*I.J. Intelligent Systems and Applications,* **2019, 8, 44-55**
Published Online August 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2019.08.05

# Efficient Intelligent Framework for Selection of Initial Cluster Centers

**Bikram K. Mishra**
Veer Surendra Sai University of Technology, Burla
E-mail: bikrammishra2012@gmail.com

**Amiya K. Rath**
Veer Surendra Sai University of Technology, Burla
E-mail: akrath_cse@vssut.ac.in

**Santosh K. Nanda**
Flytxt Mobile Solutions Pvt. Ltd., 7th Floor, Leela Infopark, Technopark Rd, Technopark Campus, Thiruvananthapuram, Kerala 695581
E-mail: santoshnanda@live.in

**Ritik R. Baidyanath**
Silicon Institute of Technology, Bhubaneswar
E-mail: ranjanritik219@gmail.com

Received: 26 February 2019; Revised: 27 March 2019; Accepted: 11 April 2019; Published: 08 August 2019

***Abstract***—At present majority of research is on cluster analysis which is based on information retrieval from data that portrays the objects and their association among them. When there is a talk on good cluster formation, then selection of an optimal cluster core or center is the necessary criteria. This is because an inefficient center may result in unpredicted outcomes. Hence, a sincere attempt had been made to offer few suggestions for discovering the near optimal cluster centers. We have looked at few versatile approaches of data clustering like K-Means, TLBOC, FEKM, FECA and MCKM which differs in their initial center selection procedure. They have been implemented on diverse data sets and their inter and intra cluster formation efficiency were tested using different validity indices. The clustering accuracy was also conducted using Rand index criteria. All the algorithms computational complexity was analyzed and finally their computation time was also recorded. As expected, mostly FECA and to some extend FEKM and MCKM confers better clustering results as compared to K-Means and TLBOC as the former ones manages to obtain near optimal cluster centers. More specifically, the accuracy percentage of FECA is higher than the other techniques however, it's computational complexity and running time is moderately higher.

***Index Terms***—Optimal Cluster Center, Performance index, Clustering Accuracy, K-Means, Modified Center K-Means and Far Efficient K-Means.

## I. Introduction

During every facet of our day to day necessities it is frequently required to sagaciously manage data into their appropriate groups. This assists us to locate them from their relevant collection more rapidly. Consequently, the most significant thing that is required to be thought about is their accurate placing in their relevant groups. In order to attain a precise group with utmost accuracy, a group containing some data may perhaps be further splitted into minor subgroups. Thus, for tracing out any vital information from a subgroup, quite a few methods have been practiced, one of them is clustering [3, 20]. Clustering is carried out by means of the similarity criteria intended towards maximizing the similarity of objects inside a cluster and minimizing their similarity involving other clusters. Clustering categorizes the objects convincingly and obtains the unknown samples that may be present in the datasets [32]. As a result, the focus of today's research revolve more or less on different clustering approaches with an effort towards – improving the structure and pattern of superior cluster conception [2, 29, 33], curtailing noise from data present in clusters [15, 34, 39], deciding near optimal number of centroids [21, 36] and initiating cluster as an approach in a variety of domains.

Another major issue of clustering relates to making a decision towards the selection of near optimal cluster centers [11, 13, 35, 37] to evade malicious clustering.

This is because, wrongly chosen centers can drag the data elsewhere and effects in bad cluster creation. The results initially expected will nowhere be closer the final consequences obtained. This may lead to the discovery of erroneous patterns form the group and will mislead the entire solution. This motivated us to work out on different approaches for obtaining near optimal centers.

In this paper, we have discussed few versatile approaches for determining the initial cluster centers - Far Efficient K-Means (FEKM) [9], TLBO means of clustering [7], and a suggested method Modified Center K-Means (MCKM). These methods may be a solution to the downsides of traditional K-Means algorithm where the initial centers are decided randomly. They were subsequently analyzed with Far Enhanced Clustering Algorithm (FECA) [6]. Once the desired clusters were formed by employing these techniques, it becomes imperative to determine how fine the groups are formed. For this reason several customary clustering validity indexes were used to test the inter-cluster and intra-cluster separations existing in a dataset [22]. The validity criterion includes Dunn's index (DI), Davies-Bouldin's index (DBI), Silhouette Coefficient (SC), C index and Calinski index (CI). The motive behind opting for a few validity measures is to get explicit cluster formation so that there are least possibilities of any biasness on the value of the selected methods. To obtain more accurate results with precise cluster formation, all the referred methods were tested on numerous datasets taken from UCI repository [10], which differ in their size, pattern, and characteristics.

Another consideration which has been used in this research is to determine the clustering accuracy. This is achieved by means of Rand index [30] which is a measure of the similarity between two data clustering. Subsequently, the time complexity or the computational complexity of each algorithm is analyzed and last but not the least, the actual time required for computing each algorithm is determined. All these parameters for evaluating an algorithm result in eliminating any possible chances of unfairness in selecting the effective clustering technique from the considered ones. As per our anticipation, the efficiency of clustering result by FECA is more acceptable than the other considered methods barring only its slightly additional computation time. But, this can be improved to a large extent by using multi-threading concept in our program.

The offerings of this paper are summarized as follows:

a) We explored on five approaches of data clustering algorithms with a sole intention for achieving near-optimal cluster centers. Thereafter, for each method, well organized cluster formation was obtained.
b) We described a way by which TLBO can be used as an optimization means for obtaining initial centers. The random selection of centers using K-Means was customized and near optimal centroids was obtained using FEKM and suggested MCKM. The FECA method uses FEKM for choosing initial centers but achieves the phase of sub-group formation faster.
c) Our evaluation of clustering effectiveness and accuracy were performed using few widely used indices. The computational complexity and execution time of individual methods were also determined for assessment. Ultimately, the pros and cons of each method were discussed. We have offered various ways by which improved clustering can be achieved on wide range of data sets.

The rest of this paper is organized as follows: Section II presents the resourceful works made by eminent researchers in this related domain. Section III offers the basics of cluster validation parameters and evaluation of accuracy of clustering results. The different clustering methods are subsequently discussed in Section IV. Simulation and results obtained are shown in Section V. At last, Section VI concludes the work giving some future enhancements that can be further carried out.

## II. Related Works

Recurrent studies are conducted by academia, researchers and business society on the initiation of upcoming thoughts employed in improving the quality of clustering methods. A few ideas which are relatable to this work are presented which are also the significant motivating factors behind this research.

The limiting aspects of K-Means were examined in [27] and a different manner by which data can be assigned to separate clusters was suggested. The method presented lessens the execution time of K-Means. "Ref. [25]" reviewed a wide variety of clustering approaches with their applications and also discussed numerous proximity standards and validity criteria that decide the outcome of cluster formation. A customized K-Means which is able to perform precise clustering without assigning the cluster numbers initially was suggested by [31]. The technique can be effectively used to ellipse-shaped data cluster which is the key factor of the research.

In one of the innovative works, [11] suggested the concept of nearest neighbour pair for deciding the initial centers for K-Means algorithm. This method finds two adjacent neighbouring pairs that are mostly dissimilar to each other and located in different clusters. This is one of the several approaches which work towards the innovation of discovering the initial cluster centroids. A further advancement towards determining the near accurate initial centers and then assigning objects to clusters was proposed by [18] but, with a limitation that initial number of cluster has to be given as input. To eliminate the random initial cluster centers selection of K-Means algorithm, [13] proposed a model in which cohesion degree of the neighbourhood of a data and coupling degree among neighbourhoods of data are defined. This model is also accompanied by a new initialization method of choosing the centers.

A modified K-medoids means for clustering can be viewed from the literature presented by [14] for acquiring the initial medoids. A distance matrix is initially computed and used for obtaining new medoids at each step. From the experimentation conducted it was found that this technique gives better performance than classical K-means.

A population-based TLBO algorithm was suggested by [5] to resolve the problem of clustering. When compared with other techniques like SA, PSO, ACO and K-Means, result provide evidences that their method offers optimum value and small standard deviation as compared to the others. When clustering was achieved by means of fuzzy c-means approach, the determination of initial centre plays an important role in its final consequence. TLBO as suggested by [4] presents a solution to this issue. TLBO was initially used to determine the near-optimal cluster centers. Results show TLBO works efficiently in selecting the finest centers for c-means. An AutoTLBO method was suggested by [38] in which K-Means was applied to TLBO to find preferable number of clusters and its effectiveness. Results confirmed the efficiency of clustering for both artificial and standard data sets.

