ISSN 2347 - 3983

Volume 9. No. 2, February 2021

International Journal of Emerging Trends in Engineering Research

Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter20922021.pdf

https://doi.org/10.30534/ijeter/2021/20922021

Improved CURE Clustering Algorithm using Shared Nearest Neighbour Technique

Nikita Kumble[1], Vandan Tewari[2]

Department of Computer Engineering, Shri Govindram Seksaria Institute of Technology
and Science, Indore, India

[1]nikitakumble@gmail.com, [2]vandantewari@gmail.com

ABSTRACT

Clustering is the unsupervised learning based grouping of data points based on the similarity between them. Traditional clustering algorithms work well with datasets which have globular/spherical shape. When it comes to non-spherical shaped clusters, they may divide large clusters into small clusters or merge two clusters.

CURE(Clustering using REpresentatives) clustering algorithm overcomes the limitations of traditional clustering algorithms for clustering non-globular shaped clusters. CURE algorithm chooses random points as representative points from each cluster and shrinks them towards the centroid of the clusters. The problem arises when data sets do not have a centroid tendency. To resolve this issue, we propose an improved CURE algorithm, where instead of shrinking of representative points, the shared neighbours between the points have been used to form clusters. The representative points which share the same neighbourhood are put together in the same clusters. This allows generating clusters of non-globular shaped clusters. Our neighbourhood based clustering does not get affected by the shape of the clusters. Experimental results of our work demonstrate that CURE clustering using shared nearest neighbours has better performance than CURE.

Key words : Clustering, Nearest neighbours, Random sampling, Representative points, Shared nearest neighbour graph, Similarity matrix.

1. INTRODUCTION

An enormous amount of data is generating continuously from various resources like corporate databases and world wide web day by day**[1]**. Handling and analysis of such large data is a complex task. Data mining is a process of Knowledge Discovery in Databases(KDD) which is used to find underlying, previously unknown and potentially beneficial information which is hidden in the data.

Clustering is unsupervised learning as the dataset is not categorized into any class[2]. This is done in such a way that for a cluster, the objects belongs to it are more similar to each other than to objects which belong to the other clusters. CURE is a hierarchical clustering algorithm which is very useful for datasets of large size[3]. It is insensitive to outliers and noise. CURE is capable of identifying the non-globular shaped clusters. For globular clusters, CURE follows a middle path between centroid based and all-point based algorithms. In centroid based technique, a centroid data point is used for labelling the cluster[4]. But, as a single data point cannot define the cluster properly, this method fails to identify an arbitrarily shaped cluster. In contrast, in all-point based method every data point is considered for clustering, due to which this method is very sensitive towards the outliers and noise data points.

To surpass the limitations of both methods, CURE acquire a compromise method between these two methods. For this, CURE chooses some well-scattered representative data points initially to describe the clusters. The representative points are selected from the whole dataset randomly. These representative points are then shrunk in the direction of the centroid of the cluster. The shrinking of representative points reduces the adverse effects of outliers and makes CURE insensitive to outliers and noises.

The shrinking of representative points towards the centroid shows that CURE has a hidden assumption of clusters to be spherical or elliptical[5]. This shrinking of representative points will not allow CURE to capture the shape of clusters which are not spherical and elliptical.

To mitigate this problem, in this work, an improved CURE algorithm has been proposed which instead of shrinking of representative points, uses the shared nearest neighbours of data points to form clusters. As the neighbourhood of a data point does not have any relation with the shape of the cluster, our proposed algorithm is capable of identifying the clusters of arbitrary shapes. Further, as an outlier is far away from the other data points, it does not share the neighbourhood with other points. This allows the CURE with Shared Nearest neighbour(SNN) to identify the outliers efficiently.

By analysing the time and space efficiency, sensitivity to parameters and quality of clustering, we calculated that improved CURE with SNN has the time and space efficiency similar to CURE. It requires very few initialisation parameters and is less sensitive towards the chosen value of parameters.

