Machine Learning with Applications 6 (2021) 100148

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

ECR-DBSCAN: An improved DBSCAN based on computational geometry

Kinsuk Giri [a,*], Tuhin Kr. Biswas [a], Pritisha Sarkar [b]

[a] Department of Computer Science and Engineering, National Institute of Technical Teachers' Training and Research Kolkata, Block - FC, Sector - III, Salt Lake, Kolkata 700106, India

[b] Department of Computer Science and Engineering, National Institute of Technology, Durgapur, West Bengal, 713209, India

ARTICLE INFO

Keywords:
Computational geometry
Empty circles
DBSCAN
KNN
Noise
Performance metrics

ABSTRACT

A new density based clustering algorithm $ECR-DBSCAN$ based on $DBSCAN$, has been presented in this paper. Computational geometry is applied to develop the modified $DBSCAN$ algorithm. It is well known that the quality of density based clustering depends on its input parameters. However, it is not easy to determine proper values of input parameters for $DBSCAN$. This paper presents three significant modifications or extensions to $DBSCAN$ related with (i) selection of hyper parameter $epsilon$ (eps) using the radii of empty or voronoi circles (ii) selection of parameter minPoints (mp) for the same epsilon and (iii) redistribution of noise points to suitable clusters using the concept of centroid hinged clustering. $ECR-DBSCAN$ is implemented with PYTHON accompanied by extensive experiment on benchmark data sets. Our experimental results establish the novelty and validity of the proposed clustering method over standard techniques.

1. Introduction

Computational geometry, which is widely used across various technical domain deal with different geometrical shapes for different computing algorithms. Spatial clustering is one of the popular clustering technique, which is essential to organize the artificial or real data set using some certain defined spatial characteristics. This work focuses on the suitable selection of the parameters related to the traditional $DBSCAN$ algorithm in context of computational geometry. The various ideas of computational geometry such as the convex hull, voronoi diagram, and empty circles have been successfully used in this work in quest of those parameters. We have also redistributed noise points into some suitable clusters generated from $DBSCAN$ using distance metrics. Various researchers have applied different techniques to enhance the performance of $DBSCAN$ clustering. A summary of these findings has been elaborated below with suitable references.

The $DBSCAN$ algorithm, a density based clustering, was originally proposed by Ester et al. (1996). They had procured the key-parameter $epsilon$ of $DBSCAN$ by finding the knee value of kth nearest neighbor's distances on the given data set. Later, a novel technique which is termed as $BDE-DBSCAN$ consisting of Binary Differential Evolution(BDE) and $DBSCAN$ clustering was proposed Karami and Johansson (2014). They presented a hybrid method with a combination of analytical $DBSCAN$ and tournament selection method that would help to choose the parameters of $DBSCAN$, viz., $epsilon$ and $minimumpoints$ in a sophisticated manner. Ren et al. (2012) represents another approach of modified $DBSCAN$ named as $DBCAMM$ (Density Based Clustering Algorithm with Mahalanobis Metric) where the Mahalanobis distance has been used instead of Euclidean distance for $DBSCAN$. Moreover they present the effective merging approach which can handle the better visualization for image segmentation as compared to the other approaches in less time. Lai et al. (2019) discovered a new method where an optimization technique was proposed. In their 'MVO-multiverse optimizer algorithm' -the parameters of $DBSCAN$ algorithm are selected through iterative variable updating. In the same context, a different approach for the optimization of time complexity for $DBSCAN$ was proposed by Jang and Jiang (2019). They termed their method as $DBSCAN++$, a step towards a fast and scale able $DBSCAN$. They provided two simple strategies: uniform and greedy k-center-based sampling in which it is shown that for simulated and real data sets, $DBSCAN++$ runs in a fraction of the time compared to the normal $DBSCAN$. Khan et al. (2018) suggested an algorithmic approach called adaptive $DBSCAN$ algorithm, to determine an appropriate epsilon and number of minimum points. However, Sander and his collaborators Sander et al. (1998) proposed a different version called $GDBSCAN$ which is a generalize version of traditional $DBSCAN$. Instead of counting the points in the neighborhood of an arbitrary object, in $GDBSCAN$ they rather focused in binary predicate and a uniquely defined metric. A detailed analysis of prototype based $DBSCAN$ is shown in Reddy's group (Edla & Jana, 2012). To discover the density varied clusters, Ram et al. (2010) has given a new density varied algorithm which turned out to be successful

* Corresponding author.
E-mail addresses: kinsuk@nitttrkol.ac.in (K. Giri), tuhin11biswas@gmail.com (T.Kr. Biswas), pritishapixie@gmail.com (P. Sarkar).
URL: http://www.nitttrkol.ac.in/kinsuk/home.html (K. Giri).

https://doi.org/10.1016/j.mlwa.2021.100148
Received 23 December 2020; Received in revised form 19 July 2021; Accepted 24 August 2021
Available online 2 September 2021

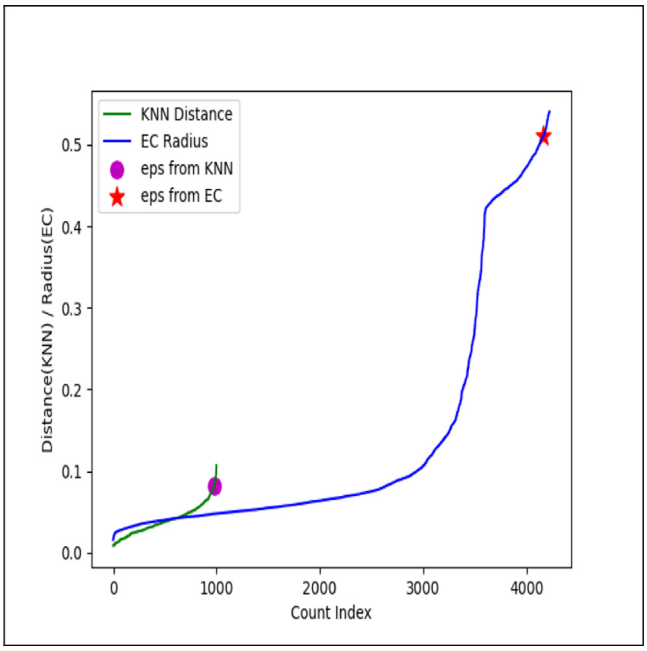

Fig. 1. Distributions for both KNN distances & ECs radii with their knee values ($epsilon$ parameter) for $Chainlink$ data set. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in handling the local density variation within the cluster. Introducing the concept of density factor, Birant and Kut (2007) shows that their density-based clustering algorithm $STDBSCAN$, not only discovers the clusters on spatial temporal data but also find noises when clusters of different densities occurs. The $VDBSCAN$(Varied Density Based Spatial Clustering of Applications with Noise) algorithm was proposed by Liu et al. (2007) and it was an improved density-based algorithm attached to effective clustering analysis of data sets with varied densities. Ruiz et al. (2007) enhanced the density-based algorithm $DBSCAN$ with constraints upon data instances –"Must Link" and "Cannot-Link" constraints. $MR-DBSCAN$ was proposed by He et al. (2011), where they analyze the concurrent parallel $DBSCAN$ algorithm, and further they also implemented an efficient parallel $DBSCAN$ algorithm, using the so called 4-stages Map Reduce paradigm. In terms of big data analysis, to identify the clusters within mixed data sets an improved $DBSCAN$ algorithm viz. $EPDCA$ was established in Liu et al. (2017), where, entropy and probability have been integrated together.

