Clustering Subway Station Arrival Patterns Using Weighted Dynamic Time Warping

Rui Wang[1], Nan Chen [1], Chen Zhang [2]
[1]Department of Industrial and Systems Engineering, National University of Singapore, Singapore
[2]Department of Industrial Engineering, Tsinghua University, Beijing, China
(r.wang@u.nus.edu, isecn@nus.edu.sg, zhangchen01@tsinghua.edu.cn)

Abstract—**To better plan and schedule public transportation resources, it is crucial to understand the travel demand from any location at any time. In this article, we focus on analyzing the demand patterns for subway stations based on the tap in data at each station entrance. It has been reported that accurately predicting the arrival rates can help improve the travel experience, and prevent over-crowding in train carriages or platforms. We proposed a weighted dynamic time warping approach (WDTW) to adaptively cluster similar patterns from multiple stations. These similarities can be exploited in improving the prediction performance because spatial temporal information is better utilized. We demonstrated our approach and its effectiveness through a real data example.**

Keywords—**Public transportation, arrival rate, dynamic time warping, hierarchical clustering**

I. INTRODUCTION

In public transportation planning and operation scheduling, one of the most important information is the travel demand, i.e., the volume of traffic at any time slot. The fast development in computing power and data management capability provides us with the opportunity to infer the travel demand from high throughput historical data. For example, in metro systems of many metropolitan cities, a fare card needs to be tapped to enter a station (origin), and the same card needs to be tapped to exit another station (destination). These entry and exit data record the number of passengers moving from one place to another in fine granularity. Such information has been proven valuable in both long term planning and short term scheduling of public transportation resources in the most cost effective way. As a result, analyzing such entry/exit data has attracted much attention in the literature [1].

Among existing works, the analysis of subway travel demand can be roughly categorized into long-term and short-term periods. Long-term analysis estimates future travel demand based on four-step transportation forecasting model [2] or regression techniques. Transportation forecasts have traditionally followed the sequential four-step model, i.e., trip generation, trip distribution, mode choice and route assignment [3], [4]. Regression models are utilized to find out the relationship between subway ridership and a series of influential factors [5], [6]. Short-term models are designed to record the regular patterns of travel flow for future predictions. Existing methods include models based on time series analysis [7], [8], as well as machine learning methods such as neural networks [9], [10] and support vector machine [11].

However, most existing approaches focus on the modeling and prediction of passenger entries from a single station. They treat the entry data from different stations as independent, and analyze them separately. Despite their simplicity, neglecting the dependency in travel demand among multiple stations has its limitations. First of all, many stations are physically connected in the transportation network. A disruption in one station can quickly propogate to other stations to change their demand patterns. Secondly, the location of the station plays a crucial role in driving the demand patterns. Clearly a station in central business district has a very different demand compared with a station in residential area. In contrast, nearby stations in the same residential areas have very similar demand patterns. The possible dependency among certain stations permits information sharing to improve the prediction accuracy for each individual station. Nevertheless, the dependency structure is unknown and not easy to derive from network structure or population distributions. Clustering based on real data of demand patterns is more feasible and flexible.

Even though clustering is a mature technique, clustering demand patterns have a few unique challenges. Initially, subway entry count is usually collected with a short time interval, which shows drastic fluctuations in the long data sequence. Then the arrival rate function cannot be directly derived from the raw data. Preprocessing techniques like resampling and smoothing show be considered. Besides, time deformations of peak hours is often seen for different stations even though they belongs to the same type of functional districts. For instance, stations in different residential areas may have different morning peak hours on the same day. This is caused by the variation of distances between the stations and business or industrial areas which provide plenty of job positions. However, in this case the stations should be placed in the same cluster, thus the influence of peak shift needs to be relieved when clustering. Finally, subway arrival patterns are quite different on weekdays and on weekends. It is easy to explain, as people mostly go to work on weekdays and take rest on weekends. Therefore, the factor of days of the week should be taken into account when analyzing travel demand.

