IJCSE International Journal of Computer Sciences and Engineering Open Access

Survey Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Survey of Clustering Methods for Large Scale Dataset

Anupama Jawale[1], Ganesh Magar[2]

[1] Narsee Monjee College of Commerce & Economics, University of Mumbai, Mumbai, India
[2] PG Department of Computer Science, Shreemati Nathibai Damodar Thackersey Women's University, Mumbai, India

DOI: https://doi.org/10.26438/ijcse/v7i5.13381344 | Available online at: www.ijcseonline.org

Accepted: 19/May/2019, Published: 31/May/2019

Abstract- This research study focuses on a comparative study of various clustering algorithms for the performance evaluation of large datasets. Analysis of large datasets is required for effective knowledge discovery. Use of data mining, machine learning techniques are often being used to refine of larger datasets. Traditional approach of processing of large datasets is inefficient and needs to consider the fast processing parallel environment to enhance the performance. This study has emphasis on four clustering algorithms, K-Means, Wards, PAM and CLARA to study performance on larger dataset of GeoJson format and CSV formats. Statistical techniques Medoid and Centroid are used for experimental work with different sample sizes to measure the performance of algorithms. Experimental work is carried out using R programming on Azure cloud for parallel computing with HDInsight Cluster. This research study provide evidence that the algorithm CLARA shows constant Medoid computations for different sample sizes compare to algorithm PAM and K-,Means. Silhouette widths of the algorithms CLARA (0.41) and Silhouette width of PAM (0.36) indicates well defined clusters are present in CLARA.
Performance of these algorithms is effectively enhanced by reducing the time of DBSCAN by 45.72%, K-means by 99.95% and CLARA by 99.96% in comparison with Ward's Algorithm for larger datasets using parallel processing environment.

Index Terms- Azure, CLARA, Clustering Algorithms, GeoJson dataset, PAM, R Studio, Ward's Method.

I. INTRODUCTION

In the field of data sciences, clustering of data is always an essential aspect. Clustering refers to the technique of grouping data points into one or more clusters. Unsupervised learning method is mostly used for statistical analysis [1].
The idea behind grouping is, all the elements belonging to a group should exhibit similar properties, where elements belonging to different groups exhibit clear, distinguishable features that separate them from each other. Such elements should possess dissimilar characteristics. Computation of medoid / centroid is one of the key roles of a clustering algorithm. Centroids are arithmetic mean of the observations whereas medoids are member of a cluster and their average dissimilarity with other members of that cluster are minimal. Medoids are similar to means or centroids, but they are always linked to the member dataset.
Clustering algorithms are divided into following four categories based on the logic of clustering of objects.

i. *Connectivity-based clustering* (hierarchical clustering - Wards)
ii. *Centroid-based clustering* (K-Means, K-Medoid)
iii. *Distribution-based clustering* (GMM)
iv. *Density-based clustering* (DBSCAN, OPTICS).

This research paper focuses on some of the popular clustering algorithms and their implementation on a large size dataset.

It is been observed by past researchers that traditional clustering algorithms, designed for limited size data, perform very poorly on large scale data [2]. Here we are trying to implement and compare below given algorithm for their performance and their time complexity based on parallel computing environment, using Microsoft Azure [2].
Section I gives brief introduction of clustering algorithms used for this study. Section III presents various methodologies and past research work done for clustering algorithms. Section IV describes experimental work and outcome of the survey. Section V describes tabular and chart description of the experimental outcome whereas conclusion is covered in section VI.

II. METHODOLOGY REVIEW

Clustering Algorithms considered for this paper are listed below.

A. DBSCAN

DBSCAN stands for - Density-Based Spatial Clustering of Applications with Noise. : Algorithm starts with random unvisited data point. All data points near to this point are classified as neighborhood points. Nearness is defined by a specific distance variable, termed as 'epsilon'. Another variable to define minimum number of points called MinPts is also required for the algorithm. To start the clustering process, MinPts number of data points are considered. The first data point will be considered as member of cluster or noise. In any case, this data point will be termed as visited. All data points with the distance epsilon will be marked this way and the process goes on until all data points are labelled within the distance epsilon. On completion of this process, algorithm starts with a new unvisited point leading to find a new cluster or a noise data point. This algorithm works towards identifying the dense region in the data.

