Titre: Title:	Comparing Clustering Methods in Recognition of Temporal Travel Pattern of Public Transport
Auteur: Author:	Zohreh Vaezi
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Vaezi, Z. (2021). Comparing Clustering Methods in Recognition of Temporal Travel Pattern of Public Transport [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/9145/

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Comparing Clustering Methods in Recognition of Temporal Travel Pattern of Public Transport

ZOHREH VAEZI

Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie industriel

Août 2021

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitule :

Comparing Clustering Methods in Recognition of Temporal Travel Pattern of Public Transport

présenté par **Zohreh VAEZI**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Bruno AGARD, président

Martin TRÉPANIER, membre et directeur de recherche

Geneviève BOISJOLY, membre

DEDICATION

To my love,

To my parents,

To myself…

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to the persons/teams/companies listed below, without whom I would not have been able to accomplish this research and even start this journey!

My husband for his patience and unconditional love and supports, my wonderful parents for their selfless sacrifices and supports, and my amazing close friends who have become my family here. Thank you all for always being willing to listen to my stories, to celebrate with me in my successes, and to comfort me in my disappointing moments, especially in this very intense and scary situation of a pandemic caused by COVID-19.

My supervisor, professor Martin Trépanier, for all of his trust, understanding, consistent support, and insightful guidance over the previous two years. He was always available for discussion and never stopped inspiring and motivating to keep going, even during the tough situation of the pandemic!

Professor Bruno Agard for allowing me to join his dynamic team and benefit from his generously provided great workshop materials and resources in data mining and programming.

The team of transportation consisting of the professors and students, for providing a great and helpful platform for all members to communicate while helping students to strengthen the skills of presentation and enrich their research contents by holding regular online meetings enabling us to participate actively.

The Réseau de transport de la Capitale (RTC) who provided the data for this study. The Natural Sciences and Engineering Research Council of Canada (NSERC), the group of THALES, the CORTEX fund, PROMPT who funded this research.

The CIRRELT group and the department of industrial engineering at Polytechnique university of Montreal.

Finally, Professor Daniel Imbeau, who accepted my request at the first step, and gave me the opportunity to participate in his research project. Even though we could not reach an agreement on working conditions, he helped me to believe in my capabilities to start a new field of research which led to so many interesting experiences down my path.

RÉSUMÉ

Les organismes de transport en commun doivent étudier les habitudes de déplacement afin d'élaborer des stratégies et des plans plus conformes aux habitudes d'utilisation de leur réseau. De nombreuses études ont été réalisées sur le comportement des passagers pour aider les autorités de transport à mieux comprendre leurs services. En facilitant la collecte des données, les systèmes de paiement par cartes à puce a rendu possible l'exploration de ces précieuses données sur les déplacements. Pour reconnaître les comportements de déplacement, des méthodes de clustering sont utilisées. En regroupant les passagers ayant le comportement le plus similaire dans un même cluster, nous pouvons alors adapter les stratégies de transport en fonction de ces groupes plutôt que les appliquer à tous les usagers . Les données des cartes à puce ayant des caractéristiques de séries temporelles, le développement de la méthode la plus appropriée pour traiter ces séquences permettra d'obtenir un processus de segmentation plus précis et pertinent.

Le choix d'une mesure de distance appropriée, ainsi que de la méthode elle-même, est crucial dans les algorithmes de clustering de séries temporelles. Dans les études précédentes sur les données des cartes à puce, les distances euclidiennes et Manhattan sont le plus souvent utilisées avec les méthodes de clustering. Cependant, toutes deux ignorent les caractéristiques des séquences et les comparent dans leurs calculs comme des données statiques, sans tenir compte de leur ordre ou de leurs corrélations. Certains auteurs ont tenté de résoudre ce problème dans leurs recherches. En proposant une méthode de projection pour transférer les données de séries temporelles vers des espaces tridimensionnels et en appliquant ensuite des techniques de clustering (Ghaemi et al., 2017) ou en choisissant la distance de corrélation croisée (CCD) comme mesure de distance plus appropriée pour la comparaison de séries temporelles (He et al., 2018). Cependant, comme ils ont utilisé le clustering hiérarchique avec CCD, ils ont été obligés d'utiliser une stratégie d'échantillonnage en raison de la limitation du clustering hiérarchique avec de grands ensembles de données. La mesure de distance *dynamic time warping* (DTW) est une autre distance appropriée pour la comparaison de séries temporelles, mais elle souffre d'une complexité temporelle.

L'objectif de cette étude est de combler cette lacune. Pour ce faire, un nouveau clustering k-shape avec une mesure de distance basée sur la forme (SBD) est testé et appliqué pour la première fois aux données des cartes à puce dans les transports publics. Cette méthode a été proposée par

Paparrizos and Gravano (2017) pour le regroupement de séries temporelles. Elle est précise, efficace et rapide. Nous développons un cadre de comparaison entre les résultats de cette nouvelle méthode, le DTW et les mesures de Distances Euclidiennes (ED) sur le même ensemble de données afin d'explorer leurs avantages et leurs inconvénients. Ce faisant, nous utilisons le regroupement k-means ainsi que les distances DTW et ED pour avoir une comparaison significative entre les performances des mesures de distance. Nous appelons nos trois méthodes DTW, SBD et ED.

Notre cadre de comparaison comporte trois critères, mais avant cela, nous avons utilisé le résultat de DTW comme vérité de base pour construire une comparaison plus structurée. Tout d'abord, DTW est comparé à SBD et ED sur la base de la distance moyenne entre les centroïdes des clusters. Celui dont la distance est la plus courte est donc considéré comme le plus compatible avec DTW. Deuxièmement, la comparaison est basée sur deux indices externes de validation des clusters : l'indice Rand ajusté (ARI) et l'indice de variation de l'information (VI). Plus l'ARI est élevé, plus les deux approches sont en accord. Un VI plus faible, en revanche, indique que les deux techniques sont plus identiques. Enfin, en plus des mesures statistiques, nous comparons les approches en fonction des modèles qu'elles ont identifiés et de la distribution de leurs clusters sur chaque jour, ainsi que de la distribution des produits tarifaires.

En outre, comme le type de vecteurs, ainsi que la méthode, ont un impact significatif sur les résultats finaux, nous utilisons trois types de profils différents pour comparer les performances des méthodes. Les données d'entrée de nos trois approches sont des vecteurs carte-jour (utilisateur-jour), arrêt-jour et itinéraire-jour, et nous pouvons voir si la comparaison est restée la même lorsque les vecteurs ont changé. Ces vecteurs sont basés sur le temps d'embarquement, c'est-à-dire le moment où les passagers effectuent une transaction lorsqu'ils utilisent les transports publics au quotidien. Les données de cette étude proviennent du Réseau de transport de la Capitale (RTC), une agence de transport qui offre des services de transport en commun à Québec, et ont été recueillies sur le réseau d'autobus pendant un mois, en février 2019.

Les résultats de cette recherche contribuent non seulement à la littérature croissante sur les données des cartes à puce, mais confirment également que la distance euclidienne, malgré sa popularité, ne fonctionne pas bien dans la reconnaissance de modèles bien définis lorsqu'il s'agit de données de séries temporelles. En outre, bien que le clustering k-means avec la distance DTW soit une approche appropriée pour la segmentation des séries temporelles, il souffre de sa complexité

temporelle. Le clustering K-shape avec SBD est une méthode puissante de reconnaissance des formes, qui a été testée pour la première fois pour le clustering de données de cartes à puce. Cependant, elle ne prend pas en compte le décalage temporel, ce qui pourrait avoir pour conséquence de mettre des séries temporelles de même forme dans le même groupe alors qu'elles pourraient avoir un décalage temporel. Il a cependant produit des résultats compétitifs dans la création de clusters de routes lorsque le décalage temporel était moindre.

Dans l'ensemble, cette constatation démontre qu'il n'existe pas de solution unique au problème du regroupement des séries chronologiques. Chaque méthode présente des avantages et des inconvénients qui doivent être examinés en fonction du type et du volume de données, du type de distorsions imposées aux données, de l'objectif de l'étude et de la durée d'application de la méthode. SBD est une nouvelle méthode de regroupement de séries temporelles qui a été introduite dans cette thèse et qui peut être utilisée pour une variété d'objectifs dans le domaine du transport, comme l'étude des fluctuations. Elle peut également être réglée pour être contrainte au décalage temporel ou même combinée avec d'autres méthodes.

ABSTRACT

Transit agencies need to investigate travel patterns in order to develop strategies and plans that are more in line with usage patterns. There has been a lot of study done on passenger behaviour to help transportation authorities gain a better understanding of their services. By facilitating data collection, smart card system has made these studies more possible to explore valuable detailed travel data. For recognizing travel behaviour, clustering methods are used. By grouping passengers with the most similar behaviour in the same cluster, we can then adapt the transport strategies based on these groups rather than a large number of individuals. Since smart card data has the characteristics of time-series, developing the most suitable method to handle these sequences will results in more accurate segmentation process.

Choosing a proper distance measure, as well as the method itself, is crucial in time-series clustering algorithms. In previous studies of smart card data, Euclidean and Manhattan distances are most often used with clustering methods. However, both of them ignore the characteristics of sequences and compare them in their calculations as static data without considering their order or correlations. Some authors have tried to address this problem in their research. By proposing a projection method to transfer time-series data to three dimensional spaces and then applying clustering techniques (Ghaemi et al., 2017) or by choosing Cross Correlation Distance (CCD) as a more suitable distance measure for time-series comparison (He et al., 2018). However, since they used hierarchical clustering with CCD, they were forced to plan a sampling strategy due to the limitation of hierarchical clustering with large dataset. DTW distance measure is another suitable distance for time-series comparison, but it suffers from time complexity.

The purpose of this investigation is to address this gap. To do this, a novel k-shape clustering with Shape-Based Distance measure (SBD) is tested and applied to smart card data in public transit for the first time. This method has been proposed by Paparrizos et al. (2017) for time-series clustering, which is accurate, efficient, and very fast with large dataset. We develop a comparison framework among the results of this novel method, DTW, and Euclidean distance (ED) on the same dataset in order to explore their advantages and drawbacks. In doing so, since k-shape clustering is based on k-means principles, we used k-means clustering along with DTW as a suitable distance metric for time-series, and ED as a most used distance in the literature, to have a meaningful comparison

between distance measures' performance. For simplicity, we call our three methods as DTW, SBD, and ED.

Our comparison framework has three criteria, and in order to build a more structured comparison we used DTW result as the ground truth. First, DTW is compared with SBD and ED based on the average distance between cluster centroids. The one with the shortest distance was therefore considered the most compatible with DTW. Secondly, the comparison is based on two cluster validations external indices: Adjusted Rand Index (ARI) and Variation of Information (VI) index. The higher the ARI, the closer the two approaches agree. Less VI, on the other hand, indicates that two techniques are more identical. Finally, in addition to statistical measurements, we compared the three approaches based on the usage patterns of their resulted clusters.

Furthermore, because the type of vectors, as well as the method, has a significant impact on the final outcomes, we employed three different types of profiles with different time-shifting patterns to compare the performance of the methods. The input data for our three approaches were card-day (user-day), stop-day, and route-day vectors, and we can see whether the comparison remained the same when the vectors changed. These vectors are based on the boarding time, which is when passengers make a transaction when using public transportation on a daily basis. The data for this study came from the Réseau de transport de la Capitale (RTC), a transportation agency that offers transit services in Québec City and was gathered from the bus network over one month in February 2019.

Results of this research not only contribute to the growing literature on smart card data, but also confirm that ED in spite of its popularity does not work well in recognition of well-defined patterns when it comes to time-series data. Besides, although k-means clustering with DTW distance is a proper approach for time-series segmentation, it suffers from the time complexity. K-shape clustering with SBD is a powerful method in pattern recognition, which was tested for the first time for smart card data clustering, however, it does not take into account time shifting which could result in putting time-series of the same shape in the same group while they might have a time difference. It however, produced competitive results in the creation of route clusters when the shifting in time was less. On the other hand, the application time of this method was faster than DTW impressively.

Overall, this finding demonstrates that there is no one-size-fits-all solution to the time-series clustering problem. Each method has advantages and disadvantages that should be examined based on the type and the volume of data, the type of distortions imposed to the data, the study goal, and the length of time used by the method application. SBD is a novel time-series clustering method which was introduced in this thesis that might be used for a variety of objectives in the transportation area, such as investigating the travel patterns based on fluctuations. It can also be tuned to be constrained for time-shifting or even combined with other methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

RTC	Réseau de transport de la Capitale
ED	Euclidean Distance
MD	Manhattan Distance
DTW	Dynamic Time Warping
cDTW	Constrained Dynamic Time Warping
CCD	Cross-Correlation Distance
SBD	Shape-Based Distance
ISFCS	Integrated Smart Card Fare Collection System
PCA	Principal Component Analysis
CVIs	Cluster Validation Indices
DB	Davies-Bouldin
DB*	Davies-Bouldin (modified)
Sil	Silhouette
PAM	Partition Around Medoids
DBA	DTW Barycenter Averaging
SMD	Simple Matching Distance

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Problem Statement

Nowadays, the opportunity of gathering a huge amount of data using multiple sources, such as internet searches or smart cards (Jain, 2010), offers a reliable and easier way to collect data without human intervention or effort. The growing use of smart cards in several fields enables studies from various viewpoints to examine this vast amount of data in order to improve the overall system's performance (Kim et al., 2017).

Although public transportation agencies initially adopted the Integrated Smart Card Fare Collection System (ISFCS) as an automated payment system, it is now regarded as the major source for collecting valuable data generated when passengers use their transportation cards (Pelletier et al., 2011). In transportation research, understanding passenger mobility patterns from this data allows to segment a population of people by the same characteristics in the same groups that we might find valuable to action upon. In other words, it can help with day-to-day operations and long-term planning for the transportation system, such as route design, urban planning, location-based services, network growth, marketing, and so on (Pelletier et al., 2011; Zhao et al., 2008). Regarding the fact that, this smart card data contains detailed information such as the time of transaction, its location and direction of the route or even most of the time the type of card etc., we can divide this information and its subsequent analysis into three main categories: temporal, spatial and spatiotemporal types (Ghaemi et al., 2017). In our study, however, the former is our mission, which means that we want to propose a method to characterise the temporal pattern of passenger behaviour.

In temporal side of analysis, Agard et al. (2006) showed the use of data mining techniques to identify temporal behaviour patterns of passengers. They used clustering method to extract similar groups so that there would be the most similarity within the members of the same group and the most dissimilarity between members of distinct groups. Hence, the adaptation of services based on the needs of these identified groups with the most similar behaviour will be more reasonable and applicable instead of considering each individual needs. In temporal clustering, one of the challenges is dealing with complicated high dimensional time-series data (Nantes et al., 2016). To address this problem, Kim et al. (2017) proposed Principal Component Analysis (PCA) for

reducing the number of variables, and they used Euclidean distance measure with k-means clustering. However, another challenge is the choice of proper distance measure for comparing time-series and despite many metrics are defined for this comparison in the literature such as Euclidean, Manhattan, Hamming, etc. none of them can consider the dynamic characteristics of temporal vectors. In this regards Ghaemi et al. (2017) in an attempt to deal with this issue, proposed a novel projection of time-series into three-dimensional clocklike space which could retain the temporal distance. He et al. (2018) also proposed to use Cross-correlation distance (CCD) and compared it with Dynamic Time Warping (DTW) as two proper distances for time-series comparison. They then, applied hierarchical clustering using a sampling procedure trying to cover its limitation of working with large dataset. However, the instability and the challenges of sampling procedure were the limitation of their study. Finally, they discussed that CCD outperformed DTW with their dataset.

Even though there are lots of research that have been carried out for smart card pattern extraction in the field of transportation and they proposed several solutions to deal with the challenges working with high dimensional time-series data, it seems there is still a need to select a proper distance measure along with a clustering method to tackle these challenges in a better way. Paparrizos et al. (2017), proposed a novel algorithm for time-series clustering using k-shape clustering with Shape-Based Distance (SBD) measure which is a normalised version of CCD. After applying their method on several dataset, they claimed that their method is a parameter-free, fast, accurate, and efficient compared with other existing time-series clustering algorithms. Beyond doubt, there is not a best distance measure, or a best clustering method resulted in the best well-defined clusters in all cases, but nevertheless there is always a wise choice with compromising the advantages and drawbacks considering the conditions. Therefore, we decided to test the novel SBD measure with k-shape clustering on our smart card data as a first attempt of applying this method in the field of transportation for pattern recognition.

According to the literature, combining DTW distance measure with k-means clustering is one of the proper choices for time-series clustering problems. On the other hand, k-means clustering with Euclidean distance (ED) have been claimed that is not a proper choice for time-series comparison despite its popularity. Thus, in our study we perform three clustering methods with three different distance measures, and we propose a comparison framework to compare their outcomes revealing

their pros and cons while applying to the same dataset. In order to do so, we first consider DTW as the ground truth we then compare it with SBD and ED from three perspectives. Firstly, according to the average distance measure between clusters centroids, secondly based on two cluster validation external metrics and finally based on their clusters pattern.

Most research works have focused on user segmentation such as all studies discussed so far. There is less works have been undertaken aiming station segmentation (Gan et al., 2018; Reades et al., 2016), or routes grouping. In this study, we analyse temporal pattern of travel not only for users, but also for stations and routes aiming to see differences in performances of three methods in different scenarios with three types of vectors.

1.2 Thesis Objectives

Having in mind the main purpose of characterizing and understanding temporal travel behaviour in public transit, in particular analysing smart card data over one month in the bus network, we are also pursuing the following sub-objectives:

- Test and adapt a new approach of k-shape clustering with SBD for time-series analysis in public transit in order to obtain more accuracy and less time complexity.
- Perform k-means clustering with DTW distance metric and k-means clustering with ED.
- Compare the performance of k-shape with SBD and k-means clustering with ED with k-means clustering DTW distance measure.
- Reveal the advantages and disadvantages of the methods in time-series clustering.
- Explore three different types of objects users, stops and routes.

1.3 Thesis Structure

Following our objectives, this thesis is organized in 7 chapters. Chapter 1, holds the introduction, comprising existing problems in analysing travel patterns in public transit while a bit referring to some previous studies besides our solutions to address these problems then we list our objectives.

Chapter 2 consists of discussing smart card data, clustering methods specifically k-means algorithm, time-series characteristics, distance measures, averaging technique, cluster validation

methods are presented. The previous studies in this field, and their contributions and limitations are discussed, afterwards. The details of the procedure of our three methods also present at the end of this chapter.

Chapter 3 provides the steps we follow in our proposed algorithm and how the dataset is prepared for the further analysis and comparison. Chapter 4, 5 and 6 are devoted to the analysis of vectors of users, stops and routes, separately. In these chapters, the resulted groups from applying our three clustering methods characterised and compared with each other.

Finally, chapter 7 concludes this study by summarizing the main results obtained and highlighting the contributions made by performing our suggested framework with respect to three different objects of analysis. The limitations of our research are also noted, and future perspectives are then provided to enlighten a possible follow-up and take advantages of using k-shape clustering method in public transit area with different objectives.

CHAPTER 2 LITERATURE REVIEW

This chapter provides a review of the relevant literature in order to establish a framework and assess prior studies in our research area, allowing for a better understanding of the challenges that exist. This state-of-the-art will assist us in identifying gaps. It also reinforces the foundation and justification for the methodological approach we propose in this thesis.

2.1 Smart Card Data in Public Transit

The data collected using the smart card in public transit has been the target of many studies in recent years. The main questions arise here are:

- How is this data collected using smart card? How is it stored? And what are its characteristics?
- What are the benefits of having this data? How can it be used to improve the system of transport?

To answer these questions, we first present the procedure of data collection and reservation by the smart card system showing in Figure 2-1, for more detail explanation please refer to Pelletier et al. (2011).

In addition to the primary goal of using smart cards as a fare collection system, Pelletier et al. (2011) categorised the applications of using this data into three groups of operational, tactical, and strategic levels in public transportation management. For instance, operational research has been conducted with the objective of improving the daily performance of the system. In addition, estimation of accessibility (Arbex & Cunha, 2020; Cavallaro & Dianin, 2020) crowding valuation (Yap et al., 2018) can be considered in this category. In tactical studies, according to user needs, public transport services will be scheduled and customised. In this regard, Seo et al. (2020) analyzed overlapping origin-destination pairs between bus stations resulting to help enhance the efficiency of transit operation, eventually. Demand estimation and forecasting by identifying public transit corridors (Zhang et al., 2018) costumer behaviour analysis (Agard et al., 2006) are also part of strategic studies improving long-term planning.

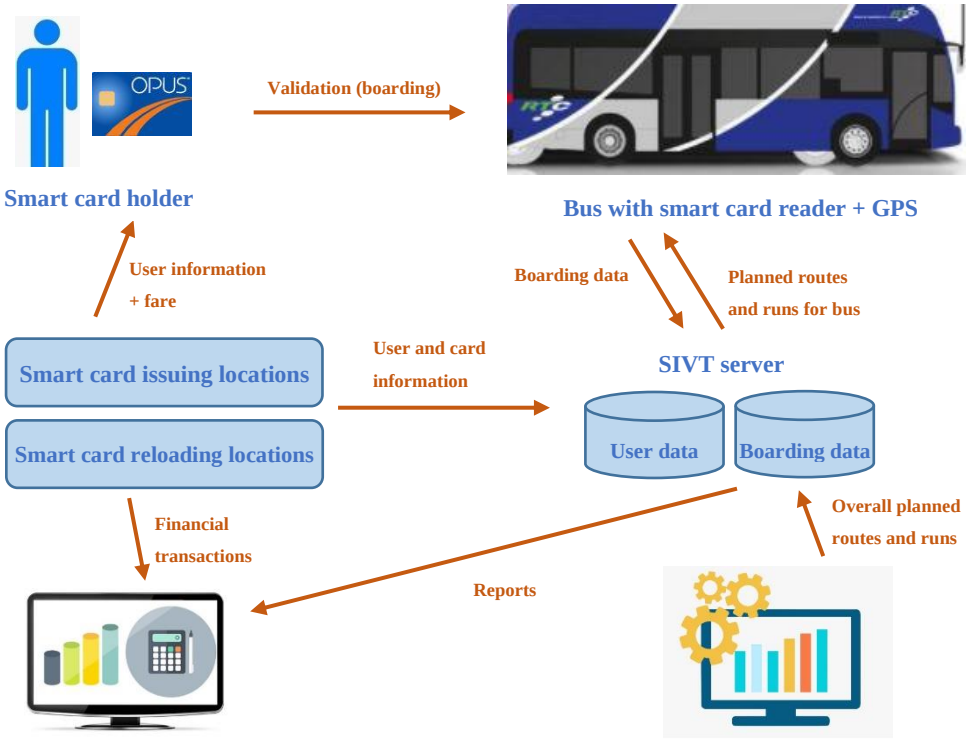


Figure 2-1: RTC smart card information system (Pelletier et al., 2011)

Among all these studies, investigating user behaviour in public transit has been great of interest and there are many authors that have suggested different tools, methods, and algorithms to help better understanding of this main issue.

Regarding the type of information that be obtained by smart card data, Ghaemi et al. (2017) divided the research have investigated user behaviour into three categories: (1) spatial pattern in which the focus is on the location of transactions, such as bus stop information, (2) in temporal patterns, analysis the time of transactions is taken into account, (3) spatiotemporal patten as can be gotten from its name, it could be considered as a hybrid of the previous two steps or as a stand-alone new way to dealing with spatiotemporal behavioural patterns.

Because the focus of this thesis is on analysing the temporal pattern of user behaviour, which has the same characteristics as time-series data, the next section includes a brief definition of time-series data before digging deeper into the clustering algorithm to deal with this type of data for pattern extraction.

2.2 Time-Series Clustering Algorithms

Clustering is the most popular data mining method in analysing behaviour in such a way that by comparing data points and creating some groups/clusters of them while there is the most similarity within data points in the same group and the most dissimilarity with the members of other groups (Han & Kamber, 2006). In the literature, various clustering algorithms have been developed for static data and there is not a direct method for times-series type of data. Thus, for time-series clustering we have to either convert it to static type and use the existing methods directly or the method has to be modified to be able to deal with the temporal type of data. Regarding that, Warren Liao (2005) divided the clustering methods into two major categories. (1) Raw-based methods; directly work with the raw time-series data and they are based on a distance measure for calculating (dis) similarity between time-series. (2) Feature/model-based methods; first convert time-series data into the selected features for reducing dimension or the number of model parameters and then apply clustering. Feature/model-based techniques are more flexible to capture complex patterns but on the other hand, raw-based (or distance-based) approaches are simpler and easier to adapt (Ghaemi et al., 2017). In this study our focus is on the distance-based clustering methods.

Agglomerative hierarchical, spectral, density-based, and partitional are the four most popular distance-based clustering methods. Partitional clustering includes the two main heuristic well-known methods, k-means and k-medoids. Since each method expresses homogeneity and separation of clusters differently and also they have different computational cost from another, the choice of them is a difficult task due to their different effect on the accuracy and efficiency of clustering (Paparrizos, 2018). On the other hand, the quality of clustering method and resulted clusters not only depends on this choice, but also selecting a compatible distance measure for (dis) similarity comparison is another challenging step. Besides, when it comes to temporal data because of the sequential characteristics, there are also some distortions and invariances which either need to be satisfied with the choice of proper distance measure or to be removed before applying clustering (Batista et al., 2013).

