IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 19, 2019, accepted December 4, 2019, date of publication December 9, 2019, date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2958378

Time-Series Representation and Clustering Approaches for Sharing Bike Usage Mining

DUO LI, YIFEI ZHAO, AND YAN LI
School of Highway, Chang'an University, Xi'an 710064, China

Corresponding author: Yan Li (lyan@chd.edu.cn)

ABSTRACT Massive bike-sharing systems (BSS) usage and performance data have been collected for years over various locations. Nevertheless, researchers encountered several challenges while dealing with massive BSS data. The challenges that could be enhanced in the previous studies are 1) reducing high dimensionality and noise of BSS time series data and 2) extracting informative usage patterns out of massive BSS data. This paper extracts patterns and reduce data dimensions of BSS usage by exploring time series representation and clustering of BSS usage data. A reduced dimension allows us to efficiently approximate the BSS usage with reasonable accuracy, which can be further used for bike usage clustering, classification and prediction. We employ a non-data adaptive representation technique -Discrete Wavelet Transform (DWT) to reduce dimensionality and filter out random errors of the raw time series. Time series are clustered using k-means based on similarities measured by Dynamic Time Warping (DTW) and prototypes computed using DTW barycenter averaging (DBA). The proposed approaches are applied on a 3-month bike usage dataset acquired on the BSS of Chicago. The analysis results show that DWT can effectively reduce dimensionality, filter out random errors and reveal the main characteristics of the raw time series. The clustering approach offers the ability to differentiate and discover bike usage patterns across different stations.

INDEX TERMS Sharing bike system, time series data mining, dynamic time warping (DWT), DTW barycenter averaging (DBA).

I. INTRODUCTION

As an affordable, convenient, and sustainable travel option with various benefits, BSSs have received increasing attention in the past decade. These systems offer the potential to decrease car usage in dense neighborhoods, promote healthy living, increase accessibility and provide much needed last-mile connection. BSS can be grouped into two categories, namely, dock-based and dockless systems. The former allows users to lock and unlock bicycles from docking stations distributed around the city, while the latter enables users to lock and unlock bicycles virtually anywhere. In the rest of this paper we will only be interested in dock-based BSS. The analysis of dockless BSS would likely only work if virtual station "areas" were created, which is beyond the scope of this research.

Compared against other forms of shared-use mobility, BSS has several unique features. First, unlike classic ride-sharing (e.g., carpooling) and ride-sourcing (e.g., Uber), bikes in BSSs are typically unattended. During vacant hours, bikes are stored at various stations where operations of check-in or check-out are performed via a backbone network. Second, bike-sharing differs from conventional public transit (e.g., subways and buses). The former provides transportation based on demand with a decentralized structure, while the latter is operated following a regular schedule and pre-determined routes [1]. Nevertheless, these two distinct features may result in characteristic issues while operating and managing BSSs. One common issue faced by BSSs is bike imbalance across stations, i.e., uncontrolled, nonstationary demand may cause an uneven bike distribution across different stations.

Various attempts have been made to alleviate bike imbalances, including demand prediction [2], [3] and inventory rebalancing [4], [5]. Here, inventory rebalancing is referred to as the operation that rebalances the bikes over time such that the proper number of bikes and open docks are available to users. Meanwhile, BSS data mining plays a key role while designing prediction models and redistribution strategies. Several studies [6]–[8] have shown that mining of BSS data was useful in extracting usage patterns, identifying

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

incentives for distribution of bikes and therefore obtaining a better understanding of BSS.

This research presented a Discrete Wavelet Transform (DWT) based data representation method and an integrated-time series clustering method that used Dynamic Time Warping (DTW), DTW barycenter averaging (DBA) and k-means for measuring similarity, computing prototype and partitioning data respectively. The proposed methods were verified against a 3-month bike usage dataset collected from the Chicago BSS.

The contributions of this work include: 1) to the best of our knowledge, this is the first work on dimension reduction of count series data. Such a data representation approach leads to significant reduction of noise and computational cost during the BSS analysis. 2) we propose a two-phase framework to process BSS count series data. In the data representation stage, data dimensionality is reduced, while still preserving the fundamental characteristics of a particular dataset. In the clustering stage, an integrated time series clustering method consisting of DTW, DBA and k-means is proposed to cluster stations over the network.

II. RELATED WORK

Papers related to analytics of BSS mainly focus on two objectives: the first is to identify relations between the demand and some factors, whereas the second is to predict the future demand of the system. The first papers always use generalized linear and generalized linear mixed models as their core statistical method. This might be because linear models are easy to interpret when compared against non-linear and non-parametric ones. The most common factors considered in these papers include 1) temporal factors (the hour, the week day, the month, etc) [8], [9], 2) meteorological factors (temperature, humidity, wind speed, etc) [10], 3) socio-demographic factors [11], and 4) neighborhood of the station [6], [7].

Papers in the second category always employ non-parametric statistical methods, such as non-homogeneous Poisson process [12], gradient boosted machines [13], neural networks [14], bayesian classifiers [15], etc. Although a few studies attempted to model the traffic per station [16], [17], the majority of the papers focus on the global prediction [18], [19]. The demand prediction per station problem is more challenging since the stochasticity is much more present. A few papers try to reduce the problem using clustering which can be performed geographically [20] or only using station behavior [21].

