A New Hybrid Clustering Technique Based On Mini-batch K-means And K-means++ For Analysing Big Data

G. Yamini[1], Dr. B. Renuka Devi[2]

[1]M. tech Scholar, Computer Science & Engineering, Vignan's Nirula Institute of technology & Science for Woman, Pedapalakaluru Guntur, Andhra Pradesh, India

[2]Assistant Professor, Computer Science & Engineering, Vignan's Nirula Institute of technology & Science for Woman, Pedapalakaluru Guntur, Andhra Pradesh, India

Abstract: In the real time scenario, the volume of data increases day by day. By the year 2020 the volume of big data would reach up to 44 trillion GB as per International Data Corporation (IDC). The clustering of datasets has becoming a big challenging issue in the field of big data analytics, but there are difficulties for applying traditional clustering algorithms on big data due to increasing the volume of data day by day. We have K-Means clustering algorithm, it is very simple and easy to implement, but has a drawback is to choose initial centroids so we can overcome this problem using K-Means++, but this algorithm doesn't give accurate results when we apply on large data sets. In this paper we introduce a new hybrid clustering algorithm which combines the Mini-batch k-means and k-means++, here we have two phases first one is to initialize the centroids using k-means++, This algorithm first choose one centroid from initial data set randomly and then compute d(x) is distance between data point and centroid that has been already chosen. Choose another data point as centroid using Probability distribution, then repeat these steps until k centroids have been chosen. Second one is to draw the random samples and assigned to nearest centroid then update the centroids using Gradient descent. This algorithm reduces the computational time ant it can handle large data sets. This paper compares the proposed algorithm with other traditional algorithms which are k-means, k-means with k-means++ and traditional Mini-batch clustering algorithms.

Keywords: Big data, Clustering, k-means, k-means++, Mini-batch k-means.

I. INTRODUCTION

In recent years the term big data has become very popular word in every field. The volume of data increases day by day. As per International Data Corporation (IDC), the volume of Big Data would reach up to 40zb, by the year 2020. Big data comes from three major sources, which are machine data this data comes from industrial equipment, Social data this data comes from Facebook likes, comments, sharing, Twitter tweets retweets and YouTube views. Which are growing in an unconceivable range. Public web is another source of social data, this can be increase the volume of big data. Transactional data this data comes from payment orders, delivery records, storage records and invoices. There is boundless amount of data that has been generated by the systems using sensors by Data accession techniques. Big data refers to complex and large datasets which cannot be processed using Traditional databases. These type of data varies primarily by 6Vs; volume, velocity, variety, value and veracity. Volume refers Data in rest, Terabytes to Exabyte's of existing data to process. Organizations collects the data from relative sources, the data includes, social media, business transactions and information from Machine-to-machine or sensor data. Velocity means Data in motion, Refers to the speed at which new data is generated and the speed at which data moves around. Determines real potential in the data. Variety of Data in many Forms either structured data that neatly fitted into tables or relational databases. Unstructured data such as text documents, email, video, audio, images, voice, or semi structured such as webpages, xml, and information integration. Value of data having access to big data is no good unless we can turn it into value. The data can be statistical, events, correlations, and hypothetical. Veracity refers Uncertainty due to data inaccurate, inclination, and abnormality. The quality of the data being captured depends on the veracity of the source data.Big data has very low density which means one single observation does not have any significance of its own. With the massive amount of data being generated by people and organizations today. The rest of the paper is organized as follows section 2 describes related work, section 3 describes clustering techniques, section 4 describes proposed method section 5 describes results and its explanation and section 6 conclusion of this paper.