Advantages: Pre definition of number of clusters is not required in DBSCAN

Limitations: This algorithm does not work well for clusters of varying densities. Also, defining epsilon and MinPts becomes difficult with large amount of data.

B. Wards Algorithm

Wards Algorithms: Wards algorithm looks at hierarchical cluster analysis as an analysis of variance problem, instead of using distance metrics or measures of association. This algorithm starts with all data items, clustered individually. Initially in the algorithm, n - 1 clusters are formed, one of size 2 and the remaining of size 1. sample units are combined into a single large cluster of size n. Conceptually the method uses I) Error Sum of Squares or ii) Total Sum of Squares or iii) R-Square method to determine whether the given element belongs to its assigned cluster or not[1, 2].

Error Sum of Squares: The ESS is the sum of the squares of the differences of the predicted values and the mean value of the response variable

$$ESS= \sum i \sum j \sum k \left|X_{ijk}-\overline{x_{ik}}\right|^2 \qquad (1)$$

where i, j, k =1, 2, 3.....n are the dimensions of data

Total Sum of Squares: The TSS is the sum of the squares of the differences of the actual values and the mean value of the response variable

$$TSS= \sum i \sum j \sum k \left|X_{ijk}-\overline{x}_k\right|^2 \qquad (2)$$

where i, j, k =1, 2, 3.....n are the dimensions of data

R-Square: The coefficient of determination $R^2$is the ratio of ESS to TSS

$$R^2= \frac{ESS}{TSS} \qquad (3)$$

Distance matrix in hierarchical clustering can be computed using several methods. The different methods are single-link, complete link, average link, centroid link, and Ward's method. Single-link distance between clusters is computed as the distance between the two closest elements of the clusters. The complete-link distance between clusters is computed by the distance between the most distant elements of the clusters. The average-link distance between clusters is computed by the distance between the averages of all pairwise distances between clusters. The centroid-link distance between clusters is computed by the distance between the centroids of the two clusters. Ward's Method to define the distance between clusters is computed by the difference between the variance of the two clusters [4].

Advantages: This algorithm works on entire dataset with mathematical computations of dissimilarity among data items. It provides different methods for calculating distance matrix. Distance matrix gives clear measure of dissimilarity among data items.

Limitations: Ward's algorithm shows poor computation time performance. Major execution time and space requirement over calculation of distance matrix

C. CLARA

CLARA (Kaufmann and Rousseeuw in 1990) stands for Clustering Large Applications. It is an obvious way to cluster larger datasets. Instead of finding medoids for the entire data set it draws a small sample from the data and applies the PAM algorithm to generate an optimal set of medoids for the sample. The quality of the resulting medoids is measured by the average dissimilarity between every object in the entire data set and the medoid of its cluster [5].

If the sample is representative, the medoids of the sample should approximate the medoids of the entire dataset.

To improve the approximation, multiple samples are drawn and the best clustering is returned as the output .The clustering accuracy is measured by the average dissimilarity of all objects in the entire dataset.

Advantages: Due to sampling logic, this algorithm is ideal for large datasets [5].

Limitations: Sampling technique may be inefficient in case of largely dissimilar dataset [6].

D. PAM

PAM stands for Partition Around Medoids. The pam-algorithm is based on the search for k representative objects or medoids among the observations of the dataset. These observations should represent the structure of the data. After finding a set of k medoids, k clusters are constructed by

assigning each observation to the nearest medoids. The goal is to find 'k' number of representative objects which minimize the sum of the dissimilarities of the observations to their closest representative object [7].
Advantages: Algorithm possesses faster computations
Limitations: PAM algorithm is not suitable for large datasets

E. K-Means

This is the oldest algorithm used in data sciences. It starts with some random k number of Centroid data points. For all data points in the given data set, the nearest to centroid are checked. For finding the nearest centroid, some measurement technique, for ex Euclidian / Cosine Distance is used
Next, for every centroid, recalculation is done. New centroid is moved to the average of all data points assigned to that centroid. This process is repeated until centroid stops changing its value [5].
Advantages: K means is the simplest algorithm and it is very easy to implement. Algorithm is fast because of lesser computations
Limitation: Consistency is affected as centroids are chosen randomly [6].

