Distributed and Parallel Databases (2021) 39:813–832
https://doi.org/10.1007/s10619-021-07324-3

An efficient sampling-based visualization technique for big data clustering with crisp partitions

K. Rajendra Prasad[1] · Moulana Mohammed[2] · L. V. Narasimha Prasad[3] · Dinesh Kumar Anguraj[2]

Accepted: 29 January 2021 / Published online: 19 February 2021

Abstract

The data cluster tendency is an emerging need for exploring the big data cluster analysis tasks. The data are evaluated based on the number of clusters is known as cluster tendency. Many visualization techniques have been developed for the detection of cluster tendency. Some of the existing techniques include Visual Assessment Tendency (VAT), spectral-based VAT (SpecVAT), and improved VAT (iVAT), are considerably succeeded for an assessment of cluster tendency for small datasets. A bigVAT is another method that was recently developed for the estimation of cluster tendency of big data. It is perfect for deriving the clustering tendency in visual form for big data. However, it is intractable to explore the data clusters for large volumes of data objects. The proposed work addresses the clustering problem of bigVAT with the derivation of sampling-based crisp partitions. The crisp partitions will accurately predict the cluster labels of data objects. This research is based on big synthetic and big real-life datasets for demonstrating the performance efficiency of the proposed work.

Keywords Cluster tendency · Visualization techniques · Data clustering · Crisp partitions · Sampling

✉ K. Rajendra Prasad
krprgm@gmail.com

Moulana Mohammed
moulanaphd@gmail.com

L. V. Narasimha Prasad
lvnprasad@yahoo.com

Dinesh Kumar Anguraj
dineshngpit@gmail.com

[1] Department of CSE, Rajeev Gandhi Memorial College of Engineering & Technology, Nandyal, Andhra Pradesh, India

[2] Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India

[3] Department of CSE, Institute of Aeronautical Engineering, Hyderabad, Telengana, India

1 Introduction

Big data [3] is a massive representation of vast volumes of either regular or complex datasets. Big data clustering problem depends on two key steps: cluster tendency and cluster partitions of big data. The cluster tendency [4] accesses the preliminary information about the number of clusters. An efficient k-means [5] algorithm discovers the cluster quality when prior cluster tendency is known (i.e., k is known). The value of cluster tendency is assigned either by the user or any external interference. Generally, such 'k' is intractable in the k-means algorithm for big data. Visualization techniques [6, 7], are used for determining the prior information about cluster tendency. Visual Access Tendency (VAT) is proposed by Bezdek et al. [1] for accessing the information about several clusters with the dissimilarity matrix [2] of data objects. The dissimilarity matrix is computed by finding the dissimilarity features of data objects using distance metrics [7]. These two distance metrics are recommended in many data clustering applications, which are Euclidean and cosine metrics [8]. In data clustering, distance metrics play a critical role in dissimilarity features computation among the different data objects. The other visualization techniques such as SpecVAT and iVAT are used for the detection of cluster tendency for complex datasets. The bigVAT enhances visualization techniques for accessing the clustering tendency of big data. Visualization techniques have shown the clusters in visual form as 'square-shaped dark colored blocks'. The number of these blocks refers to the clustering tendency. The existing techniques inadequate to address the clustering problem completely, i.e. they detect only the cluster tendency and cannot derive the data clustering results. A combination of visualization techniques with either k-means or minimum-spanning tree clustering can solve the clustering problem. However, these hybrid approaches are expensive for big data. The proposed work derives the sampling-based crisp partitions from the visual clusters (i.e., square-shape dark colored blocks) and also it predicts the cluster labels of data objects. The bigVAT is extended with sampling procedure and crisp partitions in the proposed work, and it is significantly essential for the cluster tendency assessment and cluster partitions of big data. Figure 1 shows the critical procedural steps of big data clustering with sampling-based crisp bigVAT (SC-bigVAT). Intercluster data objects are selected by determining the data objects with the maximum dissimilarity value which are nearest objects and those objects are derived for the inter-clustering data objects. It is performed with a min–max sampling procedure [9]. The best samples of inter-cluster data objects are selected to calculate the dissimilarity matrix for the set of selected sample data objects. A bigVAT is applied on the dissimilarity matrix of sample inter-cluster data objects to visualize the visual image clusters which are indicated with black colored square blocks. Squareness and edginess values are derived for the visual image clusters, useful for the crisp partitions derivations.

The crisp partitions are derived for sufficient samples of big data. The crisp partitions are derived from the visual image clusters for finding the cluster labels of big data objects. The crisp-partitions depict the aligned k-partitions of the

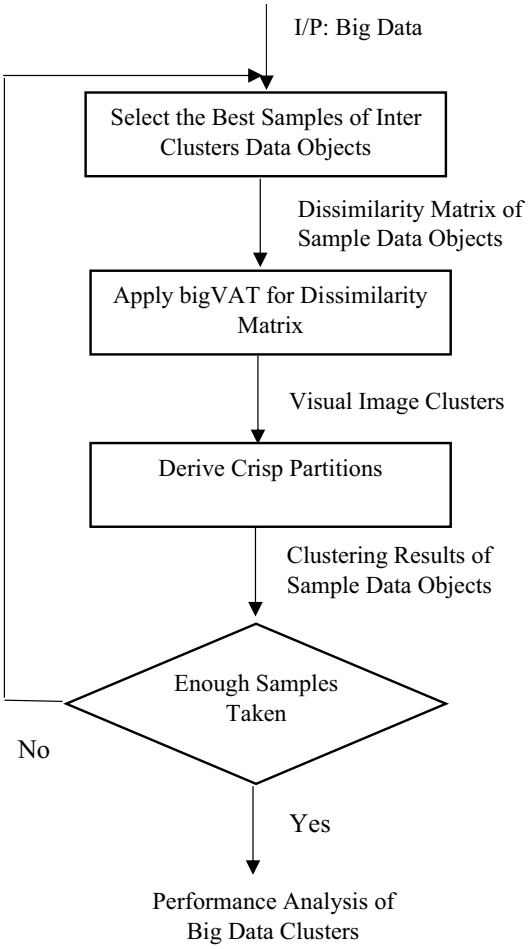

Fig. 1 Sampling-based Crisp bigVAT

visual image and it is derived from the detection of obtained square-blocks (dark-colored) along the diagonal in the visual image. The square edginess of the diagonal determines the crisp partitions and also it computes the difference between diagonal and non-diagonal square-blocks in the visual image.

