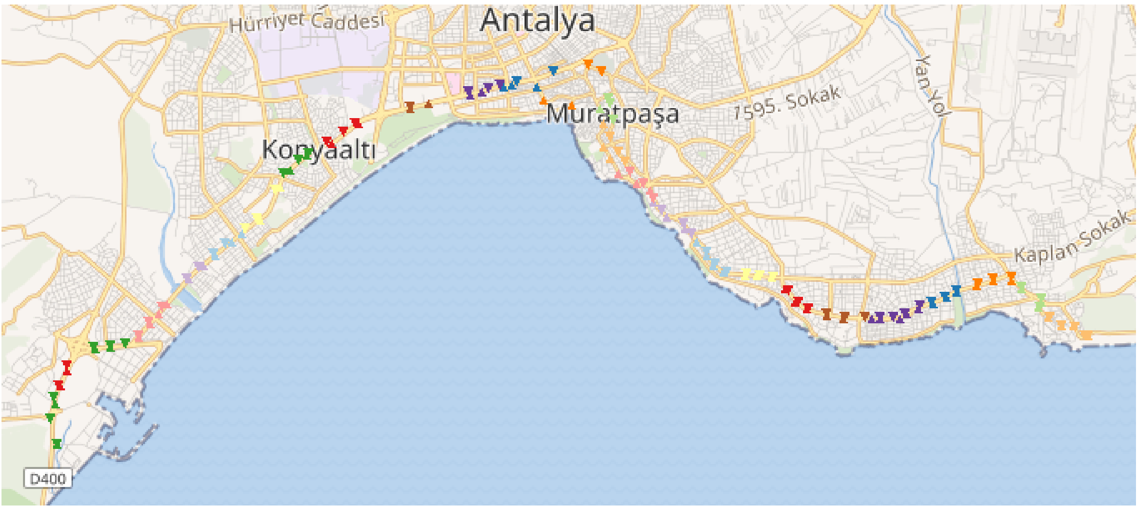

Figure 3.3. BSG Results of KL08

In order to determine which bus stop belongs to which BSG, the location of a bus stop on route has to be divided into grouping distance. Floor of this division gives the BSG index of bus stop for the bus stops belong to forward route of bus line. For calculating the BSG indices of backward route bus stops, this value should be extracted from $N_{BSG} - 1$. So a BSG index refers a particular area on the bus line.

The formula for BSG index can be seen in Equation 3.5 where L_{Rj} is j^{th} bus stop's location on the i^{th} route and I_{BSGj} is the BSD index of this bus stop.

$$I_{BSGj} = \begin{cases} \lfloor \frac{L_{Rj}}{D_{Gi}} \rfloor, & \text{if } i = 0 \\ N_{BSG} - \lfloor \frac{L_{Rj}}{D_{Gi}} \rfloor - 1, & \text{if } i = 1 \end{cases} \tag{3.5}$$

The results of the bus stop aggregation for bus line KL08 can be seen in Figure 3.3. Every bus stop belongs to KL08 are marked as triangles and colored by their BSG indices. Upward and downward triangles are used for the forward and backward routes respectively.

3.4.3. Aggregated Reverse Route

The idea of alighting estimation via Aggregated Reverse Route (ARR) originates from passengers that use public transportation only for work or school at rush hours. This type of passengers uses route with $direction = d$ when going to work on morning and route with $direction = (d + 1)(mod2)$ when returning from work on afternoon, where $d = 0$ for forward and $d = 1$ for backward route of preferred line. Which means it is highly likely to a passenger's alighting bus stop on morning is the opposing bus stop of the passenger's boarding bus stop on afternoon. This will be probably misleading for individual passengers since public transportation is used for much more various purposes but it can be used for a probabilistic approach for predicting the alighting bus stop of passengers that boarded from particular bus stops and particular time of day.

In previous section, opposing bus stops are estimated via bus stop aggregation. So it is highly likely to the alighting bus stop on morning will be in the same BSG within the boarding bus stop on afternoon. But, afternoon peaks are differ for the bus lines (Figure 3.2). Because of this reason, using a fixed morning and afternoon time boundaries may not be give accurate results for every bus line. Thus, STSC is used to determine the time boundaries of morning and afternoon time frames. STSC is used for the day of which alighting data are estimated and the time slots that contain morning and afternoon peaks are used as the morning and afternoon time frames.

After reversing the route and time, the resulting table consists of columns as time frame and route combinations (4 columns) and lines as BSGs. Every cell in table contains the boarding counts from a particular BSG at a particular time frame from a particular route of considered line.

3.4.4. Direct Reverse Route

Directly Reversed Route (DRR) is a what if scenario of ARR of which discards the time and bus stop aggregations. Alighting estimation is done by directly reversing the route and takes account passengers that boarded one time for each route of line instead of calculating time slots.

Route reversing is achieved by reversing the bus stop sequence of the backward route. As it is mentioned that the transportation network does not designed for opposing bus stop couples and that makes the number of bus stops for each route of bus lines are not equal. But in DRR, this problem is solved by adjusting the sequence matching based on middle of the bus stop sequences rather than start or end. For example, KL08 has 71 bus stops in forward route and 74 bus stops in backward route. After reversing the sequence of the backward route, 74^{th} bus stop matched with the 1^{st} bus stop of forward route. In order to match middle points of route, the matching of 1^{st} bus stop of the forward route has to be adjusted. This adjustment can be done with Equation 3.6 where Δ is the dislocation from starting bus stop.

$$\Delta = \lceil \frac{\mid N_{BSforward} - N_{BSbackward} \mid}{2} \rceil \tag{3.6}$$

After the calculation, the last bus stop of shorter route is matched with $(\Delta + 1)^{th}$ bus stop of the longer route. With this way, the boarding occurred on a bus stop in a route is counted as alighting occurred on its matched bus stop of reverse route.

In order to compare the methods, bus stop aggregation is used after the calculations.

3.4.5. Trip Chaining Method

Trip Chaining (TC) is well known method for alighting estimation. TC uses the trips of individual passengers that travel more than one time and checks the boarding stop of the next trip in order to estimate the alighting location of a trip. The distance between these two stops has to be lower than the walking distance.

TC is used for verification of ARR and DRR. Since daily data is used in this section, TC is modified to estimate a daily data and also to improve speed. In order to estimate a daily data, it is assumed that destination of every passengers last trip of a day is origin of their first trip of the same day. In original TC methodology it should be the next day instead of the same day.

Python scripts in KNIME can be implemented via Python script nodes. These nodes can take two input tables at most. Boarding data and bus stop of route table is mandatory for TC. But, calculation with only these two tables cause the execution takes long time. In order to improve performance, the haversine distance matrix of all bus stops in database is calculated beforehand. Because of this reason, Jupyter Notebook is used for TC calculations.

Like DRR, the results of modified TC is converted to the same format of ARR output in order to be used for comparison.

Table 3.1. Probability Matrix Example

	BS_2	BS_3	BS_4	BS_5	BS_j	BS_n
BS_1	$P_A(1,2)$	$P_A(1,3)$	$P_A(1,4)$	$P_A(1,5)$	...	$P_A(1,n)$
BS_2		$P_A(2,3)$	$P_A(2,4)$	$P_A(2,5)$	...	$P_A(2,n)$
BS_3			$P_A(3,4)$	$P_A(3,5)$	...	$P_A(3,n)$
BS_4				$P_A(4,5)$	...	$P_A(4,n)$
BS_i	...	...	...	...	$P_A(i,j)$	...
BS_{n-1}					...	$P_A(n-1,n)$

3.4.6. Probability Matrix

Probability matrix that used for reverse routes consist of boarding and alighting bus stops as lines and columns respectively (B. Li 2009). Each cell contains a passenger's alighting probability on jth bus stop or BSG, if that passenger boarded from ith bus stop or BSG.

A passenger has to alight at a bus stop that comes after than the boarding bus stop. So, a probability matrix (Table 3.1) has to be a triangular matrix since $j > i$ condition has to be met. This is also means that the sum of probabilities of each line in probability matrix has to be equal to 1. So, the alighting probability ($P_A(i,j)$) of a passenger who boarded from i and alighted at j is calculated with Equation 3.7 where n is the number of BSGs or bus stops and PC_A is the passenger count values for each BSG that stored in the column that used for alightings are based on.

$$P_A(i,j) = \frac{PC_A(j)}{\sum_{k=i+1}^{n} PC_A(k)} \tag{3.7}$$

The alighting calculations are based on the columns: afternoon backward boardings, backward boardings and forward alightings for ARR, DRR and TC respectively.

3.4.7. Origin Destination Matrix

Origin-Destination (OD) matrix is the real output for the alighting estimation. It is produced based on the probabilities on probability matrix with boarding data. The each cell in OD matrix refers to count of passenger who boarded from ith bus stop and alighted to jth bus stop which is notated as $PC_{OD}(i, j)$.

$$PC_{OD}(i, j) = PC_B(i) * P_A(i, j) \tag{3.8}$$

$PC_{OD}(i, j)$ is calculated with Equation 3.8 where PC_B is boarding passenger counts for each BSG. These columnas are morning forward boardings, forward boardings and forward boardings for ARR, DRR and TC respectively. But the daily data of 18 December 2019 will be used as boarding passenger counts for comparison phase.

3.4.8. Occupancy Calculation

Occupancy is the number of passengers traveling inside a bus at some point on route. Bus occupancy on a bus stop can be calculated by extracting the alightings from boardings that occurred until that bus stop. So before the calculation, number of alightings occurred from each stop (PC_A) has to be calculated (Equation 3.9). After the alighting calculation, occupancy calculation can be calculated with Equation 3.10.

$$PC_A(j) = \sum_{i=1}^{j-1} PC_{OD}(i, j) \tag{3.9}$$

$$Occupancy(i) = \sum_{k=1}^{i} PC_B(k) - \sum_{k=1}^{i} PC_A(k) \tag{3.10}$$

3.4.9. Verification

A real life data is necessary in order to verify the results of the alighting estimation methods. But, survey method is risky because of the study has been done in pandemic era. Thus, the only possible way to obtain real life data is video recordings of the security cameras inside the busses. The only video recordings available on the time is the video recordings from 01 September 2020. Because of the date of video recordings and the date of estimations based on are not the same dates, alighting and bus occupancy predictions will be verified of these three methods.

Obtained video recordings are belong to 4 most crowded trips of KL08, the most populated bus line of Antalya. For each route, 2 video recordings obtained.

Analyze of the video recordings are done by visually counting the boarding and alighting passengers without using the image processing techniques. That's because the video recordings are not taken from the suitable angles for programmatically counting boarding and alighting passengers or passengers that were travelling in the bus.

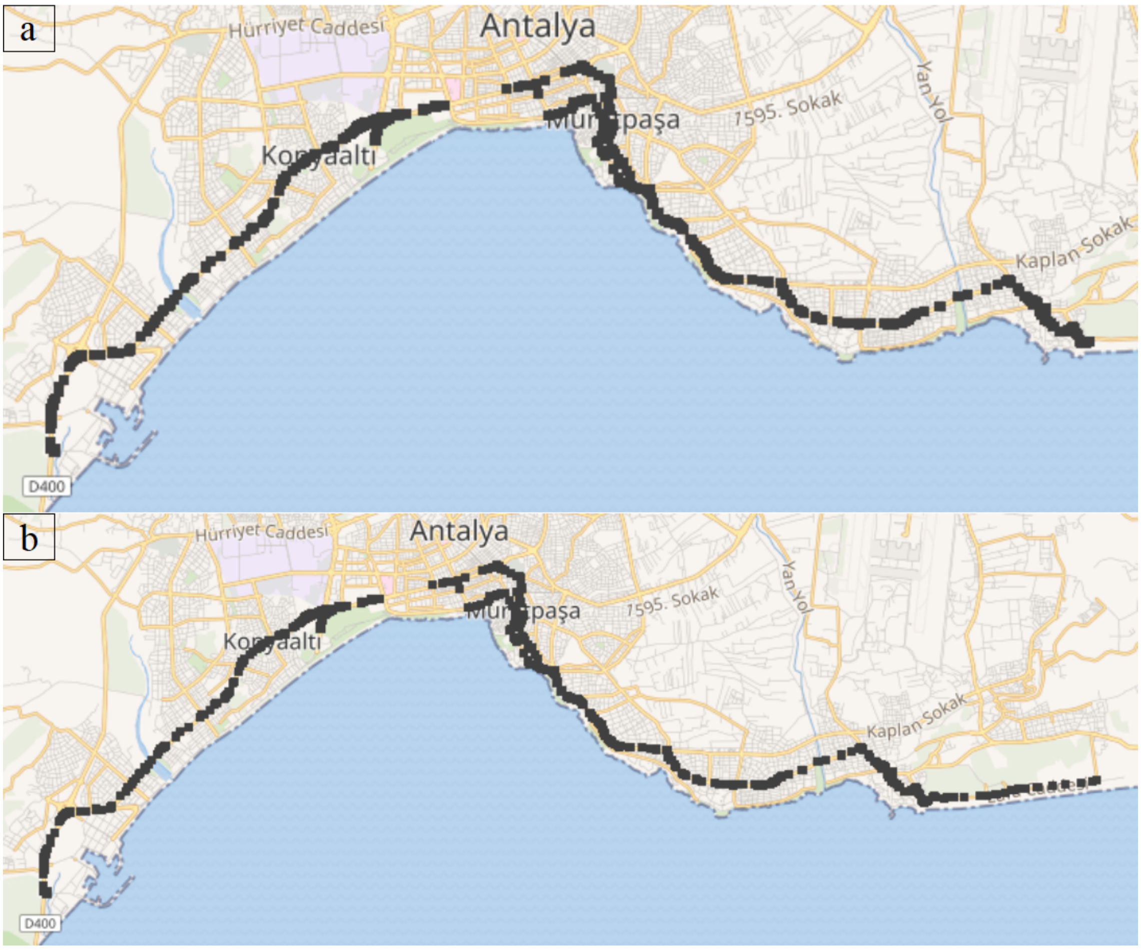

Figure 3.4. KL08 Coverage Difference Between 2019 and 2020; **a)** 2019; **b)** 2020

KL08 coverage was slightly changed from 18 December 2019 (Figure 3.4a) to 01 September 2020 (Figure 3.4b). At the end of forward and start of backward routes there were additional bus stops while the other side remained same. And these bus stop and route relation updates is not stored in data that obtained on 31 January 2020. As a reminder, the boarding data used does not contain the boardings occurred on 01 September 2020 because of this reason. Fortunately, there are not any boarding and alighting is occurred on these additional bus stops for forward route. Thus, verification can be done with predictions made for forward route of KL08 by simply ignoring these additional bus stops.