II. RELATED WORK

This section provides review of related works

Ramprasad Raghavan and Darshika G. Perera [1] they were propose an architecture for k-means clustering with Fast and scalable based parallel processing architecture, consider a profound research on K-means algorithm calculation of streamlining. They were set a forward the primary chose starting bunching focus of K-means calculation, toward this end. Asawari patil [2] et.al they have combine the Mini-batch k-means and regular k-means clustering algorithm. This method can apply on large date sets using this

clustering algorithms. They apply K-Means and Mini batch K-Means clustering algorithms on large data sets separately. And then give empirical results which algorithm is best suited for large data sets. Chuan Liu [3], et.al they proposed a profound research on K-implies calculation of advancement. they set forward the main chose introductory grouping focal point of K-means calculation, toward this end, a novel half and half calculation in view of K-means algorithm and Hybrid Rice Optimization Algorithm they were proposed to quickly locate the ideal bunch focuses and abstain from getting into nearby ideal. Caiquan Xiong [4] et.al proposed a k-means algorithm, traditional algorithm has drawback to choose initial centroids. They were introduce an enhanced k-means algorithm, this algorithm initially figures the density of every data object in dataset and after that dole out which dataset is near to centroid. Jungkyu Han [5] et.al proposed a quick k-means strategy in light of factual bootstrapping method. They were propose strategy accomplishes approximately 100 times speedup and comparative precision contrasted with Lloyd calculation it is the well-known k-means calculation in modern field. Javier Béjar[6] et.al they were concentrate an empirical comparisons of k-means and Mini-batch k means here mini batch k means clustering algorithm performs well than k-means algorithm but it has drawback which is centroid initialization it takes more time and it gives bad centroids. Hyuk Cho and Min Kyung An [7] they were develop an online incremental co-clustering algorithm, it focus on updating for row and column clustering statistists. This algorithm can handle stream data. They was focus only implementation than theoretical analysis for this algorithm. Xiao long [8] they presented an enhanced k-means algorithm. This algorithm developed a density based detection methods. K-means algorithm is very efficiently but it has deficiencies which are the number clusters needs, centroid selection and noisy data points. Here they were use noise data filter to advanced k-means algorithm. Bashar Aubaidan [9] et.al they were compare k-means and k-means++ clustering algorithms on crime data. K-means clustering algorithm is easy but it has drawback to choose centroids, k-means++ clustering algorithm overcome this problem. And they was use different distance metrics. David Arthur and Sergei Vassilvitskii [10] they were focus on k-means++ clustering algorithm, this algorithm aims to minimize the average distance between data objects in the same cluster. This algorithm for initialize centroids for traditional k-means clustering algorithm. Which gives unique centroids for k-means. Aditi Anand Shetkar [11] et.al explained k-means and k-means++ and applied k-means and k-means++ on text documents separately and then compared which algorithm is best suit for large data sets. Text categorization is one of the technique and it used for sorting the documents into sets. They were categorize the text documents using k-means and k-means++ clustering algorithm.

III. CLUSTERING TECHNIQUES

In this section we describe partitioning algorithms k-means, k-means++ clustering algorithms, K-means with k-means ++, and Mini-batch k-means.

A. K-Means Clustering Algorithm

K-Means is one of the most popular and commonly used algorithm. Proposed by McQUeen in 1976.K-Means is a partition based clustering algorithm, this algorithm is proved a very efficient way then it can produce good clustering results. It is probing data analysis technique that is it explore the complete data set. K-means implements non-hierarchical method of grouping objects together. But it will take the data set as coming and then it will group them. The data set is divided into K groups based on attributes of the object. Here K is any optimistic integer which represents the number of clusters in the algorithm. K-Means goal is to discover the places of the clusters, which can limit the separation from data objects to cluster.