It is well known that the $DBSCAN$ algorithm is capable enough for noise detection using a density-based approach. However, it was found to be very much sensitive on two parameters, viz., $epsilon$ and number of minimum points ($minPoints$). In both of the original $DBSCAN$ and most of the above-mentioned modified $DBSCAN$ algorithms, the parameter $minPoints$ has been treated as a free parameter. Typically, the thumb rule for taking $minPoints$ is that it would be greater or equal to the dimensionality of data set. Apart from that, the most important parameter epsilon (hereafter eps) of $DBSCAN$ is conventionally computed using the kth Nearest Neighbor distances. However we would like to present a contrast by proposing a novel approach to find those two

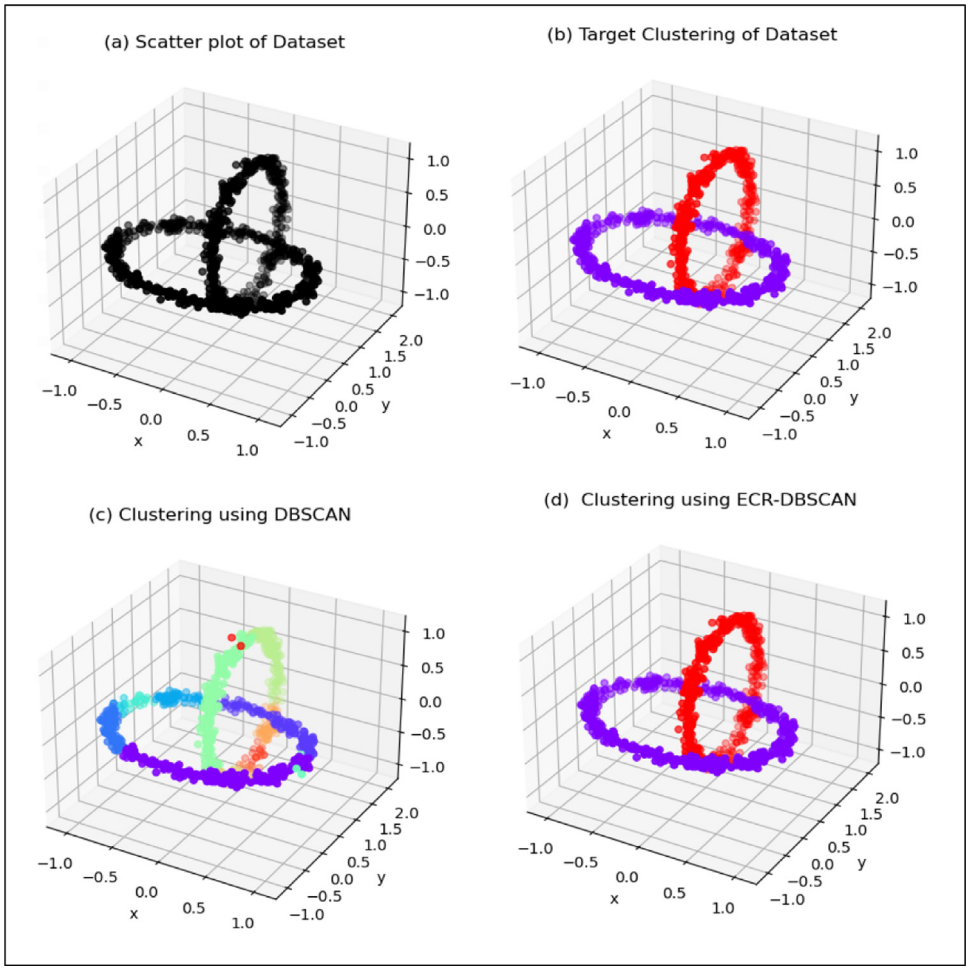

Fig. 2. (a) Scatter plot of data set (b) Target clustering (c) Data set clustered with $DBSCAN$ (d) Data set clustered with $ECR-DBSCAN$ for $Chainlink$ data set.

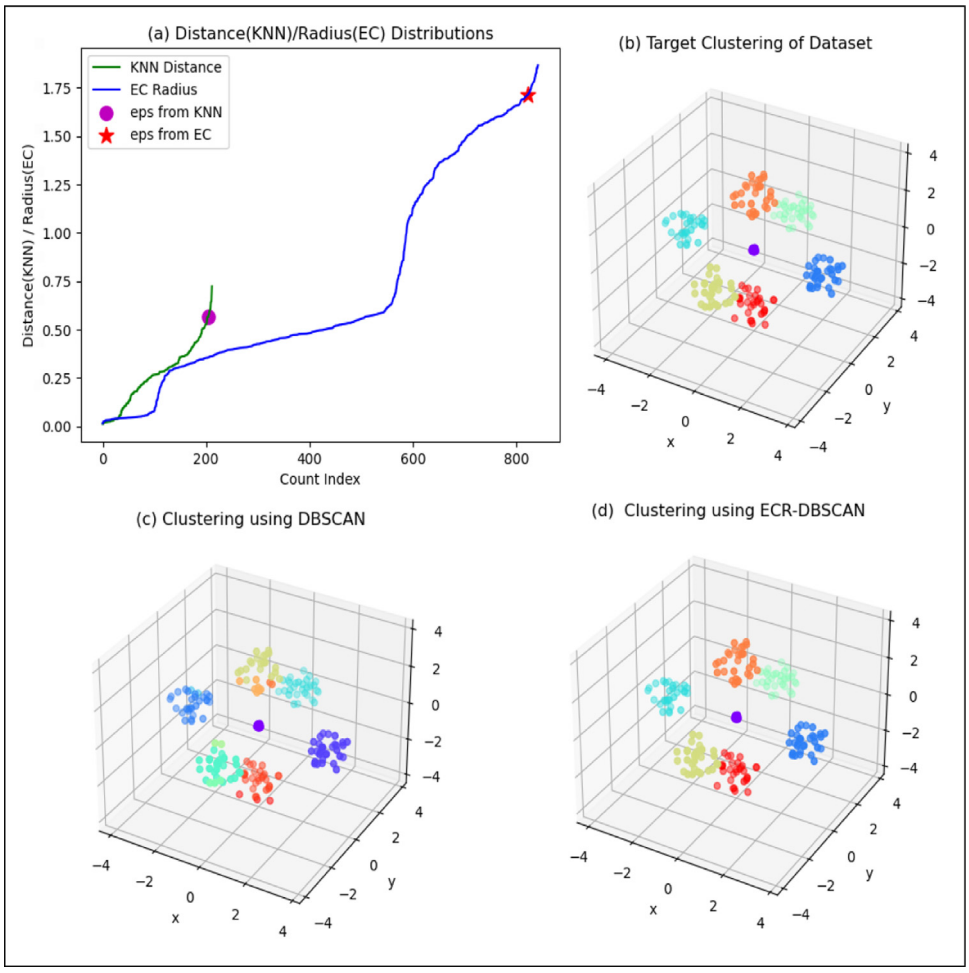

Fig. 3. (a) Distributions for both KNN distances & ECs radii with their knee values (b) Target clustering (c) Data set clustered with $DBSCAN$ (d) Data set clustered with $ECR-DBSCAN$ for $Hepta$ data set.

parameters ($epsilon$ and $minPoints$) in the light of computational geometry. In Section 3, the detailed proposed algorithm has been described. In order to check and verify the quality of our proposed $ECR-DBSCAN$, we have compared our results with traditional $DBSCAN$.