The highlights of contributions of the work are described as follows:

1. Cluster tendency of big data is determined for understanding the prior information about big data clusters
2. Best samples of big data are selected from the derived inter-clusters
3. Visualize the image clusters for the big data in the form of square-shaped blocks
4. Compute the crisp partitions for discovering the big data clusters
5. Performance analysis is conducted for illustrating the efficiency of proposed work

The layout of this article is organized as follows. Section 2 describes the literature work. The proposed sampling-based visualization techniques are explained in Sect. 3. The experimental results and performance analysis are presented in Sect. 4. Finally, Sect. 5 draws the conclusion and scope of the research work.

2 Literature work

Data clustering problems are used in the applications like pattern recognition [19], text clustering [20], image clustering [21], trajectory detection [22], bio-mining [23] etc. The k-means and hierarchical clustering methods are the prime clustering methods [10] that can effectively generate the data partitions for unlabelled data. In such methods, the user interference is necessary for the pre-assignment of the 'k' value (also known as cluster tendency). The k-means is the most popular technique due to its applicability in information or data science. These clustering methods suffer from the issues of cluster tendency. Visualization techniques are useful for the assessment of cluster tendency of unlabelled datasets. VAT [11] is the basic visualization technique, and it is developed for the detection of cluster tendency. It finds the dissimilarity features among the data objects and places the values in matrix form, called a Dissimilarity Matrix (DM). Similarity features computation shows the major impact in data clustering problems. Most of the data clustering algorithms derive the similarity (or dissimilarity features) among the data objects with either Euclidean or cosine metrics, which are shown in Eq. (1) Moreover, (2), respectively, the dissimilarity (or similarity) is computed among the two objects 'o1′ and 'o2′ for the 'n' properties.

$$\mathrm{D}(\mathrm{o1},\mathrm{o2}) = \sqrt{(\mathrm{x1}-\mathrm{y1})^2 + \ldots + (\mathrm{xn}-\mathrm{yn})^2} \tag{1}$$

$$\text{Similarity } (\mathrm{o1},\ \mathrm{o2}) = 1 - \mathrm{D}(\mathrm{o1},\mathrm{o2}) \tag{2}$$

$$\mathrm{Cosine}(\mathrm{o1},\mathrm{o2}) = \frac{\mathrm{o1.o2}}{\mathrm{o1o2}} \tag{3}$$

$$\text{Similarity} = \text{Normalized } (\mathrm{Cosine}(\mathrm{o1},\mathrm{o2})) \tag{4}$$

$$\mathrm{MVS}(\mathrm{o1},\mathrm{o2}) = \mathrm{avg}\left(\sum_{\substack{vpno=1,\\ vp \in O-\{\mathrm{o1},\mathrm{o2}\}}}^{\mathrm{N}-2} \mathrm{Sim}(\mathrm{d1},\ \mathrm{d2})\right) \tag{5}$$

Equations (3) and (4) have shown the similarity features obtained from Euclidean and Cosine results, respectively. The dissimilarity (or similarity) computation among the data objects is performed based on the magnitude and direction of data vectors

for the data objects, whereas in Euclidean, consider only the magnitude among the data objects. Cosine-based data clustering succeeds specifically in text data clustering problems [18]. The extension of cosine metric is developed in [19] for finding the most accurate similarity features among the data objects, which is a multi-viewpoint cosine-based similarity metric (MVS) [25] and it is shown in Eq. (5), in which sim (d1, d2) = cos (o1-vp, o2-vp). Traditional cosine uses the single viewpoint as the reference in similarity features computation among any two objects, whereas MVS uses the multiple viewpoints in similarity features computation. In the text data clustering applications, it shows the robust performance compared to Euclidean and traditional cosine metrics.

VAT is enhanced as iVAT[12] for detection of cluster tendency for path-shaped datasets. The basic VAT approach is shown in Algorithm 1. Initially, it takes the dissimilarity matrix 'dissM[][]' for the set of data objects 'n'. The aim is to find the re-ordered dissimilarity matrix (RDM) based on the ordering of distances among the various data objects. Finally, displays the visual image of RDM for showing the clusters as square-shaped dark colored blocks. Another method, say, SpecVAT[13] finds the best spectral features by deriving the Eigenvectors [14]. Affinities among the data objects are computed for finding the Laplacian matrix of unlabelled data. The largest k-Eigen vectors are used for the selection of spectral features of data.

```
Algorithm 1: VAT (int dissM[ ][ ],int n) [1]

Step1:
        Let IV= { } ; JV={0,1,.....n-1}
        Determine  max of dissM[ ] [ ], and its index
        cell is (i,j)
        P(0)=i; IV={i},JV=JV-{IV};
Step2:
        for (s=1;s<n;s++)
        {
          Find(i,j) from min {dissM[i][j], where
                                i€ IV, j є{JV}}
          IV= {IV} U{j}; JV={JV}-{IV};
          P(s)=j;
        }
Step3:
/* Reordered Dissimilarity Matrix Comutation*/
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
              RDM=dissM(P[i],P[j]);
Step 4:
    Display Image(RDM)
```

The Dissimilarity matrix is computed for the 'n' data objects concerning the spectral features in SpecVAT. The high-dimensional clusters assessment is performed well by finding the spectral features in SpecVAT. However, both VAT and SpecVAT are unable to handle complex datasets like path-shaped datasets. It also presents the best assessment of cluster tendency for various datasets. Another technique, bigVAT [15] is scalable, and overcomes the computational issue in the assessment of cluster

tendency for the big data. State-of-the-art visualization techniques are adequate for finding the clustering tendency for both complex and big datasets. These are significantly used for the pre-estimation of cluster tendency in data clustering methods. Both the visualization techniques and k-means are recommended for the data clustering problems. These two types of hybrid techniques are proposed in [16] for performing data clustering with known cluster tendency, which is VAT-based k-means and VAT-based MST-clustering [17]. The VAT-based k-means is efficient and rapid for data clustering when compared to VAT-based-MST-clustering. In such methods, the normal or medium-sized data sets are suitable for finding both cluster tendency and discovering the quality of data clusters. Big data is expensive when considering the time and space values due to running both visualization techniques and data clustering algorithms. In this article, sampling-based crisp partitions are derived from the visual image clusters, which groups the data objects into clusters according to predicted cluster labels. It is less expensive for big data than hybrid approaches, and its methodology is explained in the next section.