In this study, to overcome the above challenges we

978-1-5386-6786-6/18/$31.00 ©2018 IEEE

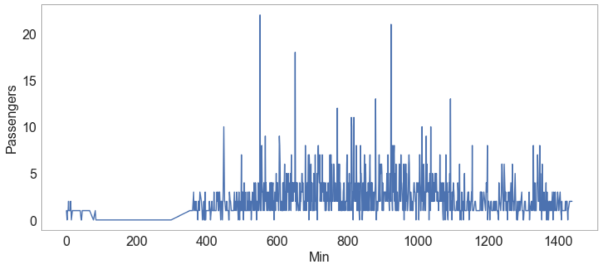
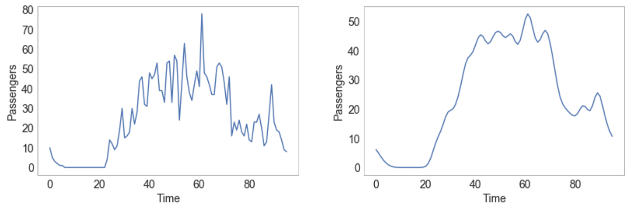


Fig. 1. Original entry rate data in one day.



(a) Resampled data (b) Smoothed data

Fig. 2. Example of data smoothing.

accomplish the clustering task with a distance measure based on dynamic time warping (DTW) [12]. DTW is a robust method for calculating the dissimilarity between temporal sequences, which is originally proposed for automatic speech recognition to deal with different speaking speeds. It allows non-linear alignment to find the optimal warping path between two sequences with the minimum cumulative distance. To penalize the phase difference between aligned points, we add a weight in the calculation of DTW distance. We propose an arrival pattern clustering framework that contains three steps: data smoothing, distance computation and clustering. We first use smoothing kernels to estimate the arrival rates for different stations on all the days. A proposed distance measure which combines the information for different days of the week is then utilized to calculate pairwise distance between stations. Hierarchical clustering is further applied for the construction of clusters.

The remainder of this article is organized as follows. Section II gives detailed explanation of the proposed framework for clustering subway demand patterns. In section III, we discuss our results of the proposed method on real-world data. Section IV summarizes the article with potential future directions.

II. Methodology

A. Data Smoothing

In real applications, the sampling frequency of smart card transaction data is often high, e.g., every one-minute, which causes fluctuations in the curve. Figure 1 shows an example of the entry rate collected at one station during one day. The number of passengers was counted separately in each one minute. Thus, there are totally 1440 sample points within one day. The long sequence of time series data along with extensive fluctuations will increase the computation load and meanwhile reduce clustering accuracy, which calls for a need to improve smoothness of raw data at the preprocessing stage.

In order to reduce the length of sequence, we first apply a resampling method. Figure 2a gives an illustration of resampling at the interval of 15 min. After resampling, only 96 sampling points remain in a single day. For data smoothing, we apply the classic nonparametric approach based on smoothing kernels. Let $x(t)$ denote the resampled passenger entry count at each time point t, and $y(t)$ denote the curve after smoothing. The smoothed curve for station i on day j is estimated in the following form:

$$y_{ij}(t_k) = \frac{\sum_{l=1}^{n_{ij}} \mathcal{K}(\frac{t_k - t_l}{b}) x_{ij}(t_l)}{\sum_{l=1}^{n_{ij}} \mathcal{K}(\frac{t_k - t_l}{b})}, \tag{1}$$

where n_{ij} is the length of the sequence x_{ij}. $\mathcal{K}(\cdot)$ is the kernel smoother function and b is the kernel bandwith which controls the length scale of the smoothing window. Here we apply Gaussian kernel smoother for this approach

$$\mathcal{K}(t) = \exp(-t^2/2). \tag{2}$$

Figure 2b draws the entry rate data after proprecessing, it can be shown that the curve is quite smooth and interpretable and the time-variant information is largely retained.