The paper has been organized in the following way. In Section 2, we will be discussing some preliminaries concepts during our work. Our proposed method have been described at Section 3. Our proposed method has been laid out in Section 3. In Section 4 & 5, we present our results and subsequent discussions respectively. Finally, conclusions have been drawn out in Section 6 followed by the acknowledgments and references.

2. Preliminaries

The basic concepts and terms used throughout this paper are given below.

2.1. Convex hull

The convex hull (hereafter CH) is the boundary among a set of data points such that all the points remain inside the boundary, with some points lying on the boundary line, considering as hull points. It is the smallest convex polygon that contains maximum data points in its interior and the rest points are the vertices of the polygon itself. Few popular algorithms are there to find the CH of a given set of data points, those are Graham Scan (Graham, 1972), Jarvis March (Jarvis, 1973) etc.

2.2. Voronoi diagram

The Voronoi diagram (hereafter VD) was invented by Russian mathematician Georgy Voronoi. It was defined as the partitioning of given set of points in a plane into some convex polygons or cells such that every point inside that convex polygon is closer to its generating point than any other point on the plane (Voronoi, 0000). Fortune (1987) proposed a sweep line algorithm in finding the VD for a given set of points. A sweep line sweeps across all the points and when it touches any point a beach line is created. Thus the point on which the beach line is created and the point that are being scanned should be equidistant. Let we have n distinct points on a plane. The site of points have been defined by the parameter $P = P1, P2, P3, P4, \ldots ..., Pn$. The task is to define the voronoi regions for these sites. The voronoi region for these site P which is defined as the subdivision in the plane such that every point inside that convex region or voronoi region is closer to its generating point than any other point on that plane.

2.3. Voronoi circle/empty circle

The empty circle (hereafter EC) or the voronoi circle (hereafter VC) is a circle that can be drawn for a given set of points on a plane such that no other point lies in the interior of the circle. The circle is constructed using the voronoi vertices (hereafter VVs) which are considered as the candidate centers of the circles, extracted from voronoi diagram for a given set of points, An early solution of the problem is an algorithm with worst-case running time is $O(n^3)$ (Dasarathy & White, 1975). Preparata and Shamos (1985) showed that, this problem can be

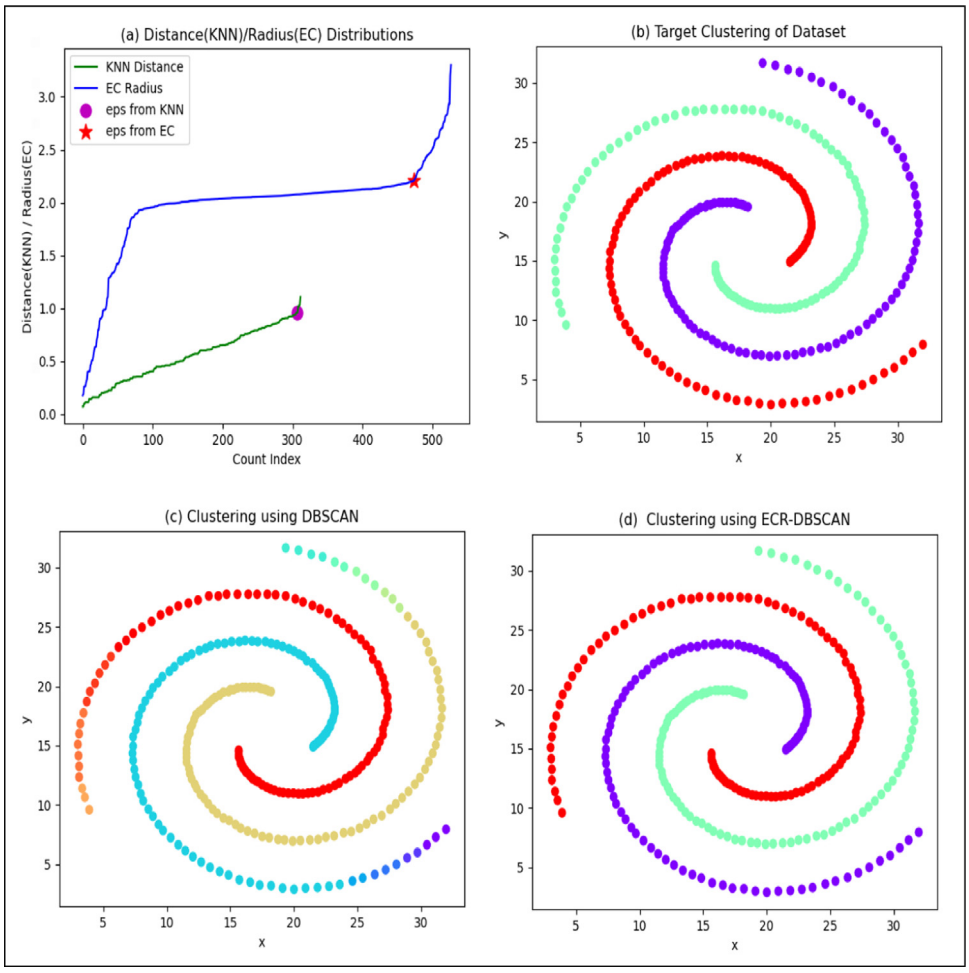

Fig. 4. (a) Distributions for both KNN distances & ECs radii with their knee values (b) Target clustering (c) Data set clustered with $DBSCAN$ (d) Data set clustered with $ECR-DBSCAN$ for *Spiral* data set.

solved in $O(nlogn)$ time with the help of voronoi diagram. The $O(nlogn)$ solution is optimal for this problem. For every voronoi vertex, there exists a unique circle centered at that vertex and passes through three or more sites.

2.4. DBSCAN algorithm

$DBSCAN$ (Ester et al., 1996) is one of the most applied clustering algorithm, which is density-based spatial clustering of applications with noise. It is one of the foremost common clustering algorithm works based on the density of objects. It depends on two important parameters viz., $epsilon$, the radius of neighborhood around a data point p and $minPts$, the minimum number of data points during a neighborhood to define a cluster. The $DBSCAN$ algorithm find the clusters in a density based approach where a core point consisting of $minPts$ number of points within the radius of $epsilon$ as well as a border does not contain $minPts$ number of points within $epsilon$ but one point within the radius of $epsilon$ should be a core point. The $DBSCAN$ algorithm classifies a point as noise point if that point is neither a border point nor core point.

2.5. K-nearest neighbors algorithm

$K-nearest$ neighbor's algorithm (Altman, 1992) (hereafter KNN) is another type of classification algorithm using which the k number of nearest points are selected from a particular point. The distances of those k number of nearest points are also extracted in the due process. In this paper, we use these KNN distances and achieve the $knee$ or $elbow$ value of this distances by using the elbow method (Thorndike, 1953), where a sharp change of the graphical plot of the distances have been measured.

