CIRRELT
1971 – 2021

CIRRELT-2021-40

Comparing Clustering Methods in Recognising of Temporal Travel Pattern of Public Transport Users

Zohreh Vaezi
Martin Trépanier

October 2021

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Comparing Clustering Methods in Recognising of Temporal Travel Pattern of Public Transport Users

Zohreh Vaezi*, Martin Trépanier

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Mathematical and Industrial Enginnering, Polytechnique Montréal

Abstract. Segmenting passengers with the most similar behaviours enable transit agencies to establish their strategies based on groups' needs rather than individuals', resulting in more efficient and effective services. The smart card system has made this investigation of travel patterns more possible by facilitating data collection. Because smart card data has the characteristics of time-series, it is crucial to develop the most suitable clustering method and a proper distance measure to handle these sequences. However, this has been largely overlooked in previous research. In this paper, we tried to test and adapt a novel k-shape clustering method with Shape-Based Distance (SBD) measure on smart card data as the first attempt. We developed a comparison framework between the results of this method with k-means clustering along with two most used distance measures: Euclidean and Dynamic Time Warping (DTW). To enrich this framework, we not only performed user segmentation but also stop and route to see whether the comparison remains the same when the type of vectors changes. This study confirmed that Euclidean distance, despite its popularity, does not work well in recognition of well-defined patterns for time-series data. In contrast, k-shape clustering works well in this regard. Although k-shape does not consider time-shifting in comparison, it yielded competitive partitions in creating route clusters when this shift was minor.

Keywords: smart card, time-series, k-shape clustering, prototyping.

Acknowledgements. The authors wish to acknowledge the collaboration of the Réseau de Transport de la Capitale for providing data and the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the group of THALES, the CORTEX fund, and PROMPT.

---

* Corresponding author: zohreh.vaezi@polymtl.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2021

INTRODUCTION

Understanding mobility patterns from smart card data not only helps day-to-day operations but also long-term planning for the transportation system, including route design, urban planning, location-based services, network growth, marketing, etc. [1, 2]. Regarding the fact that smart card data contains detailed information such as transaction time and location, route direction, and the card type, we can divide this information and its subsequent analysis into three primary categories: temporal, spatial and spatiotemporal [3].

The temporal data gathered by smart cards has time-series characteristics. Thus, for having more accurate segmentation results, choosing a suitable clustering method and a proper distance measure to handle this type of data is crucial. However, most employed metrics in the literature, such as Euclidean distance, are incapable of dealing with the dynamic characteristics of temporal vectors.

In this paper, we decided to test and adapt a novel k-shape clustering technique with Shape-Based Distance (SBD) measure, recently proposed by Paparrizos and Gravano [4], on our public transit smart card dataset for the first time. The dataset of this study is from one month of February 2019 provided by *Réseau de transport de la Capitale* (RTC), a transit authority offering regular public transit services in the greater Quebec City area.

We develop a comparative framework among the results of this novel method for time-series clustering and two other popular approaches on the same dataset to explore and highlight their advantages and drawbacks. Since the principles of k-shape clustering are based on k-means, we use k-means clustering once with Dynamic Time Warping (DTW) distance and then with Euclidean distance (ED). As a result, we will be able to compare the performance of three distance measures on time-series comparison.

Since the type of vectors also has a significant impact on the outcomes, we employ three different types of profiles to compare the performances of the methods. Therefore, card-day (user-day), stop-day, and route-day vectors based on the daily boarding time are created as input data to see whether the comparison of our methods partitions remains the same when the vectors change.

The following is a breakdown of the paper's structure. The literature review begins by providing some background on smart card data. Then a description of the dataset and the proposed methodology are presented. The results for only one type of vector (user-day) are discussed in detail, and we only confine to preset other vector results on the conclusion part. Finally, the paper is finished by the contributions and limitations of this study and a possible follow-up for future research.

LITERATURE REVIEW

Smart Card Data in Public Transit

In addition to the primary goal of using smart cards as a fare collection system, Pelletier, et al. [1] categorised the applications of using this data into three groups of operational, tactical, and strategic levels. For instance, operational research has been conducted with the objective of improving the daily performance of the system. Estimation of accessibility [5, 6], and crowding valuation [7] 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. [8] analysed overlapping origin-destination pairs between bus stations resulting to help enhance the efficiency of transit operation. Demand estimation and forecasting by identifying public transit corridors [9], and costumer behaviour analysis [10] are among strategic studies improving long-term planning.

Most research has focused on user segmentation and there is less works have been undertaken aiming station segmentation [11, 12] or routes grouping. In this study, we analyse temporal patterns of travel for users, and for stations and routes.

Time-Series Clustering Algorithms

Clustering is the most popular data mining method in analysing behaviour by grouping data points in such a way that there is the most similarity within data points in the same group and the most dissimilarity with the members of other groups [13]. Various clustering algorithms have been developed for static data in the literature, and there is no direct method for times-series data. A time-series is a sequence in which every element is resulted from recording a measurement varying by the time [14]. Thus, for clustering time-series data, we must either convert it to static type and use the existing methods, or modify the method to deal with this data [15].

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 [14]. The quality of the resulted clusters not only depends on the choice of the method, but also selecting a compatible distance measure. Besides, when it comes to temporal data because of the sequential characteristics, there are also some distortions and invariances such as shifting, scaling, and translation [4] which either need to be satisfied with the choice of proper distance measure or to be removed before applying clustering [16]. Among aforementioned methods, k-means is more efficient and can scale linearly with the size of datasets. It is known as one of the most influential data mining algorithms of all time [4].

Time-Series Clustering in Public Transit

The most studies in transportation area are aimed to recognise the travel pattern. Agard, et al. [10] used the clustering technique based on the boarding time of transactions to identify temporal characteristics of passenger behaviour on a weekly basis. Then, they measured changes in cluster membership to explore intrapersonal variability in transit usage as a 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, Morency, et al. [17] investigated spatial variability of transit users through the frequency of usage of bus stops. Deschaintres, et al. [18] 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 [19] 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 the number of trips and the time and origin of the trip. Viallard, et al. [20] used k-means clustering observed the evolution of users' behaviours 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 usage [21].

As pointed out before, traditional distancing metrics are not suitable to handle time-series. Some researchers have tried to address this issue. Ghaemi, et al. [3] 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 measures as the proper methods for sequence comparison [22]. However, since they used hierarchical clustering, due to its limitation for large datasets, they forced to take samples.

