GS International Conference on Computer Science on Engineering 2020 (GSICCSE 2020), August 2, 2020

An Improved Version of K-Means Clustering Algorithm

Muhammad Zahid Hussain
Department of Computer Science
Riphah International University
Lahore, Pakistan
m.zahid@riphah.edu.pk

Imran Ahmad
Department of Computer Science
Riphah International University
Lahore, Pakistan
imran.ahmad@riphah.edu.pk

***Abstract*— Big data refers to a very strong growth of heterogeneous data flows due to the increased use of new technologies. Although the massive volume of data may be very useful for humanity and corporations. Therefore, a large volume of data or big data has its own drawback as well. Due to large volume, huge storages are required and operations such as analytical operations, process operations and retrieval operations etc. takes lot of time and occupy so many resource for a long period of time comparatively. To overcome certain difficulties and to avoid from hazards, clustering techniques are introduced. Clustering is the process of grouping the data based on their similar properties. One of the most popular clustering method is k-means. The results of the algorithm are influenced by the initial centroids. Various initial configurations might lead to different final clusters. The cluster's center is defined as the mean of the items in a cluster. In standard k-means algorithm there are most computations. The more iterations are performed, less centroids deviate from their current position. We proposed an improved k-means algorithm to get better performance. We have tried to find data objects that may not change their cluster during next iteration to avoid unnecessary computations. We tried to improve in terms of running time when dealing with large volumes of data.**

Keywords*— k-means algorithm, Big data, Clustering*

I. INTRODUCTION

"Big Data" is assumed as huge amount of data with difficult structure, difficult storing capacities and visualization for more processing. Uses of new technologies are main source for generating big data because of the use of social network, online transactions, growth of the web, and lot of other communicating objects are growing faster every day [1]. Clustering is procedure of grouping the data, on bases of their same properties. All the elements in each cluster should be similar [3]. Data mining algorithm clustering, dimension reduction, parallel clustering and map reduced based clustering are types of clustering [1]. Data mining is the exploration of data and it is technique to use software for finding patterns and uniformities of data set. One of very important and wide field in data mining is clustering. Clustering is process of grouping the elements of data sets on the basis of their similar properties [12].Partitioned based clustering is one of important type of data mining algorithm clustering in which different algorithms like k means, k modes, k medoids, CLARANS, PAM, CLARA and FCM are used [4]. For clustering implementation, one important and famous algorithm is k means clustering algorithm [20]. Due to best performance for big data sets, the usage of this algorithm is very common [5], [6]. In standard k means method, first we choose k points as initial centroids, each centroid represents a cluster. Then we assign all objects of data set to centroid having minimum distance. Mostly Euclidean distance is used for this purpose. After allocation of all data items, centroids are calculated again. We repeat the process of objects allocation and centroids recalculation until no further objects change their cluster [7]. K means clustering algorithm has become a mainstay for the data analyst in lot of fields. The k means method is one of famous and widely used algorithm for clustering in area of "Data Mining" [14], [16].

Steps of Standard K Means Clustering Algorithm

Step 1st: Choose k points as initial centroids, each centroid represent a cluster.
Step 2nd: Allocate each object of data set to nearest (closest) centroid.
Step 3rd: After allocation of all objects, centroids are recalculated.
Step 4th: Second and third steps are repeated until centroids change their positions [7].
In standard k means clustering algorithm, the allocation of all points to related clusters or centroids takes maximum execution time for large data sets, because we need to reassign all data items during each iteration, so there is need to improve this algorithm to reduce execution time and it should be suitable for all types of data [8]. Our new proposed k-means clustering algorithm removes this deficiency. It takes less execution time because it eliminates unnecessary distance computations.

Problem Statement

Proposed an improved version of k means clustering algorithm to avoid unnecessary computations and it takes less execution time. Actually 2nd step of standard k means clustering algorithm takes maximum execution time. In this step, distance is calculated between all points of data set and centroids. After certain number of iterations, a lot of points do not change their clusters but distance between all points and centroids is calculated again.