2.6. PCA analysis

Principal Component Analysis (hereafter PCA) is an unsupervised linear transformation technique. PCA is a projection-based method (Chang, 1983) that transforms data by projecting it onto a set of orthogonal axes. Basically, PCA is the method for dimension reduction into a certain range.

2.7. Silhouette score

The Silhouette Score (hereafter SH) is basically a measure of ratio between inter and intra-cluster distances. In this paper the SH is taken into our consideration. This technique provides a succinct graphical representation of how well each object has been classified (Rousseeuw, 1987).

2.8. Manhattan distance & Euclidean distance

Manhattan distance (MD) is the distance between two points by aggregating the pairwise absolute difference between each variable, while Euclidean distance (ED) captures the same by aggregating the squared difference in each variable.

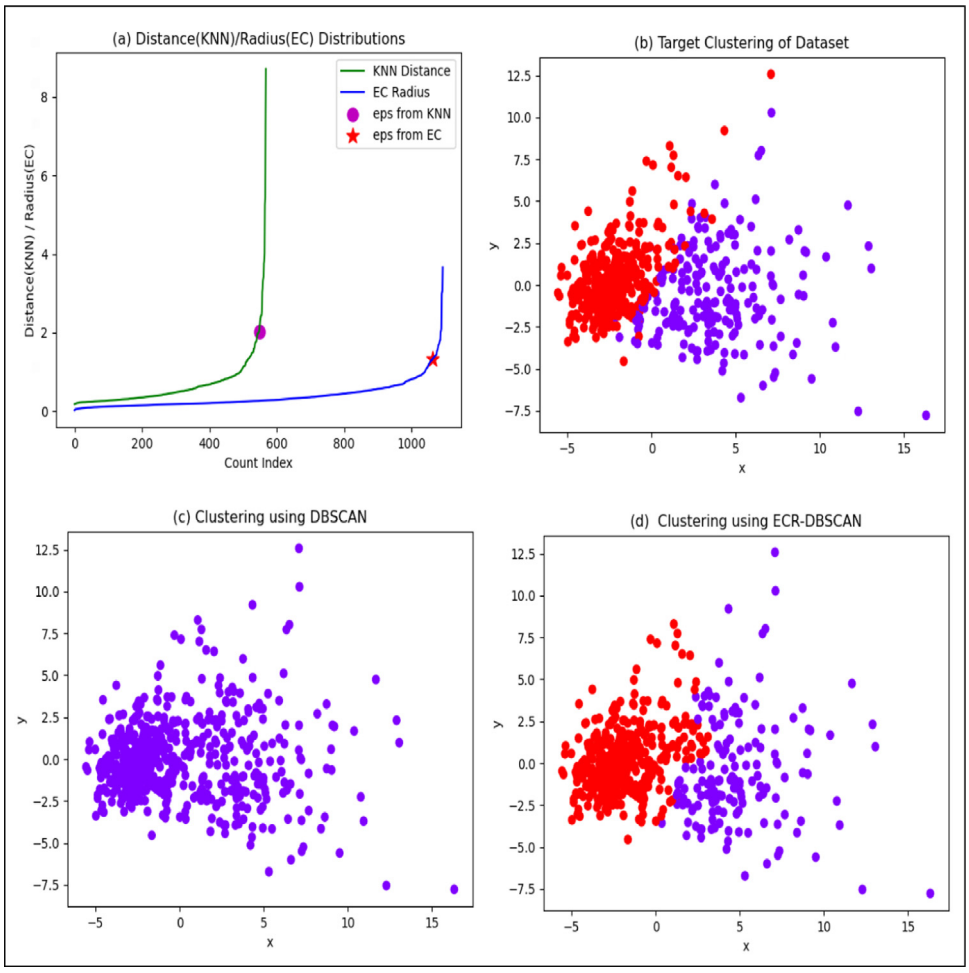

Fig. 5. (a) Distributions for both KNN distances & ECs radii with their knee values (b) Target clustering (c) Data set clustered with $DBSCAN$ (d) Data set clustered with $ECR-DBSCAN$ for *Breast_Cancer* data set.

2.9. Completeness & homogeneity

Completeness (CM) for any clustering is a measure of perfection that indicates the data points from the same label of class should be clustered or grouped together into the same cluster. Homogeneity (HM) for clustering is also a measure for perfect clustering. A clustering method satisfies homogeneity if it makes the groups or clusters with the data points such that each group/cluster contains only the data points having same class labels. The inter and intra cluster distance should be maximum and minimum respectively. Moreover, higher the values of completeness and homogeneity indicate better quality of clustering.

3. The algorithm ECR-DBSCAN

Our proposed algorithm is based on two important aspects. The first aspect is the selection of two key parameters of $DBSCAN$ algorithm, viz., (a) *epsilon* (hereafter *eps*) which is the radius of neighborhood around a point and (b) *minPoints* (hereafter *mp*) that represents the minimum number of data points with in the neighborhood with radius *eps*. The second part of our algorithm is to categorize noise points (if any) into the specific existing clusters according to the concept of centroid based clustering. Now we would like to describe our algorithm in details.

Let, P is any given data set. The convex hull $CH(P)$ and the voronoi diagram $VD(P)$ are constructed. From $VD(P)$, we have taken all the VVs and from $CH(P)$ all the hull points are collected. Hence, we stored them into two arrays viz., vor_ver&$hull_points$ respectively. It is relevant to note that we have considered the feasible zone for any real data set and accepted those VVs that remain inside the CH. The steps to find the VVs interior to the CH are provided in algorithm 1.

Algorithm 1: Steps to find the VVs interior to the CH

```
Compute CH(P)
Obtain the hull points and store into an array called hull_points
Compute VD(P)
Obtain all VVs and store into an array called vor_ver
for i ← 0 to length(vor_ver) do
    compute convex hull (data set P & vor_ver[i])
    if hull_points remains unchanged then
        the vor_ver[i] is interior
    else
        break;
    end
end
```

At the next stage, we have constructed the ECs for all the VVs that are interior to the CH. All the VVs inside CH are considered as the possible candidate centers of ECs. In order to construct the ECs, we have used following steps. (i) Construct $CH(P)$ and $VD(P)$; (ii) Extract all the VVs inside CH into an array *in_vor*; (iii) Those VVs are to be considered as the possible candidate centers of ECs; (iv) For all these VVs, we find the minimum distances such that no point overlap within this distance; (v) Finally, considering minimum distances as the radius of ECs for all VVs, we construct the corresponding ECs.