Among hierarchical, spectral, and k-medoids methods, k-means is more efficient and can scale linearly with the size of the datasets. Moreover, because it is a simple and efficient algorithm with a wide range of applicability, it is known as one of the most influential data mining algorithms of

all time (Paparrizos et al., 2017). To understand how k-means clustering works with temporal type of data we first define time-series and then we organize some sub-sections as follows: first we briefly describe the procedure of k-means algorithm, afterwards, since along with the choice of clustering method itself, a suitable distance measure, and a proper averaging method depend on the specific characteristics of time-series dataset we work on, we discuss these characteristics in separate sub-section called time-series invariances. The final section belongs to the techniques are used to evaluate the performance of the clustering method.

A time-series is a set of observations indexed in time order. To be specific, it is a sequence in which every element is resulted of recoding a measurement varying by the time. There are two types of time-series, univariate is a series with a single time-dependent variable whereas several measurements varying over time make a multivariate time-series (Paparrizos, 2018). In our thesis, we work with univariate time-series, hence, we refer it simply as time-series from now on.

2.2.1 K-means Clustering for Time-Series

K-means is a partitional clustering method based on an iterative refinement procedure. For time-series clustering, k-means starts with k artificial sequences as centres (or centroids), assigns sequences to the closest centroids, and produces k groups, then calculates new centres for those groups, and this process is repeated until no changes in centroid selection are feasible. The algorithm proceeds as follows:

1. Select an initial k clusters centroid.
2. Assign each object to its closest cluster centroid which generates a new partition.
3. Compute the centroid of the new partition.
4. Repeat steps 2, and 3 until convergence is obtained.

2.2.2 Time-Series Invariances

As pointed out before, when we compare time-series data we need to select a distance measure to be able to satisfy invariances resulting in distortion elimination. In this section we review the most common invariances. Depending on the case, one or more of these invariances should be addressed to gain the better time-series clustering results (Batista et al., 2013; Paparrizos et al., 2017).

- Scaling and translation invariances: In many cases, regardless of differences in amplitude (scaling) and offset (translation) of two time-series we still need to consider them similar in comparison. In simple mathematical way, transforming a sequence $\vec{x}$ to $\vec{x}' = a\vec{x} + b$, when a and b are constant, should keep the similarity of $\vec{x}$ to others.
- Shift invariance: When two sequences are similar but differ in phases (global alignment) or when there are regions of the sequences that are aligned and others are not (local alignment), in some cases despite these differences considering them similar is still necessary.
- Uniform scaling invariance: In some cases, due to differences in the length of sequences the matches will be poor. So, we try to stretch the shorter one or shrink the longer to have the better comparison.
- Occlusion invariance: When there is missing in some sequences and we still need to consider them similar, and we ignore the missing parts.
- Complexity invariance: Sometimes sequences have similar shape but different complexities, based on the application we consider them low or high similar.

2.2.3 Time-Series Distance Measures

As we mentioned before, when we aim to compare two time-series, we have to calculate the (dis) similarity between them. For doing so, converting data to vectors, and then calculating the distance between data points in vector space can create a distance matrix. Based on the study of Rakthanmanon et al. (2011), the selection of distance measure is so important in capturing inherent distortions of sequences. In other words, the more proper selection of distance measure, the more satisfied are distortions and the better results are obtained by clustering method. This section contains reviewing the most common and popular distance measures.

Suppose that we have two time-series, $\vec{x} = (x_1, \dots, x_i, \dots, x_n)$ and $\vec{y} = (y_1, \dots, y_j, \dots, y_m)$ where *m* and *n* represent their length.

- Manhattan distance (MD): When $m = n$, using the following formula will give us the dissimilarity between them based on MD (Mori et al., 2016):

$$MD(\vec{x},\vec{y}) = \sum_{i,j=1}^{m} |x_i - y_j| \quad (2\text{-}1)$$

- Euclidean distance (ED): Euclidean is a competitive well-known distance measure which computes the dissimilarity between $\vec{x}$ and $\vec{y}$ ($m = n$), as bellows (Faloutsos et al., 1994):

$$ED(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \quad (2\text{-}2)$$

- Dynamic time warping (DTW): It is a popular and proper adapted distance measure for time-series, and it performs elastic alignments. This distance actually tries to find the optimum warping curve between sequences under certain constraints (Paparrizos, 2018).

  In a case of $m = n$, between all two points of these series, ED or MD is calculated and create a *m*-by-*m* matrix, we call it *M*. Then, a wrapping path, $W = \{w_1, w_2, \dots, w_k\}$, with $k \geq m$, based on the distances in matrix *M* aligns the elements of $\vec{x}$ and $\vec{y}$, such that the minimum distance be chosen (Keogh & Ratanamahatana, 2005):

$$DTW(\vec{x},\vec{y}) = min\sqrt{\sum_{i=1}^{r} w_i} \quad (2\text{-}3)$$

  This path can be obtained by dynamic programming, as bellows:

$$\gamma(i,j) = ED(i,j) + min\begin{cases}\gamma(i-1,j-1)\\ \gamma(i-1,j)\\ \gamma(i,j-1)\end{cases} \quad (2\text{-}4)$$

  In Figure 2-2, we depicted the procedure followed by DTW between two series $\vec{x} = (1,2,1,1,3,1,1,1)$, $\vec{y} = (1,1,2,1,1,3,1,1)$. First for creating matrix *M*, shown in Figure 2-2 (a), the distance between each element of series is calculated by Equation 2-4. For instance, the distance between the fifth element of $\vec{x}$, and the third one of $\vec{y}$ computed as bellows:

  $D(3,2) = ED(3,2) + \min\,(D(1,1), D(1,2), D(3,1))$

  $D(3,2) = 1 + \min(1,2,3)$

  $D(3,2) = 2$

Second, the warping path is mapped according to Equation 2-3, Figure 2-2 (b). In some cases, there are many possible warping paths which makes the searching cumbersome and expensive in time and memory consuming. Hence, for optimising DTW's performance there are some constrains to limit the area of matrix *M* for mapping wrapping path which is called *warping window*. This method has been called constrained dynamic time wrapping (cDTW), and it is more efficient than DTW (Paparrizos, 2018).

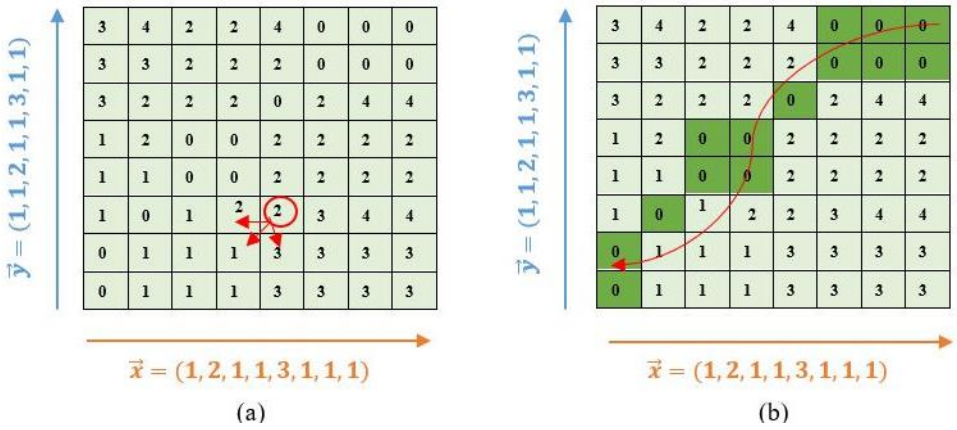


Figure 2-2: Similarity computation by DTW between two series $(\vec{x}\,, \vec{y})$: (a) matrix M, (b) warping path

There are many types of windows and the most popular one is the Sakoe-Chiba window which is visualised in Figure 2-3.

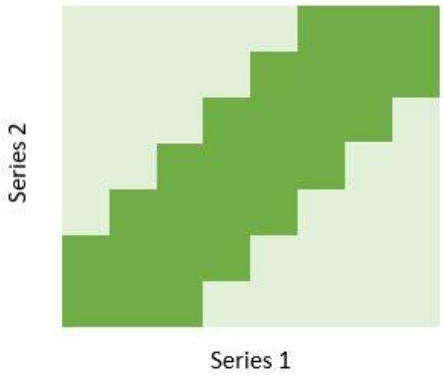


Figure 2-3: Sakoe-Chiba window warping (Giorgino, 2009)

- Cross-correlation distance (CCD): Cross-correlation is another proper and widely used distance measure in comparing time-series data and in contrast to DTW which provides

local alignment as we mentioned before, it compares sequences that differ in phase (global alignment).

To find the similarity between, $\vec{x} = (x_1, \dots, x_m)$ and $\vec{y} = (y_1, \dots, y_m)$, this method shifts one of them to find the maximum cross-correlation with another one. If we call this shift, *s*, and slides $\vec{x}$ over $\vec{y}$ then (Paparrizos et al., 2017):

$$\vec{x}_{(s)} = \begin{cases} (\overbrace{0, \dots, 0}^{|s|}, x_1, x_2, \dots, x_{m-s}), & s \geq 0 \\ (x_{1-s}, \dots, x_{m-1}, x_m, \underbrace{0, \dots, 0}_{|s|}), & s < 0 \end{cases} \qquad 2\text{-}5$$

Considering all possible *s* between [-*m*, *m*], we have the cross-correlation sequence as bellows:

$$CC_w(\vec{x}, \vec{y}) = R_{w-m}(\vec{x}, \vec{y}), \qquad w \in \{1,2, \dots, 2m-1\} \qquad 2\text{-}6$$

Where $R_{w-m}(\vec{x}, \vec{y})$, is as follows:

$$R_k(\vec{x}, \vec{y}) = \begin{cases} \sum_{l=1}^{m-k} x_{l+k}.y_l & k \geq 0 \\ R_{-k}(\vec{x}, \vec{y}) & k < 0 \end{cases} \qquad 2\text{-}7$$

The amount of *w* which makes the $CC_w(\vec{x}, \vec{y})$ maximum will be the objective and based on that the optimal shift is $s = m - w$.

2.2.4 Time-Series Averaging Techniques

Another important part of time-series clustering is the choice of averaging method (prototyping) In this regard, since the goal of clustering is having the most similar series in one cluster, there should be one sequence representing the most characteristics of other sequences in that given cluster. The choice of averaging function is closely related to the choice of distance measure. Additionally, when it comes to partitional clustering, because resulting average series are used as centroids, this choice will be even more critical. There are several strategies for time-series

averaging and the choice of proper one is not easy (Sardá-Espinosa, 2018). In this section we briefly review some of the common ones.

- Mean and median: The arithmetic mean is a common and easiest approach for averaging and mostly combined with Euclidean distance to create a competitive combination for k-mean clustering. However, due to the characteristics of time-series this approach could even perturb convergence of a clustering algorithm and give poor result (Petitjean et al., 2011). In addition to mean, median is also can be used as the averaging method.

- Partition around medoids (PAM): Partition around medoids (PAM) is another popular approach uses an object in a cluster as a medoid whose average distance to all other objects in that cluster is minimal. In some cases, PAM is preferred over mean or median due to its originality in the cluster. In other words, since it is chosen from the data points instead of artificial creation by arithmetic mean, the structure of time-series does not change. In addition, since the data is not altered, precomputing the whole distance matrix once and reusing it on each iteration, and even across different number of clusters and random repetitions would be possible which indicates another advantage of this method (Sardá-Espinosa, 2018).

- DTW barycentre averaging (DBA): This is an iterative global prototyping method which starts with an initial average sequence as a centroid and refines it by minimising the distance between the average sequence and other sequences in the cluster. Precisely, the distance between each element (or coordinate) of the average sequence and all elements of other series in the cluster is computed based on DTW and a mean is computed for each centroid coordinate. It is necessary to repeat this process several times with a new centroid in a way that its elements be closer (under DTW) to the elements it averages. This is iteratively repeated until a certain number of iterations are reached, or until convergence is assumed (Petitjean et al., 2011; Sardá-Espinosa, 2018)

2.2.5 Clustering Validation Techniques

After performing clustering, it is common to see how well it worked in creation of true clusters. There are two types of metrics; Internal and external measures, to assess clustering performance.

On the other hand, in some clustering methods in particular partitional type like k-means and k-medoids, there is a need to specify the number of clusters (or k) when the method is applied. Since the correct choice of k relies on the shape and scale of the distribution of data, in the beginning of the process it is challenging and ambiguous. Overestimating the number of k, would lead to interpretation problems and underestimating might risk generalising our groups. Although considering too few number of k cause worse performance than in overestimation (Rodriguez et al., 2019), as we mentioned both situations will increase the amount of error in the final results. Therefore, detecting the optimal number is a critical choice. There are several methods to address this issue, cluster validation internal metrics as we discuss in the following are among the popular ones.

2.2.5.1 Internal Metrics

Internal indices are based on the intrinsic information lies within the data and tries to measure the quality of partitions formed by the algorithm. Previous studies have declared that there is no best single measure for clustering validation, thus a better way is to use several techniques and compared their results to have a more robust output (Arbelaitz, Gurrutxaga, Muguerza, Pérez, et al., 2013). Among all, we review three well-known internal cluster validation techniques; Davies-Bouldin (DB), modified version of Davies-Bouldin (DB*), and Silhouette (Sil).

Internal validation measures generally reflect (1) cohesion (or intra-cluster distance) which calculate the similarity of a data point to all other data in the same cluster, and (2) separation (or inter-cluster distance) which is the similarity of data point to other members of other clusters. We describe briefly these indices based on the study of Arbelaitz, Gurrutxaga, Muguerza, Pérez, et al. (2013).

- DB (to be minimised): This index is one of the most used cluster validation indices for consistency estimation of the resulted clusters. The lower the DB index value, the better is the resulted clusters. For *k* number of clusters, DB index is obtained by the following equation:

$$DB = \frac{1}{k} \sum_{c_k \in C} max \left\{ \frac{S(c_k)+S(c_l)}{d(\bar{c}_k,\bar{c}_l)} \right\} \qquad 2\text{-}8$$

Where $S(c_k) = \frac{1}{|c_k|}\sum_{x_i \in c_k} d(x_i, \bar{c}_k)$, is the intra-cluster distance of cluster c_k which is the distance of all points of c_k to the centroid of c_k, and $d(\bar{c}_k, \bar{c}_l)$ is the inter-cluster distance which is the distance between centroids of clusters c_k and c_l.

- DB*(to be minimised): This is the modified version of DB.

$$DB^* = \frac{1}{k}\sum_{c_k \in C} \frac{\max\{S(c_k)+S(c_l)\}}{min\{d(\bar{c}_k,\bar{c}_l)\}} \quad \text{2-9}$$

- Sil (to be maximised): This index calculates the cohesion based on the distance between all points in the same cluster and the separation based on the nearest neighbour distance.

  If data has been grouped in *k* clusters, then for data point x_i in cluster C_k, a(x_i) which is the mean distance between x_i and all other data points in the same cluster is calculated as below:

$$a(x_i) = \frac{1}{|C_k|-1}\sum_{x_j \epsilon c_k} d(x_i, x_j)$$

  And b(x_i) which is the mean distance between x_i and all data points in any other cluster C_l, of which x_i is not a member ($C_l \neq C_k$) is calculated by:

$$b(x_i) = \min_{k \neq l}\left\{\frac{1}{|C_l|}\sum_{x_j \epsilon C_l} d(x_i, x_j)\right\}$$

  The cluster with the smallest mean dissimilarity is called the "neighbouring cluster" for x_i as it is the next best fit well cluster. With calculating *a* and *b*, we can obtain Silhouette value by the following equation:

$$Sil = \frac{1}{N}\sum_{C_k \epsilon C}\sum_{x_i \epsilon C_k} \frac{b(x_i)-a(x_i)}{max\{a(x_i),b(x_i)\}} \quad \text{2-10}$$

  The value of Sil index is between -1 and 1, where a high value indicates that the data points are well matched to their own clusters and poorly matched to neighbouring clusters.

2.2.5.2 External Metrics

External measures are useful when we have information about the correct partitions of a dataset as ground truth. We can compare it to our results from applying a clustering method, assuming that the more similar the method's partitions are to the ground truth, the better the method. On the other hand, using these external measures is also common when we want to compare the results of several clustering methods applied to the same dataset. There are several measures, however, in the following part, we review the two most prominent ones Adjusted Rand Index (ARI) (Rabbany & Zaïane, 2015; Santos & Embrechts, 2009), and Variation of Information (VI) (Meilă, 2007).

- ARI (to be maximised): This index is based on counting the pairs of objects that two clustering methods agree/disagree on. This means that no matter what the individual labels are, this index evaluates the set overlap. When it comes to comparison, there are some bias in terms of the number of clusters leading to change the results of external indices, ARI though tends to be indifferent which considered an advantage of this index (Rodriguez et al., 2019).

  Given a set of *n* data, $D = \{d_1, d_2, \dots, d_n\}$, suppose that $V = \{v_1, v_2, \dots, v_C\}$ and $U = \{u_1, u_2, \dots, u_R\}$ represent two different resulted clusters from D such that $U_{j=1}^{C} v_j = D = U_{i=1}^{R} u_i$. The contingency table of these partitions is as follows:

  Table 2-1: Contingency table between partitions U and V

Partitions		**V**				
		v_1	v_2	…	v_C	**Sum**
U	u_1	n_{11}	n_{12}	…	n_{1C}	$n_{1.}$
	u_2	n_{21}	n_{22}	…	n_{2C}	$n_{2.}$
	…	…	…	…	…	…
	u_R	n_{R1}	n_{R2}	…	n_{RC}	$n_{R.}$
	Sum	$n_{R.}$	$n_{R.}$	…	n_{RC}	n

  From this table, we can calculate *a* and *d*, which are the number of pairs that are in the same/different partitions in *U* and *V*. Besides, b and c sum up of those that belong to the same/different partitions according to *U* but are in different same/partitions in *V*.

Table 2-2: Simplified contingency table between U and V

Partition	V	
U	Pair in same group	Pair in different group
Pair in same group	*a*	*b*
Pair in different group	*c*	*d*

When $n_{ij} = |U_i \cap V_j|$, $n_{i.} = \sum_j n_{ij}$, $n_{.j} = \sum_i n_{ij}$, and $\binom{n}{2}$ is the total number of possible combinations of pairs in two partitions *U* and *V* then for the calculation of *a*, *b*, *c*, and *d* we have:

$$a = \sum_{i=1}^{R} \sum_{j=1}^{C} \binom{n_{ij}}{2}$$

$$b = \sum_{i=1}^{R} \binom{n_{i.}}{2} - a$$

$$c = \sum_{j=1}^{C} \binom{n_{.j}}{2} - a$$

$$d = \binom{n}{2} - a - b - c$$

Therefore, ARI is equal to:

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]} \qquad 2\text{-}11$$

- VI (to be minimised): This index is based on entropy. If we call $H(C)$ as the entropy associated with clustering C, then we have:

$$H(C) = -\sum_{k=1}^{K} p(k) \log\, p(k) \qquad 2\text{-}12$$

When $p(k) = \frac{n_k}{n}$ is the probability that a data point being classified in cluster C_k while n_k is the number of points in this cluster and n is the number of total points. Entropy equals to 0, means there is only one cluster and then no uncertainty.

If we call $I(C, C')$ as mutual information between two clustering methods; the information that one clustering has about the other, we will have:

$$I(C,C') = \sum_{k=1}^{K}\sum_{k'=1}^{K'} p(k,k') log \frac{p(k,k')}{p(k)p'(k')} \quad 2\text{-}13$$

When $p(k,k') = \frac{(C_k \cap C'_{k'})}{n}$ is the probability that a point belongs to C_k in clustering C and to $C'_{k'}$ in clustering C'. Having the entropy and mutual information, VI is calculated as following:

$$VI(C,C') = [H(C) - I(C,C')] + [H(C') - I(C,C')] \quad 2\text{-}14$$

The first and the second part of this equation are called conditional entropies. The first one; $H(C|C')$, measures the amount of information about C that we loose, while the second one $H(C'|C)$, measures the information about C' that we have to gain, we are going from clustering C to C', these are called joint entropy. Figure 2-4 illustrates the concept and the relation between information entropies, mutual information, and variation of information more clearly.

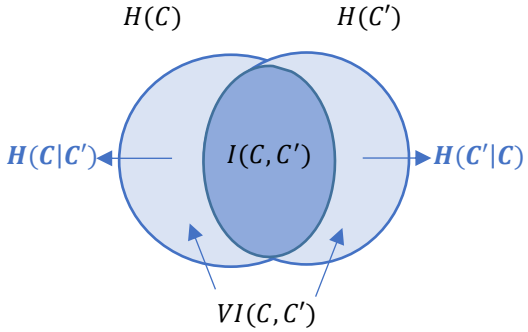


Figure 2-4: Information diagram (Meilă, 2007)

2.3 Time-Series Clustering in Public Transit

There are many research in the field of transportation with several objectives such as alighting location estimation (He & Trépanier, 2015; Trépanier et al., 2007) future trend estimation (Park et al., 2008), trip purpose inferences (Lee & Hickman, 2013).

The most of smart card studies in transportation area are aimed to recognise the travel pattern. Agard et al. (2006) used the clustering technique based on the boarding time of transactions to

identify temporal characteristics of passenger behaviour on weekly basis. Then, they measured changes in cluster membership to explore intrapersonal variability in transit usage as the first research in this area. Ever since, several studies have been carried out to measure the variability and the evolution of cluster composition. Besides measuring the temporal variability by applying clustering on boarding time, Morency et al. (2007) investigated spatial variability of transit users through the frequency of usage of bus stops. Using smart card data from Montréal, Deschaintres et al. (2019) focused on weekly variability in daily travel rate. A week typology is constructed using the K-means clustering technique, and each card is then represented as a succession of week clusters over 12 months. After that, the sequences are utilised to cluster interpersonal variability and measure intrapersonal variability as well. Egu and Bonnel (2020) assessed simultaneously interpersonal and intrapersonal day to day variability of user behaviour. They used hierarchical clustering with simple matching distance (SMD) for interpersonal variability and intrapersonal variability was evaluated with trip-based similarity metric which is the similarity of two days based on number of trips and the time and origin of starting the trip. Viallard et al. (2019) used k-means clustering observed the evolution of users' behaviors by experimental of multi-week travel patterns. Using Euclidean distance, authors has measured the sequential stability of the cluster's membership over the period of use (Moradi & Trépanier, 2018).

As pointed out before, the use of clustering for user segmentation allows passengers to be identified and grouped into clusters with similar behaviour. This data gathered by smart card, has the characteristics of time-series data. Traditional distancing metrics, therefore, are not suitable for this dynamic type of data. Some researchers have tried to address this issue by either transferring data into static one or modifying the clustering methods to be able to handle time-series. Ghaemi et al. (2017) for discovering temporal pattern of public transit users suggested a hierarchical clustering algorithm along with the novel projection to reduce the data space into a three-dimensional clocklike. In another research, authors used cross-correlation distance (CCD) and dynamic time warping (DTW) distance measure as the proper methods for sequence comparison (He et al., 2018). However, since they used hierarchical clustering, due to its limitation for large dataset, they forced to take samples for applying the methods.

2.4 Synthesis

To have a clear picture of previous research done for smart card data analysis identifying the travel behaviour, we present some of the main studies in Table 2-3. As can be seen, in all of these studies, the authors either have tried to transfer time-series data to static one to use traditional methods or have attempted to use a more appropriate distance metric for time-series comparison. However, a sufficient clustering technique that is consistent with sequences has yet to be established.

Table 2-3: Previous studies in smart card analysis in public transit

Study	Target object	Vector	Distance measure	Clustering method	Averaging method	Objective /Contributions
(He et al., 2018)	Card-day	Boarding time (binary vector)	CCD and DTW	Hierarchical	-	-
(Kim et al., 2017)	Stop-day	Boarding and alighting time	-	K-means	-	Investigation of local environment effects on human behaviour
(Agard et al., 2006)	Card-week	-	Euclidean	K-means	-	-
(Ghaemi et al., 2017)	Card-day	Boarding time (binary vector)	SCP, CCD, and ACD (autocorrelation)	Hierarchical	-	Proposed a semi-circle projection (SCP) method
(Deschaintres et al., 2019)	Card-week	7 dispersion indicators (number of trips per day) and one intensity indicator (average number of trips)	Euclidean	K-means	-	-
(Viallard et al., 2019)	Card-week	Number of trips each day of the week	Euclidean	K-means++	-	The experimental method allows the evolution of the centres through time, while the traditional method considers them stationary
(Egu et al., 2020)	Card-day	Bording and alighting time (binary vectors)	Simple Matching Distance (SMD)	Hierarchical	-	Assessing simultaneously interpersonal and intrapersonal variability of user behaviour
(Gan et al., 2018)	Stop-day	Bording and alighting time		K-means	-	This study presents one of the first attempts of exploring the relationship between local LCLU and metro ridership patterns
(Chen et al., 2009)	Stop-day	-	Euclidean	K-means	-	Investigating whether station ridership's diurnal pattern is closely related to the local built environment
(Reades et al., 2016)	Stop-day	Bording time	-	PCA + K-means	PAM	-
(Agard et al., 2013)	Card-day	Bording time (binary vectors)	Euclidean	K-means	-	Dimensionality reduction

CHAPTER 3 METHODOLOGY

In this chapter, we first introduce the dataset, and the transit agency that provided it; Réseau de transport de la Capitale (RTC). We then, provide the framework of our research and the steps we follow to reach our objectives. Afterwards, the methods we have employed be presented in detail, including three objects of analysis; users, stops and routes separately.