We present some of the main studies on travel behaviour in TABLE 1. In all these studies, the authors either used traditional distances and treat time-series data as static one or have attempted to use a more appropriate distance metric. However, a sufficient clustering technique that is consistent with sequences has yet to be established.

TABLE 1. Previous studies in smart card analysis in public transit

Study	*Target object*	*Vector*	*Distance measure*	*Clustering method*	*Averaging method*	*Objective /Contributions*
[22]	Card-day	Boarding time (binary vector)	CCD and DTW	Hierarchical	-	-
[23]	Stop-day	Boarding and alighting time	-	K-means	-	Investigation of local environmental effects on human behaviour
[10]	Card-week	-	Euclidean	K-means	-	-
[3]	Card-day	Boarding time (binary vector)	SCP, CCD, and ACD (autocorrelation)	Hierarchical	-	Proposed a semicircle projection (SCP) method
[18]	Card-week	7 dispersion indicators (number of trips per day) and one intensity indicator (average number of trips)	Euclidean	K-means	-	-
[20]	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
[19]	Card-day	Boarding and alighting time (binary vectors)	Simple Matching Distance (SMD)	Hierarchical	-	Assessing simultaneously interpersonal and intrapersonal variability of user behaviour
[11]	Stop-day	Boarding and alighting time		K-means	-	This study presents one of the first attempts of exploring the relationship between local LCLU and metro ridership patterns
[24]	Stop-day	-	Euclidean	K-means	-	Investigating whether station ridership's diurnal pattern is closely related to the local built environment
[12]	Stop-day	Boarding time	-	PCA + K-means	PAM	-
[25]	Card-day	Boarding time (binary vectors)	Euclidean	K-means	-	Dimensionality reduction

DATA & METHODOLOGY

Dataset Presentation

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 dataset contains 3,233,580 smart card transactions that were generated by 159,499 cards from 2019/02/01 to 2019/02/28. Each observation contains a validation id, transaction date and time, a line number and a direction, as well as an anonymized OPUS id; representing the card number, which is unique for each passenger, besides fare-types information.

Proposed Methodology Framework

Our proposed methodology consists of three main steps: (1) preprocessing, (2) applying the methods, and (3) comparison parts. These steps with their details are depicted in FIGURE 1.

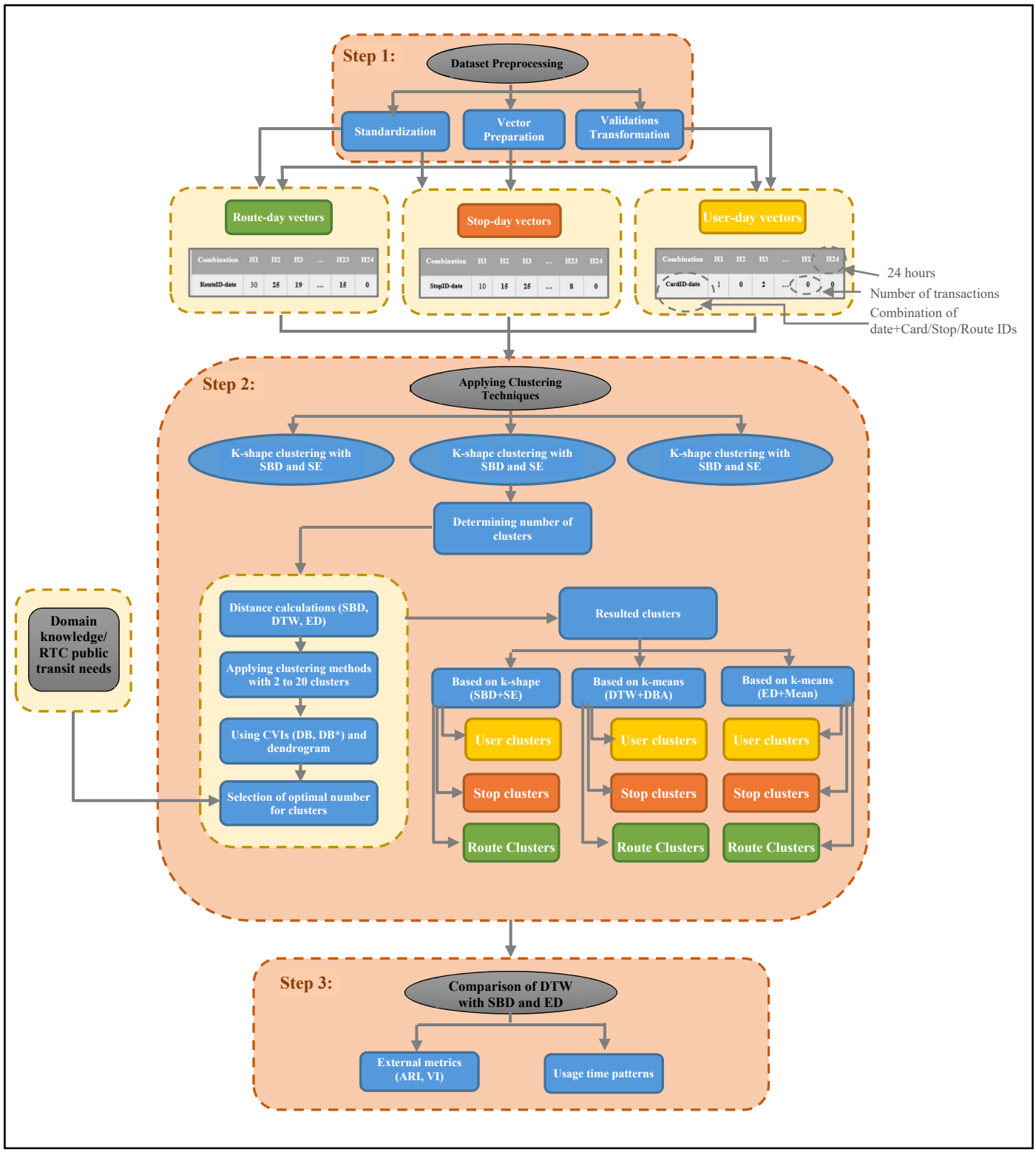

FIGURE 1. Proposed methodology diagram

STEP 1-Dataset Preprocessing

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). Since we aim to analyse the boarding time (origin) of the trips, 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 [18, 19]. 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.

