On the Solutions of The Big Data Timeliness Problem

ABSTRACT

Big Data is increasingly used on almost the entire planet, both online and offline. It is not related only to computers. It makes a new trend in the decision-making process and the analysis of this data will predict the results based on the explored knowledge of big data using Clustering algorithms. The response time of performance and speed presents an important challenge to classify this monstrous data. K-means and big k-mean algorithms solve this problem. In this paper, researcher find the best K value using the elbow method, then use two ways in the first sequential processing and the second is parallel processing, then apply the K-mean algorithm and the big K-mean on shared memory to make a comparative study find which one is the best in different data sizes. The analysis performed by R studio environment.

Keywords: Big Data, k means, big k means, sequential processing, parallel processing

1 INTRODUCTION

Sahu, S., and Y. Dhote[1] the authors talk about Big Data, it is now used almost everywhere in our daily life. Big Data is a popular term used to describe a huge volume of data, which is so large that is difficult to store and process with traditional database management systems. Nowadays terabytes or petabytes of data are growing exponentially into organizations. Thus, big data refers to data sets whose volume is beyond the ability of most current hardware and software technologies to be managed and processed within a reasonable response time. So, the flip side of size is speed. The time taken for analyzing the data is proportional to the size of the data set to be processed. The system design that deals with the massive volume of the data will also result in a system that can process a specific size of data set faster. Recently, Rehioui, H., et al [2] the authors propose The informational revolution has generated more terabytes of heterogeneous data every day. According to an investigation made by the institute IDC1, in the digital world 1.8 zettabyte data was created in 2011, 2.8 zettabytes in 2012 and it will increase up to 40 zettabytes in 2020 and more. This large quantity of complex data, which can be part of Big data, needs a more developed technology to better store, use and analyze. There are several propositions to define Big Data. It can be defined with the following properties associated with it, but these five V's are extended as seven V's later by adding two more V's such as value and Complexity They are following (Volume- Velocity- Variety- Variability- Veracity- Value- Complexity).

According to O. Kurasova, V. Marcinkevicius, V. Medvedev, A. Rapecka, and P. Stefanovic [3] as proposed the Cluster analysis is an unsupervised way to gain data insight into the world of Big Data. It will show you relationships in data that you may not realize are there. When dealing with big data, a data clustering problem is one of the most important issues. Often data sets, especially big data sets, consist of some groups (clusters) and it is necessary to find the groups. Thus, W. M. Rand[4] the author propose the

Data Clustering the similarity between objects in the same cluster (intra-class) must be small and large between the different data clusters (inter-class). This similarity is considered as a distance measure. Mathematically, the final goal of data clustering is to partition a set of unlabeled objects O = {o1, o2, ..., on} into k clusters. Each object is characterized by a feature vector X = {x1, x2, ..., xn}, where n is its dimension[5].

Yuan, C. and H. Yang [6] the authors propose four kinds of choosing K-value, like Elbow Method, Gap Statistic, Silhouette Coefficient, and Canopy. each of the four algorithms has its own characteristics. For the clustering of small data sets, the four techniques mentioned in the paper can meet the requirements and that, for large and complex data sets, it is obvious that the Canopy algorithm is a better selection. Next, we will use the real-world multidimensional data containing complex information fields for experimental verification to deeply explore the advantages and disadvantages of each algorithm or to improve the performance of the algorithm. The clustering algorithms used in this paper k-mean and parallel big k-means. the researchers used two way to obtain the best value of k and apply the k-means clustering, the first way is sequential processing and the second way is parallel processing after that identify the optimal value of k to use in cluster technique.

The rest of this paper is organized into five sections: initially, we focus on the introduction and related work of paper in the first section. The second section discusses the selection of bigdata clustering algorithms. In the third section, discuss the methodology of the proposed model. Results and dialogues of the experimental works are assumed in the fourth section, whereas the fifth section introduces the conclusion and recommended future work.

2 SELECTION OF BIG DATA CLUSTERING ALGORITHMS

The selected machine learning algorithms depend on the dataset. Accordingly, a comparison was made between the two different clustering algorithms, such as K-means and Big k-means, in order to evaluate the performance of the algorithms by Sum of Squared Errors (SSE) and computational time[7].