Four Excel sheets are generated after analyzing the video recordings. The data in Excel sheets consist of bus stop id, BSG indices, bus stop names, bus stop visit time, bus stop exit time, boarding count, alighting count. Newly added bus stops does not have BSG indices compatible with the predictions.

4. RESULTS AND DISCUSSION

4.1. Boarding Pattern Clustering

It is mentioned that size of the time intervals selected as 30 minutes. Unpopular bus lines and bus lines that have more that 30 minute trip frequency may not be have a clear boarding patterns for shorter time intervals. In order to eliminate these bus lines, a passenger count requirement is defined. This passenger requirement is set to 200000 passengers for 62-day data. 38 of 121 active bus lines are satisfied the passenger requirement and are considered for this section.

After the elimination, number of clusters for each BPC algorithm is selected to be 4 since it is assumed that boarding pattern characteristics are not varied too much among the 38 bus line.

As a reminder from Figure 3.2, All of these boarding patters have their critical decisive points where the variance is too high and which are peaks, dips or middle points of continuing demand trends. So, most of the comparisons will based on these critical points at 07:30, 10:00, 13:30, 15:30, 17:00, 18:00 and 22:00.

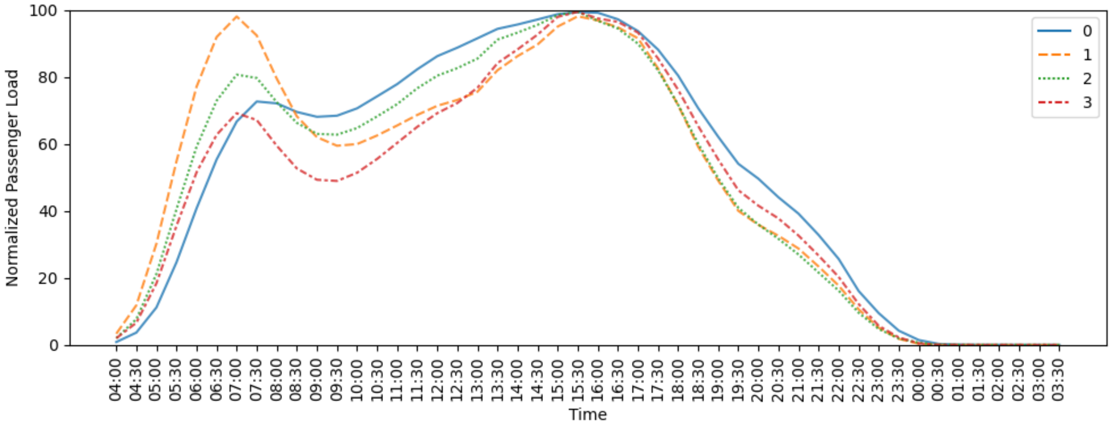

Figure 4.5. Results of ED Based on Cluster Averages

Visual observations can be made from the results of ED in Figures 4.5 and 4.6. According to the visual observations, the resulting clusters are separated from each other on morning peak (around 07:30), morning dip (around 10:00) and noon trend (around

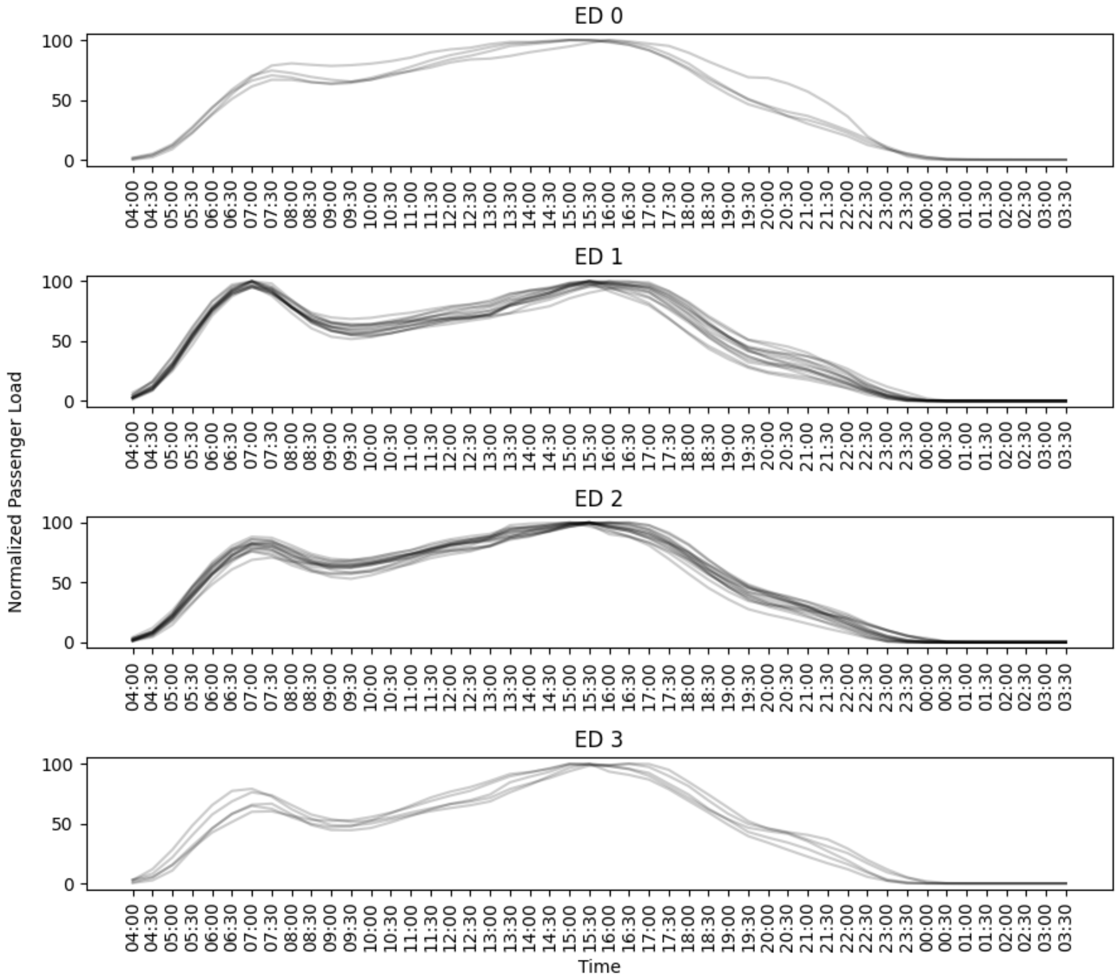

Figure 4.6. Results of ED for Each Cluster

13:30) hours. It can be said that, the morning dip of ED-0 is not an actual dip since it is not a significant drop in demand. So the demand of ED-0 constantly increases until the afternoon peak at 15:30 and decreases after from that point. This type of average pattern separates the ED-0 from the other clusters.

Even though, ED-1 and ED-3 mostly shows the same characteristics, they separate from each other at morning peak. ED-2's morning peak normalized passenger load (NPL) levels can be questioned as if it takes the bus lines that may be assigned either ED-1 or ED-3. But ED-2 diminishes from others at noon trend hours because of its higher NPL levels.

So according to the average cluster values (Figure 4.5), it can be said that the clustering algorithm separated the bus lines well but it may not be perfect. Because high variance between the NPL levels between bus lines of a cluster at some hours makes the results questionable. These high variance areas can be observed at (Figure 4.6):

i) Late afternoon and evening areas of ED-1 and ED-2.

ii) Morning, late afternoon and evening of ED-3.

These high variance areas are not significant for separation of clusters. Thus, the results of ED can be assumed as quite logical according to the visual observations.

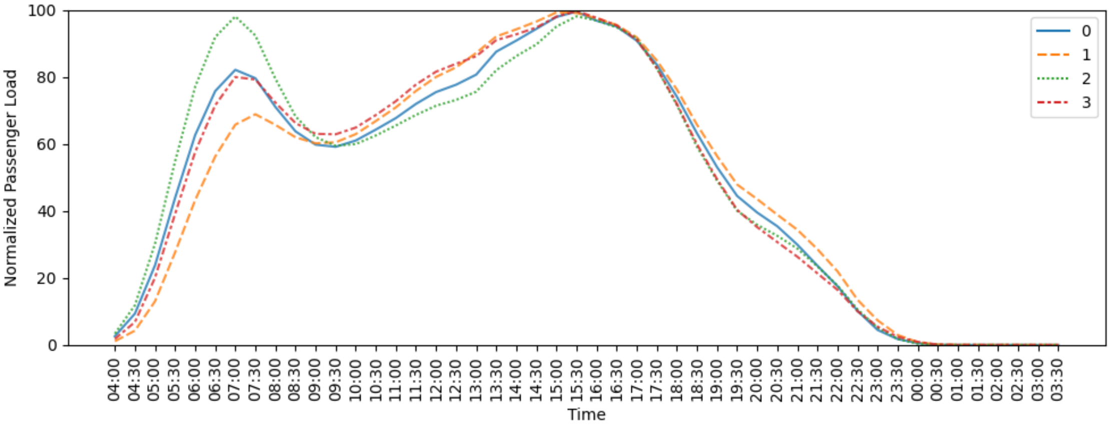

Figure 4.7. Results of DTW Based on Cluster Averages

The main and only significant separator time frame is morning peak for the resulting cluster averages of DTW method (Figure 4.7). Especially the resulting clusters DTW-0 and DTW-3 are quite similar cluster averages that only separates from each other at noon and evening. So, the resulting cluster averages of DTW does not seem quite logical but cluster variances of DTW (Figure 4.8) is quite interesting since some areas have significantly low variance:

i) All times before afternoon peak for DTW-2 and DTW-3

ii) Early evening of DTW-0

iii) Time before morning peak on general.

This type of result, low variance and not much distinguishable cluster averages, is quite logical when thinking the main idea of DTW which is calculating the differences between time series according to the shapes of the time series. But still, high variance areas like noons of DTW-0 and DTW-1 and evenings of DTW-1 and DTW-2 (Figure 4.8).

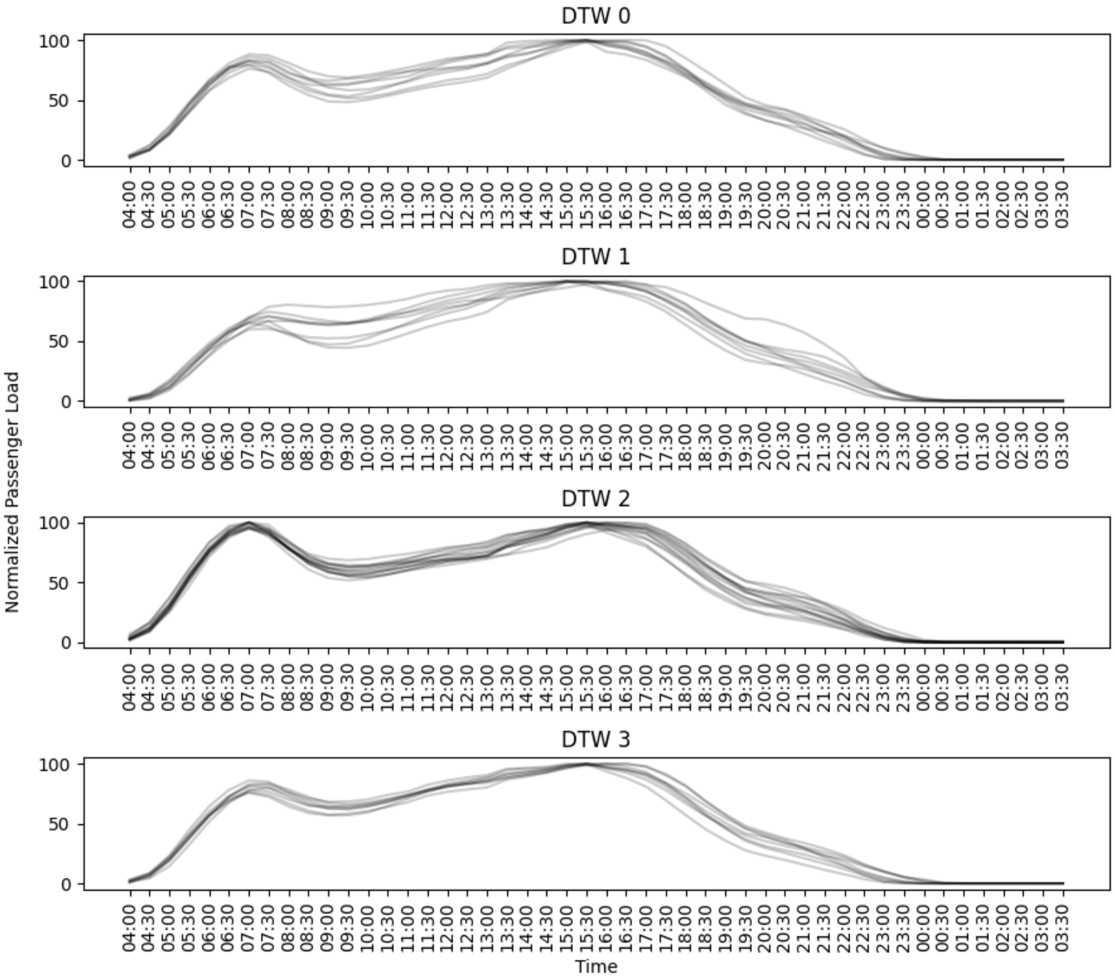

Figure 4.8. Results of DTW for Each Cluster

The results of BD in terms of cluster averages (Figure 4.9) are quite different than the other two method. Since it is more distinguishable based of demand trend levels than peak or dip areas. The separation areas of the DTW clusters are:

i) BD-0: From start point of the time series to afternoon peak at 15:30

ii) BD-1: Same with BD-3 before afternoon peak and same with BD-0 after afternoon peak

iii) BD-2: From morning dip at 9:30 to afternoon peak at 15:30

iv) BD-3: Same with BD-1 before afternoon peak and same with BD-2 after afternoon peak