Algorithm for K-Means

1. Choose k, the number of clusters to be generated.
2. Choose m data objects at random as initial centroids.
  2.1 For each data object
    2.1.1 Calculate its distance from centroid to each data object using Euclidean distance metric;
    2.1.2 Then, assign the each data object to its nearest centroid;
3. Compute means of each cluster and update its centroid;

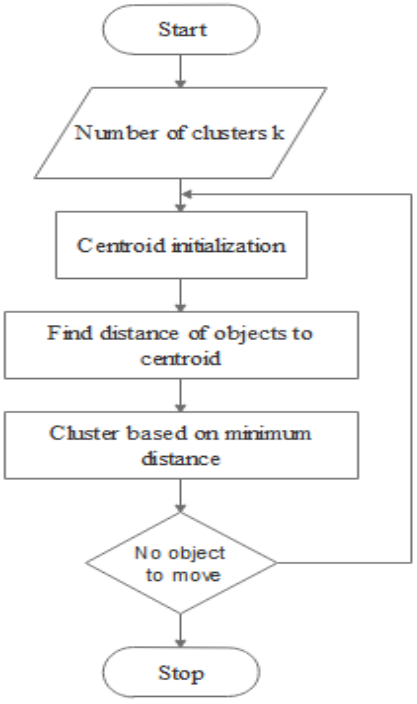

Fig.1 Flow chart for K-Means

4. Iterate until the cluster centroids don't change anymore;

End.

This algorithm has two steps first step is to choose the k the number of clusters that we need. Next we have to choose random data objects from dataset as initial centroids. And then calculate distance between centroid and data objects using Euclidean distance, then after assign each data object to the nearest centroids based on minimum Euclidean

distance. At the point when every one of the data objects are relegated to some clusters, now first step is done and primary alignment is finished. Second step is to recalculate the centroid of the cluster. Along these lines k centroids are iteratively changes their positions in every move until there is no variation in centroid values.

B. K-Means++ Clustering Algorithm

This K-Means ++ is one of the partitioning based clustering algorithm. Proposed by David Arthur and Sergei vassilvitskii in 2007. K-Means++ is improved k-means with a robust centroid initialization procedure, and easy to understand. This algorithms aims to initialize the centroids for traditional k-means clustering algorithm. Because k-means has drawback is to choose centroids randomly so, k-means ++ can overcome this problem.

Algorithm for K-Means++

Step-1 Choose one data point as centroid from initial data set randomly.

Step-2 For each data point x compute D(x), is the distance between x and nearest centroid that has been already chosen.

Step-3 Choose another data point as second centroid using probability distribution.

Step-4 Repeat steps 2 & 3 until required centroids have been chosen.

Step-5 Now initial centroids have been chosen, proceed using k-means clustering algorithm.

This algorithm first choose one centroid from initial data sets randomly and then compute d(x) is distance between data point and centroid that has been already chosen. choose another data point as centroid using Probability distribution, then repeat this steps until k centroids have been chosen whenever initial centroids has selected we proceed using standard k-means clustering algorithm.

C. K-Means with K-Means ++ Clustering Algorithm

K-Means clustering algorithm very simple and easy to understand but the primary concern of the k-means problem is to choose the initial centroids from initial data set. We can overcome this problem using K-Means++ clustering algorithm. Here first we apply k-means++ on data sets for initial centroids, whenever we get require centroids then we have to apply traditional k-means.

Algorithm for K-Means with K-Means++

Input: The number of clusters k;

Dataset X of objects x;

Output: a set of k clusters;

Begin

1. Choose data point c_1 as initial centroid from data set d randomly
2. Repeat until all k centroids have been found
   - **2.1.** For each data point compute D(x)
   - **2.2.** Select another centroid $c_i = x \in X$ with probability p_i where $D(x')^2 / \Sigma\, x \in X\ D(x)^2$ represents the shortest distance from data point to centroid that has been already chosen
3. Run K-Means with selected centroids $c_1, c_2, c_3 \ldots . c_n$ as initialization.
   - **3.1.** Calculate its distance from centroid c_n to each data object x_n using Euclidean distance metric;
   - **3.2.** Then, assign the each data object to its nearest centroid;
4. Compute means of each cluster and update its centroid;
5. Iterate until the cluster centroids don't change anymore;

This algorithm has two phases first one is to initialize centroids using K-Means++ here first choose first centroid from initial data set randomly. Next compute the D(x) is the distance between data object to centroid that has been already chosen. After that we have to choose another centroid using Probability distribution, then repeat this steps until k centroids have been chosen. Whenever initial centroids has been selected we have to apply k-means further steps. We have to calculate distance between data object to centroid using Euclidean distance metric. Then assign each data object to nearest centroid. Again we would compute the means of each cluster update its centroid. We have to repeat this procedure until cluster centroids don't change any more. This algorithm gives slightly different results and also it doesn't give accurate results for large data sets. Handling outliers is also difficult in this algorithm. We can overcome this problems by using Mini-batch k-means clustering algorithm.