"Ref. [1]" proposed WFA_selection, a modified weight-based firefly selection algorithm for obtaining optimal clusters. This algorithm merges a group of selected clusters to produce clusters of better superiority. Result obtained shows this algorithm producing new compressed clusters as compared to few other approaches. In a recent work, [33] reviewed the classical DBSCAN algorithm, analyzed its limitations and proposed a method to overcome them. The proposed technique finds the maximum density level permitted inside each cluster and this allowed DBSCAN to assess clusters with different densities. When both methods were compared it was confirmed that the proposed one determines the actual cluster effectively.

## III. Clustering Approaches, Validation and Accuracy

Cluster analysis has evolved as a general term for methods which are related to the problem: given a number of data points, the initial approach is how to determine the near optimal solution to obtain the required *K* number of cluster centers. And once the centroids are obtained the next procedure is how to select those data points which are nearest to the chosen centroids and are far away from the rest data points. Most research aims at finding some usual means for discovering the cluster centroids [11, 18, 13], while some intend towards framing and then assessing the sub-groups created [3, 20]. This article solely concentrates towards the development of few methods which suggest some efficient means of detection of initial centroids of a cluster and thereafter offers a means for effectively allowing the data points to form their sub-groups. The sub-groups are formed by minimizing an objective function which decides the "closeness" between the data points and the centroids. The clustering approach can be termed as a triplet *( D, C, m )* where *D* is a set of *N* data points to be grouped, $D = \{ d_1, d_2, ..., d_N \}$, and *C* is the cluster groups formed till *K* numbers denoted by $C = \{ c_1, c_2, ..., c_K \}$, where each cluster is a set of data points $C_k = \{ d_{k1}, d_{k2}, ..., d_{knk} \}$ such that $\sum n_k = N$ and $n_k \geq 1$ for $k = 1, 2, ..., K$. *m* is the method of selecting a particular *C* given the set *D*.

When cluster formation is the issue then, how soundly they are created is the talking point behind it. In one way, their assessment can be done by using the cluster validity indices. Cluster validation [14, 22] is a means for assessing the clustering outcome of an algorithm. The data items with analogous features are placed closer to each other in a sub-group and are far away from those present outside the sub-group. The validity measures determine the "goodness" of a cluster framed. A few 'internal validity indices' that have been used in this work are discussed below.

### A. Dunn's Index (DI)

DI [16] determines the density and well segregated clusters. It is equal to the minimum inter-cluster distance divided by maximum size of cluster.

Let, the cluster size *C* is denoted by $\Delta_c$ and distance between custers *i* and *j* be $\delta(C_i, C_j)$. Then DI is given by:

$$DI = \frac{\min\limits_{1 \leq i \leq j \leq m} \delta(C_i, C_j)}{\max\limits_{1 \leq k \leq m} \Delta_k} \quad (1)$$

Normally, a larger value of DI indicates the cluster is compact and is well-separated from other clusters.

### B. Davies-Bouldin Index (DBI)

DBI [12] is an internal evaluation method which is the sum of ratio of within cluster distribution, to the between cluster separation. DBI is specified as:

$$DBI = \frac{1}{N} \sum_{i=1}^{N} \max_{j \neq i} \left( \frac{S_i + S_j}{M_{i,j}} \right) \quad (2)$$

where, *N* is the number of clusters, $M_{i,j}$ is the separation between the $i^{th}$ and the $j^{th}$ clusters, and $S_i$ is the within cluster distribution for cluster *i*.

A smaller value of DBI implies the better separation of clusters and the 'compactness' within the clusters.

### C. Silhouette Coefficient (SC)

In SC [23], for any data point $d_i$ initially, the average distance from $d_i$ to all other data points belonging to its own cluster is determined, which is *a*. Then, the minimum average distance from $d_i$ to all other data points present in other clusters are determined, which is *b*. Finally, SC for a data point is given by:

$$s = \begin{cases} 1 - a/b & \text{if } a < b \\ 0 & \text{if } a = b \\ b/a - 1 & \text{if } a > b \end{cases} \quad (3)$$

The value of *s* ranges between 0 and 1. When *s* is

closer to 1, it is considered as "well classified".

*D. C Index*

C index was proposed by [19] for determining the intra-cluster separation.

$$\text{It is defined as: } C_{index} = \left( \frac{S - S_{min}}{S_{max} - S_{min}} \right) \quad (4)$$

where, $S$ is the sum of distances of all pairs of data points present within a cluster, $S_{min}$ is the sum of $n$ smallest distance from all data pairs, $S_{max}$ is the sum of $n$ largest distance from all data pairs, and $n$ is the number of those pairs. The C-index is bounded to the interval between [0, 1] and should be closer towards zero for superior result.

*E. Calinski Index (CI)*

CI [28] determines well framed clusters that evaluate 'separation' *between-group sum of squares* (BGSS), and 'closeness' *within-group sum of squares* (WGSS). If $n$ is the number of data points and $k$ is the number of clusters to be formed then, the variance ratio criterion, VRC is given by:

$$VRC = \frac{BGSS}{k-1} \Big/ \frac{WGSS}{n-k} = \left(1 + \frac{n-k}{k-1} a_k\right) \Big/ \left(1 - a_k\right) \quad (5)$$

where, $a_k = A_k / d^{-2}$, which is the weighted mean of the differences between the general and within-group mean squared distances. $a_k$ closer to 1 suggests the creation of well formed cluster.

One method of evaluating the outcome of clustering algorithm is to assess how "well" they form these clusters once they execute. The "goodness" of a method can be said as the number of data points misclassified, and is calculated as a parameter of error percentage [30]. In data clustering approach three assumptions are made: first, each data point is unambiguously allocated to a particular cluster. Second, clusters are formed with those data points which they do not include, as by those data points which they include. Third, every data point has equal significance in the formation of clusters. From these above considerations it is clear that if the objects of a data-pair are placed together in a cluster in each of the two clustering, or if they are assigned to different clusters in both clustering, this indicates a similarity between the clustering.

Given $N$ data points in a set $D=\{D_1, D_2, ..., D_N\}$ and two clustering of them to compare $C=\{C_1, C_2, ..., C_{K1}\}$ and $C'=\{C'_1, C'_2, ..., C'_{K1}\}$, we define Rand index, $R$ as:

$$R = (a + b) / (a + b + c + d) \quad (6)$$

where, $a$ is number of data pairs in $D$ that are in the same subsets of both $C$ and $C'$, $b$ is number of data pairs in $D$ that are in different subsets of both $C$ and $C'$, $c$ is number of data pairs in $D$ that are in same subset of $C$ and different subsets in $C'$, $d$ is number of data pairs in $D$ that are in different subset of $C$ and same subsets in $C'$.

The value for Rand index is between 0 and 1, with 0 representing the two clustering do not agree on any pair of data points and 1 representing the data clustering are precisely identical.

## IV. METHODS

The prime focus following this research is to explore few data clustering techniques. The aim is to work on several aspects for improving the major limitation of traditional K-Means approach where the initial centers were selected randomly. Few suggestions regarding this issue were earlier given by us and in this work we too suggest one more approach for it. The subsequent part of our research relates to an efficient way of creation of sub-groups from a whole lot of data objects, and had earlier succeeded in this aspect.

Now, the intension is to make an assessment of all these methods by considering few parameters for comparison viz, validity indices for verifying the so called "goodness" of clustering outcome, determining the clustering "accuracy" of each methods by using Rand index, analyzing the time complexity of each algorithms, and finally computing the actual time each algorithm takes to converge. Here, the data clustering means that have been used in this work are discussed:

*A. Method – I: K-Means*

This is unsupervised, prototype-based, partitioned clustering [17] technique which locates a user-chosen number of clusters ($K$), which are represented by their cluster centers. Initially, $K$ numbers of initial cluster centers, specified by the user are randomly selected. Each data point is allocated to its nearest center. The data points which are assigned to its closest centers form an independent cluster. The cluster center is then updated by taking the mean of the data present in it. This process of assigning the data to its nearby centers and then updating the cluster centers are repeated until the centers remain intact. Usually, for determining the notion "closest" or "nearest" some proximity measures like Euclidean distances are used.