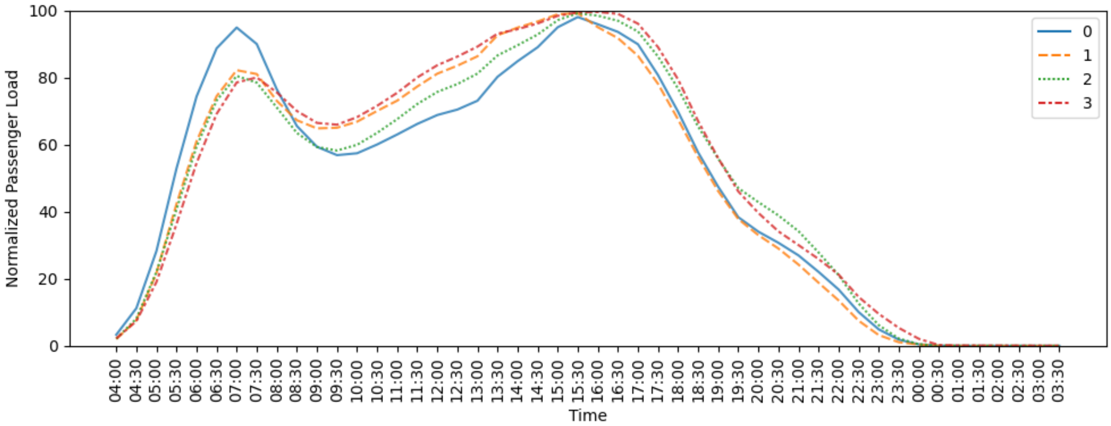

Figure 4.9. Results of BD Based on Cluster Averages

Same with DTW result clusters, BD result clusters also have low variance areas in their cluster results (Figure 4.10). But in BD, low variance areas have much lower variance as well as high variance areas have much higher variance.

Examples for low variance areas in cluster results of BD:

i) BD-0: Initial time interval to afternoon peak except morning dip and except having one outlier.

ii) BD-1: Morning dip to afternoon peak

iii) BD-3: All time intervals except morning peak

Examples for high variance areas in cluster results of BD:

i) BD-0: After afternoon peak

ii) Morning peaks of BD-1, BD-2 and BD-3

So it can be said that, BD method clusters the boarding patterns of bus lines based on demand trends since low variance areas generally on trends which have high amount of sequential time intervals and high variance areas are generally on peaks or dips.

Mentioned critical time intervals are the main decisive points for the BPC methods. It can be confirmed with clustering results since bus lines in clusters shows the similar characteristics either at time intervals that represent peak and dip areas or time intervals that represent demand trend areas.

Aside from the visual observations, this can be also proven numerically. Table 4.2 shows the average errors and standard deviations on critical time intervals and in general

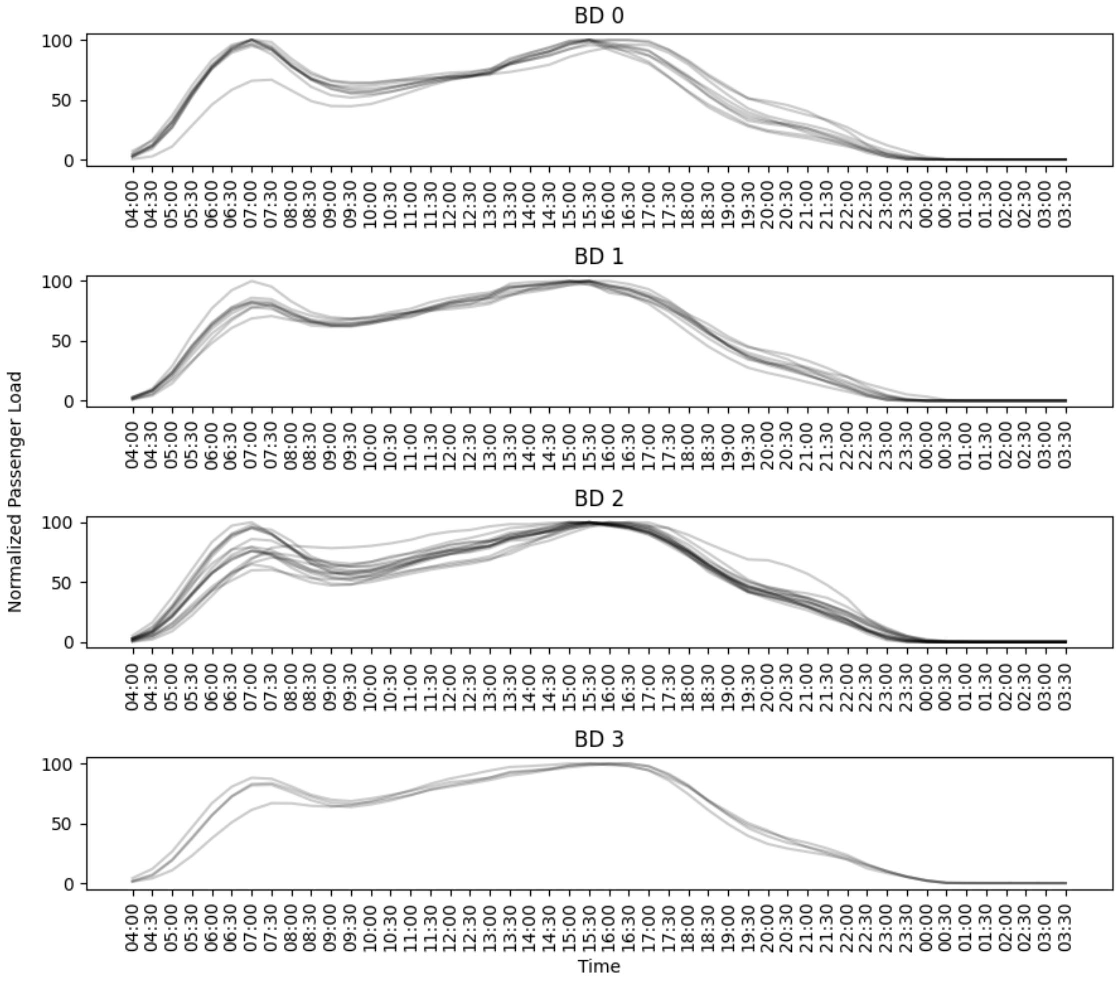

Figure 4.10. Results of BD for Each Cluster

for each BPC method.

The general peak areas for boarding patterns are 07:30 and 15:30 time interval. In these time intervals, ED shows the lowest mean errors while the standard deviation of DTW is slightly lower and BD has the highest mean error and standard deviation values.

In morning dip at 10:00, ED has the lowest mean error as well while BD has the lowest standard deviation value. This time, DTW has the largest values on both.

The evening peak area at 22:00 that can be observed in Figure 3.2 is disappeared after normalization. It is a minor peak and not much significant effect since the data is mainly consist of the months May and October in which tourists do not have a bigger effect on public transportation. Thus, 22:00 can be considered as demand trend area.

Table 4.2. Mean and Standard Deviation of Clustering Method Errors

Time Interval	ED Mean	DTW Mean	BD Mean	ED Std	DTW Std	BD Std
07:30	31.84	32.72	63.31	4.64	4.60	8.79
10:00	33.82	49.27	36.07	4.71	6.68	4.29
13:30	36.57	38.61	30.62	5.17	5.17	4.00
15:30	11.23	11.14	12.03	1.36	1.31	1.60
17:00	39.28	39.52	29.65	4.74	5.36	3.81
18:00	52.74	52.80	40.71	6.92	7.27	5.70
22:00	40.40	42.64	36.91	5.96	5.79	4.32
Daily Average	34.98	38.10	35.61	4.78	5.17	4.65

In other than peak and dip areas, BD has the lowest mean error and standard deviation values while DTW has the highest in general. However, ED has the lowest mean error in daily average while BD has the lowest standard deviation but difference between ED and BD in those values are quite low.

Table 4.3. Bus Lines Included in Same Clusters by All Methods

Core Clusters	ED	DTW	BD
LC07, KC06, LC07A	0	1	2
VL13A, VS18, DC15A, DC15, FL82, 511, AF04, VC59	1	2	0
VF02, VL13, TCP45, TC16A, AF04A	1	2	2
VF66, TB72, KPZ83	2	0	1
TC16, CV14	2	0	2
UC11, MF40, MC12, CV47	2	3	1
KM61, TK36	2	3	2
LF09, LF10	2	3	3
ML22, KF52	3	0	2
KL21, TCD49	3	1	2

Lastly it will be beneficial to mention that DTW might have big error numbers according to this type of comparison since it might cluster time series with sliding a time interval or two. But from transportation planner's point of view, planning with a DTW cluster of bus lines may needs time interval sliding in addition to maximum demand scaling.

Another interesting result can be shown with the intersection matrix in Figure 4.11. The intersection matrix shows that the bus lines are included together in clusters by which BPC method. Clustering methods are represented as RGB colors, ED is blue (Hex code:#0000FF), DTW is green (Hex code:#00FF00) and BD is red (Hex code:#FF0000). Colors are indexed as binary numbers by changing "00" and "FF" in their hex codes to 0 and 1 respectively. Thus, each digit of the binary number represents a clustering method that two bus stops coexist in same cluster. For example, KL08 and KC06 belongs to same clusters in ED and DTW but not BD, so their coexistence is 011 (3) and color is cyan (Hex code:#00FFFF). However in Figure 4.11, white (Hex code:#FFFFFF) and black (Hex code:#000000) are reversed for a better display.

The black diagonal in Figure 4.11 is inevitable. But other black coexistence (Hex code:#000000, Binary:111, Index:7) bus line couples are included in same clusters by all BPC methods. These black coexistences are named as core clusters and summarized in Table 4.3. The table shows, which core clusters are included in which result clusters in BPC methods.

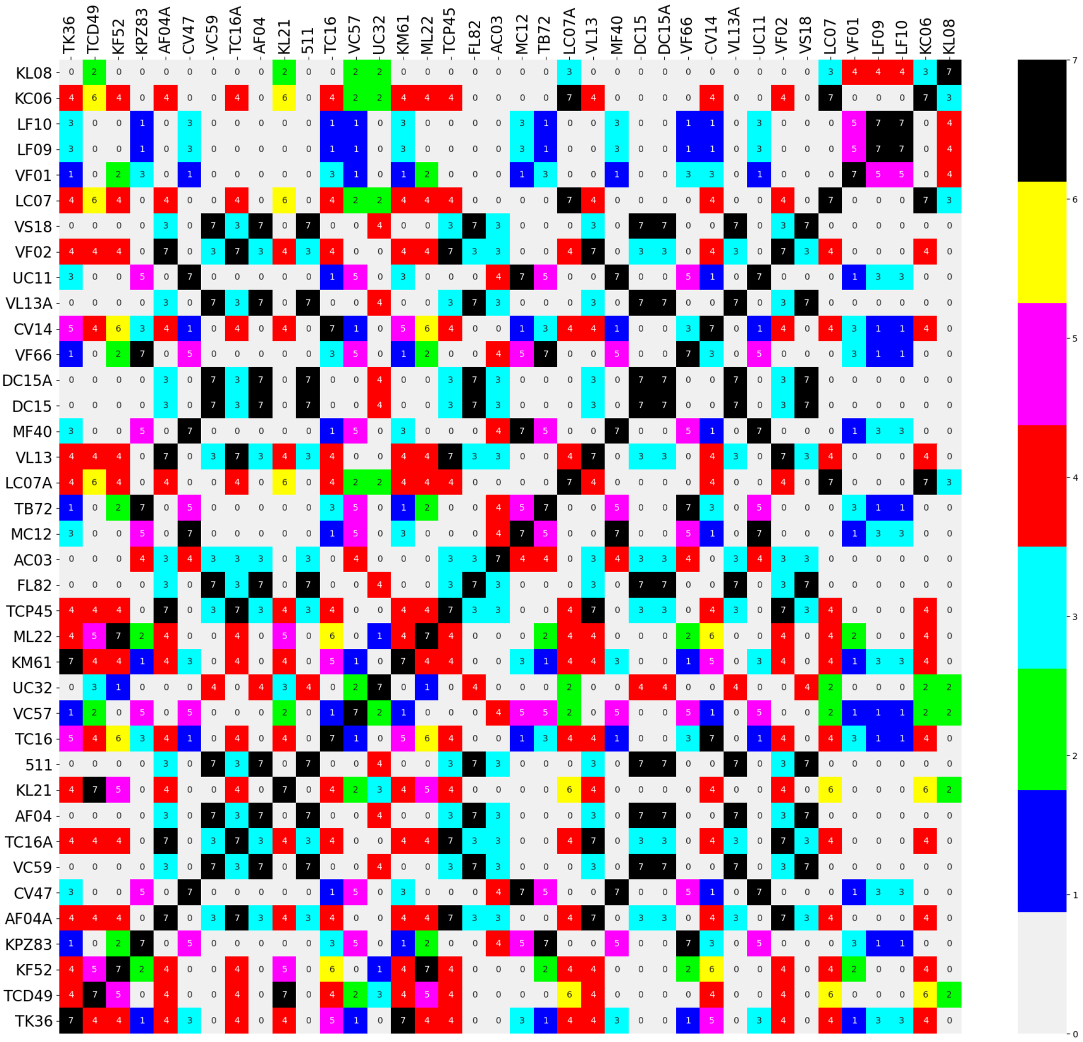

Figure 4.11. Intersection Matrix of BPC results

4.2. Time Slot Clustering

TSC is a clustering approach to determine time slots of a bus line. Thus, two of the most popular bus lines are selected in order to test the results of TSC. These bus lines are KL08 and VF01.

