Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 562194, 12 pages
http://dx.doi.org/10.1155/2014/562194

Research Article

A Hybrid Algorithm for Clustering of Time Series Data Based on Affinity Search Technique

Saeed Aghabozorgi, Teh Ying Wah, Tutut Herawan, Hamid A. Jalab, Mohammad Amin Shaygan, and Alireza Jalali

Faculty of Computer Science & Information Technology Building, University of Malaya, 50603 Kuala Lumpur, Malaysia

Correspondence should be addressed to Saeed Aghabozorgi; saeed@um.edu.my

Received 4 October 2013; Accepted 2 February 2014; Published 25 March 2014

Academic Editors: H. Chen, P. Ji, and Y. Zeng

Time series clustering is an important solution to various problems in numerous fields of research, including business, medical science, and finance. However, conventional clustering algorithms are not practical for time series data because they are essentially designed for static data. This impracticality results in poor clustering accuracy in several systems. In this paper, a new hybrid clustering algorithm is proposed based on the similarity in shape of time series data. Time series data are first grouped as subclusters based on similarity in time. The subclusters are then merged using the k-Medoids algorithm based on similarity in shape. This model has two contributions: (1) it is more accurate than other conventional and hybrid approaches and (2) it determines the similarity in shape among time series data with a low complexity. To evaluate the accuracy of the proposed model, the model is tested extensively using syntactic and real-world time series datasets.

1. Introduction

Clustering is considered the most important unsupervised learning problem. The clustering of time series data is particularly advantageous in exploratory data analysis and summary generation. Time series clustering is also a preprocessing step in either another time series mining task or as part of a complex system. Researchers have shown that using well-known conventional algorithms in the clustering of static data, such as partitional and hierarchical clustering, generates clusters with an acceptable structural quality and consistency and is partially efficient in terms of execution time and accuracy [1]. However, classic machine learning and data mining algorithms are ineffective with regard to time series data because of the unique structure of time series, that is, its high dimensionality, very high feature correlation, and (typically) large amount of noise [2–4]. Accordingly, numerous research efforts have been conducted to present an efficient approach to time series clustering. However, the focus on the efficiency and scalability of these methods in handling time series data has come at the expense of losing the usability and effectiveness of clustering [5].

The clustering of time series data can be broadly classified into conventional approaches and hybrid approaches. Conventional approaches employed in the clustering of time series data are typically partitioning, hierarchical, or model-based algorithms. In hierarchical clustering, a nested hierarchy of similar objects is constructed based on a pairwise distance matrix [6]. Hierarchical clustering has great visualization power in time series clustering [7]. This characteristic has made hierarchical clustering very suitable for time series clustering [8, 9]. Additionally, hierarchical clustering does not require the number of clusters as an initial parameter, in contrast to most algorithms. This characteristic is a well-known and outstanding feature of this algorithm and is a strength point in time series clustering because defining the number of clusters is often difficult in real-world problems. However, hierarchical clustering is cumbersome when handling large time series datasets [10] because of its quadratic computational complexity. As a result of its poor scalability, hierarchical clustering is restricted to small datasets. On the other hand, partitioning algorithms, such as the well-known k-Means [11] or k-Medoids algorithm [12], are among the most used algorithms in this domain.

k-Means and k-Medoids algorithms are very fast compared with hierarchical clustering [13], making them very suitable for time series clustering. Therefore, these algorithms have been used in several works, either in their "crispy" manner [3, 14–18] or in their "fuzzy" manner (Fuzzy c-Means and Fuzzy c-Medoids) [17–20]. Model-based clustering assumes a model for each cluster and determines the best data fit for that model. The model obtained from the generated data defines the clusters [21]. A few articles use model-based time series clustering [22–26]; however, two typical drawbacks have been discovered. First, the parameters should be set, and the parameter setting is based on the user's assumptions, which may be false and may result in inaccurate clusters. Second, model-based clustering has a slow processing time (especially neural networks) with respect to large datasets [27].

Aside from all of these conventional approaches, some new articles emphasize the enhancement of algorithms and present customized models (typically as a hybrid method) for time series data clustering. One of the latest works is an article by Lai et al. [28], who describe the problem of overlooked information as a result of dimension reduction. Lai et al. claim that the overlooked information can result in time series clustering results that have a different meaning. To solve this issue, they adopt a two-level clustering method, where both the whole time series and the subsequence of the time series are considered in the first and second levels, respectively. Lai et al. employed Symbolic Aggregate ApproXimation (SAX) [29] transformation as a dimension reduction method and the Cluster Affinity Search Technique (CAST) [30] as a first-level clustering algorithm to group first-level data. To measure distances between time series data in the second level, Dynamic Time Warping (DTW) [31] was used on data with varying lengths, and Euclidean distance (ED) was used on data of equal length. However, CAST algorithm is used twice in this approach, once to generate initial clusters and the other to split each cluster into subclusters, which is rather complex.

The authors in [32] also propose a new multilevel approach for shape-based time series clustering. First, time series data are selected from a generated one-nearest-neighbor network. To generate the time series network, the authors propose a triangle distance measurement to calculate the similarity between time series data. Hierarchical clustering is then performed on the selected time series data. Second, the data size is reduced by approximately 10% using this approach. This algorithm requires a nearest-neighbor network in the first level. The complexity in generating a nearest-neighbor network is $O(n^2)$, which is rather high. As a result, the authors attempt to reduce the search area by data preclustering (using k-Means) and limit the search to each cluster only to reduce the creation network. However, generating the network itself remains costly, rendering it inapplicable in large datasets. Additionally, the solution to the challenge of generating the prototypes via k-Means when the triangle is used as a distance measure is unclear.

In this study, the low quality problem in existing works is addressed by the proposal of a new Two-step Time series Clustering (TTC) algorithm, which has a reasonable complexity. In the first step of the model, all the time series data are segmented into subclusters. Each subcluster is represented by a prototype generated based on the time series affinity factor. In the second step, the prototypes are combined to construct the ultimate clusters.

To evaluate the accuracy of the proposed model, TTC is tested extensively using published time series datasets from diverse domains. This model is shown to be more accurate than any of the existing works and overcomes the limitations of conventional clustering algorithms in determining the clusters of time series data that are similar in shape. With TTC, the clustering of time series data based on similarity in shape does not require calculation of the exact distances among all the time series data in a dataset; instead, accurate clusters can be obtained using prototypes of similar time series data.

The rest of this paper is organized as follows. In Section 2, some concepts and definitions are explained. In Section 3, the proposed model is described. In Section 4, the algorithm is applied on diverse time series datasets and the experimental results are analyzed. In Section 5, conclusions are drawn and future perspectives are discussed.

2. Concepts and Definitions

The key terms used in this study are presented in this section. The objects in the dataset related to the problem at hand are time series data of similar lengths.

Definition 1 (time series). A time series $F_i = \{f_1, \ldots, f_t, \ldots, f_n\}$ is an ordered set of numbers that indicate the temporal characteristics of objects at any time t of the total track life T [33].