3.1 Dataset presentation

In this section we present the data and its provider.

3.1.1 Réseau de transport de la Capitale (RTC)

The data of this study has been provided by Réseau de transport de la Capitale (RTC), a transit authority offering regular public transit services for 575,000 inhabitants in the greater Quebec City area. The RTC has started to use smart card fare collection system since 2010 in its 563-bus network. 97% of Quebec City residents live within 800 meters of a bus stop. The map of RTC bus network is presented in Appendix B.

3.1.2 Dataset Structure

Table 3-1 is an excerpt from raw dataset transaction. As can be seen, each transaction has some properties which we presented with the exact names provided for this study. We then briefly describe them in the following part.

Table 3-1: Excerpts from raw smart card dataset

Id-val	Dateoperat	Codeligneo	Direction	Stop-o	Stop-d	Codeod	Tempstraje	Distanceod	Opus-id	Code-produ
28706	02/01/2019 00:19:54	801	Est	1207	1935	21	15	4703	1878149	TB2-G
30946	02/01/2019 00:03:30	807	Est	1455	2709	11	14	4237	1010684	TLM-E
30947	02/01/2019 00:03:29	807	Est	1455	2709	11	14	4237	1002188	TB2-E
34641	02/01/2019 00:09:26	807	Ouest	1394	1424	21	28	8575	1760187	TLM-G

- Id-val: Indicating validation ID which represents a unique record for each transaction.
- Dateoperat: This shows the date and time of boarding indicating “day/month/year” and “hour:minute:second”. The period runs from 2019/02/01 to 2019/02/28 containing 24 hours of a day.
- Codeligneo: The number of bus lines which are 992 in the RTC's public transport network.
- Direction: Indicating the four directions of east, west, north, south.
- Stop-o/Stop-d: The number of stops, representing the location of boarding/alighting where the passengers embark on/end the trip. In RTC bus network, there were a total of 4538 stops at the time of our processing.
- Codeod: It contains thirteen codes (11, 12, 21, …) describing the type of destination. For more details please refer to the work of He et al. (2015).
- Tempstraje:
- Distanceod: The distance between the starting point and the destination of the same direction of the same line.
- Opus-id: Representing the card number, which is unique for each passenger and ensures analysing the travel activity of each individual. However, for confidentiality reasons, we do not have access to personal data and all transactions are anonymized.
- Code-produ: Containing 44 different types of fares provided by RTC.

3.1.3 Summary Statistics on the Dataset

The dataset contains 3,233,580 smart card transactions were generated by 159,499 cards from 2019/02/01 to 2019/02/28. As the first step of analysis, plotting could give us a first impression of the distribution of our data.

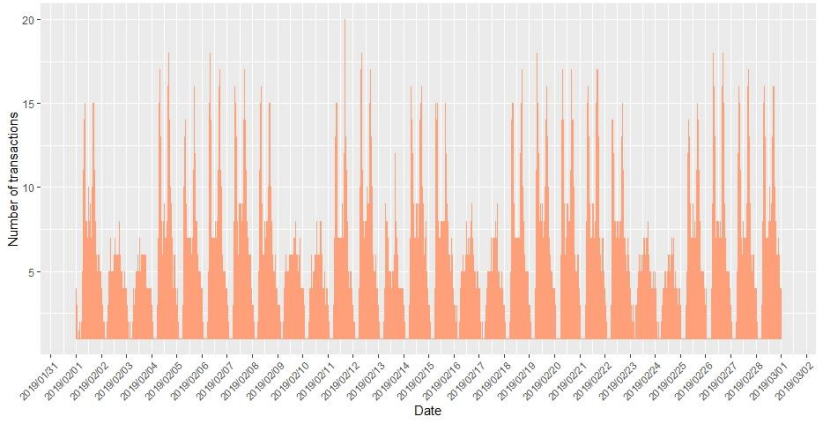


Figure 3-1: Hourly distribution per day in one month

Figure 3-2 illustrates the number of trips per day as well as the peak hours which is around 7:00 in the morning and 16:00 in the evening.

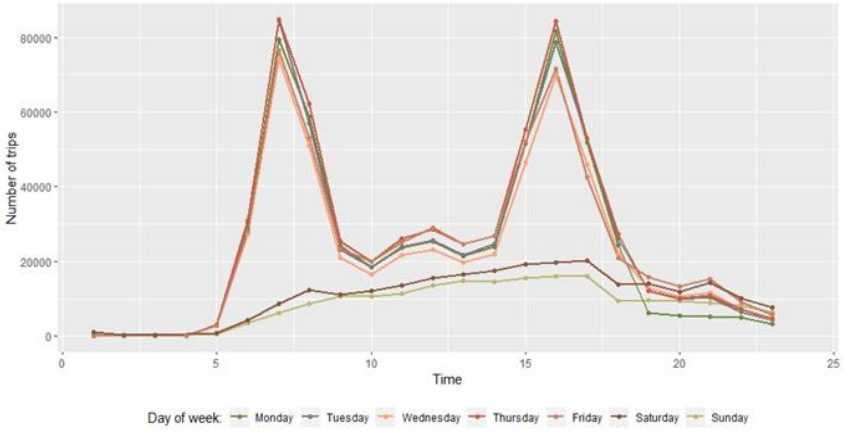


Figure 3-2: Hourly distribution in days of the week

3.2 Proposed Algorithm Structure

Based on what we discussed in Chapter 2, the gaps in previous research, and having in mind the aim of this research, which is to propose a suitable clustering approach for behaviour analysis in public transportation, we have designed the following steps to enlighten more the path of time-series data analysis in public transit by exploring a new method and comparing its results with two popular old ones. Therefore, as a usual need for preparation of dataset before applying any method, we must first do some preprocessing, which in our case consists of four sub-steps: transformation of validations into trips, vector preparation, classification of fare-types, and standardization. All the details of these sub-steps are presented separately in the following sections. Then, step 2 provides the three clustering methods; k-shape with SBD, k-means with DTW, k-means with ED distance, we describe the methods as well as how we use the *dtwclust* package in R to implement them. In step 3, we present our comparison framework which includes three perspectives.

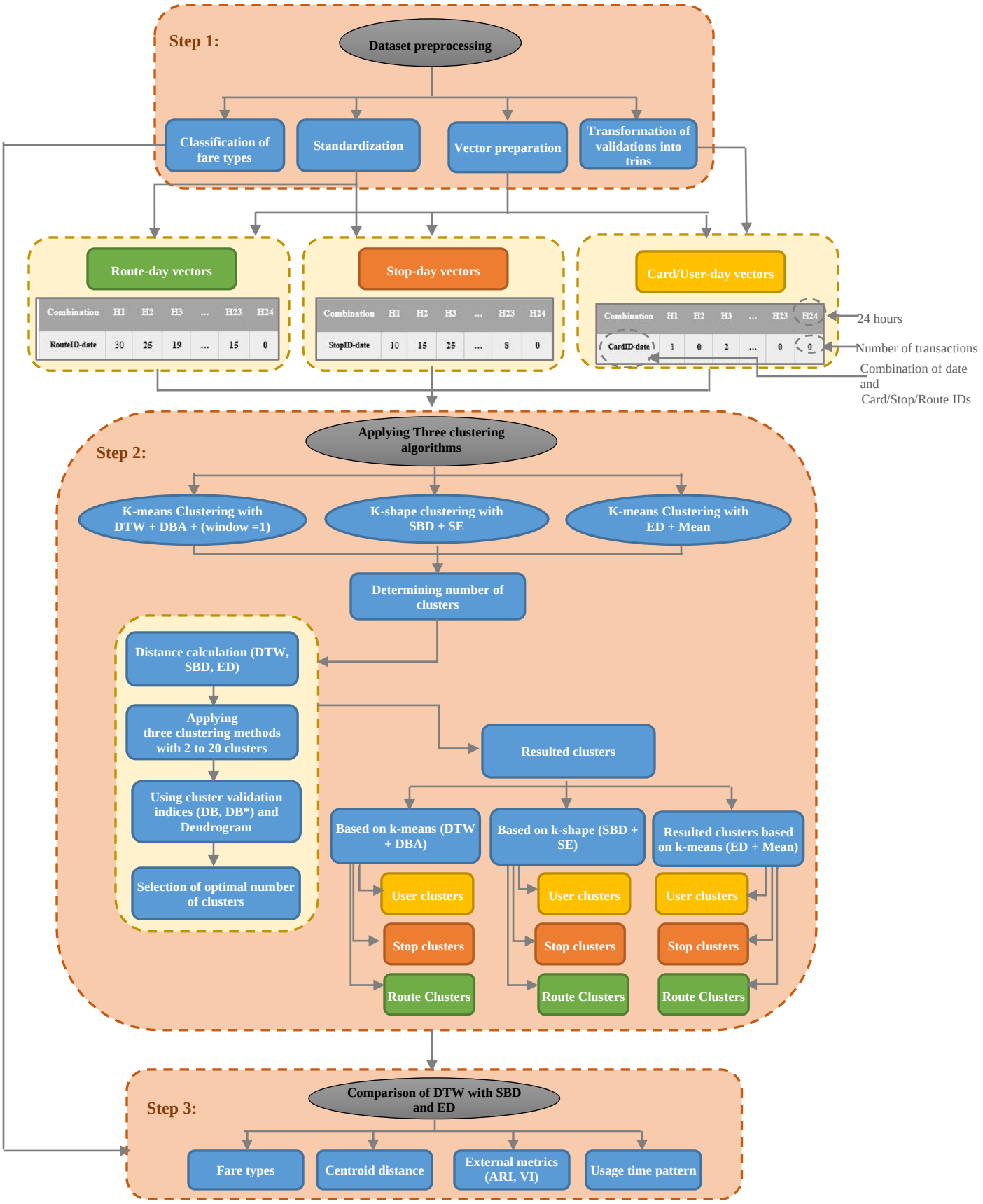


Figure 3-3: Schematic diagram of proposed algorithm

3.3 Proposed Algorithm Implementation

To perform our algorithm, we use R programming versions 3.6.1 and 4.0.5, and mainly *dtwclust* package.

3.3.1 Step 1-Dataset Preprocessing

Before any analysis, the smart card data should be preprocessed through some techniques such as data cleaning, data transformation, data reduction and, etc. In this study, since one of our objectives is extracting the travel behaviour of passengers through the beginning of the trips (boarding time) and the database contains some transfers between lines which should not be considered as independent trips, we need to apply some rules to distinguish between trips and transfers. In the next section, we present the rules we implemented to do so. The other important aspect for analysis is categorising the broad range of fare types such as annual, monthly, unlimited weekends, one day pass or some special passes offering by RTC to its passengers. Therefore, in section 3.3.1.2, we discuss how we categorised these fare types. Vector preparation is another necessary preprocessing step we should perform and since the type of vectors as the input of the model will affect the quality of the results, it is considered a critical step. In section 3.3.1.3 we present the procedure, we follow for creating vectors for three objects of users, stops, and routes.

3.3.1.1 Transformation of Validations into Trips

Whenever passengers use the bus services by tapping their smart card on the board, a validation is created. Regarding the fact that, some passengers might use their card between their origin and destination of their trips for changing the bus/line, they also create validations for the transfers. Thus, this is hard to distinguish validation as the origin of a trip or as a transfer (a part of the same trip). Given the fact that, we aim to analyse the boarding time (origin) of the trips for extracting passenger travel behaviour, we applied the following business rules of RTC's fare policy: (1) the first validation of a day is always the beginning of a new trip, (2) two validations that occur within 90 min and are made in different lines, are considered as part of the same trip (Deschaintres et al., 2019; Egu et al., 2020). In other words, for further user analysis and segmentation, the validation that meets the second rule considered as part of the same trip and will be deleted. The 90 min rule

is used in the calculation of the RTC's revenue allocation stipulating that a single ticket is valid up to 90 min from the previous validation.

3.3.1.2 Classification of Fare-Types

RTC's product code dictionary consists of 100 different types of fares which we presented in Appendix A. However, in the provided database only 44 fare types were used by users. To simplify further analysis, we first categorised fare types into 18 groups illustrated in Table 3-2.

Table 3-2: Fare product classification into 18 groups

Passenger	District	Time
Student	Inner city	Long-term (1)
		Short-term (2)
		Ticket (3)
	Outskirts	Long-term (4)
		Short-term (5)
		Ticket (6)
Adult	Inner city	Long-term (7)
		Short-term (8)
		Ticket (9)
	Outskirts	Long-term (10)
		Short-term (11)
		Ticket (12)
Senior	Inner city	Long-term (13)
		Short-term (14)
		Ticket (15)
	Outskirts	Long-term (16)
		Short-term (17)
		Ticket (18)

We first divided products into 3 base categories of passenger, district, and time. The group of "Passenger" contains 3 subcategories of student, adult, and senior. The "Student fare" holds the age of 6 to 18 and 19 and over, and "Senior fare" contains the age of 65 and over.

"District" represents 2 groups of inner city and outskirt for the areas covered by RTC's transport services. "Inner city" consists of Québec City area and "outskirt" or "Metropolitain" covers Québec City and Lévis according to RTC fare schedule.

“Time” represents the different times the passes are valid. The available passes for RTC services are annual, monthly, 5 consecutive days, unlimited weekend, one day pass and, etc. We divided time into 3 subcategories of long duration, short duration, and tickets. “Long-term” holds the annual and monthly passes and other passes are considered in “short-term”.

After grouping, we calculated the percentage of each fare-type during one month of analysis. As can be identified from the , some types of fare have zero or too small portion and around 9% of fare types have missing values. Regarding this, we decided to modify 18 groups in order to have more clear interpretation in the analysis part. In doing so, the inner and outskirt have merged, besides, the ticket groups have considered in short-term category. The resulted 6 groups consisting of “Student-long term”, ”Student-short term”, “Adult-long term”, “Adult-short term”, “Senior-long term”, “Senior-short term” with their frequencies is shown in Figure 3-5.

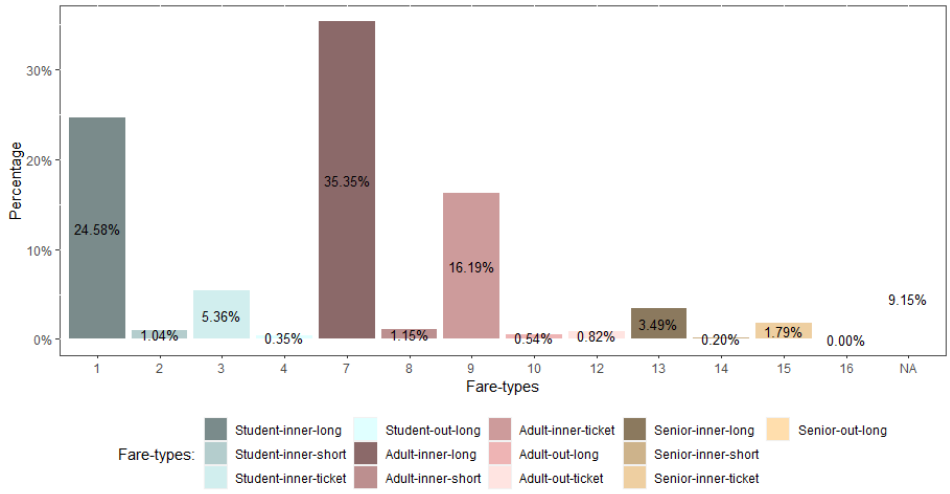


Figure 3-4: Fare-type percentage

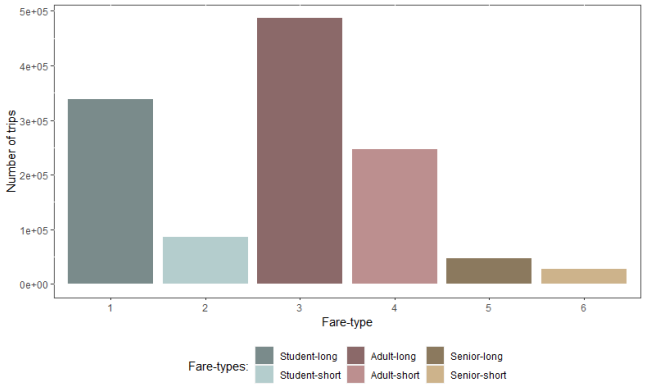


Figure 3-5: Fare-types frequencies

3.3.1.3 Preparation of Vectors

One of the main steps in clustering is the definition of vector space (Egu et al., 2020). In this study, we prepare vectors on 3 objects of card, stop, and route. For showing daily transit usage patterns, we decide to create a vector $V_n = \{v_1, v_2, \dots, v_n\}$, where $n = 24$ stands for 24 hours of a day and each v_i, takes the value representing the number of trips at the given hour of i. To be specific, when $v_8 = 3$, it means that between 8:00 and 8:59 the number of trips is equal to 3. In the following sections, we discuss in detail the procedure of vector creation for card/user, stop, and route separately.

3.3.1.3.1 Card-day (or User-day)

We assume that there is an unambiguous relation between users and cards (1 card = 1 user). Therefore, we simply refer to card-day as user-day profile. From the dataset with 3,233,580 smart card transactions, based on what we discussed in section 3.3.1.1, the total of 2,502,141 trips (boarding time) was obtained. Considering all trips for one user in one day as a one vector, a total of 1,356,537 user-day profiles (or vectors) was created with the 24 variables (or length) with the same interval represented each hour of a day. It means that user 1 in date 1 and user 1 in date 2 represent two different vectors. To do so, we combined columns of "Opus-id" and "Dateoperat" (date part) in one column and named it "idate". We then, based on column "Dateoperat" (hour part), created 24 variables each represent daily hours and the values are the number of trips in the corresponding hour. Table 3-3 shows an example of four resulted vectors. For instance, user "1000010" had one trip between 11:00 and 11:59, so H_{11}has the value of one.

Table 3-3: Example dataset of user-day

idate	$\mathbf{H_1}$	$\mathbf{H_2}$	…	$\mathbf{H_6}$	$\mathbf{H_7}$	…	$\mathbf{H_{11}}$	$\mathbf{H_{12}}$	…	$\mathbf{H_{24}}$
1000010_2019-02-25	0	0	…	0	0	…	0	0	…	0
1000010_2019-02-26	0	0	…	0	0	…	1	0	…	1
1000015_2019-02-01	0	0	…	1	0	…	0	1	…	0
1000015_2019-02-04	0	0	…	1	0	…	1	0	…	0
…										

3.3.1.3.2 Stop-day

There are 4,106 stops on the RTC bus-network that we aim to group them based on similar behaviour based on the number of trips in each hour of a day. For creating stop-day profiles, we used all transactions not only the trips. From 3,233,580 smart card transactions, we created 78,639 stop-day vectors with the length of 24, the same as what we did for user-day vectors. However, here instead of column "Opus-id" we used "Stop-o" which illustrates a unique number for each stop. From combination of "Stop-o" and "Dateoperat" (date part), we created a new column called "sdate". Table 3-4 shows an example of four stop-day vectors.

Table 3-4: Example dataset of stop-day

sdate	H_1	H_2	…	H_7	H_8	…	H_{12}	H_{13}	…	H_{24}
1005_2019-02-27	0	0	…	2	3	…	0	1	…	0
1005_2019-02-28	0	0	…	1	4	…	0	1	…	1
1006_2019-02-01	0	0	…	3	0	…	0	1	…	0
1006_2019-02-02	0	0	…	0	0	…	0	0	…	0
…										

3.3.1.3.3 Route-day

There are 195 lines on the RTC bus-network. For creating route-day vectors, we followed the same procedure as previous sections. From 3,233,580 smart card transactions, we created 3,912 route-day vectors with the length of 24. This time instead of "Opus-id" we combined the columns of "Codeligneo" and Dateoperat" (date part), we created a new column what we called "rdate".

Table 3-5: Example dataset of route-day

rdate	H_1	H_2	…	H_7	H_8	…	H_{15}	H_{16}	…	H_{24}
133_2019-02-27	0	0	…	170	70	…	48	143	…	0
133_2019-02-28	0	0	…	123	79	…	48	148	…	1
136_2019-02-01	0	0	…	172	72	…	47	134	…	0
136_2019-02-04	0	0	…	182	76	…	51	127	…	0
…										

3.3.1.4 Standardisation of Data

Data standardisation is often considered as a pre-processing step in cluster analysis. Since in creation of clusters a distance is used, this distance can be affected by dimension, scale, and unit

of the variables and change the whole results of clustering. Standardisation of raw data by converting them into a specific range using a linear transformation such that they have mean zero and standard deviation one, not only can create clusters with better quality but also it improves clustering accuracy (Mohamad & Usman, 2013).

There are three methods for data normalisation consisting of Z-score, Min-Max, and Decimal scaling. Mohamad et al. (2013) in their study compared the effect of these three normalisation methods on the k-means clustering results and they concluded Z-score is the most powerful one for improving clustering efficiency and accuracy.

In our dataset, for analysing stop-day and route-day vectors since the range of variables changes between 0 and more than 1000, we use Z-score method to standardise data we then apply clustering algorithms. However, in user-day analysis we do not use Z-score due to the small difference in the range of variables (0-12).

Suppose $Y = \{X_1, X_2, \dots, X_n\}$ is a d-dimensional raw data set then the data matrix is a $n \times d$ matrix as below:

$$X_1, X_2, \dots, X_n = \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nd} \end{pmatrix}$$

The Z-score of these data is obtained by:

$$x_{ij} = Z(x_{ij}) = \frac{x_{ij} - \bar{x}_j}{\sigma_j} \qquad 3\text{-}1$$

3.3.2 Step 2-Applying Three Clustering Techniques

In this section, we first discuss our methods, and how we apply them. Earlier mentioned, one of our goals is to test and adapt a novel k-shape clustering method with SBD, on the smart card data in public transit as the first attempt. To make a better comparison, we use the k-means clustering approach using the DTW and ED distance metrics. In this section, we go through the three methods we will be investigating at our research. We then present the techniques we employed for determining the optimal number of clusters for both of our proposed clustering methods.

3.3.2.1 K-shape clustering

K-shape clustering has built on the same successful k-means principles. It is based on an iterative refining technique that is similar to the k-means algorithm (more detail in Section 2.2.1) but differs significantly. Unlike k-means, k-Shape uses a different distance metric (SBD), and a different approach for centroid computation (SE) that we describe as following.

3.3.2.1.1 Shape-based distance (SBD)

Shape-based distance measure (SBD) is a normalised version of cross-correlation distance proposed by Paparrizos et al. (2017) to obtain shift-invariance. They used coefficient normalisation, $NCC_c(\vec{x},\vec{y}) = \frac{CC_w(\vec{x},\vec{y})}{\sqrt{R_0(\vec{x},\vec{x}).R_0(\vec{y},\vec{y})}}$, with the resulted values between [-1, 1]. Since it is sensitive to scale, authors also recommend *z*-normalisation of the sequence to have scale invariance as well. Once the amount of w in which $NCC_c(\vec{x},\vec{y})$ is maximum is determined, SBD will be calculated as follows:

$$SBD(\vec{x},\vec{y}) = 1 - \max_w \left(\frac{CC_w(\vec{x},\vec{y})}{\sqrt{R_0(\vec{x},\vec{x}).R_0(\vec{y},\vec{y})}}\right), \qquad 0 \le SBD \le 2 \tag{3-2}$$

Where 2 reflects the most dissimilarity while 0 indicates perfect similarity between $\vec{x}$ and $\vec{y}$.