3 Sampling based visualization technique for big data clustering

Algorithm 2: SC-bigVAT

Input :

N - Number of data objects
OBJ - set of objects {o1,o2,……oN}

Output :

K - cluster tendency (or number of clusters)
C - data clusters, {c1, c2,…ck}

Methodology :

1. Sample_data_objects = SICP(OBJ)
2. Use bigVAT on Sample_data_Objects for applying the following steps.
   a. Visual_Image = Image(Reordered _dissimilarity_matrix (Sample_data_objects))
   b. Extract the Visual_Profiles from Visual_Image of Sample_data_objects
   c. Access the cluster tendency 'k' with the count information of appeared square-shaped-dark-colored-blocks of the visual image
   d. Determine the crisp partitions from the visual profiles by the following step

      C = Crisp_Partitions(Visual_Profiles,k)
3. Discover the data clusters 'C.'

The best samples are returned to the calling procedure of sampling inter-cluster viewpoints (SICP) and it is shown in step 1 of algorithm 2. The Re-Ordered Dissimilarity Matrix (RDM) is computed for the selected sample data objects of SICP and obtains the image of RDM. The profiles of RDM visual images are extracted with bigVAT for faster assessment of cluster tendency ‘k’. The crisp partitions of the sample data are obtained for the ’k’ clusters by calling procedure Crisp_partitions in

step 2. Finally, the cluster labels of data objects determined in the Crisp_partitions' procedure of Crisp_partitions returns the same for discovering data clusters in step 3.

In SICP, initially, a random object is selected among the 'N' data object, and it is considered as the centroid of the initial cluster. The distance between the other objects are computed concerning the initial cluster centroid and choose the maximum distance for defining another inter-cluster data object (or another centroid). Further, the distance of objects is updated according to the determined centroid. The distinguished sample viewpoints are selected from the centroids based created clusters using SICP. A few sample viewpoints of each cluster are considered instead of selecting the samples from the entire data. The sample viewpoints by themselves denote the self-organized clustering structures hence, it is sufficient for the representation of big clustering structures. These viewpoints are selected as the reference objects when measuring the dissimilarity (or dissimilarity) among the different data objects. The samples are selected based on the minimum distance. The algorithm 3 illustrates these steps for determining the centroids of clusters and the data objects of respective cluster centroids. The sample inter-cluster data objects are returned at the end of SICP.

Algorithm 3: SICP(OBJ)

1. Choose the initial centroid for the cluster by random selection object 'o_r' from {OBJ} or {o1, o2, o3,…oN}
2. Compute the distances between o_r to other objects of OBJ in order to find the index of object based on maximum distance values, max_index=argmax $_{I \in \{1,2,..N\}}$ {distance(o_r, o_I)}and max_dist= distance(o_r, o_I)), max_index show the index of data object and has selected as centroid

   Sample_Object_Indices = {max_index}
   SOI = Sample_Object_Indices
3. Find other centroids for the remaining predicted clusters.
   a. For i = 1 to N
      Ds_i=min(max_dist, distance(o_{max_index}, o_i))
   b. Update the other centroids
      Index of centroid is derived from argmax $_{I \in \{1,2,..N\}}$ {Ds_I}, update max_index and max_dist,
      SOI=SOI U{max_index}
      Repeat Step 3 until obtaining the sufficient centroids
4. The remaining data objects are moved to the nearest centroids of the clusters.
5. Select the random samples of inter-cluster data objects, S
6. Return S

Algorithm 4: Crisp_Partitions (Visual_Profiles)

1. Visual_Image = Image(Visual_Profiles)
2. Derive the difference between diagonal and non-diagonal square-shaped blocks of the Visual_Image based on the following steps of crisp partitions
   a. The crisp partition matrix, U_{kn} denotes the 'k' clusters of n data objects (for the profiles)
   b. The partitions are obtained by maximizing the function *f(U, D)*, where *f(U,D)* is represented in the following Eqn. (5) [24]
   c. Extract the aligned k-partitions, 'C.'
3. Return the data objects of C

The visual image consists of black or grey colored square-shaped blocks, and the difference between diagonal and non-diagonal blocks is computed using Eq. (6).

$$f(U,D) = \left(\frac{\sum_{i=1}^{k} \sum_{si,tnoti} d_{st}^{*}}{\sum_{i=1}^{k} (n - n_i) n_i} \right) - \left(\frac{\sum_{i=1}^{k} \sum_{s,ti,s \neq t} d_{st}^{*}}{\sum_{i=1}^{k} (n_i^2 - n_i)} \right) \quad (6)$$

The function *f(U,D)* is maximized for the k-aligned partitions. These steps are illustrated in Algorithm 4 for returning the data objects of derived clusters 'C' from the crisp partitions.

The experimental details of the proposed work and performance study are described in the following sub-section.

4 Experimental and performance analysis

The performance analysis is demonstrated for the visualization techniques in the data clustering of big data.

4.1 Details of datasets

In the experiment, six big synthetic and four big real datasets were used, in which the efficiency of the proposed visualization technique is analyzed for the lakhs of data objects. Table 1 presents the details of the big data used in the experimental study.

The six synthetic datasets are generated in MATLAB 2020 with the setting of required gaussian parameters and shown visually in Fig. 2. These datasets are created in two-dimensional space. There are four real datasets are freely available in the UCI Machine Learning Data Repository [26] and [27], in which the features are scaled between [0, 1].

Table 1 Description of synthetic and real datasets

S. No	Nature of the data	Name of the dataset	Number of data objects
1	Synthetic Data	S-1 (2-Clusters Full Moon Data)	100,000
2	Synthetic Data	S-2 (2-Clusters half-Kernel Data)	100,000
3	Synthetic Data	S-3 (3-Clusters Gaussian Data)	150,000
4	Synthetic Data	S-4 (4-Clusters Corners Data)	200,000
5	Synthetic Data	S-5(4-Clusters Outlier Data)	200,000
6	Synthetic Data	S-6 (5-Clusters Gaussian Data)	250,000
7	Real Data	MiniBooNE (2-Clusters Data)	130,064
8	Real Data	FOREST (7-Clusters Data)	581,012
9	Real Data	MNIST (10-Clusters Data)	70,000
10	Real Data	KDDCUP (23-Clusters Data)	4,898,431