After the completion of the above two steps, we try to make a bridge between computational geometry and the spatial clustering. From all the above ECs, we have extracted their corresponding $radii$, then sorted them into an ascending order and later stored them in an array called $radius$. Next, we plotted the array $radius$ and subsequently we used the $elbow$ or $Knee$ method to find the $elbow$ / $knee$ value (where the graph has a sharp change). We denote this knee value as eps_ec. In our $ECR-DBSCAN$, we used this eps_ec as the $epsilon$ of traditional $DBSCAN$. Once we found eps_ec, we then turned our focus on finding the best mp for this eps_ec. To do it, keeping $eps = eps\_ec$, we run $DBSCAN$ varying the values of mp from 2 to 50. For each of the run we calculate the SH scores for the clustered data. We choose the value of mp where the SH is best (maximum). We denoted the best mp that gives maximum SH as $bestmin$. So finally, in brief, we successfully found two parameters of $DBSCAN$ algorithm, viz., $eps = eps\_ec$ and $mp = bestmin$ using computational geometry. Now, for a given data set P, we run $DBSCAN$ with eps_ec and $bestmin$. Let, P contains n data points and our $ECR-DBSCAN$ clustered it into k number of cluster with $m(m < n)$ number noise points(say). We will now focus into the second part of our algorithm that describes the proper distribution of these m number of noise points into existing k number of cluster so that the final clustering will be "$noise\ free$". To explain this, we will describe a particular example.

Let we got three clusters ($k = 3$) and m number of noise points for a data set of length n. Next, we compute the centroid for each the clusters. Next, we will compute the Euclidean distance between every noise points and every clusters' centroid. Here we have assigned one particular noise points into a specific cluster where the distance between that particular noise point and the specific cluster center is minimum. We can broadly describe this as: $k1$, $k2$, $k3$ are three clusters and their corresponding centroids are $c1$, $c2$, $c3$ as well as five (let $m = 5$) noise points are $np1$, $np2$, $np3$, $np4$, $np5$. Now for $np1$, compute three different distances between $np1$ and $c1$, $np1$ and $c2$ and $np1$ and $c3$. Among these three distances, let the distance between $np1$ and $c2$ is minimum. Hence, we assign the noise $np1$ into cluster $k2$. In a similar manner, we assign the rest noise points into the specific clusters and finally we got the clusters. The pseudo code for $ECR-DBSCAN$ is given at algorithm 2.

Algorithm 2: Pseudo Code for our proposed $ECR-DBSCAN$

Input: A set of data (P) containing n number of data points.
Output: The k number of cluster with the data points assigned into each specific cluster.
Data: **The following functions & variables are used:**
CH(P): Convex hull of data set P, **VD(P)**: Voronoi diagram of data set P, **VVs**: Voronoi vertices from $VD(P)$, **hull_in**: Function to compute the VVs interior to the $CH(P)$, **in_vor**: Array to store the all VVs interior to the convex hull, **ECs**: Empty circles, **circle_in**: Functions to compute the ECs for each in_vor as candidate centers, **radius**: array to store the radius of all interior hull ECs in ascending order constructed using $circle\_in$, **e_dist**: Function to compute Euclidean distances b/t pair of data points, **SH**: function to compute silhouette score, **noise_points**: array to store the all noise points obtained from $DBSCAN$ algorithm, **clus_cen**: array to store centroid of the data points in each cluster, **eps**: Parameter "$Epsilon$" for $DBSCAN$ algorithm, **mp**: Parameter "Minimum Samples" for $DBSCAN$ algorithm.

```
Take the data set P containing n data points with d number of features.
Compute the CH(P) and VD(P).
Call hull_in.
Extract the VVs and assign them into in_vor.
Call circle_in.
Take their radius and sort them in ascending order and assign into radius.
Make a graphical plot of radius.
Find the knee or elbow value (where the graph has a sharp change) of the graph using elbow/Knee method
Take this knee value as eps_ec (a parameter of DBSCAN).
for i ← 2 to 50 do
    compute DBSCAN of P with eps = eps_ec && mp = i
    compute SH
    if SH = maximum(SH) then
        store i into bestmin
    else
        continue;
    end
end
Run traditional DBSCAN using eps = eps_ec && mp = bestmin.
Extract the label of clustered data and store in label_initial.
Extract the noise points from label_initial having the label as '−1' and store them into noise_points.
Compute the centroid of each cluster and store into clus_cen.
for k ← 0 to length(noise_points) do
    compute e_dist (noise_points[k] , clus_cen)
    for each clus_cen do
        where e_dist (noise_points[k] , clus_cen) is minimum
        assign noise_points[k] into that cluster.
    end
end
Distribute all the noise points such way and get the final clustered data set with final label as label_final.
```

4. Results

Our proposed algorithm $ECR-DBSCAN$ has been implemented using PYTHON. The basic libraries for constructing CH, VD etc. are already available, but there are no libraries in PYTHON to find the ECs. To validate the result produced by our proposed method and to compare our method with traditional $DBSCAN$, we have computed few performance metrics, viz., CM, HM, ED and MD. We have used both artificial synthetic benchmark data and real world data for our experiment. It is relevant to mention here that for the data sets having more than three features or attributes we have used the PCA method to reduce the dimension of it.

seven sets of data, five synthetic and two real-life, were used for experimentation. These datasets are taken from UCI machine learning repository (Dua & Graff, 2017), GitHub data repository (Git-Hub) and FCPS data set (Ultsch, 2005). The artificial data sets are $Chainlink$, $Hepta$, $Spiral$, $D31$, $S-sets(S1)$ while the real-life data sets are $Breast\_Cancer$ and $Aggregation$. The details of these are given at Table 1.

Table 1
Description of dataset.

Data set	Data size	No. of attribute	No. of classes
Chainlink	1000	3	2
Hepta	212	3	7
Spiral	312	2	3
Breast_Cancer	569	30	2
Aggregation	788	2	7
D31	3100	2	31
S-sets (S1)	5000	2	15

First, we run our code for $Chainlink$ data set. We computed all the ECs (we got 4228 ECs inside the CH) and their $radii$ for the data set. We sort the $radii$ at increasing order and plot this array of $radii$. From this plot, we used $elbow/Knee$ method and found the $knee$ value of the plot. This $knee$ value is chosen as the $epsilon$ parameter for $ECR-DBSCAN$. In order to compare with normal $DBSCAN$, we also computed KNN distances for the same data set. In Fig. 1, both the distributions of KNN distances (green curve) and $ECs\ radii$ (blue curve) are plotted. The $knee$ value for the KNN distances is marked as the filled violet round shape, while the red asterisk sign is the $knee$ value coming from $ECs\ radii$. Hence, we found two $epsilon$ (eps): the round violet shape one for $DBSCAN$ and the red asterisk one for our