B. Weighted Dynamic Time Warping

In this section, we discuss how to calculate the distance between each pair of stations based on a weighted dynamic time warping (WDTW) approach. We start from the classic dynamic time warping (DTW). DTW is a popular shape-based similarity measure for time series data which breaks the limitation of one-to-one alignment. Also, it allows unequal length of sequences for distance calculation. The idea of DTW is as follows. Assume $s1 = (u_1, u_2, \ldots, u_m)$ is a sequence of length m and $s2 = (v_1, v_2, \ldots, v_n)$ is a sequence of length n, the pairwise distance between u_i and v_j is first computed and stored in a $m \times n$ matrix. The best warping path is found to be the one with the lowest distance path after the alignment of one sequence to the other subject to the constraints: (a) Endpoint constraint (b) Continuity constraint (c) Step size constraint [13]. The optimal alignment is calculated recursively based on the cost distance function:

$$D(i,j) = d(i,j) + \min(D(i-1,j-1), D(i-1,j), D(i,j-1)). \tag{3}$$

where $d(i,j) = |u_i - v_j|$. The DTW distance is then defined as:

$$DTW(s1,s2) = \min_{p \in P} \sqrt{\sum_{k=1}^{K} d(p_k)}, \tag{4}$$

where P is the set of all possible warping paths, p_k is the position (i,j) at kth observation of a warping path, and K is the length of the warping path [14].

The standard DTW calcuates the distance of all points without considering the influence of phase difference. DTW may choose to make alignment which covers a large temporal range to obtain the minimum distance. However, in real application, the distance measure needs to balance between shape matching and temporal alignment. For example, in the analysis of subway entry rate curves, phase difference indeed matters, as it is inappropriate to match the point in the morning to the one in the afternoon when phase difference is large. Therefore, we first apply a modified distance function by a weight similar to [15] when creating the $m\times n$ matrix. Let $w_p(|i-j|)$ denote the postive weight value between the two points u_i and v_j, the weight value is calculated based on the phase difference $|i-j|$. In other words, smaller weights should be imposed for nearer point pairs. For generalization, when it comes to the case of unequal sampling intervals or discontinuous sampling points, using time point index directly is not a proper choise. Thus, we modify the notation of phase difference to $|t_i - t_j|$, which computes the elapsed time between the points in difference time series. Finally, the distance between paired points is

$$d_w(i,j) = w_p(|t_i - t_j|)d(i,j) = w_p(|t_i, t_j|)|u_i - v_j|. \quad (5)$$

Therefore, the weighted DTW distance between two time series sequences is

$$WDTW(s1,s2) = \min_{p\in P} \sqrt{\sum_{k=1}^{K} d_w(p_k)}. \quad (6)$$

Here we apply the logistic weight function which is defined as

$$w_p(x) = \frac{w_{max}}{1+\exp(-\alpha(x-\beta))}, \quad (7)$$

where w_{max} is the upper bound of the weight function. α controls the slope of the function, β is the midpoint of the sequence which will give weight 0.5 when w_{max} is 1.

Figure 3 shows the behavior of the weight curve when choosing different α. In accordance with the previous examples, the length of each time series is 96. Then $|t_i - t_j|$ takes value in the range $[0, 96]$. β here is fixed to be 48, which is the midpoint of the phase difference sequence. It should be noted that other β values are also applicable in order to change the center point of symmetry. w_{max} here equals 1 for simplicity. It is shown that all the curves are symmetric around the midpoint β. When $\alpha = 0$, the weight function reaches a constant value. When $\alpha = 0.05$, the curve is approximately linear with respect to the phase difference. When α is set to a large value, 1 for example, the middle part becomes quite steep, resulting in two distinct weight values separated by the midpoint.

C. Distance Measure Based on Weighted DTW

For public transportation, the entry rate is often periodic at the interval of seven days (one week). This can be explained by intuition that people live in similar patterns every week, e.g., go to company on weekdays

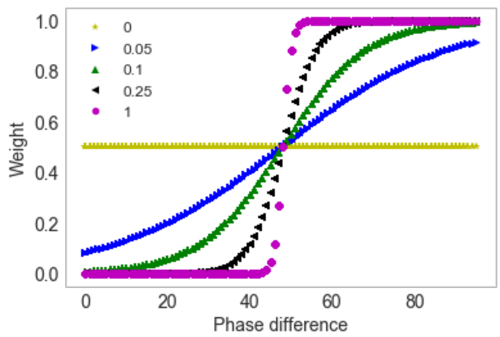


Fig. 3. Weight function for different values of α

and go shopping at weekends. Also, a subway station may be crowded at peak hours on weekdays but relatively lonesome at weekends. Thus, to look at the data at different days independently is not the best way to capture the features of arrival patterns for different stations.