The remaining paper is categorized as follows: In Section **2,** it contains the overview of work done related to clustering by various algorithms, section **3**, contains the overview of CURE algorithm and its drawbacks. In section **4** the summarization of the working of our proposed algorithm is done. Experimental analysis of our CURE using SNN is given in section **5**. Finally, the conclusion of our work and the summarization of our algorithm are stated in section **6.**

2. RELATED WORK

To handle large data, it is categorized or classified into a set of group, partitions or clusters. Clustering is an unsupervised learning technique in which the dataset available does not contain predefined class labels.

Traditional clustering algorithms are categorized into partitional based, hierarchical based and density based algorithms. The partitional algorithms are either centroid based or medoid based[6]. It uses the sum of the square error to generate the clusters. k-means works well for spherical shape clusters but for clusters of varying size and densities, k-means does not identify clusters accurately. Also, it is difficult for k-means to handle noisy data and outliers.

Hierarchical clustering algorithms follow a hierarchy in the clustering. It has a core idea that the data points near to each other are more alike to each other than the data points at a farther distance. BIRCH hierarchical algorithm can extent while handling voluminous data. It is robust against noise and outliers. But BIRCH is not able to deal with clusters of non-globular shapes as to decide the boundaries of clusters, it uses the principle of diameter measure[7]. ROCK works well for increased in dimensionality[8]. For categorical attributes, it uses links between a pair of data points for proximity measurement between them. ROCK cannot accurately define the clusters of different size and shapes. CURE[3] is discussed in the next section briefly, which can identify non-globular shaped clusters. It is very less sensitive towards outliers and noise. CHAMELEON[9] has more power for determining arbitrary shaped clusters of high quality then CURE and BIRCH. Although, the running time required to it is more than CURE and BIRCH. In the hierarchical clustering algorithm, after each merge operation, it cannot be undone. This impacts the globular optimization of the generation of clusters.

The density based algorithms efficiently identify clusters of arbitrary shape and efficient to handle noise better than hierarchical and partitioning algorithms. But the efficiency of the algorithm decreases on increasing the dimensionality. DBSCAN[10] gives excellent results against noise, but it targets low dimensional spatial data. Also, the quality of the clusters decreases if the clusters are of varying densities. DENCLUE[11] is based on the concept of density and hill climbing. This algorithm is capable to identify and converge clusters of unpredictable shapes but, it is sensitive towards the input parameters. Also, for high dimensional data, it suffers from the curse of dimensionality.

3. CURE CLUSTERING ALGORITHM[3]

CURE(Clustering Using REpresentatives) is a hierarchical clustering algorithm which follows an intermediate scheme from centroid based and all-point based. Figure 1 shows the flow of the CURE algorithm. Rather than using a single centroid or all-points, it chooses a predetermined number of representative points to represent the cluster. From all the data points, it chooses randomly scattered points. These scattered points are then shrunk in the direction of the mean of the cluster to form the partial clusters of the dataset. CURE is more robust towards the outliers as shrinking of the outlier towards the mean is typically larger than other data points. So CURE identifies outliers more efficiently.

The synopsis of the CURE algorithm is given below:

1. The dataset of size N and the expected number of clusters k are given as input.
2. Randomly choose representative points from the input data and shrunk them towards the centroid to form the partial clusters.
3. Merge the two clusters which are at the minimum distance. Randomly choose the new representative points for the merged cluster.