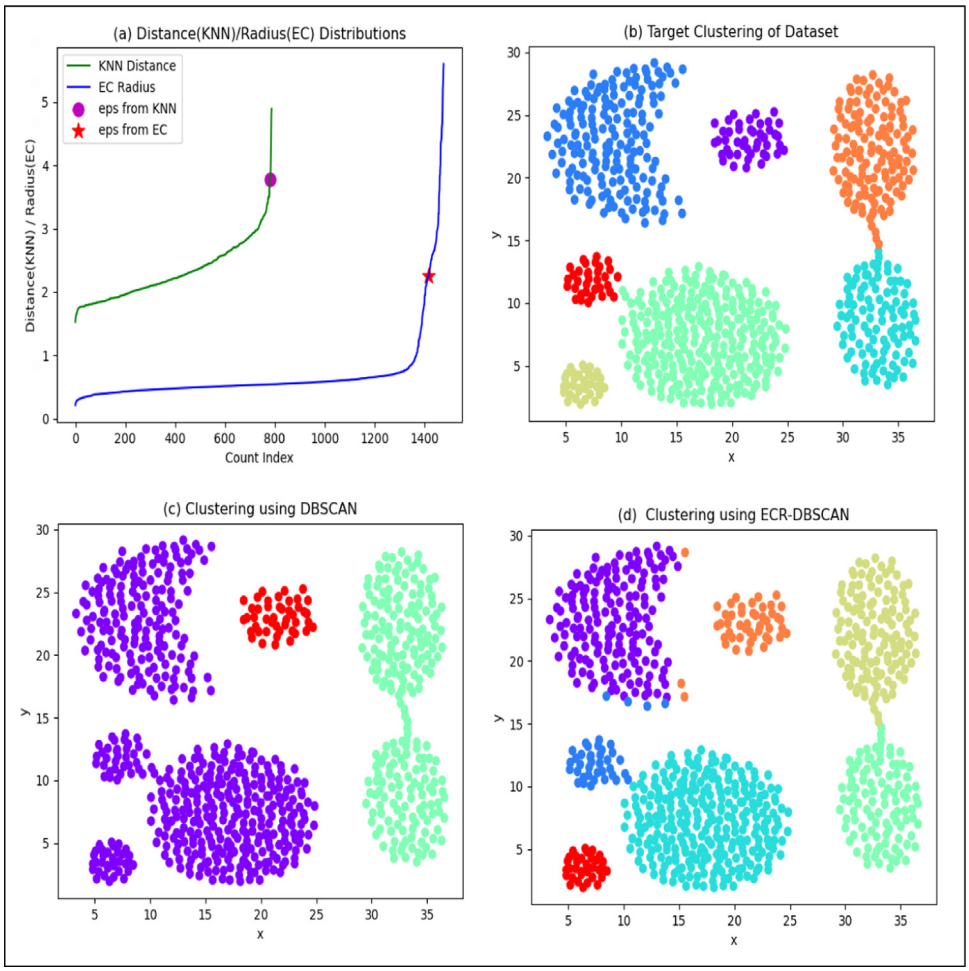

Fig. 6. (a) Distributions for both KNN distances & ECs radii with their knee values (b) Target clustering (c) Data set clustered with $DBSCAN$ (d) Data set clustered with $ECR-DBSCAN$ for $Aggregation$ data set.

$ECR-DBSCAN$. Subsequently, as per our algorithm at Section 3, we compute "$bestmp$". All the values of the parameters those are found by us are presented at Table 2. We will now present the clustering results for both of these two methods. In Fig. 2(a) the simple scatter plot of $Chainlink$ data set are shown, Fig. 2(b) represents the clustered data set using target or ground truth label, in Fig. 2(c) shows the clustering result for traditional $DBSCAN$ and finally, Fig. 2(d) represents the clustering results for our proposed $ECR-DBSCAN$. From Fig. 2(d) and Table 2 it is clearly visible that, our proposed $ECR-DBSCAN$ provide perfect clustering as similar as the ground truth of $Chainlink$ (Fig. 2(b)), while, the normal $DBSCAN$ fails to form the clusters (Fig. 2(c)) as in target (Fig. 2(b)). This clearly shows the novelty of our proposed algorithm.

In a similar manner, we now ran our code with the rest six data sets. In Fig. 3, we proceed to show the result for $Hepta$ data set. As described for Fig. 1, at Fig. 3(a) the same has been visualized for $Hepta$ data set. However, Fig. 3(b) represents the clustered data set using target or ground truth label for $Hepta$, Fig. 3(c) shows the clustering result of $Hepta$ for traditional $DBSCAN$ and finally, Fig. 3(d)shows the clustering results of $Hepta$ for our proposed $ECR-DBSCAN$. Similarly, the results for $Spiral$, $Breast\_Cancer$, $Aggregation$, $D31$ and $S-Sets(S1)$ data set are shown at Fig. 4, Fig. 5, Fig. 6, Fig. 7 and Fig. 8 respectively. All the values of the parameters those are extracted by us for these seven data sets have been laid out in Table 2.

5. Discussions

In Table 2, it is clear that for all data sets used in experiment, $ECR-DBSCAN$ absolutely exceeds of $DBSCAN$ on (a) getting low ED and MD (b) getting high CM and HM. Hence, with respect to all performance metrics, our proposed $ECR-DBSCAN$ is significantly better than traditional $DBSCAN$. Most importantly, from Fig. 2 to Fig. 8, we have discovered that, $ECR-DBSCAN$ provides all the target clusters prominently, while, the traditional $DBSCAN$ fails to recognize as well as unsuccessful in finding the exact numbers of target clusters. Now, we clarify the differences of our proposed approach with few of the existing literature.

Recently, in context of large data, $Rough-DBSCAN\&I-DBSCAN$ were introduced in Luchi et al. (2019). It is noted that $Rough-DBSCAN\&I-DBSCAN$ can perform better on large data sets without a need of additional parameter for $DBSCAN$ algorithm. On a similar manner, we have clearly found the quality improvement of $DBSCAN$ at our $ECR-DBSCAN$ without using any additional parameter. In the context of modifying $DBSCAN$, Ankerst et al. (1999) has proposed a concept of cluster ordering that can retrieve the cluster information and internal structure for finding the density based structure, if any. Contrarily, in our proposed method, we have concentrated on searching the "best" $epsilon$ for $DBSCAN$ in context of computational geometry. A non parametric clustering technique was introduced in Averbuch-Elor et al. (2020), where, instead of finding parameter, the authors considered the entire clustering process as a layered architecture. In the same work, the border points are considered as an external layer and it is claimed that this layer can explicitly identify the actual clusters. In contrast of Averbuch-Elor et al. (2020), instead of considering border points as a separate layer, we have redistributed the noise points obtained from $DBSCAN$ into the existing clusters with distance-neighborhood approach. An improved $DBSCAN$ with hierarchical