Thus, we assume the daily distances computed in each week are identically and independently distributed. We can first separate the collected data on the weekly basis. For station i, let $Y_i = \{y_{i1}, y_{i2}, \dots\}$ denote the data collected at different date. To introduce the idea of week in the notation, we revise the elements of Y_i to be y_{iwd} which represents the curve collected at the wth week and dth day of the week. Thus, $Y_i = \{y_{i11}, y_{i12}, \dots, y_{i1n_d}, \dots, y_{in_w1}, y_{in_w1}, \dots, y_{in_wn_d}\}$, where n_w denotes the number of total weeks of the collected data and $n_d = 7$ is the number of days within one week. The distance measure initially computes WDTW on each date respectively and then averages on the week level to obtain the summarized distance value for each day of the week. The summarized distance between station i and station i' on day d of the week is

$$D_d(i,i',d) = \frac{1}{n_w}\sum_{w=1}^{n_w} WDTW(y_{iwd}, y_{i'wd}), \quad (8)$$

$d \in \{1, 2, \dots, 7\}$, which means the rolled sequence of {Monday, Tuesday,. . . ,Sunday}.

Finally, the station level distance is calculated with the 2-norm of the distance of all the days inside the week. Let $D_s(i,i')$ denote the station level distance between station i and station i'.

$$D_s(i,i') = \sqrt{\sum_{d=1}^{7} D_d(i,i',d)^2}. \quad (9)$$

Here we simply assume equal weights for each day of the week. Alternatively, different weights can be applied, for example, for weekdays and weekends. Similarly, we can look at the behavior of stations on weekdays or weekends separately. For the weekday distance measure, the station level distance measure (9) can be modified to calculate the 2-norm of $D_d(i,i',\cdot)$ with five days from Monday to Friday in d.

D. Hierarchical Clustering

In this section, we use hierarchical clustering for the clustering task. This method builds the hierarchy from the individual elements by progressively merging clusters based on the distance measure. Distances between sets of objects are updates with linkage criterion [16]. For pairwise distance stored in a matrix, rows and columns are merged as the clusters grow and the distances update. The application of hierarchical clustering does not need the prior knowledge of number of clusters. Also, it avoids the calculation of centroid of each cluster which is hard and time-consuming to be accurate when applying methods like k-means.

For each cluster, we find the medoid to represent the main features of the whole cluster. Assume there are N members inside cluster c, the medoid of cluster c is find to be

$$M(c) = \arg\min_{i} \sum_{i \neq j} D_s(c(i), c(j)), i = 1, \ldots, N. \quad (10)$$

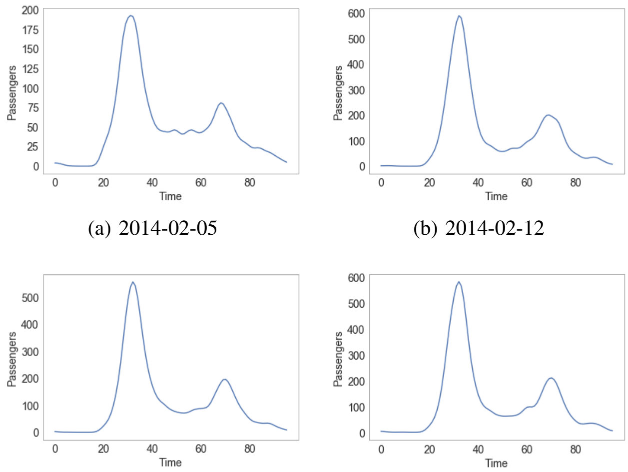
III. Experiments

A. Data Description

The dataset we use for analysis was published by Bostons Massachusetts Bay Transit Authority (MBTA) who operates the 4th busiest subway system in the U.S. after New York, Washington, and Chicago. The dataset contains passenger entry rate (per-minute count) at 63 stations along the red, orange, and blue line of the Boston's subway system, based on records from turnstiles for payment, for totally 30 days in 2014. For the benefit of weekly analysis, we select 28 consecutive days (four weeks) of data for our experiments, range from 2 Feb to 1 Mar. For raw analysis, we first use the smoothing kernels in Section II-A to estimate the entry rate function for every station everyday. After the processing, the smoothed sequence has 96 points with time interval 15 min, from 0AM in to 0AM+ on the next day. In this study, we assume the entry rate follows similar patterns every seven days. Figure 4 shows the smoothed entry rate curves of the station Alewife on four Wednesdays in February. It is shown that the entry rate functions share similar shape with a high peak in the morning and a small peak in the evening. This validates our assumption to treat the data in a periodic way.