Creation of vectors. We prepare vectors on three 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. We assume that there is an unambiguous relation between users and cards (1 card = 1 user). For the creation of user-day vector, from the dataset with 3,233,580 smart card transactions, based on what we discussed in the previous section, a total of 2,502,141 trips were obtained. Considering all trips for one user in one day as a one vector, a total of 1,356,537 user-day profiles were created with the 24 variables 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 "Date" in one column and named it "idate." We then, based on column "time" (hour), created 24 variables each represents daily hours, and the values are the number of trips in the corresponding hour. TABLE 2 shows an example of user-day vectors. For instance, user "1000010" had one trip between 11:00 and 11:59, so H_{11}has the value of one. The same procedure is followed for the creation of stop and route vectors.

TABLE 2. Example dataset of user-day

idate	H_1	H_2	…	H_6	H_7	…	H_{11}	H_{12}	…	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

Standardisation of data. In our dataset, we utilise the Z-score approach [26] to standardise data before applying clustering algorithms to analyse stop-day and route-day vectors because the range of variables varies between 0 to 1000. We do not use it for user-day analysis since its range of 0 to 12 is too narrow.

STEP 2-Applying Clustering Techniques

K-means Clustering with DTW and ED

K-means performs two steps in time-series clustering: (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 which should represent the most characteristics of other sequences in that given cluster, are modified using the prototyping (averaging) function. The choice of this function is closely related to the choice of distance measure [27].

These two processes in k-means repeat until the cluster membership does not change or the maximum number of iterations is reached [4]. In the following parts, we present the two distances and two prototyping techniques we use along with k-means.

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.

Euclidean distance (ED). It is a competitive well-known distance measure which computes the dissimilarity between $\vec{x}$ and $\vec{y}$ ($m = n$), as bellows [28]:

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

Dynamic time warping (DTW). This 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 [14]. In a case of $m = n$, between all two points of these series, ED 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 [29]:

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

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

$$\gamma(i,j) = ED(i,j) + min\begin{cases}\gamma(i-1, j-1)\\ \gamma(i-1, j)\\ \gamma(i, j-1)\end{cases} \tag{3}$$

Since there are many possible warping paths, for optimising DTW's performance we can put a constraint to limit the area of matrix *M* for mapping which is called *warping window*.

Mean. 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 give poor result [30].

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 [27, 30].

K-shape Clustering with SBD

K-shape is based on an iterative refining technique that is similar to the k-means algorithm, but it uses a different distance metric (SBD), and a different approach for centroid computation (SE).

Cross-correlation distance (CCD). This is a proper and widely used distance measure in comparing time-series data. 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 [4]:

$$\vec{x}_{(s)} = \begin{cases} (0, \dots, 0, x_1, x_2, \dots, x_{m-s}), & s \geq 0 \\ (x_{1-s}, \dots, x_{m-1}, x_m, 0, \dots, 0), & s < 0 \end{cases} \quad (4)$$

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

$$CC_w(\vec{x}, \vec{y}) = R_{w-m}(\vec{x}, \vec{y}), \qquad w \in \{1,2, \dots, 2m-1\} \quad (5)$$

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

$$R_k(\vec{x}, \vec{y}) = \begin{cases} \sum_{l=1}^{m-k} x_{l+k} . y_l & k \geq 0 \\ R_{-k}(\vec{x}, \vec{y}) & k < 0 \end{cases} \quad (6)$$

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 $s = m - w$.

Shape-based distance measure (SBD). This is a normalised version of cross-correlation distance proposed by Paparrizos and Gravano [4] 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]. Once the amount of w in which $NCC_c(\vec{x}, \vec{y})$ is maximum is determined, SBD will be calculated as follows:

$$SBD(\vec{x}, \vec{y}) = 1 - \max_w \left(\frac{CC_w(\vec{x}, \vec{y})}{\sqrt{R_0(\vec{x}, \vec{x}) . R_0(\vec{y}, \vec{y})}} \right), \qquad 0 \leq SBD \leq 2 \quad (7)$$

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

Shape-extraction (SE). This method has been proposed by Paparrizos and Gravano [4]. They suggested using the concept of optimisation problem; the minimum within-cluster sum of squared distances, but since shape-based and cross-correlation distance capture similarity - rather than dissimilarity - of sequences, it changes to maximisers:

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

$$= \underset{\vec{\mu}_k}{\operatorname{argmax}} \sum_{\vec{x}_i \in P_k} \left(\max_w \frac{CC_w(\vec{x}, \vec{y})}{\sqrt{R_0(\vec{x}, \vec{x}) . R_0(\vec{y}, \vec{y})}} \right)^2 \quad (8)$$

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. 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 this equation. Then, by combining equations 5 and 6, we will have:

$$\vec{\mu}_k^* = \underset{\vec{\mu}_k}{\operatorname{argmax}} \sum_{\vec{x}_i \in P_k} \left(\sum_{l \in [1,m]} x_{il} . \mu_{kl} \right)^2 \quad (9)$$

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

$$\vec{\mu}_k^* = \underset{\vec{\mu}_k}{\operatorname{argmax}} \vec{\mu}_k{}^T . \sum_{\vec{x}_i \in P_k} (\vec{x}_i . \vec{x}_i^T) . \vec{\mu}_k \quad (10)$$

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 it 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:

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

$$\vec{\mu}_k^* = \underset{\vec{\mu}_k}{\operatorname{argmax}} \frac{\vec{\mu}_k^T . M . \vec{\mu}_k}{\vec{\mu}_k^T . \vec{\mu}_k} \quad (11)$$

Where $M = Q^T . S . Q$. Using the preceding transformations, equation 10 was simplified to the optimisation of this equation, which is a well-known problem called maximisation of the Rayleigh Quotient [4].

Since we use R programing and mainly *dtwclust* package, to apply each of these three clustering approaches, we only need to change the distances to DTW, SBD, and ED, as well as the prototyping methodologies to DBA, SE, and Mean, respectively which all are implemented in *tsclust* function. In DTW, we put warping window equals to 1. For simplicity, we refer to our three methods by their distances: DTW, SBD, and ED.

Clustering Validation Techniques

After performing clustering, it is common to see how well it performed in creation of true clusters. There are two types of metrics: Internal and external measures. Moreover, in k-means and k-shape, there is a need to specify the number of clusters when the method is applied. There are several methods to address this issue, cluster validation internal metrics are among the popular ones.