2.1 Determine the K-Value

For the K-means clustering algorithm, the choice of the optimal number of clusters depends on the setting of determining the K value. In practice, the K value is generally difficult to define Specifically in big data so that, the choice of K value directly determines the data cluster that needs to be clustered into multiple clusters. At the beginning of the algorithm, the researchers use the “shooting the head” method to determine the K value, which is estimated to later give many improvements proposed for optimization algorithms. This paper mainly shows the method of K-value choice with specific representativeness and gives more analysis and experimental verification.[6].

2.1.1 The Elbow Method Algorithm

The main idea of the elbow method rule is to use a square of the distance between the sample points in each cluster and the centroid of the cluster to give a series of K values. The sum of squared errors (SSE) is used as a performance pointer. Iterate across the K-value and calculate the SSE. Smaller values indicate that each cluster is more convergent. When the number of clusters is set to approach the number of valid clusters, SSE shows a rapid decline. When the number of clusters exceeds the number of real clusters, SSE will continue to decrease but it will quickly become slower [6]. The pseudo-code of the algorithm is as follows:

Table 1. pseudo-code of the algorithm elbow method

```
Algorithm 1: Elbow Method
Input: traffic = datasets. load_ traffic (), X = traffic. Data [: 10:]
Output: d, k
 d = [];                                          [1]
for k = 10, k in rang (10, 30) do                 [2]
        $SSE = \sum_{i=1}^{k} \sum_{X \in c_i} dist^2 \ (C_i, X)$      [3]
 return d, k;                                     [4]
```

2.2 the basic k-mean clustering algorithm

K-means is one of the known clustering methods due to its simplicity. It is based on the concept of centroids which are used to define clusters in this work. It partitions a given data set into k clusters using the distance from each data point to k different centroids (or means). The term “k-means” was first used by (MacQueen, 1967). Even though the k-means is efficient, it may converge to local minima producing counterintuitive results, mainly due to the randomness in its initialization. Also, it has a very flexible control of cluster sizes. The name comes from representing each of the k clusters Cj by the weighted mean Cj of its data points, the so-called centroid. While this representation does not work well with categorical attributes, it makes good sense from a statistical perspective for numerical attributes. The sum of distances between elements of a set of data points and their centroid expressed through an appropriate distance function is used as an objective function[8].

2.3 Basic K-mean algorithm

In very general terms, the K-Means algorithm aims to divide a set of observations into k clusters so that each observation belongs to the cluster that has the closest average. Simply find the k different groups that have the maximum dissimilarity[9]. Since the mean is used as a measure to estimate the centroid, it is not free from the presence of extreme outlier data. Therefore, it is necessary to verify the presence of outliers in a data set before executing the k-means clustering. We can the scale method of identifying the presence of any outliers in the dataset. In the simplest form of the algorithm, it has two steps:

I. **Assignment.** Assign each observation to the cluster that provides the minimum within-cluster sum of squares (WCSS).
II. **Update.** Update the centroid by taking the mean of all the observations in the cluster.

These two steps are iteratively executed until the assignments in any two consecutive iterations don't change, meaning either a point of local or global optima (not always guaranteed) is reached. The main idea of the basic k-means clustering algorithm is to classify a given dataset in value of k number of disjoint clusters, where the value of k is fixed in advance. The above algorithm consists of two separate phases: the first phase is to define k centroids, one for each cluster. The next phase is to take each point that belongs to the given datasets and establish and associate it with the nearest centroid. Euclidean distance is usually considered to determine the distance between data points and centroids. When all points are included in some clusters, the first step is completed and early grouping is done. At this point, we need to recalculate the new centroids, since the addition of new points can lead to a change in the centroids of the clusters. Once we find new centroids, a new link will be created between the same data points and the new nearest centroid, generating a loop. As a result of this loop, the centroids can change their position step by step. Finally, a situation will be reached in which the centroids no longer move. This means the convergence criterion for the clustering. The pseudocode for the k-means clustering algorithm is listed as the first Algorithm [10] is as follows.