The formal steps involved in the clustering process are given as follows:-

1. Randomly select $K$ data points as initial cluster centers
2. **repeat**
3. Assign each data point to a cluster that is nearest to any center.
4. Recompute the center of each cluster.
5. **until** centers no longer change.

When the initial centroids are chosen randomly, different runs of K-Means may produce different outcome. Selection of suitable initial centers is the significant step of traditional K-Means procedure because, the wrong centers selected can mislead the cluster formation which is the prime weakness of this algorithm. Similarly, there is a chance for empty clusters to be produced if no data points are allocated to a cluster

during step (3).

The computational time of K-Means algorithm is essentially linear in the number of data present. The time complexity can be *O (i * K * n * a),* where, *i* corresponds to the number of iterations, *n* is the number of data points and *a* is the number of attributes of the data set. *i* value may be small more often as the majority of changes in the cluster formation happens in the initial iterations. K-Means performs efficiently if *K* is appreciably less than *n*.

*B. Method – II: Far Efficient K-Means (FEKM)*

Keeping in view the limitations of K-Means, we came up with FEKM [9], for efficiently selecting the initial cluster centers. The pseudo-code for the technique is given as follows:

*Pseudo-code:*

```
                                                        Complexity
FEKM (data_set,  k):
1.  //Determine two data points with max distance apart      n^2
    for data_i in data_set:
      for data_j in data_set:
        dist [i, j] = Eucl_dist (data_i, data_j)
2.  center[1], center[2] = max (dist [i, j])                 n^2
    //Grouping data nearest to center[1] and center[2]       1
    till threshold
3.  set i = 0
4.  while ( i < (0.5 * (no_data / k) ): // threshold         n
        dist1 = Eucl_dist (center[1], data_set[i])
        dist2 = Eucl_dist (center[2], data_set[i])
        if (dist1 <= dist2):
            add data_set [i] to cluster[1]
            remove data_set [i] from data_set
        else:
            add data_set [i] to cluster[2]
            remove data_set [i] from data_set
        increment i
    //   end of while loop
    // Update centers                                       1
5.  center[1] = mean (cluster[1])
6.  center[2] = mean (cluster[2])                            1
    //Selecting the remaining (K - 2) centers                1
7.  set i = 3
8.  while (i <= k):                                          n
      for each d_ind in range (0, i):
        set j = 0
        for each data in data_set:
          if  d_ind = 0:
            dist = Eucl_dist (center[d_ind], data)
            add dist to min_list[ ]
          else:
            dist = Eucl_dist (center[d_ind], data)
          if (min_list[j] > dist):
            min_list[j] = dist
          increment j
          add max (min_list) to center
      //   end of while loop
    // Performing clustering using K-Means                  n
      till convergence
9.  clusters =  exec_kmean (data_set,  k, center)
```

Step (1) and (2) calculates the Euclidian measure among each pair of data points present in the data set and their furthest pair found is initially treated as two cluster centers. Step (4) assigns the data to their nearest two clusters till a given threshold is reached. Once the data are assigned to their respective clusters, they are removed from the data set. This is a highlighting feature of this algorithm since these data are not considered for their subsequent formation of sub-groups and largely affects the computational efficiency of the algorithm. Step (5) and (6) are used to update the cluster centers. Step (8) which is the core of this algorithm is intended towards attaining the remaining *(K–2)* centers by considering, *max (min (distance ({$d_i$ ,$c_1$ } , { $d_i$ ,$c_2$ })))* condition. For the remaining *K–2* centers, traditional K-Means is performed to assign the left over data points to their respective clusters. Experimentation performed of a variety of data sets shows that, FEKM is an efficient method of determining the initial cluster centers and effectively solves the limitations of selecting random centers as in K-Means.

In step 1, Euclidean distance of each data point from every other data point in the data set is computed making the loop run $n^2$ times. So its complexity is $\Theta(n^2)$. From those $n^2$ distances the maximum is found to determine the farthest points making step 2 complexity $\Theta(n^2)$. Step 5 and 6 are initialization of first two centers so their complexity is *O (1).* Then, remaining *K*– 2 centers are computed in step 8, traversing through entire data set making the loop run (*n * (K*– 2)) times. As *K* is very small so the complexity is *O (n).* At last the traditional K-Mean clustering is executed to cluster data using computed *K* centers, making step 9 complexities *O (n).*

Considering the computational factor, we can say, K-means meets its convergence slightly earlier than FEKM, which is the latter's only weakness.

*C. Method–III:Far Enhanced Clustering Algorithm (FECA)*

FECA has been suggested by [6] with an intension towards enhancing the efficacy of FEKM [9] and ECM [8]. Without considering the limitations of the two methods, their fundamental concepts are considered in designing FECA. FEKM is used to decide the near optimal cluster centers and ECM is used for the subgroup formation. The steps for designing FECA are as follows:

*Pseudo-code:*

```
                                                        Complexity
FECA (data_set, k):
  // Finding initial cluster centers
1.  center = FEKM (data_set, k)                              n^2
2.  Assign each data to its nearby cluster centers.          n
3.  Construct two lists certer_ref=[ ] and dist_ref=[ ]      1
  // Creating center reference                               n
4.  for data in data_set:
    set i = 0
    for c in cluster:
      if data in c:
        add i to center_ref
      increment i
```

```
//Creating distance reference                                   1
   5.  set i=0
6. for data in data_set:                                        n
     add Eucl_dist (center [center_ref[i], data)
     increment i
7. Recalculate cluster centers by taking their mean             n
8. repeat step 7 till convergence:                              n
   set i=0
   for data in data_set:
    dist = Eucl_dist (center[center_ref[i]], data)
    if (dist > dist_ref[i]):
      set dist_list = [ ]
      for d_ind in center:
        add Eucl_dist (d_ind, data) to dist_list
        dist_ref[i] = min(dist_list)
        remove data from its present cluster
        center_ref[i] = indexof (min(dist_list))
        add data to cluster[center_ref[i]]
     increment i
   Recalculate the centroids
```

Initially, the $K$ distinct initial centers are computed using FEKM in step 1 with the computational complexity of $\Theta(n^2)$. All the data points are assigned to their nearest centers, so the loop runs $(n - (0.5 * (n / k)))$ times as $(0.5 * (n / k))$ number of data points are removed from data set in FEKM. So, the complexity of the step 2 is $O(n)$. Step 3 is a static declaration, so its complexity is $O(1)$. In step 4 and step 6 *center_ref* and *distance_ref* are created by traversing through clusters which requires $\Theta(n)$ time to compute. Cluster centers are recomputed by taking their mean in step 7 with complexity $\Theta(n)$. Then the clustering is done where the convergence criterion is check using indexes which makes the computation faster. So the complexity of step 8 is $O(n)$. Therefore, the time complexity of FECA is $\Theta(n^2)$.

### D. Method – IV: TLBO Clustering (TLBOC)

The initial concept of TLBO was proposed by [24]. This population based optimization concept was used in data clustering [7]. The process was carried out using two stages:

1. Using TLBO for attaining the initial cluster centers.
2. Using enhanced clustering approach for performing clustering.

*Phase I: Achieving near optimal centroids using TLBO:*

To achieve the initial cluster centers, TLBO have been used as an optimization means and it is combined with ECM [8] to obtain the sub-groups. The $K$ numbers of data points with minimum quantization error values as suggested by [26] are ideally selected as the initial cluster centers. The appropriate learners, considered as the quantization error is given as:

$$J_e = \frac{\sum_{j=1}^{N_C}\left[\sum_{\forall Z_p \in C_{ij}} d\left(Z_p, m_{ij}\right) / \left|C_{ij}\right|\right]}{N_C} \quad (7)$$

where, $N_c$ is no. of cluster centroid vector, $m_{ij}$ is $j^{th}$ cluster center vector of the $i^{th}$ particle in cluster $C_{ij}$ , $z_p$ is $p^{th}$ data vector and $d(Z_p, m_{ij})$ is distance matrix to all $C_{ij}$ .