Internal indices. These metrics 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 [31, 32]. Among all, we review two well-known internal cluster validation indices (CVIs): Davies-Bouldin (DB) and modified Davies-Bouldin (DB*).

(1) DB 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:

$$DB = \frac{1}{k} \sum_{c_k \in C} max \left\{ \frac{S(c_k) + S(c_l)}{d(\bar{c}_k, \bar{c}_l)} \right\} \quad (12)$$

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.

(2) DB* is the modified version of DB:

$$DB^* = \frac{1}{k}\sum_{c_k \in C} \frac{\max\{S(c_k)+S(c_l)\}}{min\{d(\bar{c}_k,\bar{c}_l)\}} \quad (13)$$

External indices. These 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. Furthermore, using these external measures is also common when we want to compare the results of several clustering methods applied to the same dataset. In the following part, we review the two most prominent ones.

(1) ARI [33-35], is based on counting the pairs of objects that two clustering methods agree/disagree on. 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 simplified contingency table of these partitions is as follows:

TABLE 3. Simplified contingency table

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 $a = \sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{ij}}{2}$, $b = \sum_{i=1}^{R}\binom{n_{i.}}{2} - a$, $c = \sum_{j=1}^{C}\binom{n_{.j}}{2} - a$, $d = \binom{n}{2} - a - b - c$. Therefore, ARI is equal to:

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]} \quad (14)$$

(2) VI is based on entropy. If we call $H(C)$ as the entropy associated with clustering C, then we have:

$$H(C) = -\sum_{k=1}^{K} p(k)\log p(k) \quad (15)$$

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:

$$I(C, C') = \sum_{k=1}^{K}\sum_{k'=1}^{K'} p(k,k') log \frac{p(k,k')}{p(k)p'(k')} \quad (16)$$

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:

$$VI(C, C') = [H(C) - I(C, C')] + [H(C') - I(C, C')] \quad (17)$$

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 illustrates the concept and the relation between information entropies, mutual information, and variation of information more clearly [36].

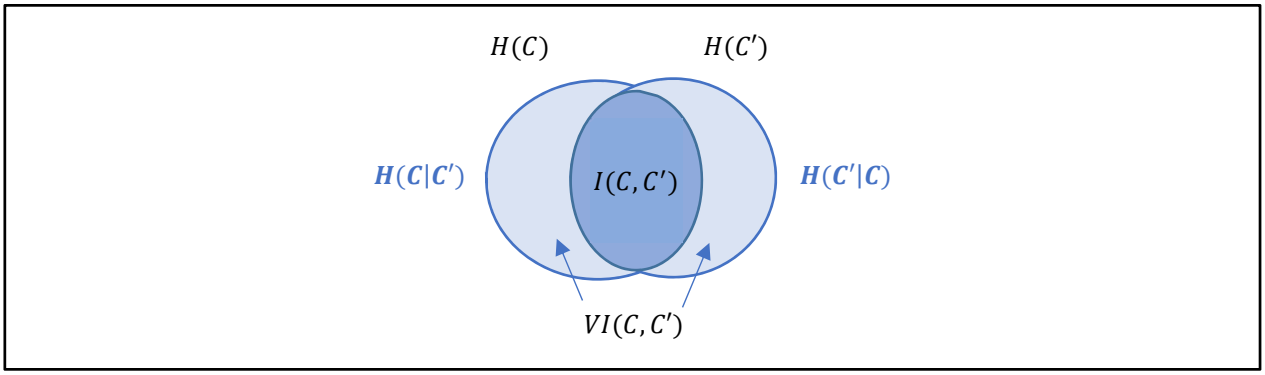

FIGURE 2. Information diagram

STEP 3-Comparison of Clustering Techniques

First, DTW is selected as the ground truth. The development of our comparative framework was then directed by two criteria. The first criterion is based on statistical measures including the two external measurements we described. Each approach has a larger ARI value, and a lower VI value is more compatible with DTW partitions. Second, we compare the clustering methods based on their patterns, rather than solely using statistical concept. This would help us to see the differences between methods' partitions in detail.

RESULTS & ANALYSIS

User-Day Analysis

After the preprocessing step, the three clustering algorithms of DTW, SBD, and ED are applied to the user-day vector. For optimal number of clusters, DB and DB* are used. In doing so, clustering methods are applied by considering numbers for clusters from 2 to 20, and these indices are calculated for each result. The value of these indices is then used to compare the quality of the partitions. The given number of clusters that yield to the better resulted indices would be the best choice as a prior cluster number. In addition to CVIs, we employ a dendrogram approach across 30 centres. The prior number of clusters is set to 30, and then the clustering algorithm is applied to the entire dataset. The 30 centroids are then plotted on a dendrogram [17].

FIGURE 3 (a) shows the minimum amounts of DB and DB* corresponding 6, 19, 9, and 12 number of clusters for DTW. In this figure (b), looking top to bottom of the dendrogram, 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. The same procedure was followed for SBD and ED and the number of 9 and 10 were chosen respectively.

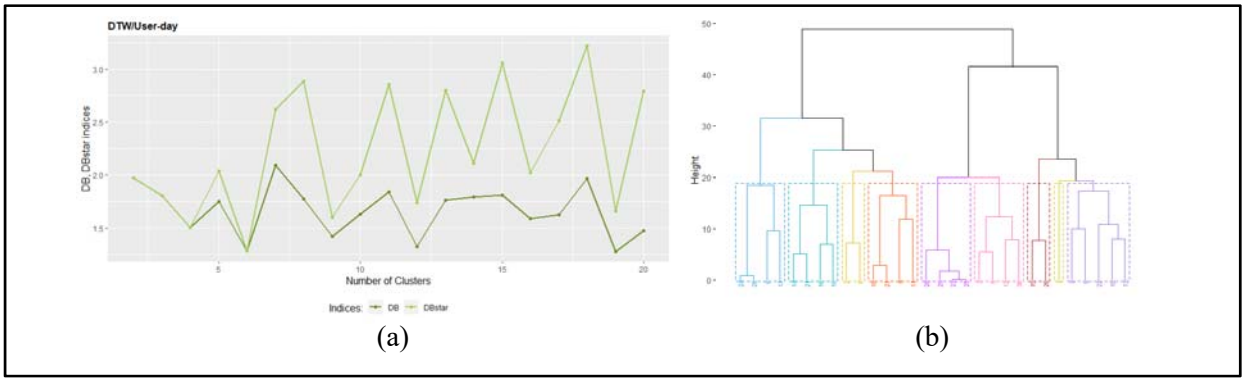

FIGURE 3. Selection of the optimal number of clusters for users under DTW by: (a) DB and DB* (b) dendrogram