Also, based on logic of the algorithm, clustering algorithms are further subdivided into categories given below

Table 1: Categories of clustering algorithm

Type of Clustering	**Algorithm**
Partition Based	k-Means, k-Medoids, PAM,CLARA
Hierarchical Clustering	Wards Method
Fuzzy Clustering	Fuzzy c-Means
Density Based Clustering	DBSCAN
Hybrid Clustering	Hierarchical k-means, HCPC

Outcomes of all these algorithms are based on the respective dataset used for experiments. For example, performance of k-means clustering is better over hierarchical clustering [9].
For the iterative clustering algorithms, limiting the iterations in bisecting K-means leads higher efficiency while maintaining clustering quality [10]
For the larger dataset computations, parallel execution is required. Cloud computing is the proven tool for the same. Many cloud services are available in the market, out of which Microsoft Azure is chosen for the experiments described in this paper. Increasing number of Hadoop clusters in the HDInsight azure Cloud increases the performance of algorithm with respect to its time complexity [11].

III. EXPERIMENTAL WORK

Four different datasets are used in the experiment.

The first dataset used for the computation time comparison in the research is the freeware dataset available on "Chicago Data Portal". Approximate size of dataset is 2.20 GB. After cleansing and preparation phase, reduced data size is 250 MB. This dataset contains 6782766 number of records showing type of crime, Latitude and Longitude of the location, which is used for clustering. 60000 records are considered for experiment, at the largest where the distance matrix size is computed around 1799970000 elements, equal to 14.6 GB.
Two more datasets used are in the format GeoJson (Geographic – JavaScript Object Notation). GeoJson is an open standard format for encoding a variety of geographic data structures [15]. The objects in the dataset are in the format of type, geometry, Coordinates and properties. All of the above datasets contain 3 variables, 1 categorical and 2 numerical. Numerical variables indicate Longitude and Latitude of the location mentioned in the observation [16].
Fourth dataset is of randomly generated numbers with large deviation.

Datasets used for this experiment are summarized as follows

Table-2: Data set with different size and format

Dataset	Format of dataset	Size in MB / GB	Number of Records	Number of records after data cleaning
Crimes_-_2001_to_present	CSV	2.20 GB	6800000	6782766
Crime_Incidents_in_2009	GeoJson	20MB	93852	93852
Invasive_Species	GeoJson	7 MB	2700	2700
randomsample	CSV	.2 MB	10070	10070

R Studio release 3.5.3 is used for experiments on local machine whereas Microsoft Azure HDInsight cloud web service portal is used to perform clustering on larger amount of data. Azure HD Insight is a Hadoop service offering hosted in Azure that enables clusters of managed Hadoop instances. Azure HDInsight deploys and provisions Apache Hadoop clusters in the cloud, providing a software framework designed to manage, analyze, and report on big data with high reliability and availability. Azure HD Insight uses the Hortonworks Data Platform (HDP) Hadoop distribution [11].
For Azure storage, datacenter is located at Central India, with storage account name 'chic-storage. Hadoop HDInsight cluster of different core sizes such as 4cores and 8 Cores are used for the experiment [17].
For computing time taken for clustering the dataset, actual time required for execution of the algorithm is noted. There are three parameters for time measurement of execution of an algorithm. 'User CPU time' gives the CPU time spent by the current and "system CPU time' gives the CPU time spent by the kernel of the operating system on behalf of the

current process. ‘Time Elapsed’ is the total of User Time and System Time.