Table 2. **pseudo-code of the k-means algorithm**

Algorithm 2: Generalized pseudocode of the traditional k-means Algorithm

```
Input: D = {d1, d2, ......, dn}     // where d1,d2,...,dn are the set of n data items.
K    // Number of desired clusters
Output: A set of k clusters.
Steps:
choose k data items from D as initial centroids randomly [1]
Repeat until no observation change [2]
Assign each item $d_i$ to the cluster which has the closest centroid [3]
Calculate the new mean for each cluster [4]
Until convergence criteria are met [5]
```

Although it is a computationally difficult problem, there are very efficient implementations to quickly find the local optimum. In an optimization problem, the optimum is the value that maximizes or minimizes the condition that we are looking for. Given a set of observations $X = \{x_{1\ ,}\ x_2 , \dots\dots\dots ,\ x_k\}$, K-Means clustering aims to partition the N observations into $K\ (\leq N)$ sets $S = \{S_{1\ ,}\ S_2 , \dots\dots\dots ,\ S_k\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$j = \sum_{j=1}^{k} \underbrace{\left\| x_i^{(j)} - c_j \right\|}_{\text{Distance Function}}{}^{2} \qquad \text{Eq. (1)}$$

J is an objective function and k is a number of clusters, n is a number of cases, $X_i^{(j)}$ X is a case i and C_j is a centroid for cluster j.

2.4 Big K-means clustering algorithm with shared memory

Big k means (x, centers), where x is a numerical data set (large data matrix object), and the centers are the number of clusters to extracted and determine the best k. The big k means returns the cluster memberships, centroids, within-cluster sum of squares (WCSS), and cluster sizes. The big k means also works with ordinary matrix objects, offering a faster calculation than the k means clustering[11]. hence, using the shared memory to create, store, access, and manipulate massive data matrices. Matrices, by default, are allocated to shared memory and can use mapped files assigned to memory. The use of these packages in parallel environments can provide substantial speed and memory efficiencies. Big Memory also provides a C ++ framework for the development of new tools that can work with both large matrix and native matrix objects. Multi-gigabyte data sets the challenge and frustrates users, even in well-equipped hardware. The use of C/C++ can provide efficiencies, but is cumbersome for interactive data analysis and lacks the flexibility and power of 's rich statistical programming environment. The big memory and associated big analytics, synchronicity, big tabulate, and big algebra bridge this gap, implementing massive matrices and supporting their manipulation and exploration. The data structures may be allocated to shared memory, allowing Distributed processes on the same computer to share access to a single copy of the data set. The data structures may also be file-supported, allowing users to easily manage and analyze data sets larger than available RAM and share them across nodes of a cluster. These features of the Big memory open the way for powerful and memory-efficient parallel analyses and data mining of massive data sets[12].

3 PROPOSED MODEL

The main objective of this study is to make a fair judgment between different clustering techniques in different sizes of the data set. This paper used standard K-means and Big k-mean. The study advanced in certain phases described in the experimentation framework as shown in figure 1. The framework includes many phases: data collection, preprocessing and data selection, transformation phase, selection of big data tools, selection of programming language and selection of big data clustering algorithms.

