

Effective Clustering of Time-Series Data Using FCM

Saeed Aghabozorgi and Teh Ying Wah

Abstract—Today, wide important advances in clustering time series have been obtained in the field of data mining. A large part of these successes are due to the novel achieves in dimensionality reduction and distance measurements of time series data. However, addressing the problem of time series clustering through conventional approach has not solved the issue completely, especially when the class label of time series are vague. In this paper, a two-level fuzzy clustering strategy is employed in order to achieve the objective. In the first level, upon dimensionality reduction by a symbolic representation, time series data are clustered in a high-level phase using the longest common subsequence as similarity measurement. Then, by utilizing an efficient method, prototypes are made based on constructed clusters and passed to the next level to be reused as initial centroids. Afterwards, a fuzzy clustering approach is utilized to justify the clusters precisely. We will present the benefits of the proposed system by implementing a real application: Credit card Transactions Clustering.

Index Terms—Clustering, time series, fuzzy C-mean, longest common subsequence.

I. INTRODUCTION

In the context of data mining, clustering techniques are typically used in order to group data based upon their similarities. Most of developed clustering algorithms are compatible with static data which their values do not change or change insignificantly over time. On the other hand, a time series is classified as dynamic data because its feature values changes as a function of time. The data in various systems - like finances, medicine, biology, economics, the web and business - is stored as time series data. As a result, this kind of dynamic data is of interest for clustering in various domains.

There are several researches involving time series clustering which generally fall into three different main categories, namely shape-based, feature-based and model-based [1]. Shape-based algorithms typically employ conventional clustering methods for static data which their dissimilarity/similarity measure has been modified with an appropriate one for time series. In the feature-based approach, the raw time series are converted into a feature vector of lower dimension. Afterwards, a conventional clustering algorithm is applied to the extracted feature vectors. In Model-based methods, a raw time series is transformed into model parameters and then a suitable model distance and a conventional clustering algorithm is chosen and applied to the extracted model parameters. In this research we focus on

shape based clustering. Considering literature in shape based clustering area, in the most of works, researchers consider a crucial challenge in time series clustering: Costly execution time of similarity calculation of time series data due to its high dimensional characteristic. As a result, a vast number of researches have published in order to present a way to represent time series in a lower dimension space compatible with conventional clustering algorithms. Moreover, different efforts have been taken on presenting a distance measurement depended on the represented time series.

Researches have shown that typically clustering by well-known conventional algorithms such as K-Means, SOM (Self Organization Map), FCM (Fuzzy C-Mean) and Hierarchical clustering generate clusters with acceptable structural quality, consistency and partially efficient in terms of execution time and accuracy [2]. However, we could not achieve fast and accurate results by applying these algorithms to time series data. We consider the problem involving low accuracy, costly execution time and the stationary manner of the final models in the time series clustering approaches. To shed light on the drawbacks of using conventional clustering algorithms in this domain, it is worth mentioning two typical strategies:

The first strategy in the time series clustering is using raw data and standard Euclidean distance. Unexpectedly, a recent research [3] has shown that the results gained when Euclidean distance is used as similarity measure is surprisingly competitive in terms of accuracy (regardless of low stability in face of noisy data). However, the dimension (or length of raw time series) is a big milestone for conventional clustering approaches even with considering the increased power of processing in recent years. That is, adopting conventional clustering algorithms is infeasible due to its high complexity of distance calculation of raw time series datasets. Specifically, in algorithms that are heuristic and iterative based such as K-Means or FCM, it is more critical. For example, k-Means is very likely slow to converge in high dimensional data [4], [5]. Therefore, this strategy has the drawback of being “computationally costly” and has the advantage of being “accurately reliable” to some extent.

In second strategy, clustering is performed as batch on the dimensionality reduced time series. There are many researches and advances in this area which is explained briefly in next section, though most of them are computationally costly or not accurate enough to be used in distance measurement. For example, one of the most popular approaches employed to dimensionality reduction and representation of time series is Symbolic Aggregate AppRoXimation (SAX) developed by Keogh *et al.* [6]. However, in this approach the distance among time series or between a time series and prototype is calculated by minimum distance which is not accurate. Minimum distance

Manuscript received October 9, 2013; revised December 10, 2013.

Saeed Aghabozorgi is with the Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia (e-mail: saeed@um.edu.my).

Teh Ying Wah is with the Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia (e-mail: tehyw@um.edu.my).

is a distance measurement which approximate the distance between a couples of time series based on SAX representation and lower bounds the Euclidian distance [6] and essentially is defined for using in indexing of time series dataset. Moreover, using these distance measurements, defining prototypes in most conventional clustering algorithms (e.g. k-Means, k-medoid, FCM, etc) is impossible or if it is feasible, it is not accurate enough.