The inputs for TSC are the daily average demands of bus lines in time series format (Figure 3.1) and threshold values for determining the boundaries of the resulting time slots.

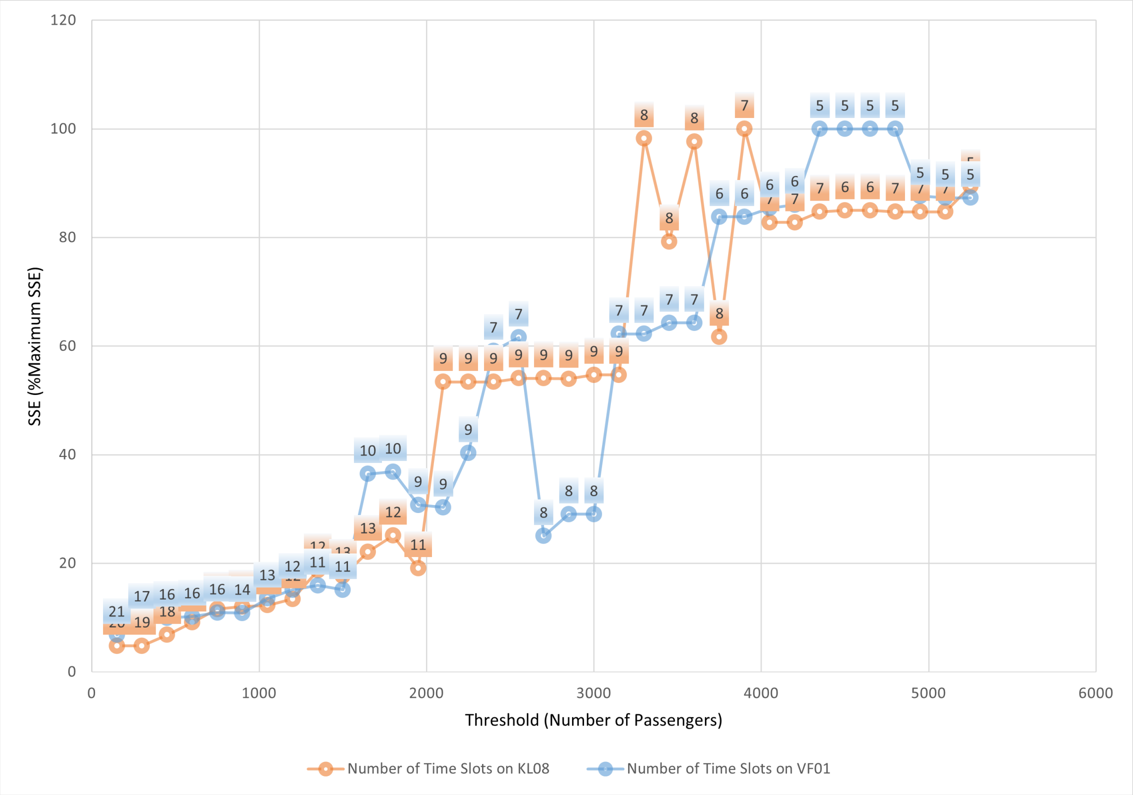

Figure 4.12. Elbow Method Results of KL08 and VF01 for STSC Threshold

Elbow method-like approach is used to determine the thresholds for the TSC algorithms. In Figure 4.12, result of the elbow method approach for STSC is displayed. STSC algorithm executed 35 times with increasing the passenger threshold by 150 in every iteration for both of the bus lines. As mentioned in methodology section, passenger threshold with the largest decrement is selected as the STSC input. The selected passenger thresholds are 3750 and 3000 passengers with 8 time slots each for KL08 and VF01 respectively.

The results of the STSC method on KL08 with given inputs are displayed on Figure 4.13 and Table 4.4. It can be said that the algorithm divides the KL08 daily demand quite fine as peak, dip and trend areas. The highest peak of KL08 is occurred at 15:30 which resides in time slot 3 and the second highest peak is occurred at 18:00 with 99.23% normalized passenger load which resides in time slot 4. The evening minor peak area (time slot 5), the morning dip area (time slot 1) and the noon increasing trend (time slot 2) are also identified as well.

Aside from the shape of the boarding pattern, average number of boarding passengers in time slot 3 is higher than in time slot 4 even though time slot 4 is longer and more passengers boarded in time slot 4.

The results of the STSC method on VF01 with given inputs are displayed on Figure 4.14 and Table 4.5. Like the STSC results of KL08, all demand areas identified well. The only exception is time slots 1 and 2 identify the both sides of a single peak. Thus, these two time slots may be merged in an additional post processing phase.

The maximum number of boarding passengers is belongs to time slot 3 but times slot 4 and merged time slots 1 and 2 have bigger average number of boarding passengers since both of these time slots identify the peak areas and time slot 3 identifies increasing noon trend.

It can be said that STSC algorithm achieves its purpose since it identifies the demand areas of the individual bus lines that have different boarding patterns. Even though the occurring time intervals of the peaks are nearly the same, areas are quite different and STSC algorithm achieves the identify differences. Tables 4.4 and 4.5 are also show that the time slot start and end time intervals do not overlap between the bus lines.

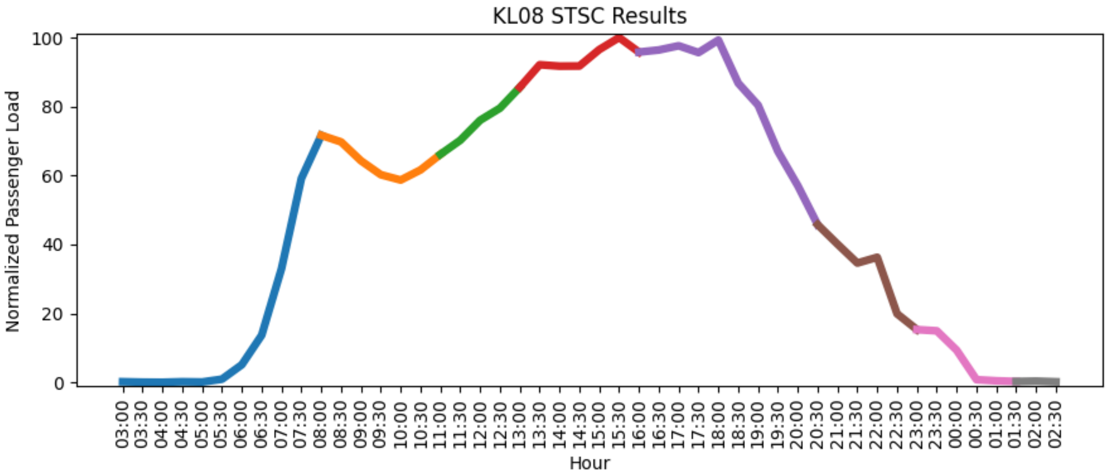

Figure 4.13. Results of STSC on KL08

Table 4.4. Results of STSC on KL08

Time Slots	Start Time	End Time	# of Passengers	Average # of Passengers
0	03:00	07:30	1002	100
1	08:00	10:30	3443	574
2	11:00	12:30	2603	651
3	13:00	15:30	4973	829
4	16:00	20:00	6919	769
5	20:30	22:30	1574	315
6	23:00	01:00	363	73
7	01:30	02:30	7	2

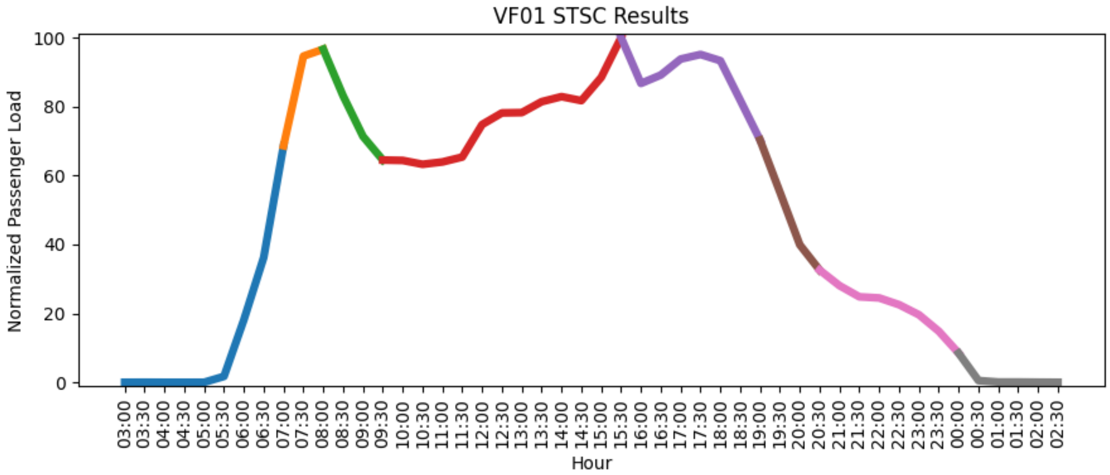

Figure 4.14. Results of STSC on VF01

Table 4.5. Results of STSC on VF01

Time Slots	Start Time	End Time	# of Passengers	Average # of Passengers
0	03:30	06:30	296	42
1	07:00	07:30	862	431
2	08:00	09:00	1326	442
3	09:30	15:00	4683	390
4	15:30	18:30	3378	483
5	19:00	20:00	873	291
6	20:30	23:30	881	126
7	00:00	02:30	48	8

Elbow method is used to determine the tolerance angle threshold for ATSC algorithm. Tolerance angles for bus lines KL08 and VF01 are determined as 13°with 6 time slots and 27°with 8 time slots respectively.

The results of ATSC on KL08 within given inputs are displayed on Figure 4.15 and Table 4.6. ATSC identifies the demand trends on time slots 0, 1, 3 and 5. At first glance, it can be said that ATSC do not identify the decreasing demand trend between 8:00 and 10:00. But, ATSC uses linear regression to identify the demand trends. Thus, mini trend breaks can be supposed as mini trend pauses that occurs within the boundaries that determined by the tolerance angle. So for a long running demand trend of time slot 1, it can be assumed as an insignificant trend pause that eventually breaks the trend at morning dip instead of morning peak.

The micro trends in time slots of 2 and 4 have only two time intervals each. So, the time intervals inside these time slots can be distributed to their neighbouring time slots in an additional post processing phase.

The results of ATSC on VF01 within given inputs are displayed on Figure 4.16 and Table 4.7. ATSC is resulted with more time slots than needed because of the volatile demand during daytime causes ATSC to identify too many micro trends. These micro trends may also be merged during an additional post processing phase. The result of the merge process can be changed according to the transportation planner's needs.

Overall, ATSC has quite fine results but it can be improved. It seems like it can be affected by outliers severely, so an additional post processing phase may become mandatory to interpret the results in planning phase. This may be also solved by adjusting the tolerance angle. But increasing it too much may cause the starting trend to be more powerful and this makes it much difficult to break the first demand trend.

