An Optimal Approach for Parallel and Dynamic Clustering

Sandeep Deepak, Amit Kumar Mishra
Student, Assistant Prof.
Department of Computer Science, SBCET, Jaipur Rajasthan, India

Abstract - **Clustering is an approach that classifies the raw data logically and searches the hidden patterns that may be present in datasets. It is procedure of collecting data items into disjointed clusters so that the data items in the same cluster are similar. K-means algorithm is effective in producing clusters for many real time practical applications. In this Work a Dynamic parallel Clustering algorithm based on K-Means algorithm is proposed that overcome the problem of fixed number of input clusters and their serial execution. K-Means is the widely used algorithm for clustering with fixed number of input clusters which is impractical in real scenario. K-Means also perform the computations for clustering between data objects in a serial execution manner. These limitation of K-Means algorithm becomes motivation to develop a new approach that overcome the limitations of K-Means algorithm. This research presents an algorithm for dynamic clustering based on K-Means algorithm which is capable to calculate the number of clusters dynamically and further executes the algorithm in parallel using the capabilities of Microsoft's Task Parallel Libraries. The original K-Means and Proposed Dynamic algorithm based on K-Means are performed for the two dimensional raw data consisting different numbers of records. From the results it is clear that the proposed algorithm is better in all the scenarios either increase the numbers of clusters or change the number of records in raw data. For the same number of input clusters and different data sets in original K-Means and Proposed algorithm, the performance of Proposed Dynamic algorithm is 18 to 46 percent better than the original K-Means in terms of Execution time and Speedup.**

Index Terms - **K-Means, Parallel K-Means, Task Parallel Libraries, Dunn's index.**

I. INTRODUCTION

Clustering is a method of unsupervised classification, where data points are grouped into cluster based on their similarity [1] .Clustering can define as a method of partition some given set of objects in the form of the disjoint clusters. Clustering is used to create the classes of similar data objects by grouping these data objects in such a way that cluster while dissimilar objects are in separate cluster [2]. K-Means clustering is widely used algorithm for clustering of data objects which is used in several areas which includes information retrieval, and pattern recognition & computer vision. In standard K-Means algorithm have to give the value of k in the number of cluster in advance. In practical scenario it is not possible to identify the clusters in advance. The chances of empty clusters increases if the numbers of clusters are fix in advance. K-Means algorithm spends more execution time in computing the distance between data objects and cluster canters.

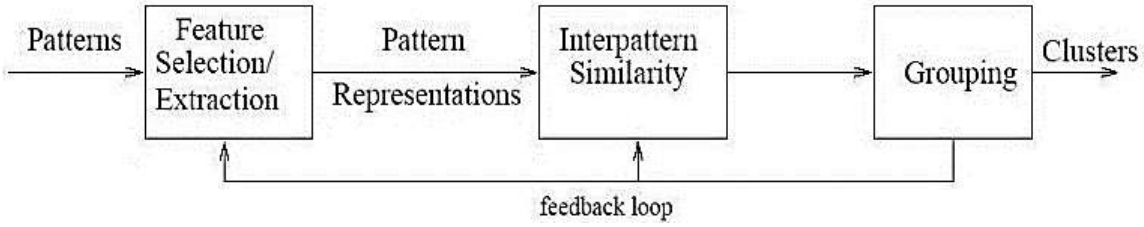

Figure 1: Clustering procedure Steps.

Relational databases are not able to scale at very large scale as compare to NoSQL databases.

1.1 Requirements of Clustering in Data Mining

The given facts describes why clustering is much needed in data mining:

a) ***Scalability:*** it needs most effective and scalable algorithms for clustering that can deal with huge databases.

b) ***Capability to handle different Attributes:*** Algorithms must have consists the ability to be applied on any type of data set such as interval-based (numerical) data, binary and categorical data.

c) ***Cluster discovery based on attribute shape:*** The clustering algorithm has the ability of identifying the clusters with random shapes. Algorithms must not be bounded to single distance measures that generally find the circular clusters made up of small sizes.

d) ***High dimensionality:*** The clustering algorithm must be capable to handle low-dimensional data as well as the high dimensional space.

e) ***Capability to handle noisy data:*** Databases generally contain noisy, modified and erroneous data. Very few algorithms are sensitive to these kinds of data so they may lead to low quality of clusters.

f) ***Interpretability:*** Result and outcomes quality must be interpretable, usable and comprehensible.