*Phase II: Enhanced Clustering Method for Clustering:*

Once the $K$ optimized cluster centers are obtained from Phase I, this phase involves grouping the data points to their respective clusters using ECM. Once TLBO converges, the resulting vector found from the learner group with minimum quantization error is treated as initial cluster centroids. After that, every data is assigned to a centre nearer to it and indexes about its current cluster position along with its distance from its centre are stored in two separate matrices. The centers are again recalculated by taking their mean within a cluster. Within each subgroup the distance between every data point to its new center is computed. If new distance is less than the old distance then the data point continue to stay in that cluster. Or else, distance of that data point with other remaining cluster centers are calculated and is assigned to that cluster whose center is nearer to it. This process is repeated till convergence.

The TLBO used for determining the cluster centers has the time complexity of $O(N * D * Gen_{max})$, where $N$ is the number of population, $D$ is the number of dimensions, and $Gen_{max}$ is the maximum number of generations. And enhanced clustering method used for the process of formation of $K$ number of clusters has the time complexity $O(n)$. Hence, the computational complexity of TLBOC is $O(n)$.

### E. Method – V: Modified Center K-Means (MCKM)

At first, the number of clusters $K$ is assigned by user. The distance from all data points to the last data point is calculated. The computed distance values are stored in ascending order. The dataset is then distributed into $K$ number of subgroups and the last data present in each subgroup forms their respective cluster centers. Then the mean data points are calculated and the new center is formed. This process is repeated till the stopping criterion is reached. The pseudo-code of this method is as follow:

*Pseudo-code:*

```
                                                              Complexity
MCKM (data_set, k):
1. set cent = [ ]                                               1
   //last element of data set is treated initial center         1
2. cent.append (data_set [0])
   //finding distance of all data from the last one             1
3. set i = 0
```

```
   //storing distance of each data from initial center                n
4. for r in data_set:
       dist = Eucl_dist (cent[0] , r)
       dist_list.append (dist)
       increament i
5. dist_list.sort( )          //sorting the data                    n*log(n)
   // dividing the sorted list into k equal halves                   n
    s = split_list (dist_list, k)
   // finding remaining k-1 centers                                  1
6. set i = 1
7. for i < k:                                                        n
       cent.append ( mean(s[i])
   // Performing clustering using k-means till convergence           n
8. clusters= exec_kmean (data_set, k, cent)
```

Step 1, 2, and 3 are static initialization and declaration of variables with a complexity of *O (1).* In step 4, a list storing distance of each data from the first element is computed causing a for loop run exactly '*n*' times. So, it has a complexity of *Θ(n).* The data is sorted as per distance list using *trim sort* making step 5 complexity as *O (n * log (n))*. Sorted data is divided in *k* equal parts for which the whole data set is traversed having a complexity of *Θ(n)* in step 6. Then in step 8, mean of the groups is computed to find the initial centers with complexity *Θ(n).*K-means algorithm is executed with a constrained cluster loop in step 9 for further clustering which executes till *O(n).* Therefore, the complexity of MCKM algorithm is *O(n * log (n)).*

By using of all these above methods it is feasible to obtain the required subgroups of any dataset. Any number of clusters can be framed taking different values of *K*. The computational efficiency of all these methods is discussed. Now, these techniques are further evaluated for determining the clustering accuracy and goodness of the clusters created.

## V. Results and Discussion

We evaluated the results of K-Means, FEKM, TLBOC FECA and MCKM methods of clustering on versatile data sets [10]. The characteristics of the datasets used are given in Table 1.

Table 1. Characteristics of Data Sets Used

| Datasets | No. of Attributes | No. of Classes | Instances present |
|---|---|---|---|
| Iris | 4 | 3 | 150 |
| Wine | 13 | 3 | 178 |
| Seed | 7 | 3 | 210 |
| Balance | 4 | 3 | 625 |
| Mushroom | 22 | 2 | 8124 |
| Abalone | 8 | 3 | 4177 |
| Glass | 11 | 2 | 214 |
| TAE | 5 | 3 | 151 |

The datasets considered here varies in their characteristics, size, attributes, categories of belongingness and number of occurrences. If these diverse datasets are given as input to the clustering algorithms then the efficiency of the methods can be known. Iris dataset consists of three classes of flowers - *setosa, virginica* and *versicolor* and each class has 50 samples. The attributes of the flowers includes length and width of sepals and length and width of petals. Likewise, the dataset of Mushroom contains some hypothetical samples of 22 species belonging to *Agaricus* and *Lepiota* family. Each type is labeled as either edible or poisonous. Similarly, Seed dataset comprises of wheat kernels of three classes - *Kama*, *Rosa* and *Canadian* and each variety has 70 instances each. Balance dataset is for representing psychological experimental results. The attributes present in it are left weight, left distance, right weight, and right distance. All objects remain in three classes - scale tip to right, tip to left, or balanced. TAE contains teaching performance evaluation of 151 teaching assistants. The scores are separated into three categories of low, medium, and high which is the class variables.

The clustering outcomes of all algorithms are evaluated using the cluster validity indices viz, DI, DBI, SC, C index and Calinski. These validity measures assess the inter-cluster and intra-cluster distances among the objects of the subgroups. The number of clusters *K* to be formed is initialized by the user. One more evaluation criteria conducted in this research is judging the clustering accuracy. For this reason, Rand index has been considered which measures the similarity between two data clustering. Consequently, the time complexity of each algorithm is analyzed to assess their computation speed and last but not the least, the actual time required for computing each algorithm is calculated.

From Table 2 it can be seen that, limiting the number of iterations to 20 and initially selecting *K* as 3 by the user, most of the validity indices for FECA method of clustering shows encouraging results for almost all datasets baring only a few. For DI, SC and Calinski Index, which are used for inter-cluster validation, a larger value generally closer to 1 indicates better cluster formation, it can be observed that almost all of their values are larger for FECA when compared with other clustering methods. Similarly, for DBI and C Index which are used for assessing the intra-cluster or within-cluster configuration, a smaller value usually closer to 0 signifies superior clustering, it can be noticed that most entries for FECA for different datasets are lesser than other considered approaches.

Again, to check whether FECA is the better technique and performs quality clustering we changed the stopping criteria and now limited the number of iterations to 10 and same *K* as 3, the result can be seen from Table 3. More or less the same consequences were obtained as achieved when the number of iterations was 20.

The effect of validity indices of all the discussed methods with number of iterations as 20 and $K = 3$ on different datasets is shown in Fig. 1(a) to Fig. 1(e) and those with number of iterations as 10 and $K = 3$ is shown in Fig. 2(a) to Fig. 2(e). Both the figures, Fig. 1 and 2 are the graphical view of the results obtained from Table 2

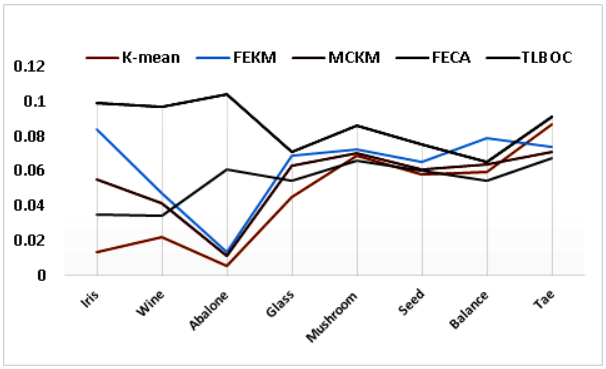
and 3 respectively and show a comparative analysis of all the discussed methods when evaluated with different validity indices for versatile data sets.