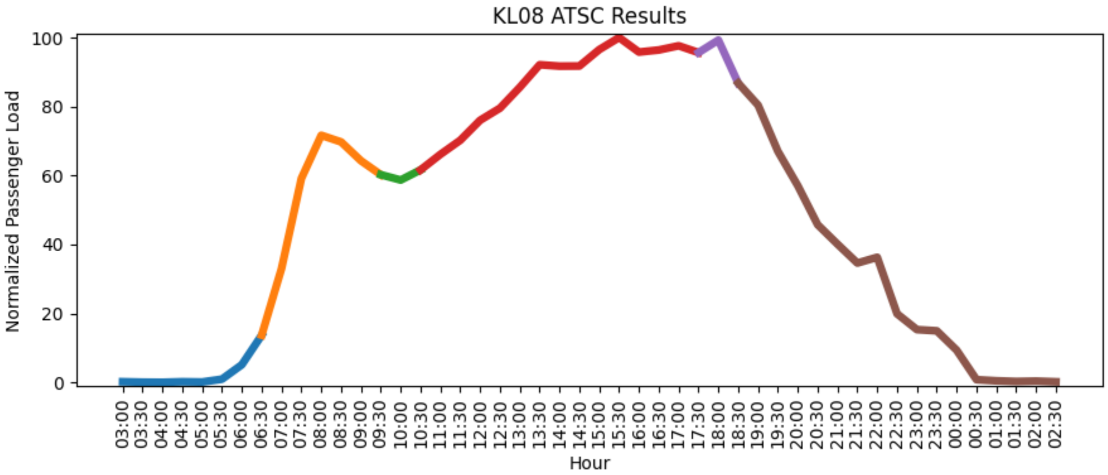

Figure 4.15. Results of ATSC on KL08

Table 4.6. Results of ATSC on KL08

Time Slots	Start Time	End Time	# of Passengers	Average # of Passengers
0	03:00	06:00	58	8
1	06:30	09:00	2778	463
2	09:30	10:00	1060	530
3	10:30	17:00	10710	765
4	17:30	18:00	1738	869
5	18:30	02:30	4541	267

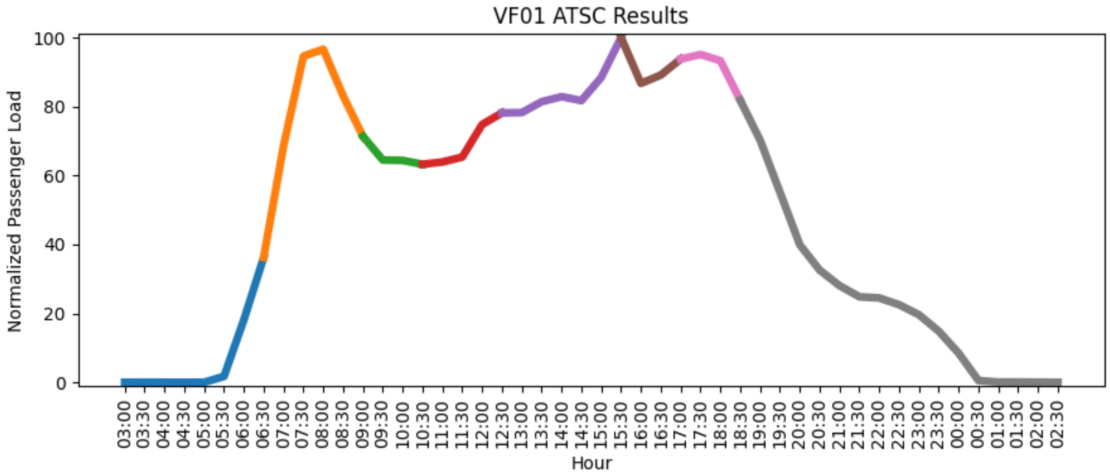

Figure 4.16. Results of ATSC on VF01

Table 4.7. Results of ATSC on VF01

Time Slots	Start Time	End Time	# of Passengers	Average # of Passengers
0	03:30	06:00	105	17
1	06:30	08:30	2002	400
2	09:00	10:00	1057	352
3	10:30	12:00	1411	353
4	12:30	15:00	2592	432
5	15:30	16:30	1457	486
6	17:00	18:00	1522	507
7	18:30	02:30	2234	131

The resulting time slots can be used for bus frequency planning. Identifiers of time slots, mean passenger count and linear regression of passenger counts within time intervals, may used to determine the bus frequencies within each time slot. To compare the frequency setting with TSC methods and general frequency approaches:

i) Fixed Schedule (FS): It is assumed that a fixed bus schedule is used for the entire day. The bus frequency is adjusted to satisfy the demand of the daily average Ceder 2016.

ii) Given Schedule (GS): It is assumed that a predetermined bus schedules are used for rush hours and non-rush hours. Determined rush hours in literature are 6:00-7:00 and 16:30-17:30 by T. Li et al. 2018, around 12:00 and between 16:00–18:00 by Mohamed et al. 2016. Rush hours for GS is selected as 7:00-9:00 and 16:00-18:00 since most of the peaks are occurred during these hours. It is assumed that the bus frequencies in GS are the average passenger count for rush hours and non-rush hours as two group.

iii) Stepped Schedule (SS): In SS, the results of the STSC are being used. Since average passenger counts are considered for STSC, bus frequencies are adjusted to satisfy these average demands for SS.

iv) Adaptive Schedule (AS): Same with SS. But bus frequencies in each time interval are adjusted to fit the linear regressions of the relevant time slots.

The estimated demand levels of each method are displayed on the average daily demand of KL08 in Figure 4.17. Bus frequency setting with FF is the most inefficient since the most of the daytime it will fail to satisfy the demand and in the night time it will be costly to run services for low demand.

As for GS, the only reliable time is the frequency setting in morning rush hour. But for other daytime hours, operating busses will be overloaded.

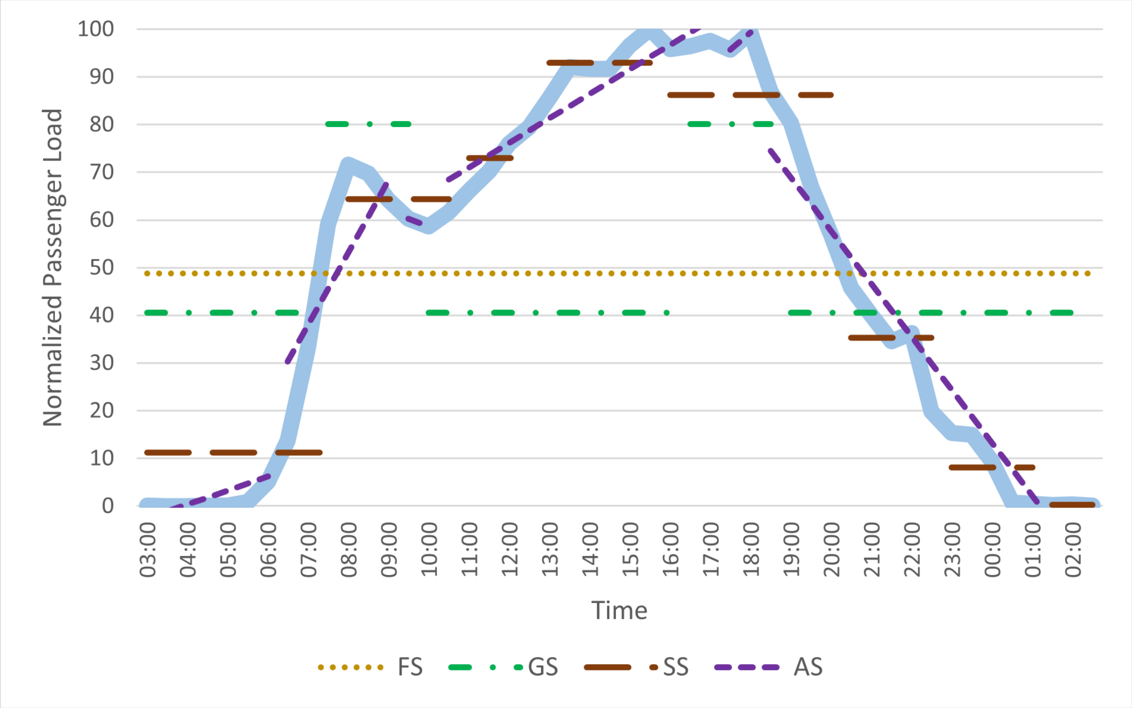

Figure 4.17. Comparison of The Methods on KL08

If bus frequency setting is done with SS, the demand in most of the time intervals will be satisfies. Inefficiency may be occurred at starting and ending time intervals of the time slots. Since some of these time intervals, the demand is increased or decreased dramatically. For example, operating busses will be overloaded at 07:00 and busses will operate for low demand at 20:00.

This problem can be solved in bus frequency setting with AS. Because of the frequency is fit to linear regression of time slots, bus frequency is adjusted dynamically over time intervals. As observed from the Figure 4.17, bus frequency is always set to satisfy at least 80% and at most %120 of the demand on each time interval.

4.3. Alighting Estimation and Occupancy

In order to compare three alighting estimation methods (ARR, DRR and TC), the most popular and balanced bus lines are selected. The criteria for selecting these bus lines are having more than 3000 daily passengers on each route and the difference between boarding counts of routes has to be less than 10%. The bus lines that satisfy this criteria are KL08, KC06, LC07, LF10, UC11 and VL13.

As mentioned in methodology section, the only bus line that have real life data is KL08. So only KL08 used to compare methods with real life data. The selected 6 bus lines are only used for comparing the methods with each other.

The comparison with each other is done by two error calculations:

i) Distance from Mean (DFM): The total distance from the average alighting estimations made by all of 3 methods for each BSG. DFM is used as error value for each method.

ii) Distance from Trip Chaining (DFTC): The total distance from the estimations made by TC for each BSG. DFTC is used as error value for comparing only ARR and DRR since both of these methods have potential to produce close alighting estimation results.

DFM errors can be observed for each method in Figures 4.18 and 4.19 as box and whiskers plots. The plot shows the distribution of DFM values, the middle line for each box represents the median value. Each of the box and whiskers plot scaled same (between -600 and 800 passenger miscalculation) in order to observe the error variation differences.

According to the Figures 4.18 and 4.19, DRR has much less DFM errors in general since it has low margin of errors and have much less errors that the other two methods. ARR has lower DFM error margin for estimations on VL13 and LC07. Estimations of TC has the most distant values from the three method averages.

The same results also can be said with the Table 4.8 which shows the average absolute values of the DFM errors. ARR has the lowest averages on VL13 and LC07 and DRR has the lowest average DFM errors on the other 4 bus lines. It can be also noted that the VL13 and LC07 has the 1^{st} and 3^{rd} lowest daily passenger counts among the selected bus lines.

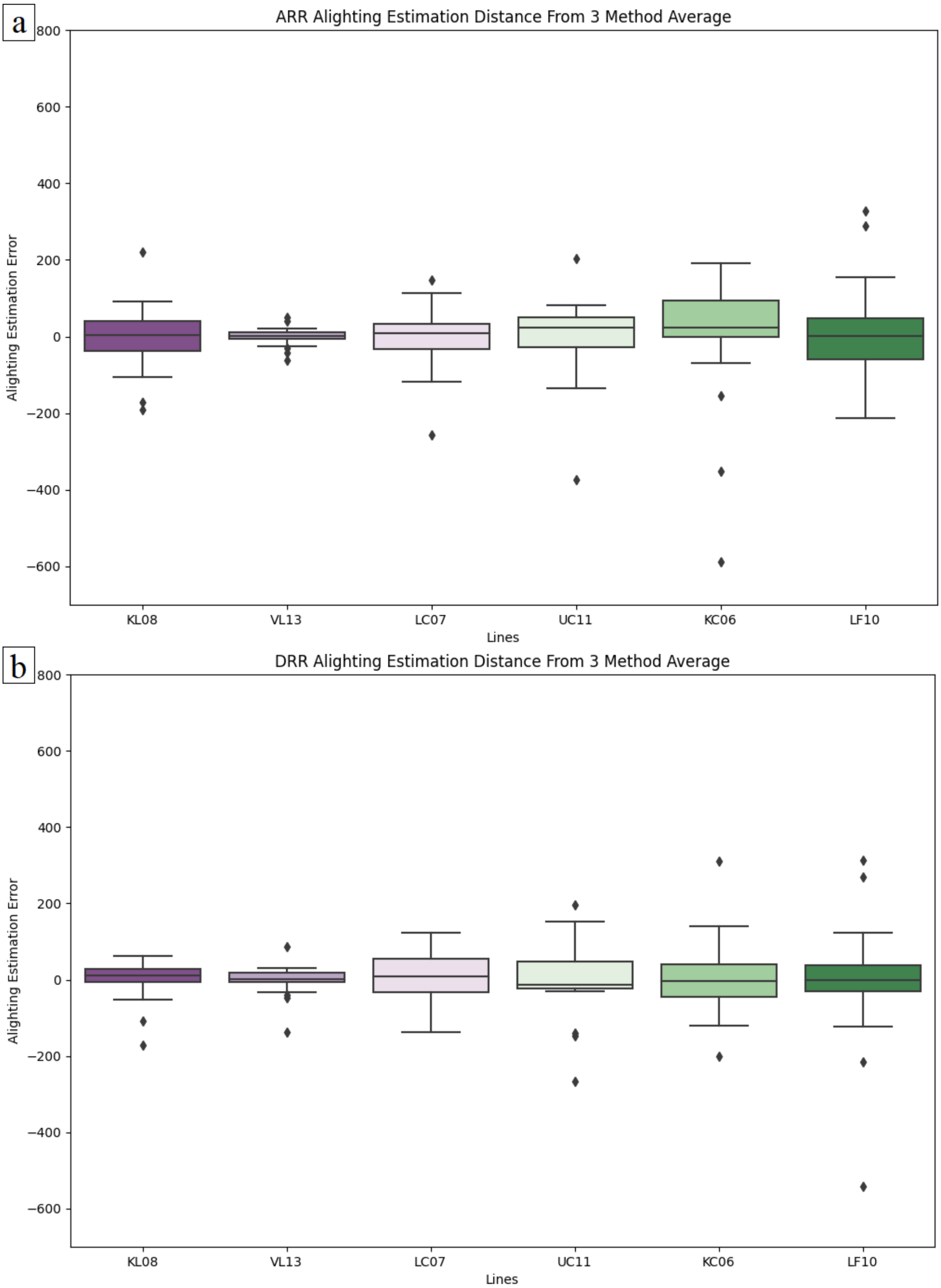

Figure 4.18. DFM Error Distribution on Box and Whiskers Plot; **a)** ARR; **b)** DRR