As a result, in despite of advances in representation and distance measure techniques, selecting adequate representation and distance measurement is a challenge which needs a trade-off that should be fined. However, one of the aspects of time series clustering is selecting efficient clustering algorithm which is as important as selecting appropriate time series representation and distance measure. The performance and the accuracy of most utilized approaches in the literature are not efficient enough because they are designed based on static data and then are customized for time series data. In these systems, using conventional clustering approaches is not practical for time series data and the absence of a clustering algorithm that is adequate for time series is felt strongly.

II. RELATED WORKS

Although different techniques have been applied to analyze time series datasets, clustering is one of the most frequently used techniques [7], due to its exploratory nature, and its application as a pre-processing phase in more complex data mining algorithms. Prior studies that have noted different approaches and importance of clustering of time series are listed as such: [8]–[16]. Additionally, there are many reviews and surveys that focus on comparative aspects of time series clustering experiments such as [1], [3], [17], [18]. In reviewing the literature, one can conclude that most works related to clustering time series are classified into “whole time series clustering” and “subsequence clustering” categories as mentioned in [19]. “Whole time series clustering” is considered as clustering of a set of individual time series with respect to their similarity. “Subsequence clustering” means clustering on sub sequences of a time series that are extracted via a sliding window. In this paper we deal with “Whole time series data”. Considering works which propose new algorithms for clustering whole time series data, a two-factor fuzzy time series is presented in [20] to predict the temperature. Wang *et al.* combine High-order fuzzy logical relationships and genetic-simulated annealing techniques in their work. First, an automatic clustering algorithm is used to cluster the historical data into intervals of different lengths. Then, a method is developed to deal with the temperature prediction based on two-factor high-order fuzzy time series. In another work, Cheng-Ping Lai *et al.*, in [21], adopt a two-level clustering method, where both the whole time series, and the subsequence of time series are taken into account in the first and second level respectively. They use SAX transformation and CAST clustering in the first level in order to group level-1 data. For calculating distances between level-2 data, Dynamic Warp Timing (DWT) is used for varying length data, and Euclidean distance for equal length data. At last, Level-2 data, of all the time series, are then grouped by a clustering algorithm.

However, Using CAST algorithm is not computationally as perfect as other clustering algorithms like partitioned clustering algorithms. Additionally, this method is not efficient in front of noisy data because of using DWT as distance measurement in second level.

Considering multi resolution time series, Vlachos *et al.* [22] developed a method to cluster time series based on the k-Means algorithm. They extended the k-Means algorithm to perform clustering of time series incrementally at different resolutions based on DWT decomposition. At first, they used Haar wavelet transformations to decompose all the time series. After that, they apply the k-Means clustering algorithm on various regulations from a dirty to a finer level.

Although the above approaches have been fairly effective in clustering time series data, there are some well-recognized shortcomings in the presented methods:

One of the most frequently used clustering algorithms in time series clustering, is the k-Means. This approach is rather fast in comparison to other algorithms [23] and is usually used in time series clustering. However, k-Means groups objects through an iterative partitioning manner and in case of high dimensional data, it is not very efficient. That is, k-Means is unable to deal efficiently with long time series due to poor scalability. Besides, defining prototype for time series data is not as easy as static objects when k-Means is the chosen algorithm for clustering.

In many cases, the authors use hierarchical clustering because of its simplicity and nested results. However, hierarchical clustering algorithm is not efficient in large datasets of high dimensional data and is proper for small datasets due to its quadratic computational complexity [14].

Given the mentioned limitations in the reviewed literature, we present a method that is simple, flexible and accurate. In this research, a two-level clustering method is presented, where time series are grouped in different resolutions. At first, we use SAX transformation in the first level to convert the time series into a symbolic representation. Then, we calculate the similarity between the two symbolized time series using LCSS approach. In order to group data, Hierarchical clustering approaches is used due to its scalability performance in lower dimension data. Then, for each group, a prototype is made and reused as an initial center in the next level. In order to cluster time series in low-level step, LCSS distance measurement is chosen to calculate similarity between data and prototypes. At last fuzzy-clustering is used for clustering time series data.

The rest of this paper is organized as follows: After a brief review of basic concepts, the proposed model is described in Section III. Then, the proposed algorithm is applied on real time series datasets and the experimental results are reported in Sections IV. In Section V, the efficiency and effectiveness of algorithm will be discussed. Finally, conclusions and future perspectives are drawn in Section B. Here, some basic notations and preliminary definitions are stated.

Definition 1. Time series: A time series $F_i = \{f_1, \dots, f_t, \dots, f_T\}$ is an ordered set of flow vectors which indicate the spatiotemporal characteristics of moving objects at any time t of the total track life T [24]. A flow vector or feature vector $f_t = [X, Y, Z, \dots]$ generally represents location and dynamics in the domain. However, we limit ourselves to just a spatial location $f_t = [X]$ in this work for the sake of

simplicity.