Our Proposed algorithm

New proposed k means algorithm is applied on datasets which are composed of two dimensional points. In our

proposed algorithm, we have tried to find the points which may not change their cluster during next iteration to avoid calculations for these points and we saved execution time. Second important thing which reduce execution time is width intervals of large data set. All points of data set were grouped into wide interval. Instead of visiting all points of data set, we do it against a point of group.

Shortcoming Covered in Proposed Algorithm
Our new methodology for the k means algorithm, removes the deficiency of standard k means algorithm. It takes less execution time because it eliminates unnecessary distance computation.

Evaluation of Proposed Algorithm
Our implementation on specific data set was evaluated. We composed data sets of two dimensional points. There were generated randomly data sets of different points and random number generator was used for this purpose. Execution time of improved k-mean was measured on different wide intervals to find appropriate wide intervals with minimum execution time. Each data set was split in twelve, sixteen and twenty clusters. Running time of improved k means has been compared with the running time of the standard k means clustering algorithm. Compared results showed an encouraging reduction in execution time.

II. LITERATURE REVIEW

Lot of people tried many time to improve the efficiency and productivity of k means clustering algorithm. Some people tried to improve accuracy of algorithm, quality of clusters is also improved by some methods and some optimized variations improved the running time of k means algorithm [9], [10], [11].

Some deficiencies of basic k means was removed by Pallavi Purohit et al [24]. They proposed a new methodology for basic k means clustering algorithm. According to this approach, the algorithm first computes k number of initial points that are known as centroids according to the requirements. The proposed algorithm gives best clustering results without losing accuracy.

A heuristic approach for implementing k means clustering algorithm was proposed by D. Mariammal et al [21] on multidimensional data, which was based on attribute with maximum range. It was enhanced version of standard k means to obtain clustering results more accurate. In this research there was presented an algorithm to find initial centroids.

To assign data items of data set to appreciate cluster, there was proposed a new version of k means method by Fahim A M et al [10]. For this purpose Fahim's technique creates use of two distance functions. First one is same as basic the k means clustering method and second one function is investigative to decrease the number of computations. The employment of proposed method is very easy because the algorithm requiring a simple data structure. The results of improved algorithm were compared with basic k means clustering, execution time and quality of clusters was improved.

R.Ranga Raj et al [22] proposed an improved clustering method which increases the accuracy and proficiency. In their research, the improved k means clustering algorithm was implemented on the large student dataset to determine the different groups and categories. But there is need to give as an input the number of required clusters, regardless of the separation of the data objects.

There was proposed less similarity based clustering algorithm by Gomathi. D et al [23]. The enhanced method was able to find the better initial centroids and to be responsible for an effective way of allocating the data objects to appropriate clusters. The time complexity was also reduced in this algorithm. The core objective emphases on the technique of using a reduced amount of similarity based clustering algorithm, to discover the initial centroids proficiently.
Chunfei Zhang et al [19] improved k means clustering algorithm. Their improvement was based on choosing initial points, and elaborates the method of improving the K-means clustering algorithm based on improve the initial focal point and determine the K value. Results of their experiments showed that the upgraded k means clustering method is more reliable for clustering, further enhanced k means algorithm escape the influence of the noise data in the dataset object to guarantee that the ending clustering results are more correct and effective.

Another version of k means algorithm was proposed by Kohei Arai et al [16] named as hierarchical k means clustering algorithm. It employs all results of k means clustering in the definite times, however lot of them touch the local optima. They converted all center points of clustering results by merging with hierarchical method to decide the initial center points known as centroids for the k means clustering.

Kajal C. Agrawal et al [13] presented a modified k means clustering algorithm. The central idea of algorithm is to use two data structures, to hold the markers or labels of all centroids or cluster, also to retain the distance of all points to the closest group through the each repetition that may use in coming iteration.