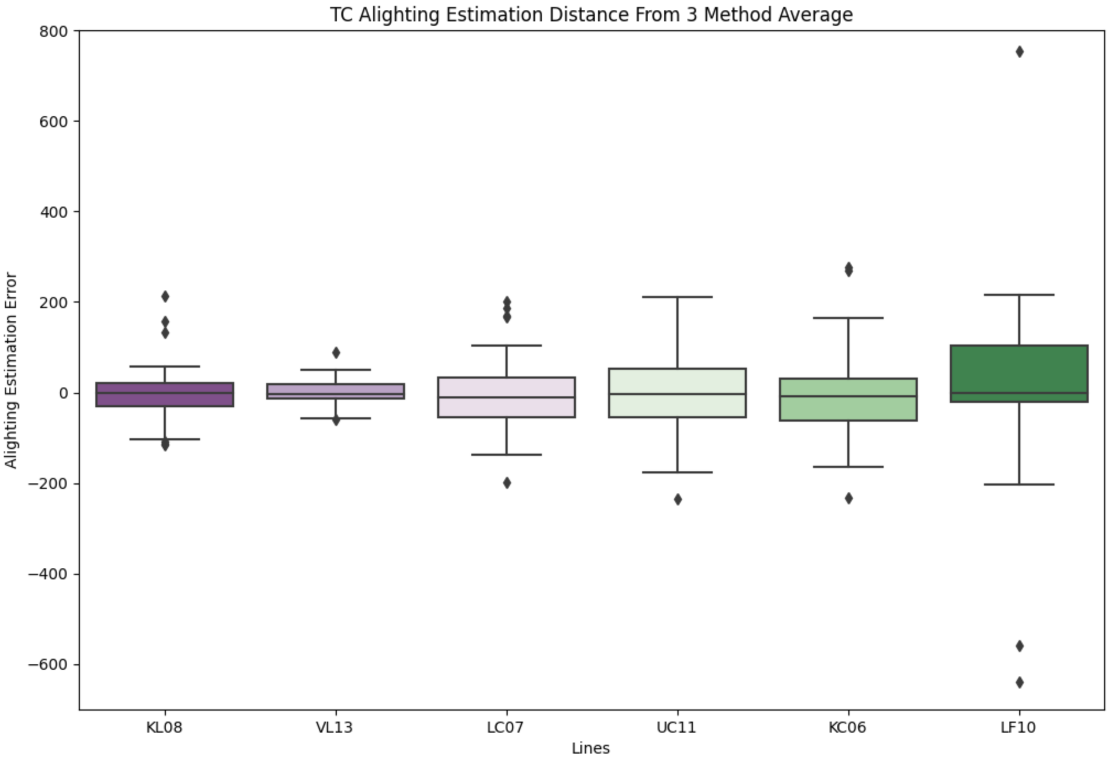

Figure 4.19. DFM Error Distribution on Box and Whiskers Plot (TC)

Table 4.8. Sum of Absolute Values of DFM Errors

Bus Line	ARR	DRR	TC	Passenger Count
KL08	56.37	31.94	51.51	9724
VL13	15.49	22.27	23.64	3225
LC07	48.47	57.89	68.47	5129
UC11	72.51	67.71	73.37	4541
KC06	95.15	65.69	79.03	9111
LF10	94.90	86.63	142.22	8304

As mentioned above ARR and DRR have potential to produce close results. This may be cause the DFM errors of the two methods will be lower than the TC. It is like, taking the average of two small numbers and one large number. The average will be always closer to the two small numbers than the large number. Thus, DFM results are only valid when the average estimations of the 3 method assumed as the real alighting counts.

That makes an another error definition necessary. In DFTC, it is assumed that the real alighting counts are the estimations made via TC method. DFTC results are also represented same with DFM results in Figure 4.20 and Table 4.9. The results are same with DFM and error values are bigger as expected. DFTC errors are increased to approximately 2 times from DFM error. As it can be seen from the scale of the box and whiskers plots are increased to -1500 and 1000 from -600 and 800.

According to the results of alighting estimations, it can be said that DRR estimations generally closer to the TC estimations. But, ARR tends to produce closer estimations when daily passenger count is relatively lower.

Table 4.9. Sum of Absolute Values of DFTC Errors

Bus Line	ARR	DRR	Passenger Count
KL08	105.04	59.65	9724
VL13	29.63	42.19	3225
LC07	105.93	119.20	5129
UC11	138.38	105.71	4541
KC06	169.24	95.84	9111
LF10	231.88	218.00	8304

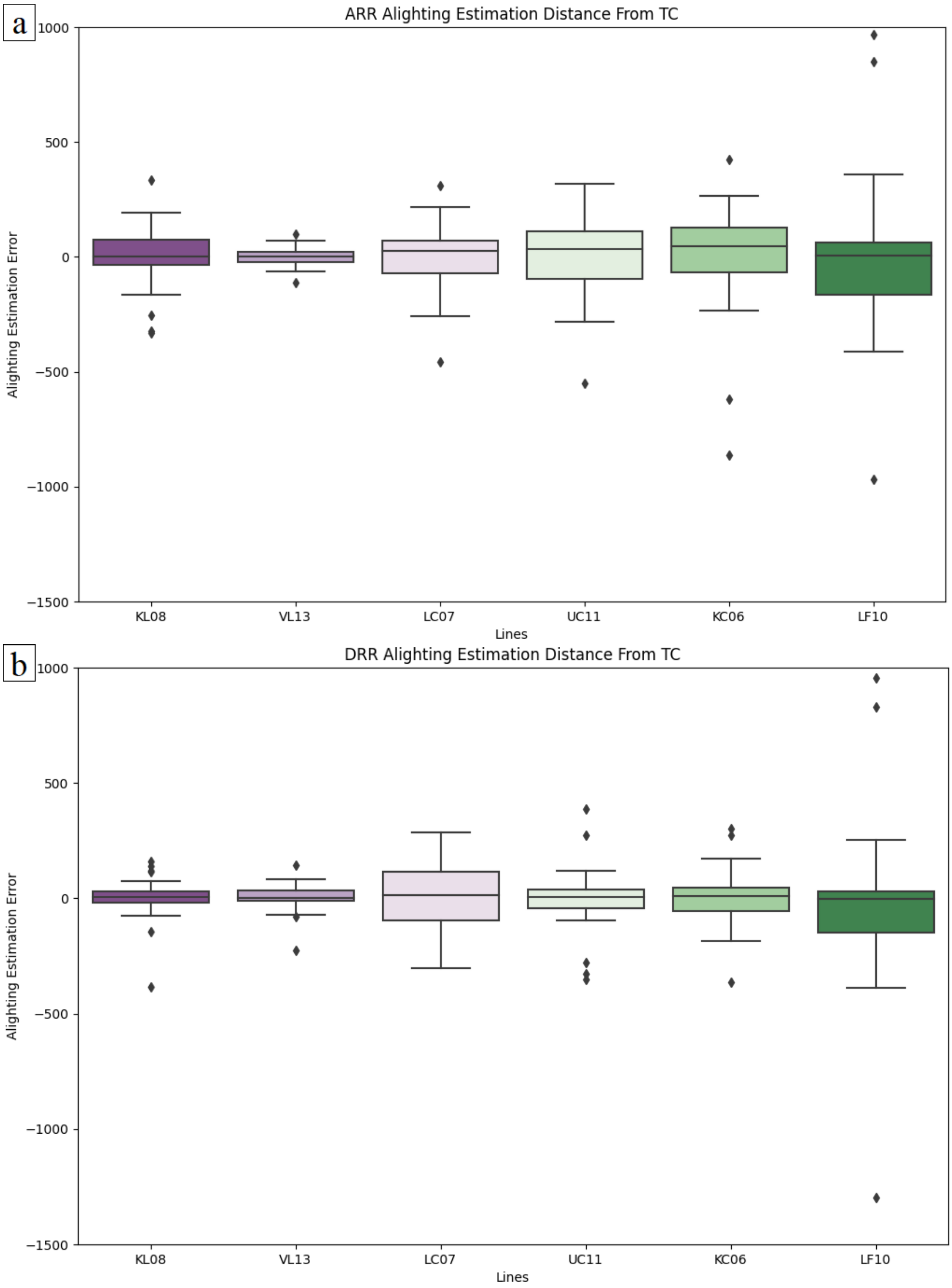

Figure 4.20. DFTC Error Distribution on Box and Whiskers Plot; **a)** ARR; **b)** DRR

A more clear comparison can be made on occupancy calculations by the alighting estimations of these methods. The occupancy estimation results can be observed in Figures 4.21 and 4.22. The black line on figures is the average occupancy on BSGs.

i) KL08: The general occupancy pattern is followed by all three methods but ARR produces higher occupancy values for the middle BSGs.

ii) LF10: Same with KL08. Occupancy pattern is followed but the drop on 13^{th} BSG on ARR and DRR is occurred 14^{th} BSD on TC.

iii) VL13: Same with KL08. On middle BSGs, DRR has higher occupancy than ARR. Later it drops and give similar results with TC.

iv) KC06: DRR and TC produces similar results but ARR can be even considered as outlier since its results are exceptionally higher. Results of all three methods are following the same pattern.

v) LC07: All of the methods produces different results and occupancy patterns. But results of TC are more distant.

vi) UC11: Same with LC07.

Overall, each of the three methods have their own results but generally ARR tends to produce higher and TC tends to produce lower occupancy results. Event though, they all followed same occupancy patterns most of the time, it is hard to say anything about accuracy without a real life data.

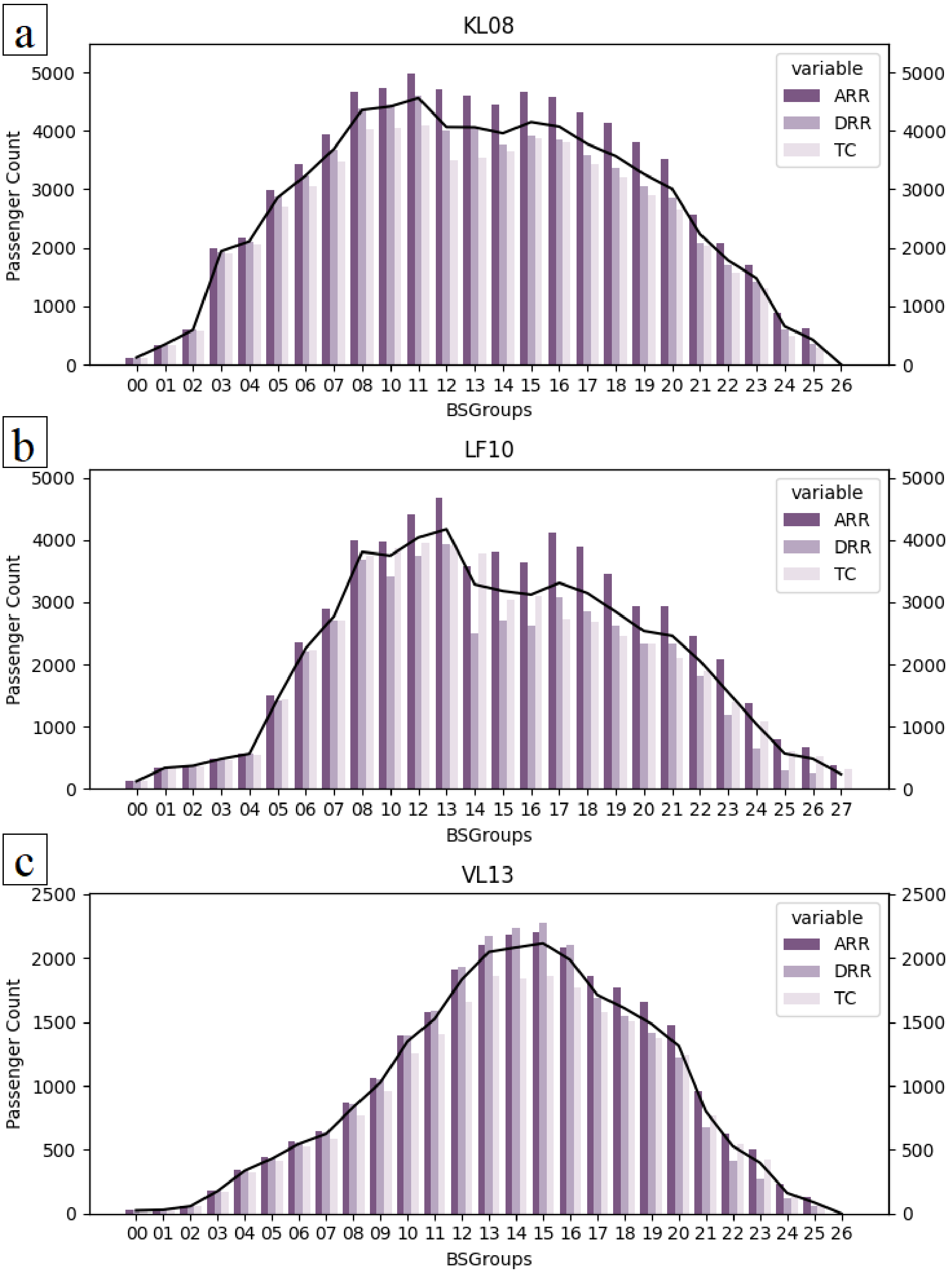

Figure 4.21. Occupancy Estimation Results of; **a)** KL08; **b)** LF10; **c)** VL13