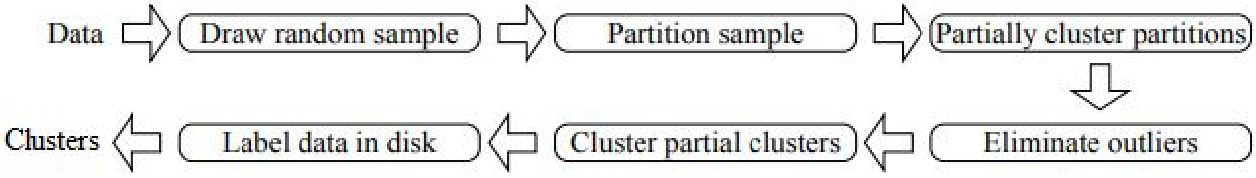


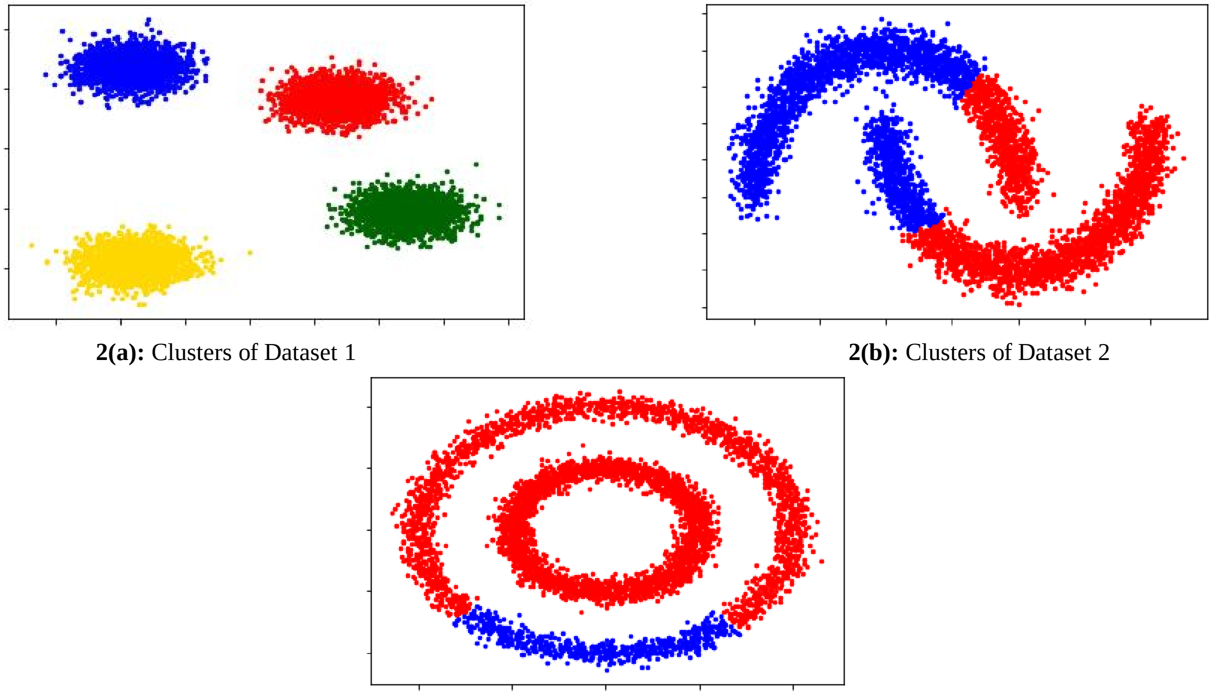
Figure 1: Flow diagram of CURE Algorithm[3]

4. If the desired number of clusters are generated or there is no cluster is being merged, the algorithm stops. Otherwise, repeat step 3.

For a database of large size, CURE applies random sampling and partitioning methods. Firstly, a random sample is chosen from the dataset that fits in the main memory. To speed up the execution time, the partitions of the random sample is done.

All the partitions are partially clustered simultaneously to reduce the execution time. After the generation of clusters of the random sample, the remaining data points of the dataset are clustered by calculating their distance from the representative points of the clusters of the random sample.

