International Journal of Innovative Computing, Information and Control
Volume **10**, Number **4**, August **2014**

ICIC International ©2014 ISSN 1349-4198
pp. 1457–1470

ENSEMBLE-BASED TIME SERIES DATA CLUSTERING FOR HIGH DIMENSIONAL DATA

Sampasetty Saravanan[1] and Gulam Mohideen Kadhar Nawaz[2]

[1]Department of Master of Computer Applications
Adhiyamaan College of Engineering Hosur
Krishnagiri District, Tamil Nadu 635109, India
saravanan0675@gmail.com

[2]Department of Computer Applications
Sona College of Technology
Thiagarajar Polytechnic College Road, Salem, Tamil Nadu 636005, India

Received April 2013; revised December 2013

Abstract. *The time series clustering analysis provides an effective way to discover the intrinsic structure. In most of the time, the series of data mining algorithms uses similarity search as the core subroutine, and hence the time taken for similarity search becomes complicated, due to the large data sets. In this paper, we have developed an approach for clustering the temporal data via the ensemble of cluster weight for multiple partitions developed by initial clustering analysis on two types of representations. Initially, time series data sets are converted into representations in which each partition is used to reduce the dimension and subsequently, the clustering algorithm is applied. The different types of weight algorithms are applied to each of the representation. By considering the weight and the representation matrix, we develop the final clustering. Finally the experimentations are carried out on the time series data sets, and the simulation results demonstrate that our approach gives the desired results in clustering analysis of time series data.*

Keywords: Representations, Time series data, Representation clustered matrix, Weighted consensus function, Fuzzy-C-means (FCM), Kernel function, Temporal data clustering

1. **Introduction.** Clustering [1] has been premeditated expansively for more than forty years, transversely with a lot of disciplines due to its wide applications. Clustering is the procedure for transmission of data stuff into a set of displaces groups called clusters, where objects in each cluster are more analogous to all the stuff other than dissimilar clusters. The prose presents with a huge number of algorithms for well-organized clustering of data. These algorithms can be categorized into nearest-neighbor clustering algorithm, hairy clustering algorithm, partitional clustering algorithm, and hierarchical clustering algorithm, reproduction neural networks for clustering algorithm, statistical clustering algorithm, and density-based clustering algorithm and so on. In these algorithms, hierarchical and partitional clustering algorithms are the two most important approaches of growing interest in research communities. Hierarchical clustering algorithms can typically display satisfied clustering outcome domino effect. Even though the hierarchical clustering technique is often portrayed as a superior worth clustering approach, this practice does not hold any condition for the reallocation of entities, which have been inadequately confidential to the nearest of the beginning stage. Additionally, most of the hierarchical algorithms are very computationally concentrated and necessitate much reminiscence space [2].

Normally, clustering [27] is essential when no labeled data are easy to obtain, in spite of the data being double, definite, numerical, period, ordinal, relational, textual, spatial, chronological, spatio-temporal, figure, multimedia, or mixtures of the above data types. Data are known as fixed if all their aspect values do not modify with time, or modify negligibly. The mass of clustering analyses have been performed on static data. As motionless data clustering, circumstance chain clustering also requires a clustering algorithm or progression to form clusters, with the specified set of unlabeled data matter and in which the choice of clustering algorithm depends together on the category of data obtainable and on the meticulous reason and its application. As far as the time sequence data are troubled, distinctions can be completed when the data are discrete-valued or real-valued, homogeneously or non-uniformly sampled, univariate or multivariate, and even if the data sequence are of identical or uneven length. Non-uniform sampled data must be rehabilitated into uniformed data before clustering operation can be performed. This can be achieved by an extensive series of methods, from trouble-free downhill variety based on the roughest sampling hiatus to a complicated modeling and judgment approach [3].

The behavior of data reliance, accessible time sequence clustering algorithms can be categorized as temporal-proximity-based, model-based, and representation-based clustering methodologies. Temporal-proximity and model-based clustering algorithms are used as the crow flies functioning on time sequence, where sequential association is dealt straightly during the clustering psychiatry by revenue of sequential correspondence events [4,5], e.g., forceful time warping, or dynamic models [6,7], like Hidden Markov Model (HMM). In dissimilarity, a representation-based algorithm converts time sequence into minor dimensionality of facet space, where any stationary data- clustering algorithm is pertinent to time sequence clustering, which is particularly resourceful in calculation [8]. Time sequence clustering study provides an efficient method to determine the inherent structure and to compress/recapitulate the sequence conveyed in sequential data [28] by exploring the energetic behaviors concealed with original time sequence in an unsubstantiated erudition model. The eventual intention of time sequence clustering analysis is done to panel a set of unlabeled time sequence into groups or clusters where all the sequences grouped in the identical cluster should be consistent or harmonized. There are two basis evils in clustering analysis, i.e., copy assortment and appropriate federation. The earlier method is to estimate the essential number of clusters, which is a temporal dataset, and the concluding demands, where an appropriate consortium rule together with the consistent sequences mutually forms a cluster [8].