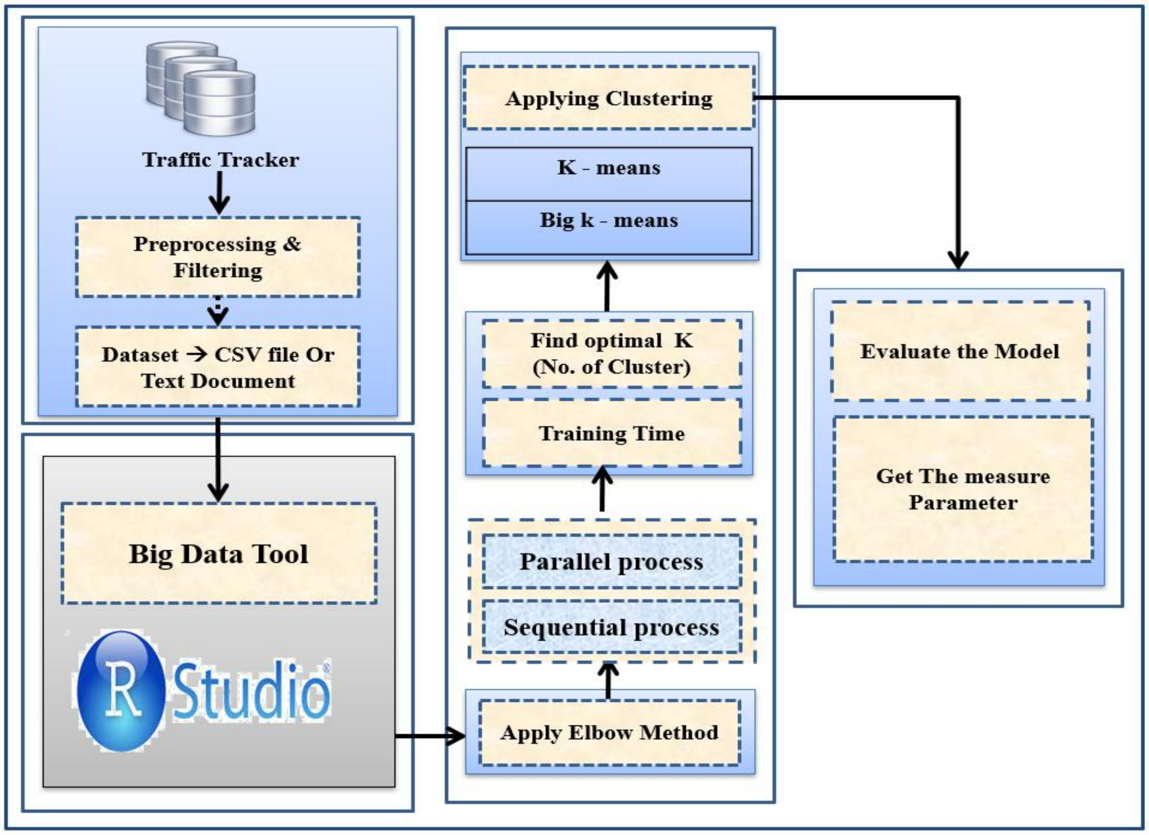

figure. 1. The conceptual framework

3.1 Data Collection and Selection Phase

In this experiment, the data set contains the estimated historical congestion for more than 1,000 traffic segments, beginning approximately in March 2018. The Chicago Traffic Tracker estimates traffic congestion on Chicago's arterial streets (streets without highways) in real-time by continuously monitoring and analyzing the GPS tracks received from the buses of the Chicago Transit Authority (CTA). There are two types of congestion estimates every 10 minutes: 1) by traffic segments and 2) by regions or traffic zones. Traffic congestion estimates give the speed typically observed for half a mile of a street in a traffic direction. Traffic Segment level congestion is available for approximately 300 miles of major arteries. The data set contains 78.1 million records, 22 characteristics, and each row is a segment time segment.[13].

3.2 Data Preprocessing and Transformation Phase

In this step, the dataset goes through many stages, such as data cleaning, data integration, and data transformation. The gathered data was saved as text documents. The cleaning process is required in order to analyze the data based on selected cluster algorithms, in which data with missing values are eliminated, inconsistent data is corrected, outliers are identified, and duplicate data is removed. The data was exemplified by numbers and stored in the form of a CSV file so it can be introduced to the data mining tool.

The dataset goes through many stages:

I. Convert the data to numeric.

II. Remove the NAN and missing value.

III. **Data Normalization:** The point of normalization is to change their observations so that they can be described as a normal distribution, also known as the bell curve, is a specific statistical distribution where an approximately equal observation falls above and below the average, the mean and median are the same, and there are more observations closer to the mean.

$$x' = \frac{x - x_{mean}}{x_{max} - x_{min}} \quad \text{Eq. (2)}$$

where x' is the original feature vector, x_{mean} is the mean of that feature vector, and σ is its standard deviation. For normalization, the maximum value you can get after applying the formula is 1, and the minimum value is 0. So, all the values will be between 0 and 1.

IV. **Data scale:** An alternative approach to the normalization (or standardization) of the Z score is the so-called Min-Max scaling (often also simply called "normalization", a common cause of ambiguities). In this approach, the data is scaled to a fixed range. generally, from 0 to 1. The cost of having this limited range, in contrast to standardization, is that we will end up with smaller standard deviations, which can suppress the effect of outlier's data. Thus, if there is a need for outliers to get weighted more than the other values, the z-score standardization technique suits better A Min-Max scaling is typically done via the following equation:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \text{Eq. (3)}$$

V. Convert data set to Data Matrix.

3.3 Evaluation

The comparison of the different Clustering algorithms mentioned in Section 3.2 is based on the following measured parameters [14]:

I. **Training Time**: As an accomplished machine learning algorithm are measured. Time taken to build the model is called training time. This varies on the implementations of the algorithms.

II. **Sum of Squared Error (SSE)**: It is the variance between the estimator and what is estimated. It is a risk function, consistent with the estimated value of the squared error loss or quadratic loss.