The clusters generated by CURE are mainly depended on the representative point. Multiple representative points conquer the shapes and extend of the cluster. It allows CURE to discover the non-globular clusters. The shrinking of scattered points towards the centroid of the cluster by fraction α diminishes the impact of outliers. The fraction α lies from 0 to 1. For α=1, CURE becomes similar to the centroid based approach, whereas, for α=0, it reduces to all point based approach. To handle outliers, CURE assumes that outliers are at the farthest distance from the centroid, so the shrinking of representative points forms clusters of good quality by eliminating the outliers. It enables CURE to accurately determine the clusters in figure 2(a). But for arbitrary shape clusters, this is not always confirmed. For such clusters, the shrinking of representative points not only place the outlier in a cluster, but it also split the cluster by identifying the data point at a farther distance as an outlier. Thus, this shrinking operation consequence in many issues in handling some specifically shaped clusters, as illustrated in figure 2(b-c).



2(a): Clusters of Dataset 1

2(b): Clusters of Dataset 2

2(c): Clusters of Dataset 3

Figure 2: Resultant clusters of datasets generated by Cure algorithm

4. CURE WITH SNN

In CURE algorithm, the representative points shrink towards the centroid to form the clusters. But, for clusters of non-globular shapes, shrinking of the representative points towards the centroid is not appropriate. It shows that CURE somehow has hidden assumption that the clusters have centroid so they are spherical. To overcome this limitation, we proposed to cluster the data using shared nearest neighbour graph which is not centroid based. The flow of CURE algorithm with SNN is shown in figure 3 which shows all the modules of our work.

Cure with SNN is more efficient than CURE algorithm as it can identify the clusters which cannot be identified by CURE correctly as shown in figure 4(b) and 4(c). The shrinking of representative points towards the centroid of the cluster of CURE algorithm results in many problems to deal with such shape clusters. Whereas, in our proposed work, the data points are clustered based on their neighbourhood, which allows Cure with SNN to identify arbitrary shaped clusters more efficiently.