Clustering time sequence data [32,33] is a complicated mission, where the applications has a wide-range hodgepodge of fields, and has newly concerned a huge amount of research. The study provides a way to investigate the existing algorithms and techniques for clustering of time series data streams and helps to provide guidelines for prospect improvement. Prospect research can be directed to the following aspects. (1) Cluster time series data can be used in high dimensional data by growing the rapidity. (2) Computation attempt can be improved in high-dimensional records using clipping practice. (3) An efficient loom can be increased to guess the prospect value in time sequence data. (4) Since, time sequence data deals with raw arrangement, which is luxurious in requisites of dispensation and storage [9].

We have proposed an approach of ensemble based time series data clustering, for high dimensional data. Initially, the time series temporal data are modeled by the representations to eliminate basic drawback in the representation based temporal data clustering analysis. Then, the initial clustering is applied to each representation, and the initial clustering algorithm is done by the existing clustering techniques. The outcome of initial clustering gives the multiple partitions and it is used to assign the weights for each partition through the proposed clustering validation standard, and subsequently final weight is also found out with the help of all the weights. Finally, we can achieve the final partition with the help of the final weight, and the representation matrix generated from various representation and multiple partitions from the initial clustering.

The rest of the paper is organized as follows. A brief review of few literature works in the relational data mining is presented in Section 2. The contribution of the paper is given in Section 3 and the proposed methodology for ensemble-based time series data clustering, for high dimensional data is given in Section 4. The experimental results and performance analysis discussion are provided in Section 5. Finally, the conclusions are summed up in Section 6.

2. **Review of Related Works.** Literature presents an assortment of techniques for occasion of sequence data clustering, using dissimilar methods. Here, we present an appraisal for few of the algorithms. K. Premalatha and A. M. Natarajan [10] have proposed an advancement for data clustering based on PSO with confined investigation. They constructed an alteration approach for the element swarm optimization (PSO) algorithm and applied it in the record sets. They also provided a technique for particles to maneuver clearly off from the local stagnation, and the local investigations were useful to get better decency of fitting. The efficiency of this thought was established by group psychoanalysis. They showed that the copy provides improved presentation and maintains more assortments in the group, and thereby allows the particles to be vigorous to trace the altering environment.

S. Das *et al.* [11] have presented a routine clustering, using an enhanced discrepancy evolution algorithm. They described a request of DE to the routine clustering of huge unlabeled data sets. The advantage of this technique was established by comparing it with two newly developed partitional clustering techniques, and one trendy hierarchical clustering algorithm. The partitional clustering algorithms were based on two influential well-known optimization algorithms, specifically the hereditary algorithm, and the element swarm optimization. Also a motivating real-world request for the future method to routine segmentation of descriptions is reported.

Y.-T. Kao *et al.* [12] have presented a hybridized, statistics clustering. They planned a mixture method by combining the K-means algorithm, Nelder-Mead simplex investigate, and constituent part cloud optimization, called K-NM-PSO. The K-NM-PSO searches for cluster centers of random data set as the K-means algorithm, which can efficiently and competently find the worldwide optima. K-NM-PSO algorithm was experienced on nine data sets, and its presentation was compared with those of PSO, NM-PSO, K-PSO and K-means clustering. The conclusion shows that, K-NM-PSO was vigorous for conducting data clustering. Pallavi and S. Godara [17] have presented an enhanced clustering coming up with time series data set. They projected BIRCH hierarchical clustering technique and used it, on great quantities of arithmetical data by incorporation of hierarchical clustering and other clustering methods such as iterative partitioning methods like k-means and k-medoids and their assessments.

G. Zhao and W. Deng [13] have an accessible HMM-based hierarchical clustering of genetic material appearance time sequence data. They planned a Hidden Markov Model-based Hierarchical Clustering (HMM-HC) method to analyze gene expression time series data. Gene appearance time sequence data were preprocessed and HMMs were used, to replicate the preprocessed data to take advantage of the time dependency between different time points in the genetic material profile. The built HMM models were clustered with hierarchical approach to accomplish the clustering of data. Their consequence showed that the technique can be used to create high-quality clusters and also to establish the suitable number of clusters.

Y. Yang and K. Chen [14] have proposed a chronological data clustering via prejudiced clustering band with dissimilar representations. They introduced a chronological data clustering structure via a prejudiced clustering company, of numerous partitions fashioned by preliminary clustering investigation on dissimilar temporal data representations. They planned for a prejudiced agreement purpose, guided by the clustering corroboration criteria to take rights on preliminary partitions of the contender agreement partitions from dissimilar perspectives, and then to initiate a conformity occupation to the additional settlement of those applicants agreement partitions of a concluding divider. As a consequence, the future weighted clustering band algorithm provided an effectual enabling method for the combined use of dissimilar representations, which cuts down the sequence defeat in a solitary depiction and exploits a variety of sequence to the source of fundamental chronological data.

Y. Yang and K. Chen [15] have prepared an accessible time sequence clustering via RPCL System Company with dissimilar representations. They obtained an unverified ensemble erudition approach to time sequence clustering by combining rival-penalized competitive learning (RPCL) networks with different representations of time sequence. In their approach, the RPCL system collection was employed for clustering analysis based on dissimilar representations of time sequence whenever it was accessible, and the most favorable assortment purpose was to find out a concluding agreement partition from manifold partition candidates yielded by applying a variety of agreement functions for the mixture of aggressive learning consequences. Their approach had been evaluated with 16 standard time sequence data mining tasks, along with contrast to state-of-the-art time sequence clustering techniques.