Definition 2 (time series clustering). Given a dataset of N objects, $D = \{F_1, F_2, \ldots, F_N\}$, where F_i is a time series. The unsupervised partitioning process of D into $C = \{C_1, C_2, \ldots, C_k\}$ occurs such that homogenous time series data are grouped together based on similarity in shape, a grouping that is called time series clustering. C_i is then called a cluster, where $D = \bigcup_{i=1}^{k} C_i$ and $C_i \cap C_j = \emptyset$, for $i \neq j$.

Definition 3 (similarity in time). The similarity between two time series data is based on the similarity in each time step.

Definition 4 (similarity in shape). The similarity between two time series is based on the similarities between their subsequences or their common trends regardless of time occurrence.

Definition 5 (subcluster). A subcluster SC_i is a set of individual time series data that are similar in time and are represented as a single prototype. Time series data are attached to a new subcluster based on their affinity to the subcluster. Thus, $V = \{\mathrm{SC}_1, \mathrm{SC}_2, \ldots, \mathrm{SC}_i, \ldots, \mathrm{SC}_M\}$ is the set of all subclusters, where $k < M << N$.

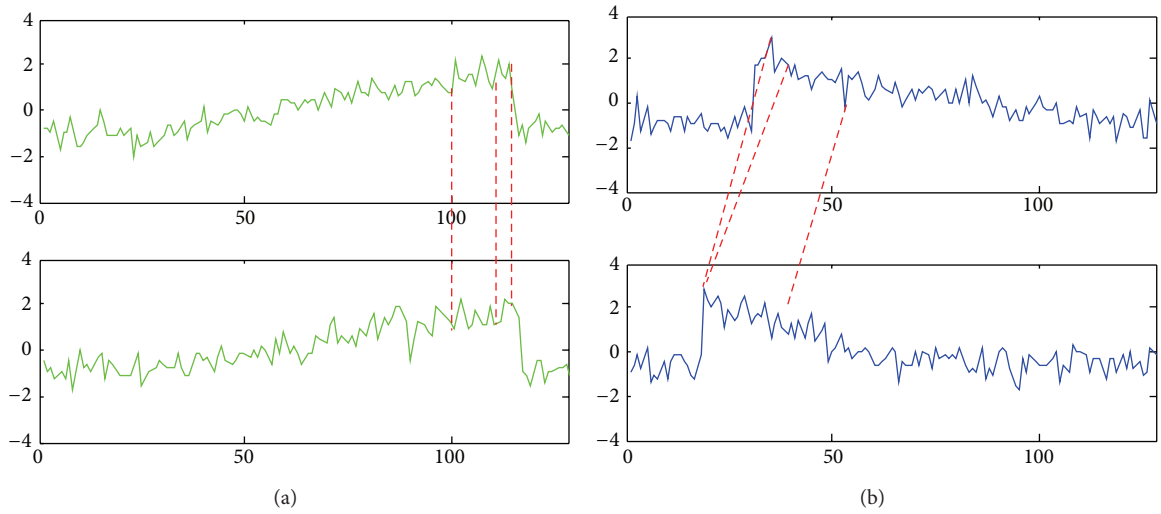

Figure 1: Similarity in shape (b) and similarity in time (a) between two time series data.

Definition 6 (affinity). The affinity of a time series F_x with a subcluster SC_i is defined as follows:

$$a_i(F_x) = \frac{\sum_{y \in SC_i} A_{xy}}{|SC_i|}, \quad (1)$$

where A_{xy} is the similarity between time series F_x and F_y and $|SC_i|$ is the number of time series data that exist in the subcluster SC_i. This value is used to distinguish the time series data that have a low affinity by placing them into a new subcluster.

Definition 7 (prototype). The prototype is a time series $R_i = \{r_1, \ldots, r_x, \ldots, r_n\}$, which represents the most typical time point of a finite set of time series data in subcluster SC_i. The prototype of each subcluster is constructed with regard to the affinity of each time series with the subcluster.

Time series clustering relies highly on a distance measure. Several distance measures have been proposed by researchers in the literature [34–42]. However, ED and DTW are revealed to be the most common methods used in time series clustering because of the efficiency of ED and the effectiveness of DTW in similarity measurement. Simple and fast, ED is used as benchmark in numerous studies (approximately 80%) [34, 43–45] because it is parameter-free. However, it is not the best choice as a distance function because it is extremely dependent on the domain of the problem at hand and the dataset's time series characteristics. In fact, ED is very weak and sensitive to slight shifts across the time axis [46–49], which limits it in terms of determining time series data that are *similar in time*.

In contrast to ED, which proposes one-to-one matching between time points, DTW is suggested as a one-to-many measurement. DTW is a generalization of ED, which solves the local shift problem in the time series data to be compared (see Figure 1). The local shift problem is a time scale issue that characterizes most time series data. Handling local shifts allows similar shapes to be matched even if they are out of phase in the time axis; that is, they are *similar in shape*.

Using this definition, time series clusters with similar patterns of change are constructed regardless of time points, for example, to cluster share prices related to different companies that have a common stock pattern independent of time series occurrence [22, 50]. DTW is thus superior to ED [31, 39, 41, 51, 52], as the latter can only determine time series that are similar in time.

DTW "warps" the time axis to achieve the best alignment between data points within the series. Dynamic programming is generally used to effectively determine the warping path. However, warping causes a scalability problem that requires quadratic computation, which is a huge challenge for DTW [53]. However, we do not need to calculate all of the distances when the proposed algorithm previously mentioned is used; therefore, DTW can be adopted without affecting clustering efficiency.

3. The Proposed Algorithm

The detailed description of the proposed algorithm is presented in this section. Figure 2 shows the block diagram for the proposed TTC algorithm. First, the size of the time series dataset is reduced (i.e., data reduction) using the concept of affinity. A prototype is then generated for each subcluster. Consequently, subclusters are merged using k-Medoids clustering.

According to the steps above, the activities of the TTC are explained in the following sections.