1.2 K-means Algorithm

In order to solve the Clustering problem K-Means is most known, simplest and widely used learning algorithm. The approach of K-Means classify the data objects in a very simple and easy way to calculate an optimal number of clusters (suppose k clusters) that are already fixed. The core concept is to define k centroids, only one per cluster. The placement of these clusters must be in a cunning way due to different location because of different result. Hence, the good choice is to place these far away as much as possible [2]. The coming steps takes each point which belongs to a predefined set of data and assign them the possible nearest centroid. When it's done with all points, very first step is complete and an early group age is done. After this the need to again calculate new k centroids as barycenter for all the clusters that results from the first step. Finally we have all these new k centroids, the need for new binding has to be complete between the similar data set objects and the new centroid which is nearest. A loop for this can be created. In the results of this loop we come across that the k centroids in steps changes their location until no new changes takes place. In different words we can say centroids don't change any more.

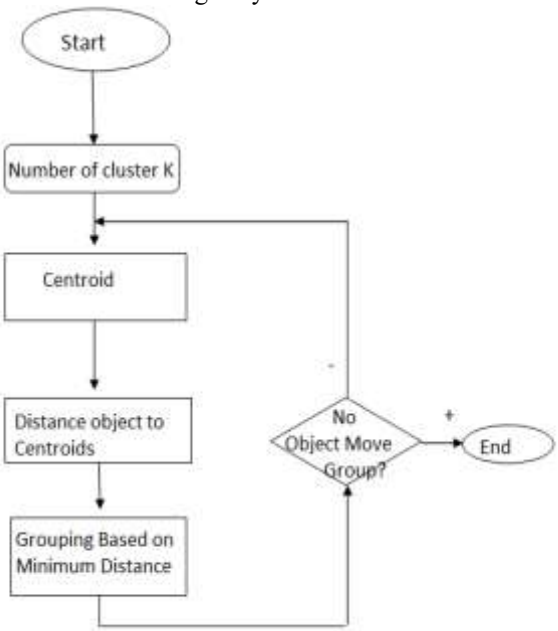

Figure 2: Steps of K-Means Algorithm

1.3 K-Means Algorithm

K-means algorithm can be defined using simple steps, as follows

Input:

D= [d1,d2,…dn] ///sets of n data items

K = number of desired clusters

Output:

A set of k clusters.

Steps:

1. Select k data items from D as initial centroids;
2. Repeat the process of selecting the items
3. Assign the each item di to the cluster which has the nearest and the suitable centroids;
4. Calculate the new mean value for each cluster;
5. Repeat the process until the criteria is satisfied. [3], [10], [11].

The working of Algorithm can be explained clearly with the help of an example, in the first step there are two sets of objects. Then the centroids of both sets are determined. According to the centroid again the clusters are formed which gave the different clusters of dataset. This process repeats until the best clusters are achieved. [9]

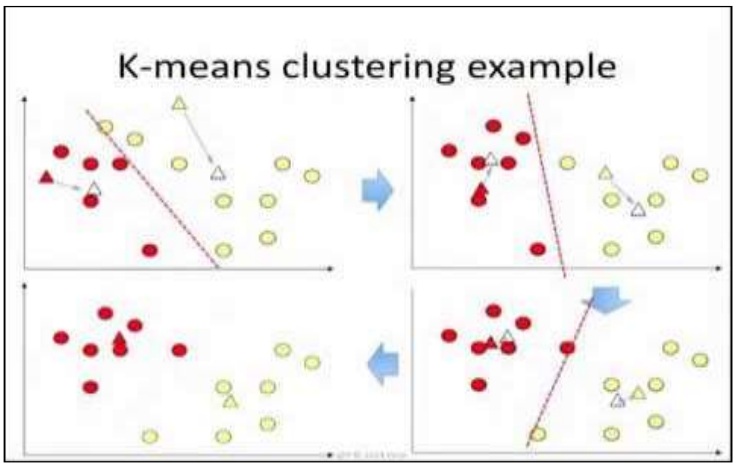

Figure 3: Working of K-Means Algorithm

1.4 K-Means Limitations

K-means clustering has some of the limitations which need to get overcome. Several people got multiple limitations while working on their research with K-means algorithm. Some of the common limitations are discussed below

1.4.1 Number of clusters

Determining the number of clusters in advance is always been a challenging task for K-means clustering approach. It is beneficial to determine the correct number of clusters in the beginning. It has been observed that sometimes the numbers of clusters are assigned according to the number of classes present in the dataset. Still it is an issue that on what basis the number of clusters should be assigned [12].