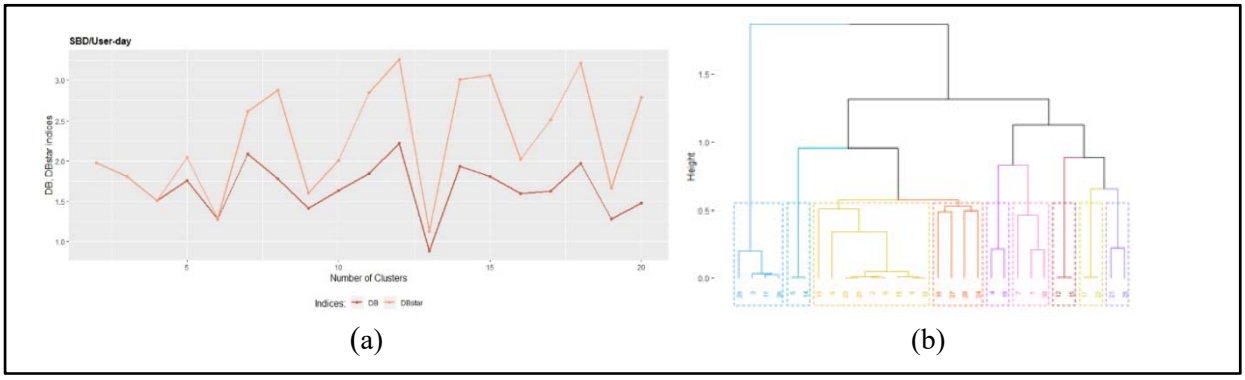

FIGURE 4. Selection of the optimal number of clusters for users under SBD by: (a) DB and DB*, (b) dendrogram

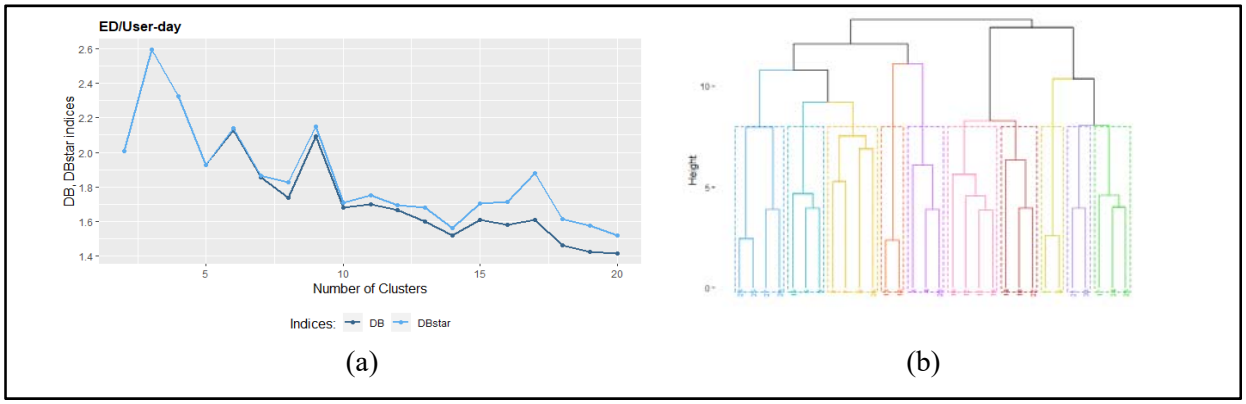

FIGURE 5. Selection of the optimal number of clusters for users under ED by: (a) DB and DB*, (b) dendrogram

External metric comparison. In this study since we mainly use *dtwclust* package in R, ARI and VI index are implemented in the main function of cvi. So, we do not need to use the contingency tables directly for the calculation. 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.

For SBD and ED, ARI and VI were calculated while putting DTW partitions as the ground truth. ARI and VI for SBD are equal to 0.184 and 1.322 and for ED are 0.099 and 1.254, respectively. The bigger amount of ARI for SBD, shows higher agreement between SBD and DTW than ED and DTW, whereas the smaller VI for ED challenges this conclusion.

Pattern comparison. The resulted cluster centroid patterns from applying DTW, SBD, and ED are plotted over 24 hours of a day in FIGURE 7,FIGURE 6, andFIGURE 9. We also plotted the pie charts in FIGURE 8.