Although the study of BSS is increasing in popularity, previous papers leave several challenges that could be enhanced. The first is extracting informative usage patterns out of massive BSS data. Without the help of automatic algorithms which extract usage patterns and give a synthetic view of the information, it is difficult to acquire knowledge from the cluttered display and overlaps of hundreds of time series. The second is dimensionality reduction of BSS count series data. High dimension and noise are characteristics of BSS time series data. There are two reasons for reducing dimensionality before conducting further analysis: firstly, computing high dimensional raw data is computationally expensive, and dimension reduction can dramatically speed up the analysis process [22]. Secondly, because the high-dimensional data often contains a significant amount of noise, highly unintuitive results may be garnered by using raw time-series [23]. As a result, one may extract usage patterns which are similar in noise rather than extracting them based on similarity in shape. Dimension reduction can be achieved by transforming the raw time-series data into a lower dimensional space. This process is called "data representation".

III. TIME-SERIES DATA REPRESENTATION

Selecting a proper time series data representation method is greatly important since it impacts the accuracy and efficiency of the solution. Various representation techniques have been proposed to reduce data dimensionality, while still preserving the fundamental characteristics of a particular dataset, such as Discrete Fourier Transformation (DFT) [24], Single Value Decomposition (SVD) [24], Discrete Wavelet Transformation (DWT) [25], Piecewise Aggregate Approximation (PAA) [26], Adaptive Piecewise Constant Approximation (APCA) [27], Chebyshev polynomials (CHEB) [28], Symbolic Aggregate approXimation (SAX) [29] and etc.

In this study, we used a non-data adaptive method, DWT, to reduce the dimensionality of raw time series. DWT has the useful multiresolution property which makes it very appropriate for noise filtering, dimensionality reduction, and singularity detection. More specifically, wavelets are a set of mathematical functions that resemble the shape of time series. Therefore, the transformed data can be dispersed to multiple scales and analyzed at multiple resolution levels. The Haar wavelet, a sequence of rescaled "square-shaped" functions forming a wavelet family or basis, was selected as the mother wavelet for this study because: 1) it allows good approximation with a subset of coefficients, 2) it can be computed quickly and easily, and 3) it preserves Euclidean distance [30].

IV. TIME-SERIES CLUSTERING

There are mainly three components involved in time series clustering, namely, *similarity measure*, *prototype computation* and *clustering algorithm*. Usually, a *clustering algorithm* is employed to partition time-series based on similarities among time-series. The similarity is computed by a *similarity measure*. In the clustering process, several time series called "*prototypes*" are created to summarize the important characteristics of all series in given clusters.

A. SIMILARITY MEASURE

Euclidean Distance (ED) [24] and Dynamic Time Warping (DTW) [31] are the most widely used methods for measuring similarity in time-series clustering. ED is a relatively simple measure which compares two time-series $x = (x_1, x_2, \ldots, x_t)$

and $y = (y_1, y_2, \ldots, y_t)$ of length t as follows

$$\mathrm{ED}\,(\mathrm{x}, \mathrm{y}) = \sqrt{\sum\nolimits_{i=1}^{t} (x_i - y_i)^2} \tag{1}$$

DTW is considered as an extension of Euclidean distance that provides a local (non-linear) alignment. For given time series x and y of length t, a t-by-t matrix T is built, with the Euclidean distance between any two x_i and y_j. A *warping path* $E=\{e_1, e_2, \ldots, e_l\}$, with $t < l$, is a contiguous set of matrix elements that defines a mapping between x and y:

$$DTW\,(x, y) = min\sqrt{\sum\nolimits_{i=1}^{t} e_i} \tag{2}$$

The warping path can be calculated on matrix T with dynamic programming for the evaluation of the following recurrence:

$$r(i, j) = ED(i, j) + min\{r(i-1, j-1),\, r(i-1, j),\, r(i, j-1)\} \tag{3}$$

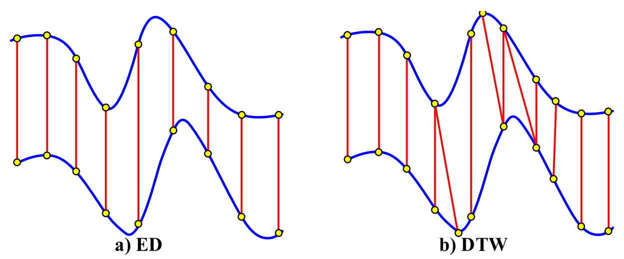


FIGURE 1. **Alignments between two-time series provided by a) ED and b) DTW.**

Fig. 1 shows the alignments between two-time series provided by ED and DTW. It can be observed that although the shapes of the two-time series are not aligned in the time axis, they have the similar overall shape. Due to its sensitivity to small distortions in the time axis, ED may fail to yield an intuitive measure of similarity. Whereas, DTW aims at identifying similar patterns of change regardless of the time axis, which makes it possible to produce a more intuitively correct measure of similarity. Thus, DTW is chosen as the similarity measure in this study.