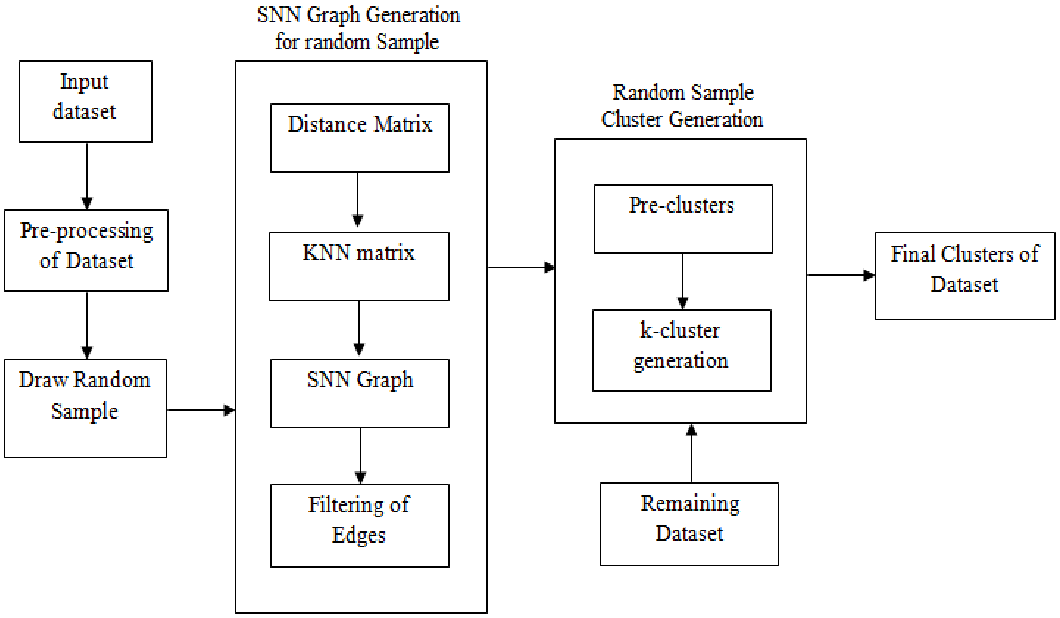


Figure 3:Flow diagram of CURE with SNN

Initially, for each data point from the random sample, for each pair of data points, the distance between them is calculated to find the p nearest neighbours of each point. For each data point, the shared nearest neighbours are calculated from all data points from their neighbours to generate SNN graph. In the SNN graph, the links between data points which shares neighbours less than the threshold are removed. The data points which are linked are merged into one cluster as they share neighbours more than the threshold. This allows clustering of the data points without getting affected by the shape of clusters. Once the pre-clusters are generated from the SNN graph, to generate the required number of clusters, the pre-clusters are merged. The merging of the clusters is based on the distance between them. For each pre-cluster, representative points are selected randomly. The minimum distance between the representative points is considered as the distance between two clusters. The clusters at the minimal distance are merged to make a bigger cluster. The merging of clusters continues until the required number of clusters are formed.

The labelling of disk data in Cure with SNN is similar to CURE algorithm. As the k clusters are generated from the random sample, rep_point representative points are selected from each cluster for the labelling of remaining dataset. For each data point, its distance is evaluated from the the representative points of clusters. The data point is allocated to the clusters which contain the representative point which is at the minimum distance from the data point.

4.1 Algorithm

Input: random sample s, required number of clusters k
Output: k clusters of random sample

```
n=size of random sample
p=number of nearest neighbours
knn_val=threshold for shared nearest neighbours
m=number of pre-clusters
rep_point=representative points
START
for i=0 to n do     (*Compute distance between each pair of data point*)
  for j=i+1 to n do
    $d_{ij}=\sum|x_i-x_j|^2$;
for i=0 to n do     (*Find p most similar data points for each point*)
  while (data point in $knn\_list_{i)}$ !=p do
    for j=i+1 to n do
      $knn\_list_i$=min($d_{ij}$);
for i=0 to n do
  for j=i+1 to n do
    $SNNGraph_{ij}$= $knn\_list_i \cap knn\_list_j$     ((*SNNGraph containing shared nearest neighbours between points*)
if $SNNGraph_{ij}$<knn_val do
  $SNNGraph_{ij}$=0;     (*Filter the edges of SNNGraph less than threshold*)
else
  $pre\_cluster_i$=insert($point_j$) AND $pre\_cluster_j$=insert($point_i$); (*Insert the data points in pre-clusters*)
#Generation of k clusters from Pre-clusters
while (m>k) do
  for each i € pre_cluster do
    $p_i$=rep_points($pre\_cluster_i$);     (*Choose rep_point representative points from each cluster*)
  for each pair of cluster compute do
    $dist_{ij}$=cluster_distance($p_i$,$p_j$);     (*minimum distance between each pair of clusters*)
```

```
   distance=mini($dist_{ij}$);   (*u,v clusters at minimum distance*)
   u=i
   v=j
   merge($pre\_cluster_u$,$pre\_cluster_j$); (*Merge clusters at minimum distance*)
(*end of while*)
END
```

5. EXPERIMENTAL RESULTS

For testing the proposed algorithm, Python has been used to implement the algorithm and verifying the effects and results. The experiment hardware platform used: CPU is Intel(R) Pentium(R) CPU 1.06GHz, memory is 2GB; the operating system is Windows 8.1 Pro.

We have used Euclidean distance as the distance metric for all of our experiments. The results of our work for different datasets are shown in figure 4. It shows the clusters generated by CURE with SNN from various datasets.

5.1 Datasets

We performed experiments with four datasets. Here we are reporting three datasets. The attributes of the datasets are shown in table 1. Dataset 1 contains 8000 data points with 3 features. It consists of four blobs shaped clusters. This dataset contains 10% noise data points of the whole dataset. Dataset 2 contains two moons one partly “inserted” in the other. This dataset includes 8% noise data points of the whole dataset. Dataset 3 consists of two circles with the same centres. This dataset includes 10% noise data points of the whole dataset. Experimental outcomes show that our proposed method clusters these datasets correctly. The clusters of dataset 1 are shown in figure 4(a). The output of our work for dataset 2 is presented in figure 4(b). And, the clusters of dataset 3 are shown in figure 4(c). The CURE clustering algorithm with shared nearest neighbours clusters the datasets very efficiently.