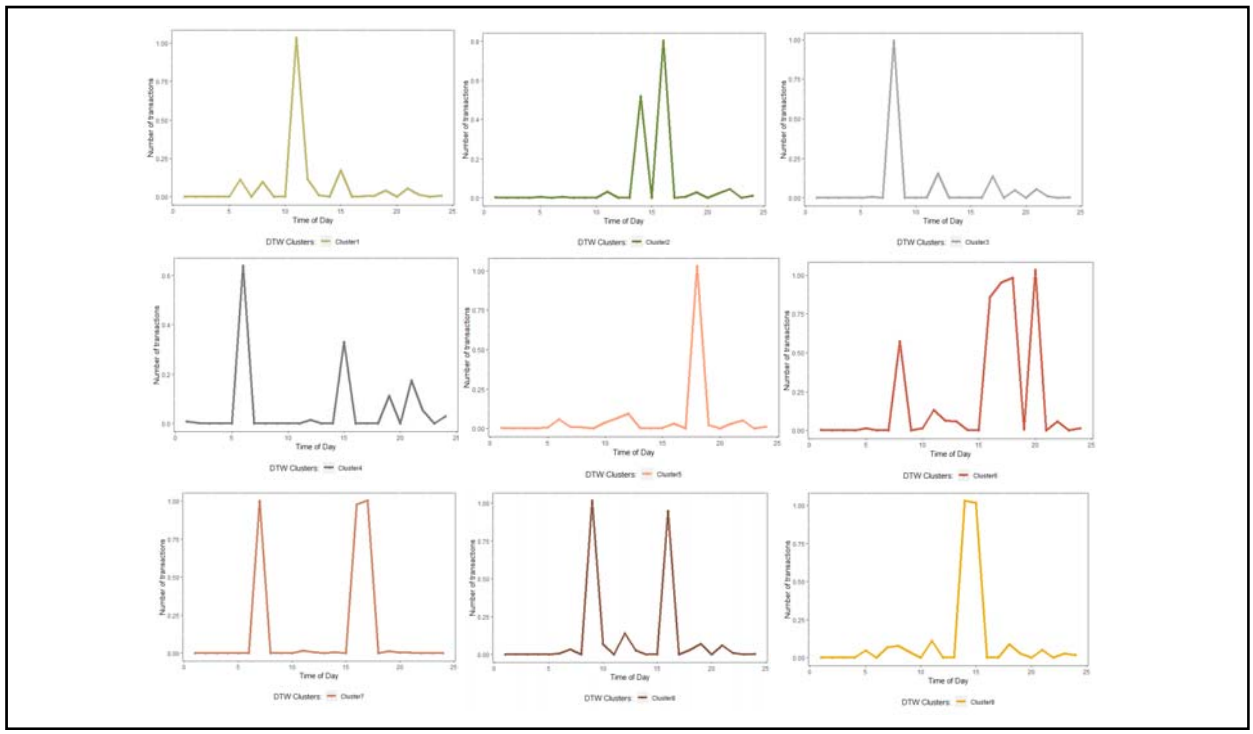

FIGURE 6. DTW user clusters' patterns

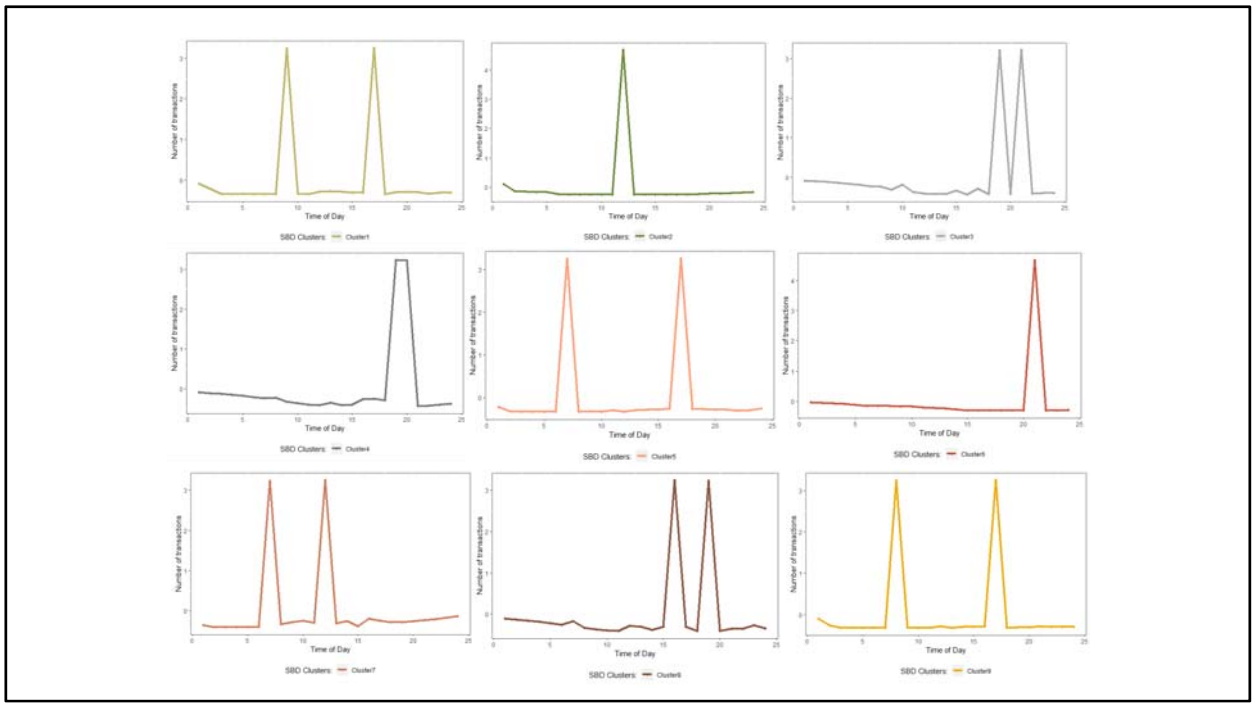

FIGURE 7. SBD user clusters' patterns

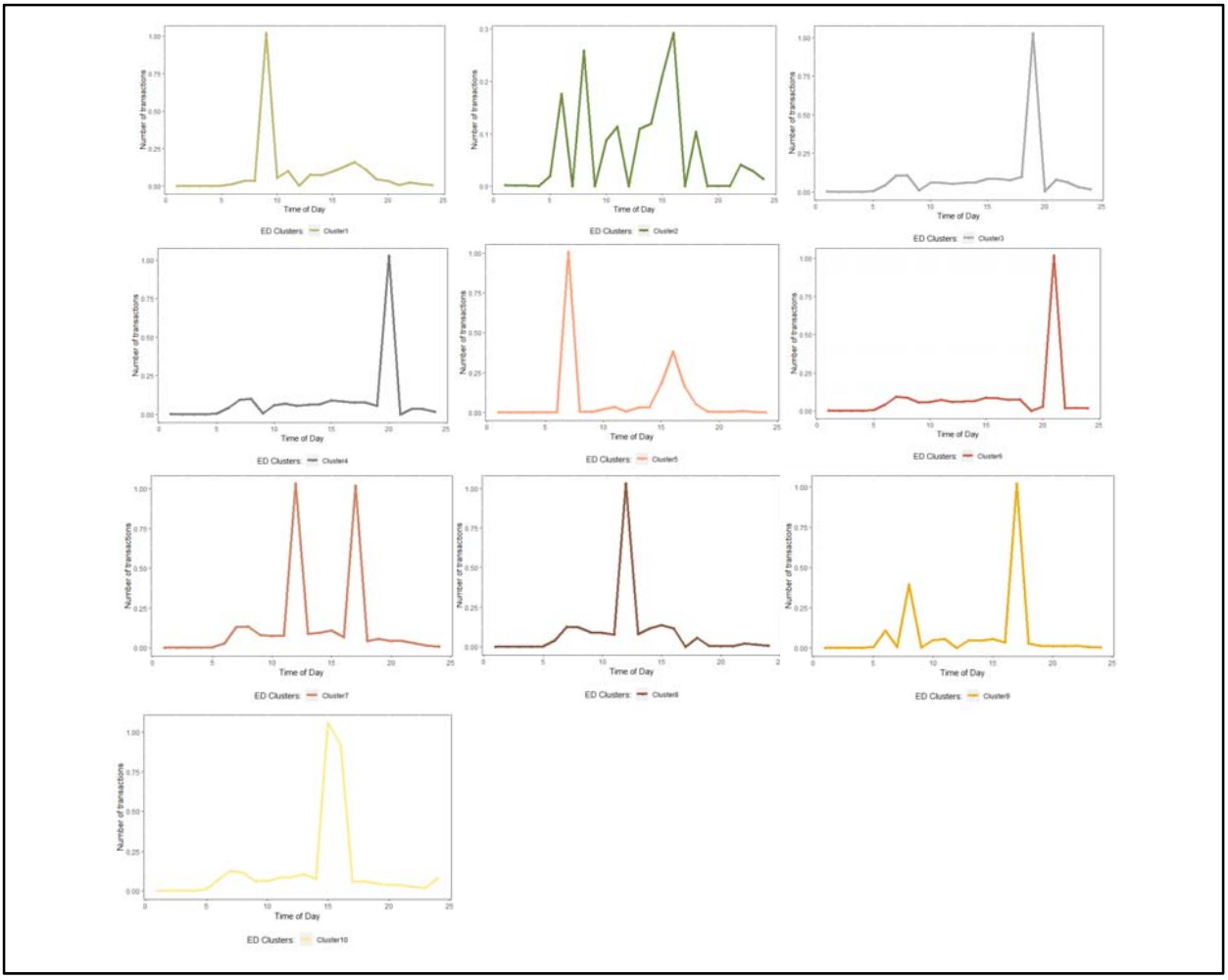

FIGURE 8. ED user clusters' patterns

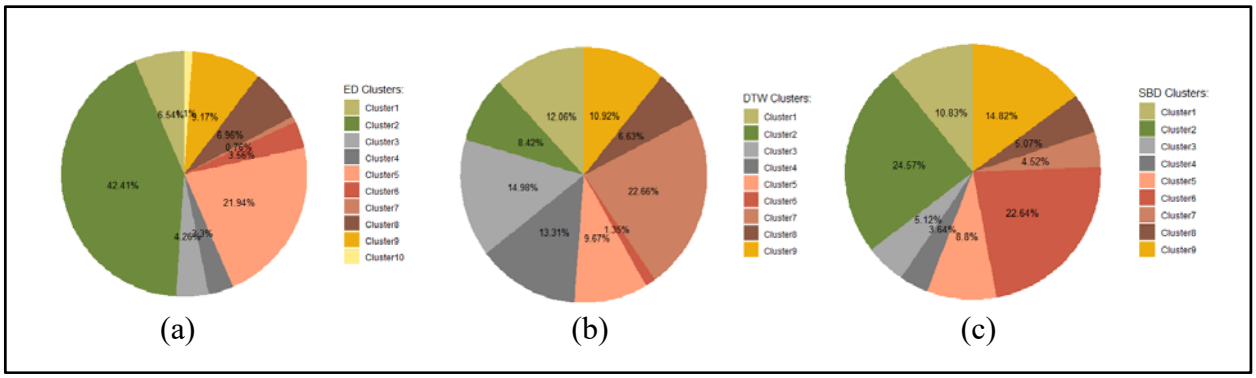

FIGURE 9. Clusters' portions: (a) DTW, (b) SBD, (c) ED

According to the patterns, we categorised them in four main groups; considering their portions in pie charts, the clusters' 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. SBD clusters 1, 5, 7, and 9 are identified as having the regular pattern with the 44.04% portion of all users. 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. The only cluster in SBD that has the same characteristics of this group is cluster 2 with the portion of 24.57%. 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, clusters 6, 3, 4, and 8 are also having the same characteristics of this group with the total portion of 36.47%. Clusters 3, 4, and 6 is the only cluster from ED are identified in 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. Based on SBD patterns, there is no cluster having the same pattern of this group. But ED has cluster 1 with the portion of 6.54%.

In terms of DTW and SBD comparison, we 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 case 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 a slight shift in time. 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.

CONCLUSION

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, the variations and distortions of time-series data in segmentation process has received less attention in smart card studies.

In this paper, we used k-shape clustering with the SBD method 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, and the most commonly used one, ED along with k-means clustering. 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 but might result in a big difference for two sequences with the same shape.

As a ground truth, we had the method of DTW which considers both the shape and the shift of time-series but suffers from time complexity. This comprehensive examination of three methodologies, each with its own set of features, can help to pave the way for smart card data analysis in transportation research, allowing selecting a method that is more compatible with the specific characteristics of data and the objective of the study.