P. Maji and S. Paul [16] have proposed a microarray time-sequence data clustering using rough-hairy C-means algorithm. They obtained a request of uneven – furry c-means (RFCM) algorithm to realize co-expressed genetic material clusters. One of the main issues of RFCM is based on the microarray data clustering, which was used to choose preliminary prototypes of dissimilar clusters. To conquer this restriction, a technique was anticipated to choose preliminary cluster centers. A technique was also introduced based on the Dunn's cluster validity directory to recognize most favorable values of dissimilar parameters of the initialization technique and the RFCM algorithm. The efficiency of the RFCM algorithm, and the length of contrast with other connected methods, were established on five yeast genetic material appearance time-series data sets using Silhouette index, Davies-Bouldin directory, and genetic material ontology based analysis.

3. **Contribution of the Paper.**

The major contributions of the paper are discussed as follows:

- Model the time series datasets into different types of representation to remove the basic demerits of the temporal data clustering analysis.
- Initial clustering analysis on different types of representation makes the multiple partitions.
- New representation matrix with the help of multiple partition and different types of representation.
- Utilize the different types of clustering validation standards of evaluation to weight the initial clustering results.
- Comparison of our proposed method with the existing techniques evaluates the effectiveness and efficiency of the proposed approach using 16 benchmark datasets.

4. **The Proposed Approach of Ensemble-Based Time Series Data Clustering for High Dimensional Data.** In this paper, with the intention of conquering the primary limitation of the representation based temporal data clustering analysis, we present an approach to temporal data clustering with different representations. These representations are fed to the initial clustering process; the initial clustering is completed by the existing algorithms, and the outcome of the initial clustering is multiple partitions. We propose the weighted algorithm clustering ensemble in two ways: one is to assign weights for each partition through the several types of clustering validation standards. Then, representation matrix is generated and final clustering is achieved through the representation matrix and weight.

Our ultimate aim is to cluster the temporal data via ensemble of cluster weight for multiple partitions developed by initial clustering analysis on two types of representations. We proposed a clustering method through normal and PCA representations, and the proposed method consists of four modules.

1. *Extraction of representation*
2. *Initial clustering*
3. *Computation of weight to each partition*
4. *Ensemble of weighted for final clustering*

The algorithmic description of the proposed approach is given in Table 1.

Table 1. Pseudo code

```
Algorithm: Ensemble-based time series clustering
Input: Time series data, $T$
Output: Clustered result
Step 1. Find PCA components of input data, $T$
Step 2. Find wavelet components of input data, $T$
Step 3. Do $k$-means clustering on PCA, Wavelet and data space individually
Step 4. Find weights of every partitions using MHI, FWA, KFWA
Step 5. Construct representation matrix based on the output of Step 3
Step 6. Do $k$-means clustering on representation matrix with new distance formulae
Step 7. Get the final clusters
Step 8. Stop
```

4.1. Extraction of representations.

Formally, the time series data are defined as the set of pairs of data points and corresponding time stamps such as $T = \{(x_i, t_i)\}$ where $(1 \leq i \leq n)$. Typically the time series datasets are highly dimensional and clustering or grouping directly to its original format becomes extremely affected due to the cost in terms of computing and storage. In order to remove such types of difficulties here, we utilize the representation methods that can reduce the dimensionality of the time series data. In this paper, we utilize the PCA representation to reduce the dimension of the raw data and we also use the normal representation. The normal representation denotes the original format of the datasets.

4.1.1. *PCA representation.* The PCA representation helps to convert the raw temporal data format into feature vectors of fixed dimensionality for initial clustering.

Principal Component (PC)

Theoretically, a principal component can be described as a linear association of optimally weighted monitored variables which increases the variance of the linear association and which has zero covariance with the preceding PCs. The PCs are calculated by Eigen

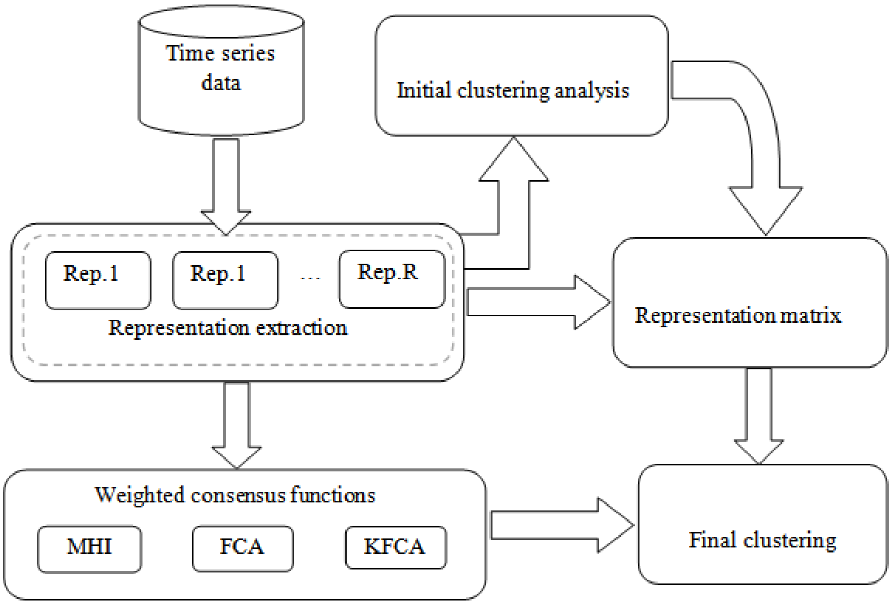

Figure 1. The block diagram of the proposed approach

value decomposition of a data covariance matrix or singular value decomposition of data matrix, which is usually done after performing data centering for each attribute.