Table 2. Assessment of K-MEANS, FEKM, MCKM, FECA and TLBOC by Taking into Consideration DI, DBI, SC, C INDEX and CI When Number of Iterations is 20 and $K = 3$

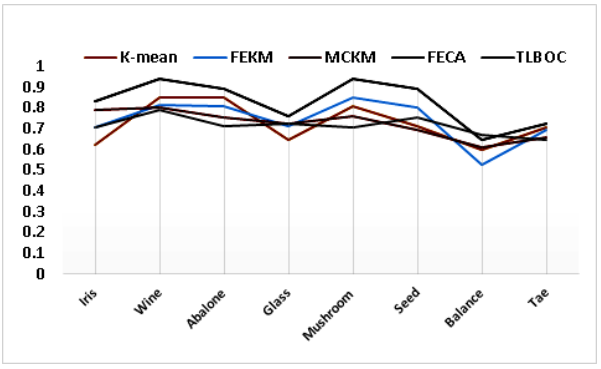
| Algorithms Used → | K-Means | | | | | FEKM | | | | | MCKM | | | | | FECA | | | | | TLBOC | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Datasets ↓ | DI | DBI | SC | C Index | CI | DI | DBI | SC | C Index | CI | DI | DBI | SC | C Index | CI | DI | DBI | SC | C Index | CI | DI | DBI | SC | C Index | CI |
| Iris | 0.013 | 0.612 | 0.621 | 0.422 | 0.613 | 0.084 | 0.066 | 0.703 | 0.402 | 0.801 | 0.055 | 0.076 | 0.789 | 0.508 | 0.752 | 0.099 | 0.051 | 0.831 | 0.313 | 0.822 | 0.035 | 0.076 | 0.706 | 0.512 | 0.713 |
| Wine | 0.022 | 0.554 | 0.846 | 0.565 | 0.644 | 0.047 | 0.066 | 0.814 | 0.521 | 0.843 | 0.041 | 0.073 | 0.801 | 0.569 | 0.835 | 0.097 | 0.068 | 0.94 | 0.44 | 0.974 | 0.034 | 0.073 | 0.791 | 0.508 | 0.803 |
| Abalone | 0.005 | 0.795 | 0.848 | 0.705 | 0.795 | 0.013 | 0.084 | 0.805 | 0.885 | 0.605 | 0.011 | 0.998 | 0.755 | 0.803 | 0.521 | 0.104 | 0.198 | 0.891 | 0.716 | 0.779 | 0.061 | 0.142 | 0.711 | 0.883 | 0.641 |
| Glass | 0.045 | 0.046 | 0.646 | 0.575 | 0.868 | 0.069 | 0.055 | 0.711 | 0.668 | 0.898 | 0.063 | 0.068 | 0.742 | 0.522 | 0.810 | 0.071 | 0.043 | 0.706 | 0.414 | 0.957 | 0.054 | 0.071 | 0.725 | 0.802 | 0.798 |
| Mushroom | 0.069 | 0.132 | 0.804 | 0.894 | 0.742 | 0.072 | 0.086 | 0.848 | 0.763 | 0.711 | 0.07 | 0.094 | 0.756 | 0.712 | 0.702 | 0.086 | 0.09 | 0.937 | 0.671 | 0.786 | 0.066 | 0.106 | 0.704 | 0.763 | 0.712 |
| Seed | 0.058 | 0.342 | 0.712 | 0.622 | 0.776 | 0.065 | 0.068 | 0.799 | 0.511 | 0.913 | 0.061 | 0.077 | 0.691 | 0.691 | 0.753 | 0.075 | 0.051 | 0.889 | 0.542 | 0.889 | 0.06 | 0.069 | 0.753 | 0.704 | 0.804 |
| Balance | 0.059 | 0.283 | 0.599 | 0.559 | 0.601 | 0.079 | 0.04 | 0.522 | 0.574 | 0.616 | 0.064 | 0.052 | 0.606 | 0.671 | 0.515 | 0.065 | 0.043 | 0.646 | 0.468 | 0.699 | 0.054 | 0.059 | 0.666 | 0.692 | 0.612 |
| Tae | 0.087 | 0.176 | 0.707 | 0.391 | 0.841 | 0.074 | 0.041 | 0.693 | 0.312 | 0.684 | 0.071 | 0.155 | 0.655 | 0.356 | 0.679 | 0.091 | 0.036 | 0.722 | 0.302 | 0.847 | 0.067 | 0.082 | 0.642 | 0.416 | 0.726 |

Table 3. Assessment of K-MEANS, FEKM, MCKM, FECA and TLBOC by Taking into Consideration DI, DBI, SC, C INDEX and CI When Number of Iterations is 10 and $K = 3$

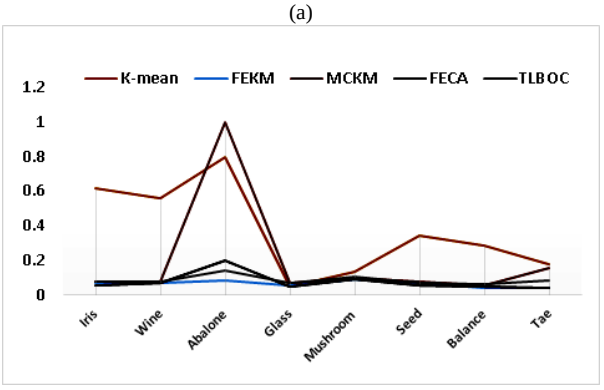
| Algorithms Used → | K-Means | | | | | FEKM | | | | | MCKM | | | | | FECA | | | | | TLBOC | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Datasets ↓ | DI | DBI | SC | C Index | CI | DI | DBI | SC | C Index | CI | DI | DBI | SC | C Index | CI | DI | DBI | SC | C Index | CI | DI | DBI | SC | C Index | CI |
| Iris | 0.023 | 0.612 | 0.621 | 0.42 | 0.562 | 0.084 | 0.066 | 0.703 | 0.402 | 0.801 | 0.055 | 0.076 | 0.789 | 0.513 | 0.752 | 0.099 | 0.051 | 0.831 | 0.313 | 0.822 | 0.035 | 0.076 | 0.706 | 0.518 | 0.713 |
| Wine | 0.019 | 0.561 | 0.842 | 0.564 | 0.623 | 0.042 | 0.073 | 0.81 | 0.523 | 0.84 | 0.038 | 0.066 | 0.801 | 0.561 | 0.831 | 0.095 | 0.071 | 0.937 | 0.46 | 0.974 | 0.034 | 0.078 | 0.786 | 0.51 | 0.798 |
| Abalone | 0.008 | 0.796 | 0.845 | 0.721 | 0.745 | 0.011 | 0.084 | 0.8 | 0.884 | 0.6 | 0.012 | 0.998 | 0.755 | 0.81 | 0.516 | 0.104 | 0.2 | 0.888 | 0.719 | 0.634 | 0.058 | 0.146 | 0.711 | 0.886 | 0.638 |
| Glass | 0.041 | 0.046 | 0.641 | 0.578 | 0.851 | 0.064 | 0.056 | 0.708 | 0.67 | 0.895 | 0.06 | 0.068 | 0.76 | 0.524 | 0.802 | 0.071 | 0.043 | 0.751 | 0.528 | 0.957 | 0.053 | 0.076 | 0.722 | 0.805 | 0.791 |
| Mushroom | 0.07 | 0.086 | 0.8 | 0.89 | 0.681 | 0.07 | 0.117 | 0.846 | 0.76 | 0.7 | 0.068 | 0.094 | 0.751 | 0.716 | 0.706 | 0.082 | 0.09 | 0.84 | 0.673 | 0.701 | 0.066 | 0.106 | 0.7 | 0.766 | 0.691 |
| Seed | 0.052 | 0.341 | 0.71 | 0.622 | 0.712 | 0.061 | 0.068 | 0.796 | 0.52 | 0.913 | 0.058 | 0.077 | 0.69 | 0.693 | 0.746 | 0.073 | 0.051 | 0.882 | 0.548 | 0.885 | 0.06 | 0.069 | 0.75 | 0.71 | 0.804 |
| Balance | 0.055 | 0.283 | 0.595 | 0.556 | 0.593 | 0.074 | 0.046 | 0.52 | 0.581 | 0.606 | 0.064 | 0.052 | 0.602 | 0.674 | 0.511 | 0.062 | 0.05 | 0.641 | 0.472 | 0.601 | 0.052 | 0.063 | 0.66 | 0.694 | 0.602 |
| Tae | 0.081 | 0.175 | 0.7 | 0.4 | 0.726 | 0.069 | 0.046 | 0.693 | 0.322 | 0.68 | 0.07 | 0.155 | 0.655 | 0.36 | 0.672 | 0.089 | 0.036 | 0.653 | 0.328 | 0.847 | 0.063 | 0.086 | 0.64 | 0.42 | 0.73 |