3.1. Step 1: Data Reduction. The main objective of this TTC step is to reduce the size of the dataset by defining a prototype for each group of very similar time series data, which significantly decreases the complexity of TTC. The time series data are first standardized using z-score (z-normalization) [54], which causes the time series data to be invariant to scale

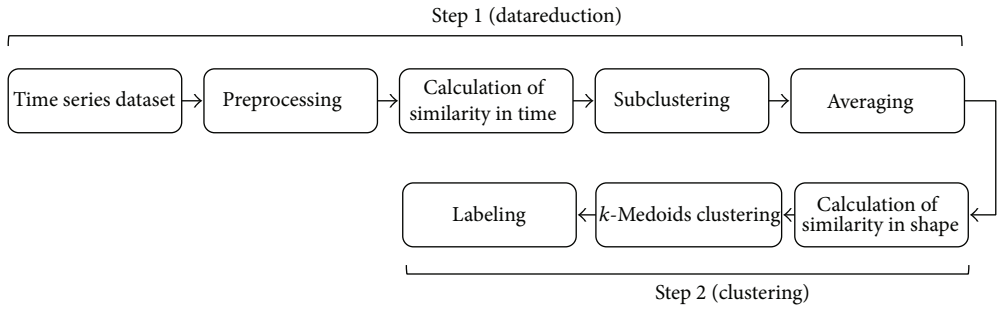

Figure 2: Block diagram for the proposed TTC algorithm.

and offset. Supposing that $F_i = \{f_1, .., f_t, .., f_n\}$ is a time series with T data points, z-normalization is defined as

$$z\text{-Normalization}\left(F_i, \mu_i, \text{sd}\right) = \frac{f_t - \mu_i}{\text{sd}}, \tag{2}$$

where

$$\mu_i = \frac{\sum_{t=1}^{n} f_t}{n}, \qquad \text{sd} = \sqrt{\frac{\sum_{t=1}^{n} f_t \left(f_t - \mu_i\right)^2}{n}}, \tag{3}$$

where μ_i is an arithmetic mean of data points f_1 through f_n and sd is the standard deviation of all the data points in the given time series.

Subsequently, all the data are clustered as a whole based on similarity in time. In this step, the affinity search technique concept in CAST [30] is borrowed to generate the subclusters. CAST was essentially introduced into the bioinformatics domain for gene expression clustering; it is used in this step because the number of clusters does not need to be predetermined in CAST. In contrast to numerous algorithms that require the number of clusters to be predefined in advance, the mechanism used by the CAST algorithm can determine clusters dynamically and deal effectively with outliers. CAST works based on the pairwise similarity matrix of objects. The similarities between time series data are calculated and stored in an N-by-N similarity matrix ($A_{N\times N}$), where A_{ij} is the similarity between time series F_i and time series F_j. ED is used as the dissimilarity measure to calculate the similarity (similarity in time) between time series data. Figure 3 illustrates the reasoning behind the use of ED to construct subclusters in the first step. $A'_{N\times N}$ is assumed to be the pairwise distance matrix, where A'_{ij} is the Euclidian distance between F_i and F_j. This distance is mathematically defined as

$$A'_{ij} = \text{dis}_{\text{ED}}\left(F_i, F_j\right) = \sqrt{\sum_{i=1}^{n} \left(f_i - f_j\right)^2}, \tag{4}$$

where the square root step can be removed because the square root function is monotonic and reverts to the same rankings in clustering [2]. The time complexity of this calculation can also be reduced from linear to constant by caching some

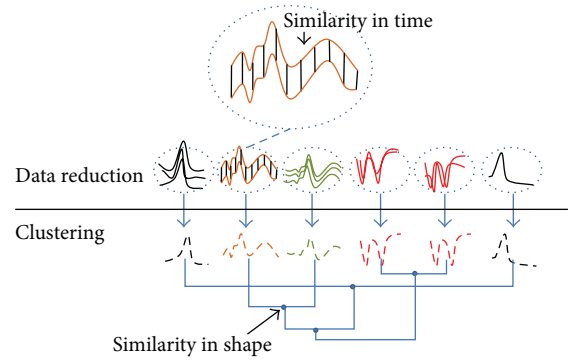

Figure 3: Reasoning behind the use of DTW in calculating similarity in shape between the prototypes of subclusters in the second step of TTC.

of the calculated results [55]. Given $A_{N\times N}$, the algorithm is performed by adding and removing time series data from a subcluster based on a threshold affinity value between 0 and 1, as defined by the user.

A new subcluster (Definition 5) is constructed by the time series datum that has the highest similarity to other time series data. Subsequently, each time series datum is added to a new subcluster based on its affinity with the subcluster (Definition 6); that is, each subcluster is constructed with a time series datum and is gradually completed by the addition of new time series data to the subcluster based on the average similarity (affinity) between the unassigned time series data and the current subcluster members. As previously mentioned, subclusters are formed sequentially with an affinity threshold. By defining the specific threshold value, the cluster accepts the high affinity time series datum. The affinity threshold α is specified to determine what is considered significantly similar. This parameter controls the number and sizes of the produced subclusters. After a subcluster is formed, CAST deletes the low affinity objects from the subcluster. The process of adding to and removing from a subcluster is performed consecutively until no further changes occur in the subcluster.

After each subcluster is constructed, a prototype is defined for each subcluster. The construction of an effective time series prototype is a vexing problem [56, 57]. In the current study, we propose a novel approach to represent time

series data in a cluster. The prototype of each subcluster is calculated based on the affinity of each time series datum with the subcluster. An affinity set is maintained during the subclustering process for all the time series data, denoted as a_i (Definition 6). The affinity of a time series datum evidently implies its weight in the construction of the prototype. Given the subcluster SC_i, its prototype is defined by a time series $R_i = \{r_1, \ldots, r_x, \ldots, r_n\}$. r_x is then calculated as

$$r_x = \frac{\sum_{y \in SC_i} a_i\left(F_y\right) * f_{yx}}{|SC_i|}, \quad (5)$$

where $F_y = \{f_{y1}, \ldots, f_{yx}, \ldots, f_{yn}\}$ is a time series datum in CS_i and $|SC_i|$ indicates the number of time series data in the subcluster.

3.2. Step 2: Clustering. In the first step, the time series data are grouped based on the similarity in time. However, two time series data that are not similar in time may be similar in shape. Similarity in shape is desirable in time series clustering because the constructed clusters are very close to the ground truth and are more meaningful. However, the methods that have this feature, such as DTW, are often costly [53] in the similarity evaluation of time series data. As a result, several researchers, such as [47, 58–61], try to accelerate the process, typically by proposing efficient lower bound approximations of DTW distance to reduce its complexity. However, most of these works are under the classification problem (the search area is pruned using a lower bound distance of DTW) and are not suitable for several clustering algorithms, where the dissimilarity matrix must be fully calculated. For example, in clustering algorithms such as k-Medoids or Unweighted Pair-Group Method with Arithmetic Mean [62], all distances must be calculated and no pruning can be performed. In such cases, the clustering process benefits from a fast and accurate similarity measure [63]. However, we do not have to calculate similarity in shape between all time series data in the TTC because the very close time series data (similar in time) are similar in shape as well. That is, the dissimilarity matrix does not need to be fully calculated using an expensive similarity measure such as DTW. As a result, only a small part of the matrix is calculated by DTW using the prototypes of the subclusters, which are small in size, in the first step of TTC (instead of all the data as a whole). Figure 3 depicts the reasoning behind the use of ED and DTW in the first and second steps of TTC, respectively. As this figure shows, the intersimilarity between the time series data in the subclusters is computed based on similarity in time, and intrasimilarity is calculated based on similarity in shape.