Elimination Methods of Unnecessary PCs

The number of PCs equivalent to the number of original variables is generated when the dataset is transformed to the new principal component axis. As most of the variances are described by many of the first PCs, the remaining can be discarded with negligible loss of information. The number of PCs that must be preserved for interpretation is determined using the several criteria that follow:

- *Scree Diagram*, automatically discards the PCs with extremely low variances by plotting the variances in percentage pertaining to the PCs.
- PCs having variance less than a specified threshold value can be discarded.
- PCs having Eigen values less than a specified fraction of the mean Eigen value can be*eliminated.*

The dimensionality of the high dimension dataset is reduced in such a way that dataset containing i attributes are approximately reduced to j attributes, where $i > j$. Later, we integrate the Constraint-Partitioning K-means clustering algorithm to the reduced dataset to produce good and accurate clusters.

Perform Dimensionality Reduction using PCA

Principal Components Analysis method [20-25] performs covariance analysis between factors that decrease the dimensionality of the data. It automatically fits for data sets in multiple dimensions, such as UCI datasets, Parkinson's dataset and Ionosphere dataset. For our proposed work, we are performing PCA using the covariance method. Following is a comprehensive explanation of PCA using the covariance method:

(a) Organize the original dataset in a matrix form.
(b) Subtract the mean from each of the data dimensions.
(c) Find the data covariance matrix.
(d) Find the eigenvectors and Eigen values of the covariance matrix.
(e) Rearrange the eigenvectors and Eigen values in decreasing order.
(f) Remove weaker components from PCs and form transformation matrix consisting of significant PCs.

(g) Find the reduced data set using the reduced PCs.

Suppose we have a sample data X comprising of 'A' records and each having 'B' attributes, and we want to reduce the data such that each record will have only 'C' attributes in such a way that $C < B$.

a) **Organize the high dimensional dataset $\boldsymbol{X}$ in a matrix $\boldsymbol{S}$:** Arrange data as a set of 'N' data vectors $S_1, S_2, \cdots, S_N$ where each S_N represents a single grouped record of the 'B' attributes. Write $S_1, S_2, \cdots, S_N$ as column vectors, each of which has B rows. Place the column vectors into a single matrix S of dimensions $B \times N$.

b) **Subtract the mean from each of the data dimensions:** For PCA to work properly, we have to subtract the mean from each of the data dimensions. Then find the mean along each dimension $b = 1, 2, \cdots, B$ and place the calculated mean values into mean vector of dimensions $B \times I$. Later, subtract the mean vector from each S_N values of the data matrix S. Now, store the mean-subtracted data in the $B \times N$ matrix U. This produces a data set whose mean is zero.

c) **Find the data covariance matrix $\boldsymbol{C}$:** Find the $B \times B$ covariance matrix C from the outer product of matrix U with itself.

d)

$$C = \frac{1}{N} \sum UU^* \tag{1}$$

e) **Find the eigenvectors and Eigen values of the covariance matrix $\boldsymbol{C}$:** Compute the matrix V of eigenvectors which diagnose the covariance matrix C

$$V^{-1}CV = D \tag{2}$$

D, is the diagonal matrix containing Eigen values of C which will take the form of $B \times B$ diagonal matrix. Matrix V is also of dimension $B \times B$ which contains B column vectors corresponding to the B eigenvectors of the covariance matrix C. For a covariance matrix, the eigenvectors correspond to principal components and, the Eigen values to the variance are explained by the principal components.

f) **Rearrange the eigenvectors and Eigen values in decreasing order:** The eigenvector with the *highest* Eigen value is the *principle component* of the data set. Thus, assemble the columns of the eigenvector matrix V and Eigen value matrix D in the decreasing order of Eigen values. This gives us the components in order of significance.

g) **Remove weaker components from PCs and form transformation matrix consisting of significant PCs:** Selecting the number of PCs is a significant question. The largest Eigen values correspond to the principal-components that are related with a large amount of the co-variability, among a number of observed data. Hence, we will remove the weaker principal components from the set of components obtained. For the removal purpose, perform any one of the three suitable methods explained in Section 5.2, and thus generating the transformation matrix P with reduced PCs is formed.

h) **Find the reduced data set using the reduced PCs:** The transformation matrix P is applied to the original data set X to produce the new reduced projected dataset H which we can make use for data clustering.

4.1.2. *Wavelet representation.* Discrete wavelet transform, turns out to be an effective multi-scale analysis tool. Like the preprocessing in the PLS representation, time series is blocked into a set of segments with a window of size. We apply the DWT to each segment for a multi-scale analysis in order to capture local details in a more accurate way. The DWT decomposes time series via the successive use of low-pass and high-pass filtering at appropriate levels. At level j, the coefficients of high-pass filters encode the detailed information, while those of low-pass filters encode, characterize coarse information. For the nth segment with a multi-scale analysis of J levels, the application of the DWT leads to a piecewise representation with all coefficients collectively.