The performance of the algorithms are evaluated on the basis of Medoids. According to the *definitions.net*, medoid is defined as representative objects of a data set or a cluster with a data set whose average dissimilarity to all the objects in the cluster is minimal. Medoids are similar in concept to means or centroids, but medoids are always members of the data set. The term is used in computer science in data clustering algorithms.

Silhouette analysis [18] is also used to study the dissimilarity distance between the resulting clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. This measure has a range of {-1 to 1} [19].

Silhouette coefficients (as these values are referred to as) near +1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster [19].

For measuring and comparing the performance, algorithms CLARA, PAM are executed on different sample sizes and the medoids computation is recorded. In the said experiment, medoids consistency for different sample sizes is compared for different types of datasets. Sample sizes considered vary from smallest to largest possible.

IV. RESULTS DISCUSSION AND INTERPRETATION

The outcome of the research work on the basis of experiments can be presented as below in Table-3

For Execution time Analysis

In case of hierarchical clustering, distance matrix computation is the task of highest time complexity [20]. To reduce the time required, parallel execution is opted for larger dataset. Experimental results are shown in the following tables

Table-3: Distance Matrix Computation for Wards Algorithm:

Size of Dataset – Number of Records	Number of nodes for computation	Time required in Seconds Sec.
20000	Standalone	289.58
20000	4 Nodes	10.013
60000	Standalone	Could not compute
60000	4 Nodes	92.993

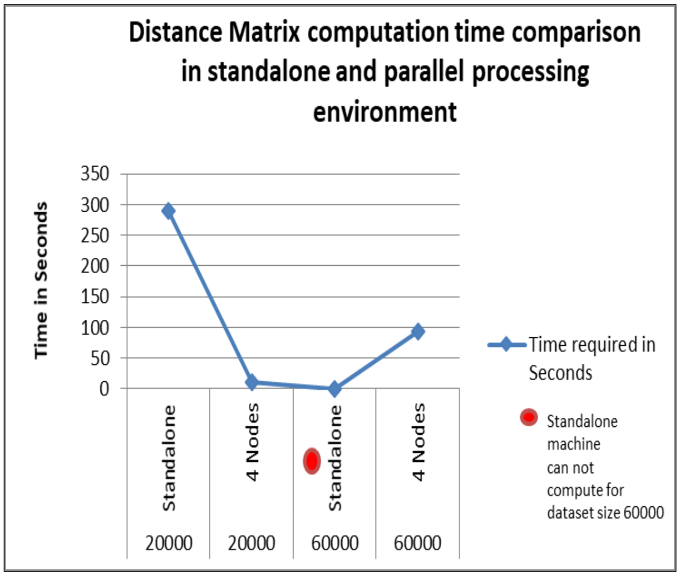

Figure 1 Distance Matrix computation time comparison

Table-4: Different Linkage methods of Wards Algorithm and their execution time.

Wards Linkage Method	User Time (sec)	System Time (sec)	Elapsed Time (sec)
complete	47.476	3.374	50.836
average	48.899	3.301	52.185
ward	50.062	3.273	53.32
ward.d2	50.68	3.302	53.967

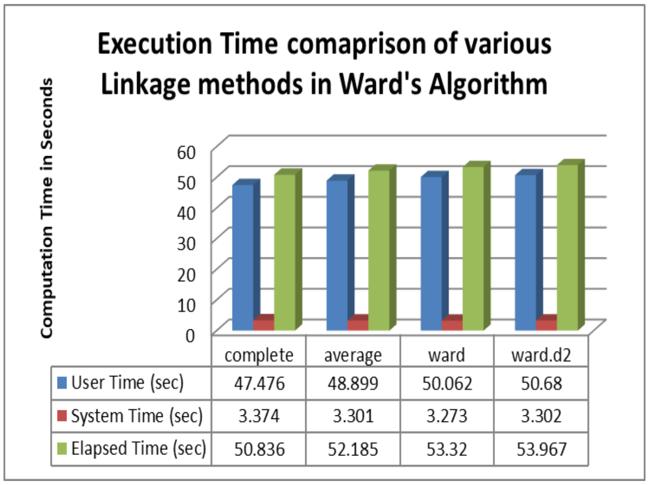

Figure 2: Comparison of execution time of various linkage methods in Wards Algorithm