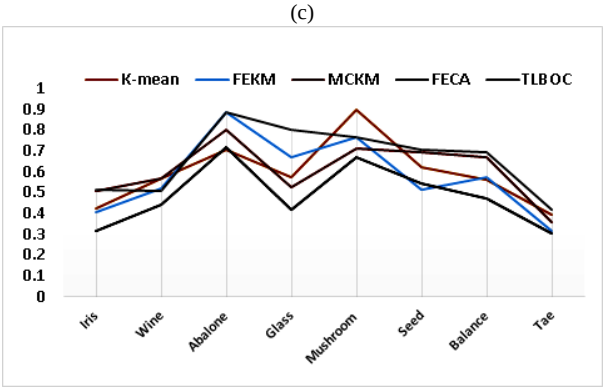

(a)



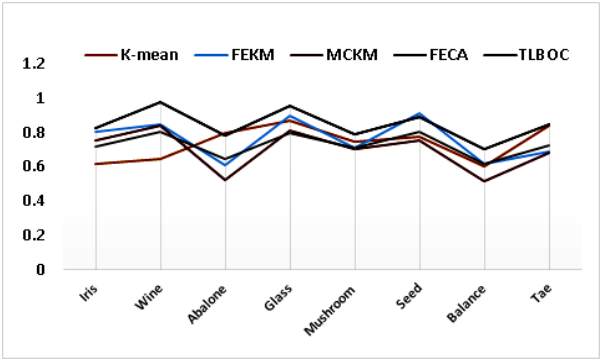

(c)




(b)

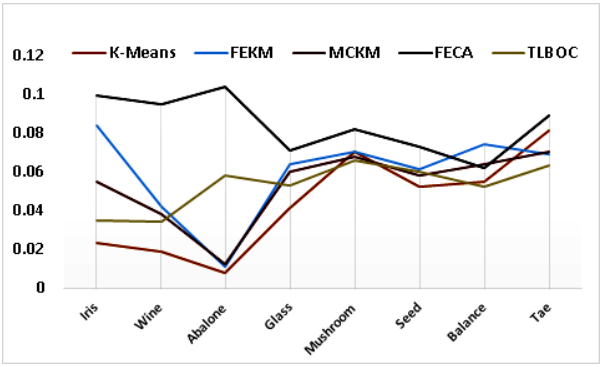



(d)

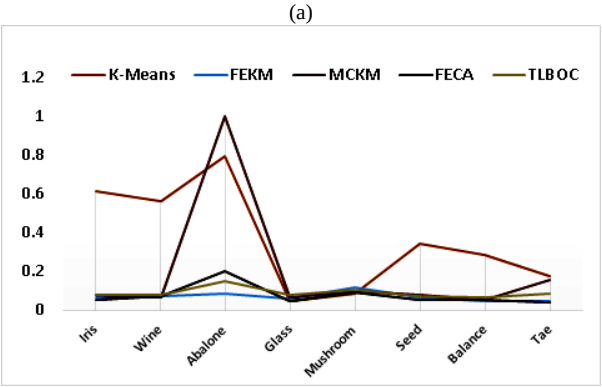



(e)

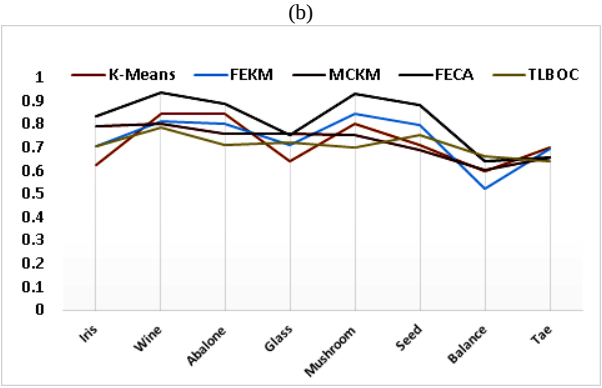
Fig.1. Performance Based on (a) DI values (b) DBI values (c) SC values (d) C Index values (e) Calinski Index values (Number of iterations = 20 and $K$ = 3



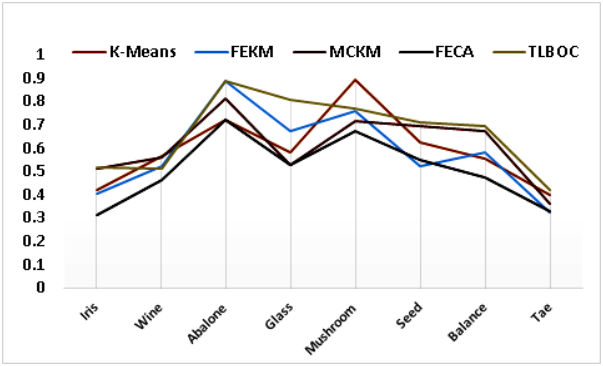

(a)



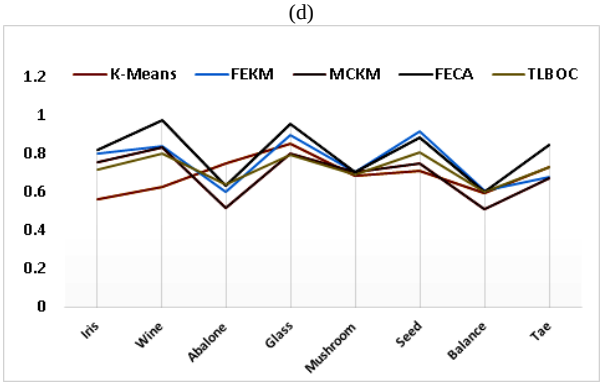

(b)




(c)




(d)




(e)

Fig.2. Performance Based on (a) DI values (b) DBI values (c) SC values (d) C Index values (e) Calinski Index values (Number of iterations = 10 and $K$ = 3)

Experiments were also conducted with different iterations and number of clusters $K$, but only a few results is shown in this paper. The next parameter for evaluation of all discussed algorithm relates to the accuracy with which the clusters were created. This was evaluated by considering the well known Rand index criteria. As discussed earlier, the values obtained for Rand index normally stay within 0 and 1. Any value closer to 0 indicates that the two clustering do not agree on any pair of data points and, any value closer to 1 represents the data clustering are precisely identical. Table 4 illustrates the accuracy of subgroups formed by the help of different algorithms considered by limiting the number of iterations to 20 with $K$=3. It can be noticed that, a majority values of Rand index for FECA are nearer to 1 for most of the datasets than the other methods, only a few being the exclusions. Fig. 3 clarifies this fact. Similarly, Rand index was also computed for all techniques by limiting the number of iterations to 10 with $K$=3. Similar kinds of outcome obtained with the majority values of FECA are closer to 1 than the others. This suggests that the accuracy percentage is more for FECA. Table 5 illustrates this fact and Fig. 4 shows the Rand index analysis of all the given methods.

Table 4. Clustering Accuracy of K-MEANS, FEKM, TLBOC, MCKM and FECA by Considering RAND INDEX (Iterations = 20)

| Datasets ↓ | | K-Means | FEKM | TLBOC | MCKM | FECA |
|---|---|---|---|---|---|---|
| Iris | (*K*=3) | 0.7153 | 0.8859 | 0.8693 | 0.8878 | 0.8985 |
| Wine | (*K*=3) | 0.7186 | 0.7286 | 0.6994 | 0.7186 | 0.7640 |
| Abalone | (*K*=3) | 0.5554 | 0.6235 | 0.6139 | 0.6096 | 0.6835 |
| Glass | (*K*=2) | 0.7372 | 0.7749 | 0.7016 | 0.7475 | 0.7552 |
| Mushroom | (*K*=2) | 0.6002 | 0.6601 | 0.6367 | 0.6648 | 0.6459 |
| Seed | (*K*=3) | 0.6903 | 0.86675 | 0.7259 | 0.7581 | 0.8856 |
| Balance | (*K*=3) | 0.5906 | 0.6495 | 0.5892 | 0.6268 | 0.6043 |
| TAE | (*K*=3) | 0.5432 | 0.6436 | 0.6016 | 0.6102 | 0.6939 |

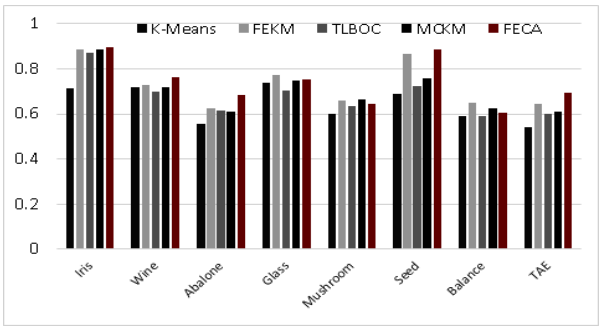



Fig.3. Clustering accuracy using Rand Index (Iteration =20, *K*=3)