4.1.3. *Normal representation.* The normal representation is the direct representation of the time series dataset (i.e., original format of the dataset), and this representation helps to find out the difference between, the original format with the other representation.

4.2. **Initial clustering.** The initial clustering algorithm is applied to each representation obtained from the representation of the extracted module. As the results of initial clustering process, we get the partition of the given time series dataset generated as per representations. Consider the K-means clustering algorithm [18,19] utilized for initial clustering. By proceeding with the initial clustering algorithm on variant of initialization conditions, the multiple partitions based on the representation are produced. These multiple partitions help to obtain the weighted clustering ensemble for final partition.

Consider the set of given time series temporal data $T = \{(x_i, t_i)\}$ where $(1 \le i \le N)$, due to the high dimensionality of raw data, that agree to be normal and, the PCA representation. The PCA representation helps to reduce the high dimensionality of the raw data into convenient format and the normal representation signifies the same raw time series data for comparing the PCA representation clustering and normal representation clustering. These two representations are fed to the initial clustering process to generate the multiple partitions $P = (p_k)$ where $(1 \le k \le m)$. The obtained partitions from the initial clustering are dissimilar due to the representations.

4.3. **Computation of weights.** The obtained partitions (p_k) offer a weight $w^{\alpha}(p_k)$ for each partition by weighting method. Our partition weighting method allocates the weights to each partition based on the clustering validation standard α. In this paper, we utilize the following clustering validation standards like *Modified Huber's index* (MHI), *Fuzzy Weighting Algorithm* (FWA), *Kernal Fuzzy Weighting Algorithm* $(KFWA)$.

The each partition consists of, set of objects $P_k = \{(x_i, t_i)\}$ where $1 \le i \le n$, in order to find out the MHI clustering validation standard. Initially we find out the $n \times n$ cluster distance matrix D_{st} which is computed from each partition and the elements of the distance matrix D_{st} denote the distance between the centroid of the clusters where objects x_s and x_t belong to. The following Equation (3) helps to find out the clustering validation standards [14].

$$\left(\alpha^1\right) \big| \, MHI\,(p_k) = \frac{n\,(n-1)}{2} \sum_{s=1}^{n-1} \sum_{t=s+1}^{n} P_{st}\, D_{st} \tag{3}$$

From the above Equation (3), we can obtain the weights for all partitions, from all the weights of the partitions we choose one weight which is high MHI value that corresponding partition has compact and well-separated clustering structure.

The next clustering validation standard is fuzzy weighting algorithm FWA, which helps to describe the relation between the objects and corresponding cluster centroid. This fuzzy weighting algorithm helps to find out the amount of compactness of the cluster. The following Equations (4) and (5) help to find out the FWA clustering validation standard.

$$\left(\alpha^2\right) \big| \, FWA\,(p_k) = \frac{\sum\limits_{i=1}^{n} x_{ij}^{k}}{n\left(x_{ij}^{k}\right)} \tag{4}$$

$$\left(x_{ij}^{k}\right) = \sum_{q=1}^{C} \left(\frac{\left\| x_i - c_q \right\|^2}{\left\| x_i - c_j \right\|^2} \right) \tag{5}$$

In the above Equation (5), i stands for objects j signifies, the corresponding cluster centroid, q denotes the other cluster centroid, C represents the total number of clusters and $n(x_{ij}^{k})$ is the total number of $\left(x_{ij}^{k}\right)$. From the above Equation (4), we get the weight of each partition from that, we can select the highest FWA value among them. Since the high FWA value represents the well separated clustering structure.

The next cluster validation standard is kernel fuzzy weighting algorithm $KFWA$, which helps to find out the intensity of the object, of the cluster centroid for cluster. In following Equation (6), i stands for objects, j signifies corresponding cluster centroid, and $n(x_{ij}^{k})$ is total number of $\left(x_{ij}^{k}\right)$. From the above Equation (5), we get the weights for all of the partitions from which, the weight is selected from where the high $KFWA$ value is chosen. Since the high $KFWA$ value of the partition has the high intensity value of clustering. The following Equations (6) and (7) help to find out the $KFWA$ clustering validation standard.

$$\left(\alpha^{3}\right)\Big|\, KFWA\left(p_{k}\right) = \frac{\sum\limits_{i=1}^{n} x_{ij}^{k}}{n(x_{ij}^{k})} \tag{6}$$

$$\left(x_{ij}^{k}\right) = \exp\left(-\left\|x_{i} - x_{j}\right\|\right) \tag{7}$$

From the outcome of the three types of clustering validation standards, we achieve the weights of the partition of each representation. The final weight of the all partition is calculated using the following formula.

$$Fin.W = \frac{\sum\limits_{i=1}^{n} \alpha}{n} \tag{8}$$

4.4. **Ensemble of weighted for final clustering.** In our algorithm, there are three types of weighting algorithm; normally these partitions are not the same, so we make the resolution through the generation of representation matrix. This representation matrix is the input for final clustering; the following Figure 2 signifies the representation matrix RM at size $[N \times R]$ where N signifies the number of data point present in the time series dataset and R represents number of representations.