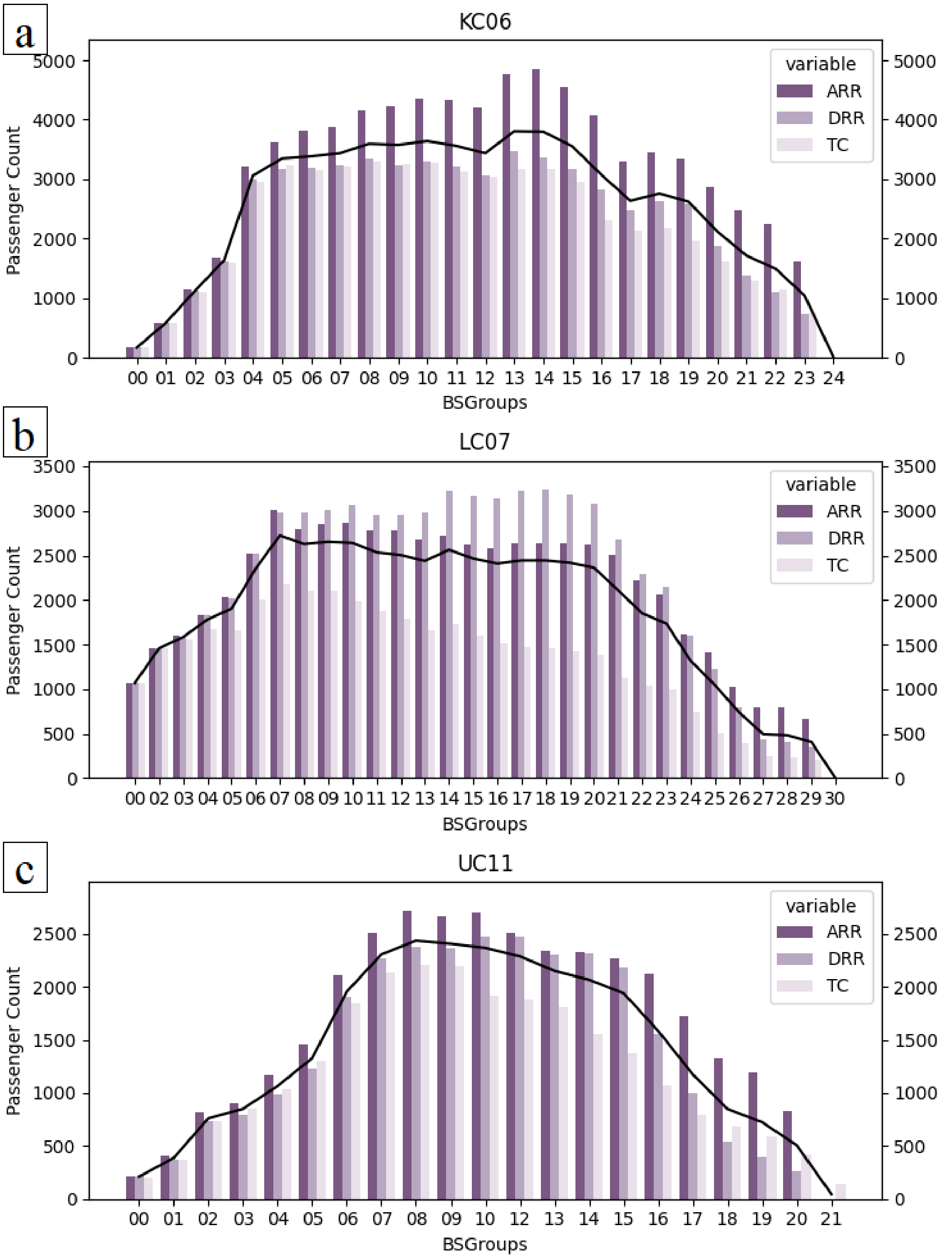

Figure 4.22. Occupancy Estimation Results of; **a)** KC06; **b)** LC07; **c)** UC11

The details about the real life data are already given in methodology section. Prediction matrices for each method that produced from 18 December 2019 are used to predict the alightings of trips that have the video records. (Figure 4.23a)

Box and whiskers plot of the errors are given in Figure 4.23b. According to the plot, ARR has the most accurate predictions and error variance is lowest among the three methods. Even the variance seems pretty higher, the median of the errors of DRR is closer to 0 than the median of TC.

This can be also verified from the Table 4.10 where the average and standard deviation of the absolute values of alighting estimation and occupancy errors are listed.

After the occupancy estimations are calculated (Figure 4.24a, in where the black line represents the real life occupancy). The results are similar to the occupancy estimations done when comparing methods with each other. All methods follows the occupancy pattern but ARR produce higher and TC produces lower results. Interestingly, all methods have failed to estimate closely the last BSGs of KL08.

In occupancy error comparison (Figure 4.24b), it can be seen that the median of the DRR is close the 0. But the estimations of ARR seems to closer to 0.

However, the average absolute occupancy error of DRR is the lowest among the three methods' predictions (Table 4.10). Thus, occupancy-vise DRR produced the most accurate results. The standard deviation of occupancy errors of ARR is the lowest and occupancy estimation in most of the BSGs are higher than the actual occupancy (Figure 4.24). That means, even ARR does not have the highest accuracy, the occupancy predictions of ARR follows the real life occupancy patterns better.

Table 4.10. Estimation Error on Real Life Data

Method	A.E.E. Mean	A.E.E. St.Dev.	O.E. Mean	O.E. St.Dev.
ARR	3.19	2.94	8.23	5.46
DRR	3.62	3.72	6.73	6.14
TC	4.23	4.32	9.15	7.96

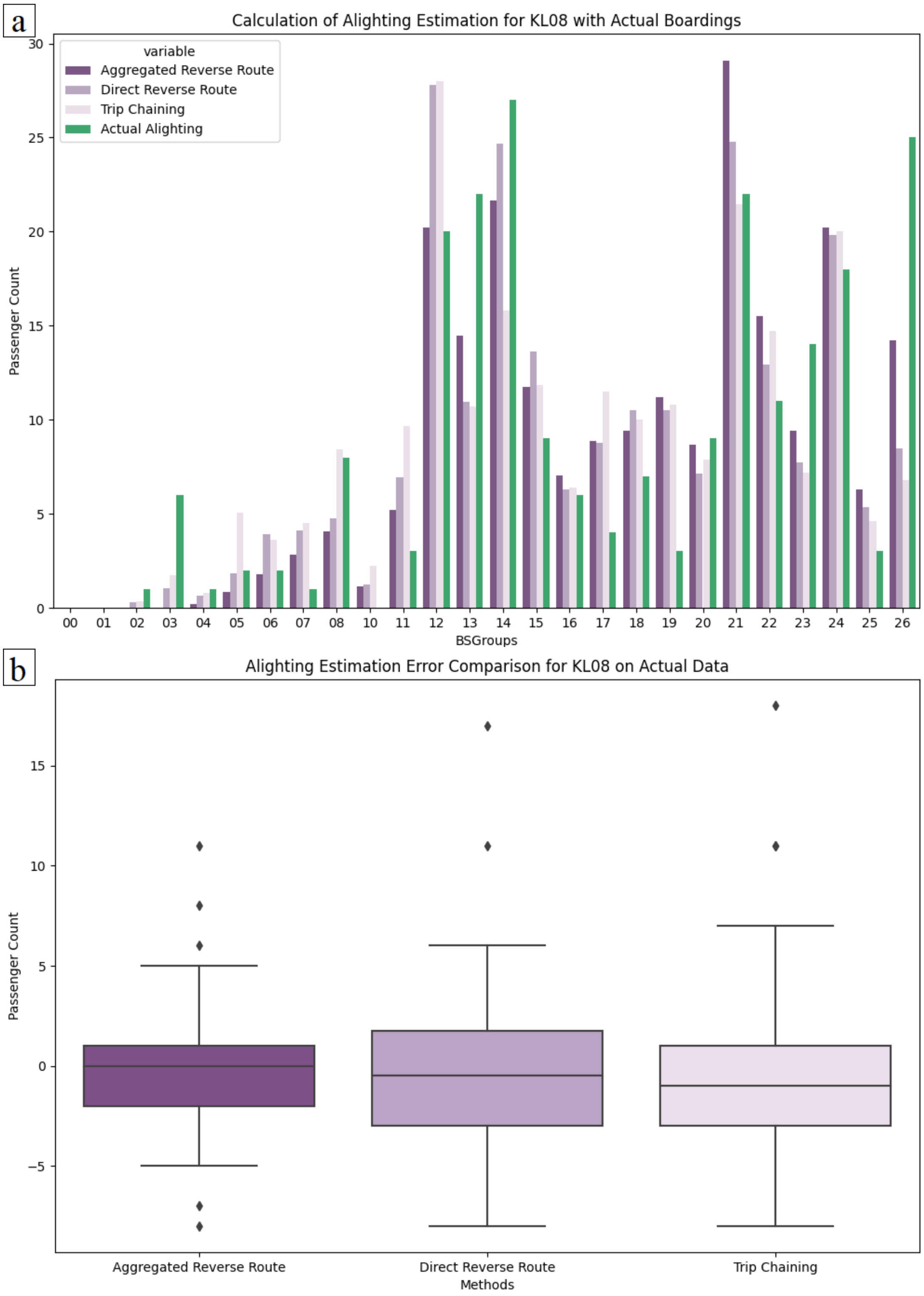

Figure 4.23. Alighting Estimation Verification with Real Life Data; **a)** Alighting Calculations; **b)** Alighting Error Comparison Between Methods

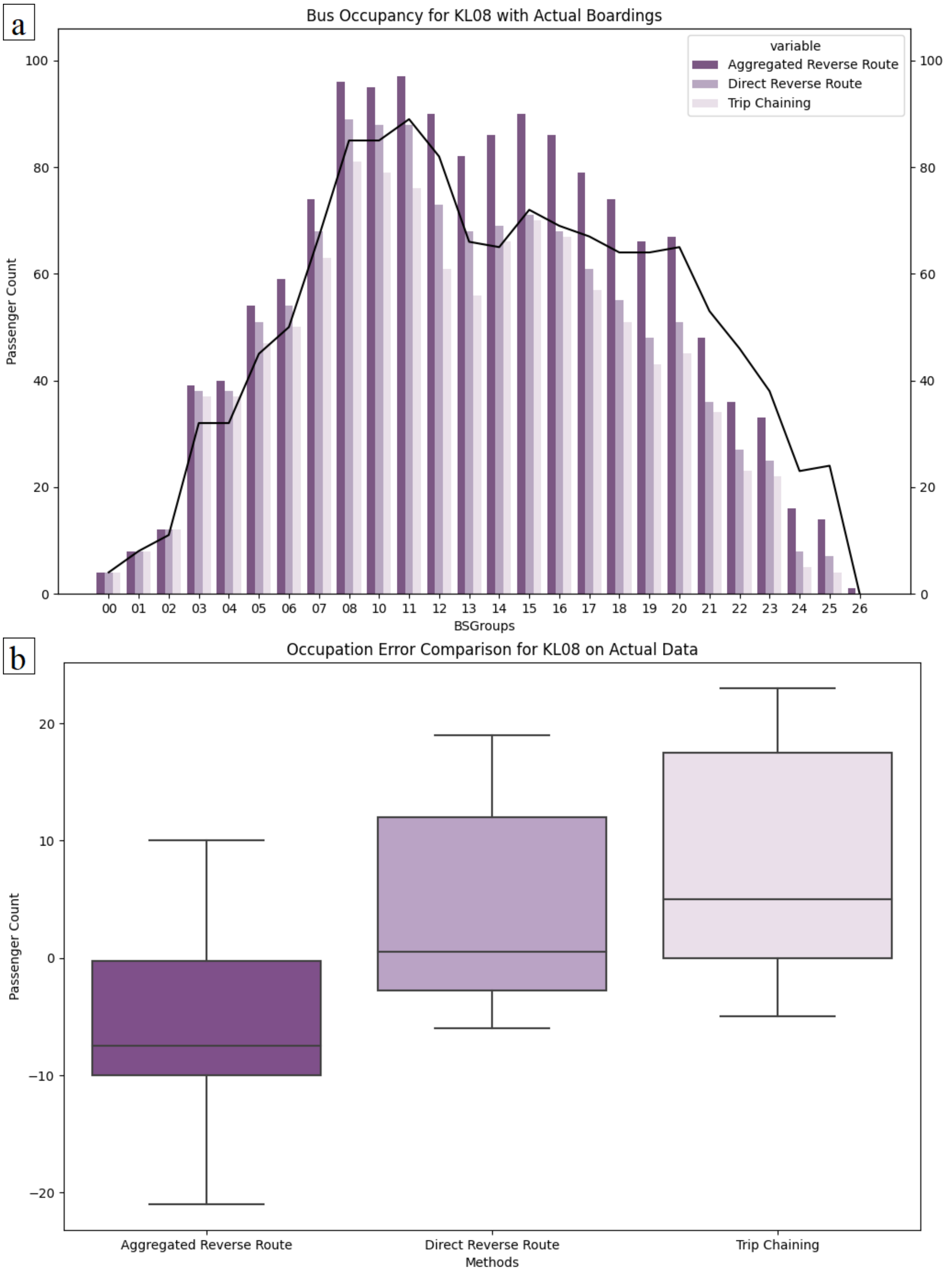

Figure 4.24. Occupancy Verification with Real Life Data; **a)** Occupancy Calculations; **b)** Occupancy Error Comparison Between Methods

5. CONCLUSION

This study proposes various types of methods that can be used to optimize or redesign bus lines of a public transportation system. The daily demand of bus lines can be represented as boarding patterns when the weight of popularity ignored by normalizing each bus line according to its maximum number of boarding counts. Clustering these boarding patterns may help to detect bus lines with similar types of passenger activities. This type of information has high potential that may enable transportation planner to come up with new approaches for strategical or tactical planning decisions such as frequency setting problem.