Table-5: Partition and Density based Algorithm time analysis

Method	User Time (sec)	System Time (sec)	Elapsed Time(sec)	Reduction in time with respect to Wards Method
DBSCAN	25.62	2.921	28.536	45.72%
K means	0.027	0	0.026	99.95%
CLARA	0.018	0	0.018	99.96%

To analyze performances of K means, PAM and CLARA algorithms, centroid and medoid computations are performed for the variable 'Latitude' and 'Longitude' in the dataset. Outcome of CLARA algorithm is tested for three different sample sizes.

Table-6: Centroid and Medoid computation of based on variables 'Latitude' and 'Longitude' for different sample sizes of CLARA , algorithm PAM and k-means where 4 clusters are formed with below centroid/medoid for the dataset GeoJson Dataset with 2700 records.

Kmeans Centroid	PAM - Medoid	CLARA Medoid with sample size 20	CLARA Medoid with sample size 1000	CLARA Medoid with sample size 2500
23.2	14.7245	20	14.72842	14.72509
43.33	43.2272	43.2048	43.2272	43.22664
-102.75	-122.55	-122.2	-122.553	-122.5513
8.214	-73.183	-73.371	-73.187	-73.18277

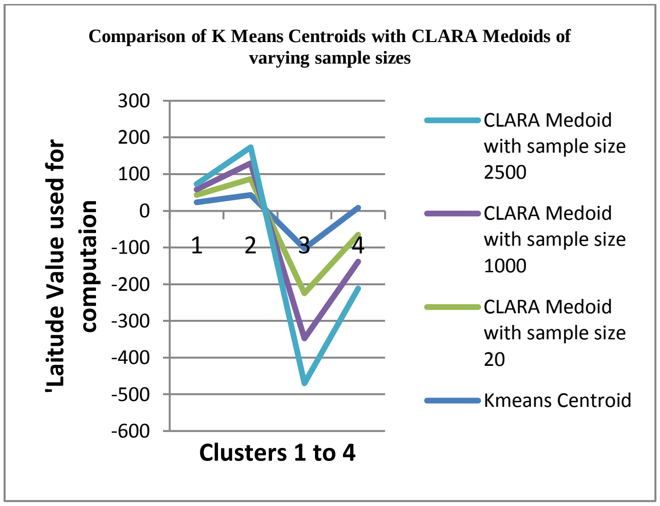

Figure 3: Centroids and Medoids comparison with different sample sizes.

Centroid and Medoid computation of based on variables 'Latitude' and 'Longitude' for different sample sizes of CLARA , algorithm PAM and k-means where 4 clusters are formed with below centroid/medoid for the dataset GeoJson Dataset with 93852 records

Kmeans Centroid	PAM - Medoid	CLARA Medoid with sample size 1000	CLARA Medoid with sample size 2500	CLARA Medoid with sample size 15000
76.97547	5	5	5	5
38.9	9	9	9	9
6.67	3	38.9	38.9	38.9
-77.035	6	-77	-77.01	-77.01

Table-8: Silhouette comparison in PAM and CLARA for 4 clusters for the GeoJson Dataset of 93852 records

PAM	CLARA Medoid with sample size 100	CLARA Medoid with sample size 2500	CLARA Medoid with sample size 15000
0.36	0.41	0.41	0.41

Silhouette comparison shows the neighboring cluster of a data element for which the average dissimilarity between its observations and neighbor cluster is minimal. Higher Silhouette value indicates higher dissimilarity [21-23]. Silhouette width can be defined as

$$s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))} \qquad (4)$$

Where a(i) = average dissimilarity between i and all other points of the cluster to which i belongs, b(i) = dissimilarity between i and its neighbor cluster.