B. PROTOTYPE COMPUTATION

Prototype computation is an essential subroutine in the clustering process. The performance of clustering algorithms, especially partitioning algorithms, relies highly on prototype computation functions, since the prototypes are used as cluster centroids. The *mean* and *partition around medoids (PAM)* are very commonly used prototype computation functions. The *mean* simply takes average of the time-series at each time point. So, it is often used in conjunction with the Euclidean distance measure. *PAM* calculates a representative object called ''medoid'' from a cluster, which minimizes the sum of squared distances to all other objects within the same cluster. Although both *mean* and *PAM* can work with a number of similarity measures, they do not take into account the unique feature of DTW. Thus, we adapted a DTW based prototyping function called DTW barycenter averaging (DBA) [32]. DBA is the most robust time series averaging method so far.

DBA is a global averaging method consisting in iteratively refining an initially averaging time series so as to minimize its squared DTW to other series. DBA starts with randomly choosing a time series as the initial averaging series. For each iteration, DBA calculates DTW between every individual series and the averaging series, so as to find associations between time-points of the averaging series and time-points of other series. Due to the warping path in DTW, it can be that several time-points from a given series map to a single time-point in the averaging series. Thus, for every time-point in the averaging series, corresponding values from all other series are grouped together based on the DTW alignments, consequently the mean value is calculated for every point in the averaging series using the values stored in every group. This process is iteratively repeated until it reaches convergence or a maximum number of iterations.

C. CLUSTERING ALGORITHM

Reviewing existing works in the literature, it is implied that partitioning and hierarchical algorithms [33] are broadly used in time series clustering. Other groups of algorithms for time series clustering include Grid-based (e.g., [34], [35]), Model-based (e.g., [36], [37]), Density-based (e.g., [38], [39]) and Multi-step clustering algorithms (e.g., [40], [41]), which have been comprehensively reviewed by [42].

Hierarchical clustering algorithms, as their name suggest, aim to construct a hierarchy of groups in which, with the increase of the level in the hierarchy, clusters are constructed by merging the lower level clusters, consequently an ordered sequence of groupings is produced. A hierarchical algorithm can be agglomerative or divisive. The former treats every time series as a cluster, and then gradually merges the clusters (bottom-up); while, the letter considers all series as one cluster and then gradually divides the cluster (top-down). Hierarchical algorithms suffer from a lack of flexibility since they cannot adjust the clusters once a split or merger has been performed.

Partitional clustering algorithms assign n unlabeled time series to k clusters through minimizing the intra-cluster similarity and the inter-cluster dissimilarity. Partitional clustering algorithms begin by randomly selecting k initial prototypes (centroids) that are assigned to individual clusters. Then the similarity between all time-series and prototypes are computed and each time-series is assigned to the cluster with its closest prototype. Similarity and prototype are updated iteratively until it meets a maximum number of iterations or convergence. The most prominent examples of such algorithm are k-means and k-Medoids (PAM). Prototype in the former is the mean of series in a given cluster; whereas, the latter uses the prototype provided by PAM. k-means was employed for this study, since we chose DBA, an averaging method, to compute prototype in the clustering process.

k-means requires three user-specified parameters: cluster initialization, similarity measure, and number of clusters *k*. Different initializations may result in different final clustering because *k*-means, which is a greedy algorithm, only reaches a local minimum. In order to overcome the local minima, we ran the *k*-means algorithm with 10 different initial partitions and chose the partition with the best performance. The number of clusters *k* is the most critical parameter in *k*-means. Davies–Bouldin index (DB) was chosen to determine the correct number of clusters.

V. CASE STUDY

Our main dataset was acquired on the BSS of Chicago. This dataset contains start station id, end station id, start timestamp, end timestamp and trip duration for each bike trip. We collected 3-month rental bike statistics spanning from 1st June to 31th August 2018. Totally 572 bike stations were included. Fig 2 and 3 shows demand distribution and daily profile of Chicago BSS.

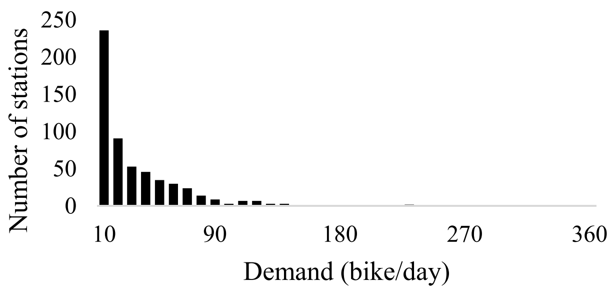


FIGURE 2. Demand distribution.

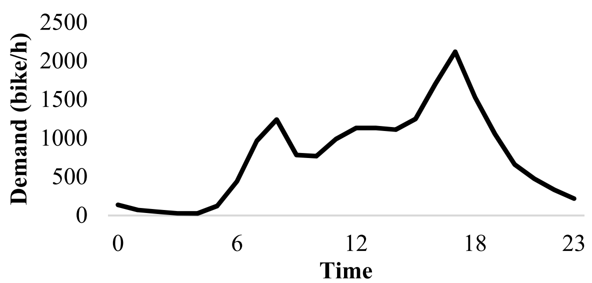


FIGURE 3. Daily demand profile.