From the previously mentioned measured parameters, we can compare the SSE provided by all the algorithms on a dataset. Here, the focus is mainly on comparing major parameters like SSE and training time in order to decide which Clustering is better suited for a selected type of data.

4 EXPERIMENTAL WORK

the researcher applies both the algorithms original and proposed for the different number of records. Both the algorithms original and proposed need number of clusters as an input.in this paper, the researcher defines mainly a method for determining the optimal number of clusters for k-means. In the basic K-means clustering algorithm, a set of initial centroids is required. The proposed method finds the initial centroids systematically. The proposed method requires only the data sets and the number of clusters as inputs. The basic K-means clustering algorithm is executed more than times for the different data sets values of the initial centroids. In each experiment, time was calculated and the average time of all experiments was taken. Table 4 shows the performance comparison of the Basic and proposed k-mean clustering algorithms. The experiment results show that the proposed algorithm is producing better results in fewer amounts of computational time with small SSE compared to the basic k-means algorithm.

The experimental simulation environment is DELL T5600 CACHE 40MB 32 CORE Threads, Intel Xeon E5-2660, 20M Cache, 2.20 GHz, 8.00 GT/s, 2 processor, Max Turbo Frequency 3.00GHz, 48G memory, 500G hard disk space. The experiment used the Chicago data portal Machine Learning Repository machine to learn the Chicago Traffic Tracker estimates traffic data set in the data set (it is explained in section 4).

4.1 Determining the Optimal Number of Clusters

The research experiments many techniques to determine K value, such as the elbow method, the average silhouette method, and the Gap statistical method. Memory size does not fit all algorithms, it just fits the elbow method. The elbow method is a technique to choose the best value of k. This method uses homogeneity within the group or heterogeneity within the group to evaluate variability. In other words, the research is interested in the percentage of the variance explained by each group. You can expect the variability to increase with the number of clusters, alternatively, the heterogeneity decreases. Our challenge is to find the k value that is beyond diminishing returns. we find this point using the heterogeneity measure. The Total within clusters sum of squares (WSS) is the tot. wittiness in the list return by k-means. The result of the technique can show that in figure [2].