III. RESEARCH METHODOLGY

A. Finding the step of standard k means that causes most computations

Obviously step number two of K-Means algorithm takes biggest time of execution for large data set because distance of all objects from all centroids is calculated.

One of most important thing to save our execution time is "is it necessary to visit all points for large data set?" For this purpose, two dimensional objects were used to describe or represent data objects. The algorithm was specially designed to change the positions of its centroids. We analyze that as long as the execution progresses, the centroids after changing their positions continuously from initial position and reaches

at-most closer to the previously found final position. Most of the time after some iterations, the centroids closely move towards their final position during last iteration. Hence we can infer after having deep observations that most of the data objects that belong to the same cluster whose centroids moves slightly. Because of slight movement of centroids, the data objects should remain closer to the last position and also the part of the same cluster for the next iteration. So it is understandable that the movement of centroids will affect the position of the points. As long as the centroids move, the points will be get affected as much. We have focused to identify that which of the data objects may be affected by the move, this finding can lead us towards a very essential improvement during step number 2 wherein we may not reiterate the entire data set, and we may have to visit a very small data set. Therefor we can save our execution time if we find the way to avoid unnecessary computations for the objects that do not change their position as well as centroids for next iterations.

B. Proposed solution to avoid unnecessary computations

Being able to determine which objects of large data set could not change their centroids during next iteration, we have no need to visit complete data set, only small part of data set that may change centroids, should be visited. We denote these objects as "boundary layer". Before deciding which objects of large data set should be part of this boundary layer, we need to establish a criteria for the finalization that which data objects should be the part of boundary layer by fulfilling the criteria it may be considered a boundary element. Let we have three clusters X, Y and Z. We have point "pt" which is part of cluster Z. We omit all other points for our best presentation. As point pt is part of cluster Z, so the distance of point pt to cluster Z is less than distance of point pt to cluster X and cluster Y. At end of iteration, all centroids will move towards new positions. We assume that centroid X moved to X_1, centroid Y moved to Y_1 and centroid Z moved to Z_1. In worst situation for point pt, point Z became further away from point pt which is $|ZZ_1|$ while point A became closer $|XX_1|$ and point B became closer $|YY_1|$. The distance of point pt to its edge is $e_{pt,}$ mean point pt is e_{pt} away to switch in new cluster. The conditions that point pt will stay in the cluster C would be:

$$e_{pt} > |ZZ_1| + |XX_1| \quad \text{and} \quad e_{pt} > |ZZ_1| + |YY_1|$$

In this way we found a criteria to check whether the point is part of boundary layer or not. We are not found complete solution yet, still we need to compute every object of data set for every iteration to include an object in boundary layer. To escape these computations, all objects of our data set grouped into wider intervals as shown in proposed algorithm. Instead of visiting entire data set, proposed algorithm groups the points which are near the edge and we can do it for a complete group. So we can say that the constant WIDTH has a great impact on improvement. If we increase or decrease value of width then there will large effect on number of computations, and directly effect on execution time.

C. Improved k means clustering algorithm

1. Define Constant WIDTH
2. Defined intervals li and calculated li as I * WIDTH, (i increment by 1) * WIDTH
3. Mark all data set as visited
4. For each point to b traversed
5. Calculate the value of e
6. Find out points with I * WIDTH that should be less than value of e, and value of e should be less than value of (i+1) * WIDTH, i is positive integer.
7. Calculate new centroids
8. After that the value of li will be updated by subtracting 2 * D (points taken by the intervals got closer to the edge by 2 * D).
9. Hold the value of all points inside the intervals whose tag is less or equal to 0, and repeat step 4 to traverse it again.

D. Assessment of Improved K-Means Algorithm

i. Generation of data sets

For this purpose there were generated 2D points with coordinates between 0 and 1. Then there were generated randomly data sets of 100000 points, 200000 points, 300000 points, 400000 points and 500000 points. The random number generator was used for this purpose.