Various studies observed that the bike usage exhibits different patterns on weekdays and weekends. We also observed that the bike usage demands were significantly different on weekdays and weekends. Weekdays' bike demands were more fluctuant and slightly higher than these on weekends. Thus, we analyzed these two scenarios separately. The results showed that the proposed method was effective in analyzing both scenarios. For conciseness, only the analysis on weekdays' demands is presented in this section. The data was aggregated at 60-min interval which was used by most paper for BSS data analysis. Note that due to the page limitation, we took a randomly chosen station (id: 85) for example.

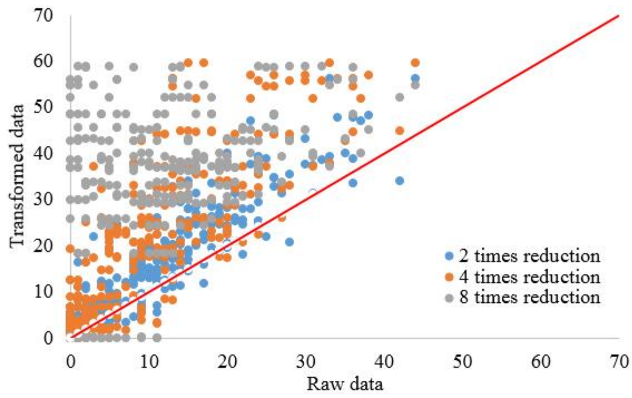


FIGURE 4. Transformed datasets against raw dataset.

The impact of different data resolution levels on clustering performance was analyzed. The transformed datasets data points were obtained by performing 2, 4 and 8 times dimension reduction respectively. As depicted in Fig. 4, the correlation between the transformed and raw datasets decreased with dimension reduction. The correlation efficient values were 0.86, 0.73 and 0.36 for 2, 4, and 8 times reduced datasets respectively. Therefore, 4 times dimension reduction is recommended, which can decrease computational cost while ensuring proper data representation and clustering performance.

We reduced the dimensionality of the raw dataset by 4 times using DWT, which means that bike usage of each day was presented by 6 data points in the transformed dataset instead of 24 data points in the raw dataset. To further investigate the impact of wavelets transformation on the time series, we decomposed the raw and transformed datasets using an additive model. For a given time series $(x_1, x_2, \ldots, x_i, \ldots, x_t)$, the addictive model is given by

$$x_i = S_i + T_i + R_i \tag{4}$$

where S_i is the seasonal component at time i, T_t is the trend component at time i,, and R_i is the random error component at time i.

To verify our selection, we compared DWT with two popular representation methods, namely, DFT and SAX. Tightness of lower bounds (TLB) that was considered as a meaningful measure in the literature [26], [28] was used to measure representation performance. Note that the value of TLB ranges between 0 and 1. Higher TLB value indicates better representation performance. Fig. 5 shows computation time and TLB values of the tested methods computed using 3-month data from the Chicago BSS system. It can be observed that DFT produced the worst representation performance among all the methods. There was very little difference between representations yielded by DWT and SAX. However, DFT and SAX required nearly 5 and 2 times longer computation time than DWT respectively, which means DWT can produce acceptable representation performance within relatively short computation time.

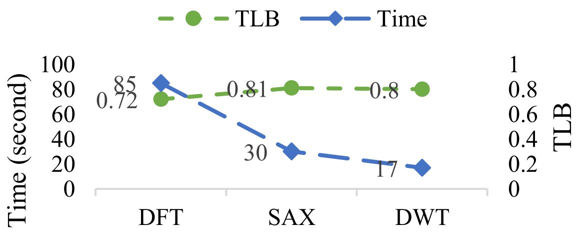
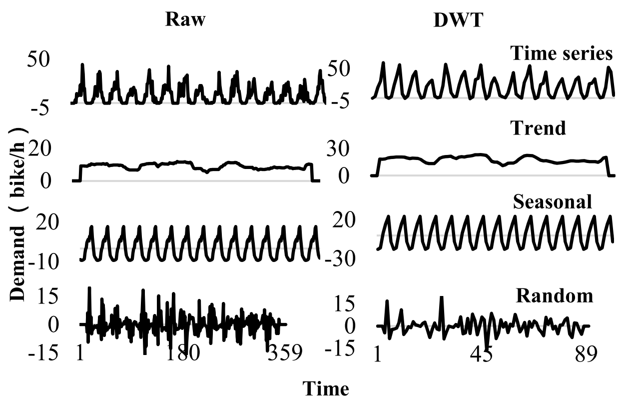


FIGURE 5. Time consumption and TLB values for different representation methods.



DWT series when compared against the raw series.

FIGURE 6. Decomposition plots of raw and DWT series.