1.4.2 Empty clusters in K-means

If no points are allocated to a cluster during the starting step, then the empty clusters problem occurs. It was a former problem with the traditional K-means clustering algorithm [8].

1.4.3 Shortcomings of K-Means algorithm

From the above analysis of algorithms, the algorithm has to calculate the distance from each data object to every cluster center in each iteration. However, by experiments it finds that it is not necessary for us to calculate that distance each time. Assuming that cluster C formed after the first j iterations, the data object x is assigned to cluster C, but in a few iterations, the data object x is still assigned to the cluster C.
Along with this if there is no point allocated to a cluster during the starting step, and then the empty clusters problem occurs.

1.5 Problem Statement

The Traditional K-Means algorithm accepts the number of clusters (K) as input from the user [6]. Practically it is very difficult to fix the value of k in advance and it also leads to a poor quality cluster and sometimes ends up with the problem of empty clusters [13]. Another major problem with K-Means algorithm is, it spends a lot time while execution because of computing the distances between data objects and cluster centers in a serial manner that leads to a very expensive execution time.

II. LITERATURE SURVEY

2.1 Dunn's Index

The Dunn index (DI) was introduced by J. C. Dunn in 1974 can be defined as a metric for evaluation of clustering algorithms [1]. This is part of a group of validity indices, in that it is an internal evaluation method, where all the outcomes are based on the clustered data.
As do all other such indices, the aim is to determine the sets of clusters which are compact, with a very small variance between data objects of the cluster, and well separated, where the means of different clusters are completely far apart, as compared to the internal cluster variance.
For a given assignment of clusters, a higher Dunn index indicates improved clustering. One of the major drawback of using this approach is the computational cost as the number of clusters and dimensionality of the data increase.

2.1.1 Preliminaries

There are many ways to define the size or diameter of a cluster. It could be the distance between the farthest two points inside a cluster; and it could also be the mean of all the pair wise distances between data points inside the cluster, or it could as well be the distance of each data point from the cluster centroid. Each of these formulations is mathematically shown below:

Let Ci be a cluster of vectors. Let x and y be any two n dimensional feature vectors assigned to the same cluster Ci.

$$\Delta_i = \max_{x,y \in C_i} d(x,y) \quad \text{..... (i)}$$

Which calculates the maximum distance.

$$\Delta_i = \frac{1}{|C_i||C_i - 1|} \sum_{x,y \in C_i, x \neq y} d(x,y) \quad \text{….. (ii)}$$

Which calculates the mean distance between all pairs.

$$\Delta_i = \frac{\sum_{x \in C_i} d(x,\mu)}{|C_i|}, \mu = \frac{\sum_{x \in C_i} x}{|C_i|} \quad \text{…….. (iii)}$$

Calculates distance of all the points from the mean.

This can also be said about the inter cluster distance, where similar formulations can be made, using either the closest two data points, one in each cluster, or the farthest two, or the distance between the centroids and so on. The definition of the index includes any such formulation, and the family of indices are formed called Dunn-like Indices. Let

$\delta(C_i, C_j)$ Be this inter cluster distance metric, between clusters Ci and Cj.

With the above notation, if there are m clusters, then the Dunn Index for the set is defined as:

$$DI_m - \frac{\min_{1 \leqslant i < j \leqslant m} \delta(C_i, C_j)}{\max_{1 \leqslant k \leqslant m} \Delta_k} \quad \text{…. (iv)}$$

2.2 Task Parallelism (Task Parallel Library)

The Task Parallel Library (TPL) is based on the concept of a task, which represents an asynchronous operation. In some ways, a task resembles a thread or Thread Pool work item, but at a higher level of abstraction. The term task parallelism refers to one or more independent tasks running concurrently. Tasks provide two primary benefits: More efficient and more scalable uses of system resources.

Behind the scenes, tasks are queued to the Thread Pool, which has been enhanced with algorithms that determine and adjust to the number of threads and that provide load balancing to maximize throughput. This makes tasks relatively lightweight, and can create many of them to enable fine-grained parallelism.

More programmatic control than is possible with a thread or work item.

Tasks and the framework built around them provide a rich set of APIs that support waiting, cancellation, continuations, robust exception handling, detailed status, custom scheduling, and more.

For both of these reasons, in the .NET Framework, TPL is the preferred API for writing multi-threaded, asynchronous, and parallel code.