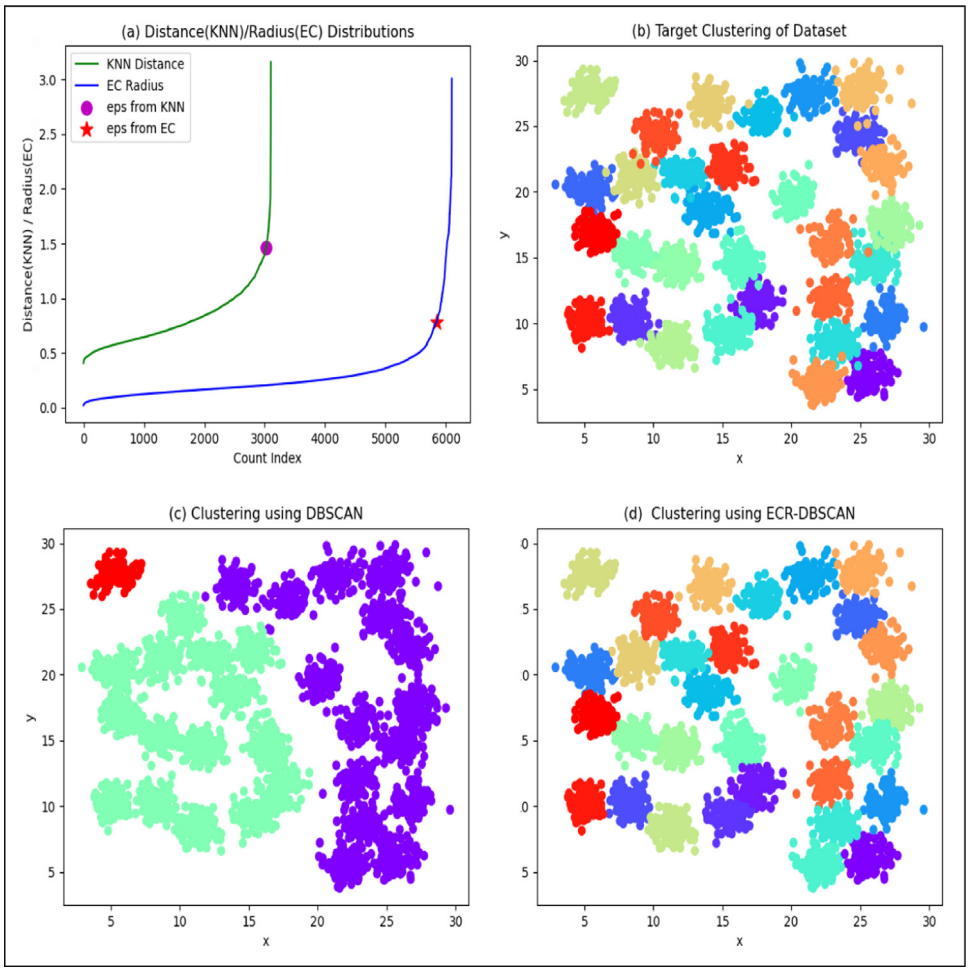

Fig. 7. (a) Distributions for both KNN distances & ECs radii with their knee values (b) Target clustering (c) Data set clustered with $DBSCAN$ (d) Data set clustered with $ECR-DBSCAN$ for $D31$ data set.

Table 2
Multi-parameter analysis performed in five different datasets. The extracted values of eps, mp, noc, CM, HM, ED, MD are given here. See text for details.

Data Set Name	Algorithm	*eps*	*mp*	*noc*	*CM*	*HM*	*ED*	*MD*
Chainlink	DBSCAN	0.082	2	14	0.32	0.99	170.00	4266.00
	ECR-DBSCAN	**0.510**	**2**	**2**	**1.00**	**1.00**	**0.00**	**0.00**
Hepta	DBSCAN	0.565	2	13	0.88	1.00	35.00	323.00
	ECR-DBSCAN	**1.712**	**2**	**7**	**1.00**	**1.00**	**0.00**	**0.00**
Spiral	DBSCAN	0.955	2	14	0.68	0.99	150.00	2311.00
	ECR-DBSCAN	**2.206**	**2**	**3**	**1.00**	**1.00**	**25.00**	**418.00**
Breast_Cancer	DBSCAN	2.762	23	1	1.00	0.00	19.00	357.00
	ECR-DBSCAN	**1.315**	**23**	**2**	**1.00**	**1.00**	**8.00**	**63.00**
Aggregation	DBSCAN	3.775	23	3	0.99	0.47	82.00	2041.00
	ECR-DBSCAN	**2.249**	**23**	**7**	**1.00**	**1.00**	**53.00**	**1146.00**
D31	DBSCAN	1.461	23	3	0.99	0.24	945.00	44803.00
	ECR-DBSCAN	**0.783**	**23**	**31**	**1.00**	**1.00**	**201.00**	**5147.00**
S-Sets(S1)	DBSCAN	36037.129	9	10	0.99	0.81	359.00	15224.00
	ECR-DBSCAN	**26800.329**	**9**	**15**	**1.00**	**1.00**	**246.00**	**12208.00**

technique named as $HDBSCAN$ was established in Campello et al. (2015). In $HDBSCAN$, the final clusters are derived within every level of hierarchy that includes the concept of cluster tree. However, $HDBSCAN$ was a non parametric clustering technique which combines the idea of the hierarchical & density based clustering. Based on K-nearest neighbor algorithm (KNN), a domain adaptive approach was introduced in Chen and Yu (2021) to define a new density measurement method. They detected various density peaks of various density regions. It is relevant to mention that, none of these above four works are directly comparable with our $ECR-DBSCAN$ as none of these works is related to *epsilon* finding. In addition, none of them used computational geometry in order to find *epsilon*. We now compare our $ECR-DBSCAN$ with the existing work of Pavlis et al. (2017), where, they introduced a sparse graph representation with a KNN distance matrix and subsequent decomposition which was done with depth first search (DFS) algorithm. The work Pavlis et al. (2017) was applied in a retail estimation model. They show the improvement on clustering using a control switch on radius parameter into local scale. Hence, there is a similarity in between our work and Pavlis et al. (2017). Both the ideas were focused on the key parameter *epsilon* although

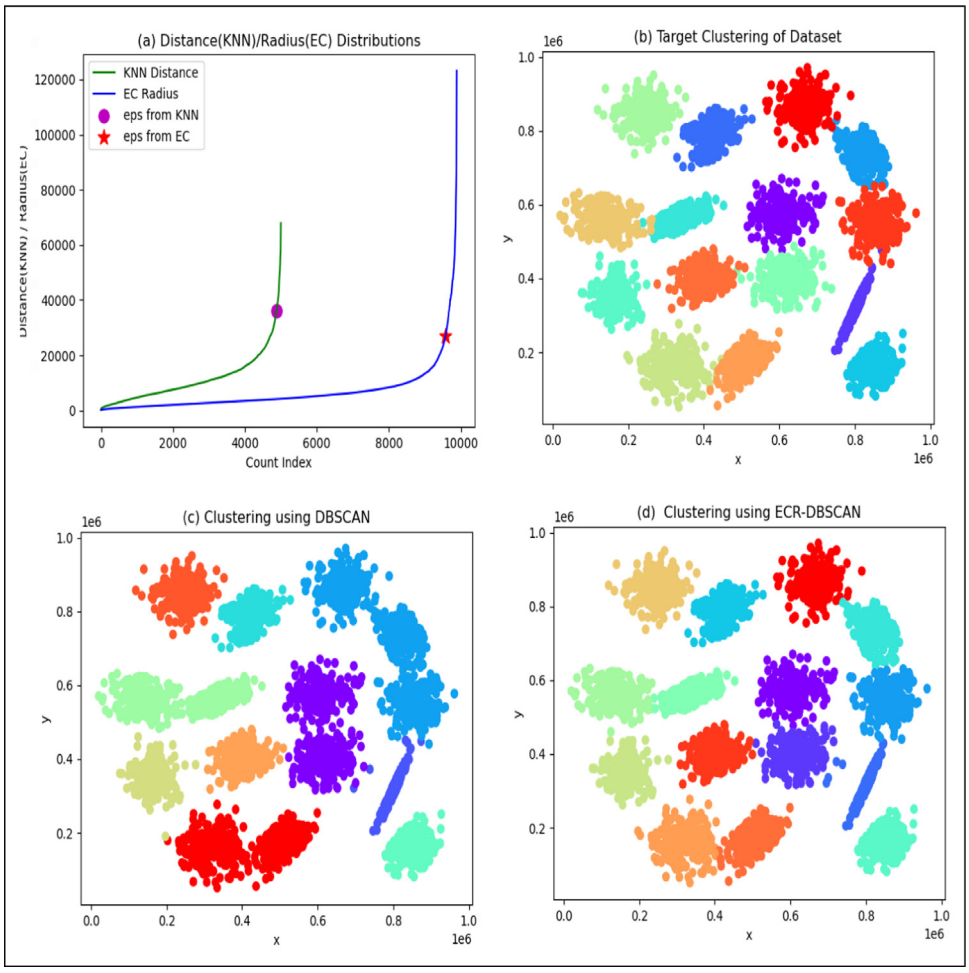

Fig. 8. (a) Distributions for both KNN distances & ECs radii with their knee values (b) Target clustering (c) Data set clustered with $DBSCAN$ (d) Data set clustered with $ECR-DBSCAN$ for $S-sets(S1)$ data set.