(For more detail please refer to Paparrizos et al. (2017))

3.3.2.1.2 Shape extraction (SE)

Shape-extraction is a recently proposed method to calculate time-series prototypes which is a part of the novel *k*-Shape algorithm described in the study of Paparrizos et al. (2017) and it is based on SBD. As mentioned before, although the easiest way to capture the average of sequences is arithmetic mean, it is not a proper choice for averaging due to its lack of ability to extract all characteristics of time-series data. Therefore, Paparrizos et al. (2017) suggested to use the concept of optimisation problem; the minimum within-cluster sum of squared distances, which can be indicated as bellows.

$$P^* = \underset{p}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{\vec{x}_i \in p_j} dist\left(\vec{x}_i, \vec{c}_j\right)^2 \tag{3-3}$$

Where, $P = \{p_1, \dots, p_k\}$ is the number of clusters (partitions), $\vec{c}_j$ is the centroid of partition $p_j \in P$, $X = \{\vec{x}_1, \dots, \vec{x}_i, \dots, \vec{x}_n\}$ is the set of n observations. In other words, the objective is minimising distances between each centroid in a partition with all observations in that given partition. However, in the case of shape extraction, since shape-based and cross-correlation distance capture similarity - rather than dissimilarity - of sequences, Equation 3-3 changes to maximisers and based on Equation 3-2 we have:

$$\vec{\mu}_k^* = \underset{\vec{\mu}_k}{\operatorname{argmax}} \sum_{\vec{x}_i \in P_k} NCC_c(\vec{x}_i, \vec{\mu}_k)^2$$

$$= \underset{\vec{\mu}_k}{\operatorname{argmax}} \sum_{\vec{x}_i \in P_k} \left(\max_w \frac{CC_w(\vec{x}, \vec{y})}{\sqrt{R_0(\vec{x}, \vec{x}).R_0(\vec{y}, \vec{y})}} \right)^2 \qquad 3\text{-}4$$

This equation requires the computation of an optimal shift for every $\vec{x}_i \in P_k$. We use the previously computed centroid as a reference and align all sequences using SBD towards this reference sequence according to the context of iterative clustering. Since before the computation of the centroids, sequences are already aligned towards a reference sequence, we can omit the denominator of Equation 3-4. Then, by combining Equations 2-6 and 2-7, we will have:

$$\vec{\mu}_k^* = \underset{\vec{\mu}_k}{\operatorname{argmax}} \sum_{\vec{x}_i \in P_k} \left(\sum_{l \in [1,m]} x_{il}.\mu_{kl} \right)^2$$

For simplicity, this equation can be expressed with vectors and assume that the $\vec{x}_i$ sequences have already been z-normalised to handle the differences in amplitude.

$$\vec{\mu}_k^* = \underset{\vec{\mu}_k}{\operatorname{argmax}} \vec{\mu}_k{}^T . \sum_{\vec{x}_i \in P_k} (\vec{x}_i . \vec{x}_i^T) . \vec{\mu}_k \qquad 3\text{-}5$$

In this equation only $\vec{\mu}_k$ is not z-normalised. To handle the centring, we set $\vec{\mu}_k = \vec{\mu}_k . Q$, where $Q = I - \frac{1}{m} O$, I is the identity matrix and O is the matrix with all ones. Moreover, for making $\vec{\mu}_k$ to have a unit norm, we divide Equation 3-5 by $\vec{\mu}_k^T . \vec{\mu}_k$. Finally, by subtracting S for $\sum_{\vec{x}_i \in P_k} (\vec{x}_i . \vec{x}_i^T)$, we obtain:

$$\vec{\mu}_k^* = \underset{\vec{\mu}_k}{\operatorname{argmax}} \frac{\vec{\mu}_k^T . Q^T . S . Q . \vec{\mu}_k}{\vec{\mu}_k^T . \vec{\mu}_k}$$

$$\vec{\mu}_k^* = \underset{\vec{\mu}_k}{\operatorname{argmax}} \frac{\vec{\mu}_k^T . M . \vec{\mu}_k}{\vec{\mu}_k^T . \vec{\mu}_k} \qquad 3\text{-}6$$

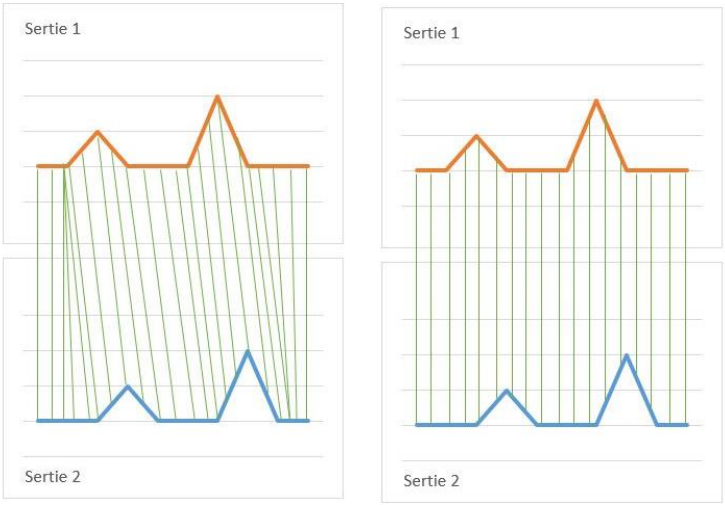
Where $M = Q^T . S . Q$. Using the preceding transformations, Equation 3-5 was simplified to the optimisation of this equation, which is a well-known problem called maximisation of the Rayleigh Quotient (Paparrizos et al., 2017).

3.3.2.2 K-means clustering with DTW and ED distances

As stated in Section 2.2.1, k-means performs two steps: (1) assignment step, which updates the cluster memberships by comparing each time-series based on a distance measure with all centroids and assigning each to the closest centroid; (2) refinement step, To reflect the changes in cluster memberships in the preceding stage, the cluster centroids are modified using the prototyping approach. It repeats these two processes until the cluster membership does not change or the maximum number of iterations is reached.

In this thesis, along with the k-shape clustering with SBD, we also perform k-means clustering with DTW distance and DBA prototyping procedure (more detail in Section 2.2.4), and k-means clustering with ED and mean as prototyping technique.

To understand the characteristics and differences of ED and DTW distance measures clearly, we compare them based on their alignment procedure. As can be seen from Figure 3-6, the two series are pretty similar in shape but slightly differ in phase (time). ED alignment without considering this possible variance in time series data, calculates a one-by-one distance between each element of sequence whereas DTW calculates a pairwise distance between all elements. Having this advantage, it provides more meaningful comparison while considering the possible shift. However, without any constraint in time shifting DTW will perform as Figure 3-7 illustrated. It is clear from this figure, without considering any restrictions on the warping path, the alignment can get stuck in similar features and ignore the difference. DTW's ability to determine a window size makes it a suitable distance measure particularly when the amount of shift (lag) is an important factor in time series comparison.



(a) (b)

Figure 3-6: Alignment between two series:
(a) under DTW, (b) under ED

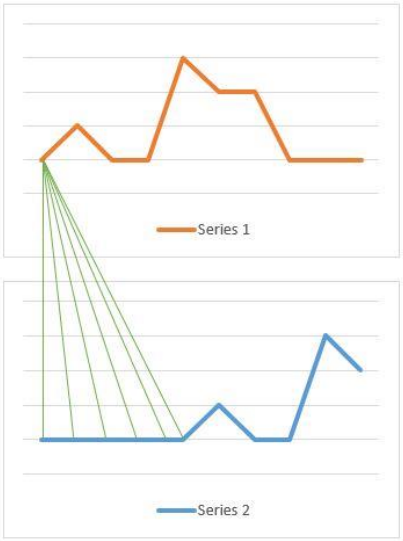


Figure 3-7: DTW alignment without
constraint

Although DTW is the most used and proper similarity measure for comparing sequences, because of the lack of a suitable averaging method, its applicability was reduced until Petitjean et al. (2011) proposed a global averaging method, called DTW barycentre averaging (DBA). In their study, they showed that this method not only covers all drawbacks of previous averaging method under DTW

space, but also its behaviour is robust and outperforms other techniques. Since then, combining k-means with DBA which relies heavily on averaging, provides a reliable method for clustering time-series data using DTW distance measure.

All in all, we presented in this chapter different distance measures and prototyping techniques while emphasising their advantages and disadvantages based on the literature. In the following chapters, we will apply our three methods with different distance measures and averaging techniques on three different types of vectors to highlight their benefits, specifically on our dataset.

For the sake of simplicity, we name the three clustering approaches "DTW," which stands for k-means clustering with DTW distance measure, "SBD," which represents k-shape clustering with SBD, and "ED," which refers to k-means clustering with Euclidean distance.

To apply each of these methods, we only need to change the distances to DTW, SBD, and ED, as well as the prototyping methodologies to DBA, SE, and Mean, respectively, and in case of DTW, we set 1 for parameter window. Since we use R programing and *dtwclust* package, all of these options are implemented in *tsclust* function.

Furthermore, because k-shape clustering is based on k-means and it needs the number of clusters as an input, we follow the procedure of determining the optimal number of clusters which has been presented in the next section.

3.3.2.3 Determining the optimal number of clusters

As previously stated, k-means and k-shape require a preceding number for clusters. Cluster validation indices are one method for determining this value. In doing so, clustering is used by considering different numbers for clusters, these indices are calculated for each result. The number of these indices, which are an indicator of purity and well-separated segments, is then used to compare the quality of the resulting clusters. The given number of clusters that yield to the better resulted indices, would be the best choice as a prior cluster number. In our study, we used internal cluster validation indices (CVIs) that was discussed in detail in Section 2.2.5. Nevertheless, along with using this method, domain knowledge can play a crucial role.

Figure 3-8 illustrates an example of resulted DB and DB* indices considering different numbers for clusters (2 to 20). As can be seen, the minimum has been occurred in 5, 9, 13 for both lines meaning that by choosing each of them as cluster number considering other conditions, the resulted clusters contain more purity and are well separated which indicates the good performance of clustering algorithm when choosing 5, 9 or 13 as a prior number for clusters.

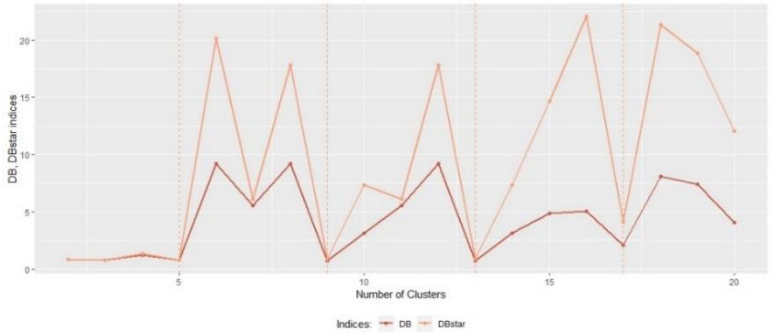


Figure 3-8: Example of resulted DB and DB* indices for 2 to 20 clusters

Figure 3-9 also shows an example of resulted Sil index. In this index since the maximum amount indicates the better result of clustering, the optimal choices are, 10, 9, 11, 6 and 4.

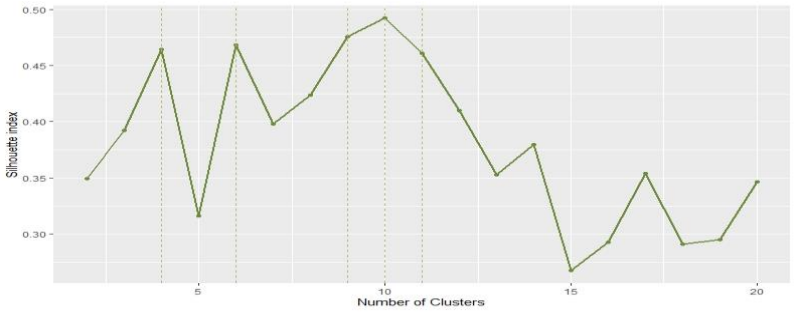


Figure 3-9: Example of resulted Sil index for 2 to 20 clusters

Regarding the fact that, Silhouette index calculates the entire distance matrix between the series in the data, which can be prohibitive for methods with random centroid selection techniques, such as SE and DBA (Sardá-Espinosa, 2018), we employ a dendrogram approach across 30 centres. The prior number of clusters is set to 30 in this two-step procedure, and then the clustering algorithm is

applied to the entire dataset. The 30 centroids are then plotted on a dendrogram. The ideal number of clusters is chosen based on this dendrogram (Morency et al., 2017).

Figure 3-10, shows an example of the resulted dendrogram has been drawn over 30 vectors as the centres of the 30 clusters.

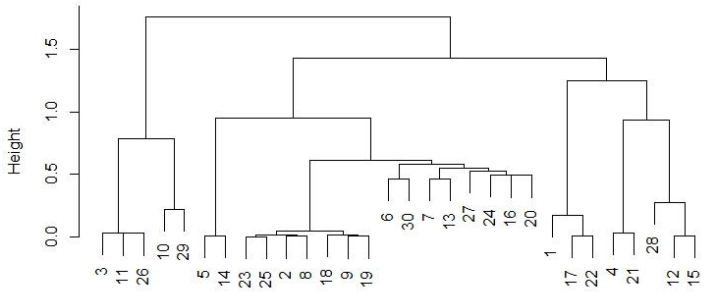


Figure 3-10: Example of resulted dendrogram over 30 centroids

3.3.3 Step 3-Comparison of Three Clustering Techniques

Three criteria guided the development of our comparison system. First, the DTW distance between the centroids of DTW clusters and the centroids of SBD and ED clusters is calculated. After that, we take an average of the resulted distances. The one with the minimum average distance would agree more with DTW partitions.

Second, as we explained in Section 2.2.5.2, the comparison is based on two external measurements. Each approach has a larger ARI value, and a lower VI value is more compatible with DTW partitions.

Finally, we compare the clustering methods based on their patterns, rather than solely using statistical measurements. This would help us to see the differences in detail and with each cluster patterns, portions, and distributions. For the user-day vectors, we also use the difference of the methods in fare-type distribution.

CHAPTER 4 CARD-DAY ANALYSIS

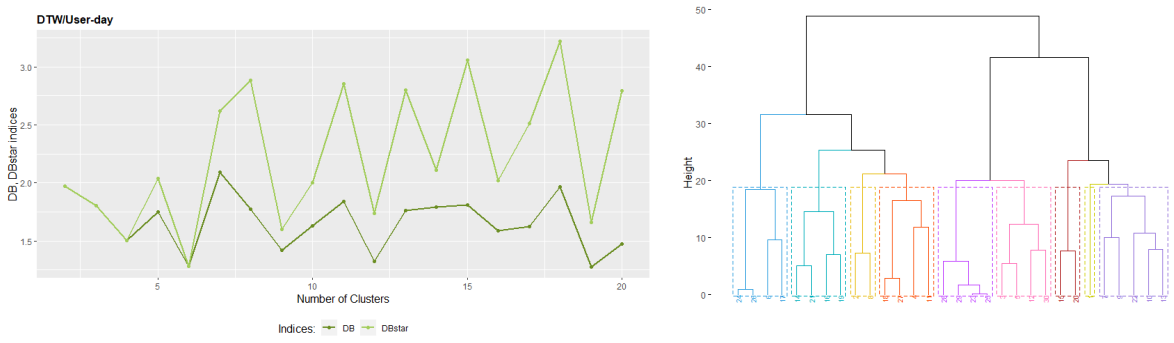
In this chapter, the three clustering algorithms of DTW, SBD, and ED applying to the user-day vector will be presented and compared. We first discuss how we decide on the optimal number of clusters for these methods. The resulting clusters are then compared from four perspectives: distance between centroids, external measures, usage time, and fare-type.

4.1 Number of Groups

As explained in the previous chapter, we used DB and DB* techniques, as well as a dendrogram, to discover the optimal number of clusters. Since there is no one-size-fits-all solution in this regard, the best strategy is to try out different methods and make a decision based on their agreement.

For DB and DB*, we applied DTW, SBD, and ED on our dataset with setting the number of clusters from 2 to 20. And then, the amounts of DB and DB* corresponding to each cluster number were calculated as shown in part (a) from Figure 4-1,Figure 4-2, and Figure 4-3. In part (b) of these figures, the dendrogram obtained from 30 cluster centres resulting from DTW, SBD, and ED is illustrated.

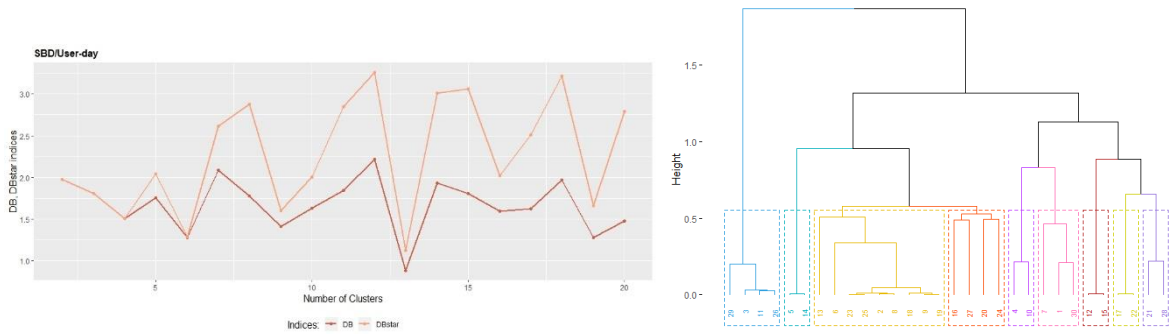
Figure 4-1 (a) shows the minimum amounts of DB and DB* when the number of clusters in DTW is 6, 19, 9, and 12. Also, Figure 4-1 (b) in the resulted dendrogram, looking top to bottom, we observe that the split to 6 clusters causes a significant drop in the amount of error and the biggest successive splits occur at 9 and 12 clusters; 12 is also a good choice but a negligible difference in comparison to 9. In this case, the number 9, which is neither too big nor too small, appears to be a good choice.



(a) (b)

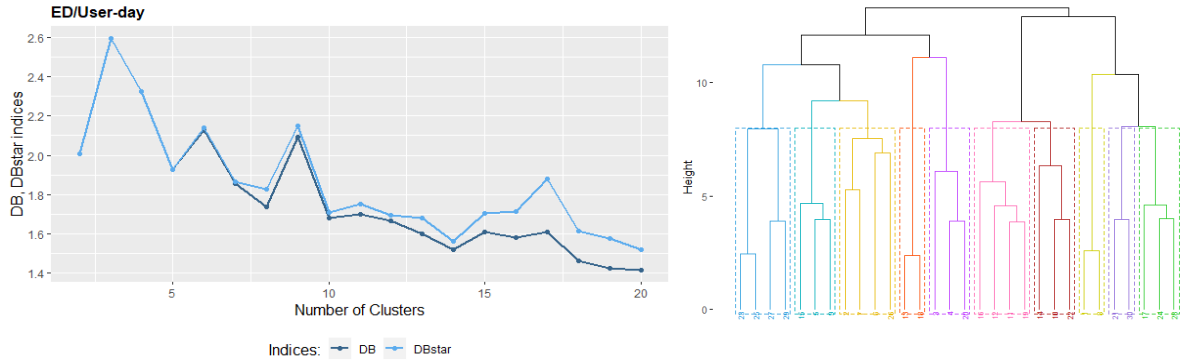
Figure 4-1: Selection of the optimal number of clusters for users under DTW by: (a) DB and DB* (b) dendrogram

As illustrated in Figure 4-2 (a), applying SBD for 2 to 20 clusters, the minimum amounts of DB and DB* occur in 13, 6, 9, and 19 clusters. Figure 4-2 (b) also displays the resulted dendrogram over 30 centroids, and likewise DTW, it seems cutting the dendrogram into 6 and then into 9 clusters reduce the amount of error noticeably. Although the reduction in error is also visible in 13 groups, we can ignore this difference and choose 9 as the best number. The same procedure was followed for ED, Figure 4-3, and we decided 10 clusters is the best choice.



(a) (b)

Figure 4-2: Selection of the optimal number of clusters for users under SBD by: (a) DB and DB*, (b) dendrogram



(a) (b)

Figure 4-3: Selection of the optimal number of clusters for users under ED by: (a) DB and DB*, (b) dendrogram

4.2 Comparison of SBD and ED with DTW

We compare and assess the resulting clusters of applying DTW, SBD, and ED methods from four perspectives in this section. First, the comparison is based on the mean distance between the clusters' centroids, then the amount of two external measures is considered, followed by the time of usage, and finally, the type of tariff.

4.2.1 Based on Distance Between Clusters

As we emphasised in this study, distance measure is used for calculating (dis)similarity between two objects. Thus, one way to compare SBD and ED with DTW is calculating the distance between their clusters and see which of them are closer to DTW indicating more agreement. With the aim of doing so, there are three approaches: (1) Single linkage; measures the distance between the closest members of the clusters, (2) Complete linkage; measures the distance between the most distant members, and (3) Centroid comparison; measures the distance between the centers of the clusters (Pérez, 2020).We decided to use the third one in our case.

Figure 4-4 shows the heat map illustrating the distance between the cluster centroids of DTW with SBD and ED. The more two centers are closer; the more colour is lighter and shows the better choice for matching. To calculate the distance, we used DTW distance measure with window equals to 1, and due to the difference in the range of centroids, we standardised them before calculating their distance for more meaningful comparison.

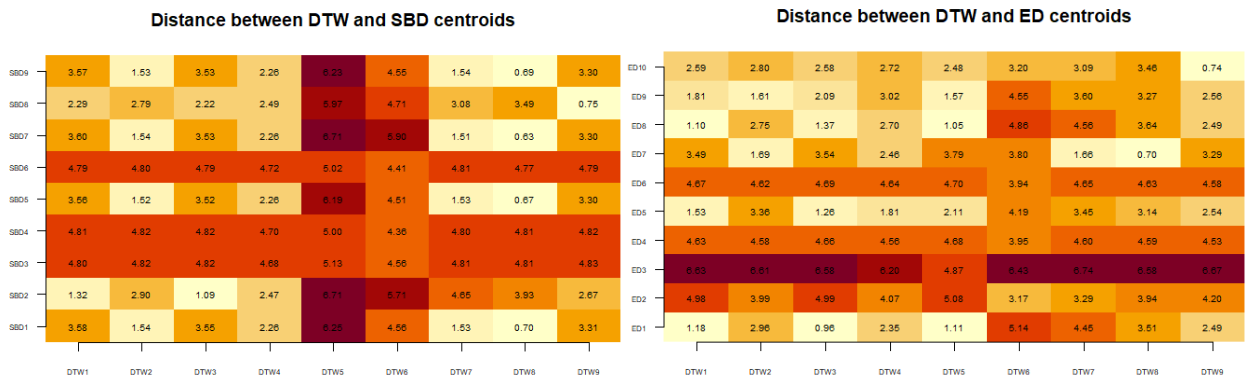


Figure 4-4: Distance between DTW and:
(a) SBD, (b) ED, user clusters

The final distance was calculated using the average, producing 3.696 and 3.619 for SBD and ED, respectively. Since the distance between DTW and ED is smaller than the one between SBD and ED, it can be assumed that the results of DTW and ED are more in agreement. However, it is preferable to compare them from other perspectives due to the tiny difference between them.

4.2.1.1 Label Matching

K-means and k-shape algorithms assign labels to groups randomly, therefore cluster 1 might be generated with different patterns in each run. As a result, it is useful to match SBD and ED cluster labels with DTW in order to make a more explicit comparison, mainly in a graphical format. As stated in the previous part, measuring the distance is one form of comparison.; we thus, matched the labels of SBD and ED clusters with DTW using the minimum distance between the centroids showing in the heat maps Figure 4-4.

Table 4-1 and Table 4-2, show the steps we followed to determine the best match according to these heat maps. First, we ordered the clusters from the least to the highest distance amount for every SBD clusters. We then, detected the clusters are identified as the best matches for more than one cluster. For instance, in Table 4-1, DTW cluster 6 has been determined as the best match for SBD clusters 3, 4, and 6 (green-coloured). Observing the heat map, we saw the distance between DTW cluster 6 from these clusters is 4.56, 4.36, and 4.41, respectively. So, DTW cluster 6 would be the best match for SBD cluster 4 with a distance of 4.36. We kept DTW cluster 6 for SBD cluster 4 and switched SBD clusters 3 and 6 to their second-best matches, and this process was repeated until all of the clusters' matches are unique.

Table 4-1: SBD matched labels by distance

Partition		SBD								
DTW	Matched labels (distance based)	1	2	3	4	5	6	7	8	9
		8,7,2,4,9,3,1,6,5	3,1,4,9,2,8,7,6,5	6,4,1,7,8,2,3,9,5	6,4,7,8,1,2,3,9,5	8,2,7,4,9,3,1,6,5	6,4,8,3,9,1,2,7,5	8,7,2,4,9,3,1,6,5	9,3,1,4,2,7,8,6,5	8,2,7,4,9,3,1,6,5
		8	3	6	6	3	6	8	9	8
		7	1	4	6	3	4	7	9	8
		2	1	4	6	3	8	7	9	8
		2	1	4	6	3	3	7	9	8
		2	1	4	6	3	9	7	9	8
		2	1	4	6	3	1	7	9	8
		2	1	4	6	3	2	7	9	8
		2	1	4	6	3	7	7	9	8
		2	1	4	6	3	5	7	9	8

Table 4-2: ED Matched labels by distance

Partition		ED									
DTW	Matched labels (distance based)	1	2	3	4	5	6	7	8	9	10
		3,5,1,4,9,2,8,7,6	6,7,8,2,4,9,1,3,5	5,4,6,8,3,2,1,9	6,9,4,2,8,7,1,2,5	3,1,4,5,9,8,2,7,6	6,9,2,8,4,7,1,3,5	8,7,2,4,9,1,3,5,6	5,1,3,9,4,2,8,7,6	5,2,1,3,9,4,8,7,6	9,5,3,1,4,2,7,6,8
		3	6	5	6	3	6	8	5	5	9
		3	6	4	9	1	9	8	5	2	9
		3	6	4	4	1	2	8	5	2	9
		3	6	6	4	1	8	8	5	2	9
		3	6	8	4	1	4	8	5	2	9
		3	6	3	4	1	7	8	5	2	9

Furthermore, there is another method for matching the labels which is easier to use and more straightforward. The function of *matchLabels* in the R package *WGCNA* which is not dependent on distance but rather it considers the objects are grouped in the clusters based on the contingency table. In other words, this function is based on Fisher's exact test to determine if there are non-random associations between clusters. The resulted matched labels from distance and Fisher's exact test are different. This method unlike the centroid distance which only considers the distance between clusters' centroids, compares them based on the assigned objects that seems a more reliable approach. So, we decided to use the labels in Table 4-3 and Table 4-4 for the comparison in the following sections.