A. Brief Review of SAX

Footnotes should be Times New Roman 9-point, and justified to the full width of the column. High dimensionality and noise are characteristics of most time series data. Dealing with this kind of data is a big challenge for clustering algorithms in comparison with static objects. Dimension reduction methods are usually used in whole clustering in order to alleviate these issues and promote the performance. There are different approaches for dimensionality reduction of objects in order to cluster them effectively such as perceptually important point [9], piecewise aggregate approximation (PAA), normalized spectrum [25], time domain and frequency domain representation [26], wavelet transforms [27], etc. In this paper we use Symbolic Aggregate Approximation (SAX) transformation in order to reduce the dimension of time series data. SAX is a symbolic representation of time series developed by Keogh *et al.* [6]. This method, transfers a time series into the piecewise aggregate approximation (PAA) representation and then it maps the coefficients to symbols.

B. Brief Review of LCSS

One of the problems in time series clustering is measuring the similarity among time series. There are various distance measures designed for specifying similarity between time series. The dynamic time warping (DTW) [28], [29], Euclidean distance, and longest common subsequence (LCSS) are of the most popular distance measurement methods used for time series data. Zhang *et al.* [30] has performed a complete survey on different distance measures and compare them in different applications. Euclidian distance is simple, fast and used as benchmark in many works, because it is parameter free. However, it is sensitive to noise and requires that the time series being compared are of exactly the same dimensionality. DTW and LCSS, on the other hand, are other famous algorithms for measuring similarity between two sequences with irregular-lengths. The LCSS is more robust to noise and outliers in comparison to DTW because all points of two time series do not need to be matched. That is, instead of a one-to-one mapping between points, a point with no good match can be ignored to prevent unfair biasing. As a result, the LCSS is employed as distance measure for our methodology in both levels. Nevertheless, min distance can also be used as distance measurement in the high-level clustering in the case that time series have equal length.

Definition 2. Given F_i as a time series and f_t as feature vector at time t in time series F_i , if f_{qt} is the feature q -th of time series for $q = \{1, \dots, p\}$ at time t and if p is number of features describing each object, then the LCSS distance is defined as [31]:

$$LCSS(F_i, F_j) = \begin{cases} 0, & T_i = 0 \mid T_j = 0 \\ 1 + LCSS(F_i^{T_i-1}, F_j^{T_j-1}), & d_E(f_i, T_i, f_j, T_j) < \varepsilon \text{ and } |T_i - T_j| < \delta \\ \max(LCSS(F_i^{T_i-1}, F_j^{T_j}), LCSS(F_i^{T_i}, F_j^{T_j-1})), & \text{otherwise} \end{cases}$$

where the $LCSS(F_i, F_j)$ value states the number of matching points between two time series and $F_i = \{f_1, \dots, f_t\}$ specifies

all the flow vectors in time series F_i up to time t . In this equation, δ is an integer value which constricts the length of the warping and $0 < \varepsilon < 1$ is a real number as the spatial matching threshold to cover elements with real values. In this paper, a modified distance measure is defined low-level step as:

$$D_{LCSS}(F_i, F_j) = 1 - \frac{LCSS(F_i, F_j)}{\text{mean}(T_i, T_j)}$$

where, using $\text{mean}(T_i, T_j)$ instead of $\min(T_i, T_j)$ results in taking the length of both time series into account. It means, in the case that length of time series is different, if one of them is shorter than other one, the mean help us to consider both time series.

C. Brief Review of Fuzzy C-Means (FCM) Algorithm

There are a wide variety of clustering algorithms used for time series clustering such as Hierarchical based [6], [17], [32], partitional based [33], [34] and density based [35]. We have focused on the fuzzy c-means (FCM) [36] largely due to the advantage of the degree of membership of a time series to the clusters in clustering process. It is used to facilitate the detection of changes in prototypes. Additionally, fuzzy sets have a more realistic approach to address the concept of similarity than classical sets. A fuzzy set is a set with fuzzy boundaries where each element is given a degree of membership to each set. The FCM works by partitioning a collection of n vectors into c fuzzy groups and finds a cluster center in each group such that the cost function of dissimilarity measure is minimized. Bezdek introduced the idea of a ‘‘fuzzification parameter’’ (m) in the range $[1, n]$ which determines the degree of fuzziness (weighted coefficient) in the clusters. Normally m is in the range $[1.25, 2]$ inclusively and controls the permeability of the cluster horizon which can be viewed as an n -dimensional cloud moving out from a cluster center [37]. The Fuzzy C-means (FCM) has been applied to time series to find the clusters in some works in literature. For example, in [38], authors use FCM to cluster time series for speaker verification. Authors in [39] use fuzzy variant to cluster similar object motions that were observed in a video collection. We use the concept of FCM with time-series data using LCSS as distance measure and a new prototype updating approach. Let the centers be $v_j = \{v_1, \dots, v_c\}$ and each time series F_i that $i = \{1, \dots, n\}$ and $D_{LCSS}(F_i, v_j)$ as distances between centers and time series. Therefore, the membership values μ_{ij} are obtained with:

$$\mu_j(x_i) = \frac{\left(\frac{1}{D_{LCSS}(F_i, v_j)}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^p \left(\frac{1}{D_{LCSS}(F_i, v_k)}\right)^{\frac{1}{m-1}}}$$

where the sum of cluster memberships for a time series equals 1. Then, the FCM objective function (standard loss) that is attempted to be minimized takes the form:

$$J = \sum_{j=1}^c J_j = \sum_{j=1}^c \sum_{i=1}^n [\mu_{ij}]^m D_{LCSS}(F_i, v_j)$$

where μ_{ij} is a numerical value between [0; 1]; $D_{LCSS}(F_i, v_j)$ is the LCSS distance between the j_{th} prototype and the i_{th} time series; and m is the exponential weight which influences the degree of fuzziness of the membership matrix. In different iterations, the membership values of the time series are calculated, and then the prototypes (cluster centers) are recomputed. The iterations will continued to a terminate criterion. Finally, with the application of the fuzzy clustering algorithm to time series, a set of clusters is created so that each cluster, C_i , includes a subset of time series with similar patterns (common subsequence).

III. PROPOSED MODEL

In this section, a general view of the methodology is explained based on the fuzzy clustering discussed in Section C. We propose a two-level clustering where at first, clusters are made in high-level phase and then clusters are fined in low-level phase. The following figure (Fig. 1) illustrates the pseudo code the overall view of methodology.

<p>Input c: number of clusters , T: time series data , w: word length, s: segment size Initialization n : number of time series , DC: distance matrix [n × c] , F: fuzziness ,FC: membership function matrix STEP 1: High-level Clustering 1.1. T_{sax}=Use SAX to reduce the dimension of time series T 1.2. HC=Perform the hierarchal pre-clustering (c,T_{sax}) // HC is crisp membership Matrix 1.3. C=Construct the initial prototypes (HC) // C is matrix of cluster prototypes STEP 2: Low-Level Clustering 2.1. DC=Calculate Distance Matrix (T, C) 2.2. FC=Calculate fuzzy partition matrix (DC, C) Repeat 2.3 C=Update the cluster prototypes (FC,C) 2.4 DC=Update the distances Matrix (DC,C) 2.5 FC=Update the fuzzy partition matrix(DC, C) until convergence</p>

Fig 1. Pseudo code for two-phase time series fuzzy clustering.

Step 1.1: Reduce dimension: There is an issue with the clustering of time series in comparison with standard object clustering that need to be addressed. The size of the data in the time series is highly problematic. A dimensional reduction technique should be employed to reduce the dimension through mapping the time series data to a new feature space of a lower dimension. As mentioned, SAX presentation is utilized as transformation in this research because it is as efficient as well-known representations, while requiring less storage space. However, other well-known methods such as APP, DWT, or DFT also can be utilized for this purpose. For using SAX, the word and segment size of time series needs to be declared as input to avoid unreasonable level symbolization.

Step 1.2: Pre-clustering: In this step, Hierarchal clustering is employed in order to build initial prototypes in high-level clustering, Hierarchical clustering is desired because it does not need prototype objects initially, though it can be done by any other techniques which determine such clusters. In this approach, the distance between two clusters is measured by averaging distances between all pairs of time series, where each pair is made up of one time series from each group. The

LCSS is utilized as distance measure in this step. The output is approximate groups of data which are not very accurate, but are constructed very fast.

Step 1.3: In this step, each approximated time series is replaced with its raw time series. Then, the prototype of the cluster is constructed based on a new fuzzy-based prototype defined in [40]. In this approach, shortest common super sequence (SCSS) is used to make the prototype. In the SCSS problem, the two sequences F_i and F_j are given and the objective is to find the shortest possible common super sequence of these sequences. In general, the SCSS is not unique, but it is possible to find the optimal [40]. The SCSS problem is closely related to the LCSS problem and for two input sequences, the SCSS can be easily formed from their LCSS.

Step 2.1: Calculate Distance Matrix: the distance between each time series and the prototypes provided in high-level step are computed in this step. LCSS is utilized as distance measurement in this step for handling noises in raw data and time series with various lengths.

Step 2.2: Calculate fuzzy partition matrix: the algorithm computes its coefficients of the clusters, using the formula mentioned in section C. Then the algorithm repeats steps 2.3, 2.4 and 2.5 until a termination condition (that is, the coefficients' change between two iterations is no more than ϵ , a sensitivity threshold which can be defined by user). Although the execution cost of each iteration is partially high (because of using raw data), in overall, the cost of the algorithm is better than clustering time series from scratch. It is due to its less iteration to converge than standard fuzzy algorithms because of providing partially precise and fit initial prototypes.