The Task Parallel Library (TPL) is a set of public types and APIs in the System. Threading and System.Threading.Tasks namespaces.

The purpose of the TPL is to make developers more productive by simplifying the process of adding parallelism and concurrency to applications. The TPL scales the degree of concurrency dynamically to most efficiently use all the processors that are available. In addition, the TPL handles the partitioning of the work, the scheduling of threads on the ThreadPool, cancellation support, state management, and other low-level details. By using TPL, you can maximize the performance of your code while focusing on the work that your program is designed to accomplish.

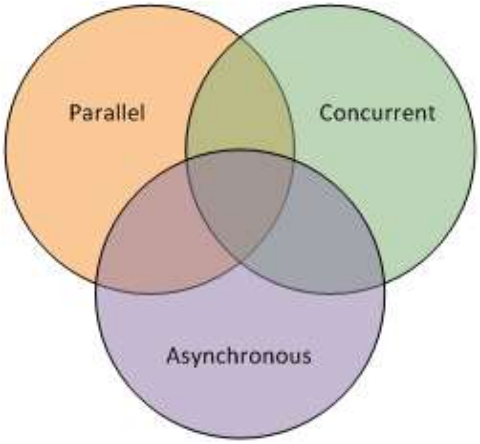

Figure 4: Features of Task Parallel Library

Starting with the .NET Framework 4, the TPL is the preferred way to write multithreaded and parallel code. However, not all code is suitable for parallelization; for example, if a loop performs only a small amount of work on each iteration, or it doesn't run for many iterations, then the overhead of parallelization can cause the code to run more slowly. Furthermore, parallelization like any multithreaded code adds complexity to program execution. Although the TPL simplifies multithreaded scenarios, it recommends that you have a basic
Understanding of threading concepts, for example, locks, deadlocks, and race conditions, so that use the TPL effectively.

2.3 Related Work

In this section the related works that propose the problems of K-Means clustering algorithm are discussed. A number of papers have been published regarding the improvement of quality of k-means clustering, here are few of them that are found most appropriate with this research work.

Jeyhun Karimov and Murat Ozbayoglu in year 2015 in their research titled "Clustering Quality Improvement of k-means using a Hybrid Evolutionary Model" presented an approach for choosing good candidates for the initial centroid selection process for compact clustering algorithms, such as k-means, is essential for clustering quality and performance [14].
In their research they proposed a novel hybrid evolutionary model for k-means clustering (HE-kmeans). Their model uses meta-heuristic methods to identify the “good candidates” for initial centroid selection in k-means clustering method. The results indicate that the clustering quality is improved by approximately 30% compared to the standard random selection of initial centroids.

They improved the clustering quality but did not solve the problem of fixed numbers of static cluster as input. They also did not targeted any approach with parallel computation of clusters.

GrigoriosTzortzis and AristidisLikas in year 2014 under their research titled "The MinMax k-Means clustering algorithm" proposed the Min Max k-Means algorithm, a method that assigns weights to the clusters relative to their variance and optimizes a weighted version of the k-Means objective [15].
In their approach Weights are learned together with the cluster assignments, through an iterative procedure. The proposed weighting scheme limits the emergence of large variance clusters and allows high quality solutions to be systematically uncovered, irrespective of the initialization.
They performed some Experiments to verify the effectiveness of their approach and its robustness over bad initializations, and they compared it with both k-Means and other methods from the literature that consider the k-Means initialization problem.
They targeted the K-means initialization problem but did not considered the parallel execution and computation of clusters.

Ahamed Shafeeq B M and Hareesha K S in 2012 under their research titled "Dynamic Clustering of Data with Modified K-Means Algorithm" proposed an approach for modified Kmeans algorithm with the intension of improving cluster quality and to fix the optimal number of cluster. With their approach user has the flexibility either to fix the number of clusters or input the minimum number of clusters required.
Finally they shown that how the modified k-mean algorithm will increase the quality of clusters compared to the K-means algorithm.

III. PROPOSED APPROACH

The standard k-means algorithm need to calculate the distance between every data object and the centers of k clusters when it executes the iteration every time; it takes up more execution time mainly for large datasets as it executes in serial manner.

Proposed approach overcomes this problem as its uses Dunn’s index approach for the calculation of finding the total number of clusters from given data objects.

This approach also speed ups the execution time as it performs the execution in parallel manner with the help of Microsoft’s task parallel libraries.