Fig 6. shows the decomposition plots of both raw and DWT time series. The four plots are the original time series, seasonal component, trend component and random errors. It can be observed that there were certainly some seasonal variations in the daily bike usage which had the similar magnitude across time. The trend was relatively flat. The seasonal component shown that there was a peak in the daytime, and a trough in the night. The shape of the DWT series is nearly the same to that of the raw series. Meanwhile, DWT can efficiently capture typical profiles of the trend and seasonal variations of the raw series. It was noticed that DWT showed the ability to filter out random errors. A significant reduction in random errors was recorded in the DWT series when compared against the raw series.

k-means may produce different final clustering using different initializations. Thus, we ran k-means with 10 different initial partitions (seed 1-10) and adapted DB index to measure the clustering performance, as shown in Table 1. Note that the lower the DB value is, the better the separation of the clusters and the 'tightness' inside the clusters are. In this study, we selected the cases that produced the lowest DB values using the same number of clusters. Meanwhile, the cases yielding DB values that were above 2 were removed. The chosen cases were highlighted in Table 1.

Fig. 7 presents the final clustering of 2-cluster, 3-cluster and 4-cluster cases. Note that only the prototypes were plotted. As only few clusters included in these cases, the interpretation is straightforward. For 2-cluster case, data was divided into high and low demand time series. For 3-cluster case, data was divided into high, medium and low demand time series. And for 4-cluster case, data was divided into high, medium, stable-low and fluctuating-low demand series. In general, the overall demand level rather than seasonal variation is the main contributor to the final clustering in these cases due to the limited number of clusters.

A further analysis was performed on 5-cluster case. We zoomed in to daily profiles of each cluster. For each cluster, daily prototypes and their mean value were shown in Fig. 8. Time series in cluster 1 exhibited a low bike usage level which reached its peak at around 6 pm. Cluster 2 and 5 differed from each other in terms of demand level and peak

TABLE 1. Davies–Bouldin index values.

Cluster number	2	3	4	5	6	7	8	9	10
Seed1	1.86	2.47	2.23	3.55	2.48	2.87	2.3	2.88	2.85
Seed2	1.45	2.09	2.34	2.42	2.1	3.13	3.07	2.88	2.84
Seed3	2.99	1.4	1.87	2.37	3.37	3.25	2.59	2.67	2.92
Seed4	2.99	2.47	2.25	2.24	3.85	2.91	2.57	3.34	3.11
Seed5	1.45	2.2	2.46	3.53	3.13	3.25	2.58	2.19	2.61
Seed6	2.99	2.03	3.77	1.74	2.98	2.77	2.7	3.17	2.65
Seed7	1.85	2.47	2.61	2.57	2.15	3.37	2.65	2.66	2.31
Seed8	1.86	2.3	2.71	2.47	2.62	2.25	3.62	1.99	3.12
Seed9	1.86	2.3	2.29	2.86	2.29	2.4	2.47	3.42	3.24
Seed10	1.86	2.3	2.68	2.6	2.1	2.88	2.67	2.72	3.21

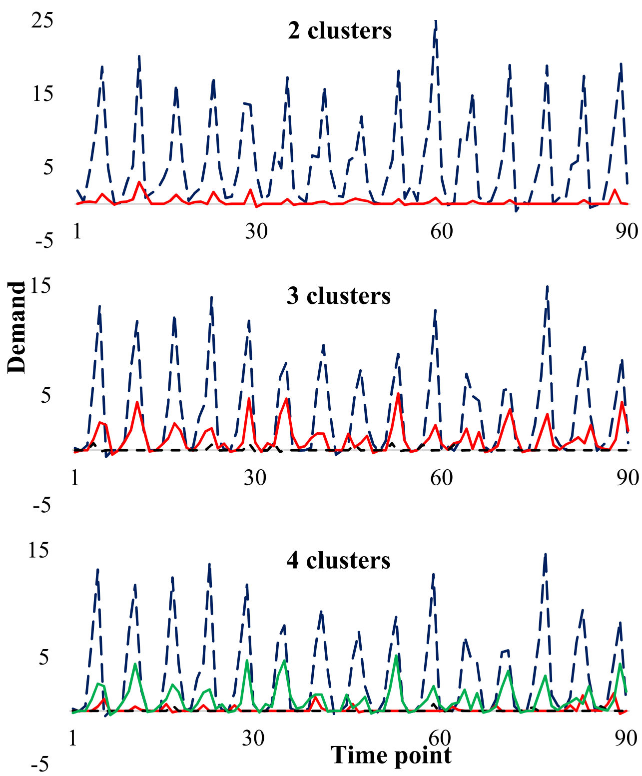


FIGURE 7. Final clustering of 2-4 cluster cases.