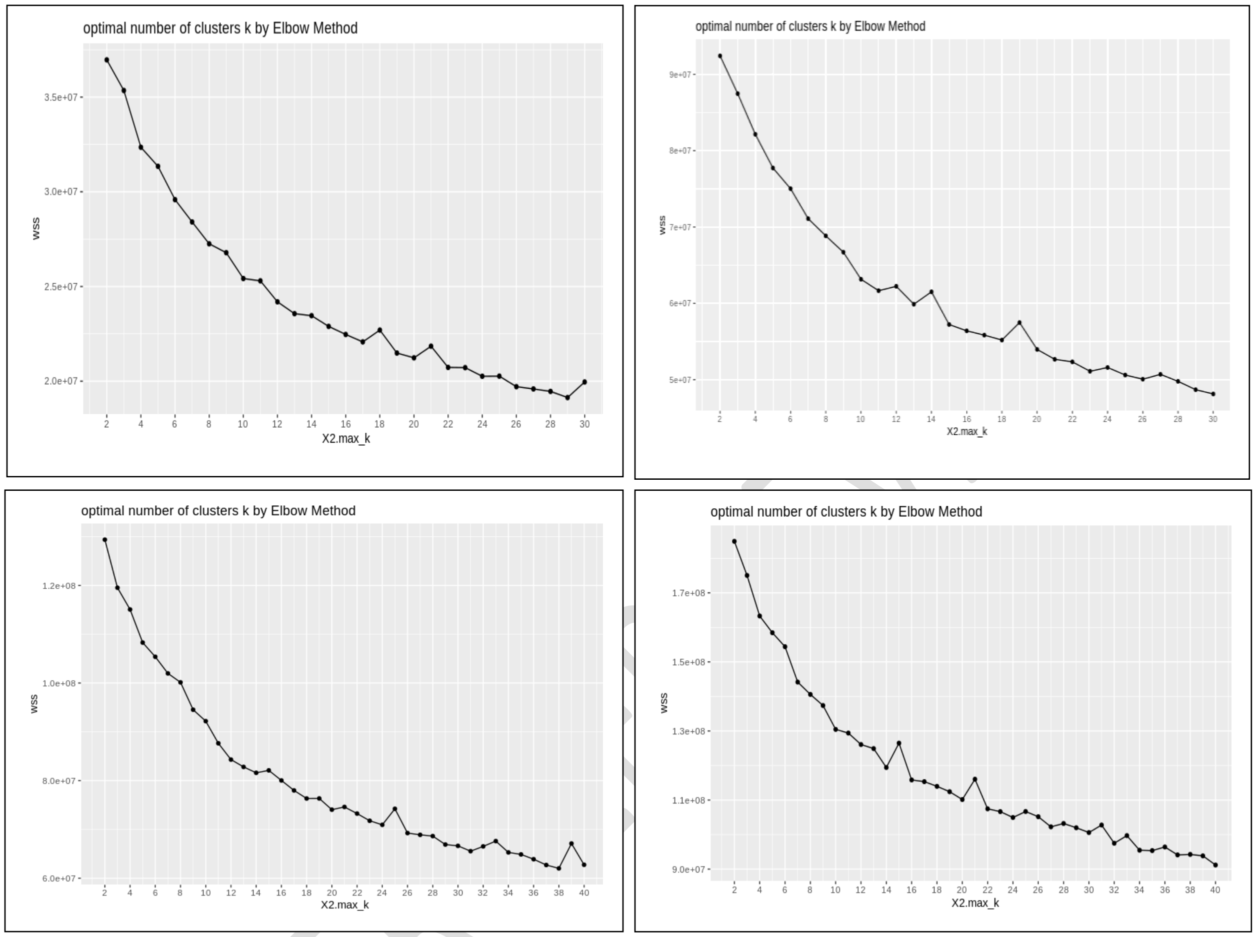

Figure. 2.: Plot the within cluster sums of squares versus the number of clusters according to the size of the dataset extracted

No. of Records	Rang of No. Of Cluster	system. Time (elbow) in Sec			system. Time (parallel elbow) in Sec		
		user	**system**	**elapsed**	**user**	**system**	**elapsed**
2000000	10, 11, 14	321.655	19.218	340.81	161.843	11.817	174.986
5000000	11	1353.627	83.364	1436.8	687.285	44.591	963.147
7000000	14, 12	1894.274	119.891	2013.791	950.333	62.01	1024.687
10000000	14	2724.464	199.675	2924.005	2199.837	190.874	1547.724

Table 3. Comparing runtimes between Elbow method and Parallel Elbow method

In the table 3, the results show that the execution is done on a single processor and parallel processing for different data size to measurement time token and chose the optimal value of k, where the curve begins to have a diminishing yield as shown as in above Figure 2. The Elbow Method algorithm uses SSE as a performance metric, traverses the K value, finds the inflection point, and has a simple complexity. The inadequacy is that the inflection point depends on the relationship between the K value and the distance value. If the inflection point is not obvious, the K value cannot be determined. Once the researchers have the optimal k, they can now rerun the algorithm and evaluate the clusters. figure 3 shows the training time between the Elbow method and parallel elbow.