D. Mini-batch K-Means Algorithm

There is a modified k-means clustering algorithm which is more efficient than traditional algorithm, is called Mini-batch k-means clustering algorithm. It is mostly useful in web applications where the amount of data can be huge, and the time available for clustering maybe limited. This methodology is also used in other domains such as artificial neural network as a mean to reduce training time for the back propagation algorithm. This algorithm is one of the unsupervised learning algorithm. It can solve the well-known clustering problem. The aim of this algorithm is to represent the dataset by a smaller subset of the data. Mini Batch K-means has been proposed as an alternative to the K-means algorithm for clustering massive data sets. The advantage of this algorithm is to reduce the computational cost by not using all the dataset each iteration but a subsample of a fixed size.

Algorithm for Mini-batch k-means

Step 1 Samples drawn randomly assigned to nearest centroid

Step 2 Repeat for each example: Increment per center, get per center learning rate, $\eta \leftarrow 1 / v[c]$

Step-3 Take Gradient Descent $c \leftarrow (1- \eta)c + \eta x$

This algorithm repeats between two steps, similar to vanilla k-means. In the first step, we would draw samples randomly from the initial dataset, to form a mini-batch. Then these are assigned to the nearest centroid. In the next second step, the centroids are updated. In contrast to k-means, this is done on an each sample basis. For each sample in the mini-batch, the assigned centroid is updated by taking the streaming average of the sample. And all previous samples assigned to that centroid. This algorithm also has the problem is to choose initial centroids, we can detect this problem using k-means++ for initializing the centroids.

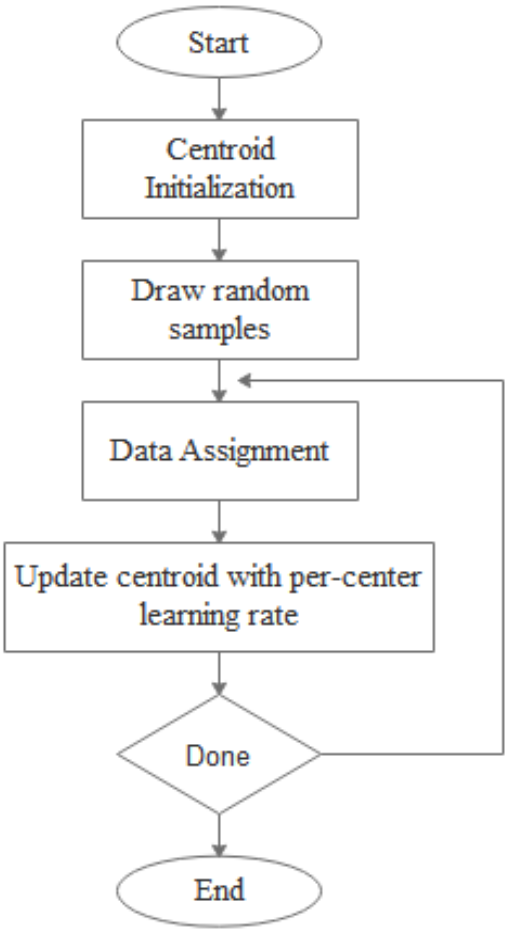

Fig.2 Flow chart for Mini-batch K-Means

IV. PROPOSED METHOD

In this section we describes a proposed method that is a hybrid clustering algorithm which combines Mini-batch k-means and k-means++ clustering algorithms.