$$RM = \begin{bmatrix} p_{11}^{k} & p_{12}^{k} & \cdots & p_{1r}^{k} & \cdots & p_{1R}^{k} \\ p_{21}^{k} & p_{22}^{k} & \cdots & p_{2r}^{k} & \cdots & p_{2R}^{k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{n1}^{k} & p_{n2}^{k} & \cdots & p_{nr}^{k} & \cdots & p_{nR}^{k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{N1}^{k} & p_{N2}^{k} & \cdots & p_{Nr}^{k} & \cdots & p_{NR}^{k} \end{bmatrix}$$

Figure 2. The representation matrix at size $[N \times R]$

This representation matrix is input to the final clustering here, the final clustering is K-means algorithm, but the way of finding the distance matrix is different from normal K-means algorithm by considering the final weight $(Fin.W)$. The following Formula (9) is used to calculate the distance matrix.

$$DM = \left(\sqrt{Fin.W\left(p_{11} - p_{21}\right)^{2} + Fin.W\left(p_{12} - p_{22}\right)^{2}}\right) \tag{9}$$

In the above Equation (9), the distance of the point is multiplied with the final weights. Based on the weights the groups are generated, from where we achieve the accurate results.

As the result of the final clustering, we achieve the number of clusters as per the user given and their elements. The first row of the representation matrix signifies the first data of the times series data, based on the clusters found.

5. Experimental Results and Performance Analysis Discussion.

The experimental results of the proposed approach for clustering of time series data are presented in this section. The proposed approach has been implemented using MATLAB (Matlab7.10) and the performance of the proposed system is analyzed using the accuracy and running time. With the help of the reference [14], we calculate the accuracy of the clustering algorithm for each dataset in the benchmark datasets [26]. The running time is evaluated by how much time (seconds) the proposed system takes for clustering each dataset in the benchmark.

5.1. Experimental environment and dataset.

We have used the real dataset for clustering of time series data and experimentation is performed on a 3.0 GHz dual core PC machine with 2 GB main memory. We have utilized the real dataset called time series benchmarks which consists of nine time series datasets [26] that has been collected to evaluate the time series clustering and the clustering algorithm, informs about the dataset given in the following Table 2 which consists of number of classes for each time series dataset and the size of each dataset includes training and testing the length of the dataset.

Table 2. Information about the benchmark time series datasets

Dataset	Number of class	Size of dataset (Trainning+Testing)	Length
CBF	3	300+900	128
ECG 200	2	100+100	96
Face Four	4	24+88	350
Gun-Point	2	50+150	150
Lighting2	2	60+61	637
Lighting7	7	70+73	319
OSU Leaf	6	200+242	427
Trace	4	100+100	275
Yoga	2	300+3000	426

5.2. Evaluation of running time based on the number of clusters.

The running time of the proposed system varies depending on the number of clusters for each data set. In this paper, we evaluate the running time of the proposed approach in two categories. First one is based on the number of the class present in the data set of the benchmark and the second one is based on the number of classes of the data set given by us. Using each dataset from the benchmark that consists of the number of classes, which is based on the class of the dataset, we can calculate the running time.

From Figures 3-5, we can analyse the running time of the proposed system, which does not vary so much based on the number of clusters. The running time of the proposed system is almost similar, by varying the number of clustering of data sets in the benchmark. The size of the each dataset is different since it affects the running time of the proposed system. We consider the dataset “yoga” which is, the highest dataset consisting of 3000 and above. Since it takes more running time than the other datasets, while comparing with the other dataset such as the CBF, ECG and Gun-Points are having less number of data, because their running time is very less than other datasets.

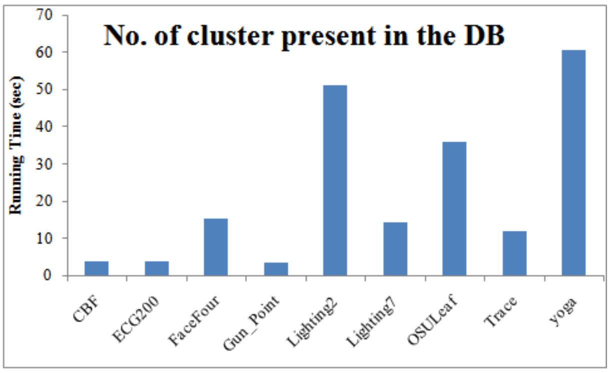

Figure 3. The time taken of clustering for No. of cluster present in each dataset