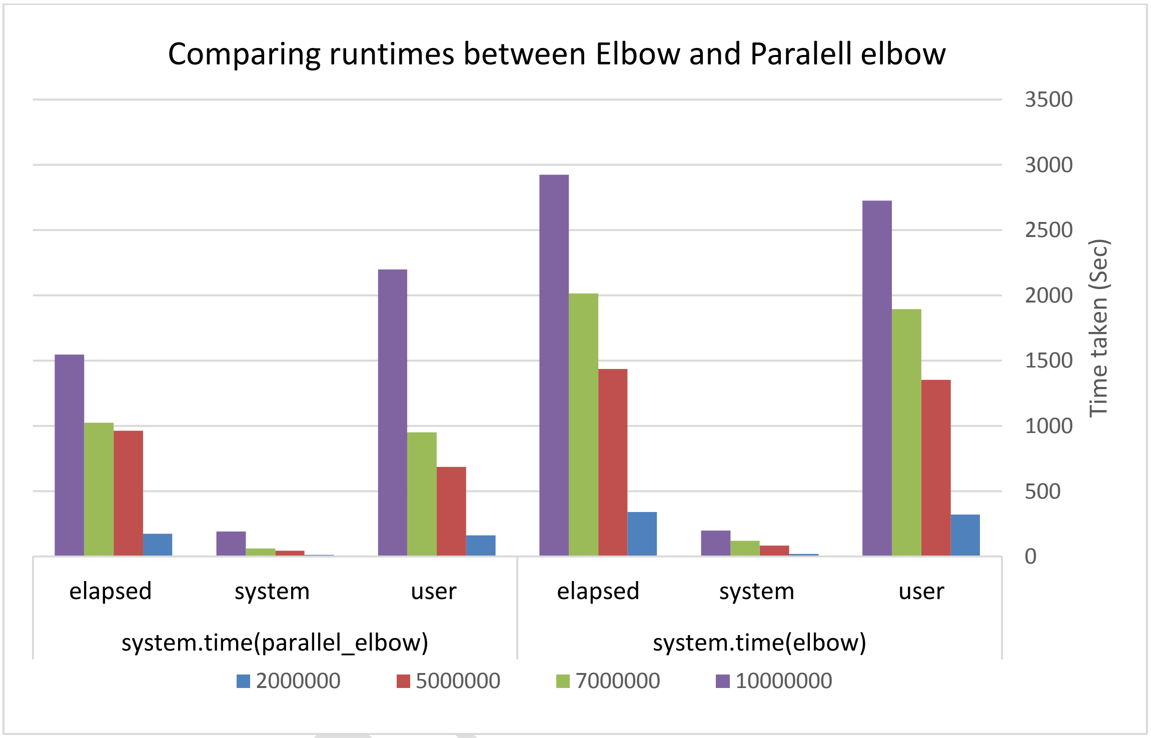

Figure .3: training time between Elbow method and parallel Elbow Method

4.2 Comparison of k-means and big k-means

In the table 4. apply k means and big k means clustering in the different data size with rang of cluster and number of iterations, each run shows the SSE and training time, however, we can see two such points in the above graph figure 2 one at 10,11 and another at 14 we go for the one with small SSE And less time Execution, so k=10 is more accurate. Thus, we select k as 10 using the elbow method And Big k Means. in figure 4 Calculating the average execution time of the k means and big k means takes substantial time. The preceding figure, however, reveals that big k means works more efficiently with larger datasets than the k means, thus reducing the calculation time of R in the analysis.

No. of Records	No. Of Cluster	No. Of iteration	Some square error (SSE)	K-means Execution time in mins	Big k-means Execution time in mins
2000000	10	7	42.6%	7.059539	2.098157
	11	7	46.4%	9.094598	2.312489
	14	10	47.4 %	8.461808	3.802602

5000000	11	6	44.0%	20.6393	5.71434
7000000	14	4	47.1%	35.74134	6.67423
	12	6	45.3%	29.5375	7.8523
10000000	14	5	45.6%	56.25175	11.21251