Step 2.3: Update the cluster prototypes: In order to update the prototypes by conventional fuzzy clustering, the prototypes move according to the membership of all objects. However, if all the time series are considered to update the prototypes, after a few iterations, prototypes turn into a rather straight line due to the weakness of creating prototypes from high dimensional objects. Authors in [40] present an approach for moving prototypes based on time series data clusters. This algorithm is used in incremental clustering as well [41] to update prototypes. Their results show that it is fast and accurate enough when fuzzy techniques are utilized as the algorithm for clustering.

In this approach, the fuzziness of candidate time series is utilized in order to update prototypes in this approach. The candidate time series are defined using a thresholds, μ_{min} , to update only the time series with memberships more than μ_{min} . This threshold is required in order to ignore the noise of time series with a lower fuzziness value. Additionally, it prevents prototypes from stretching incrementally over time. μ_{min} equals to the inverse of the class number value:

$$\mu_{min} = \frac{1}{c}$$

In fact, in accordance with the definition, only a part of whole time series, so called candidate time series, are considered in the calculation of prototypes. Then, a parts of candidate time series which, although they do not have match points with the prototype, they have higher memberships than

$\mu_{\text{mean},j}$ ($\mu_{ij} > \mu_{\text{mean},j}$) are imported in prototype calculation. In Fig. 2, the algorithm used is shown to explain the details of updating prototypes. In this algorithm, $F_i = \{f_{i1}, \dots, f_{it}, \dots, f_{ip}\}$ is a time series with length p and time points f_{it} . The matrix $\text{LCSS}(F_i, V_j)_{r \times 2}$ indicates match points (LCSS) of time series F_i and prototype V_j where $\text{LCSS}(F_i, V_j)$ has a dimension of $r \times 2$ (r is the number of match points).

Step 2.4. Update distance matrix: this step is for updating distance of all time series data from moved prototype in previous step.

Step 2.5. Update the fuzzy partition matrix. This step is based on the FCM algorithm for updating the fuzzy partition matrix.

```

Method Name: Prototype_Update
Input:
c: cluster members
Initialization:
initialize  $\mu_{\min} = \frac{1}{c}$ 
initialize  $\mu_{\text{mean},j}$ 
initialize a set U, length=length( $V_j$ )
Algorithm:
for each time series  $F_i$  with  $\mu_{ij} > \mu_{\min}$ ;
  for each pair of match point in  $\text{LCSS}(F_i, V_j)$ 
     $P_{\text{LCSS}(F_i, V_j)_{r,1}, i} = \text{LCSS}(F_i, V_j)_{r,2}$ ;
     $U = \{U, P_{\text{LCSS}(F_i, V_j)_{r,1}, i}\}$ 
  end for
end for
for every time series  $F_i$  whose  $\mu_{ij} > \mu_{\text{mean}}$ ;
   $U = \text{prototype}(F_i, U, P, \mu_{ij})$ 
end
for each  $P_h$  in U
   $V'_{jt} = \frac{\sum_{i=1}^b (\mu_{ij})^m (f_{ix})}{\sum_{i=1}^b (\mu_{ij})^m}$ ,  $b = \text{length}(P_h)$ 
end for
    
```

Fig. 2. Pseudo code to update prototype based on time series inside a cluster.

IV. EXPERIMENTAL RESULTS

For a bank in Malaysia, similar yearly patterns of its customers, when they purchase by their credit cards, are very important. Discovering similar customers leads to make decisions to change the card types and invest in different businesses based upon that. Additionally, bank uses these patterns for fraud detection, campaigns, etc. In order to find accurate similar transactions on all credit cards, clustering is performed based on their total bills on each day. The proposed methodology is performed on different cardinalities of dataset of the credit card dataset (time series of total bills). In this section, at first, the datasets used in the experimental procedure are described and then, the methodology is applied to the real dataset to present the results. At last, achieved results in different cycles are evaluated and discussed.

The dataset is a collection of the time series which is related to one year of outstanding amount of 10,000 customer's credit card. Each time series in this dataset is presented by 200 to 365 observations. That is, time series length of customers varies from 200 to 365 time points.

At first, the window size $w=30$ is used to symbolize the time series. It reduces dimension of time series and as a result gains high speed during pre-clustering step. Then, agglomerative hierarchical clustering is used to the low-resolution time series in order to make the initial

prototypes. In this step such as many other clustering algorithms, the problem of defining the number of clusters is an obstacle. However, because a fast clustering is performed on low-resolution time series, a comprehensible view of data is constructed as a dendrogram which helps experts in banking system to choose the best cluster number. Considering approximate knowledge about the clusters, 10 clusters are chosen as main clusters. The prototypes of constructed clusters are constructed as a centroid of low-level step. In Fig. 3, some of clusters are illustrated with their prototypes.