Table 5. Clustering Accuracy of K-MEANS, FEKM, TLBOC, MCKM and FECA by Considering RAND Index (Iterations = 10)

| Datasets ↓ | | K-Means | FEKM | TLBOC | MCKM | FECA |
|---|---|---|---|---|---|---|
| Iris | (*K*=3) | 0.7654 | 0.8901 | 0.8611 | 0.8706 | 0.8991 |
| Wine | (*K*=3) | 0.6548 | 0.7238 | 0.6123 | 0.6912 | 0.7223 |
| Abalone | (*K*=3) | 0.6345 | 0.6824 | 0.5612 | 0.6123 | 0.7145 |
| Glass | (*K*=2) | 0.5213 | 0.6941 | 0.5671 | 0.5912 | 0.5992 |
| Mushroom | (*K*=2) | 0.5421 | 0.6156 | 0.6412 | 0.6831 | 0.6112 |
| Seed | (*K*=3) | 0.5903 | 0.87001 | 0.7413 | 0.8453 | 0.8812 |
| Balance | (*K*=3) | 0.6004 | 0.6042 | 0.5123 | 0.5547 | 0.6374 |
| TAE | (*K*=3) | 0.5999 | 0.5974 | 0.6012 | 0.5124 | 0.6144 |

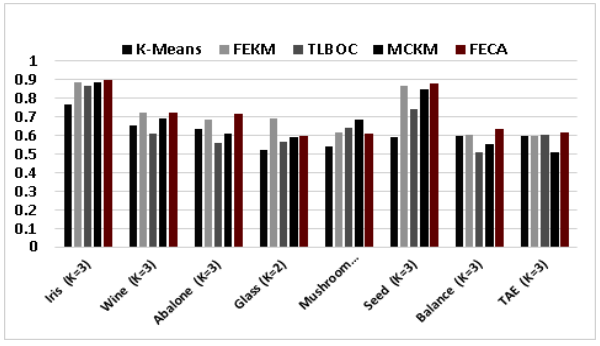



Fig.4. Clustering Accuracy Using RAND Index (Iteration =20, *K*=3)

The actual running time (in sec.) of all the clustering algorithms were calculated. A maximum number of iterations were fixed for each algorithm. However, it was observed that some methods converge before the number of iterations fixed for them. From the execution point of view, K-Means take the minimum time to converse since not much time is spent on its initial center selection which is done randomly. TLBOC and MCKM running time is quite nearer to K-Means but, FECA and FEKM take a little more time to meet their convergence than K-Means since most of their computation time is spent while determining the near optimal centers. Nevertheless, once the centers are obtained the actual formation of subgroups is created faster. All algorithms were implemented in a system with 5th Gen Intel® core i3 Processor, with 1.90 Ghz. frequency and 4 GB RAM. The execution time of different algorithms on a variety of datasets keeping the constraints on the number of iterations to 20 and $K = 3$ is shown in Table 6. Its corresponding analysis is shown in Fig. 5. The execution time was also computed for different number of iterations which are not shown in this paper.

Table 6. Running Time (in Sec.) of K-MEANS, FEKM, TLBOC, MCKM and FECA When Number of Iterations is 20 and $K = 3$

| | K-Means | FEKM | MCKM | FECA | TLBOC |
|---|---|---|---|---|---|
| Iris | 0.024 | 0.091 | 0.043 | 0.075 | 0.036 |
| Wine | 0.031 | 0.124 | 0.062 | 0.088 | 0.049 |
| Glass | 0.037 | 0.071 | 0.059 | 0.076 | 0.051 |
| Abalone | 0.982 | 1.986 | 1.427 | 1.806 | 1.294 |
| Mushroom | 1.643 | 2.509 | 1.955 | 2.414 | 1.703 |
| Seed | 0.035 | 1.221 | 0.071 | 1.005 | 0.056 |
| Balance | 0.108 | 0.935 | 0.311 | 0.821 | 0.282 |
| TAE | 0.019 | 0.296 | 0.041 | 0.254 | 0.021 |

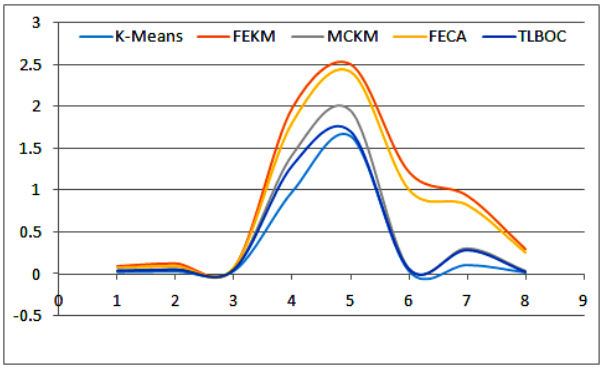



Fig.5. Running Time (in Sec.) of All Algorithms Considered (Iteration =20, *K*=3)

## VI. Observation and Conclusion

The assortment of initial centre selection plays a decisive role in the final formation of useful clusters as erroneous centroids may results in malicious clustering. In this research, an effort has been made to possibly present few approaches to decide the near optimal cluster centers. Quite a few suggested approaches viz, K-Means, FEKM, TLBOC, FECA and MCKM were analyzed considering their clustering effectiveness, correctness, complexity and actual execution time in seconds.

While analyzing all these parameters it was observed that, FEKM, TLBOC and MCKM provides different

approaches of discovering the initial centroids unlike the random method of selection practiced by K-Means. Even if the running time of MCKM is slightly more than that of traditional K-means and TLBOC however, its clustering accuracy is comparatively better and also presents an effective means of selection of initial cluster centers. Similarly, FEKM is another novel approach for choosing the initial centroids that overcomes the major downside of K-Means.

After discovering few innovative ways of obtaining the initial cluster centroids, the focus was on creation of efficient sub-groups. Thus, these methods were evaluated using various cluster validity indices like DI, DBI, Silhouette coefficient, C index and Caliliski index. Precisely, it was noticed that FECA is the better technique and performs quality clustering with different number of iterations and clusters. With Rand index parameter especially preferred for determining the accuracy of clusters created, it was noted that the percentage of accuracy for FECA is more as compared to the other methods for most of the datasets with different number of iterations. The only down side of FECA is its computation time which is to some extent more than K-Means, MCKM and TLBOC. This is because most of its execution time is spent on determining the near optimal initial centers but once it is done, it uses two reference lists to store the cluster index and distance index for performing the actual clustering which is completed in less time. That is the reason why FECA converges much earlier than FEKM when both use the same method for determining the initial centroids. However, the execution time of FECA can be reduced substantially by using multithreading concepts in the program.

We have further thought of enhancing this work and apply the multithreading approach as mentioned to reduce its execution time of FECA. Once this is achieved this technique can be functional in several domains especially in digital image processing for discovering the significant regions of interest. It can be used in the agricultural sphere to enhance our farming in a smart manner.

## ACKNOWLEDGMENT

We are extremely thankful to Sagarika Swain who provided expertise that greatly assisted the work. The authors also express gratitude to the editors and the anonymous referees for any productive suggestions on the research.

## REFERENCES

[1] A. J. Mohammed, Yusof, Y. and Husni, H. "Discovering optimal clusters using firefly algorithm", *Int. J. of Data Mining, Modelling and Management*, vol. 8, no. 4, pp.330–347, 2016.

[2] A. K. Jain, M. N. Murty and P. J. Flynn. "Data Clustering: A Review", *ACM Computing Surveys*, vol. 31, no. 3, pp 264–323, 1999.

[3] A. K. Jain, A. Topchy, M. H. C. Law and J. M. Buhmann. "Landscape of clustering algorithms", *In Proc. IAPR International conference on pattern recognition, Cambridge, UK*, pp. 260–263, 2004.

[4] A. Naik. S. C. Satpathy and K. Parvathi. "Improvement of initial cluster centre of c-means using teaching learning based optimization", *2nd Int. Conf. on Communication, Computing & Security*, pp. 428 – 435, 2012.