Table 4-3: SBD matched labels by Fisher's exact test

Partition		**SBD**								
DTW	**Matched labels (Fisher's exact test)**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
		4	1	2	8	6	5	3	9	7

Table 4-4: ED matched labels by Fisher's exact test

Partition		**ED**									
DTW	**Matched labels (Fisher's exact test)**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
		3	1	5	4	7	6	8	1	2	9

4.2.2 Based on External Measurements

As pointed out in Section 2.2.5.2, ARI calculates the number of data points in the same/different resulted groups for comparing two clustering methods and VI consider the non-overlapping parts of two methods. Table 4-5 shows the contingency table used for calculating ARI, to avoid repetition, we did not present here the contingency table for ED and DTW.

Table 4-5: Contingency table between SBD and DTW

Partition		DTW									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**SUM**
SBD	**1**	63647	19714	136776	28449	6597	6	11392	459	66218	**333258**
	2	19490	17210	4079	1534	6187	4302	3007	5187	8473	**69469**
	3	11107	3026	14254	1034	7885	767	4064	10161	9041	**61339**
	4	3979	1963	800	50070	4989	2225	54312	19285	9228	**146851**
	5	23202	55782	15112	61087	89885	1088	254	38282	22460	**307152**
	6	3619	1437	14290	3457	2466	3611	86612	1925	1897	**119314**
	7	3044	1394	7040	31997	2062	1645	144313	3934	5574	**201003**
	8	13129	2956	3745	1677	6592	1086	854	5117	14278	**49434**
	9	22373	10748	7167	1252	4531	3557	2544	5583	10962	**68717**
	SUM	**163590**	**114230**	**203263**	**180557**	**131194**	**18287**	**307352**	**89933**	**148131**	**1356537**

In this study we mainly use *dtwclust* package in R, ARI and VI index are implemented in the main function of *cvi* in this package, so we did not use the contingency tables directly. Regarding the fact that DTW is a popular most used distance measure for time-series clustering we considered its results as a ground truth to compare with the results of novel SBD and ED methods. We also calculated ARI and VI based on two results of matched labels we obtained in the previous part to see if the amounts of these index depend on the labels. Table 4-6 shows the amount of both ARI and VI does not depend on the labels and remains constant in three different label assignment scenarios.

Table 4-6: External measures for SBD and ED

External measures	**SBD**	**SBD matched labels/distance**	**SBD matched labels/Fisher**	**ED**	**ED matched labels/distance**	**ED matched labels/Fisher**
ARI	0.184	0.184	0.184	0.099	0.099	0.099
VI	1.322	1.322	1.322	1.254	1.254	1.254

ARI ranges from 0 to 1, while 0 indicating that two clustering approaches are distinct and 1 shows they are identical. VI starts at 0 for similar partitions and grows greater as the partitions become more dissimilar. The bigger amount of ARI for SBD, in Table 4-6, shows higher agreement between SBD and DTW than ED and DTW, whereas the smaller VI for ED challenges this conclusion.

4.2.3 Based on Usage Time

The resulted cluster centroid patterns from applying DTW, SBD, and ED are plotted over 24 hours. We used the matched labels for SBD and ED resulted from Fisher's exact test and coloured them as same as corresponding DTW cluster labels. For instance, cluster 6 from SBD and cluster 3 from ED, are matched with cluster 5 from DTW, all has been coloured in orange. To characterise these patterns and to have their portions of the dataset, we plotted the pie charts are shown in Figure 4-5, Figure 4-6, and Figure 4-7.

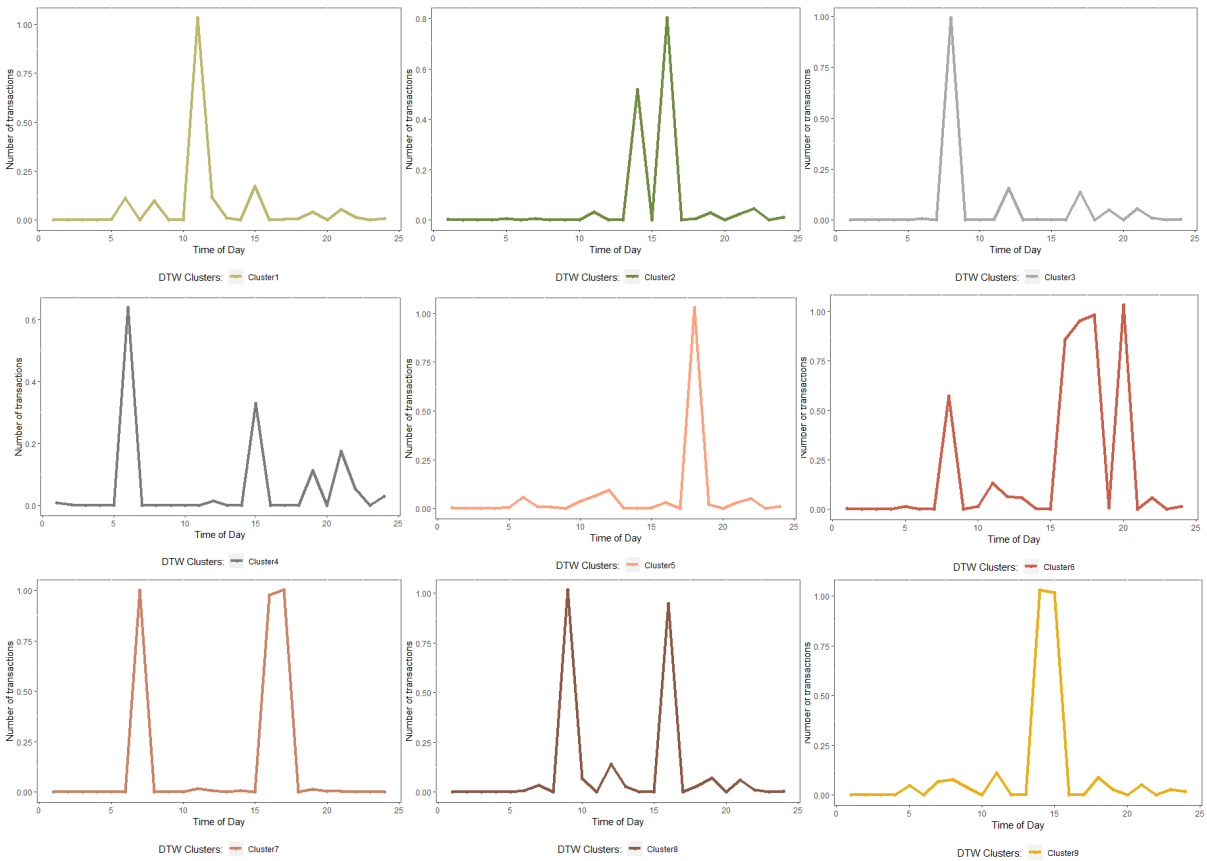


Figure 4-5: DTW user clusters' patterns

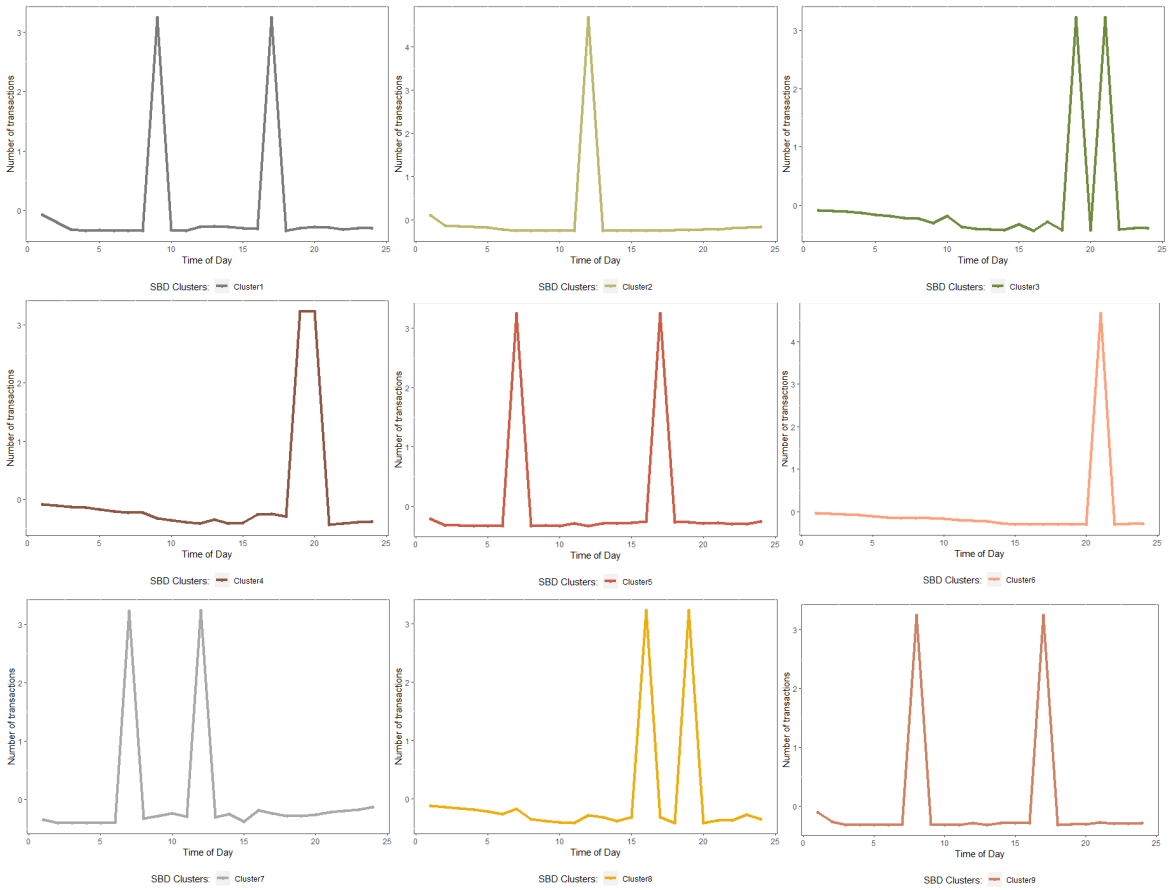


Figure 4-6: SBD user clusters' patterns

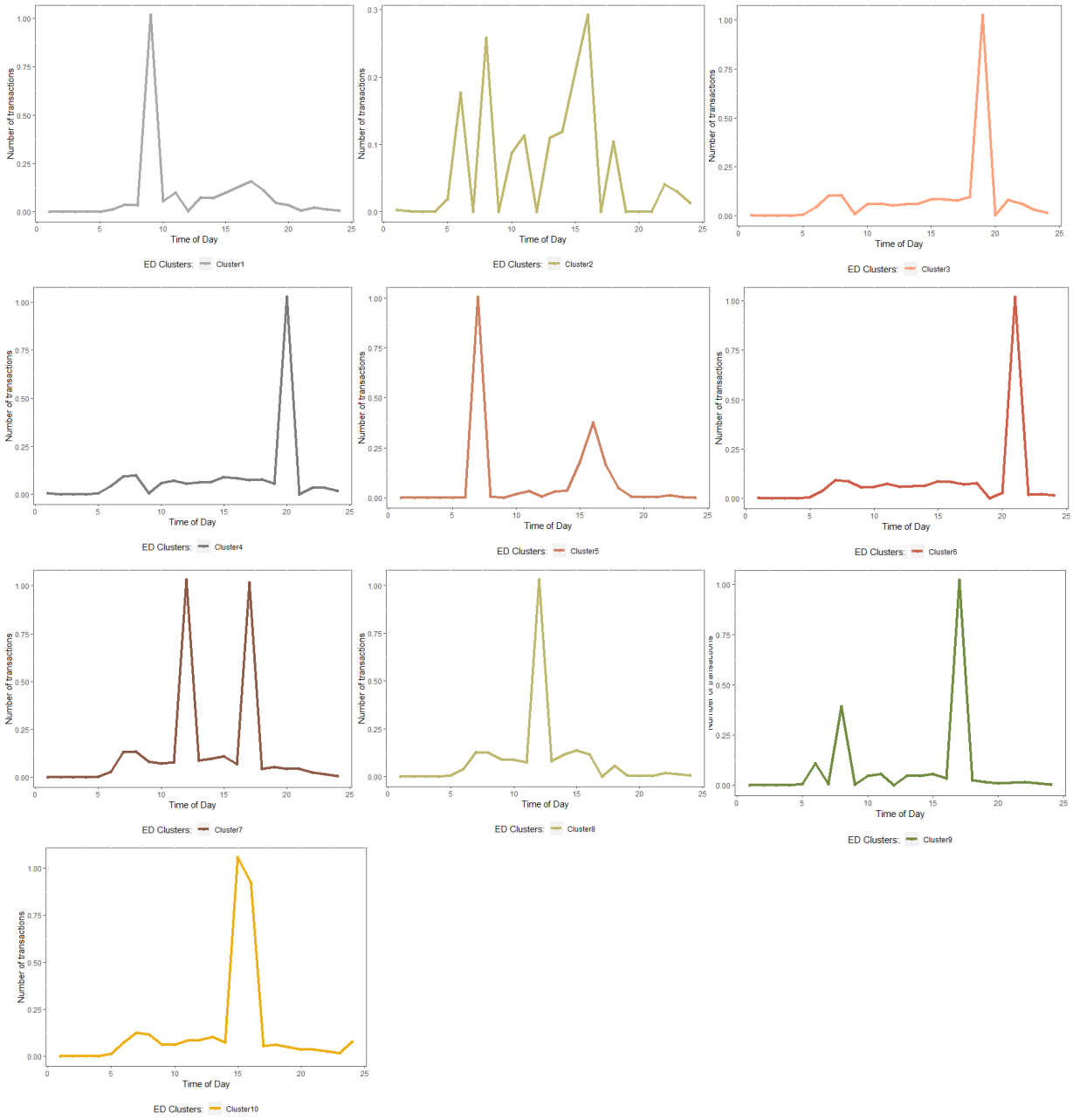
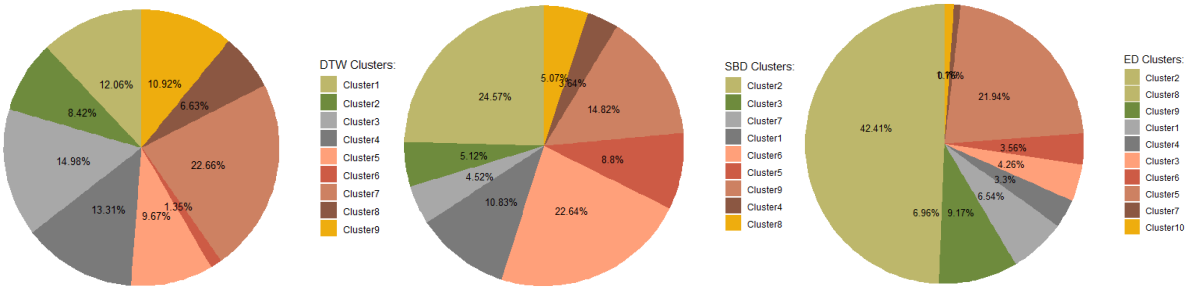


Figure 4-7: ED user clusters' patterns



(a) (b) (c)

Figure 4-8: Clusters' portions: (a) DTW, (b) SBD, (c) ED

According to Figure 4-5, the patterns of DTW clusters were categorised in four main groups; considering the portions in Figure 4-8, for all three methods clusters, the characterisation is as follows:

1. Regular commuters: This category consists of users with the frequent of usage mostly twice a day in the morning and in the afternoon. In DTW, clusters 4, 6, 7, and 8 with the 43.95% of total users belong to this group. Based on matched clusters, in SBD, clusters 1, 4, 5, and 9 should be in this category but based on the patterns, cluster 4 does not have the same characteristics of this group instead cluster 7 does. Therefore, SBD clusters 1, 5, 7, and 9 are identified as having the regular pattern with the 44.04% portion of all users. For ED, cluster 4, 5, 6, and 7 are in this group based on the matched clusters. However, it is obvious from the patterns in Figure 4-7,clusters 4 and 6 are not among this group while cluster 9 has the same characteristics of this group. So, In ED, clusters 5, 7, and 9 with the total portion of 31.87% have this group pattern.
2. Midday commuters: This group is identified as the users who use public transit mostly around the lunch time. In DTW, clusters 1, 2, and 9 are in this category with the 31.4% of total users. In SBD, based on matched labels, clusters 2, 3, and 8 belong to this category, however, clusters 3 and 8 can hardly be considered as midday commuters. So, the only cluster in SBD that have the same characteristics of this group is cluster 2 with the portion of 24.57%. Based on matched labels, clusters 2, 8, 9, and 10 from ED are categorised in this group, but cluster 2 does not have a well-defined pattern and cluster 9 is in the regular group, so for ED, clusters 8 and 10 with 8.06% of users are members of this group.

3. Late commuters: This group consists of users with the usage in late evening. In DTW, users in cluster 5 are identified as the late commuters with 9.67%. In SBD, cluster 6 is in this category according to the matched labels, though based on the patterns, clusters 3, 4, and 8 are also having the same characteristics of this group. Therefore, the total portion of 36.47% users is grouped in this category by SBD method. Cluster 3 is the only cluster from ED is identified in this group based on matched labels, but clusters 4 and 6 are also having the same pattern of this group with the total portion of 11.12%.
4. Early bird commuters: In DTW, cluster 3 belongs to this group with 14.98% of all users. In SBD, although based on the matched labels cluster 7 should be considered in this group, it more has the characteristics of regular commuters than early birds. Based on SBD patterns, Figure 4-6, there is no cluster having the same pattern of this group. But ED has cluster 1 based on matched labels and also the patterns which is among this group of commuters with the portion of 6.54%.

In terms of DTW and SBD comparison, we can observe that the most similar portion of users have been segmented in the regular commuter group by both methods. The least similar portions belong to late and early bird commuters. This reveals what we expected from the behaviour of SBD method in the creation of groups. Because SBD, unlike DTW, does not consider the shift in time and only considers the similarity in shape; in cased of significant shift, it could mistakenly assign users with the early-bird pattern to the group of late commuters or vice versa. It can, nevertheless, produce satisfactory outcomes in the case of slight shift in time, as what it did in the creation of regular and midday commuters in our case.

On the other hand, while ED shaped the groups in all four categories in the same way as DTW did, its portions in each of them differ dramatically from those in DTW. ED method also created a non-well definable pattern in cluster 2 consisting of a noticeable portion of 42.41% of users, which we could not place in any of the four categories.

The daily distribution of these four categories over one month for three methods is shown in Figure 4-9, Figure 4-10, and Figure 4-11. As can be seen, the distribution of regular commuters is similar in three methods. DTW and SBD also have roughly the same distribution in midday group while in SBD, late commuters is the dominant group consisting of late and early-bird users. Comparing

DTW and ED, we observe that although ED outperformed SBD in terms of creating late and early-bird groups, its dominant category, identified as non-well definable pattern, includes users from both regular and midday patterns, implying that this algorithm may not be effective in recognising all patterns in our case.

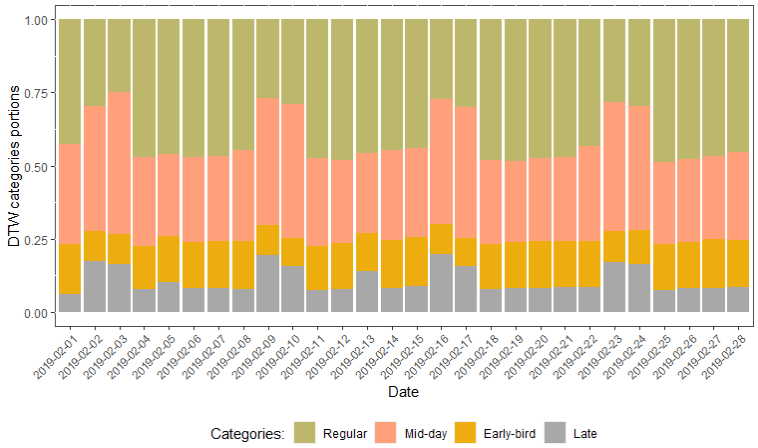


Figure 4-9: Distribution of DTW categories over one month

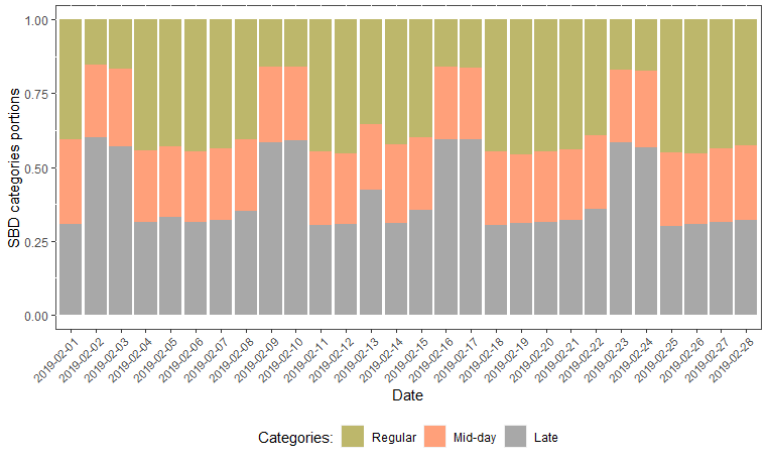


Figure 4-10: Distribution of SBD categories over one month

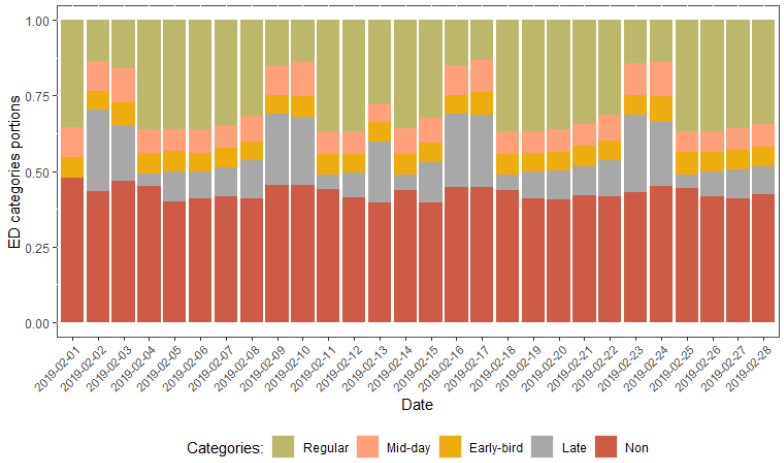


Figure 4-11: Distribution of ED categories over one month

4.2.4 Based on Fare-Type

Figure 4-12 shows the distribution of fare-type revealing that the highest percentage of users correspond to "Adult-long-term", "Student-long-term", and "Adult-short-term", respectively no matter which day of the week. Also, the lowest percentage belong to "Senior-short-term", "Senior-long-term", and "Student-short-term" respectively on weekdays, however, "Student-short-term" and "Senior-long-term" have headed similar frequency on weekends.