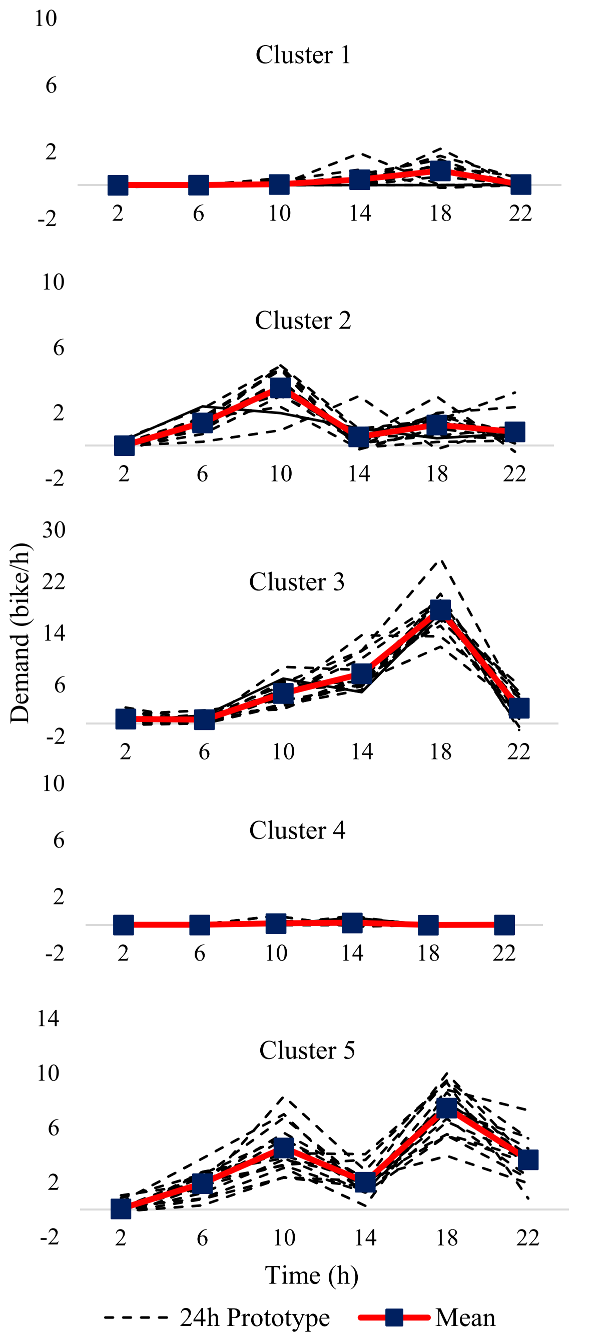


FIGURE 8. daily prototypes and their mean values for 5-cluster case.

hours: The former had a relatively low demand level and was more active during morning peak hours; while there was a higher demand in cluster 5 in which stations were busier during evening peak hours. Stations in cluster 3 were the most active ones compared against other stations. The usage of these stations began to increase around 6 am until 6 pm. In contrast, the bike demand level in cluster 4 was extremely low and there was few change across time.

Although we selected DTW, DBA and k-means as similarity measure, prototype computation and clustering algorithm respectively, it is still interesting to test different combinations on the dataset. Table 2 shows the DB values computed using different combinations for 5-cluster case. Here, the "mean" is the mean value across time series in the same cluster. PAM is a time series whose average distance to all other objects in the same cluster is minimal. The k-Shape clustering algorithm was developed by Paparrizos and Gravano [43] in 2015. It is a partitional clustering algorithm with a custom distance measure – shape based distance (SBD), as well as a custom centroid function – shape extraction (SE). Note that only the lowest DB values among 10 runs are shown in Table 2. It should also be note that k-means relies on the prototypes generated by averaging methods (e.g., mean and DBA); whereas, k-medoids can only uses the prototypes provided by PAM. The combination used in this study was highlighted (in green). It can be observed that the hierarchical algorithm produced the highest DB values indicating the worst performance among all the tested algorithms. Clustering results computed using DTW were better than those using ED in most cases. DBA based clustering outperformed PAM based clustering. Although k-shape yielded a slightly lower DB value (in red) when compared with k-means, the difference between them was not remarkable. It is worth mentioning that k-shape required about 5 times longer computational time than k-means. In general, the results revealed that the combination used in study can provide a cost-effective solution in terms of clustering performance and computation time.

TABLE 2. DB values computed using different algorithms.

		Hierarchical	k-means	k-medoids	k-shape
Mean	ED	2.18	1.81	-	-
	DTW	2.19	1.77	-	-
PAM	ED	2.21	-	1.78	-
	DTW	2.20	-	1.75	-
DBA	ED	2.05	1.76	-	-
	DTW	2.02	1.74	-	-
SE	SBD	-	-	-	1.72

TABLE 3. DB values computed using different resolution levels.

No. of clusters	2	3	4	5
2 times reduction	1.43	1.42	1.77	1.76
4 times reduction	1.45	1.4	1.87	1.74
8 times reduction	1.48	1.45	1.86	1.91

Table 3 shows the lowest DB values among 10 runs computed using the transformed datasets. It is shown that although the dataset reduced by 8 times witnessed the worst performance among all the cases, the impact of dimension reduction on clustering performance was not remarkable, i.e., the clustering performance resulted from lower dimensional datasets can approach that of higher dimensional datasets.

VI. SUMMARY AND CONCLUSION

To achieve a better understanding of BSS usage and performance, various attempts have been made to investigate the data collected from BSS. Nevertheless, researchers encountered several challenges while dealing with massive BSS data. The challenges that could be enhanced in the previous studies are 1) reducing high dimensionality and noise of BSS time series data and 2) extracting informative usage patterns out of massive BSS data.