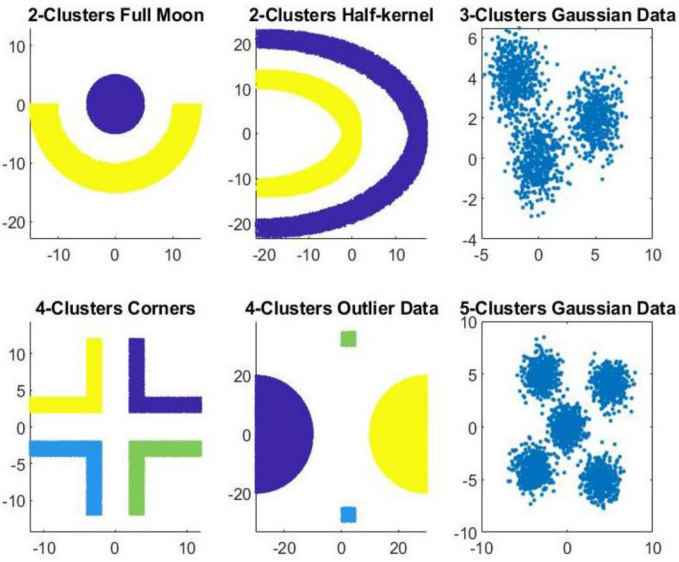

Fig. 2 Big synthetic datasets. **a** Under Euclidean. **b** Under Cosine

For the big synthetic and real datasets, the proposed SC-bigVAT visualization technique displays sample profiles' visual image, unlike the entire VAT image. Thus, its applicability is more for big data than other visualization techniques. The experimental of SC-bigVAT is conducted under two metrics, i.e., Euclidean and Cosine, and their clusters generation are shown in the form of visual image clusters, which are illustrated in Figs. 3, 4, 5, 6, 7, 8, 9, and 10, 11 for 2-clusters moon data, half-kernel data, 3-clusters gaussian, 4- clusters gaussian, 5-clusters gasussian, 2-clusters MiniBooNE, 7-clusters Forest data, and 10-clusters real MNIST data respectively.

Figure 11 shows the assessment of cluster tendency for the real big data-MNIST (10-clusters data). It also is shown those 10 visual square-shaped black colored blocks in diagonal of SC-bigVAT image. Similarly, Fig. 12 showed the assessment of KDDCUP dataset. The two visual images are obtained for finding the data object's similarities with Euclidean and cosine metrics. The dissimilarity and re-ordered dissimilarity matrices are obtained from the dissimilarity (or similarity)

(a) Under Euclidean (b) Under Cosine

Fig. 3 Assessment of 2- Clusters Moon Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance Metrics (sample = 10% of original data). **a** Under Euclidean. **b** Under Cosine

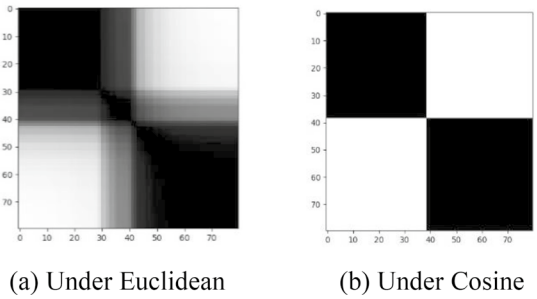

Fig. 4 Assessment of 2- Clusters half-Kernel Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance Metrics (sample = 10% of original data). **a** Under Euclidean. **b** Under Cosine

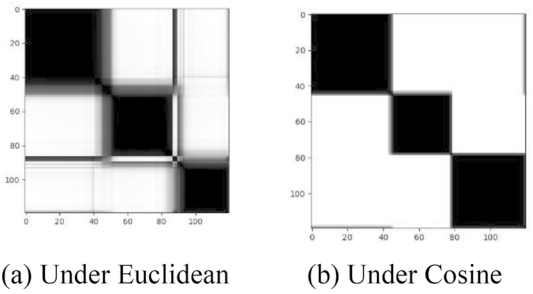

Fig. 5 Assessment of 3- Clusters Gaussian Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance Metrics (sample = 10% of original data). **a** Under Euclidean. **b** Under Cosine

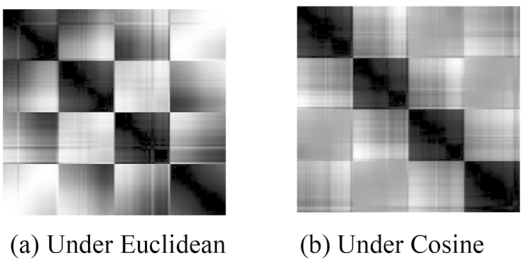

Fig. 6 Assessment of 4- Clusters Corner Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance etrics (sample = 10% of original data). **a** Under Euclidean. **b** Under Cosine

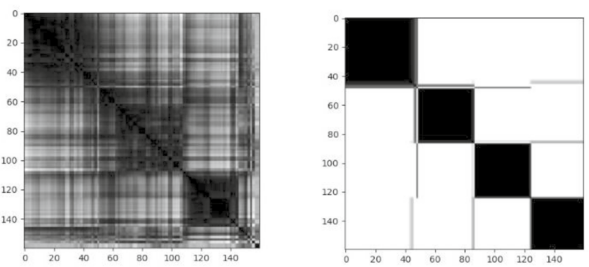

Fig. 7 Assessment of 4- Clusters Outlier Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance Metrics (sample = 10% of original data). **a** Under Euclidean. **b** Under Cosine

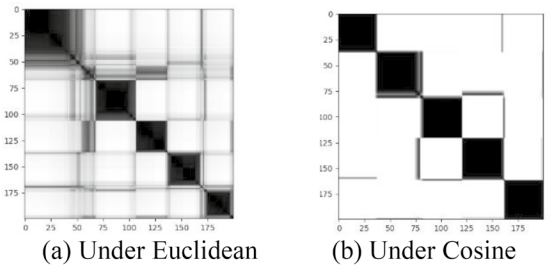

Fig. 8 Assessment of 5- Clusters Gaussian Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance Metrics (sample = 10% of original data). **a** Under Euclidean. **b** Under Cosine

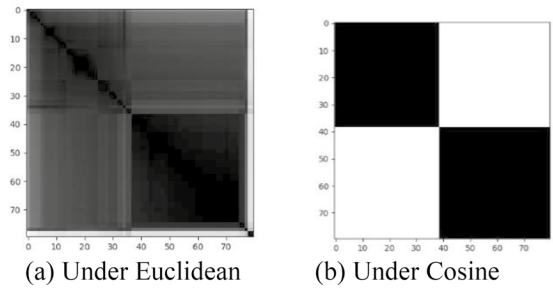

Fig. 9 Assessment of 2- Clusters MiniBooNE Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance Metrics (sample = 10% of original data). **a** Under Euclidean. **b** Under Cosine