Therefore, the similarity between subclusters is calculated and stored in an M-by-M similarity matrix $B_{M\times M}$, where B_{ij} is the similarity between the prototypes of subclusters SC_i and SC_j. First, DTW distance among the prototypes of the subclusters is calculated to construct the pairwise dissimilarity matrix $B_{M\times M}$, where B_{ij} is the DTW distance of two subclusters' prototypes, namely, prototype R_i and prototype R_j, as denoted by dis_{DTW}. Suppose that $R_x = \{r_{x1}, \ldots, r_{xi}, \ldots, r_{xn}\}$ is the prototype of SC_x, where n is the length of the prototype and r_x is calculated by (5). To compute the distance between the prototypes of SC_x and SC_y, an $n \times n$ matrix is constructed for the distance of all pairs as $Z(R_x, R_y)$, where $Z_{i,j} = \text{dis}_{\text{ED}}(r_{xi}, r_{yj})$ and $\text{dis}_{\text{ED}}(\,)$ is the Euclidean distance. Given $W = \{w_1, w_2, \ldots, w_u\}$ as a set of warping paths, where $w_u = \{(r_{x1}, r_{y1}), (r_{xi}, r_{yj}), \ldots, (r_{xn}, r_{yn})\}$ is a set of points that define a traversal of matrix Z and the DTW between the two prototypes R_x and R_y is a warping path that minimizes the distance between R_x and R_y,

$$\text{dis}_{\text{DTW}}\left(R_x, R_y\right) = \min\left(\sum_{u=1}^{U} \frac{W_u}{U}\right), \quad (6)$$

where $(r_{x1}, r_{y1}) = (1, 1)$ and $(r_{xn}, r_{yn}) = (n, n)$ and $0 \le r_{xi+1} - r_{xi} \le 1$ and $0 \le r_{yj} - r_{yj+1} \le 1$, for all $i < n$.

Given the pairwise dissimilarity matrix, different schemes can be used for clustering. k-Medoids, which has been shown to be effective in the time series clustering domain [29–33], is selected. The TTC algorithm is presented in Pseudocode 1.

4. Analysis

4.1. Evaluation Metrics and Experimental Setup. The experiment on the proposed model is conducted with one syntactic dataset and 12 real-word datasets obtained from the UCR Time Series Data Mining Archive in various domains and sizes [64]. This set is selected because it is composed of various numbers of clusters with different cluster shapes and density, contains noise points, and is used as a benchmark in several articles in previous literature.

The well-known three-class Cylinder-Bell-Funnel (CBF) dataset is used as a syntactic dataset in the experiment on 2PTC with large datasets. The CBF dataset is an artificial dataset that has temporal domain properties and was originally proposed by Saito in 2000. This dataset has been used in numerous works [32, 65, 66]. It includes three types of time series data: Cylinder (c), Bell (b), and Funnel (f). Different CBF datasets are generated and used in this study. Examples of CBF time series datasets are shown in Figure 4.

In general, evaluating extracted clusters (patterns) is not easy in the absence of data labels [66]. However, all the selected datasets in this study have class labels (ground truth) and can be applied in evaluating TTC using external indices. The most commonly used external indices in the time series clustering domain are used in evaluating the accuracy of TTC, namely, Rand Index and Entropy. (The interested reader may refer to [67, 68] for definitions.)

Rand Index is a popular quality measure [69–71] for evaluating time series clusters; it measures the agreement between two partitions, that is, how close clustering results are to the ground truth. The agreement between cluster C and ground truth G can be estimated using

$$\text{RI}(C, G) = \sqrt{\frac{|\text{TP}| + |\text{TN}|}{|\text{TP}| + |\text{TN}| + |\text{FP}| + |\text{FN}|}}, \quad (7)$$

where |TP| (True Positive) is the number of pairs belonging to one class in G (ground truth) and are clustered together in C. |TN| (True Negative) is the number of pairs that neither

Method: (D, α, K)
Input: D: the set of time-series $D = \{F_1, F_2, \ldots, F_n\}$
α: Affinity threshold
K: cluster number
Output: C: set of clusters
/*Step 1. time-series data reduction*/
(1) $D = z$-norm(D) /*z-normalization of all time-series */
(2) $A[N][N] \leftarrow$Smilarity$_{\mathrm{ED}}$ $(\widehat{D})$ /* Calculate disimilarity array based on (5)*/
(3) (SC [1 to M], a [1 to N]) $\leftarrow$ CAST(A, α) /* M is determined automatically by CAST*/
(4) **for** $i = 1$ to M; M is the number of sub-clusters
(5) $r \leftarrow$ Average (SC$_i$, a [1 to N]) /* summarize the cluster */
(6) $R \leftarrow R \cup r$ /*R: a collection of prototypes */
(7) **end for**
/*Step 2. clustering*/
(8) $B[M][M] \leftarrow$Smilarity$_{\mathrm{DTW}}(R)$
(9) $C' \leftarrow k$-medoids(k, B)
(10) $C \leftarrow$ labels(C')validate the MTC clustering results
(11) return C

Pseudocode 1: Pseudocode related to TTC.

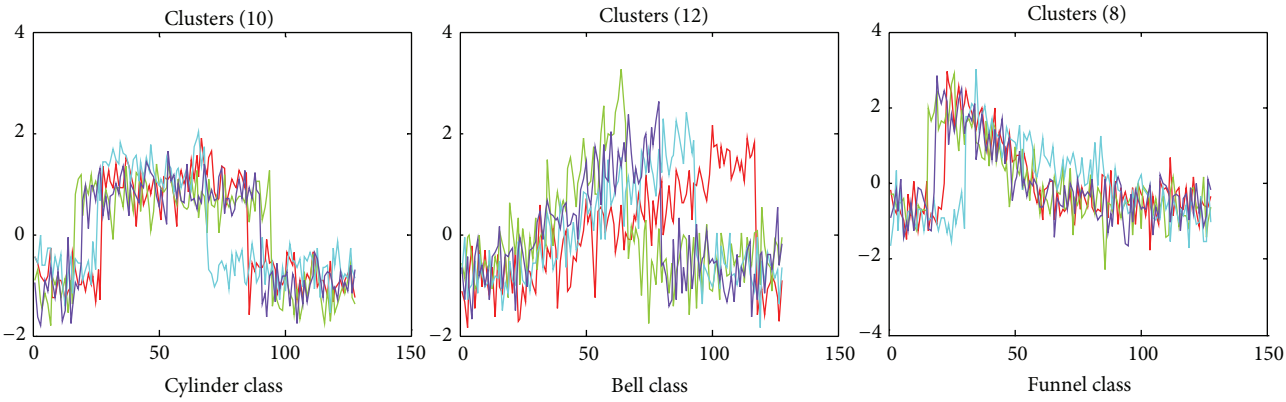

Figure 4: Three samples of each Cylinder, Bell, and Funnel (CBF) class dataset.

belong to the same class in G nor are clustered together in C. The types of error clustering are the |FN| (False Negative), which is the number of pairs that belong to one class in G but are not clustered together in C, and |FP| (False Positive), which is the number of pairs that do not belong to one class in G (dissimilar time series) but are clustered together in C. The Random Index evaluation criteria have values ranging from 0 to 1, where 1 corresponds to the case wherein ground truth and clustering result are identical and 0 corresponds to the case wherein they are completely different.