B. Results & Discussion

In this section, we present the clustering results of the stations based on the entry rate data of 28 days. Before the computation of distance measure, we first apply Z-normalization to the smoothed curve to eliminate to influence of scaling. The procedure ensures that all elements of the input are transformed so that the mean is approximately 0 while the standard deviation is in a range close to 1. This can be simply done by first subtracting the time series mean from original values and then dividing the results by the standard deviation.



(a) 2014-02-05 (b) 2014-02-12

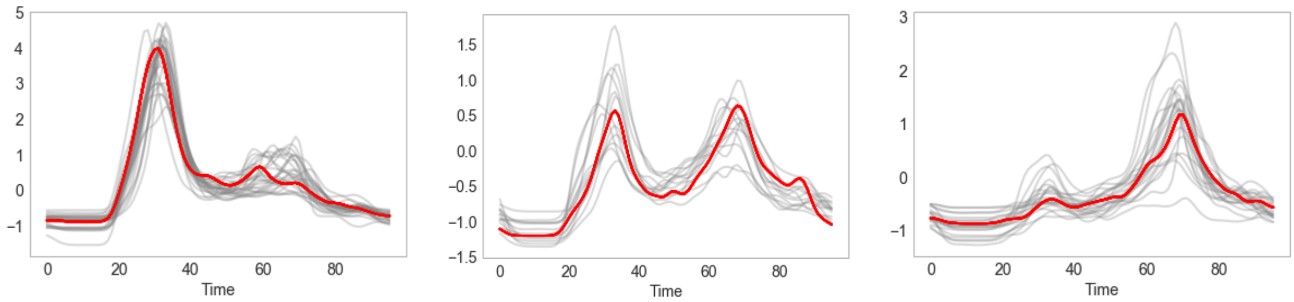
(c) 2014-02-19 (d) 2014-02-26

Fig. 4. Entry rates of station Alewife on Wednesdays.

In our experiment, we set the parameters $\alpha = 0.1$ and $\beta = 48$ for the distance measure. To better visualize results, we perform clustering on weekdays and weekends separately based on the modified distance measure of (9). The clustering results for weekdays are shown in Figure 5. The curves plotted here are from data collected on the first Wednesday (2014-02-07). The red curve represents the medoid found by (10) of each cluster. Stations in cluster 1 have morning peaks, roughly from 6AM to 8AM. Cluster 2 shows a two-peak pattern, in both morning and evening (peaking at aroung 5:30PM). cluster 3 shows one obvious evening peak, nearly symmetric with the pattern in cluster 1. Obviously, there exists deformations of peaks in each cluster, which shows the advantage of DTW. The shift can be explained by different distances from the station to the destinations, which require different department time in order to arrive on time.

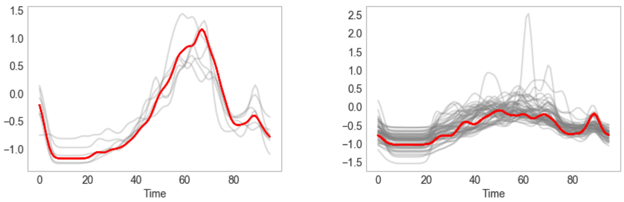
Figure 6 shows the clusters generated with data on weekends. We draw the curves on the first Saturday (2014-02-08) of all the stations in two clusters. Most of the stations belong to the second cluster, which shows a concave pattern with a relatively flat part in the middle (during daytime). This is because passengers are generally free at weekends and the departure time is no longer limited. Small fluctuations are also seen in the curves, which is caused by unpredictable behavior of passengers. Cluster 1 for weekends shows an interesting pattern, with peaks both in the evening and at midnight. Stations in this cluster are probably near entertainment venues like theaters and clubs.