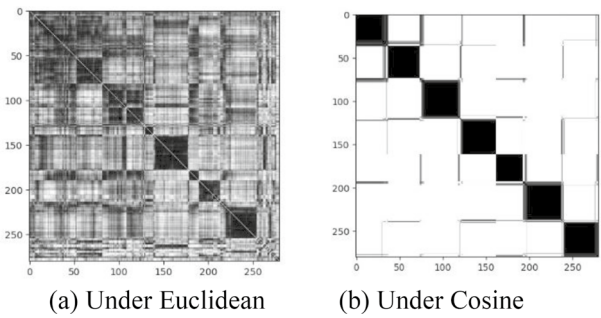

Fig. 10 Assessment of 7- Clusters Forest Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance Metrics (sample = 10% of original data)

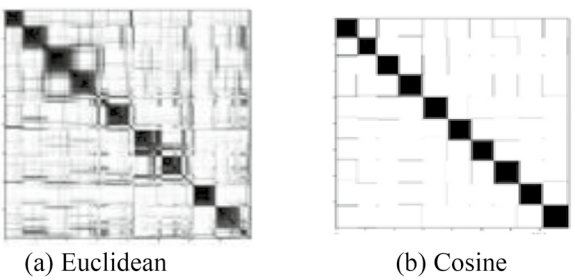

Fig. 11 Assessment of 10- Clusters Real MNIST Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance Metrics (sample = 10% of original data). **a** Euclidean. **b** Cosine

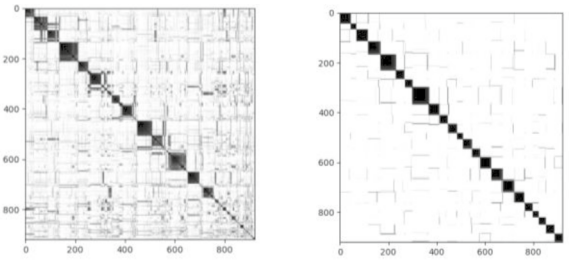

Fig. 12 Assessment of 23-Clusters KDDCUP Data using Sampling-based Crisp -bigVAT (SC-VAT) with Two Distance Metrics (sample = 10% of original data). **a** Euclidean. **b** Cosine

features of data objects. Thus, similarity metrics are showing a significant role in the assessment of cluster tendency. With the visualization results of both synthetic and real datasets, it is observed that the cosine metric poses the best assessment of cluster tendency. Cluster tendency is derived from the count of the number of diagonal squared blocks for the dataset. Cosine based SC-bigVAT justifies the clustering tendency with more clarity of visual image than Euclidean based SC-bigVAT.

Crisp partitions are identified with the finding of the square edginess of square-shaped diagonal blocks. The relevant objects information is retrieved with the finding of k-aligned partitions defined in the earlier sections of the proposed algorithms. Crisp partitions predict the information of cluster labels of data objects. These information are compared with ground truth labels for finding quality data clusters through performance measures such as cluster accuracy (CA) [28], normalized mutual information (NMI) [29], precision, recall, and F-Measure [30]. Tables 2 and 3 present the performance of visualization techniques for empirical observations.

From the performance study of the experimental, the SC-bigVAT is performed as the best when compared to the big data clustering method, faster spherical k-means (SPKM) [31], and CLARA [32]. The SPKM scans the given big data in a single time and it is a speedy version of k-means algorithm. In SPKM, the space of the cluster centers is obtained by the gradient descent approach [33]. The Clustering LARge Applications (CLARA) is designed for handling big datasets, in which representative objects are not derived. However, it uses the Partition Around Medoid (PAM) on the sample and determines the medoid of the sample. The CLARA selects the medoid samples instead of representatives of the data. The comparative scores observed that the SC-bigVAT scores the best cluster accuracy value related to the performance parameters such as NMI, precision, recall, and F-measure. The best scores are highlighted in bold in Tables 2 to 3. The proposed sampling-based crisp partition bigVAT under the cosine performed as the best when compared to CS-bigVAT under Euclidean. Thus, it recommended the big data clustering under cosine space for achieving robust data clusters.

Figures 13, 14, 15, 16, and 17 shows the performance comparison empirically for the two existing methods SPKM, CLARA. These two proposed visualization techniques for big data clustering are represented as bar graphs. Spherical k-means is efficient when compared to CLARA for the data clustering of big datasets.

With this empirical analysis of performance parameters such as CA, and NMI which are shown in Figs. 13 and 14, respectively. It is observed that the SC-bigVAT

Table 2 CA and NMI for the big data clustering methods

Dataset	CLARA	SPKM	SC-biVAT (Euclidean)	SC-bigVAT (Cosine)
Cluster accuracy (CA)				
S-1	0.483	0.499	0.721	**0.769**
S-2	0.675	0.775	0.911	**0.988**
S-3	0.755	0.725	0.922	**0.967**
S-4	0.741	0.842	0.891	**0.945**
S-5	0.754	0.952	0.822	**0.954**
S-6	0.753	**0.777**	0.753	0.680
MiniBooNE	0.483	0.422	0.583	**0.684**
FOREST	0.568	0.567	0.566	**0.600**
MNIST	0.497	0.498	**0.521**	0.511
KDDCUP	0.331	0.432	**0.491**	0.411
Normalized mutual information (NMI)				
S-1	0.607	0.607	0.580	**0.610**
S-2	0.958	0.958	0.916	**0.958**
S-3	0.903	0.903	0.874	**0.906**
S-4	0.859	0.859	0.861	**0.882**
S-5	0.851	0.851	0.906	**0.908**
S-6	0.626	0.626	0.632	**0.642**
MiniBooNE	0.585	0.585	0.590	**0.598**
FOREST	0.564	0.564	0.565	**0.570**
MNIST	0.454	0.454	0.438	**0.471**
KDDCUP	0.514	0.514	0.449	**0.520**

under the Cosine metric is outperformed with others. The precision, recall, and F-measure empirical analysis are shown in Figs. 15, 16, and 17, respectively, and here also the same observation is made that SC-bigVAT is significantly suitable for the big data when compared with traditional spherical k-means and CLARA methods.