To address these issues, this study explored time series representation and clustering approaches for BSS usage mining. We employed a non-data adaptive representation technique - Discrete Wavelet Transform (DWT) to reduce dimensionality and filter out random errors of the raw time series. Time series were clustered using k-means based on similarities measured by Dynamic Time Warping (DTW) and prototypes computed using DTW barycenter averaging (DBA). The proposed approaches were applied on a 3-month bike usage dataset acquired on the BSS of Chicago.

The obtained clustering results are useful in two aspects. Firstly, the demand prediction per station problem is challenging due to its stochasticity. A possible strategy to predict the demand is to reduce the problem using clustering. Secondly, the clustering results can provide insights that are helpful in improving BSS service level requirements.

Following conclusions can be drawn based on the results presented in the previous section:

- DWT can effectively reduce dimensionality, filter out random errors and emphasize the main characteristics of the raw time series.
- The combined approach offered the ability to differentiate and discover bike usage patterns across different stations
- Final clustering was mainly contributed by the overall demand level in the cases with fewer clusters. However, when more clusters were involved, both demand level and daily usage profile were main contributors to final clustering.

The approaches proposed in this study focus on univariate time series– rented bike count series in our case. A single time-series object can be constituted of several values that change on the same time scale, e.g., time series containing both rented and returned bike usage information. It might be useful to develop a multivariate clustering approach that takes into account both rented and returned bike count series so as to further explore the BSS usage and performance in the future research. Bike usage variations may occur daily, weekly, monthly, or quarterly. Thus, another future work will focus on multi-seasonal time series feature extraction and clustering.

REFERENCES

[1] Z. Yang, H. Ji, Y. Shu, C. Peng, and T. Moscibroda, "Mobility modeling and prediction in bike-sharing systems," presented at the 14th Annu. Int. Conf., 2016.

[2] R. Giot and R. Cherrier, "Predicting bikeshare system usage up to one day ahead," in *Proc. IEEE Symp. Comput. Intell. Vehicles Transp. Syst. (CIVTS)*, Dec. 2014, pp. 22–29.

[3] S. Ruffieux, E. Mugellini, and O. A. Khaled, "Bike usage forecasting for optimal rebalancing operations in bike-sharing systems," in *Proc. IEEE 30th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2018, pp. 854–858.

[4] F. Chiariotti, C. Pielli, A. Zanella, and M. J. S. Zorzi, "A dynamic approach to rebalancing bike-sharing systems," *Sensors*, vol. 18, no. 2, pp. 512–534, 2018.

[5] J. Schuijbroek, R. C. Hampshire, and W.-J. van Hoeve, "Inventory rebalancing and vehicle routing in bike sharing systems," *Eur. J. Oper. Res.*, vol. 257, no. 3, pp. 992–1004, Mar. 2016.

[6] A. Faghih-Imani, S. Anowar, E. J. Miller, and N. Eluru, "Hail a cab or ride a bike? A travel time comparison of taxi and bicycle-sharing systems in New York City," *Transp. Res. A Policy*, vol. 101, pp. 11–21, 2017.

[7] A. Faghih-Imani and N. Eluru, "Incorporating the impact of spatio-temporal interactions on bicycle sharing system demand: A case study of New York CitiBike system," *J. Transp. Geography*, vol. 54, pp. 218–227, 2016.

[8] K. Gebhart and R. B. Noland, "The impact of weather conditions on bikeshare trips in Washington, DC," *Transp. Res. A Policy*, vol. 41, no. 6, pp. 1205–1225, 2014.

[9] S. Wagner, T. Brandt, and D. Neumann, "In free float: Developing business analytics support for carsharing providers," *Omega*, vol. 59, 2016, Art. no. S0305048315000481.

[10] B. Caulfield, M. Omahony, W. Brazil, and P. Weldon, "Examining usage patterns of a bike-sharing scheme in a medium sized city," *Transp. Res. Part A Policy*, vol. 100, pp. 152–161, 2017.

[11] A. Faghihimani, R. Hampshire, L. Marla, and N. Eluru, "An empirical analysis of bike sharing usage and rebalancing: Evidence from Barcelona and Seville," *Transp. Res. A Policy Pract.*, vol. 97, pp. 177–191, 2017.

[12] R. Alvarez-Valdes, "Optimizing the level of service quality of a bike-sharing system," *Omega*, vol. 62, pp. 162–175, 2016.

[13] R. Regue and W. Recker, "Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem," *Transp. Res. E*, vol. 72, pp. 192–209, 2014.

[14] R. L. Cheu, J. Xu, A. G. H. Kek, W. P. Lim, and W. L. Chen, "Forecasting shared-use vehicle trips with neural networks and support vector machines," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1968, no. 1968, pp. 40–46, 2006.

[15] L. Cagliero, T. Cerquitelli, S. Chiusano, P. Garza, and X. Xiao, "Predicting critical conditions in bicycle sharing systems," *Computing*, vol. 99, no. 1, pp. 1–19, 2017.

[16] N. Lathia, S. Ahmed, and L. Capra, "Measuring the impact of opening the London shared bicycle scheme to casual users," *Transp. Res. C-Emerg. Technol.*, vol. 22, pp. 88–102, 2012.

[17] D. Li and Y. Zhao, "A multi-categorical probabilistic approach for short-term bike sharing usage prediction," *IEEE Access*, vol. 7, pp. 81364–81369, 2019.