[5] B. Amiri. "Application of Teaching-Learning-Based Optimization Algorithm on Cluster Analysis", *Journal of Basic and Applied Scientific Research*, 2(11), pp. 11795–11802, 2012.

[6] B. K. Mishra and A. K. Rath. "Improving the Efficacy of Clustering by Using Far Enhanced Clustering Algorithm", *Int. J. Data Mining, Modeling and Management*, vol. 10, issue 3, pp. 269–292, 2018.

[7] B. K. Mishra, N. R. Nayak and A. K. Rath. "Assessment of basic clustering techniques using teaching-learning-based optimization", *Int. J. Knowledge Engineering and Soft Data Paradigms*, vol. 5, no. 2, pp. 106–122, 2016.

[8] B. K. Mishra, N. R. Nayak, A. K. Rath and S. Swain. "Improving the Efficiency of Clustering by Using an Enhanced Clustering Methodology", *Int. J. of Advances in Engineering & Technology*, vol. 4, issue 2, pp. 415–424, 2012.

[9] B. K. Mishra, N. R. Nayak, A. K. Rath and S. Swain. "Far Efficient K-Means Clustering Algorithm", *Proceedings of the International Conference on Advances in Computing, Communications and Informatics, ACM*, pp. 106–110, 2012.

[10] C. Merz and P. Murphy. UCI Repository of Machine Learning Databases, Available: fttp://ftp.ics.uci.edu/pub/machine-learning-databases.

[11] C. S. Li. "Cluster Center Initialization Method for K-means Algorithm Over Data Sets with Two Clusters", *Int. Conference on Advances in Engineering, Elsevier*, vol. 24, pp. 324–328, 2011.

[12] D. L. Davies and D. W. Bouldin. "A Cluster Separation Measure", *IEEE Trans Pattern Analysis & Machine Intelligence*, vol. 1, pp. 224–227, 1979.

[13] F. Cao, J. Liang and G. Jiang. "An initialization method for the K-Means algorithm using neighbourhood model", *Computers and Mathematics with Applications*, pp. 474–483, 2009.

[14] H. S. Park and C.H. Jun. "A simple and fast algorithm for K-medoids clustering", *Expert System with Applications*, pp. 3336–3341, 2009.

[15] H. Xiong, G. Pandey, M. Steinbach and V. Kumar. "Enhancing Data Analysis with Noise Removal", *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, issue 3, pp. 304–319, 2006.

[16] J. C. Dunn. "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *J. Cybernetics*, vol. 3, pp. 32–57, 1973.

[17] J. Mac Queen. "Some methods for classification and analysis of multivariate observations", *Fifth Berkeley Symposium on Mathematics, Statistics and Probability*, University of California Press, pp. 281–297, 1967.

[18] K. A. Nazeer and M. P. Sebastian. "Improving the Accuracy and Efficiency of the k-means Clustering Algorithm", *Proceedings of the World Congress on Engineering*, vol. 1, 2009.

[19] L. J. Hubert and J. R. Levin. "A general statistical framework for accessing categorical clustering in free recall", *Psychological Bulletin 83*, pp.1072–1080, 1976.

[20] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[21] M. Erisoglu, N Calis and S Sakallioglu. "A new algorithm for initial cluster centers in k-means algorithm", *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1701–1705, 2011.

[22] M. Halkidi, Y. Batistakis and M. Vazirgiannis.

"Clustering validity checking methods: Part ii ", *SIGMOD*, record 31 (3), pp. 19–27, 2002.

[23] P. J. Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", *J. of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[24] R. V. Rao, V. J. Savsani and D.P. Vakharia. "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems", *Computer-Aided Design* 43, pp. 303–315, 2011.

[25] R. Xu and D. Wunsch. "Survey of Clustering Algorithms", *IEEE Transactions on Neural networks*, vol. 16, no. 3, pp. 645–678, 2005.

[26] S. C. Satpathy and A. Naik. "Data Clustering Based on Teaching-Learning-Based Optimization", *SEMCCO, LNCS* 7077, pp. 148–156, 2011.

[27] S. Na, L. Xumin and G. Yong. "Research on K-Means clustering algorithm – An Improved K-Means Clustering Algorithm", *IEEE 3rd Int. Symposium on Intelligent Info. Technology and Security Informatics*, pp. 63–67, 2010.

[28] T. Caliliski and Harabasz, J. "A dendrite method for cluster analysis", *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.

[29] V. E. Castro. "Why so many clustering algorithms — A Position Paper", *SIGKDD Explorations*, vol. 4, issue 1, pp 65–75, 2002.

[30] W. M. Rand. "Objective Criteria for the Evaluation of Clustering Methods". *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846 – 850, 1971.

[31] Y. M. Cheung. "A New Generalized K-Means Clustering Algorithm", *Pattern Recognition Letters*, vol. 24, issue15, pp. 2883–2893, 2003.

[32] Ahmad and S. Khan. "Survey of State-of-the-Art Mixed Data Clustering Algorithms", *IEEE Access*, vol. 7, pp. 31883–31902, 2019.

[33] A. Fahim. "Homogeneous Densities Clustering Algorithm", *I. J. Information Technology and Computer Science*, vol. 10, no. 10, pp. 1–10, 2018.

[34] A. Nag and S Karforma, "An Efficient Clustering Algorithm for Spatial Datasets with Noise", *Int. J. of Modern Education and Computer Science*, vol.10, no.7, pp.29–36, 2018.

[35] V. Kumar, J. K. Chhabra and D. Kumar. "Data Clustering Using Differential Search Algorithm", *Pertanika J. Sci. & Technol.*, vol. 24(2), pp. 295–306, 2016.

[36] X. Yao, S. Ge, H. Kong and H. Ning. "An improved Clustering Algorithm and its Application in WeChat Sports Users Analysis", *Procedia Computer Science, Elsevier*, vol. 129, pp. 166–174, 2018.

[37] D. Jian and G. Xinyue. "Bisecting K-means algorithm based on K-valued self-determining and clustering center optimization", *Journal of Computers*, vol. 13, no. 6, pp. 588–596, 2018.

[38] R. R. Kurada and K. P. Kanadam. "A Novel Evolutionary Automatic Data Clustering Algorithm using Teaching-Learning-Based Optimization", *Int. J. of Intelligent Systems and Applications*, vol. 10, no. 5, pp. 61–70, 2018.

[39] G. Gan and M. K-P. Ng. "K-means clustering with outlier removal", *Pattern Recognition Letters*, vol. 90, pp. 8–14, 2017.

## Authors' Profiles

**Bikram Keshari Mishra** is currently working as a Professor in the Department of Computer Science and Engineering at VSSUT, Burla. He presently has active involvements with novel works involving data clustering and applications. His research interests focus on data mining, knowledge discovery and image processing.

**Amiya Kumar Rath** is a Professor in the Department of Computer Science at VSSUT, Burla, India. Presently he is deputed to National Assessment and Accreditation Council (NAAC), Bangalore as adviser. He has contributed more than 70 research level papers to many national and international journals and conferences. His research interests include embedded system, ad-hoc network, evolutionary computation and data mining.

**Santosh Kumar Nanda** is working as Asst. General Manger in Analytics Center of Excellence, (R & D), FLYTXT Mobile Solution Pvt. Ltd., Trivandrum, India. He completed his PhD from National Institute of Technology, Rourkela. His research interests are Computational Intelligence, Artificial Intelligence, Image Processing Prediction Methodologies, Statistics and Data Science, Mathematical modeling, Pattern Recognition. He has more than 60 research articles in reputed International Journals and International conferences etc. He is now Editor-in-Chief of Journal of Artificial Intelligence, Associate Editor in International Journal of Intelligent System and Application. He is the member of World Federation Soft Computing, USA.

**Ritik Ranjan Baidyanath** is a research scholar in the Department of Information Technology at Silicon Institute of Technology. His research interest includes AI, Image Processing and Artificial Intelligence.

**How to cite this paper:** Bikram K. Mishra, Amiya K. Rath, Santosh K. Nanda, Ritik R. Baidyanath, "Efficient Intelligent Framework for Selection of Initial Cluster Centers", International Journal of Intelligent Systems and Applications(IJISA), Vol.11, No.8, pp.44-55, 2019. DOI: 10.5815/ijisa.2019.08.05