4.2 Computational complexity

The existing CLARA and SPKM are widely used for big data clustering. However, their computational times are depending on the selection of sample size, and the number of clusters. For the big data, the sample size 's' also extremely large and these techniques demand high computational complexities i.e. quadratic complexities. The proposed sc-bigVAT uses a few sample viewpoints and it should be less than (s/2). Thus, quadratic complexity requirements of big data clustering are reduced in our proposed sc-bigVAT.

The time and space analysis graphs are shown in Figs. 18 and 19. With this analysis, it can be stated that the proposed SC-bigVAT under cosine and

Table 3 Precision, recall, and F-measure for the big data clustering methods

Dataset	CLARA	SPKM	SC-biVAT (Euclidean)	SC-bigVAT (Cosine)
Precision (P)				
S-1	0.441	0.521	0.855	**0.922**
S-2	0.332	0.442	0.621	**0.931**
S-3	0.212	0.344	0.743	**0.923**
S-4	0.282	0,345	0.822	**0.954**
S-5	0.121	0.521	0.876	**0.966**
S-6	0.433	**0.965**	0.911	0.899
MiniBooNE	0.332	0.776	**0.876**	**0.891**
FOREST	0.452	0.672	**0.723**	**0.762**
MNIST	0.387	0.654	**0.711**	0.687
KDDCUP	0.443	0.766	**0.856**	0.812
Recall (R)				
S-1	0.526	0.521	0.978	**1.000**
S-2	0.41	0.734	0.685	**0.775**
S-3	0.327	0.68	0.977	**0.995**
S-4	0.311	0.497	0.957	**0.968**
S-5	0.264	0.495	0.943	**0.948**
S-6	0.252	0.455	0.978	**1.000**
MiniBooNE	0.201	0.306	0.68	**0.961**
FOREST	0.195	0.257	0.571	**0.687**
MNIST	0.19	0.261	0.497	**0.682**
KDDCUP	0.174	0.228	0.468	**0.611**
F-Measure (F)				
S-1	0.8	0.75	0.5	**0.812**
S-2	0.675	0.45	0.733	**0.742**
S-3	0.594	0.481	0.681	**0.602**
S-4	0.565	0.445	0.495	**0.571**
S-5	0.533	0.479	0.495	**0.541**
S-6	0.454	0.4	0.454	**0.458**
MiniBooNE	0.434	0.388	0.541	**0.451**
FOREST	0.408	0.308	0.419	**0.425**
MNIST	0.463	0.303	0.443	**0.471**
KDDCUP	0.325	0.255	0.386	**0.414**

Euclidean is a faster big data clustering technique when compared to existing SPKM and CLARA techniques. The speed of the proposed methods sc-bigVAT (Euclidean) and sc-bigVAT (cosine) were analyzed relative to SPKM and CLARA which are shown in Fig. 20. This analysis stated that the speed of sc-bigVAT is improved exponentially, hence our proposed work is more suitable for big data clustering problems.

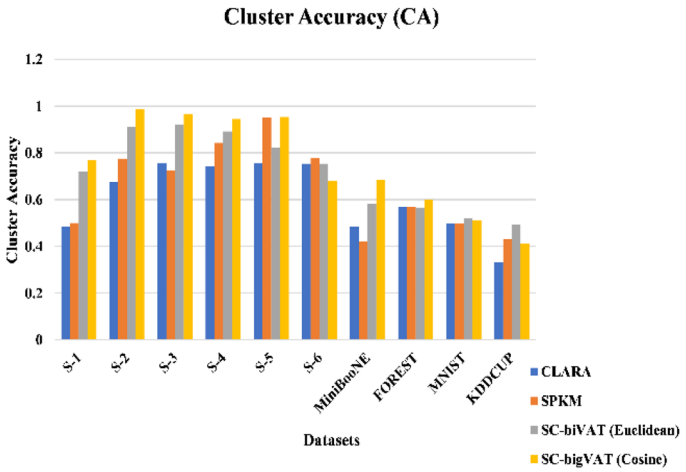

Fig. 13 Cluster accuracy comparison for big data clustering methods (existing and proposed visualization technique)

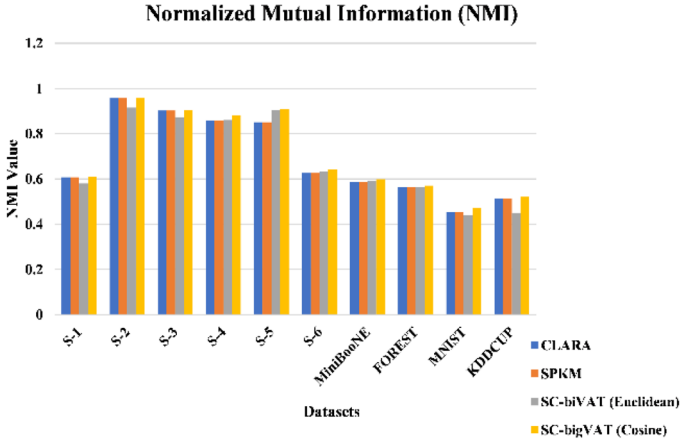

Fig. 14 Normalized mutual information comparison for big data clustering methods (existing and proposed visualization technique)

5 Conclusion and scope of the work

Cluster tendency discovers the prior information about the clusters in data clustering. The existing visualization techniques effectively determine the clustering tendency in the form of visual image clusters. However, it produces the value of cluster tendency for the specific limited size of datasets. The proposed work attempts the big data clustering problem, in which initially cluster tendency of big data is addressed with the sampling technique. The crisp partitions are derived from visual image clusters that

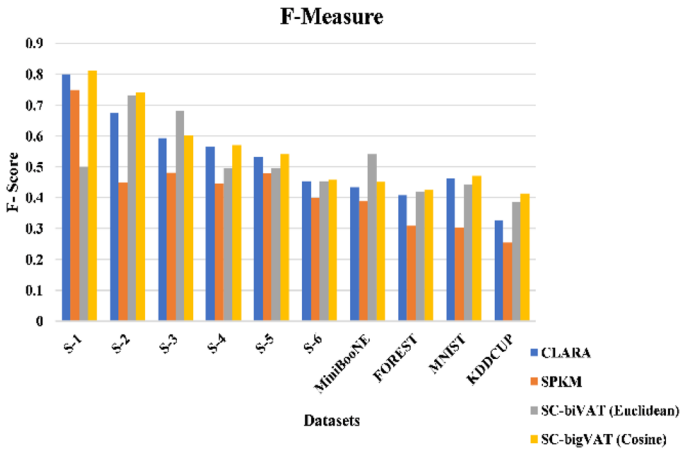

Fig. 15 Precision comparison for big data clustering methods (existing and proposed visualization technique)