IV. EXPERIMENTAL SETUP

In this research, all the tests are performed under following specifications:

1) **Host System:** Intel i5 processor with 4 GB RAM and 500 GB Hard disk.
2) **Operating Environment:** Windows 8.1
3) **C#**
4) **Microsoft Task Parallel Library**
5) **Visual Studio**

a) **Speedup:** Speedup Sp can be defined in terms of the ratio of the execution time using single core of the sequential algorithm in order to solve a problem to the time used by the algorithm implementing parallel computing while solving the same problem on ***p*** processor [7].
All the processors used by the parallel algorithm must be having same characteristics with the processor which is used in the sequential algorithm.

$$\textbf{Speedup } Sp = \frac{\textbf{Sequential time for Scaled} - \textbf{up workload}}{\textbf{Parallel time for Scaled} - \textbf{up workload}}$$

b) **Execution Time:** Execution time can be defined in terms of time consumed by an algorithm in order to solve a problem using processor p.

- **Serial runtime:** The total time consumed in the overall execution of the serial calculations. Generally denoted by TS.
- **Parallel runtime:** Total time consumed in the overall execution of parallel calculations, computations and processing elements (PE). Generally denoted by TP.

V. RESULTS AND ANALYSIS

Following results were found for serial and parallel K means while having cluster size is 6, on different size of datasets. When K=6

TABLE 1: Execution time of Serial k-means versus Parallel k-means for clusters k=6

Data	K-Means	Parallel K-Means	SpeedUP
400	1096	804	35
1200	1422	1149	24
2300	2552	1759	45

Figure 5: Running Time Comparisons of Serial k-means versus Parallel k-means, k=6 Clusters.

2. Following results were found for Serial and parallel K means while having cluster size is 9, on different size of datasets.

TABLE 2: Execution time of Serial k-means versus Parallel k-means for clusters k=9

Data	K-Means	Parallel K-Means	SpeedUP
400	986	864	13
1200	1856	1359	37
2300	2639	1742	51

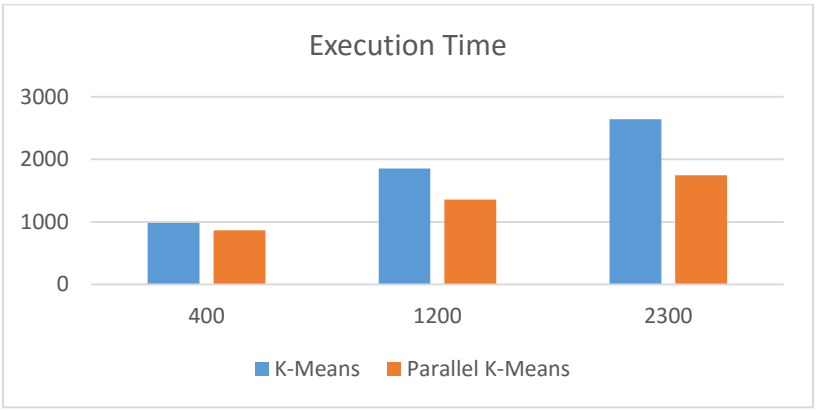

Figure 6: Running Time Comparisons of Serial k-means versus Parallel k-means, k=9 Clusters.

Following results were found for Serial and parallel K means while having cluster size is 18, on different size of datasets.

TABLE 3: Execution time of Serial k-means versus Parallel k-means for clusters k=18

Data	K-Means	Parallel K-Means	SpeedUP
400	1452	1128	28
1200	1764	1294	36
2300	2276	1686	34