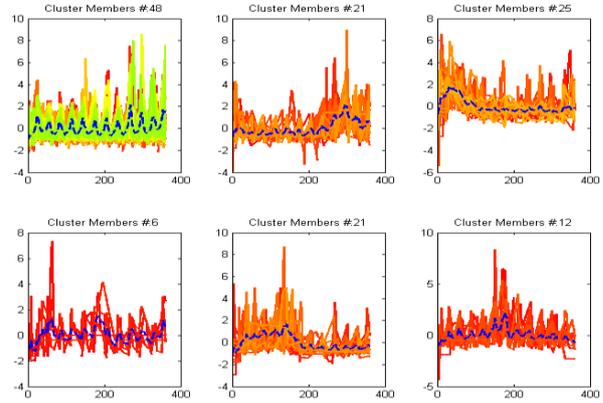


Fig. 3. The clusters of time series with their prototypes.

V. EVALUATION

A. Execution Time

In order to show reduced execution time of our algorithm, AMCT (Accurate Multilevel Clustering of Time-series), in comparison with FCM as a conventional method, both algorithms are applied to the credit card dataset. The results show that the AMCT converges with less iteration than FCM (Fig. 4). In our evaluation, the execution time is measured for the whole clustering function of FCM and, for AMCT, the timing is calculated for all steps. The steps in AMCT consist of data reduction by SAX, pre-clustering by hierarchical clustering, and the needed iterations in low-level clustering. A DELL Server with 4GB RAM and two 2.6 GHz Quad-core Processors was used to run all the tests. The chart presented in Fig. 4 shows timing result values (the mean accumulated CPU time in seconds) across the bank dataset with different cardinalities.

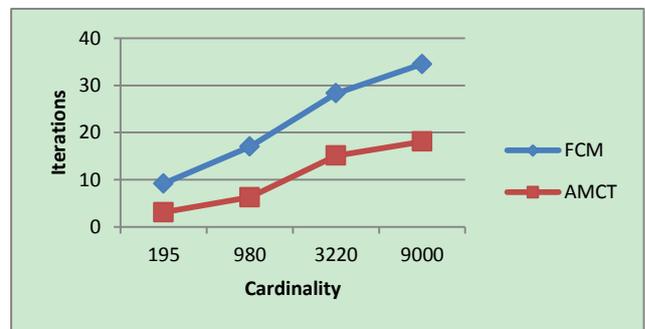


Fig. 4. Average required iterations until convergence in AMCT and FCM. The results are average of 10 running of both algorithms on different cardinalities of dataset.

Despite the required time for dimension reduction and pre-clustering (very efficient on dimensionally reduced data),

the result of this experiment shows that the speed of AMCT is higher than conventional FCM, because AMCT requires very few iterations until it converge. The experiment is repeated for other algorithms to check the efficiency of proposed algorithm. The reported execution time for AMCT shows a speedup versus k-Means and Hierarchical algorithms as well, especially in big cardinality datasets (See Fig. 5).

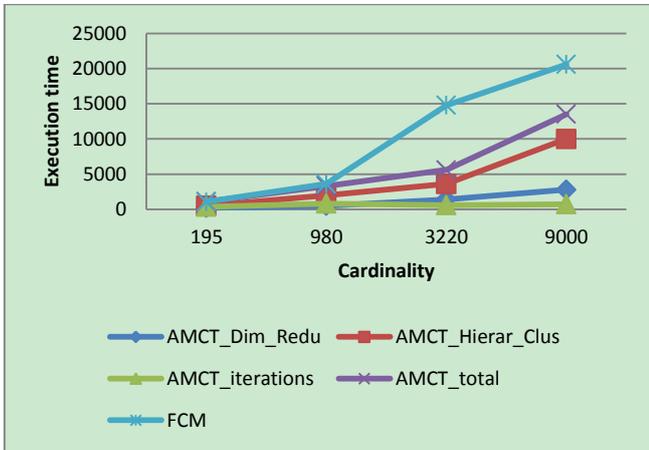


Fig. 5. Execution time of AMCT and FCM across different cardinalities of dataset.

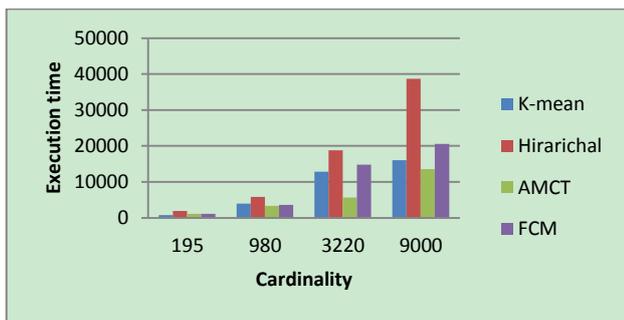


Fig. 6. Execution time of algorithms cross different cardinalities and conventional algorithms.