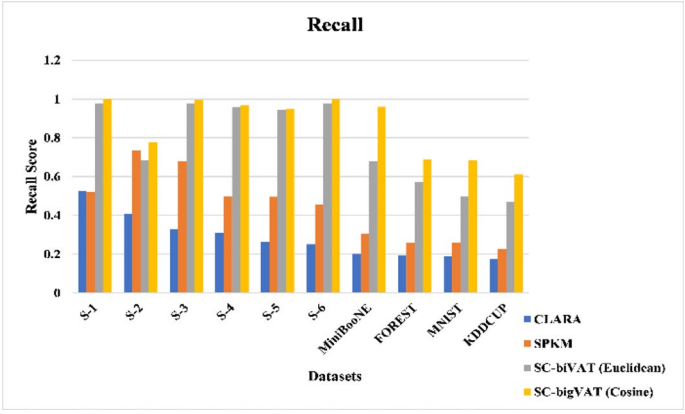

Fig. 16 Recall comparison for big data clustering methods (existing and proposed visualization technique)

accurately predict the cluster labels of data objects in big data clustering. In future work, sampling-based crisp bigVAT is to be enhanced with the subspace learning techniques for handling the scalability issues of high dimensional big data.

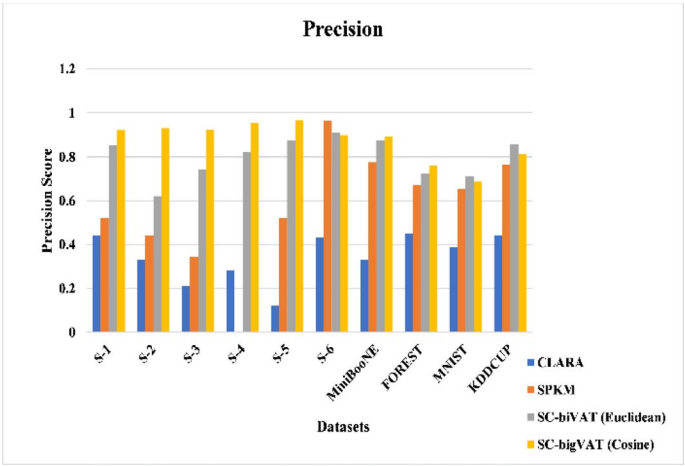

Fig. 17 F-Measure comparison for big data clustering methods (existing and proposed visualization technique)

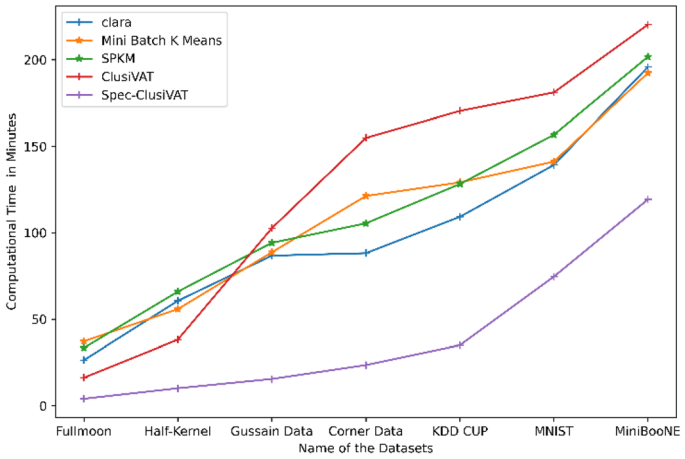

Fig. 18 Time analysis for big data clustering methods

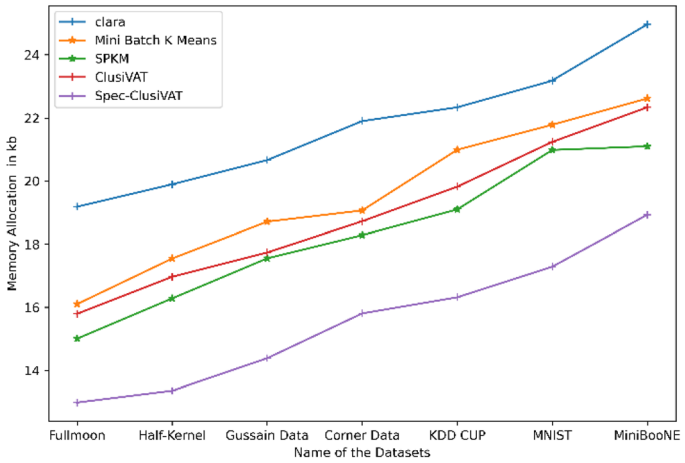

Fig. 19 Memory allocation analysis for big data clustering methods

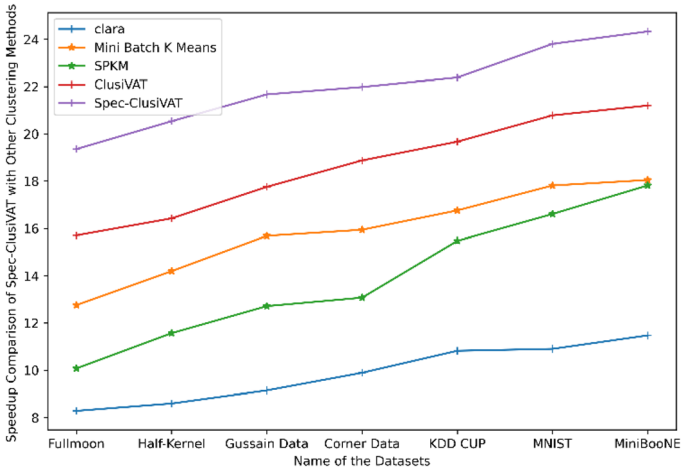

Fig. 20 Speedup analysis for big data clustering methods

Funding This study was funded by Science and Engineering Research Board (Grant No. ECR/2016/001556).