The proposed BPC methods (ED, BD, DTW) emphasizes on different aspects of boarding pattern similarities such as peak-dip areas, trend areas or balanced between these two, respectively. Each of these approaches have their own advantages or disadvantages such as decreased accuracy on non emphasized areas or in general. Also, BPC methods can be used together in order to produce core clusters, which consist of the bus lines that end up in same clusters in all methods. Thus, core clusters indicate much more similar bus lines on which the more detailed plannings can be implemented on them together.

The peak-dip areas and trend areas in daily demand of bus lines can be detected by TSC methods of STSC and ATSC respectively. A sequence of time intervals on a daily demand that clustered by these methods is named as time slot. These time slots can be used to design or optimize dynamic frequency settings for a transportation system. Using dynamic frequency settings that adapts daily passenger activities will prevent busses to overload, which causes significant decrease on passenger comfort, and decrease the cost penalty that stems from operating trips for insufficient number of passengers. Additionally, BPC and TSC can be used together to set frequencies of the bus lines that have similar daily demand characteristics.

Alighting estimation of a route can be done with a probabilistic approach that based on the values obtained by reverting the route. In reverse route mentality, opposite direction counterparts of bus stops and time slots are used. It can be done by aggregations based on BSGs and STSC or simply directly reversing the route.

Reverse route approach provides faster alighting estimations that can be implemented on every desired bus line or route rather than analyzing the trips of each passenger. Also, this approach can be used on single daily data and the boarding data that does not have sufficient transfer data. The alighting prediction matrix that calculated by reverse route methods can be used on a single trip or future boarding data. The occupancy results that based from the alighting predictions via reverse route are mostly accurate and can be used in various different transportation planning decisions.

6. REFERENCES

Ait-Ali, Abderrahman and Jonas Eliasson (2019). “Dynamic origin-destination-matrix estimation using smart card data: an entropy maximization approach”. In: *Rail-Norrköping2019. Norrköping, Sweden.*

Alsger, Azalden et al. (2018). “Public transport trip purpose inference using smart card fare data”. In: *Transportation Research Part C: Emerging Technologies* 87, pp. 123–137.

Barry, James J, Robert Freimer, and Howard Slavin (2009). “Use of entry-only automatic fare collection data to estimate linked transit trips in New York City”. In: *Transportation research record* 2112.1, pp. 53–61.

Bera, Sharminda and KV Rao (2011). “Estimation of origin-destination matrix from traffic counts: the state of the art”. In.

Berthold, Michael R et al. (2007). “KNIME: The Konstanz Information Miner”. In: *Studies in Classification, Data Analysis, and Knowledge Organization.*

Briand, Anne-Sarah et al. (2016). “A mixture model clustering approach for temporal passenger pattern characterization in public transport”. In: *International Journal of Data Science and Analytics* 1.1, pp. 37–50.

Bulut, Batuhan (Apr. 2021). “Creating Agile and Optimal Transportation System”. MA thesis. Antalya: Akdeniz University.

Ceder, Avishai (2016). *Public transit planning and operation: Modeling, practice and behavior*. CRC press.

Cheng, Zhanhong, Martin Trépanier, and Lijun Sun (2020). “Probabilistic model for destination inference and travel pattern mining from smart card data”. In: *Transportation*, pp. 1–19.

Cui, Alex (2006). “Bus passenger origin-destination matrix estimation using automated data collection systems”. PhD thesis. Massachusetts Institute of Technology.

Dubos-Golain, Aurélie, Martin Trépanier, and Catherine Morency (2017). *Understanding transit use patterns in Montreal*. Tech. rep. CIRRELT, Centre interuniversitaire de recherche sur les réseaux d’entreprise ...

Faroqi, Hamed and Mahmoud Mesbah (2021). "Inferring trip purpose by clustering sequences of smart card records". In: *Transportation Research Part C: Emerging Technologies* 127, p. 103131.

Farzin, Janine M (2008). "Constructing an automated bus origin–destination matrix using farecard and global positioning system data in Sao Paulo, Brazil". In: *Transportation research record* 2072.1, pp. 30–37.

Hadas, Yuval and Matan Shnaiderman (2012). "Public-transit frequency setting using minimum-cost approach with stochastic demand and travel time". In: *Transportation Research Part B: Methodological* 46.8, pp. 1068–1084.

Harrison, Gillian, Susan M Grant-Muller, and Frances C Hodgson (2020). "New and emerging data forms in transportation planning and policy: Opportunities and challenges for "Track and Trace" data". In: *Transportation Research Part C: Emerging Technologies* 117, p. 102672.

Hazelton, Martin L (2010). "Statistical inference for transit system origin-destination matrices". In: *Technometrics* 52.2, pp. 221–230.

He, Li, Bruno Agard, and Martin Trépanier (2020). "A classification of public transit users with smart card data based on time series distance metrics and a hierarchical clustering method". In: *Transportmetrica A: Transport Science* 16.1, pp. 56–75.

He, Li, Neema Nassir, et al. (2015). *Validating and calibrating a destination estimation algorithm for public transport smart card fare collection systems*. Vol. 52. CIRRELT.

Huang, Di et al. (2020). "A method for bus OD matrix estimation using multisource data". In: *Journal of Advanced Transportation* 2020.

Hussain, Etikaf, Ashish Bhaskar, and Edward Chung (2021). "Transit OD matrix estimation using smartcard data: Recent developments and future research challenges". In: *Transportation Research Part C: Emerging Technologies* 125, p. 103044.

Ibarra-Rojas, Omar J et al. (2015). "Planning, operation, and control of bus transport systems: A literature review". In: *Transportation Research Part B: Methodological* 77, pp. 38–75.

Lee, Sang Gu and Mark Hickman (2014). "Trip purpose inference using automated fare collection data". In: *Public Transport* 6.1-2, pp. 1–20.

Li, Baibing (2009). “Markov models for Bayesian analysis about transit route origin–destination matrices”. In: *Transportation Research Part B: Methodological* 43.3, pp. 301–310.

Li, Tian et al. (2018). “Smart card data mining of public transport destination: A literature review”. In: *Information* 9.1, p. 18.

Lu, Kai et al. (2020). “A Review of Big Data Applications in Urban Transit Systems”. In: *IEEE Transactions on Intelligent Transportation Systems*.

Ma, Xiao-lei et al. (2012). “Transit smart card data mining for passenger origin information extraction”. In: *Journal of Zhejiang University Science C* 13.10, pp. 750–760.

Ma, Xiaolei et al. (2013). “Mining smart card data for transit riders’ travel patterns”. In: *Transportation Research Part C: Emerging Technologies* 36, pp. 1–12.

Matias, Luis et al. (2010). “Validation of both number and coverage of bus Schedules using AVL data”. In: *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 131–136.

Min, Meekyung (2018). “Classification of Seoul Metro Stations Based on Boarding/Alighting Patterns Using Machine Learning Clustering”. In: *The Journal of The Institute of Internet, Broadcasting and Communication* 18.4, pp. 13–18.

Mohamed, K et al. (2016). “Clustering smart card data for urban mobility analysis”. In: *IEEE Transactions on intelligent transportation systems* 18.3, pp. 712–728.

Munizaga, Marcela A and Carolina Palma (2012). “Estimation of a disaggregate multimodal public transport Origin–Destination matrix from passive smartcard data from Santiago, Chile”. In: *Transportation Research Part C: Emerging Technologies* 24, pp. 9–18.

Nassir, Neema, Mark Hickman, and Zhen-Liang Ma (2015). “Activity detection and transfer identification for public transit fare card data”. In: *Transportation* 42.4, pp. 683–705.

Özgün, Kamer et al. (2020). “Analysis of Public Transportation for Efficiency”. In: *Artificial Intelligence and Applied Mathematics in Engineering Problems*. Ed. by D. Jude Hemanth and Utku Kose. Cham: Springer International Publishing, xxx–yyy. ISBN: 978-3-030-36178-5.

Pelletier, Marie-Pier, Martin Trépanier, and Catherine Morency (2011). “Smart card data use in public transit: A literature review”. In: *Transportation Research Part C: Emerging Technologies* 19.4, pp. 557–568.

Redman, Lauren et al. (2013). “Quality attributes of public transport that attract car users: A research review”. In: *Transport Policy* 25.C, pp. 119–127. URL: `https://doi.org/10.1016/j.tranpol.2012.11.005`.

Stewart, Colin et al. (Jan. 2016). “Perspectives on Transit: Potential Benefits of Visualizing Transit Data”. In: *Transportation Research Record Journal of the Transportation Research Board* 2544. URL: `https://doi.org/10.3141/2544-11`.

Tekin, Sezgin et al. (2018). “TRIP OPTIMIZATION FOR PUBLIC TRANSPORTATION SYSTEMS WITH LINEAR GOAL PROGRAMMING (LGP) METHOD.” In: *Sigma: Journal of Engineering & Natural Sciences/Mühendislik ve Fen Bilimleri Dergisi* 36.4.

Trépanier, Martin, Nicolas Tranchant, and Robert Chapleau (2007). “Individual trip destination estimation in a transit smart card automated fare collection system”. In: *Journal of Intelligent Transportation Systems* 11.1, pp. 1–14.

Tupper, Laura L, David S Matteson, C Lindsay Anderson, et al. (2018). “Band depth clustering for nonstationary time series and wind speed behavior”. In: *Technometrics* 60.2, pp. 245–254.

Tupper, Laura L, David S Matteson, and John C Handley (2016). “Mixed data and classification of transit stops”. In: *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 2225–2232.

Van Oort, Niels and Oded Cats (2015). “Improving public transport decision making, planning and operations by using big data: Cases from Sweden and the Netherlands”. In: *2015 IEEE 18th international conference on intelligent transportation systems*. IEEE, pp. 19–24.

Vo, Van, Jiawei Luo, and Bay Vo (2016). “Time series trend analysis based on k-means and support vector machine”. In: *Computing and Informatics* 35.1, pp. 111–127.

Wang, Wei, John P Attanucci, and Nigel HM Wilson (2011). “Bus passenger origin-destination estimation and related analyses using automated data collection systems”. In.

Wang, Yinhai and Ziqiang Zeng (2018). *Data-Driven Solutions to Transportation Problems*. Elsevier.

Wang, Yuandong et al. (2019). "Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1227–1235.

Welch, Timothy F. and Alyas Widita (2019). "Big data in public transportation: a review of sources and methods". In: *Transport Reviews* 39.6, pp. 795–818. URL: `https://doi.org/10.1080/01441647.2019.1616849`.

Więcek, Paweł et al. (2019). "Framework for onboard bus comfort level predictions using the markov chain concept". In: *Symmetry* 11.6, p. 755.

Yang, Hao and Hesham Rakha (2019). "A novel approach for estimation of dynamic from static origin–destination matrices". In: *Transportation Letters* 11.4, pp. 219–228.

Yang, Yuedi et al. (2020). "Dynamic origin-destination matrix estimation based on urban Rail transit AFC data: deep optimization framework with forward passing and backpropagation techniques". In: *Journal of Advanced Transportation* 2020.

Zhu, L. et al. (2019). "Big Data Analytics in Intelligent Transportation Systems: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems* 20.1, pp. 383–398.

CURRICULUM VITAE

Barış Doruk Başaran

doruk07@gmail.com

EDUCATION DETAILS

Master of Science 2019-2022	Akdeniz University Institute of Natural and Applied Sciences, Department of Computer Engineering, Antalya
Bachelor of Science 2012-2017	Akdeniz University Faculty of Engineering, Civil Engineering, Antalya

WORK EXPERIENCE

Research Assistant 2022-Present	Akdeniz University Faculty of Engineering, Computer Engineering, Antalya

PUBLICATIONS

Articles published in international peer-reviewed journals

1- Özgün, K., Günay, M., Başaran, B. D., Bulut, B., Yürüten, E., Baysan, F., & Kalemsiz, M. (2020, April). Analysis of public transportation for efficiency. In The International Conference on Artificial Intelligence and Applied Mathematics in Engineering (pp. 680-695). Springer, Cham. DOI:
10.1007/978-3-030-79357-9_63

2- Özgün, K., Günay, M., Başaran, B. D., Bulut, & Ledet J. L. (2022, May). Estimation of Alighting Counts for Public Transportation Vehicle Occupancy Levels Using Reverse Direction Boarding. SSRN. DOI:
10.2139/ssrn.4113026

Refereed Congress Publications in Proceedings

1- Özgün, K., Günay, M., & Başaran, B. D. (2021, October). Determination of Peak Times in Public Transportation. In 2021 Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1-6). IEEE. doi:
10.1109/ASYU52992.2021.9599009.