Each data sets split into different number of clusters
Standard k means algorithm and proposed k means algorithm were applied to split data sets in different number of clusters. Each data set was split into twelve, sixteen and twenty number of clusters.

ii. Time measured for standard k means and proposed k means

Execution time of standard k means as well as proposed k-means clustering algorithm was calculated in milliseconds for each number of clusters on different data sets.

iii. Comparison of results

The running time of proposed algorithm was compared with running time of standard k means algorithm for all number of clusters. This comparison is also represented graphically.

IV. RESULTS

For this proposed research study, there were generated randomly datasets of sizes 100000 points, 200000 points, 300000 points, 400000 points and 500000 points. Each data set was split in twelve, sixteen and twenty number of clusters. Before finding the final results, proposed algorithm was also tested on different width intervals to find appropriate width. The running time of standard k-means algorithm and proposed k-means algorithm was measured on specified datasets and number of clusters. Then we compared the running time of the proposed k-means algorithm to the running time of the standard k-means algorithm.
Details of results are as under:

Table.1. RUNNING TIME ON DIFFERENT WIDTH INTERVAL.

Twelve Clusters, Running Time of Proposed K Means on Different Wide Intervals										
Size of Datasets	**100000**		**200000**		**300000**		**400000**		**500000**	
Running Time of Proposed Algorithm on Different Width Intervals	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**
	0.05	23772	0.05	48202	0.05	73045	0.05	96158	0.05	124122
	0.07	22185	0.07	43509	0.07	64064	0.07	86744	0.07	109124
	0.1	19271	0.1	38922	0.1	58695	0.1	79384	0.1	98555
	0.13	24290	0.13	48599	0.13	70675	0.13	95231	0.13	117518
	0.15	26541	0.15	52033	0.15	78410	0.15	107397	0.15	131922
	0.17	26555	0.17	51987	0.17	79130	0.17	106497	0.17	133432
	2.0	37728	2.0	75685	2.0	115528	2.0	145765	2.0	194029
Sixteen Clusters, Running Time of Proposed K Means on Different Wide Intervals										
Size of Datasets	**100000**		**200000**		**300000**		**400000**		**500000**	
Running Time of Proposed Algorithm on Different Width Intervals	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**
	0.05	34441	0.05	68182	0.05	102596	0.05	134667	0.05	170790
	0.07	31329	0.07	62793	0.07	99444	0.07	128954	0.07	160917
	0.1	30051	0.1	59977	0.1	92040	0.1	125080	0.1	146130
	0.13	34167	0.13	68846	0.13	105399	0.13	140578	0.13	175572
	0.15	37567	0.15	76944	0.15	116103	0.15	155008	0.15	194144
	0.17	37632	0.17	76052	0.17	116131	0.17	154806	0.17	194241
	2.0	50135	2.0	99132	2.0	147076	2.0	196659	2.0	250112
Twenty Clusters, Running Time of Proposed K Means on Different Wide Intervals										
Size of Datasets	**100000**		**200000**		**300000**		**400000**		**500000**	
Running Time of Proposed Algorithm on Different Width Intervals	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**	**Width intervals**	**Running Time (Milliseconds)**
	0.05	47276	0.05	90517	0.05	138075	0.05	184429	0.05	240675
	0.07	42656	0.07	84191	0.07	128958	0.07	169386	0.07	210134
	0.1	41067	0.1	82861	0.1	119250	0.1	152156	0.1	193458
	0.13	45850	0.13	91543	0.13	138915	0.13	186690	0.13	232617
	0.15	50342	0.15	100792	0.15	151381	0.15	203502	0.15	253676
	0.17	50015	0.17	102372	0.17	151971	0.17	201931	0.17	258304
	2.0	61888	2.0	126157	2.0	192175	2.0	257650	2.0	318833

To find appropriate width, the execution time of proposed k-means algorithm was measured on different wide intervals. In table 1, it is shown that two dimensional random generated data sets of 100000, 200000, 300000, 400000 and 500000 points were split into twelve, sixteen and twenty number of clusters on different wide intervals like 0.05, 0.07, 0.1, 0.13, 0.15, 0.17 and 0.2. The best width was found by comparing the execution time on different wide intervals.