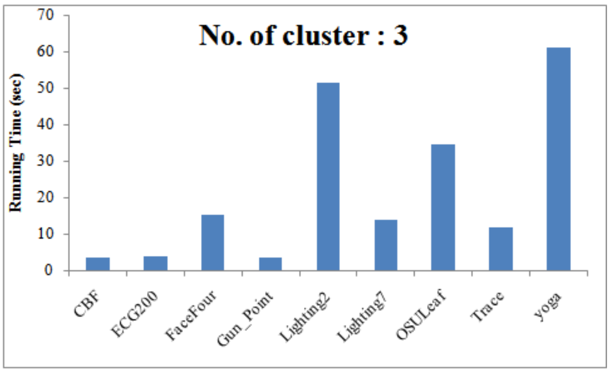

Figure 4. The time taken for clustering for No. of cluster is 3

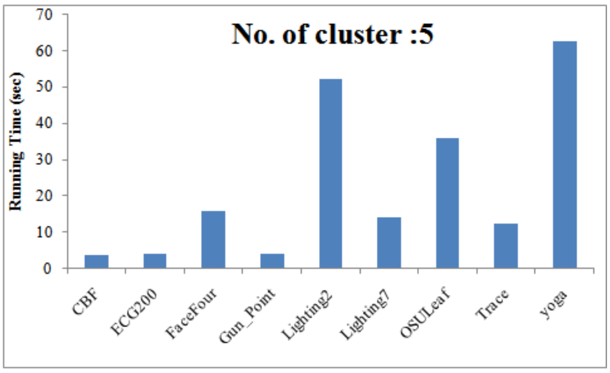

Figure 5. The time taken for clustering for No. of cluster is 5

5.3. **Evaluation of accuracy of the proposed approach through comparing with the previous algorithm.** Table 3 presents the detailed analysis of the accuracy of the proposed approach in nine different datasets. The comparison of the proposed approach was done with the eight different existing works in which, three works are related with temporal proximity-based, four works are related with single representation based on k-means clustering algorithm and final one, is based on model-based technique. From the table, the maximum accuracy for the dataset of "CBF" is 70% which has been achieved by the proposed approach while the second highest accuracy is obtained by the k-means clustering. For the "CBF", the proposed technique showed nearly 8% of performance improvement. For the "ECG 200" dataset, the proposed has achieved the accuracy of 77% which is nearly 7% improvement over the model-based existing technique.

TABLE 3. The accuracy of proposed system and previous techniques

Data Set	Temporal-Proximity based			Model based	Single Representation based on K-mean				Proposed Approach
	K-mean	HC	RPCCL	K-HMM	PCF	DFT	PLS	PDWT	
CBF	62.6	50.9	63.2	60.1	57.6	60.0	60.6	60.5	**70**
ECG 200	69.8	59.4	67.4	70.3	59.3	60.3	60.4	64.0	**77**
Face Four	66.9	62.7	65.3	**69.1**	50.7	58.6	59.6	60.4	54.16
Gunpoint	50.0	41.9	46.1	43.8	43.4	45.4	48.5	49.4	**72**
Lightening2	61.1	62.0	56.3	57.7	53.3	53.0	55.8	57.1	**75**
Lightening7	48.4	39.4	**53.0**	51.2	42.5	43.5	49.9	49.1	42.85
OSU Leaf	37.8	39.1	32.2	**44.2**	30.6	26.4	33.5	31.7	38
Trace	48.5	40.2	53.1	50.9	40.5	42.0	42.3	45.4	**76**
Yoga	51.7	44.2	50.8	48.5	45.6	48.6	47.6	48.5	**72**

The performance of the temporal proximity-based approach behaves well in the dataset of "Lightening7" and also, K-HMM provided the good accuracy in "Face Four" and "OSU Leaf" as compared with the proposed approach. For the Gunpoint and Lightening2 dataset, the proposed approach provided 72% and 75% accuracy as the existing k-means algorithm showed only 50% and 61%. Similarly, for the datasets such as, "Trace" and "Yoga", the proposed approach showed nearly 25% and 20% performance improvement as compared with existing algorithms. Overall, the proposed approach provided more improved results for the six datasets out of nine datasets experimented. The improvement ensures that the contribution made in different steps of the proposed approach is well and good for time-series data clustering.

The improved performance makes sure that the proposed approach will provide good results on different applicability. The proposed time series data clustering has good practical applicability in various fields like, medical imaging, biological science and telecommunication filed. The practical applicability of time series data clustering is discussed in [29-31] for the different fields. In [29], human motion was clustered in a temporal way and in [30], the trajectory analysis was carried out using time series data clustering. The important application for the medical field can be found out in fMRi cluster analysis [31].

6. **Conclusions.** We have developed an ensemble-based time series data clustering for high dimensional data. At first, the time series datasets are converted into different representation formats; subsequently, the representations are fed into the initial clustering algorithm, and the effect of the initial clustering gives multiple partitions of the each representation. For each partition, we have applied various clustering validation standards for evaluating each partition received from the initial clustering and then we calculate the final weight for finding the final clustering. The representation matrix is generated with the help of representation and partition, and this matrix is the input for the final clustering. While calculating the final clustering, the distance matrix is generated with the final weight received from the various clustering validation standards. The experimentation process is carried out with the help of different types of datasets from benchmark to evaluate our proposed approach and we achieve 70% of accuracy, more than the previous algorithms.

REFERENCES

[1] S. Das, A. Abraham and A. Konar, Automatic clustering using an improved differential evolution algorithm, *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol.38, no.1, 2008.

[2] H. Izakian, A. Abraham and V. Snasel, Fuzzy clustering using hybrid fuzzy c-means and fuzzy particle swarm optimization, *World Congress on Nature and Biologically Inspired Computing*, India, pp.1690-1694, 2009.

[3] T. W. Liao, Clustering of time series data – A survey, *Pattern Recognition*, vol.38, pp.1857-1874, 2005.

[4] E. Keogh and S. Kasetty, On the need for time series data mining benchmarks: Asurvey and empirical, *Knowl. Data Discov.*, vol.6, pp.102-111, 2002.