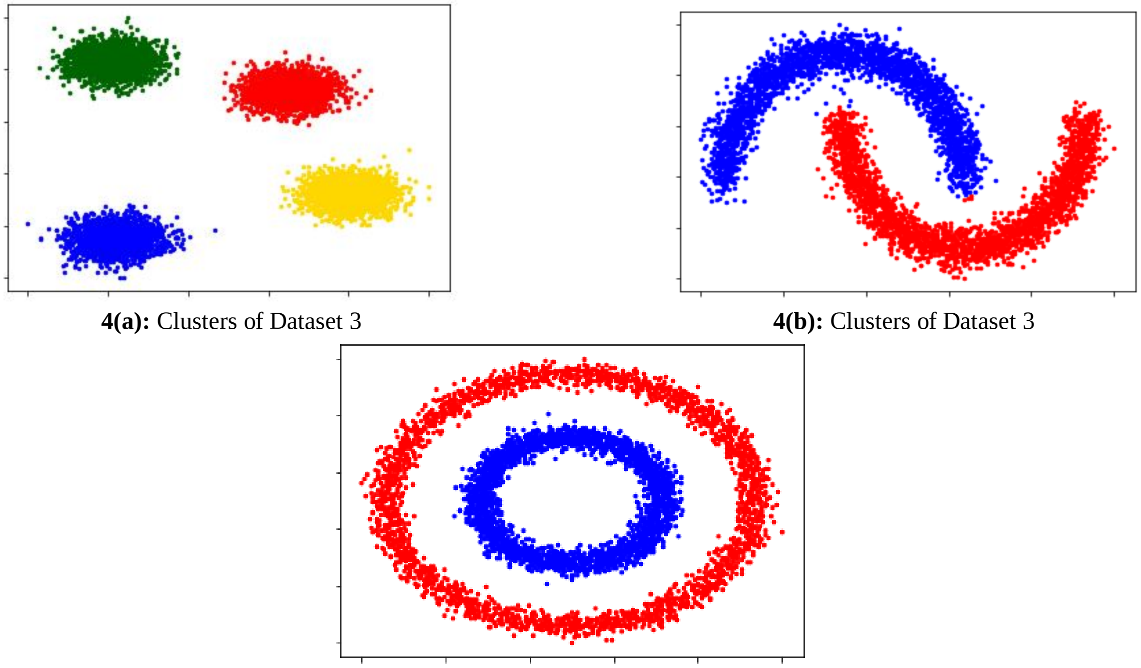
Table 1: Details of datasets

Datasets	Data points	Noise points(in % of the whole dataset)	Structure of Clusters	Number of Clusters
Dataset 1	8000	10	Blobs	4
Dataset 2	5000	8	Moons	2
Dataset 3	5000	10	Concentric Circles	2

5.2 Sensitivity to parameters

In this subdivision, we analyze for all the parameters of our algorithm which are *s*, *p*, *knn_val*, *rep_point*.

- **random sample, *s*:** We performed our experiment with different sample size starting from 100 to 1000. We found that for sample size less than 500, the quality of clusters is not good as the small sample does not acquire the geometry of the whole dataset. However, for *s* of size 500 and above, the clusters are identified correctly.



4(a): Clusters of Dataset 3

4(b): Clusters of Dataset 3

4(c): Clusters of Dataset 3

Figure 4: Resultant clusters of datasets using Cure with SNN

- **Nearest neighbours, *p*:** We have performed our algorithm for various values of p. The smallest value for which the algorithm clusters the datasets correctly varies from 5 to 20.
- **Shared neighbours, *knn_val*:** For the value of *knn_val*, we varied it from 1 to 17. For the value of *knn_val* less than 4, the algorithm takes a long time to find the clusters of the random sample. Thus, we conclude that 4 to 17 is a convenient range of value for *knn_val* to determine non-globular clusters. The time required for the clustering decreases as an increase in the value of shared neighbours, knn_val.