In the comparison section, we looked at two points of view: external measures and patterns in usage time. This allowed us to compare methods in greater depth rather than only on the basis of the indices. For instance, despite ED had better agreement with DTW outcomes than SBD in terms of VI value in user-day analysis, it performed worse than SBD in recognising well-defined patterns.

Furthermore, 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, for example, should perform better comparison where there is less shifting in time, and it met our expectations for route analysis, and produced competitive results comparing to ED.

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. When there is a time constraint and a large dataset, 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 is a quick and simple approach, it is not a wise option for time-series comparison.

Limitations

From the methodological point of view, since there was no prior information as the actual clusters, we were compelled to use the results of k-means clustering with DTW distance as a ground truth to compare it with k-shape clustering with SBD for revealing the reliability of this novel method in our case; nevertheless, the DTW method has its own imperfections in the clustering process.

Perspectives

Introducing k-shape clustering with SBD for smart-card data segmentation could open up several possibilities for future research with different objectives. In fluctuation analysis, for instance, it can produce highly competitive results in the detection of very well-defined patterns. Furthermore, in studies where time-shifting comparison is required, it is necessary to investigate a way to modify this approach to be constrained for shifting.

ACKNOWLEDGMENTS

The authors wish to acknowledge the collaboration of the *Réseau de Transport de la Capitale* for providing data and the financial support of the *Natural Sciences and Engineering Research Council of Canada (NSERC),* the group of *THALES,* the *CORTEX* fund, and *PROMPT.*

REFERENCES

[1] M.-P. Pelletier, M. Trépanier, and C. Morency, "Smart card data use in public transit: A literature review," *Transportation Research Part C: Emerging Technologies,* vol. 19, no. 4, pp. 557-568, 2011.