In combination of both clusters generated from weekday and weekend entry rate curves, we have totally four clusters. If we denote the stations in weekday cluster i and weekend cluster j using a bracket as cluster (i, j), the combined clusters are (1,2), (2,2), (3,1), (3,2). It is noted that all stations in cluster 1 and 2 for weekday analysis belong to weekend cluster 2. This joint pattern



(a) Cluster 1 (b) Cluster 2 (c) Cluster 3

Fig. 5. Clusters generated with data on weekdays.



(a) Cluster 1 (b) Cluster 2

Fig. 6. Clusters generated with data on weekends.

reveals that stations in cluster (1,2) are near residential areas where people go out for work on weekday mornings and enjoy leisure time on weekends. Similarly, cluster (2,2) represents multifunctional areas which provide both living and working functions in the district. Also, all the stations who have late night peaks on weekends as cluster 1 are all rested in weekday cluster 3. This is in accordance with our analysis of weekend clusters from Figure 6. Cluster (3,2) gives a typical example of central business distinct where people leave after work in the evenings on weekdays. To this end, we manage to separate the stations into clusters that are easy to interpret while eliminating the influence of horizontal shift of the curves.

IV. Conclusion

In this study, we analyze the pattern of subway ridership via a clustering method. WDTW is applied to eliminate the influence of the curve deformations. Influence of day of the week is considered during the analysis. Experiments show that the clustering results are meaningful and interpretable. For further studies, the combination of clustering results and network information will help analyze the interactions and connections among stations inside a transportation system. Also, the cluster information can be possibly incorporated with prediction models for more accurate passenger flow forecasting and better travel demand management.

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. G. McNally, "The four-step model," in *Handbook of Transport Modelling: 2nd Edition*. Emerald Group Publishing Limited, 2007, pp. 35–53.

[3] A. J. Horowitz, "Simplifications for single-route transit-ridership forecasting models," *Transportation*, vol. 12, no. 3, pp. 261–275, 1984.

[4] Y. Zhao and K. M. Kockelman, "The propagation of uncertainty through travel demand models: an exploratory analysis," *The Annals of regional science*, vol. 36, no. 1, pp. 145–163, 2002.

[5] S. Chan and L. Miranda-Moreno, "A station-level ridership model for the metro network in montreal, quebec," *Canadian Journal of Civil Engineering*, vol. 40, no. 3, pp. 254–262, 2013.

[6] L.-F. Chow, F. Zhao, X. Liu, M.-T. Li, and I. Ubaka, "Transit ridership model based on geographically weighted regression," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1972, pp. 105–114, 2006.

[7] X. Ma, Y.-J. Wu, Y. Wang, F. Chen, and J. Liu, "Mining smart card data for transit riders travel patterns," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 1–12, 2013.

[8] C.-H. P. Tsai, C. Mulley, G. Clifton *et al.*, "Forecasting public transport demand for the sydney greater metropolitan area: A comparison of univariate and multivariate methods," *Road & Transport Research: A Journal of Australian and New Zealand Research and Practice*, vol. 23, no. 1, p. 51, 2014.

[9] Y. Wei and M.-C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 148–162, 2012.

[10] A. Dantas, K. Yamamoto, M. V. Lamar, and Y. Yamashita, "Neural network for travel demand forecast using gis and remote sensing," in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, vol. 4. IEEE, 2000, pp. 435–440.

[11] Y. Sun, B. Leng, and W. Guan, "A novel wavelet-svm short-time passenger flow prediction in beijing subway system," *Neurocomputing*, vol. 166, pp. 109–121, 2015.

[12] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, vol. 7, no. 3, pp. 358–386, 2005.

[13] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[14] V. Niennattrakul and C. A. Ratanamahatana, "On clustering multimedia time series data using k-means and dynamic time warping," in *Multimedia and Ubiquitous Engineering, 2007. MUE'07. International Conference on*. IEEE, 2007, pp. 733–738.

[15] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, vol. 44, no. 9, pp. 2231–2240, 2011.

[16] G. J. Szekely and M. L. Rizzo, "Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method," *Journal of classification*, vol. 22, no. 2, pp. 151–183, 2005.