TABLE I: COMPARISON OF CLUSTERS GAINED BY APPLYING k-MEANS AND AMCT ON 195 TIME SERIES

cluster #	k-Means	AMCT	Similar cases	Similarity ratio
1	15	15	15	100
2	18	19	18	94.73
3	8	8	8	100
4	35	35	33	94.28
5	14	14	14	100
6	11	12	11	91.66
7	11	9	8	88.88
8	18	18	18	100
9	41	40	40	100
10	24	25	24	96
	195	195	189	189/195=96.92

B. Accuracy

Clustering is an unsupervised learning technique and there are no predefined classes to compare the clustering results [42]. As a result, it is necessary to use some validity criteria. In order to prove that the proposed approach is more effective than conventional algorithms, the k-Means and the Hierarchy algorithms are employed for comparison purpose. At first,

the result of clustering is compared with standard k-Means which has been used in many researches for benchmarking. Although k-Means is not the ideal, but close results to k-Means imply that AMCT is accurate enough. We performed the algorithm 10 times on a part of dataset with 195 time series to show the accuracy of the algorithm. The reported result (See TABLE) reveals that clusters are around 95% similar to standard k-Means algorithms. Therefore, it is fair to say that the proposed algorithm is comparable with other algorithms (See Fig. 6).

VI. CONCLUSION

In order to show experimental results, AMCT methodology was implemented on credit cards data of a bank to perform the segmentation. Moreover, we applied three more frequently used clustering algorithms on our dataset to compare them with the developed approach (AMCT) in terms of accuracy and execution time. The results of this study indicate that this method is more efficient than conventional clustering algorithms computationally. It is because of less iteration in the learning phase. Moreover, in terms of being accurate, this method is sufficiently accurate enough in comparison with traditional strategies due to its accurate prototype. However, further research needs to be done in order to evaluate AMCT in terms of speed and accuracy of data clusters in different datasets with different dimensions to understand its potentials and limitations.

ACKNOWLEDGMENT

This paper is supported by RP0061-13ICT University of Malaya Research Grant.

REFERENCES

- [1] T. Warrenliao, "Clustering of time series data--a survey," *Pattern Recognit.*, vol. 38, no. 11, pp. 1857–1874, Nov. 2005.
- [2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
- [3] 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," in *Proc. VLDB Endow.*, vol. 1, no. 2, pp. 1542–1552, 2008.
- [4] A. Vattani, "k-means requires exponentially many iterations even in the plane," in *Proc. the 25th Annual Symposium on Computational Geometry*, 2009, pp. 324–332.
- [5] D. Arthur, "How slow is the k-means method?" in *Proc. Twenty-Second Annu.*, 2006.
- [6] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Work. Res. Issues Data Min. Knowl. Discov. - DMKD '03*, pp. 2, 2003.
- [7] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf.*, vol. 17, no. 2, pp. 107–145, 2001.
- [8] P. Cotofrei and K. Stoffel, "Classification rules+ time= temporal rules," *Comput. Sci. 2002*, pp. 572–581, 2002.
- [9] T. C. Fu, F. L. Chung, V. Ng, and R. Luk, "Pattern discovery from stock time series using self-organizing maps," in *Proc. Workshop Notes of KDD2001 Workshop on Temporal Data Mining*, 2001, pp. 26–29.
- [10] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani, "Mining the stock market: which measure is best?," in *Proc. the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, pp. 487–496.
- [11] X. Jin, L. Wang, Y. Lu, and C. Shi, "Indexing and mining of the local patterns in sequence database," *Intell. Data Eng. Autom. Learn.* 2002, pp. 39–52, 2002.
- [12] E. Keogh and C. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, May 2004.