[2] M. Zhao, L. Mason, and W. Wang, "Empirical study on human mobility for mobile wireless networks," presented at the IEEE Military Communications Conference, MILCOM 2008, 2008.

[3] M. S. Ghaemi, B. Agard, M. Trépanier, and V. Partovi Nia, "A visual segmentation method for temporal smart card data," *Transportmetrica A: Transport Science,* vol. 13, no. 5, pp. 381-404, 2017.

[4] J. Paparrizos and L. Gravano, "Fast and Accurate Time-Series Clustering," *ACM Transactions on Database Systems,* vol. 42, no. 2, pp. 1-49, 2017.

[5] R. Arbex and C. B. Cunha, "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,* vol. 85, 2020.

[6] F. Cavallaro and A. Dianin, "An innovative model to estimate the accessibility of a destination by public transport," *Transportation Research Part D: Transport and Environment,* vol. 80, 2020.

[7] M. Yap, O. Cats, and B. van Arem, "Crowding valuation in urban tram and bus transportation based on smart card data," *Transportmetrica A: Transport Science,* vol. 16, no. 1, pp. 23-42, 2018.

[8] J. Seo, S.-H. Cho, D.-K. Kim, and P. Y.-J. Park, "Analysis of overlapping origin–destination pairs between bus stations to enhance the efficiency of bus operations," *IET Intelligent Transport Systems,* vol. 14, no. 6, pp. 545-553, 2020.

[9] T. Zhang, Y. Li, H. Yang, C. Cui, J. Li, and Q. Qiao, "Identifying primary public transit corridors using multi-source big transit data," *International Journal of Geographical Information Science,* vol. 34, no. 6, pp. 1137-1161, 2018.

[10] B. Agard, C. Morency, and M. Trépanier, "Mining Public Transport User Behaviour from Smart Card Data," *IFAC Proceedings Volumes,* vol. 39, no. 3, pp. 399-404, 2006.

[11] Z. Gan, M. Yang, T. Feng, and H. Timmermans, "Understanding urban mobility patterns from a spatiotemporal perspective: daily ridership profiles of metro stations," *Transportation,* vol. 47, no. 1, pp. 315-336, 2018.

[12] J. Reades, C. Zhong, E. D. Manley, R. Milton, and M. Batty, "Finding Pearls in London's Oysters," *Built Environment,* vol. 42, no. 3, pp. 365-381, 2016.

[13] J. Han and M. Kamber, *Data mining : concepts and techniques*, 2nd ed. (Morgan Kaufmann series in data management systems). Amsterdam, Pays-Bas: Elsevier : Morgan Kaufmann Publishers, 2006, pp. xxviii, 770 p.

[14] J. Paparrizos, "Fast, scalable, and accurate algorithms for time-series analysis," Doctor of Philosophy, Graduate school of arts and sciences, Columbia University, 2018.

[15] T. Warren Liao, "Clustering of time series data—a survey," *Pattern Recognition,* vol. 38, no. 11, pp. 1857-1874, 2005.

[16] G. E. A. P. A. Batista, E. J. Keogh, O. M. Tataw, and V. M. A. de Souza, "CID: an efficient complexity-invariant distance for time series," *Data Mining and Knowledge Discovery,* vol. 28, no. 3, pp. 634-669, 2013.

[17] C. Morency, M. Trépanier, and B. Agard, "Measuring transit use variability with smart-card data," *Transport Policy,* vol. 14, no. 3, pp. 193-203, 2007.

[18] E. Deschaintres, C. Morency, and M. Trépanier, "Analyzing Transit User Behavior with 51 Weeks of Smart Card Data," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2673, no. 6, pp. 33-45, 2019.

[19] O. Egu and P. Bonnel, "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,* vol. 19, pp. 112-123, 2020.

[20] A. Viallard, M. Trépanier, and C. Morency, "Assessing the Evolution of Transit User Behavior from Smart Card Data," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2673, no. 4, pp. 184-194, 2019.

[21] M. Moradi and M. Trépanier, "Assessing longitudinal stability of public transport users with smart card data," *Transportation Research Procedia,* 2018.

[22] L. He, B. Agard, and M. Trépanier, "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,* vol. 16, no. 1, pp. 56-75, 2018.

[23] M.-K. Kim, S.-P. Kim, J. Heo, and H.-G. Sohn, "Ridership patterns at subway stations of Seoul capital area and characteristics of station influence area," *KSCE Journal of Civil Engineering,* vol. 21, no. 3, pp. 964-975, 2017.

[24] C. Chen, J. Chen, and J. Barry, "Diurnal pattern of transit ridership: a case study of the New York City subway system," *Journal of Transport Geography,* vol. 17, no. 3, pp. 176-186, 2009.

[25] B. Agard, V. Partovi-Nia, and M. Trépanier, "Assessing public transport travel behaviour from smart card data with advanced data mining techniques," presented at the 13th WCTR, Rio de Janeiro, Brazil, July 15-18, 2013.

[26] I. B. Mohamad and D. Usman, "Standardisation and Its Effects on K-Means Clustering Algorithm," *Research Journal of Applied Sciences, Engineering and Technology,* vol. 6, no. 17, pp. 3299-3303, 2013.

[27] A. Sardá-Espinosa, *Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package*. 2018.

[28] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *ACM SIGMOD Record,* vol. 23, no. 2, pp. 419-429, 1994.

[29] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems,* vol. 7, no. 3, pp. 358-386, 2005/03/01 2005.

[30] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition,* vol. 44, no. 3, pp. 678-693, 2011.

[31] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. s. M. Pérez, and I. i. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition,* vol. 46, no. 1, pp. 243-256, 2013.

[32] L. A. Pérez, Á. M. García Vico, P. González, and C. J. Carmona, "Techniques for Evaluating Clustering Data in R. The Clustering Package," 2020.

[33] R. Rabbany and O. R. Zaïane, "Generalization of clustering agreements and distances for overlapping clusters and network communities," *Data Mining and Knowledge Discovery,* vol. 29, no. 5, pp. 1458-1485, 2015.

[34] M. Z. Rodriguez *et al.*, "Clustering algorithms: A comparative approach," *PLoS One,* vol. 14, no. 1, p. e0210236, 2019.

[35] J. Santos and M. Embrechts, *On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification*. 2009, pp. 175-184.

[36] M. Meilă, "Comparing clusterings—an information based distance," *Journal of Multivariate Analysis,* vol. 98, no. 5, pp. 873-895, 2007.