A. Twelve clusters, running time of standard vs. improved k means

Table. II. Twelve Clusters, Running time of proposed algorithm vs standard k-means Algorithm

Size of Data Sets	Running Time of Standard K-Means Algorithm in Milliseconds	Running Time of Proposed Algorithm in Milliseconds	Improved (%)
100000	51969	19271	62.91
200000	143492	38922	72.87
300000	188943	58695	68.93
400000	311968	79384	74.55
500000	322963	98555	69.48

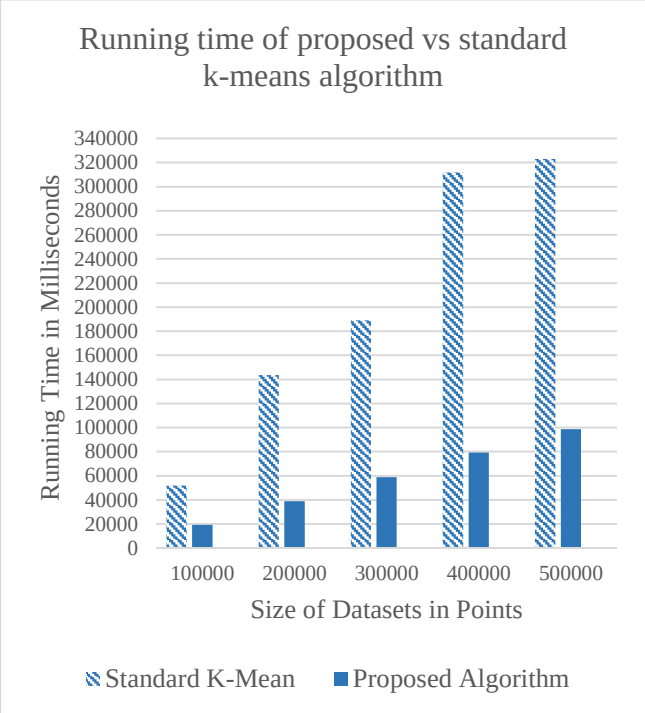

Figure. I. Twelve Clusters, Running time of proposed algorithm vs standard k-means algorithm

In figure 1, the running time of proposed k means clustering algorithm is compared with running time of standard k-means algorithm. Different datasets were split into twelve clusters. Running time is shown along y-axis in milliseconds and comparable different datasets of standard k-means algorithm as well as improved algorithm are shown along x-axis. The results are certainty encouraging.

B. Sixteen clusters, running time of standard vs. improved k means

Table. III. Sixteen Clusters, Running time of proposed algorithm vs standard k-means Algorithm

Size of Data Sets	Running Time of Standard K-Means Algorithm in Milliseconds	Running Time of Proposed Algorithm in Milliseconds	Improved (%)
100000	102989	30051	70.82
200000	208952	59977	71.29
300000	313715	92040	70.66
400000	421800	125080	70.34
500000	506653	146130	71.15

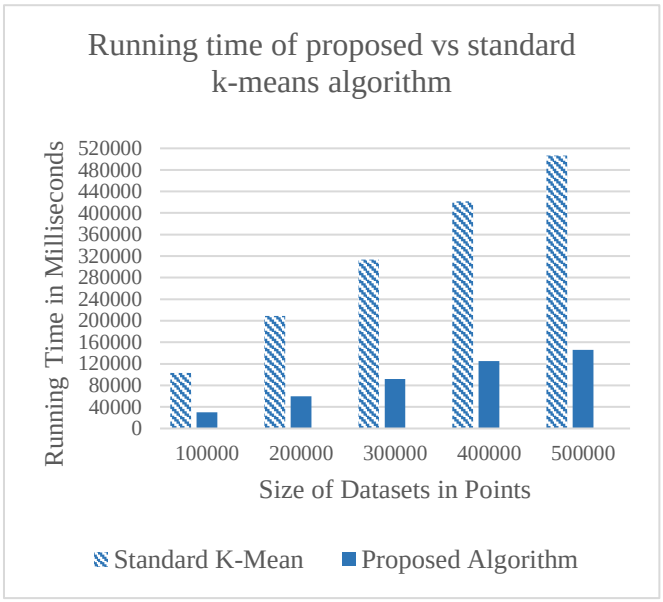

Figure. II. Sixteen Clusters, Running time of proposed algorithm vs. standard k-means algorithm.