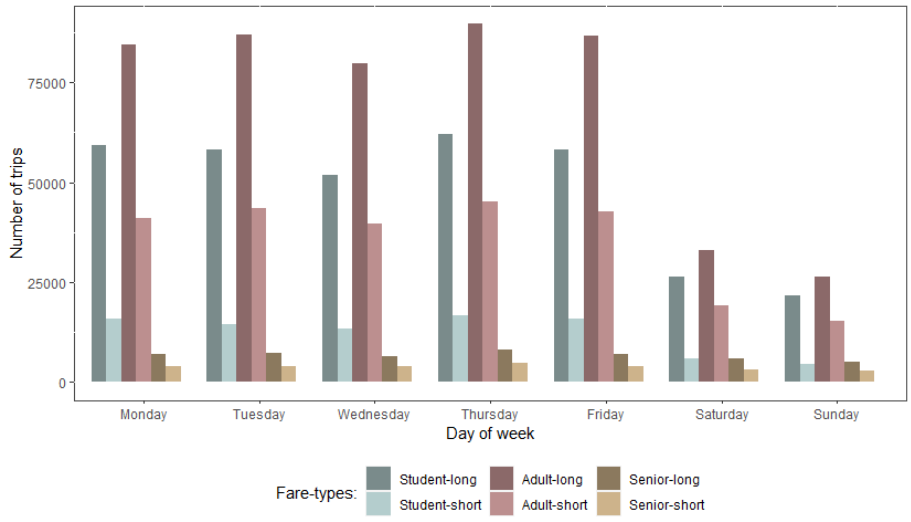
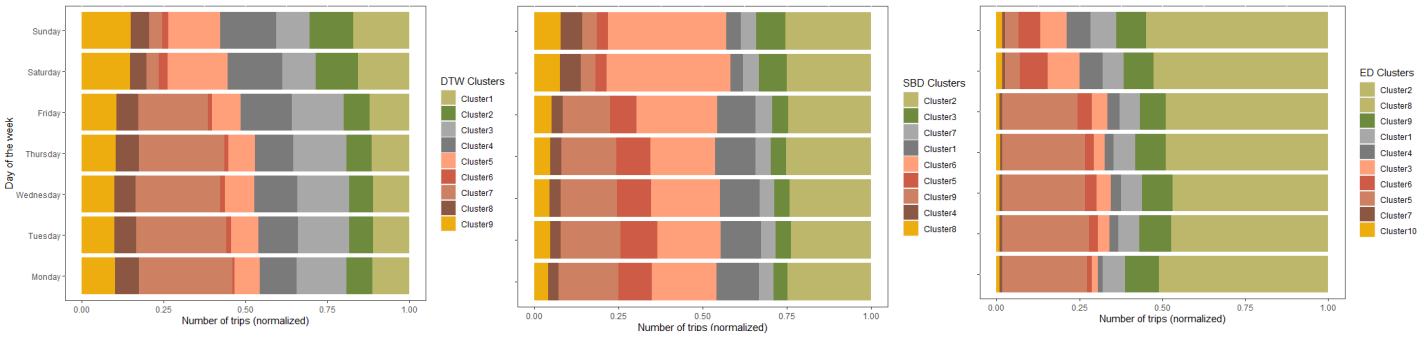


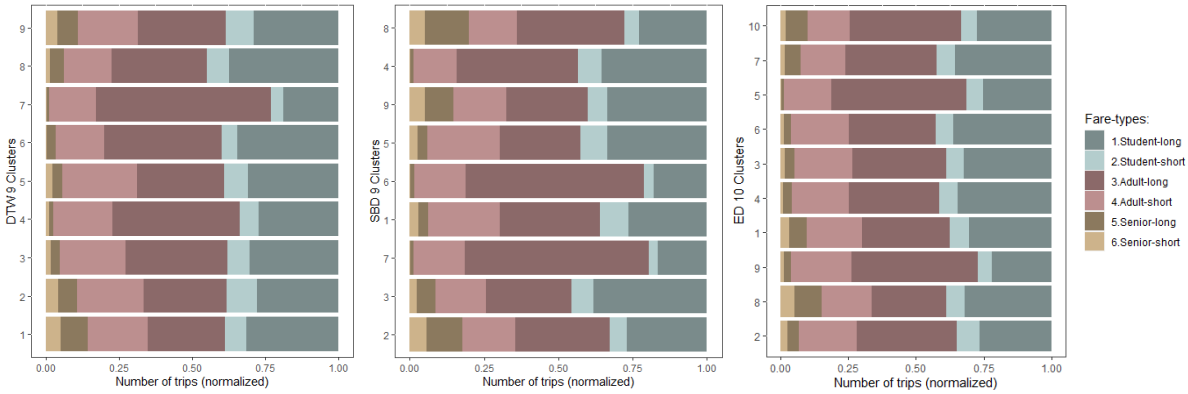
Figure 4-12: Frequency distribution of fare types by day of the week

According to Figure 4-13, on weekdays the portion of users that have been grouped in DTW cluster 7 is noticeably more than weekends. Its matched clusters in SBD, cluster 9, and in ED, cluster 5 follow the same pattern. In addition, based on ,the distribution of each fare-type for DTW cluster 7 and ED cluster 5, are the same, belonging mostly to “Adult-long-term”, “Adult-short-term”, and “Student-long-term” while SBD cluster 9 consist of all types of fares. Moreover, DTW cluster 5 and its matched ones in SBD and ED, clusters 6 and 3, respectively, have the more portions in weekends than weekdays. Again, the distribution of fare types corresponding to these clusters, reveals the more compatibility between DTW and ED than DTW and SBD. Likewise, DTW clusters 1, 2 and 9 have the more portions on weekends, their matched clusters in SBD, clusters 2, 3 and 8 have the same distribution for both clusters and fare types based on Figure 4-13 and Figure 4-14. We also can conclude that for these clusters which having the pattern of more frequency in weekends, the portion of senior fare-type besides Adult and student, is considerable. In addition, ED clusters 2, 8, and 9, are the same as DTW in fare-type distribution but different in cluster distribution.



(a) (b) (c)

Figure 4-13: Distribution of clusters by day of the week:
(a) DTW, (b) SBD, (c) ED



(a) (b) (c)

Figure 4-14: Distribution of fare-type versus clusters:
(a) DTW, (b) SBD, (c) ED

Considering all forms of comparisons, it is clear that neither ED nor SBD approaches completely outperformed the other one in the case of user-vector analysis for our dataset.

CHAPTER 5 STOP-DAY ANALYSIS

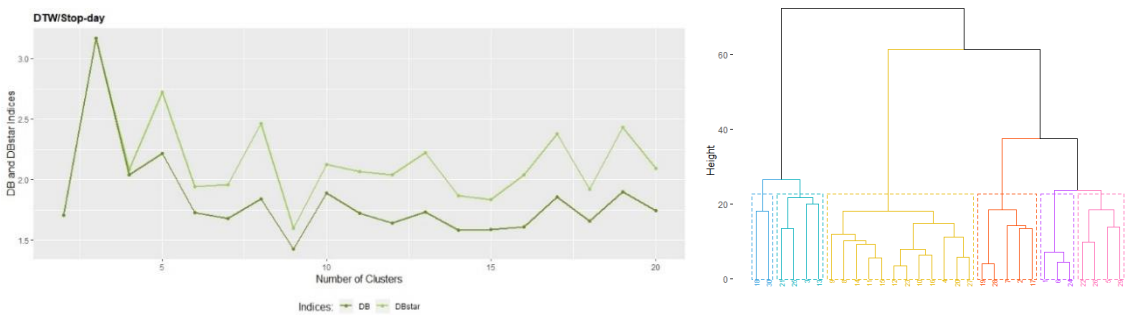
This chapter deals with the analysis of the stop-day vector. To do so, the procedure of choosing the optimal number of clusters for three methods of DTW, SBD, and ED will present and then the comparison of their resulted partitions based on three perspectives of minimum distance, external metrics and usage time will discuss. In this chapter, unlike for user-day vectors, we do not use fare-type distribution as a comparison criterion and for matching labels, we only use Fisher's exact test technique.

5.1 Number of Groups

We determined the optimal number of clusters by applying DTW, SBD, and ED methods considering the different clusters from 2 to 20, using DB and DB* indices and dendrogram. Figure 5-1, Figure 5-2, and Figure 5-3 show the number of these indices corresponding to each cluster number and the dendrogram drawn over 30 centroids.

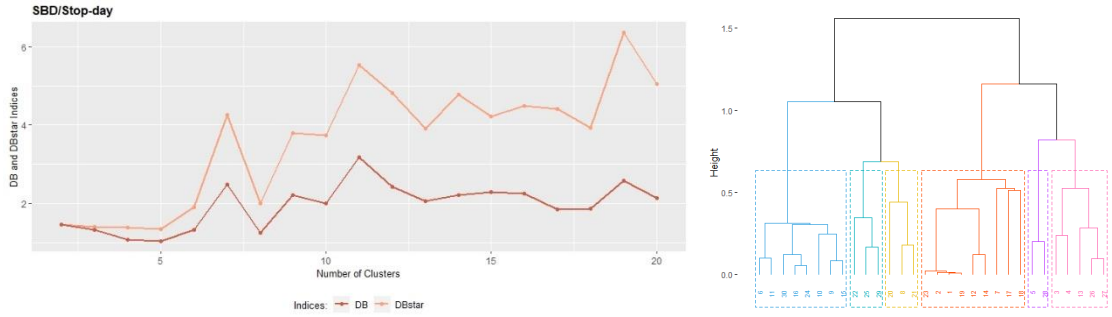
We can observe from Figure 5-1 (a), for DTW, the minimum amount of DB and DB* occurred in 9, 7, 6, 14, and 15. From the dendrogram shown in Figure 5-1 (b), it can be seen a considerable drop in error has occurred in 6 clusters while from 6 to 7, the decrease in error is not significant; however, again in 9, this elimination is noticeable.

Likewise, for SBD and ED, we followed the same procedure, and we decided to choose the optimal number of clusters equals 6 for all three methods.



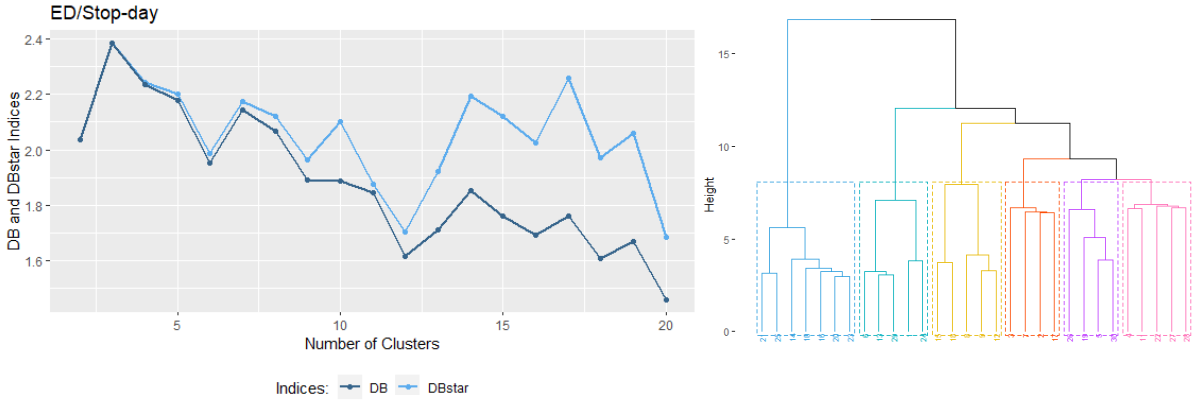
(a) (b)

Figure 5-1: Selection of the optimal number of clusters for stops under DTW by: (a) DB and DB*, (b) dendrogram over 30 centroids



(a) (b)

Figure 5-2: Selection of the optimal number of clusters for stops under SBD by: (a) DB and DB*, (b) dendrogram over 30 centroids



(a) (b)

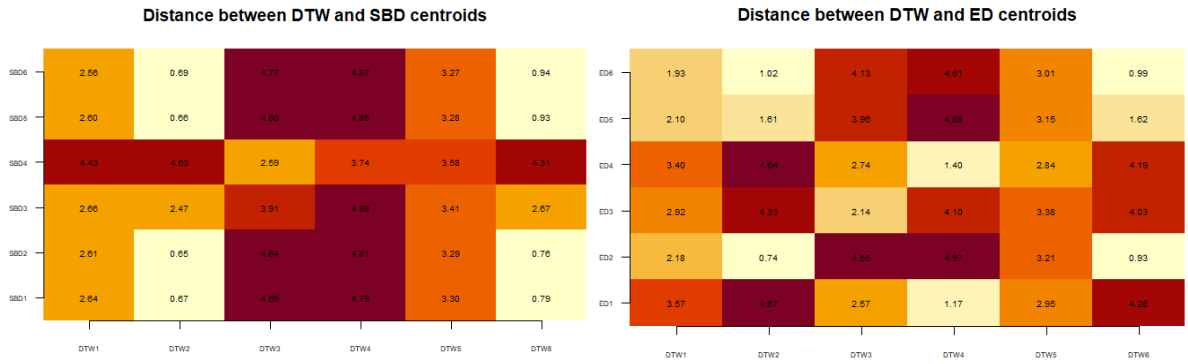
Figure 5-3: Selection of the optimal number of clusters for stops under ED by: (a) DB and DB*, (b) dendrogram over 30 centroids

5.2 Comparison of SBD and ED with DTW

In this section, we compare and discuss the resulting clusters of the application of DTW, SBD, and ED from three perspectives of distance, external measurements, and usage time.

5.2.1 Based on Distance Between Clusters

As already mentioned in Chapter 4, to calculate the distance between clusters, we used DTW distance measure between centroids shown in the heat maps (Figure 5-4). The closer the two centres are to each other, the lighter the colour.



(a) (b)

Figure 5-4: Distance between DTW clusters and:
(a) SBD clusters, (b) ED clusters

Based on Figure 5-4, the average distance between DTW and SBD is 2.998, and between DTW and ED is 3.023. The difference between these average distances is obviously insignificant, and because these distances are average, they do not provide a true comparison. Therefore, using other types of comparisons appears to be inevitable.

5.2.1.1 Label Matching

As stated in Chapter 4, we explored two methods of minimum distance and Fisher's exact test for obtaining the most similar clusters. In this chapter, however, we just employed Fisher's exact test method to match the labels of SBD and ED with DTW. The results are shown in Table 5-1 and Table 5-2.

Table 5-1: Matched labels for SBD clusters based on Fishers' exact test

Partition		**SBD**					
DTW	**Matched labels (Fisher's exact test)**	**1**	**2**	**3**	**4**	**5**	**6**
		6	4	1	5	2	3

Table 5-2: Matched labels for ED clusters based on Fisher's exact test

Partition		ED					
DTW	Matched labels (Fisher's exact test)	1	2	3	4	5	6
		4	2	5	3	6	1

5.2.2 Based on External Measurements

Regarding the last part, we calculate ARI and VI (for more detail) for original SBD and ED clusters labels. To prevent repetition, we did not display contingency tables before computing ARI in this case. It can be observed from Table 5-3, ED partitions have more agreement with DTW partitions than SBD with the more score in ARI (0.329) and lower score in VI (0.868).

Table 5-3: External measures for SBD and ED

External measures	SBD	ED
ARI	0.076	0.329
VI	1.232	0.868

5.2.3 Based on Usage Time

The cluster patterns depicted in Figure 5-5, Figure 5-6, and Figure 5-7, were obtained by applying DTW, SBD, and ED to stop-day vectors. We used the same colours for those clusters of SBD and ED that were matched with DTW according to Fisher's exact test for more clarity in comparison. **Error! Reference source not found.**8 also shows the portions of each method clusters.

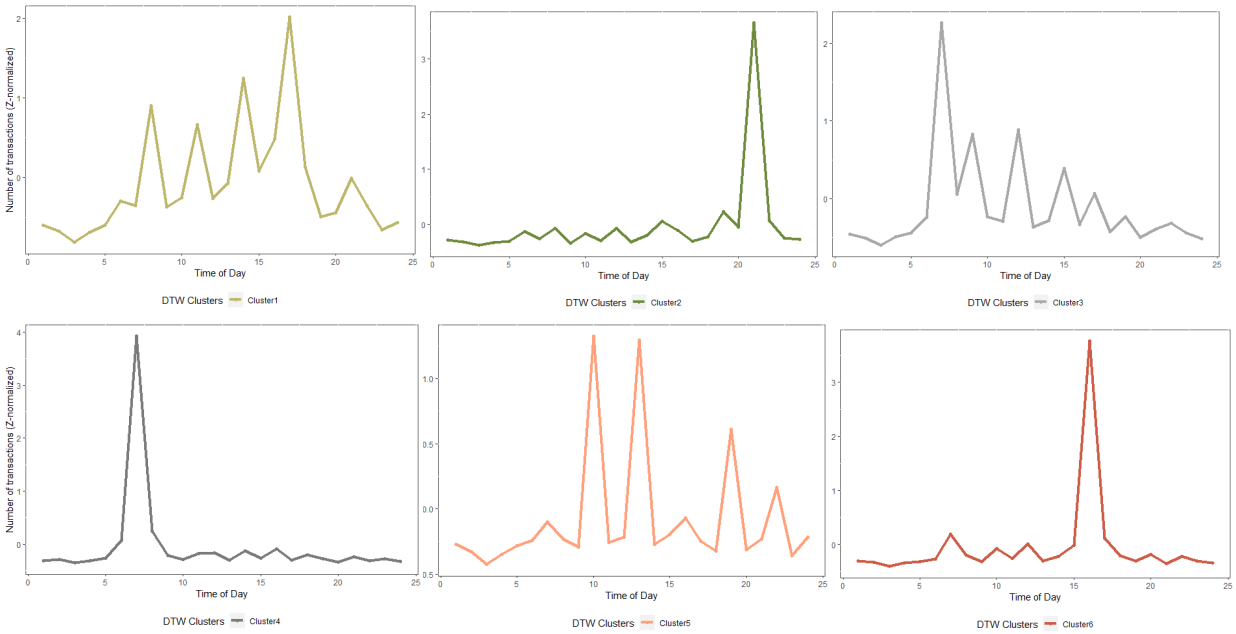


Figure 5-5: DTW stop clusters' patterns

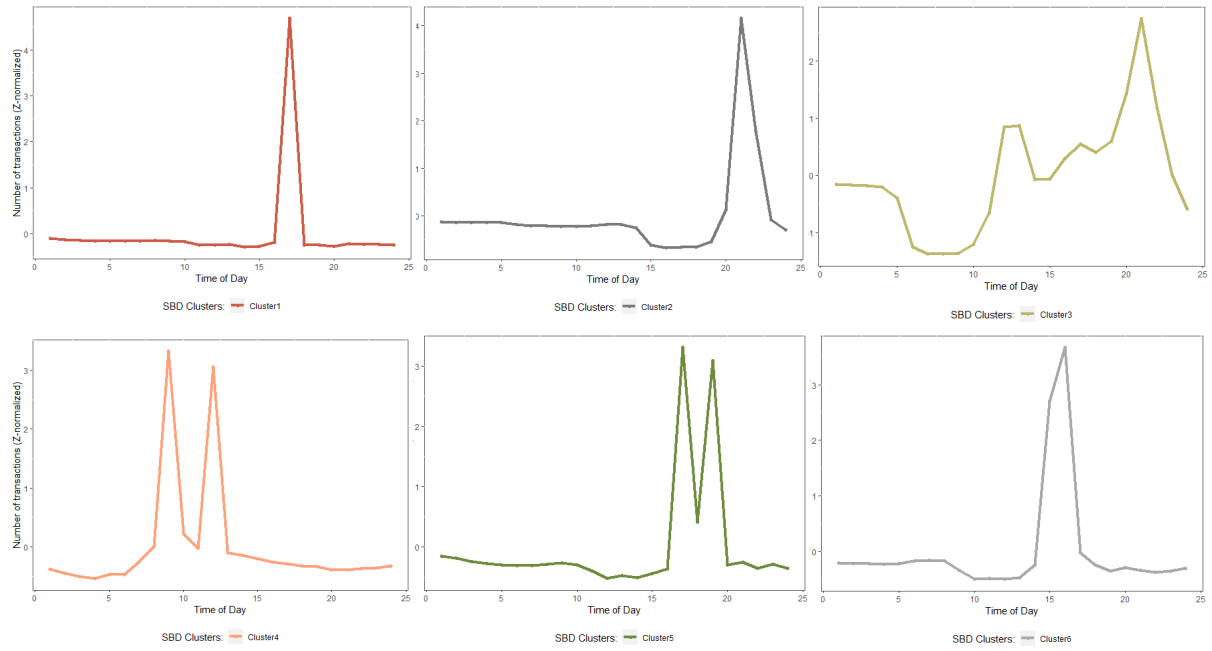


Figure 5-6: SBD stop clusters' patterns

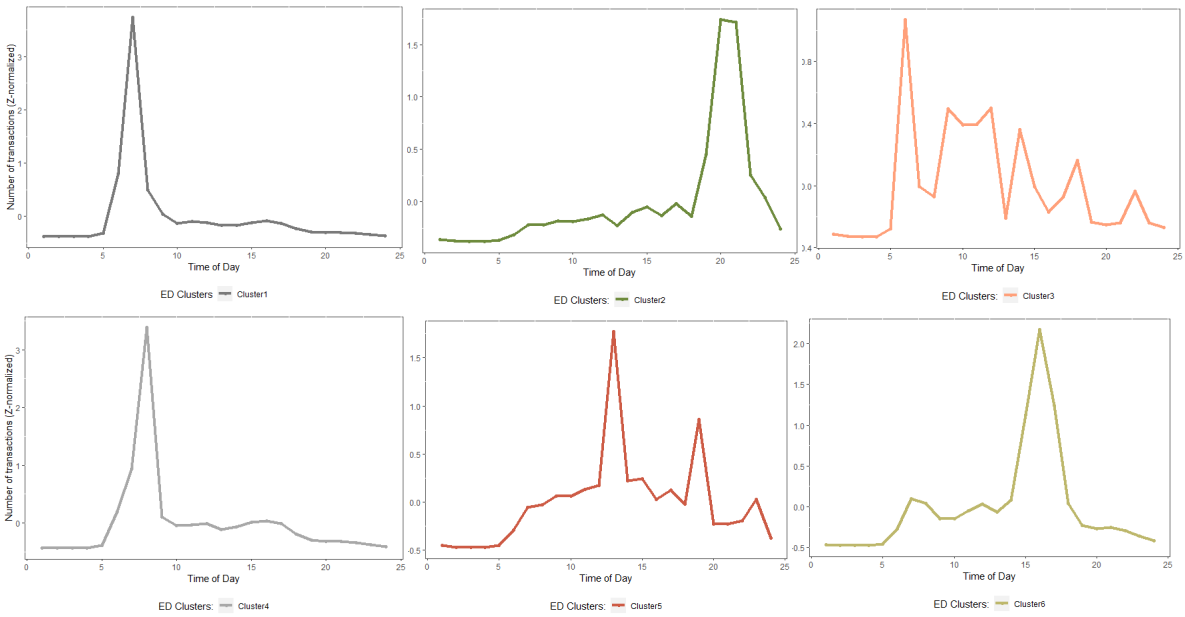
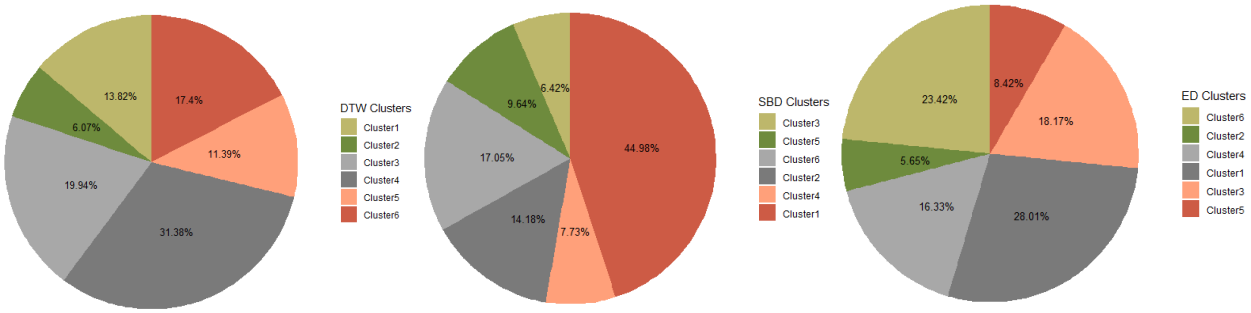


Figure 5-7: ED stops clusters' portions



(a) (b) (c)

Figure 5-8: Clusters' portions: (a) DTW, (b) SBD, (c) ED

According to what we observed for DTW patterns in Figure 5-5 and the portions in Figure 5-8, these patterns have been characterised in four main groups as follows. SBD And ED clusters have been also grouped accordingly.

1. Stops with regular commuters: DTW cluster 5 containing 11.39 percent of all stops has the pattern of usage mostly in the morning and afternoon. Its matched cluster in SBD, cluster 4, does not have a regular pattern but rather it shows most usage with two peaks both in the morning with the percentage of 7.73, less than DTW. Besides, cluster 3 in ED which is the

matched one for DTW cluster 5, has the pattern of all-day-long rather than regular. Considering ED cluster 5 in this group with 8.42% is more meaningful.

2. Stops with all-day-long commuters: In DTW clusters 1 and 3 has this pattern with the total of 33.76% of all stops. Their corresponding clusters in SBD are clusters 3 and 6. While according to their patterns, only cluster 3 can be included in this group with 6.42%. ED clusters 4 and 6 are the matched one with DTW clusters, while only cluster 6 can be considered in this category based on its pattern. The other ED cluster according to the pattern is cluster 3. Therefore, ED clusters 3 and 6 with the total of 41.59% of stops are in this group.
3. Stops with early-bird commuters: For DTW cluster 4 has this pattern with 31.38%. Its matched one in SBD is cluster 2 which has the opposite pattern of this group and should be considered in late commuters. Cluster 1 in ED and also cluster 4 are among this category with the total percentage of 44.34.
4. Stops with late commuters: This category consists of DTW clusters 2 and 6 with 23.47%. Their matched clusters in SBD are 5 and 1, but based on the patterns, clusters 2 and 6 are also among them. So, the total percentage of 85.86 of all stops, have this pattern based on SBD. For ED, although clusters 2 and 5 are the matched ones, cluster 2 can be considered in this category with 5.65%.

For better understanding the difference, we plotted the distribution of these four categories for all three methods per day over one month in Figure 5-9, Figure 5-10, and Figure 5-11.