  The value of knn_val varies with the presence of noise. For smaller noise, a high value of knn_val can be used to generate clusters faster. But, as the noise increases, for a large value of knn_val, the clusters can be merged due to noise data point. In such a case, a smaller value of knn_val generates the clusters correctly.
- **representative points, *rep_point*:** We ran our algorithm by varying the value of *rep_point* from 1 to 50. For a smaller value of *rep_point*, we found that the quality of the clusters suffers. For example, when *rep_point* is set to 7, the points do not acquire the geometry of clusters adequately. Although, for the value of *rep_point* more than 22, the algorithm found the correct clusters all the time.

In table 2, the range and default values of sensitivity parameters are presented.

Table 2: Sensitivity parameters and their values for Cure with SNN

Symbol	Meaning	Default Value	Range
S	Size of the random sample	500	500-1000
P	nearest neighbour data points	7	6-20
knn_val	Shared neighbour data points	5	4-17
rep_point	representative points	25	22-30

6. CONCLUSION

In this paper, we present an improved CURE clustering algorithm using the Shared Nearest Neighbours method, to overcome the shortcomings of the CURE algorithm where the shrinking of representative points is based on the supposition of globular shaped clusters. As contrary to the shrinking of representative points towards the centroid of clusters, we calculated the distance between the data points of the random sample to find out the shared neighbours between the data points. Since the data points which shares the neighbours are put in a cluster, the data points are clustered better than centroid based. This enables the algorithm to identify the varying non-globular shaped clusters by adjusting well to their geometry. Also, it overcomes the drawbacks of the CURE algorithm where shrinking towards the centroid can merge two interleaving clusters. From the results of our experiment, it is shown that CURE algorithm with Shared Nearest Neighbour works well for both globular and non-globular shaped clusters efficiently.

REFERENCES

1. T. Aslanidis, D. Souliou, and K. Polykrati, **"CUZ: An Improved Clustering Algorithm"**, 2008 *IEEE 8th International Conference on Computer and Information Technology Workshops*, Sydney, QLD, 2008, pp. 43-48. doi: 10.1109/CIT.2008.Workshops.118.
2. K. Bindra, and A. Mishra, **"A detailed study of clustering algorithms"**, 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, 2017, pp. 371-376, doi:10.1109/ICRITO.2017. 8342454.
3. S. Guha, R. Rastogi, K. Shim, **"CURE: An Efficient Clustering Algorithm for Large Databases"**. In: Proceedings of the 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98). Seattle, Washington, ISBN:0-89791-9955 doi: 10.1145/276304. 276312.
4. Garima, H. Gulati, and P. K. Singh, **"Clustering techniques in data mining: A comparison"**, 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2015, pp. 410-415.
5. Y. T. Qian, Q. S. Shi, and Q. Wang, **"CURE-NS: a hierarchical clustering algorithm with new shrinking scheme"**, Proceedings. International Conference on Machine Learning and Cybernetics, Beijing, China, 2002, pp. 895-899 vol.2. doi: 10.1109/ICMLC.2002.1174512.
6. S. S. Mary D. A, T. R. Selvi, 2014, **A Study of K-Means and CURE Clustering Algorithms**, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) Volume 03, Issue 02 (February 2014).
7. T. Zhang, R. Ramakrishnan, L. Miron. 1996. **BIRCH: an efficient data clustering method for very large databases**, In Proceedings of the 1996 ACM SIGMOD international conference on Management of data, p.103-114, June 04-06, 1996, Montreal, Quebec, Canada
8. S. Guha, R. Rastogi, and K. Shim, **"ROCK: a robust clustering algorithm for categorical attributes"**, Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337), Sydney, NSW, Australia, 1999, pp. 512-521, doi: 10.1109/ICDE.1999.754967.