In figure 2, the running time of proposed k means clustering algorithm is compared with running time of standard k-means algorithm. Different datasets were split into sixteen clusters. Running time is shown along y-axis in milliseconds and comparable different datasets of standard k-means algorithm as well as improved algorithm are shown along x-axis. The results are certainty encouraging.

C. Twenty clusters, running time of standard vs. improved k means.

Table. IV. Twenty Clusters, Running time of proposed algorithm vs standard k-means algorithm

Size of Data Sets	Running Time of Standard K-Means Algorithm in Milliseconds	Running Time of Proposed Algorithm in Milliseconds	Improved (%)
100000	87189	41067	52.89
200000	202338	82861	59.04
300000	308133	119250	61.29
400000	390209	152156	61.06
500000	625386	193458	69.06

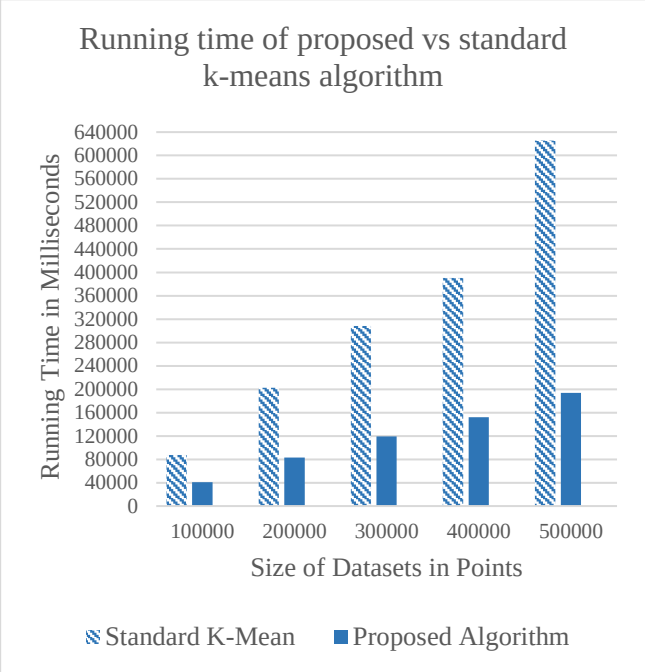

Figure. III. Twenty Clusters, Running time of proposed algorithm vs standard k-means algorithm.

In figure 3, the running time of proposed k means clustering algorithm is compared with running time of standard k-means algorithm. Different datasets were split in twenty clusters. Running time is shown along y-axis in milliseconds and comparable different datasets of standard k-means algorithm as well as improved algorithm are shown along x-axis. The results are certainty encouraging.

V. CONCLUSION

In our proposed algorithm, we suggested a boundary between the data which may change its clusters and the data which may not change its clusters during next iteration. That was very helpful to avoid extra computations and execution time decreased certainly.