The Entropy [72, 73] of a cluster shows the dispersion of classes within a cluster (this dispersion should be low) in several domains. Entropy has been adopted in the evaluation of time series clustering in literature [74, 75] as well. It is a function of the distribution of classes in the resulting clusters. For each cluster C_j, the class distribution of data is computed as the probability $\Pr(G_i \mid C_j)$, wherein an instance in C_j belongs to class G_i. Using this class distribution, the normalized entropy of C_j is computed as

$$\mathrm{Entropy}\left(C_j\right) = -\frac{1}{\log h}\sum_{i=1}^{h}\Pr\left(\left(G_i \mid G_j\right) \times \log\left(G_i \mid G_j\right)\right), \tag{8}$$

where $\Pr(G_i \mid C_j) = |C_j \cap G_i|/|C_j|$. ConEntropy is the converse of Entropy based on the definition of Entropy, wherein ConEntropy is 1 when the ground truth and the clustering result are identical. Overall ConEntropy ($E \in [0, 1]$) is defined as the sum of the individual cluster entropies weighted by the size of each cluster:

$$\mathrm{ConEntropy}\,(C, G) = 1 - \frac{1}{|D|}\sum_{j=1}^{K}\left|C_j\right| \times \mathrm{Entropy}\left(C_j\right). \tag{9}$$

Table 1: Quality of TTC approach against the standard k-Medoids with regard to raw time series data and the time series data represented by PAA.

Dataset	Number of classes	DS size	Length	k-Medoids (ED)		k-Medoids (PAA-ED)		TTC	
				RI	ConEnrtropy	RI	ConEnrtropy	RI	ConEnrtropy
50words	50	455	270	0.95	0.73	0.95	0.73	0.96	0.79
Adiac	37	391	176	0.94	0.58	0.94	0.58	0.96	0.64
CBF	3	900	128	0.67	0.29	0.7	0.41	0.88	0.78
Coffee	2	28	286	0.8	0.6	0.78	0.54	0.86	0.68
ECG200	2	100	96	0.61	0.18	0.61	0.18	0.55	0.14
FaceFour	4	88	350	0.77	0.51	0.77	0.51	0.91	0.78
FISH	7	175	463	0.77	0.32	0.8	0.32	0.83	0.48
SwedishLeaf	15	625	128	0.9	0.54	0.9	0.54	0.9	0.58
synthetic_control	6	300	60	0.82	0.56	0.87	0.79	0.92	0.95
Trace	4	100	275	0.75	0.51	0.78	0.59	0.85	0.78
Two_Patterns	4	4000	128	0.63	0.03	0.69	0.32	0.86	0.88
Wafer	2	6164	152	0.43	0.01	0.43	0.01	0.5	0.21

Based on the measures above, a good clustering solution is expected to have high ConEntropy. To avoid a biased evaluation, the conclusions are drawn based on the average value of the indices. Although the focus of this study is improving the accuracy of TTC, the scalability of the proposed model is also calculated to prove its theoretical feasibility.

4.2. Accuracy Evaluation. In this section, the results are compared with those of partitional clustering. First, the distance between the time series data is calculated using ED to compare TTC with conventional k-Medoids. The reader may wonder why DTW is not used to compare the results. In simple terms, the use of DTW does not result in a fair comparison because DTW is not practically feasible in the real world as a result of its very high complexity. The complexity of DTW in between each pair of time series data in k-Medoids is $O(Ik(N-k)^2$ and $O(n^2)$, where N is the number of time series data, k is the number of clusters, I is the number of iterations required for convergence, and n is the length of time series data. Therefore, the total computation of k-Medoids is $O(Ik(N-k)^2 \cdot n^2)$. That is, $N(N-1)/2$ distance calculation is required to calculate the confusion matrix alone (needed in clustering), where N is the number of time series. As a result, the complexity of the distance matrix alone (not the entire clustering process) equals $N(N-1)n^2/2$, which is very high. For example, given $N = 1000$ and $n = 152$ in a dataset, the number of instruction executions is 11,540,448,000. However, using TTC on the same process requires approximately 177,084,440 executions because the process operates on a fraction of the entire dataset with the reduction factor = 0.1 (see (11)).

As a fair comparison in the subsequent experiment, the raw time series data are represented by a representation method because time series data are represented by a representation method prior to clustering in the majority of previous literature. Numerous studies focusing on the representation or dimensionality reduction of time series data have been conducted [7, 55, 59]. Among these representation methods, each of which has its strong points and weaknesses, Piecewise Aggregate Approximation (PAA) [52, 75] is adopted in this study because of its strength in the representation of time series data and its low complexity [76]. The raw time series data are represented using different compression ratios ($\text{compression}_{\text{ratio}} = [4, 6, 8]$) because the accuracy of PAA itself depends on the number of segmentations. As a result, the mean of three accuracies for each dataset is calculated as the average accuracy of k-Medoids. Table 1 shows the quality of the TTC approach against quality of the k-Medoids with regard to raw time series data and the time series data represented by PAA.

As expected, the comparison of TTC with *k-Medoids (ED)* and *k-Medoids (PAA-ED)* shows that TTC is more accurate in most of the datasets. TTC outperforms *k-Medoids (ED)* because ED cannot handle the local shifts in time series data, which decreases the accuracy of the final clusters.

Furthermore, TTC is more accurate than the conventional k-Medoids on represented time series, that is, *k-Medoids (PAA-ED)*. Although several outliers and noises in raw time series data are handled in the time series represented by PAA, the proposed algorithm, namely, TTC, remains superior to *k-Medoids (PAA-ED)* because of its shift-handling mechanism. The result shows that improved clustering quality is obtainable without reducing the time series dimension by using the prototypes of very similar time series data. This result is the proof of the researcher's claim that the TTC model can outperform conventional algorithms using either raw time series data or dimensionality reduction approaches.

4.3. Comparing TTC with Hybrid Models. As mentioned in related works, one of the novel works close to the proposed model in this study is the two-level approach proposed by Lai et al. [28] called the 2LTSC. In Lai et al.'s work, SAX transformation is used as a dimension reduction method, and CAST is used as the clustering algorithm in the first level. In the second level, DTW is used to calculate distances between time series data with varying lengths, and ED is used to calculate distances between data of equal length. The

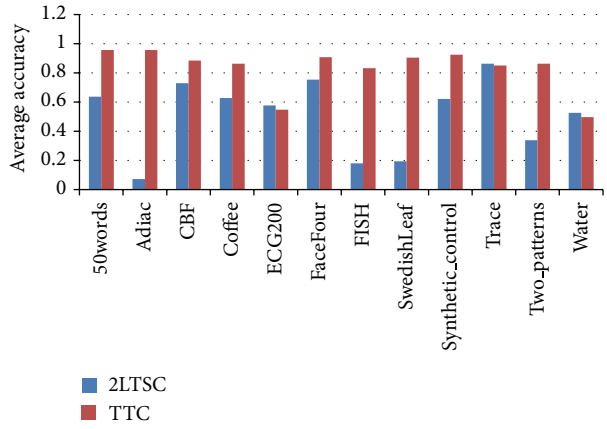

Figure 5: Comparison of 2LTSC and TTC against ground truth using the test datasets.