A New Hybrid Clustering Algorithm

This Hybrid clustering algorithm combines the Mini-batch and k-means ++ clustering algorithms. It has two phases first one is to initialize the centroids using k-means ++ clustering algorithm. Here we first choose one data object as first centroid from data set randomly, then we have to compute D(x) is shortest distance between data object to nearest centroid that has been already chosen. Now we would choose next centroid using a probability distribution. Then we have to repeat this procedure until k centroids have been chosen. Whenever initial centroids has selected we proceed using Mini-batch k-means clustering algorithm. Here we would draw the random samples from initial data set. It forms Mini-batchs then we have to assign this batches to nearest centroid. In the next the centroids updated using gradient descent. The assigned centroid is updated by taking the streaming average of the sample. And all previous samples assigned to that centroid.

Algorithm for Hybrid Clustering

Input: k,Mini-batch size b,iteration t

D= {x_1, x_2,…..,x_n} // set of data objects

Output: K= {C_1, C_2,…….C_k} //Set of k clusters

Begin

v ← 0

for i=1 to t do

M←b examples draw randomly form D

1. C← choose a data object randomly from D
2. for c < = k do
   - **2.1.** For each data object x Compute D(x)
   - **2.2.** Choose new centroid c with probability c_i =x∈X with probability p_i Where $D(x')^2 / \Sigma\, x \in X\, D(x)^2$.
   - **2.3.** End for
3. For x ∈ M do
   - **3.1.** Get cached center for this x c ← d[x];
   - **3.2.** Update per-center counts v[c] ← v[c] + 1;
   - **3.3.** Get per-center learning rate η ← 1 / v[c];
   - **3.4.** Take gradient step c ← (1 − η)c + ηx;
   - **3.5.** End for
4. End for

End

Mini batch gradient descent is similar to gradient descent algorithm that splits the initial data set into sub sets which called batchs, which are used to calculate model error and update model coefficients.

Implementations may choose to sum the gradient over the mini-batch or take the average of the gradient which further reduces the variance of the gradient. Mini-batch gradient descent seeks to find a balance between the robustness of stochastic gradient descent and the efficiency of batch gradient descent. It is the most common implementation of gradient descent used in the field of deep learning. Mini-batch sizes, commonly called "batch sizes" for brevity, are often tuned to an aspect of the computational architecture on which the implementation is being executed.

Advantages of Hybrid Clustering Algorithm:

- ✓ This Hybrid algorithm reduce the computation time, while still attempting to optimize the same objective function.

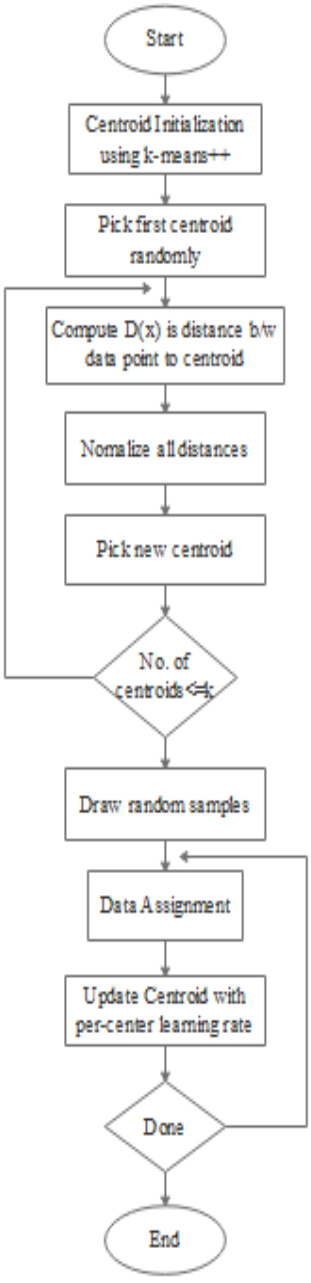

Fig.3 Flow chart for

- ✓ It can solve the well-known clustering problem.
- ✓ It gives accurate results when we apply on big data.
- ✓ Applying k-means++ we can get good centroids.
- ✓ Using Mini-batch Gradient Descent instead of batch gradient and Stochastic Gradient Descent it gives quality better than SGD k-means.