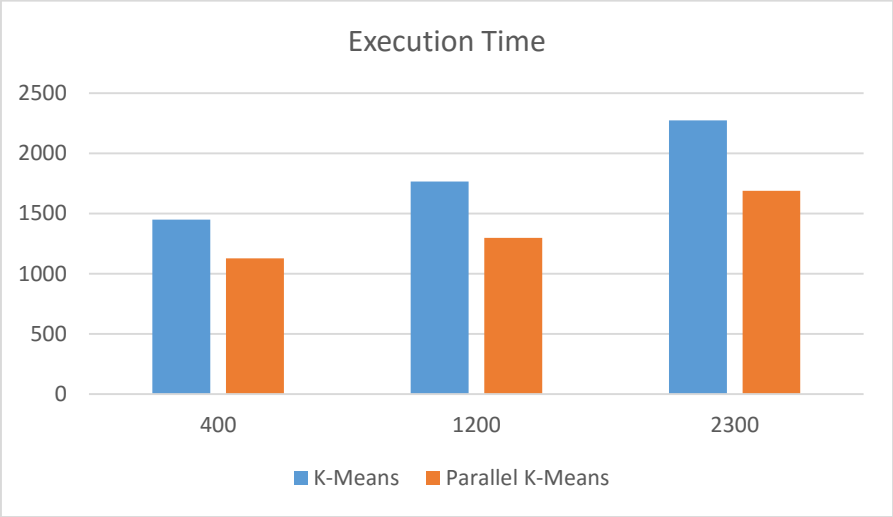

Figure 7: Running Time Comparisons of Serial k-means versus Parallel k-means, k=18 Clusters.

These results came out from the analysis between the Serial and Parallel K-Means with Dunn's index. Here operations are performed on different size of datasets and in Dynamic K-Means clusters size is calculated by using Dunn's Index.

VI. CONCLUSION

This research proposed an improved data clustering method for the anonymous data set. The algorithm works fine for the unknown data set with improved results than traditional K-means clustering. The k-means algorithm is well known for its ease and the alteration is done in the proposed method with maintenance of simplicity. The traditional K-means algorithm obtains number of clusters (K) as input from the abuser. The major problem in traditional K-means algorithm is fixing the number of clusters in advance.

The results shows that the proposed approach has overcome the problem by finding the optimal number of clusters with the help of Dunn's index, and use of parallel libraries enables the algorithm to perform better than conventional K-means algorithm in all scenarios with small or large datasets.

VII. FUTURE SCOPE

Future work can focus on how to reduce the time complexity without compromising cluster quality and optimality. More experiments can be conducted with natural datasets with different features. We can also use some more powerful parallel programming models like Microsoft's Parallel Patterns Libraries and OpenMP to obtain reduced execution time.

VIII. REFERENCES

[1] Michelin and J. Hanmorgan Kauffman, "Data mining concepts and techniques", year 2006.
[2] U.Maulik, and S.Bandyopadhyay, "Genetic Algorithm-Based Clustering Technique" Pattern Recognition 33, 1999.
[3] M.Murty & K.Krishna,"Genetic k-means Algorithm,"IEEE Transactions on System, Vol.29, No.3, 1999.
[4] Manish Verma, Mauly Srivastava, NehaChack, Atul Kumar Diswar and Nidhi Gupta, " A Comparative Study of Various Clustering Algorithms in Data Mining," International Journal of Engineering Reserch and Applications (IJERA),Vol. 2, Issue 3, pp.1379-1384, 2012
[5] Mukul Sharma and Pradeep Soni, "Comparative Study of Parallel Programming Models to Compute Complex Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 96– No.19, June 2014
[6] TIAN Jinlan ZHU Lin ZHANG Suqin, "Improvement and Parallelism of K-Means Clustering Algorithm", TSINGHUA SCIENCE AND TECHNOLOGY, ISSN 1007-0214 Pages 277-281, Volume 10, Number 3, June 2005
[7] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh et. al, "Top 10 algorithms in data mining", Knowledge and Information Systems, Volume 14, Issue 1, Pages 1-37, January 2008
[8] E. Kijsipongse and S. U-ruekolan, "Dynamic load balancing on GPU clusters for large-scale K-Means clustering", IEEE International Joint Conference on Computer Science and Software Engineering (JCSSE), Pages 346-350, 2012

[9] P.Bradley, and U.Fayyad, "Refining Initial Points for K-means Clustering,", In Proceeding of 15th International Conference on Machine Learning, 1998.
[10] K.A Abdul Nazeer and M.P Sebastian, "Improving the Accurancy and Effiency of the K-means Clustering Algorithm", WCE London, 2009
[11] O. A. Abbas, "comparisons between data clustering algorithms", the international Arab journal of information technology, vol. 5, no. 3, July 2008.
[12] Monika Sharma and Jyoti Yadav, "A Review of K-mean Algorithm", International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 7- July 2013.
[13] Grigorios Tzortzis and Aristidis Likas, "The MinMax k-Means clustering algorithm", Pattern Recognition 47(2014)2505–2516, Pages 2505-2516, Elsevier Ltd, February 2014.
[14] Jeyhun Karimov and Murat Ozbayoglu, "Clustering Quality Improvement of k-means Using a Hybrid Evolutionary Model", Procedia Computer Science, Volume 61, Pages 38–45, Complex Adaptive System, Elsevier Ltd, 2015.