REFERENCES

1. Zerhari, B., Lahcen, A. A., & Mouline, S. (2015, May). Big data clustering: Algorithms and challenges. In *Proc. of Int. Conf. on Big Data, Cloud and Applications (BDCA'15)*.
2. Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., ... & Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, *2*(3), 267-279.
3. Rehioui, H., Idrissi, A., Abourezq, M., & Zegrari, F. (2016). DENCLUE-IM: A new approach for big data clustering. *Procedia Computer Science*, *83*, 560-567.
4. Sajana, T., Rani, C. S., & Narayana, K. V. (2016). A survey on clustering techniques for big data mining. *Indian Journal of Science and Technology*, *9*(3).
5. Vrahatis, M. N., Boutsinas, B., Alevizos, P., & Pavlides, G. (2002). The new k-windows algorithm for improving the k-means clustering algorithm. *Journal of complexity*, *18*(1), 375-391.
6. Yedla, M., Pathakota, S. R., & Srinivasa, T. M. (2010). Enhancing K-means clustering algorithm with improved initial center. *International Journal of computer science and information technologies*, *1*(2), 121-125.
7. Velmurugan, T and Santhanam, T. (2011). A Survey of Partition based Clustering Algorithms in Data Mining: An Experimental Approach. *Information Technology Journal*. 10(3) 478-484.
8. Nazeer, K. A., & Sebastian, M. P. (2009, July). Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. In *Proceedings of the world congress on engineering* (Vol. 1, pp. 1-3).
9. Yuan, F., Meng, Z. H., Zhang, H. X., & Dong, C. R. (2004, August). A new algorithm to get the initial centroids. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on* (Vol. 2, pp. 1191-1193). IEEE.
10. Fahim, A. M., Salem, A. M., Torkey, F. A., & Ramadan, M. A. (2006). An efficient enhanced k-means clustering algorithm. *Journal of Zhejiang University-Science A*, *7*(10), 1626-1633.
11. Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, *2*(3), 283-304
12. Rauf, A., Sheeba, S. M., Khusro, S., & Javed, H. (2012). Enhanced k-means clustering algorithm to reduce number of iterations and time complexity. *Middle-East Journal of Scientific Research*, *12*(7), 959-963.
13. Agrawal, K. C., & Nagori, M. (2013). Clusters of Ayurvedic Medicines Using Improved K-means Algorithm. In *International Conf. on Advances in Computer Science and Electronics Engineering*.
14. Bradley, P. S., Bennett, K. P., & Demiriz, A. (2000). Constrained k-means clustering. *Microsoft Research, Redmond*, 1-8.

15. Elkan, C. (2003). Using the triangle inequality to accelerate k-means. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (pp. 147-153).
16. Arai, K., & Barakbah, A. R. (2007). Hierarchical K-means: an algorithm for centroids initialization for K-means. *Reports of the Faculty of Science and Engineering*, *36*(1), 25-31.
17. Chaturvedi, E. N., & Rajavat, E. A. (2013). An improvement in K-means clustering algorithm using better time and accuracy. *International Journal of Programming Languages and Applications*, *3*(4), 13-19.
18. Ashok, P., Nawaz, G. K., Elayaraja, E., & Vadivel, V. (2013). Improved performance of unsupervised method by renovated K-means. *arXiv preprint arXiv:1304.0725*.
19. Zhang, C., & Fang, Z. (2013). "An Improved K-means Clustering Algorithm", *Journal of Information & Computational Science*.
20. Kumar, R., Puran, R., & Dhar, J. (2011). Enhanced k-means clustering algorithm using red black tree and min-heap. *International Journal of Innovation, Management and Technology*, *2*(1), 49.
21. Mariammal, D., Gowthami, M., & Sindhujaa, N. (2013). New algorithm to get the initial centroids of clusters on multidimensional data. *IJREAT International Journal of Research in Engineering & Advanced Technology*, *1*(1)
22. Raj, R. R., & Punithavalli, M. (2012). Evaluation of Enhanced K-MEANS Algorithm to the Student Dataset. *International Journal of Advanced Networking and Applications*, *4*(2), 1578.
23. MCA, T. N. Improving the accuracy and efficiency of k-means algorithm by using less similarity based clustering technique for better initial centroids. *International Journal of Computer Science and Management Research*, Vol 1, Issue 4, November 2012.

Purohit, P., & Joshi, R. (2013). An Efficient Approach towards K-Means Clustering Algorithm. *International Journal of Computer Science & Communication Networks*, *4*(3), 125-129.