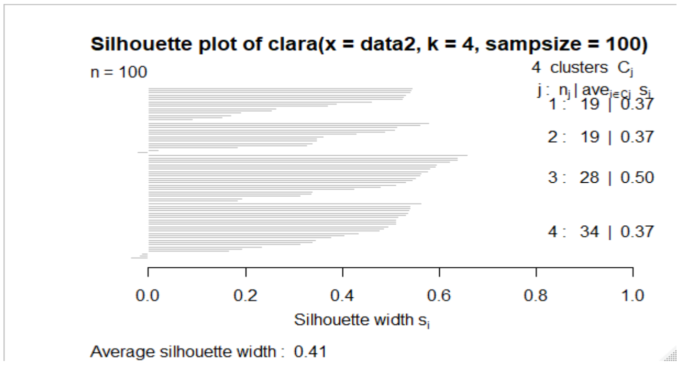

Figure-4: Sample Silhouette plot for sample size 100 Silhouette width = 0.41

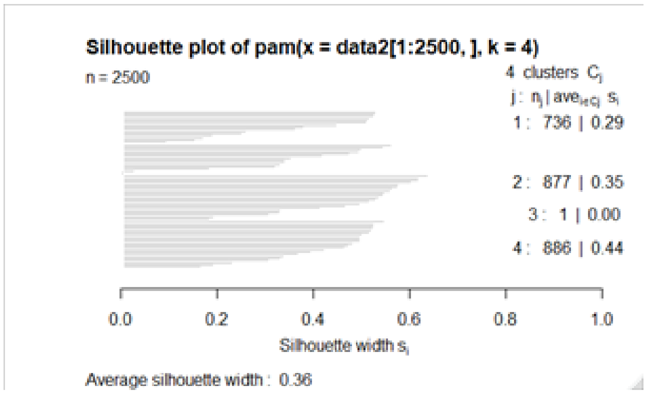

Figure 5- Sample Silhouette plot for PAM
Silhouette width = 0.36

V. Conclusion

Parallel computation on Azure platform reduces the computation time by 94.5 % in case of hierarchical algorithm applied on dataset of 20000 records. The major task in hierarchical algorithm computation is calculation of distance matrix. Parallel computation computes it effectively with reduction of 92.99% for large dataset of 60000 records that increases efficiency of hierarchical clustering algorithm. Irrespective of sample size, CLARA works consistently with small/large GeoJson dataset as well as CSV dataset. PAM does not show consistent medoid computation when applied on larger datasets.

Silhouette widths comparison of PAM (0.36) and CLARA (0.41) shows higher dissimilarity for CLARA which indicates well-formed clusters, even for large datasets.

REFERENCES:

[1] S. Miyamoto, R. Abe, Y. Endo, and J. Takeshita, “Ward method of hierarchical clustering for non-Euclidean similarity measures,” in *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, Fukuoka, Japan, 2015, pp. 60–63.

[2] Jian Yin, Zhi-Fang Tan, Jiang-Tao Ren, and Yi-Qun Chen, “An efficient clustering algorithm for mixed type attributes in large dataset,” in *2005 International Conference on Machine Learning and Cybernetics*, Guangzhou, China, 2005, pp. 1611-1614 Vol. 3.

[3] Lingling Yuan, “An effective Chinese short message texts clustering algorithm based on the ward’s method,” in *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, Deng Feng, China, 2011, pp. 1897–1899.

[4] J. Pagel, M. Campion, A. S. Nair, and P. Ranganathan, “Clustering analytics for streaming smart grid datasets,” in *2016 Clemson University Power Systems Conference (PSC)*, Clemson, SC, USA, 2016, pp. 1–8.

[5] M. K. Pakhira, “Fast Image Segmentation Using Modified CLARA Algorithm,” in *2008 International Conference on Information Technology*, Bhunaneswar, Orissa, India, 2008, pp. 14–18.

[6] S. Sreepathi, J. Kumar, R. T. Mills, F. M. Hoffman, V. Sripathi, and W. W. Hargrove, “Parallel Multivariate Spatio-Temporal Clustering of Large Ecological Datasets on Hybrid Supercomputers,” in *2017 IEEE International Conference on Cluster Computing (CLUSTER)*, Honolulu, HI, USA, 2017, pp. 267–277.