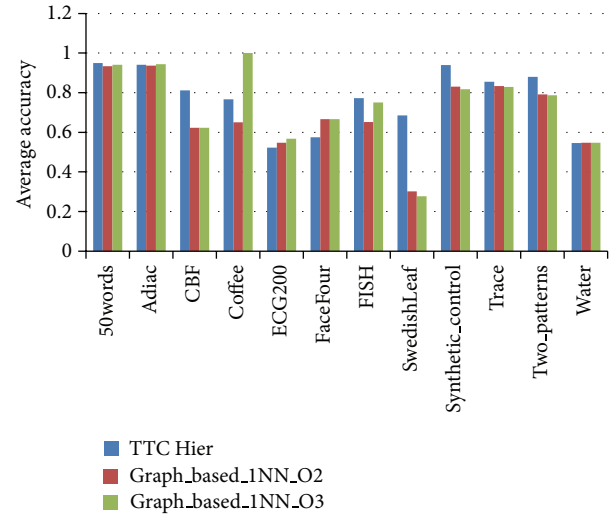

Figure 6: Quality of clustering using the graph-based approach as compared with TTC.

2LTSC works with the CAST algorithm, wherein the number of clusters is indirectly determined by a threshold. Hence, the same number of clusters (generated by 2LTSC) is also used in TTC after the 2LTSC is run. Figure 5 shows the best clustering result of both approaches.

As mentioned, a high-resolution time series is used in the TTC model, which is superior to the dimensionality reduced time series used in 2LTSC. As a result, the quality of TTC is increased after clustering occurs in the second level. The subclusters are merged in the second step of TTC, which causes the generated cluster structure to be more similar to the ground truth.

Another study that performed clustering in more than one step is Zhang et al. [32], which was discussed in the literature review. As previously mentioned, Zhang et al. proposed a new multilevel approach for shape-based time series clustering, wherein several candidate time series data are selected and clustered. To compare this approach (called the graph-based approach) with the TTC model, the quality of TTC clustering in terms of different orders of the nearest-neighbor network is calculated and shown in Figure 6. To provide fair conditions, the order of two to three is considered in the graph-based approach, which provides a reasonable reduction in the second layer.

As the result shows, TTC is superior to the graph-based algorithm in some datasets. The graph-based approach notably requires the generation of a nearest-neighbor graph, which is costly. However, the graph-based approach can be advantageous in datasets where similarity in time is essentially very important, such as the Coffee dataset (as shown in Figure 6). To summarize, the proposed model, namely, TTC, can outperform rival approaches, even with lower time complexity.

4.4. Data Reduction. To verify the effect of data reduction on final clustering, some experiments are conducted. In this experiment, we calculate the error rate in the data reduction step based on the CAST parameter, that is, the affinity threshold. Different sizes of the syntactic dataset CBF are used in this experiment.

First, a parameter is defined as reduction factor R_{factor}

$$R_{\text{factor}} = \frac{M}{N}, \tag{10}$$

where N is the size of the dataset and M is the number of subclusters generated by CAST (referred to as the number of prototypes).

The error rate E_{rate} of the subclusters is calculated based on the number of items in the same subcluster that belongs to the same class (ground truth) [77]. Given $G = \{G_1, G_2, \ldots, G_M\}$ as ground truth clusters and $V = \{\text{SC}_1, \text{SC}_2, \ldots, \text{SC}_i, \ldots, \text{SC}_M\}$ as the subclusters generated by CAST, the subcluster SC_i is assigned to the class most frequently found in the cluster to compute the error rate of cluster SC_i with respect to G. The error rate of this assignment is then measured by counting the number of misclassified time series data and dividing the result by the number of time series data in the subcluster. M is assumed to be the number of subclusters determined by CAST, and the size of cluster C_i is shown by $|\text{SC}_i|$. $\max(|\text{SC}_i \cap G_j|)$ is assumed to denote the number of items in subcluster SC_i that are not in G_j. The error rate of cluster SC_i is then given by

$$E_{\text{rate}}(\text{SC}_i) = \frac{1}{|\text{SC}_i|} \max\left(\left|\text{SC}_i \cap G_j\right|\right). \tag{11}$$

Given N as the size of the dataset, the overall error rate of the reduction step can be expressed as a weighted sum of individual cluster error rates:

$$E_{\text{rate}} = \sum_{i=1}^{M} \frac{|\text{SC}_i|}{N} E_r(\text{SC}_i). \tag{12}$$

As previously mentioned, the affinity threshold in the TTC algorithm determines the size and shape of sub clusters. If the value of the threshold is high, sub clusters are denser and the number of prototypes increases. As a result, the reduction

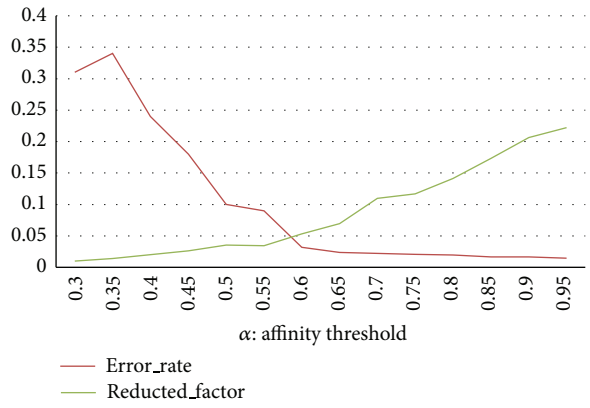

Figure 7: Reduction factor and error rate of the TTC approach across affinity threshold values for the CBF dataset.

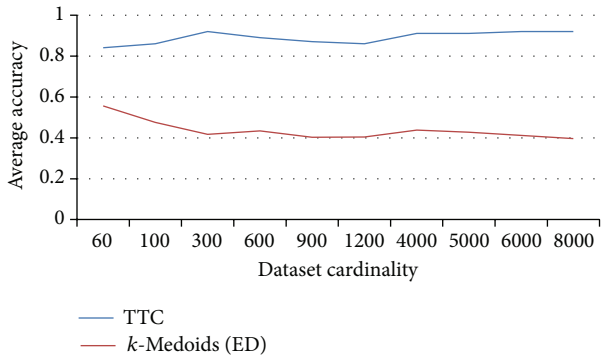

Figure 8: Accuracy of TTC across different CBF dataset sizes.

factor increases. Figure 7 shows the reduction factor and error rate of TTC across different affinity threshold values. The result shows that a good trade-off between reduction factor and error rate is obtained in thresholds above 0.7 for both datasets. As the threshold value increases, a lower error rate is encountered. The number of subclusters also increases (and reduction factor is higher). The following experiment verifies that TTC can reduce the data size by approximately 77% ($R_{\text{factor}} = 0.23$). The effectiveness of TTC is not significantly reduced; that is, the error rate is less than 0.05.