V. EXPERIMENTAL EVOLUTION

We experimented hybrid clustering algorithm which combines the Mini-batch k-means and k-means++ using python 3.6 on Anaconda navigator by using spyder. The trials were performed on an Intel Core 2 Duo Processor and

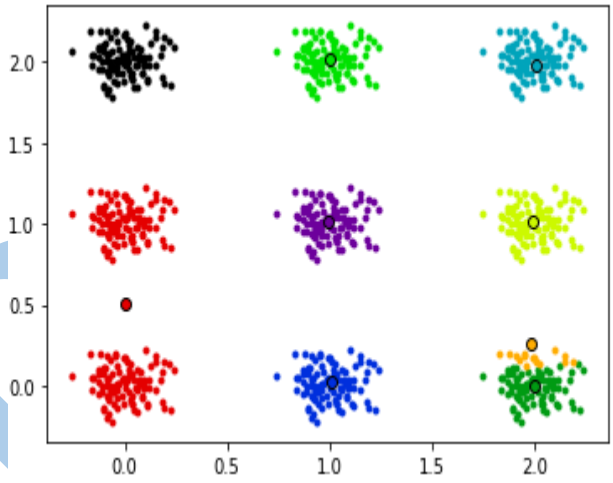

Fig.4 Mini-batch k-means with k-means++

4 GB RAM running on the stage Microsoft Windows 8. Recall that the k-means++ augments the k-means algorithm by choosing the initial cluster centroids according to the D2 metric, and not uniformly at random from the data. Overall, the new seeding method yields a much better performing algorithm, and consistently finds a better clustering with a lower potential than k-means.

We compare this hybrid clustering to other algorithms which traditional k-means, hybrid algorithm which is k-means with k-means++ and traditional Mini-batch k-means. Here we observed

This algorithms takes more time to execute. This hybrid clustering algorithm gives accurate results when we apply on large data sets. Here we used k-means++ for initializing the centroids so it gives good centroids for mini-batch k-means. In Mini-batch k-means we were use Gradient Descent for updating the centroid using this Gradient Descent we can get better quality clusters than algorithms. This hybrid clustering algorithm takes less time to execute compare to other clustering algorithms.

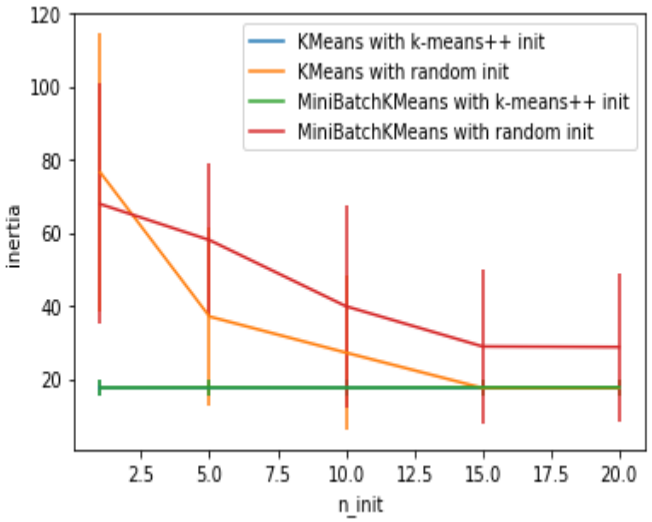

Fig.5 Comparison of existing algorithms and proposed algorithm

VI. CONCLUSION

In this paper we proposed a new Hybrid clustering algorithm which combines Mini-batch k-means and k-means++ clustering algorithms. This algorithm has two phases. First phase is to choose data object as first centroid then compute D(x) is distance between data object to centroid that has been already chosen. Now we would take new centroid using weighted probability distribution. Repeat this procedure until we get k centroids. Second phase is to draw the random samples from data set and assigned to nearest centroid. Update centroids using Gradient Descent. This algorithm gives accurate results when we apply on large data sets. Using k-means++ we can get good centroids. This hybrid clustering algorithm reduces computational time.