9. G. Karypis, E. H. Han E. H. and V. Kumar, **"Chameleon: hierarchical clustering using dynamic modeling"**, in Computer, vol. 32, no. 8, pp. 68-75, Aug. 1999, doi: 10.1109/2.781637.
10. K. Khan, S. U. Rehman, K. Aziz, S. Fong and S. Sarasvady, **"DBSCAN: Past, present and future"**, The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), Bangalore, 2014, pp. 232-238, doi: 10.1109/ICADIWT.2014.6814687.
11. J. Oyelade et al., **"Data Clustering: Algorithms and Its Applications"**, 2019 *19th International Conference on Computational Science and Its Applications (ICCSA)*, Saint Petersburg, Russia, 2019, pp. 71-81, doi: 10.1109/ICCSA.2019.000-1.
12. P. Lathiya, and R. Rani, **"Improved CURE clustering for big data using Hadoop and Mapreduce"**, 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, 2016, pp. 1-5. doi: 10.1109/INVENTIVE.2016.7830238.
13. S. Maitrey, C. K. Jha, R. Gupta and J. Singh. Article: **Enhancement of CURE Clustering Technique in Data Mining.** IJCA Proceedings on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI 2012) DRISTI(1):7-11, April 2012.
14. I. A. Sabri, M. Man, W. A. W. Abu Bakar, and A. Rose. (2019). **Web Data Extraction Approach for Deep Web using WEIDJ**. Procedia Computer Science. 163. 417-426. 10.1016/j.procs.2019.12.124.
15. M. A. T. Laksono, Y. Purwanto and A. Novianty, **"DDoS detection using CURE clustering algorithm with outlier removal clustering for handling outliers"**, 2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, 2015, pp. 12-18. doi: 10.1109/ICCEREC.2015.7337029.
16. S. Xiufeng, and C. Wei, **"Improved CURE algorithm and application of clustering for large-scale data"**, 2011 IEEE International Symposium on IT in Medicine and Education, Cuangzhou, 2011, pp. 305-308. doi: 10.1109/ITiME.2011.6130839.
17. D. Saravanan, **"CURE clustering technique suitable for video data retrieval"**, *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Chennai, 2016, pp. 1-4. doi: 10.1109/ICCIC.2016.7919592.
18. Z. Yan, **"Research of an improved cure algorithm used in enterprise competitive intelligence to dynamic identify analysis"**, 2010 *IEEE Youth Conference on Information, Computing and Telecommunications*, Beijing, 2010, pp. 299-302. doi: 10.1109/YCICT.2010.5713104.
19. S. Tiruveedhula, C. M. Rani and V. Narayana. (2016) **"A Survey on Clustering Techniques for Big Data Mining"**, Indian Journal of Science and Technology, [S.l.], feb. 2016. ISSN 0974 -5645. doi:10.17485/ijst/2016/v9i3/75971.
20. A. Idrissi, H. Rehioui, A. Laghrissi, A., and S. Retal, **"An improvement of DENCLUE algorithm for the data clustering"**, 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA), Marrakech, 2015, pp. 1-6, doi: 10.1109/ICTA.2015.7426936.
21. P. Singh and P. A. Meshram, **"Survey of density based clustering algorithms and its variants"**, 2017 *International Conference on Inventive Computing and Informatics (ICICI)*, Coimbatore, 2017, pp. 920-926, doi: 10.1109/ICICI.2017.8365272.
22. P. S. Badase, G. P. Deshbhratar and A. P. Bhagat, **"Classification and analysis of clustering algorithms for large datasets"**, 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2015, pp. 1-5, doi: 10.1109/ICIIECS.2015.7193191.
23. A. Fahad et al**., "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis"**, in IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 3, pp. 267-279, Sept. 2014, doi: 10.1109/TETC.2014.2330519.