4.5. Evaluation of TTC on Large Datasets. To confirm the effectiveness of TTC further, some experiments are conducted on large synthetic datasets. For this purpose, up to 8,000 CBF time series are generated. To evaluate the results of the proposed model on large datasets, the average accuracy of TTC with regard to different CBF data sizes is shown in Figure 8. The experiment on TTC was also conducted with respect to different numbers of subclusters. This experiment shows the accuracy of TTC on large datasets. The average accuracy of TTC with respect to different numbers of subclusters is shown in Figures 8 and 9.

As the result shows, the quality of TTC is superior to that of other algorithms. The quality of TTC reaches 90% (Figure 9) in most of the cardinalities of the dataset when 30

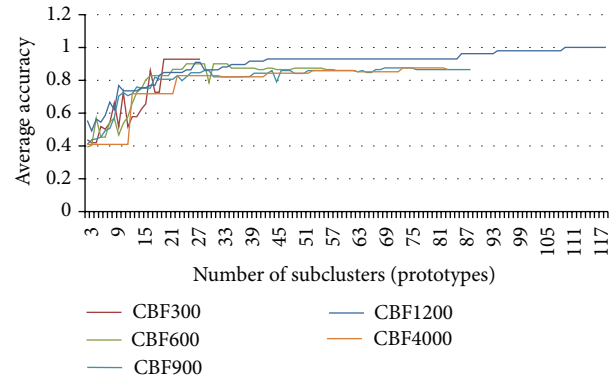

Figure 9: Accuracy of TTC across different numbers of subclusters.

subclusters are used. The maximum accuracy of conventional approaches is approximately 50% (Figure 8). The trend shows an increase in quality as the size of the dataset increases (Figure 8). Therefore, the use of DTW is not necessary in the clustering of all the data in very large datasets; it can be applied to smaller sets of a time series subset represented by prototyping instead.

5. Conclusion and Future Works

We illustrated the advantages of using some time series data as prototypes to cluster time series data based on the similarity in shape. We proposed a two-step clustering approach and showed its usage. The results obtained by applying TTC to different datasets were evaluated extensively. Clustering can be applied to a large time series dataset to generate accurate clusters. In the experiments with various datasets, different evaluation methods were used to show that TTC outperforms other conventional and hybrid clustering. Currently, we are working on a multistep approach, which is very scalable in the clustering of very large time series datasets. This approach will be performed as an anytime split-and-merge algorithm to present early results to the user and thus improve the clusters.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by High Impact Research Grant, University of Malaya, no vote UM.C/628/HIR/MOHE/SC/13/2, from the Ministry of Higher Education, Malaysia.

References

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 316–323, 1999.

[2] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, 2003.

[3] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulos, "Iterative incremental clustering of time series," in *Advances in Database Technology—EDBT 2004*, pp. 106–122, 2004.

[4] S. Rani and G. Sikka, "Recent techniques of clustering of time series data: a survey," *International Journal of Computational and Applied*, vol. 52, no. 15, pp. 1–9, 2012.

[5] C. Ratanamahatana, "Multimedia retrieval using time series representation and relevance feedback," in *Proceedings of the 8th International Conference on Asian Digital Libraries (ICADL '05)*, pp. 400–405, 2005.

[6] M. Vlachos, J. Lin, and E. Keogh, "A wavelet-based anytime algorithm for k-means clustering of time series," in *Proceedings of the Workshop on Clustering High Dimensionality Data and Its Applications*, pp. 23–30, 2003.

[7] E. Keogh and M. Pazzani, "An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback," in *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*, pp. 239–241, 1998.

[8] T. Oates, M. D. Schmill, and P. R. Cohen, "A method for clustering the experiences of a mobile robot that accords with human judgments," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 846–851, 2000.

[9] S. Hirano and S. Tsumoto, "Empirical comparison of clustering methods for long time-series databases," in *Active Mining*, vol. 3430, pp. 268–286, 2005.

[10] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data Mining and Knowledge Discovery*, vol. 13, no. 3, pp. 335–364, 2006.

[11] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.

[12] L. Kaufman, P. J. Rousseeuw, and E. Corporation, *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 39, Wiley Online Library, 1990.

[13] P. S. Bradley, U. Fayyad, and C. Reina, "Scaling clustering algorithms to large databases," in *Proceedings of the 4th International Conference on Knowledge Discovery & Data Mining (KDD '98)*, pp. 9–15, 1998.

[14] C. Guo, H. Jia, and N. Zhang, "Time series clustering based on ICA for stock data analysis," in *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 1–4, 2008.

[15] V. Hautamaki, P. Nykänen, and P. Fränti, "Time-series clustering by approximate prototypes," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, 2008.

[16] C. A. Ratanamahatana and V. Niennattrakul, "Clustering multimedia data using time series," in *Proceedings of the International Conference on Hybrid Information Technology (ICHIT '06)*, pp. 372–379, 2006.

[17] D. Tran and M. Wagner, "Fuzzy c-means clustering-based speaker verification," in *Advances in Soft Computing—AFSS 2002*, vol. 2275, pp. 318–324, 2002.

[18] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 375–381, 2003.

[19] S. Aghabozorgi, T. Y. Wah, A. Amini, and M. R. Saybani, "A new approach to present prototypes in clustering of time series," in *Proceedings of the 7th International Conference of Data Mining*, vol. 28, pp. 214–220, 2011.

[20] M. Ji, F. Xie, and Y. Ping, "A dynamic fuzzy cluster algorithm for time series," *Abstract and Applied Analysis*, vol. 2013, Article ID 183410, 7 pages, 2013.

[21] J. W. Shavlik and T. G. Dietterich, *Readings in Machine Learning*, Morgan Kaufmann, 1990.

[22] A. Bagnall and G. Janacek, "Clustering time series with clipped data," *Machine Learning*, vol. 58, no. 2-3, pp. 151–178, 2005.

[23] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 719–725, 2000.

[24] M. Ramoni, P. Sebastiani, and P. Cohen, "Multivariate clustering by dynamics," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 633–638, 2000.

[25] M. Bicego, V. Murino, and M. Figueiredo, "Similarity-based clustering of sequences using hidden Markov models," in *Machine Learning and Data Mining in Pattern Recognition*, vol. 2734, pp. 86–95, 2003.

[26] J. Hu, B. Ray, and L. Han, "An interweaved HMM/DTW approach to robust time series clustering," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 145–148, 2006.

[27] B. Andreopoulos, A. An, X. Wang, and M. Schroeder, "A roadmap of clustering algorithms: finding a match for a biomedical application," *Briefings in Bioinformatics*, vol. 10, no. 3, pp. 297–314, 2009.