[18] P. Borgnat, P. Abry, P. Flandrin, C. Robardet, J. B. Rouquier, and E. Fleury, "Shared bicycles in a city: A signal processing and data analysis perspective," *Adv. Complex Syst.*, vol. 14, no. 3, 2011, Art. no. 1100295.

[19] W. Jia, Y. Tan, and J. Li, "Hierarchical prediction based on two-level affinity propagation clustering for bike-sharing system," *IEEE Access*, vol. 6, pp. 45875–45885, 2018.

[20] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," *Adv. Geographic Inf. Syst.*, p. 33, 2015.

[21] P. Vogel, *Service Network Design of Bike Sharing Systems*. 2016.

[22] E. J. Keogh and M. J. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," *Databases Knowl. Inf. Syst.*, vol. 1805, no. 1, pp. 122–133, 2000.

[23] A. Hinneburg and D. A. Keim, "Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering," in *Proc. Int. Conf. Very Large Data Bases*, 1999.

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

[25] K. Chan and A. W. Fu, "Efficient time series matching by wavelets," in *Proc. Int. Conf. Data Eng.*, 1999, pp. 126–133.

[26] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, ''Dimensionality reduction for fast similarity search in large time series databases,'' *Knowl. Inf. Syst.*, vol. 3, no. 3, pp. 263–286, 2001.

[27] E. Keogh, K. Chakrabarti, M. J. Pazzani, and S. Mehrotra, ''Locally adaptive dimensionality reduction for indexing large time series databases,'' in *Proc. Int. Conf. Manage. Data*, 2001, vol. 30, no. 2, pp. 151–162.

[28] Y. Cai and R. T. Ng, ''Indexing spatio-temporal trajectories with Chebyshev polynomials,'' in *Proc. Int. Conf. Manage. Data*, 2004, pp. 599–610.

[29] J. Lin, E. Keogh, L. Wei, and S. Lonardi, ''Experiencing SAX: A novel symbolic representation of time series,'' *Data Mining Knowl. Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[30] K. P. Chan and W. C. Fu, ''Efficient time series matching by wavelets,'' in *Proc. Int. Conf. Data Eng.*, 1999.

[31] H. Sakoe and S. Chiba, ''Dynamic programming algorithm optimization for spoken word recognition,'' *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[32] F. Petitjean, A. Ketterlin, and P. Gançarski, ''A global averaging method for dynamic time warping, with applications to clustering,'' *Pattern Recognit.*, vol. 44, no. 3, pp. 678–693, Mar. 2011.

[33] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 2009.

[34] G. Sheikholeslami, S. Chatterjee, and A. Zhang, ''WaveCluster: A multi-resolution clustering approach for very large spatial databases,'' Tech. Rep., 1998.

[35] W. Wei, J. Yang, and R. R. Muntz, ''STING: A statistical information grid approach to spatial data mining,'' presented at the Proc. 23rd Very Large Database Conf., 1997.

[36] D. H. Fisher, ''Knowledge acquisition via incremental conceptual clustering,'' *Mach. Learn.*, vol. 2, no. 2, pp. 139–172, 1987.

[37] T. Kohonen, ''The self-organizing map,'' *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[38] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, ''OPTICS: Ordering points to identify the clustering structure,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1999.

[39] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, ''A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise,'' in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1996.

[40] S. Aghabozorgi, W. T. Ying, T. Herawan, H. A. Jalab, M. A. Shaygan, and A. Jalali, ''A hybrid algorithm for clustering of time series data based on affinity search technique,'' *Sci. World J.*, vol. 2014, no. 3, 2014, Art. no. 562194.

[41] C.-P. Lai, P.-C. Chung, and V. S. Tseng, ''A novel two-level clustering method for time series data analysis,'' *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6319–6326, 2010.

[42] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, ''Time-series clustering—A decade review,'' *Inf. Syst.*, vol. 53, pp. 16–38, 2015.

[43] J. Paparrizos and L. Gravano, ''K-shape: Efficient and accurate clustering of time series,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 69–76.

DUO LI received the B.S. degree in civil engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2010, the M.S. degree in civil engineering from the University of Queensland, Brisbane, QLD, Australia, in 2011, and the Ph.D. degree in traffic engineering from the University of Auckland, Auckland, New Zealand, in 2015.

Since 2018, he has been a Humboldt Fellow with the Institute of Traffic Systems, German Aerospace Center (DLR). Since 2015, he has been a Lecturer with the Highway School, Chang'an University.

YIFEI ZHAO received the B.S., M.S., and Ph.D. degrees from Chang'an University, Xi'an, China, in 1990, 1998, and 2009, respectively.

From 1990 to 2003, he was a Lecturer with the Highway School, Chang'an University, where he has been an Associate Professor, since 2003.

YAN LI received the B.S. and Ph.D. degrees from Southeast University, Nanjing, China, in 2006 and 2011, respectively.

From 2011 to 2013, he was a Lecturer with the Highway School, Chang'an University, where he has been an Associate Professor, since 2013. He was a Visiting Scholar with the University of California at Davis, from 2017 to 2018.