References

1. Bezdek, J.C., Hathaway, R.J.: VAT: a tool for visual assessment of (cluster) tendency. In: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02, pp. 2225–2230 (2002)

2. Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y.: A comparison study on similarity and dissimilarity measures in clustering continuous data. PLoS ONE **10**(12), e0144059 (2015)
3. Singh, S., Singh, N.: Big Data analytics. In: 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, 2012, pp. 1–4, https://doi.org/10.1109/ICCICT.2012.6398180.
4. Suleman Basha, M., Mouleeswaran, S.K., Rajendra Prasad, K.: Cluster tendency methods for visualizing the data partitions. International Journal of Innovative Technology & Exploring Engineering, 2019
5. Esteves, R.M., Hacker, T., Rong, C.: Competitive K-means, a new accurate and distributed K-means algorithm for large datasets. In: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, 2013, pp. 17–24. https://doi.org/10.1109/CloudCom.2013.89.
6. Kumar, D., Bezdek, J.C., Palaniswami, M., Rajasegarar, S., Leckie, C., Havens, T.C.: A hybrid approach to clustering in big data. IEEE Trans Cybern **46**(10), 2372–2385 (2016)
7. Rajendra Prasad, K., Mohammed, M. & Noorullah, R.M. Visual topic models for healthcare data clustering. *Evol. Intel.* (2019). https://doi.org/https://doi.org/10.1007/s12065-019-00300-y
8. Taghva, K., Veni, R.: Effects of similarity metrics on document clustering. In: 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, 2010, pp. 222–226, https://doi.org/10.1109/ITNG.2010.65.
9. Leonori, S., Martino, A., Mascioli, F.M.F., Rizzi, A.: ANFIS microgrid energy management system synthesis by hyperplane clustering supported by neurofuzzy min–max classifier. IEEE Trans. Emerg. Top. Comput. Intell. **3**(3), 193–204 (2019)
10. Rajendra Prasad, K., Mohammed, M., Noorullah, : Hybrid topic cluster models for social Healthcare Data. Int. J. Adv. Comput. Sci. Appl. **10**(11), 490–506 (2019)
11. Rathore, P., Kumar, D., Bezdek, J.C., Rajasegarar, S., Palaniswami, M.: A rapid hybrid clustering algorithm for large volumes of high dimensional data. IEEE Trans Knowledge Data Eng **31**(4), 641–654 (2019). https://doi.org/10.1109/TKDE.2018.2842191
12. Havens, T.C., Bezdek, J.C.: An efficient formulation of the improved visual assessment of cluster tendency (iVAT) algorithm. IEEE Trans Knowl Data Eng **24**(5), 813–822 (2012). https://doi.org/10.1109/TKDE.2011.33
13. Bezdek, J.L.: SpecVAT: Enhanced visual cluster analysis. In: IEEE International Conference on Data Mining, ICDM (2008)
14. Denton, P., Parke, S., Tao, T., Zhang, X.: Eigenvectors from eigenvalues. arXiv. **1908**, 03795 (2019)
15. Huband, J.M., Bezdek, J.C., Hathaway, R.J.: bigVAT: Visual assessment of cluster tendency for large data set. Pattern Recogn. **38**(11), 1875–1886 (2005)
16. Bhatnagar, V., Majhi, R., Jena, P.R.: Comparative performance evaluation of clustering algorithms for grouping manufacturing firms. Arab J Sci Eng **43**, 4071–4083 (2018)
17. Eswara Reddy, B., Rajendra Prasad, K.: Reducing runtime values in minimum spanning tree based clustering by visual access tendency. Int. J. Data Min. Knowl. Manag. Process **2**(3), 11–22 (2012)
18. Lin, Y.S., Jiang, J.Y., Lee, S.J.: A similarity measure for text classification and clustering. IEEE Trans. Knowl. Data Eng. **26**(7), 1575–1590 (2013)
19. Chow, T.W.S., Huang, D.: Data reduction for pattern recognition and data analysis. In: Fulcher, J., Jain, L.C. (eds) Computational Intelligence: A Compendium. Studies in Computational Intelligence, vol 115. Springer, Berlin (2008)
20. Shengxi, P., Jianguo, L., Jiaxiong, P., Wang, G.: The design and implementation of dip arrow plot pattern recognition system. In: [1988 Proceedings] 9th International Conference on Pattern Recognition, vol. 2, Rome, Italy, pp. 703–705. (1988). https://doi.org/10.1109/ICPR.1988.28333.
21. Tariq, A., Foroosh, H.: T-clustering: Image clustering by tensor decomposition. In: 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 4803–4807. https://doi.org/10.1109/ICIP.2015.7351719.
22. Ji, Y., Wang, L., Wu, W., Shao, H., Feng, Y.: A method for LSTM-based trajectory modeling and abnormal trajectory detection. IEEE Access **8**, 104063–104073 (2020). https://doi.org/10.1109/ACCESS.2020.2997967

23. Rajendra Prasad, K., Suleman Basha, M.: Improving the performance of speech clustering method. In: IEEE- 10th International Conference on Intelligent Systems and Control (ISCO) (2016)
24. Mahallati, S., Bezdek, J.C., Kumar, D., Popovic, M.R., Valiante, T.A.: Interpreting cluster structure in waveform data with visual assessment and Dunn's index. InFrontiers in Computational Intelligence 2018 (pp. 73–101). Springer, Cham
25. Rajendra Prasad, K., Suleman Basha, M., Rama Subbaia, B.: Speech clustering analysis by multi viewpoints cosine based similarity. Int. J. Pure Appl. Math. **116**(21), 235–241 (2017)
26. https://archive.ics.uci.edu/ml/index.php
27. https://archive.ics.uci.edu/ml/support/Pen-Based+Recognition+of+Handwritten+Digits
28. Pattanodom, M., I am-On, N., Boongoen, T.: Clustering data with the presence of missing values by ensemble approach. In: 2016 Second Asian Conference on Defense Technology (ACDT). https://doi.org/10.1109/acdt.2016.7437660
29. Alessia, A., Pizzuti, C.: Is normalized mutual information a fair measure for comparing community detection methods? In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (2015).
30. Xu, G., Meng, Y., Chen, Z., Qiu, X., Wang, C., Yao, H.: Research on topic detection and tracking for online news texts. IEEE Access **7**, 58407–58418 (2019)
31. Gulnashin F., Sharma I., Sharma H. (2019) A new deterministic method of initializing spherical K-means for document clustering. In: Pati B., Panigrahi C., Misra S., Pujari A., Bakshi S. (eds) Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 713. Springer, New York
32. Xu, D., Tian, Y.: A comprehensive survey of clustering algorithms. Ann. Data. Sci. **2**, 165–193 (2015)
33. Hitendra Sarma, T., Viswanath, P., Eswara Reddy, B.: Single pass k-means clustering method. Sadhana, Vol. 38, Part. 3, 407–419, Springer (2013)