[5] A. Jain, M. Murthy and P. Flynn, Data clustering: A review, *ACM Comput. Surv.*, vol.31, pp.264-323, 1999.

[6] H. Liu and D. E. Brown, A new point process transition density model for space-time event prediction, *IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol.34, no.3, pp.310-324, 2004.

[7] S. Policker and A. B. Geva, Nonstationary time series analysis by temporal clustering, *IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybern.*, vol.30, no.2, pp.339-343, 2000.

[8] Y. Yang and K. Chen, Senior member, time series clustering via RPCL network ensemble with different representations, *IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol.41, no.2, 2011.

[9] V. Kavitha and M. Punithavalli, Clustering time series data stream – A literature survey, *International Journal of Computer Science and Information Security*, vol.8, no.1, 2010.

[10] K. Premalatha and A. M. Natarajan, A new approach for data clustering based on PSO with local search, *Computer and Information Science*, vol.1, no.4, 2008.

[11] S. Das, A. Abraham and A. Konar, Automatic clustering using an improved differential evolution algorithm, *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol.38, no.1, pp.218-237, 2008.

[12] Y.-T. Kao, E. Zahara and I.-W. Kao, A hybridized approach to data clustering, *Expert Systems with Applications*, vol.34, no.3, pp.1754-1762, 2008.

[13] G. Zhao and W. Deng, HMM-based hierarchical clustering of gene expression time series data, *School of Computer Science & Technology*, vol.47, no.32, pp.167-169, 2011.

[14] Y. Yang and K. Chen, Temporal data clustering via weighted clustering ensemble with different representations, *IEEE Trans. on Knowledge and Data Engineering*, vol.23, no.2, pp.307-320, 2011.

[15] Y. Yang and K. Chen, Time series clustering via RPCL network ensemble with different representations, *IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol.41, no.2, pp.190-199, 2011.

[16] P. Maji and S. Paul, Microarray time-series data clustering using rough-fuzzy C-means algorithm, *Proc. of IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp.269-272, 2011.

[17] Pallavi and S. Godara, An improved clustering approach on time series data set, *Proc. of National Workshop-Cum-Conference on Recent Trends in Mathematics and Computing*, no.4, 2012.

[18] A. Vattani, K-means requires exponentially many iterations even in the plane, *Discreate and Computational Geometory*, vol.45, pp.596-616, 2011.

[19] C. Ding and X. He, K-means clustering via principal component analysis, *Proc. of International Conference of Machine Learning*, pp.225-232, 2004.

[20] A. M. Aguilera, R. Guti'errez, F. A. Ocaña and M. J. Valderrama, Computational approaches to estimation in the principal component analysis of a stochastic process, *Appl. Stoch. Models Data Anal.*, vol.11, pp.279-299, 1995.

[21] A. Ali, G. M. Clarke and K. Trustrum, Principal component analysis applied to some data from fruit nutrition experiments, *Statistician*, vol.34, pp.365-369, 1985.

[22] C. Croux and G. Haesbroeck, Principal component analysis based on robust estimators of the covariance or correlation matrix, *Influence Functions and Efficiencies. Biometrika*, vol.87, pp.603-618, 2000.

[23] S. A. Farmer, An investigation into the results of principal component analysis of data derived from random numbers, *Statistician*, vol.20, pp.63-72, 1971.

[24] G. W. Horgan, Principal component analysis of random particles, *J. Math. Imaging Vision*, vol.12, pp.169-175, 2000.

[25] S. Konishi and C. R. Rao, Principal component analysis for multivariate familial data, *Biometrika*, vol.79, pp.631-641, 1992.

[26] E. Keogh, *Temporal Data Mining Benchmarks*, http://www.cs.ucr.edu/∼eamonn/time_series_data, 2010.

[27] D. Binu and A. George, KF-PSO: Hybridization of particle swarm optimization and kernel-based fuzzy C-means algorithm, *International Conference on Advances in Computing, Communications and Informatics*, pp.512-515, 2013.

[28] A. George and D. Binu, DRL-Prefixspan: A novel pattern growth algorithm for discovering downturn, revision and launch (DRL) sequential patterns, *Central European Journal of Computer Science*, vol.2, no.4, pp.426-439, 2012.

[29] F. Zhou, F. De La Torre and J. K. Hodgins, Hierarchical aligned cluster analysis for temporal clustering of human motion, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.35, no.3, pp.582-596, 2013.

[30] W. Hu, X. Li, G. Tian, S. Maybank and Z. Zhang, An incremental DPMM-based method for trajectory clustering, modeling, and retrieval, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.35, no.5, pp.1051-1065, 2013.

[31] V. P. Oikonomou and K. Blekas, An adaptive regression mixture model for fMRI cluster analysis, *IEEE Trans. on Medical Imaging*, vol.32, no.4, pp.649-659, 2013.

[32] J. Zakaria, A. Mueen and E. Keogh, Clustering time series using unsupervised-Shapelets, *Proc. of IEEE the 12th International Conference on Data Mining*, pp.785-794, 2012.

[33] H. Kremer, S. Gunnemann, A. Held and T. Seidl, Effective and robust mining of temporal subspace clusters, *Proc. of IEEE the 12th International Conference on Data Mining*, pp.369-378, 2012.