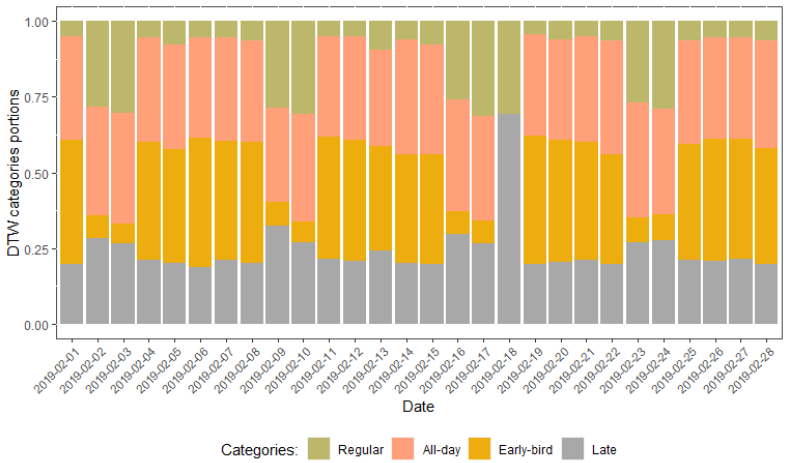


Figure 5-9: Distribution of DTW categories over one month

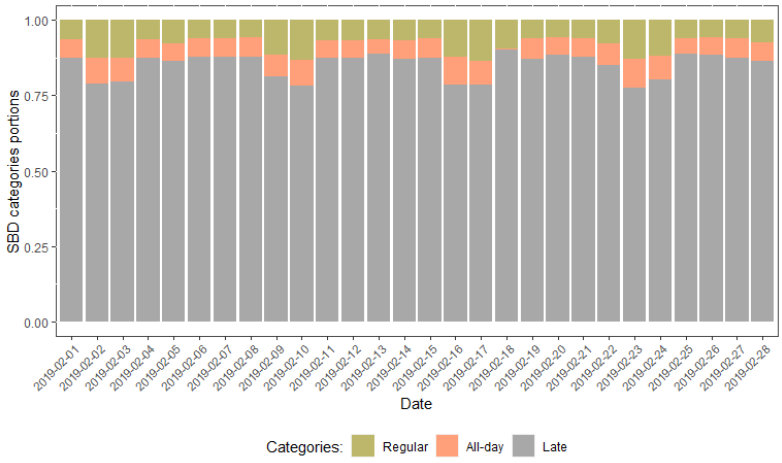


Figure 5-10: Distribution of SBD categories over one month

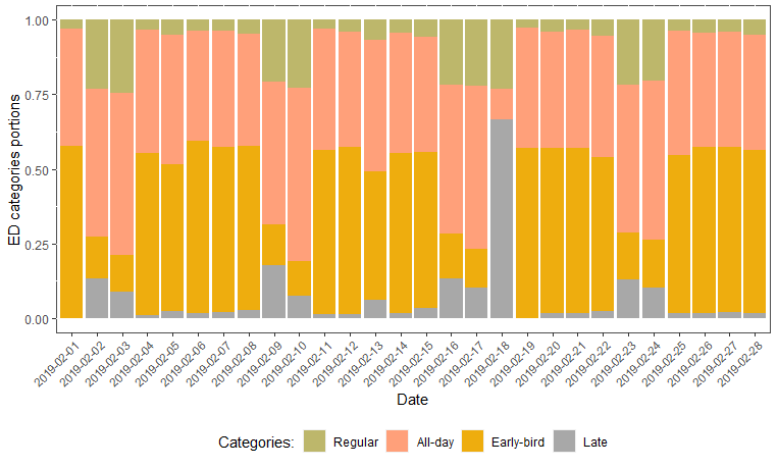


Figure 5-11: Distribution of ED categories over one month

From these figures, it can be noted that the most significant difference between DTW and SBD occurred in the grouping of early birds and late commuters. In a way, late commuters are a dominant group and there is no stop with the pattern of early birds based on SBD. This is in line with what we concluded in Chapter 4 from SBD method and its ignorance of time-shifting.

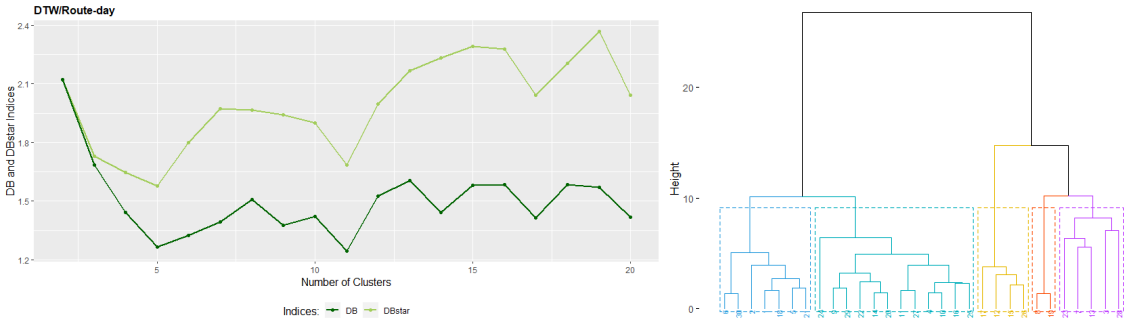
In spite of the fact that, ED outperformed SBD in the case of stop segmentation, and its distribution in four categories is more similar to DTW than SBD, its differences in all-day-long, early bird, and late commuters with DTW is noticeable.

CHAPTER 6 ROUTE-DAY ANALYSIS

This chapter aims to analyse, particularly, route-day vectors and compare the results of applying three methods of DTW, SBD, and ED. First, the procedure of choosing the optimal number of clusters, and then the comparison based on three perspectives of minimum distance, external metrics and usage time will discuss, likewise stop-day vectors in previous chapter.

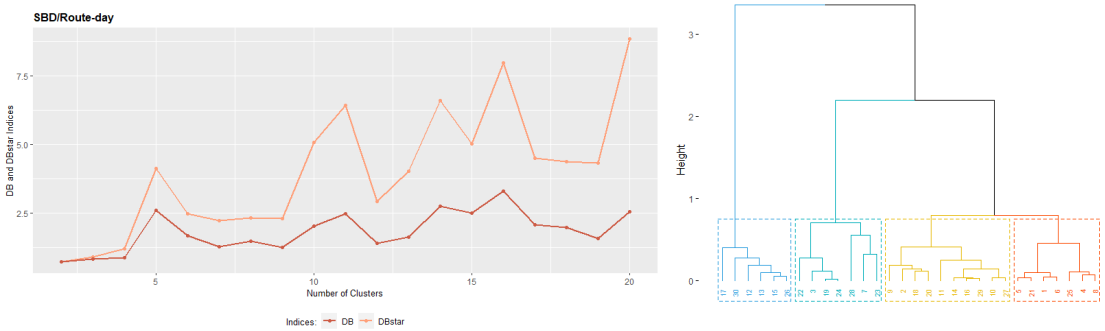
6.1 Number of Groups

As clarified in previous chapters, we used DB and DB* indices along with dendrogram to determine the prior number of clusters before applying our three methods. Figure 6-1(a), shows the minimum amounts of DB and DB* occur when the number of clusters is 3, 4, 5, and 11. In Figure 6-1(b) we see a considerable drop in error when dendrogram cuts in 5, while moving from 5 to 6 does not cause a significant decrease in the amount of error. Therefore, five groups seem good enough as the prior choice for the number of clusters. Likewise, for SBD and ED, we follow the same procedure, and the optimal number of 4 and 5, respectively, is chosen. Figure 6-2 and Figure 6-3 also confirm this selection.



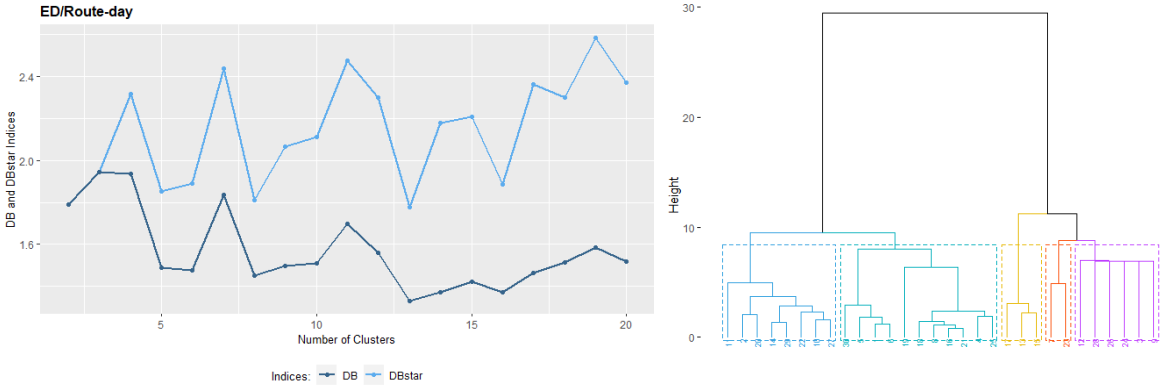
(a) (b)

Figure 6-1: Selection of the optimal number of clusters for routes under DTW by: (a) DB and DB*, (b) dendrogram over 30 centroids



(a) (b)

Figure 6-2: Selection of the optimal number of clusters for routes under SBD by: (a) DB and DB*, (b) dendrogram over 30 centroids



(a) (b)

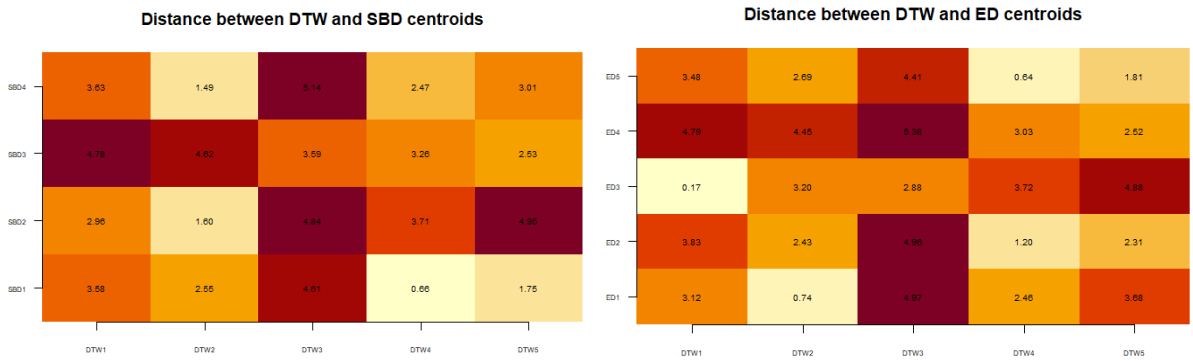
Figure 6-3: Selection of the optimal number of clusters for routes under ED by: (a) DB and DB*, (b) dendrogram over 30 centroids

6.2 Comparison of SBD and ED with DTW

In this section, we compare and evaluate the resulted clusters from the application of DTW, SBD, and ED based on three points of view. First, based on the distance between clusters centroids, and since k-means intrinsically produces random labels in each run for a more explicit comparison between three resulted partitions, we match SBD and ED clusters labels to DTW labels using Fisher's exact test. Second, calculating the amounts of ARI and VI, and then according to the time of using.

6.2.1 Based on Distance Between Clusters

Figure 6-4 shows DTW distance measure between (a) DTW clusters and SBD clusters centroids, and (b) DTW clusters and ED clusters centroids. According to these heat maps, the fewer two clusters are closer; the more colour is darker.



(a) (b)

Figure 6-4: Distance between DTW-clusters and:
(a) SBD-clusters, (b) ED-clusters

According to these heat maps, we computed the average distance for both DTW-SBD, and DTW-ED which are 3.29 and 3.11, respectively. However, this slight difference in their average distances is negligible, requiring a different method of comparison.

6.2.1.1 Label Matching

The resulted matched labels according to Fisher's exact test (for more detail see Appendix) is shown in , there is no suitable match for SBD-cluster 4 and ED cluster 2 corresponding to DTW clusters. This means that there is no non-random association between each DTW cluster with SBD and ED clusters.

Table 6-1 and Table 6-2, there is no suitable match for SBD-cluster 4 and ED cluster 2 corresponding to DTW clusters. This means that there is no non-random association between each DTW cluster with SBD and ED clusters.

Table 6-1: Matched SBD-clusters labels based on Fisher's exact test

Partition		SBD			
DTW	Matched labels (Fisher's exact test)	1	2	3	4
		4	2	1	-

Table 6-2: Matched ED-clusters labels based on Fisher's exact test

Partition		ED				
DTW	Matched labels (Fisher's exact test)	1	2	3	4	5
		2	-	3	5	4

6.2.2 Based on External Measurements

Calculating ARI and VI (for more detail) for original SBD and ED clusters labels illustrated in Table 6-3. From this table, we observe that SBD partitions compared to ED have more agreement with DTW partitions with the more score in ARI (0.499) and lower in VI (0.512).

Table 6-3: External measures for SBD and ED

External measures	SBD	ED
ARI	0.499	0.457
VI	0.512	0.603

6.2.3 Based on Usage Time

The resulted clusters' centroid patterns from applying DTW, SBD, and ED are plotted over 24 hours of a day in Figure 6-6, Figure 6-5, and Figure 6-7. We used the same colours for those clusters

of SBD and ED for more clarity, determined based on Fisher's exact test. For SBD cluster 4 and ED cluster 2 which did not match with DTW clusters we used different colour from DTW. Following these figures, we plotted the pie charts and discussed the characteristics of three resulted partitions for knowing the percentage occupied by each group.

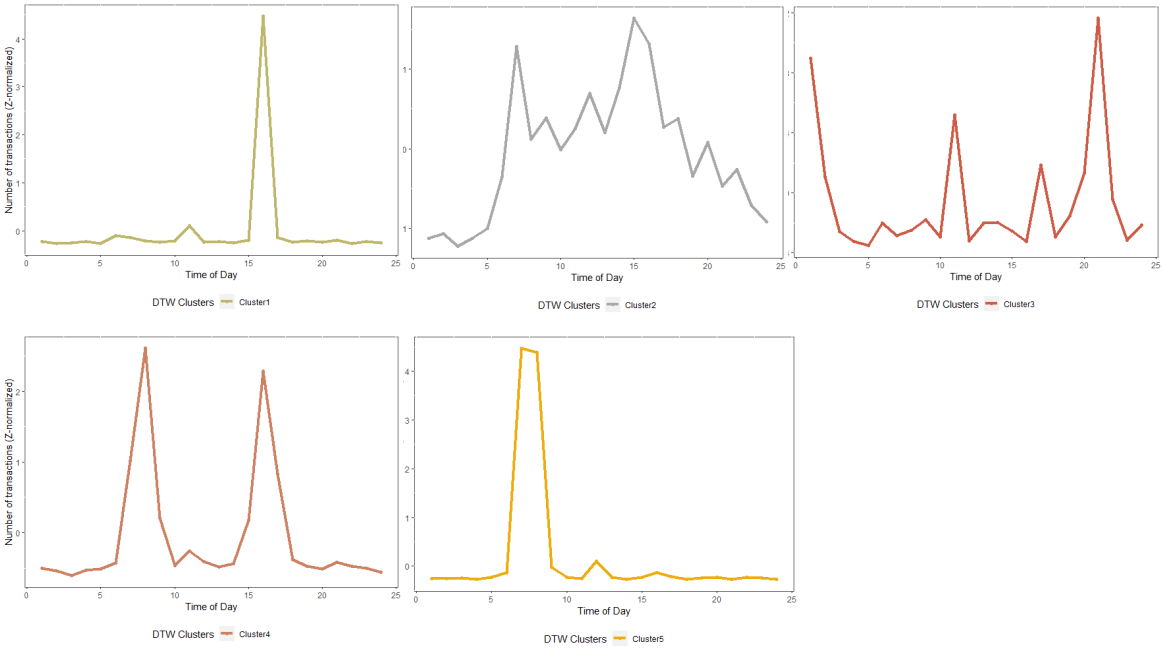


Figure 6-6: DTW route clusters' patterns

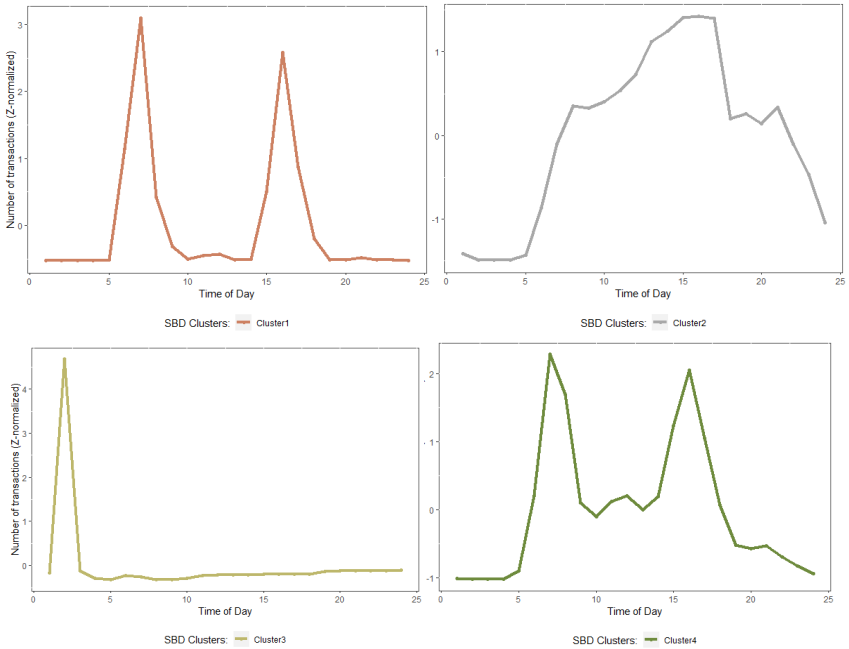


Figure 6-5: SBD route clusters' patterns

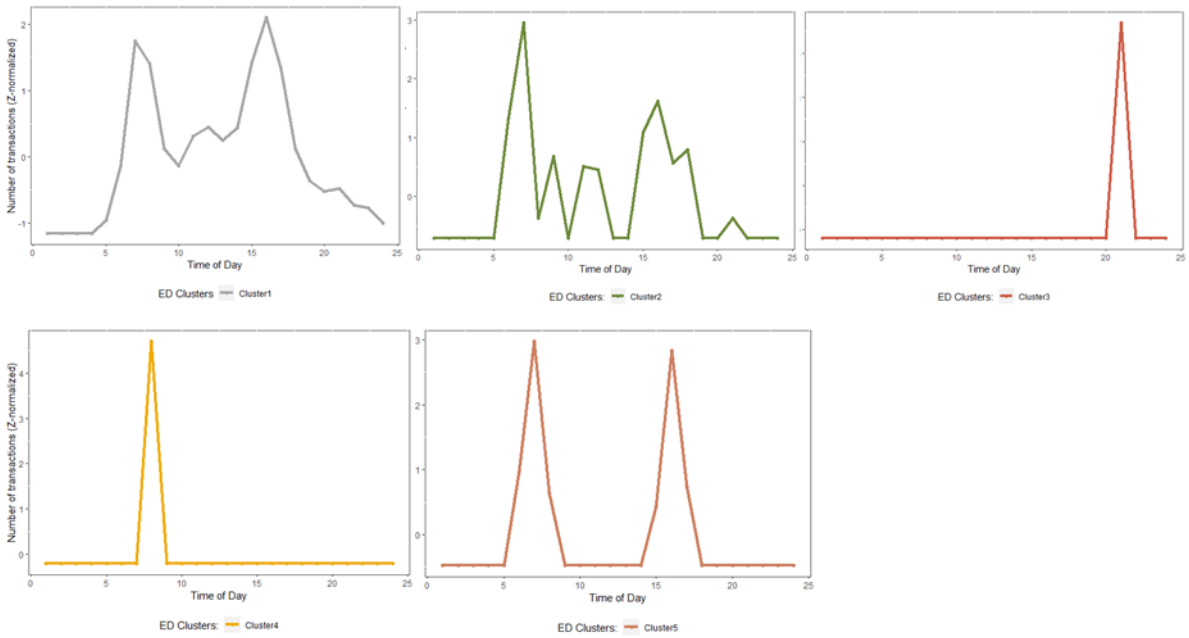
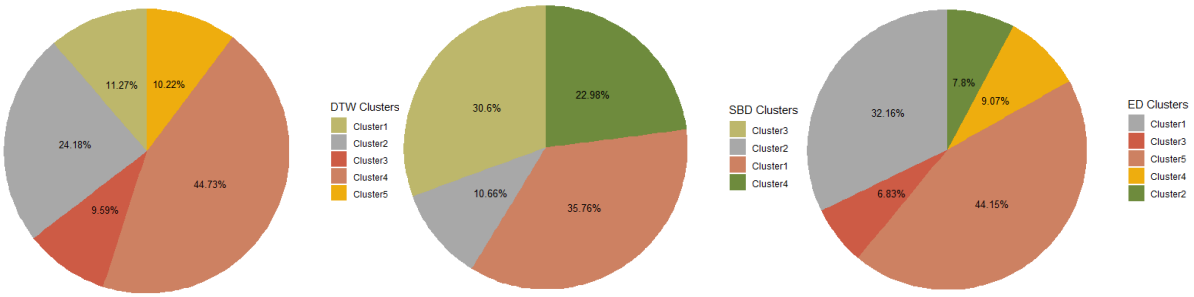


Figure 6-7: ED route clusters' patterns



(a) (b) (c)

Figure 6-8: Clusters' portions: (a) DTW, (b) SBD, (c) ED

According to the patterns shown in the figures, and considering the pie charts in Figure 6-8, we characterised the route clusters in three groups as follows:

1. Routes with regular commuters: DTW cluster 4 has this pattern of two peaks in the morning and afternoon which occupies 44.73% of all the routes. Its matched in SBD, cluster 1, is also among this group with the percentage of 35.76%. ED assigned 44.15% of routes to this group by the creation of cluster 5.

2. Routes with all-day-long commuters: DTW created 2 and 3 clusters with this pattern and assigned 33.77% of the routes to them. Based on SBD, cluster 2 which is also the matched one with DTW cluster 2, and cluster 4 based on its pattern are among this group with 33.64%. ED with the creation of clusters 1 and 2 assigned 39.96% of routes to this pattern.
3. Routes with early-bird commuters: based on DTW cluster 5 with 10.22%, based on SBD cluster 3 with 30.6%, and based on ED cluster 4 with 9.07% have this pattern.
4. Routes with late commuters: cluster 1 with 11.27% based on DTW, cluster 3 with 6.83% based on ED are with this pattern and in SBD partitions no cluster identified with this pattern.

We also depicted the distribution of the four route groups in Figure 6-9, Figure 6-11, and Figure 6-10 for more clarity. As concluded for user-day and stop-day vectors, observing the result of SBD for route-day vector, it can be claimed again when it comes to distinguishing a significant shift in time such as what we are faced in the creation of early bird and late commuters; SBD treats these two different patterns similarly. In other words, for instance, here for the early-bird commuters, SBD assigned 30% of the routes to this group which might also contain routes with the late commuter pattern.

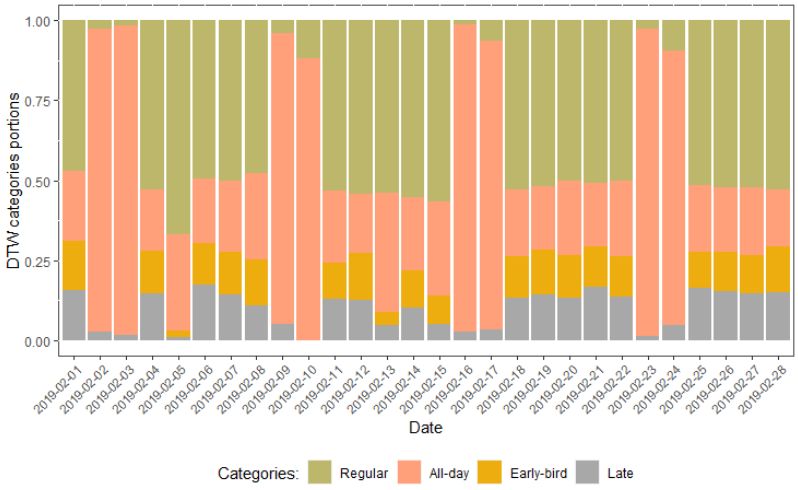


Figure 6-9: Distribution of DTW categories over one month

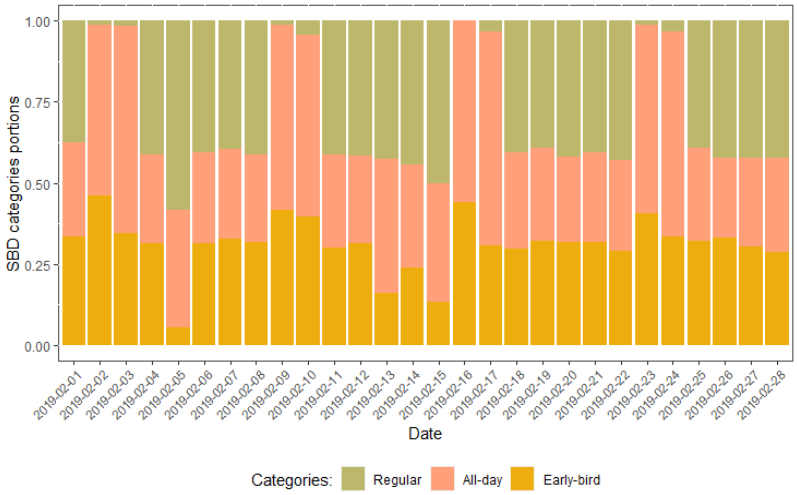


Figure 6-11: Distribution of SBD categories over one month

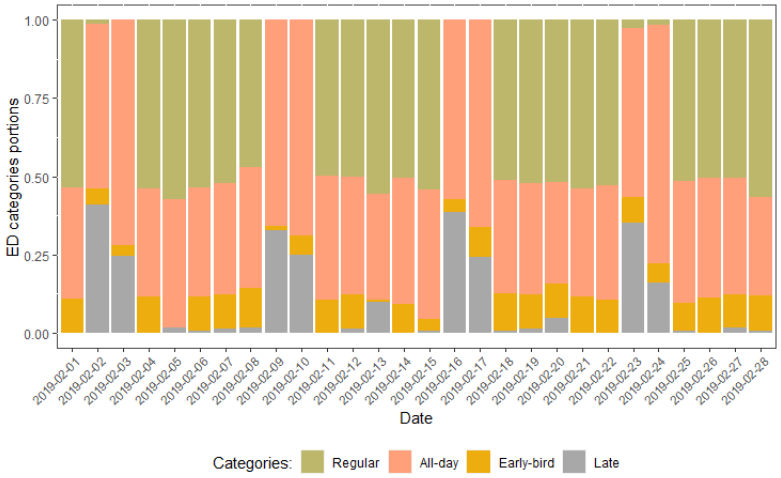


Figure 6-10: Distribution of ED categories over one month

CHAPTER 7 CONCLUSION AND RECOMMENDATIONS

This study aimed to investigate and test the results of applying a new technique of k-shape clustering with SBD on smart card data in public transit for the first time. Furthermore, the method's strengths and drawbacks were evaluated and compared to the most often used approaches; k-means clustering with DTW and Euclidean distances, in the field of transportation in previous studies. This chapter provides an overview of the three techniques and their outcomes to highlight the contributions made by this thesis. Some limitations are also stated, as well as suggestions for improvements and future explorations.