REFERENCES

[1] Raghavan, Ramprasad, and Darshika G. Perera. "A fast and scalable FPGA-based parallel processing architecture for K-means clustering". "Communications, Computers and Signal Processing (PACRIM), 2017 IEEE Pacific Rim Conference on. IEEE, 2017.

[2] Meghana m chavan1, asawari patil, et.al "Mini Batch K-Means Clustering On Large Dataset" IJSETR, ISSN 2319-8885 Vol.04, Issue.07, March-2015, Pages:1356-1358.

[3] Liu, Chuan, et al. "Improved K-means algorithm based on hybrid rice optimization algorithm." Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2017 9th IEEE International Conference on. Vol. 2. IEEE, 2017.

[4] Xiong, Caiquan, et al. "An Improved K-means text clustering algorithm By Optimizing initial cluster centers." Cloud Computing and Big Data (CCBD), 2016 7th International Conference on. IEEE, 2016.

[5] Han, Jungkyu, and Min Luo. "Bootstrapping K-means for big data analysis." Big Data (Big Data), 2014 IEEE International Conference on. IEEE, 2014.

[6] Béjar Alonso, Javier. "K-means vs Mini Batch K-means: A comparison." (2013).

[7] Cho, Hyuk, and Min Kyung An. "Co-Clustering Algorithm: Batch, Mini-Batch, and Online." International Journal of Information and Electronics Engineering 4.5 (2014): 340.

[8] Wang, Juntao, and Xiaolong Su. "An improved K-Means clustering algorithm." Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on. IEEE, 2011.

[9] Aubaidan, Bashar, et al. "Comparative study of k-means and k-means++ clustering algorithms on crime domain." Journal of Computer Science 10.7 (2014): 1197-1206.

[10] Arthur, David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding." Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics, 2007.

[11] Aditi Anand Shetkar ,"Text Categorization of Documents using K-Means and K-Means++ Clustering Algorithm", International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 4 Issue: 6.

[12] Yadav, Mr Krishna, and Mr Jwalant Baria. "Mini-Batch K-Means Clustering Using Map-Reduce in Hadoop." International Journal of Computer Science and Information Technology Research 2.2 (2014): 336-342.

[13] Kanungo, Tapas, et al. "An efficient k-means clustering algorithm: Analysis and implementation." IEEE transactions on pattern analysis and machine intelligence 24.7 (2002): 881-892.

[14] Bradley, Paul S, Usama M. Fayyad, and Cory Reina. "Scaling Clustering Algorithms to Large Databases." KDD. 1998.

[15] Sculley, David. "Web-scale k-means clustering." Proceedings of the 19th international conference on World Wide Web. ACM, 2010.

[16] Agarwal, Manu, Ragesh Jaiswal "k-means++ under Approximation Stability." Theoretical Computer Science 588 (2015): 37-51.

[17] Arthur, David, and Sergei Vassilvitskii. "Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method." Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on. IEEE, 2006.

[18] Capó, Marco, Aritz P érez, and Jose A. Lozano. "An efficient K-means clustering algorithm for massive data." arXiv preprint arXiv:1801.02949 (2018).

[19] Karimov, Jeyhun, and Murat Ozbayoglu. "Clustering quality improvement of k-means using a hybrid evolutionary model." Procedia Computer Science 61 (2015): 38-45.

[20] Gokhale, M., J. Frigo, K. Mccabe, J. Theiler and L. Dominique,"Early Experience with a Hybrid Processor: K-Means Clustering," in1st Int. Conf. On Engineering of Reconfigurable Systems and Algorithms, 2001