- [13] P. Tino, C. Schittenkopf, and G. Dorffner, "Temporal pattern recognition in noisy non-stationary time series based on quantization into symbolic streams, lessons learned from financial volatility trading," *Rep. Ser. Adapt. Inf. Syst. Manag. Econ. Manag. Sci.*, 2000.
- [14] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration," *Data Min. Knowl. Discov.*, vol. 7, no. 4, pp. 349–371, 2003.
- [15] S. Aghabozorgi and Y. W. Teh, "Stock market co-movement assessment using a three-phase clustering method," *Expert Syst. Appl.*, vol. 41, no. 4, part 1, pp. 1301–1314, 2014.
- [16] S. Aghabozorgi and T. Y. Wah, "Shape-based Clustering of Time Series Data," *J. Intell. Data Anal.*, vol. 18, no. 5, 2014.
- [17] S. Hirano and S. Tsumoto, "Empirical comparison of clustering methods for long time-series databases," *Act. Min.*, vol. 3430, pp. 268–286, 2005.
- [18] T. C. Fu, "A review on time series data mining," *Eng. Appl. Artif. Intell.*, vol. 24, no. 1, pp. 164–181, Feb. 2010.
- [19] E. Keogh and J. Lin, "Clustering of time-series subsequences is meaningless: implications for previous and future research," *Knowl. Inf. Syst.*, vol. 8, no. 2, pp. 154–177, 2005.
- [20] N.-Y. Y. Wang and S.-M. M. Chen, "Temperature prediction and TAIFEX forecasting based on automatic clustering techniques and two-factors high-order fuzzy time series," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 2143–2154, Mar. 2009.
- [21] C.-P. P. Lai, P.-C. 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, Sep. 2010.
- [22] M. Vlachos, J. Lin, and E. Keogh, "A wavelet-based anytime algorithm for k-means clustering of time series," in *Proc. Work. Clust.*, pp. 23–30, 2003.
- [23] P. S. Bradley and U. Fayyad, "Refining initial points for k-means clustering," in *Proc. the Fifteenth International Conference on Machine Learning*, 1998, vol. 66.
- [24] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in *Proc. 2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 312–319, Jun. 2009.
- [25] C. T. Shaw and G. P. King, "Using cluster analysis to classify time series," *Phys. D Nonlinear Phenom.*, vol. 58, no. 1–4, pp. 288–298, 1992.
- [26] L. M. D. Owsley, L. E. Atlas, and G. D. Bernard, "Self-organizing feature maps and hidden Markov models for machine-tool monitoring," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2787–2798, 1997.
- [27] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," *Found. Data Organ. Algorithms*, vol. 46, pp. 69–84, 1993.
- [28] D. Sankoff and J. B. Kruskal, "Time warps, string edits, and macromolecules: the theory and practice of sequence comparison," in *Read. Addison-Wesley Publ.* 1983, D. Sankoff, J. B. Krus Eds., vol. 1, 1983.
- [29] S. Chu, E. Keogh, D. Hart, M. Pazzani *et al.*, "Iterative deepening dynamic time warping for time series," in *Proc. Second SIAM International Conference on Data Mining*, 2002, pp. 195–212.
- [30] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," in *Proc. 18th Int. Conf. Pattern Recognition, ICPR 2006*, vol. 3, 2006, pp. 1135–1138.
- [31] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proc. 18th International Conference on Data Engineering*, 2002, pp. 673–684.
- [32] 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 *Proc. the national conference on artificial intelligence*, 2000, pp. 846–851.
- [33] J. Beringer and E. Hullermeier, "Online clustering of parallel data streams," *Data Knowl. Eng.*, vol. 58, no. 2, pp. 180–204, Aug. 2006.
- [34] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulos, "Iterative incremental clustering of time series," *Adv. Database Technol.* 2004, pp. 521–522, 2004.
- [35] S. Chandrakala and C. Chandra, "A density based method for multivariate time series clustering in kernel feature space," in *Proc. IEEE International Joint Conference on Neural Networks IEEE World Congress on Computational Intelligence (2008)*, 2008, pp. 1885–1890.
- [36] J. C. Bezdek, *Fuzzy Mathematics in Pattern Classification*, Cornell University, August 1973.
- [37] E. Cox, *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*, 2005.
- [38] D. Tran and M. Wagner, "Fuzzy c-means clustering-based speaker verification," *Adv. Soft Comput.* 2002, no. 2275, pp. 363–369, 2002.
- [39] J. Alon and S. Sclaroff, "Discovering clusters in motion time-series data," in *Proc. Computer Society Conference on Computer Vision and Pattern Recognition*, 2003, pp. 375–381.
- [40] S. Aghabozorgi, T. Y. Wah, A. Amini, and M. R. Saybani, "A new approach to present prototypes in clustering of time series," in *Proc. The 7th International Conference of Data Mining*, 2011, vol. 28, no. 4, pp. 214–220.
- [41] S. Aghabozorgi, M. R. Saybani, and T. Y. Wah, "Incremental Clustering of Time-series by Fuzzy Clustering," *Journal of Information Science & Engineering*, vol. 28, issue 4, pp. 671–688, 2012.
- [42] M. Halkidi and Y. Batistakis, "Clustering algorithms and validity measures," in *Proc. Thirteenth International Conference on Scientific and Statistical Database Management SSDBM*, 2001, pp. 3–22.



Saeed Aghabozorgi received his B.Sc. in computer engineering and software discipline from University of Isfahan, Iran, in 2002. He received his M.Sc. from Islamic Azad University, Iran, in 2005, and his Ph.D. from University of Malaya in 2013. Currently, he is a lecturer at the Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia. His current research area is data mining.



Ying-Wah Teh received his B.Sc. and M.Sc. from Oklahoma City University and Ph.D. from University of Malaya. He is currently an associate professor at Information Science Department, faculty of Computer Science and Information Technology, University of Malaya. His research interests include data mining, text mining, document mining, cloud computing and big data.