Table .4: Performance Comparison and Evaluation of clustering algorithms

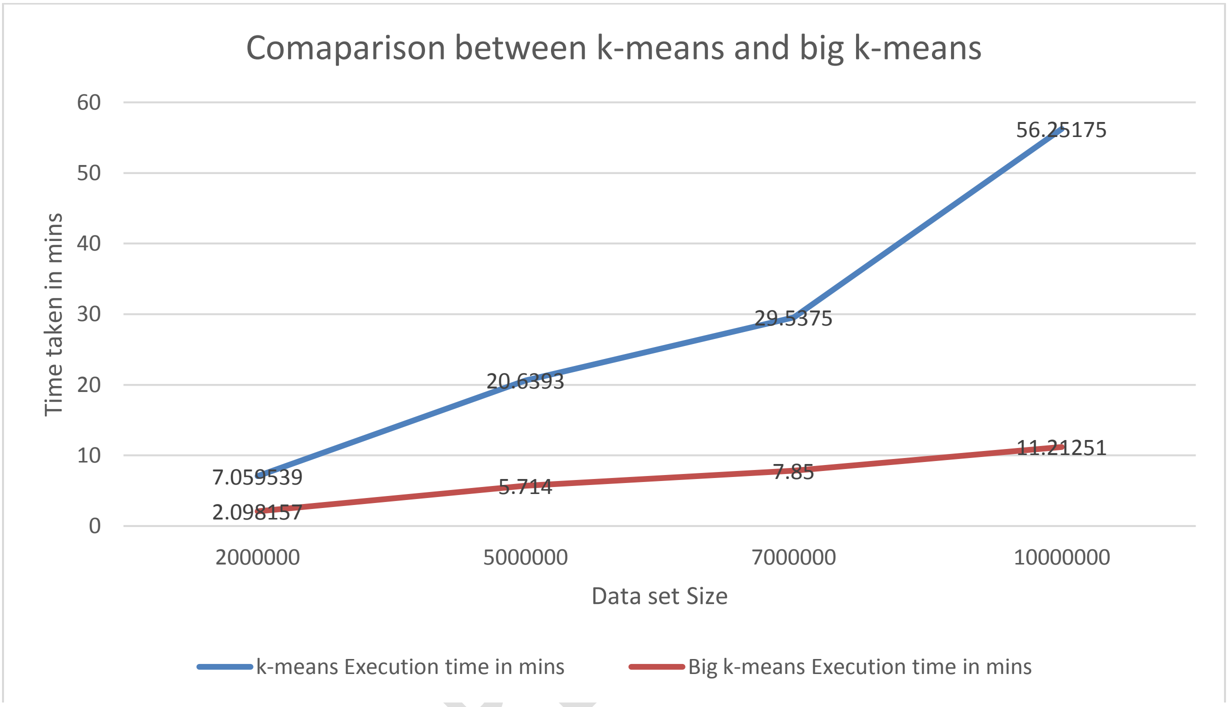

Figure .4: Execution time of the k means and big K means according to the size of the dataset

The result of the clustering depends highly on the initial centers in the K-Means clustering method. The response time of performance and speed presents an important challenge to classify this monstrous data. In this paper, the reduce time with a large amount of data resolved in two ways in the first one obtains the best value of K using the elbow method, using sequential processing and parallel processing. The result in this part shows that parallel processing is better than the sequential in reduced time by 75 %. In a second way, apply the K-mean algorithm and the big K-mean in the shared memory and found that the big K-mean is the best to reduce time by 80 % less than the K-mean. In future work, researchers will work to improve performance through the use of rough K-means and different clustering techniques. The main point in the future minimizes the total number of the cluster by using constrain K-means and integer linear programming.

REFERENCES

1.Sahu, S. and Y. Dhote, *A Study on Big Data: Issues, Challenges and Applications.* International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), 2016. **4**(6): p. 10611-10616.

2.Rehioui, H., et al., *DENCLUE-IM: A new approach for big data clustering.* Procedia Computer Science, 2016. **83**: p. 560-567.

3.Kurasova, O., et al. *Strategies for big data clustering*. in *2014 IEEE 26th international conference on tools with artificial intelligence*. 2014. IEEE.

4.Rand, W.M., *Objective criteria for the evaluation of clustering methods.* Journal of the American Statistical association, 1971. **66**(336): p. 846-850.

5.Pal, K., et al., *Relational mountain (density) clustering method and web log analysis.* International journal of intelligent systems, 2005. **20**(3): p. 375-392.

6.Yuan, C. and H. Yang, *Research on K-Value Selection Method of K-Means Clustering Algorithm.* J, 2019. **2**(2): p. 226-235.

7.Wagstaff, K., et al. *Constrained k-means clustering with background knowledge*. in *Icml*. 2001.

8.Kanungo, T., et al., *An efficient k-means clustering algorithm: Analysis and implementation.* IEEE Transactions on Pattern Analysis & Machine Intelligence, 2002(7): p. 881-892.

9.Jain, A.K., M.N. Murty, and P.J. Flynn, *Data clustering: a review.* ACM computing surveys (CSUR), 1999. **31**(3): p. 264-323.

10.Han, J., J. Pei, and M. Kamber, *Data mining: concepts and techniques*2011: Elsevier.

11.Maia, R., et al., *pavo: an R package for the analysis, visualization and organization of spectral data.* Methods in Ecology and Evolution, 2013. **4**(10): p. 906-913.

12.Kane, M.J., J.W. Emerson, and P. Haverty, *bigmemory: Manage massive matrices with shared memory and memory-mapped files.* R package version, 2010. **4**(3).

13.*Chicago Traffic Tracker - Historical Congestion Estimates by Segment - 2018-Current*. 2018 [cited 2019 2019]; Available from: https://data.cityofchicago.org/Transportation/Chicago-Traffic-TrackerHistoricalCongestionEsti/sxs8h27x?fbclid=IwAR2CwmZewIvsUIq0cMsJRbCQto2QNqNHRPYs31vzZpxMjTGjrFIqRHkQ9q8.