[7] X. Dong and Z. Zhang, “Research and implementation of PAM algorithm with time constraints,” in *Proceedings 2014 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS)*, Qingdao, China, 2014, pp. 108–111.

[8] X.-D. Wang, R.-C. Chen, F. Yan, Z.-Q. Zeng, and C.-Q. Hong, “Fast Adaptive K-Means Subspace Clustering for High-Dimensional Data,” *IEEE Access*, vol. 7, pp. 42639–42651, 2019.

[9] K. M. A. Patel and P. Thakral, “The best clustering algorithms in data mining,” in *2016 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, Tamilnadu, India, 2016, pp. 2042–2046.

[10] Li Wenchao, Z. Yong, and X. Shixiong, “A Novel Clustering Algorithm Based on Hierarchical and K-means Clustering,” in *2007 Chinese Control Conference*, Zhangjiajie, China, 2006, pp. 605–609.

[11] A. Bhardwaj, V. K. Singh, Vanraj, and Y. Narayan, “Analyzing BigData with Hadoop cluster in HDInsight azure Cloud,” in *2015 Annual IEEE India Conference (INDICON)*, New Delhi, India, 2015, pp. 1–5.

[12] C. Nishizaki, Y. Niwa, M. Imasato, and H. Motogi, “A method for feature extraction and classification of marine radar images,” in *2014 World Automation Congress (WAC)*, Waikoloa, HI, 2014, pp. 48–53.

[13] C.-Y. Kuo, C. N. Hang, P.-D. Yu, and C. W. Tan, “Parallel Counting of Triangles in Large Graphs: Pruning and Hierarchical Clustering Algorithms,” in *2018 IEEE High Performance extreme Computing Conference (HPEC)*, Waltham, MA, 2018, pp. 1–6.

[14] M. Alkathiri, J. Abdul, and M. B. Potdar, “Kluster: Application of k-means clustering to multidimensional GEO-spatial data,” in *2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC)*, Indore, 2017, pp. 1–7.

[15] S. Soor and B. S. D. Sagar, “Iterated Watersheds, A Connected Variation of K-Means for Clustering GIS Data,” p. 11.

[16] K. L. N. Eranki and A. S. Reddy, “Geo-spatial library: A geo-spatial educational tool for knowledge management and capacity building,” in *2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA)*, Kottayam, India, 2012, pp. 1–4.

[17] A. S. Sidhu, S. R. Balakrishnan, and S. K. Dhillon, “HPC&#x002B;Azure environment for bioinformatics applications,” in *2013 IEEE International Conference on Bioinformatics and Biomedicine*, Shanghai, China, 2013, pp. 12–15.

[18] C. Reinbacher, M. Ruther, and H. Bischof, “Pose Estimation of Known Objects by Efficient Silhouette Matching,” in *2010*

20th International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 1080–1083.

[19] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *MACHINE LEARNING IN PYTHON*, p. 6.

[20] Y. Zhuang, Y. Mao, and X. Chen, “A Limited-Iteration Bisecting K-Means for Fast Clustering Large Datasets,” in *2016 IEEE Trustcom/BigDataSE/ISPA*, Tianjin, China, 2016, pp. 2257–2262.

[21] S. Gupta and V. K. Srivatava, “An accelerated clustering algorithm for segmentation of grayscale images,” in *2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011)*, Allahabad, India, 2011, pp. 660–665.

[22] Marie Fernandes , “Data Mining: A Comparative Study of its Various Techniques and its Process”, International Journal of Scientific Research in Computer Science and Engineering, Vol.5, Issue.1, pp.19-23, 2017

[23] Nilamadhab Mishra , “Internet of Everything Advancement Study in Data Science and Knowledge Analytic Streams”, International Journal of Scientific Research in Computer Science and Engineering, Vol.6, Issue.1, pp.30-36, 2018.