7.1 Contributions

Knowing what type of data, we are dealing with while studying smart card data in public transit provides for a more relevant and accurate results and analysis. Since the transactions produced by smart cards have the characteristics of time-series, it is critical to choose a method that can handle this type of data well. To put it in detail, when it comes to time-series clustering problems, selecting a good distance measure which is suitable with the specific variations inherent in sequences is as important as the algorithm itself. However, in the field of transportation these variations and distortions of time-series data in segmentation process has received less attention resulting in selection of Euclidean and Manhattan distances which both ignore the characteristics of sequences in their calculation.

In this study, we used k-shape clustering with the SBD method, which works well in time-series comparison, to segment smart card data in public transportation. Moreover, to reveal this method benefits and disadvantages, we compared it with the one of the most suitable and popular distance measures for time-series comparison; DTW distance measure with k-means clustering. In addition, the most commonly used distance measure for clustering smart card data in the transportation sector, Euclidean distance with k-means clustering, was used to highlight the methods differences even more. Therefore, in one side we had a fast method of SBD which considers mostly the shape of sequences in comparison and ignores their differences in time shifting. In other side, the fast method of ED which considers shifting in its calculation and might result in a big difference for two sequences even if they have the same shape. As a ground truth, we have the slow method of DTW which considers both the shape and the shift of time-series while it can be constrained for

the maximum shift. This comprehensive examination of three methodologies, each with its own set of features, can help to bridge the gap in the literature and pave the way for smart card data analysis in transportation research, allowing selecting a method that is more compatible with the data's specific characteristics.

In the comparison section, we looked at two statistical points of view: the distance between cluster centroids and external measures, as well as patterns in usage time and fare types in detail. This provided us with a better understanding of how the techniques worked and allowed us to compare them in greater depth rather than only on the basis of the indices produced. For instance, despite ED had better agreement with DTW outcomes than SBD in terms of minimum average distance between cluster centroids and VI index in user-day analysis, it performed worse than SBD in recognising well-defined patterns.

Furthermore, not only did we not confine our research to a single technique, but we also did not limit it to a single sort of object. We applied our three methods on three types of daily vectors to segment users, stops, and routes providing us more opportunity to evaluate how our approaches behaved as the vectors changed. SBD based on its characteristics, for example, should perform better where there is less shifting, such as in stop and route segmentation. It met our route expectations. However, the results in stop clustering were not satisfactory.

In conclusion, there is not the best method which can perform well in every situation. However, we are more likely to get more relevant results if we understand the type of data and its distortions while considering the study's goal. When there is a time constraint and a large dataset, for example, DTW, despite its high performance, cannot be chosen due to its time complexity. When the goal is to recognise the shape of patterns and the temporal shift is minor or not important, SBD outperforms the other approaches and is incredibly fast with large datasets. Even though ED it is a quick approach, and also performs well, but it is not a wise option for time-series comparison.

7.2 Limitations

Each research project has its own set of limitations, and our study is no exception. The first limitation in this thesis is regarding to the data and the second one arises from the methodology.

In terms of data, we only had access to the boarding times, consequently our research was restricted to one kind of vector, though if we had access to alighting time, our comparison would be more precise.

In the preprocessing part, for creating user-day vectors, we transferred several transactions into trips based on RTC rules of 90 minutes and the same bus line, whereas some of them might consist of the return trip in the same line within 90 minutes which should have been considered as two different trips as also pointed out in the study of Deschaintres et al. (2019). Another important factor to consider is the limitations in creating vectors. We used 24 hours of the day for separating the period of usage while different spaces of the vectors could lead to different outcomes and conclusions.

From the methodological point of view, there is no prior information as the actual clusters in the real world. Here we used the results of k-means clustering with DTW distance measure as the ground truth to compare with the novel method of k-shape clustering with SBD and k-means clustering with Euclidean distance for revealing the reliability of their results in our case; nevertheless, the DTW method has its own imperfections in the clustering process.

Moreover, k-means and k-shape clustering have the drawback of requiring the number of clusters to be determined prior to application. There are various strategies for doing so, but there is no one-size-fits-all solution for every scenario. As a result, different procedures must be used, and the findings must be integrated. In addition, we only examined the results of DTW while the parameter window equals to 1 which could generate different portions and different comparison results if it changed to another amount.

Furthermore, while comparing vectors, we only took into account differences based on a distance measure, neglecting the locations and other local environmental factors such as residential, industrial, or commercial areas, population, and ageing index, etc.

7.3 Perspectives

Introducing k-shape clustering with SBD to the field of smart-card data clustering in the transportation sector opened up several possibilities for future research with different goals. This is a precise and efficient time-series clustering approach that works very fast even with big datasets.

For instance, in fluctuation analysis, it can produce highly competitive results in the detection of very well-defined patterns. Furthermore, in studies such as ours, where time-shifting is required for comparison, it is necessary to investigate a way to modify this approach to be constrained for time-shifting or perhaps to combine it with another methodology.

In this thesis, based on the work of Paparrizos and Gravano (2017), we used coefficient normalisation of CCD for obtaining SBD, while future studies can explore other way of normalisations such as biased estimator, NCC_b, or unbiased estimator, NCC_u, to gain different characteristics as a time-series distance measure. According to Equation 2-6 for these normalisations we will have:

$NCC_b = \frac{CC_w(\vec{x},\vec{y})}{m}$, $NCC_u = \frac{CC_w(\vec{x},\vec{y})}{m-|w-m|}$

REFERENCES

Agard, B., Morency, C., & Trépanier, M. (2006). Mining Public Transport User Behaviour from Smart Card Data. *IFAC Proceedings Volumes, 39*(3), 399-404. doi:10.3182/20060517-3-fr-2903.00211

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition, 46*(1), 243-256. doi:10.1016/j.patcog.2012.07.021

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. s. M., & Perona, I. i. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition, 46*(1), 243-256. doi:10.1016/j.patcog.2012.07.021

Arbex, R., & Cunha, C. B. (2020). Estimating the influence of crowding and travel time variability on accessibility to jobs in a large public transport network using smart card big data. *Journal of Transport Geography, 85*. doi:10.1016/j.jtrangeo.2020.102671

Batista, G. E. A. P. A., Keogh, E. J., Tataw, O. M., & de Souza, V. M. A. (2013). CID: an efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery, 28*(3), 634-669. doi:10.1007/s10618-013-0312-3

Cavallaro, F., & Dianin, A. (2020). An innovative model to estimate the accessibility of a destination by public transport. *Transportation Research Part D: Transport and Environment, 80*. doi:10.1016/j.trd.2020.102256

Deschaintres, E., Morency, C., & Trépanier, M. (2019). Analyzing Transit User Behavior with 51 Weeks of Smart Card Data. *Transportation Research Record: Journal of the Transportation Research Board, 2673*(6), 33-45. doi:10.1177/0361198119834917

Egu, O., & Bonnel, P. (2020). Investigating day-to-day variability of transit usage on a multimonth scale with smart card data. A case study in Lyon. *Travel Behaviour and Society, 19*, 112-123. doi:10.1016/j.tbs.2019.12.003

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. *ACM SIGMOD Record, 23*(2), 419-429. doi:10.1145/191843.191925

Gan, Z., Yang, M., Feng, T., & Timmermans, H. (2018). Understanding urban mobility patterns from a spatiotemporal perspective: daily ridership profiles of metro stations. *Transportation, 47*(1), 315-336. doi:10.1007/s11116-018-9885-4

Ghaemi, M. S., Agard, B., Trépanier, M., & Partovi Nia, V. (2017). A visual segmentation method for temporal smart card data. *Transportmetrica A: Transport Science, 13*(5), 381-404. doi:10.1080/23249935.2016.1273273

Giorgino, T. (2009). Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *Journal of Statistical Software, 31*. doi:10.18637/jss.v031.i07

Han, J., & Kamber, M. (2006). *Data mining : concepts and techniques* (2nd ed.). Amsterdam, Pays-Bas: Elsevier : Morgan Kaufmann Publishers.

He, L., Agard, B., & Trépanier, M. (2018). A classification of public transit users with smart card data based on time series distance metrics and a hierarchical clustering method. *Transportmetrica A: Transport Science, 16*(1), 56-75. doi:10.1080/23249935.2018.1479722

He, L., & Trépanier, M. (2015). Estimating the Destination of Unlinked Trips in Transit Smart Card Fare Data. *Transportation Research Record: Journal of the Transportation Research Board, 2535*(1), 97-104. doi:10.3141/2535-11

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, 31*(8), 651-666. doi:10.1016/j.patrec.2009.09.011

Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and Information Systems, 7*(3), 358-386. doi:10.1007/s10115-004-0154-9

Kim, M.-K., Kim, S.-P., Heo, J., & Sohn, H.-G. (2017). Ridership patterns at subway stations of Seoul capital area and characteristics of station influence area. *KSCE Journal of Civil Engineering, 21*(3), 964-975. doi:10.1007/s12205-016-1099-8

Lee, S. G., & Hickman, M. (2013). Trip purpose inference using automated fare collection data. *Public Transport, 6*(1-2), 1-20. doi:10.1007/s12469-013-0077-5

Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis, 98*(5), 873-895. doi:10.1016/j.jmva.2006.11.013

Mohamad, I. B., & Usman, D. (2013). Standardization and Its Effects on K-Means Clustering Algorithm. *Research Journal of Applied Sciences, Engineering and Technology, 6*(17), 3299-3303. doi:10.19026/rjaset.6.3638

Moradi, M., & Trépanier, M. (2018). Assessing longitudinal stability of public transport users with smart card data. *Transportation Research Procedia*.

Morency, C., Trépanier, M., & Agard, B. (2007). Measuring transit use variability with smart-card data. *Transport Policy, 14*(3), 193-203. doi:10.1016/j.tranpol.2007.01.001

Morency, C., Trépanier, M., Frappier, A., & Bourdeau, J.-S. (2017). *Longitudinal Analysis of Bikesharing Usage in Montreal, Canada.* Paper presented at the 96th Annual Meeting of the Transportation Research Board, Washington, D.C.

Mori, U., Mendiburu, A., & Lozano, J. (2016). Distance Measures for Time Series in R: The TSdist Package. *The R Journal, 8*. doi:10.32614/RJ-2016-058

Nantes, A., Bhaskar, A., Miska, M., Chung, E., & Ngoduy, D. (2016). Real-time traffic state estimation in urban corridors from heterogeneous data. *Transportation Research Part C: Emerging Technologies, 66*, 99-118. doi:10.1016/j.trc.2015.07.005

Paparrizos, J. (2018). *Fast, scalable, and accurate algorithms for time-series analysis.* (Doctor of Philosophy), Columbia University, Retrieved from https://academiccommons.columbia.edu/doi/10.7916/D80K3S4B

Paparrizos, J., & Gravano, L. (2017). Fast and Accurate Time-Series Clustering. *ACM Transactions on Database Systems, 42*(2), 1-49. doi:10.1145/3044711

Park, J. Y., Kim, D.-J., & Lim, Y. (2008). Use of Smart Card Data to Define Public Transit Use in Seoul, South Korea. *Transportation Research Record: Journal of the Transportation Research Board, 2063*(1), 3-9. doi:10.3141/2063-01

Pelletier, M.-P., Trépanier, M., & Morency, C. (2011). Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies, 19*(4), 557-568. doi:10.1016/j.trc.2010.12.003

Petitjean, F., Ketterlin, A., & Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition, 44*(3), 678-693. doi:10.1016/j.patcog.2010.09.013

Rabbany, R., & Zaïane, O. R. (2015). Generalization of clustering agreements and distances for overlapping clusters and network communities. *Data Mining and Knowledge Discovery, 29*(5), 1458-1485. doi:10.1007/s10618-015-0426-x

Rakthanmanon, T., Keogh, E. J., Lonardi, S., & Evans, S. (2011). Time Series Epenthesis: Clustering Time Series Streams Requires Ignoring Some Data. *11th IEEE International Conference on Data Mining*, 547-556. doi:10.1109/ICDM.2011.146

Reades, J., Zhong, C., Manley, E. D., Milton, R., & Batty, M. (2016). Finding Pearls in London's Oysters. *Built Environment, 42*(3), 365-381. doi:10.2148/benv.42.3.365

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. D. F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PLoS One, 14*(1), e0210236. doi:10.1371/journal.pone.0210236

Santos, J., & Embrechts, M. (2009). *On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification* (Vol. 2009).

Sardá-Espinosa, A. (2018). *Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package*.

Seo, J., Cho, S.-H., Kim, D.-K., & Park, P. Y.-J. (2020). Analysis of overlapping origin–destination pairs between bus stations to enhance the efficiency of bus operations. *IET Intelligent Transport Systems, 14*(6), 545-553. doi:10.1049/iet-its.2019.0158

Trépanier, M., Tranchant, N., & Chapleau, R. (2007). Individual Trip Destination Estimation in a Transit Smart Card Automated Fare Collection System. *Journal of Intelligent Transportation Systems, 11*(1), 1-14. doi:10.1080/15472450601122256

Viallard, A., Trépanier, M., & Morency, C. (2019). Assessing the Evolution of Transit User Behavior from Smart Card Data. *Transportation Research Record: Journal of the Transportation Research Board, 2673*(4), 184-194. doi:10.1177/0361198119834561

Warren Liao, T. (2005). Clustering of time series data—a survey. *Pattern Recognition, 38*(11), 1857-1874. doi:10.1016/j.patcog.2005.01.025

Yap, M., Cats, O., & van Arem, B. (2018). Crowding valuation in urban tram and bus transportation based on smart card data. *Transportmetrica A: Transport Science, 16*(1), 23-42. doi:10.1080/23249935.2018.1537319

Zhang, T., Li, Y., Yang, H., Cui, C., Li, J., & Qiao, Q. (2018). Identifying primary public transit corridors using multi-source big transit data. *International Journal of Geographical Information Science, 34*(6), 1137-1161. doi:10.1080/13658816.2018.1554812

Zhao, M., Mason, L., & Wang, W. (2008). *Empirical study on human mobility for mobile wireless networks*. Paper presented at the IEEE Military Communications Conference, MILCOM 2008.

APPENDIX A RTC FARE-TYPES

Table A-1: RTC fare-types

	CodeProduit	Produit - Nom
1	TB2-G	billets général
2	TLM-E	L-P mensuel étudiant
3	TLM-G	L-P mensuel général
4	TLA-G	Abonnebus général
5	TB2-E	billets étudiant
6	TB2-N	billets aîné
7	TLM-N	L-P mensuel aîné
8	TLA-E	Abonnebus étudiant
9	T17SE	PASSE DE SESSION
10	TL5JG	LP 5 JOURS GENERAL
11	TL5JR	LP 5 JOURS REDUIT
12	TLMVG	L-P traversiers gén
13	T10HP	10BIL HORSPOINTE 65+
14	TLA-N	Abonnebus aîné
15	200	Employé/Retraité-OPT
16	QLMRG	LP métropolitain gén
17	QLMRE	LP métropolitain étu
18	LPM-R	L-P mensuel régulier
19	LPM-P	LP mensuel privilège
20	T11JR	11 JRS FEST ÉTÉ
21	QLARG	Abonnebus mét gén
22	ABRTC	RTC Abonnement
23	T10FG	10 billets Fest ÉTÉ
24	SCOLA	Scolaire Annuel
25	TPJ21	Promotion 21T
26	LPABR	AbonneBus rég
27	TLM18	L-P 18 et - SOIR WE
28	T11JE	11 JR FESTIVAL ETUDI
29	QLMRN	LP métropolitain 65+
30	ADOBU	ADOBUS 7 À 17 ANS
31	TLS-P	L-P employé/ret
32	CBMRE	LP mét MRC CB étu
33	TLS-Q	L-P employé STQ
34	LPM-A	LP mensuel Ainé
35	CBMRG	LP mét MRC CB gén
36	QLARE	Abonnebus mét étu
37	FB4-G	FORM 4 billets gén.
38	LPABP	AbonneBus pri
39	T11JN	11 JR FESTIVAL AINE
40	TB4-G	billets général
41	QLARN	Abonnebus mét 65+
42	TB4-N	billets aîné
43	201	Employé/Retraité-AMT
44	TB4-E	billets étudiant
45	66	OPUS-TRAM2-Carnet-O
46	C10FE	10 billets Fest. ÉTÉ
47	C11JR	11 JRS FEST ÉTÉ
48	C1502	2 Bill 150 gratuit
49	C2402	2 billets promo 240J
50	C5JRS	CPO 5 JOURS GÉNÉRAL
51	CB10S	GROUPE 10
52	CB12G	billet général
53	CB1-G	billet général
54	CB2-C	2 billets courtoisie
55	CB2-G	billet général
56	CB3-G	billet général
57	CB4-E	billets étudiant
58	CB4-G	billet général
59	CB6-G	billets général
60	CB8-G	billet général
61	CBMRN	LP mét MRC CB 65+
62	CBP-2	2 billets PROMO
63	CBP-4	4 billets PROMO
64	CEG-2	2 Bill Essai gratuit
65	CL2JG	L-P deux jours
66	CL5JG	LP 5 JOURS CPO
67	CL7JG	LP 7 jours general
68	CLJ-G	L-P un jour
69	CLM-E	L-P mensuel étu. CPO
70	CLPU	L-P UNIVER. UL CPO
71	CPJ21	Promotion 21C
72	CSP3J	Spécial 3 jours
73	CSP7J	Spécial 7 jours
74	CWE	CPO WEEKEND ILLIMITÉ
75	FCB12	FORM 12 billets gén.
76	FLM-E	FORM L-P men. étu.
77	IOMRE	LP mét MRC IO étu
78	IOMRG	LP mét MRC IO gén
79	IOMRN	LP mét MRC IO 65+
80	JCMRE	LP mét MRC JC étu
81	JCMRG	LP mét MRC JC gén
82	JCMRN	LP mét MRC JC 65+
83	KBMRE	CPO met MRC CB étu
84	KLMRE	L-P metropol étu CPO
85	LPABA	AbonneBus Ainé
86	LPUUL	L-P UNIVERSITAIRE UL
87	T121	LP LIGNES 1 ET 21
88	T5JRE	LP 5 JOURS ÉTUDIA
89	T5JRN	LP 5 JOURS AINÉS
90	T5JRS	LP 5 JOURS GÉNÉRAL
91	T65WE	65 WEEKEND ILLIMITÉ
92	TEWE	E WEEKEND ILLIMITÉ
93	TL7JE	LP 7 jours etudiant
94	TL7JG	LP 7 jours general
95	TL7JN	LP 7 jours aine
96	TLANH	AB AINÉ HORS-POINTE
97	TLAVI	L-P traversiers int
98	TLJ-G	L-P un jour OPUS
99	TLMNH	L-P AINÉ HORS-POINTE
100	TWEG	TG WEEKEND ILLIMITÉ

APPENDIX B FISHER'S EXACT TEST

Fisher's exact test is a statistical significance test used for contingency tables analysis . If we have the simplified contingency as bellows, Table B-1, based on the value of p can be computed as following:

Table B-1: Contingency table

Partition	V		Sum
U	Pair in same group	Pair in different group	
Pair in same group	a	b	$a + b$
Pair in different group	c	d	$c + d$
Sum	$a + c$	$b + d$	$a + b + c + d = n$

$$p = \frac{(a+b)!\,(c+d)!\,(a+c)!\,(b+d)!}{a!\,b!\,c!\,d!\,n!}$$

Where p is the probability of non-random association between V and U.

APPENDIX C RTC NETWORK MAP

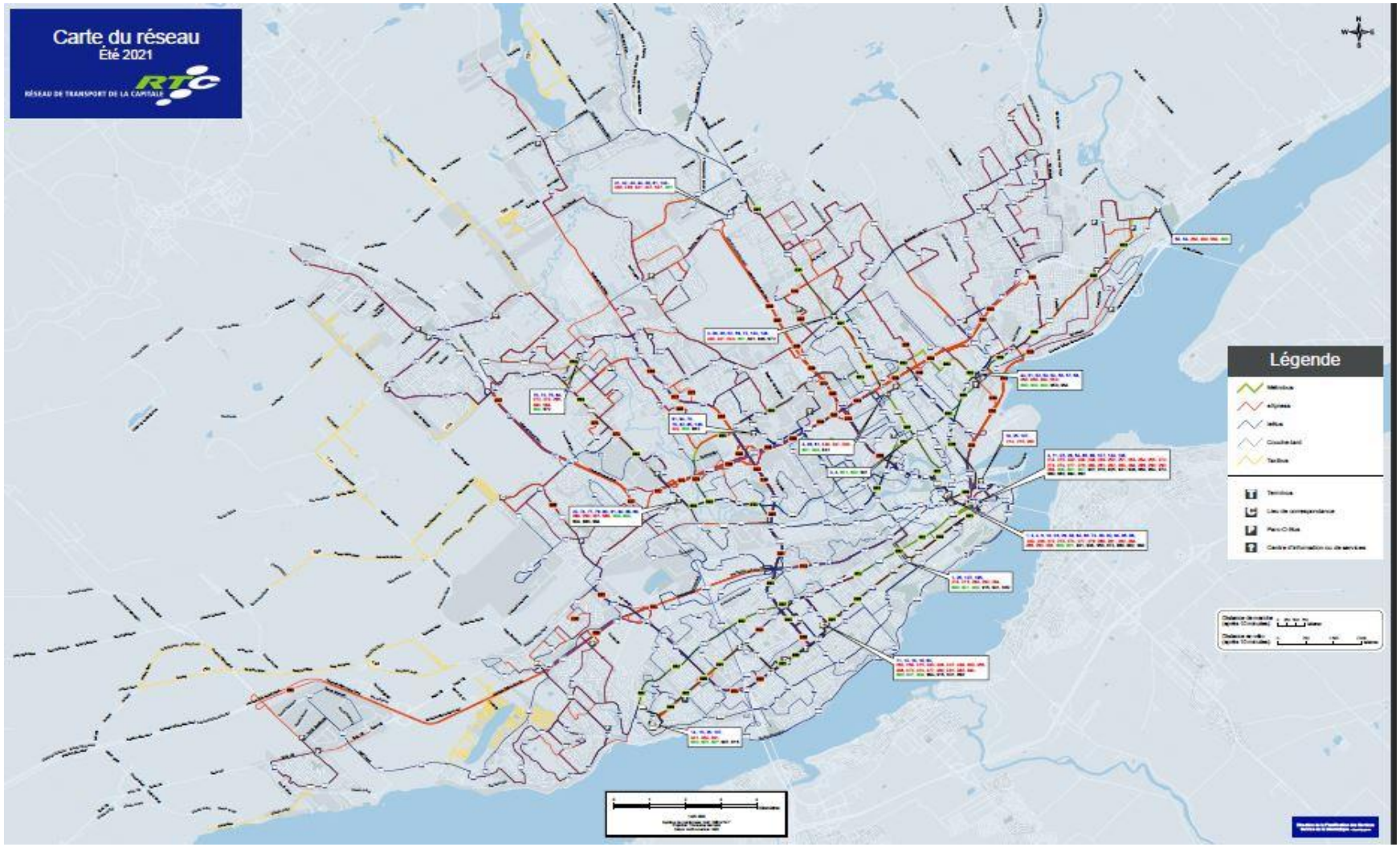


Figure C-1: RTC network map