[28] C.-P. P. Lai, P.-C. C. Chung, and V. S. Tseng, "A novel two-level clustering method for time series data analysis," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6319–6326, 2010.

[29] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards parameter-free data mining," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, vol. 22, pp. 206–215, 2004.

[30] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *Journal of Computational Biology*, vol. 6, no. 3-4, pp. 281–297, 1999.

[31] S. Chu, E. Keogh, D. Hart, M. Pazzani, and Michael, "Iterative deepening dynamic time warping for time series," in *Proceedings of the 2nd SIAM International Conference on Data Mining*, pp. 195–212, 2002.

[32] X. Zhang, J. Liu, Y. Du, and T. Lv, "A novel clustering method on time series data," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11891–11900, 2011.

[33] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: experimental studies and comparative evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 312–319, 2009.

[34] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '94)*, vol. 23, pp. 419–429, 1994.

[35] H. Sakoe and S. Chiba, "A dynamic programming approach to continuous speech recognition," in *Proceedings of the 7th International Congress on Acoustics*, vol. 3, pp. 65–69, 1971.

[36] 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.

[37] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the 30th International Conference on Very Large Data Bases*, vol. 30, pp. 792–803, 2004.

[38] J. Aßfalg, H. P. Kriegel, P. Kröger, P. Kunath, A. Pryakhin, and M. Renz, "Similarity search on time series based on threshold queries," in *Advances in Database Technology—EDBT 2006*, pp. 276–294, 2006.

[39] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proceedings of the 18th International Conference on Data Engineering*, pp. 673–684, 2002.

[40] A. Banerjee and J. Ghosh, "Clickstream clustering using weighted longest common subsequences," in *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining*, pp. 33–40, 2001.

[41] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '05)*, pp. 491–502, 2005.

[42] E. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S.-H. Lee, and J. Handley, "Compression-based data mining of sequential data," *Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. 99–129, 2007.

[43] E. Keogh, "Fast similarity search in the presence of longitudinal scaling in time series databases," in *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence*, pp. 578–584, 1997.

[44] F. K.-P. Chan, A. W.-C. Fu, and C. Yu, "Haar wavelets for efficient similarity search of time-series: with and without time warping," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 3, pp. 686–705, 2003.

[45] E. Keogh, M. Pazzani, K. Chakrabarti, and S. Mehrotra, "A simple dimensionality reduction technique for fast similarity search in large time series databases," *Knowledge and Information Systems*, vol. 1805, no. 1, pp. 122–133, 2000.

[46] C. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the International Conference on Data Mining (SDM '05)*, pp. 506–510, 2005.

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

[48] A. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pp. 1033–1040, 2006.

[49] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proceedings of the AAAI94 Workshop on Knowledge Discovery in Databases*, pp. 359–370, 1994.

[50] S. Aghabozorgi and Y. W. Teh, "Stock market co-movement assessment using a three-phase clustering method," *Expert Systems with Applications*, vol. 41, no. 4, part 1, pp. 1301–1314, 2014.

[51] J. Aach and G. M. Church, "Aligning gene expression time series with time warping algorithms," *Bioinformatics*, vol. 17, no. 6, pp. 495–508, 2001.

[52] B. K. Yi and C. Faloutsos, "Fast time sequence indexing for arbitrary Lp norms," in *Proceedings of the 26th International Conference on Very Large Data Bases*, pp. 385–394, 2000.

[53] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[54] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, Calif, USA, 2011.

[55] A. Mueen, E. Keogh, and N. Young, "Logical-shapelets: an expressive primitive for time series classification," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pp. 1154–1162, 2011.

[56] V. Niennattrakul and C. Ratanamahatana, "Inaccuracies of shape averaging method using dynamic time warping for time series data," in *Computational Science—ICCS 2007*, pp. 513–520, 2007.

[57] S. Aghabozorgi, M. R. Saybani, and T. Y. Wah, "Incremental clustering of time-series by fuzzy clustering," *Journal of Information Science and Engineering*, vol. 28, no. 4, pp. 671–688, 2012.

[58] S.-W. Kim, S. Park, and W. W. Chu, "An index-based approach for similarity search supporting time warping in large sequence databases," in *Proceedings of the 17th International Conference on Data Engineering*, pp. 607–614, 2001.

[59] B.-K. Yi, H. V. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," in *Proceedings of the 14th International Conference on Data Engineering*, pp. 201–208, 1998.

[60] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[61] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2012.

[62] R. R. Sokal, "A statistical method for evaluating systematic relationships," *University of Kansas Scientific Bulletin*, vol. 38, no. 1958, pp. 1409–1438, 1958.

[63] I. Gronau and S. Moran, "Optimal implementations of UPGMA and other common clustering algorithms," *Information Processing Letters*, vol. 104, no. 6, pp. 205–210, 2007.

[64] E. Keogh, Q. Zhu, B. Hu et al., "The UCR time series data mining archive," UCR Time Series Classification, 2011, http://www.cs.ucr.edu/~eamonn/time_series_data/.

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

[66] H. Zhang, T. B. Ho, Y. Zhang, and M.-S. Lin, "Unsupervised feature extraction for time series clustering using orthogonal wavelet transform," *Informatica*, vol. 30, no. 3, pp. 305–319, 2006.

[67] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[68] C. Ratanamahatana, E. Keogh, A. J. Bagnall, and S. Lonardi, "A novel bit level time series representation with implications for similarity search and clustering," in *Proceedings of the 9th Pacific-Asian International Conference on Knowledge Discovery and Data Mining (PAKDD '05)*, pp. 771–777, 2005.

[69] J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for K-means clustering," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 877–886, 2009.

[70] M. Chiş, S. Banerjee, and A. E. Hassanien, "Clustering time series data: an evolutionary approach," *Foundations of Computational, IntelligenceVolume 6*, vol. 206, pp. 193–207, 2009.

[71] F. Rohlf, "Methods of comparing classifications," *Annual Review of Ecology and Systematics*, vol. 5, pp. 101–113, 1974.

[72] M. Song and L. Zhang, "Comparison of cluster representations from partial second-to full fourth-order cross moments for data stream clustering," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*, pp. 560–569, 2008.

[73] F. Gullo, G. Ponti, A. Tagarelli, G. Tradigo, and P. Veltri, "A time series approach for clustering mass spectrometry data," *Journal of Computational Science*, vol. 3, no. 5, pp. 344–355, 2012.

[74] C. J. van Rijsbergen, "A non-classical logic for information retrieval," *The Computer Journal*, vol. 29, no. 6, pp. 481–485, 1986.

[75] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, 2001.

[76] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Locally adaptive dimensionality reduction for indexing large time series databases," *ACM SIGMOD Record*, vol. 30, no. 2, pp. 151–162, 2001.

[77] C. J. van Rijsbergen, *Information Retrieval*, Butterworths, London, UK, 1979.