we did not use the iterative process and sparse graph representation of clusters. Rather, we constructed the empty circles using each data points within the convex hull of the given data points. Instead of KNN based approach we have incorporated the radius (of empty circles) based concept to find the optimal value of *epsilon* parameter. While doing so we have achieved an effective improvement on clustering quality over the actual $DBSCAN$ method.

6. Conclusions

This paper proposed a novel approach on the improvement of traditional $DBSCAN$ by coupling with computational geometry. Our hybrid approach consisting empty circles (ECs) and $DBSCAN$ clustering algorithm as $ECR-DBSCAN$ to choose quickly and intelligently the input parameters $epsilon(eps)$ and $MinPts(mp)$ for $DBSCAN$. We got two significant phenomenon at our proposed method in compare with normal $DBSCAN$. First, the pattern of both the distributions ($DBSCAN's$ KNN distances and our proposed $ECR-DBSCAN's$ ECs $radii$) are more or less same, however, the knee values are different for both of them. Second, the above experiments successfully shows our $ECR-DBSCAN$ capturing target clusters efficiently in comparison to normal $DBSCAN$ for each of the data set. From all of our experimented results, it is evident that the redistribution of the noise points that we used here works successfully. It is pertinent to mention here that our proposed clustering method has some limitations. If the dimension of the data set is more than five, it takes huge time to compute all the ECs. Hence, at our present work we used PCA to reduce the dimension of those data having more than five attributes. However, for the data set having less than or equal to five attributes, even without applying PCA, $ECR-DBSCAN$ works perfectly. One of the key advantage found at our $ECR-DBSCAN$ is that it is an improved version of traditional $DBSCAN$ with respect to the clustering quality. The nearest neighbor approach is already established into various existing research work, but, for the first time, we have used computational geometry in context of finding the "best" $epsilon$ and subsequent $minPoints$. On the other hand, we would also like to draw some shortfall of our $ECR-DBSCAN$. There are plenty of modified approaches on $DBSCAN$ are available in literature using the idea of density distribution of data set. But, in our work, we have not focused on it. Rather, we have concentrated on the selection of key parameters. Beside these, we did not extend our experiment on the mixed data set having different density distributions. Also, in the present version of this work, we did not consider any separate layer/s for handling noise points which improves the quality of the clusters. Our next goal is to investigate and improve the method on computing ECs $radii$ for the data set having large number of attributes in such a way that it would be able to compute quickly. Thus in the near future we plan to continue our quest to improvise our findings and come up with further discoveries.

CRediT authorship contribution statement

Kinsuk Giri: Conceptualization, Methodology, Software, Validation, Writing – review & editing, Supervision. **Tuhin Kr. Biswas:** Software, Visualization, Writing – original draft. **Pritisha Sarkar:** Resources, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge Dipabaly Bhattacherya of IIHM Kolkata for helping on language corrections of this paper. KG acknowledges the High Performance Computing System (HPC) of NITTTR Kolkata for using it as the computational resource for this paper. Finally the authors also thank the anonymous referee for his/her invaluable comments that helped them to improve the quality of the paper.

References

Altman, N. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*, 175–185.

Ankerst, M., Breunig, M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *SIGMOD Record*, *28*, 49–60. http://dx.doi.org/10.1145/304182.304187.

Averbuch-Elor, H., Bar, N., & Cohen-Or, D. (2020). Border-peeling clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(7), 1791–1797. http://dx.doi.org/10.1109/TPAMI.2019.2924953.

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, *60*, 208–221. http://dx.doi.org/10.1016/j.datak.2006.01.013.

Campello, R. J. G. B., Moulavi, D., Zimek, A., & Sander, J. (2015). Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data*, *10*(1), http://dx.doi.org/10.1145/2733381.

Chang, W. C. (1983). On using principal components before separating a mixture of two multivariate normal distributions. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *32*(3), 267–275, URL: http://www.jstor.org/stable/2347949.

Chen, J., & Yu, P. S. (2021). A domain adaptive density clustering algorithm for data with varying density distribution. *IEEE Transactions on Knowledge and Data Engineering*, *33*, 2310–2321.

Dasarathy, B., & White, L. (1975). On some maximin location of classifier problems. In *Computer science conference*. Washington. (Unpublished Lecture).

Dua, D., & Graff, C. (2017). *UCI machine learning repository*. Irvine: University of California, School of Information and Computer Sciences, URL: http://archive.ics.uci.edu/ml.

Edla, D. R., & Jana, P. K. (2012). A prototype-based modified DBSCAN for gene clustering. *Procedia Technology*, *6*, 485–492. http://dx.doi.org/10.1016/j.protcy.2012.10.058, 2nd International Conference on Communication, Computing & amp; Security [ICCCS-2012].

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining* (pp. 226–231). AAAI Press.

Fortune, S. (1987). A sweepline algorithm for voronoi diagrams. *Algorithmica*, *2*, 153–174.

Graham, R. (1972). An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, *1*, 132–133.

He, Y., Tan, H., Luo, W., Mao, H., Ma, D., Feng, S., & Fan, J. (2011). MR-DBSCAN: An efficient parallel density-based clustering algorithm using MapReduce. In *2011 IEEE 17th international conference on parallel and distributed systems* (pp. 473–480).

Jang, J., & Jiang, H. (2019). DBSCAN++: Towards fast and scalable density clustering. In *International conference on machine learning* (pp. 3019–3029). PMLR.

Jarvis, R. (1973). On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, *2*(1), 18–21. http://dx.doi.org/10.1016/0020-0190(73)90020-3.

Karami, A., & Johansson, R. (2014). Choosing DBSCAN parameters automatically using differential evolution. *International Journal of Computer Applications*, *91*, http://dx.doi.org/10.5120/15890-5059.

Khan, M. M. R., Siddique, M. A., Arif, R., & Oishe, M. (2018). *ADBSCAN: Adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities*. http://dx.doi.org/10.1109/CEEICT.2018.8628138.

Lai, W., Zhou, M., Hu, F., Bian, K., & Song, Q. (2019). A new DBSCAN parameters determination method based on improved MVO. *IEEE Access*, *7*, 104085–104095. http://dx.doi.org/10.1109/ACCESS.2019.2931334.

Liu, X., Yang, Q., & He, L. (2017). A novel DBSCAN with entropy and probability for mixed data. *Cluster Computing*, *20*(2), 1313–1323.

Liu, P., Zhou, D., & Wu, N. (2007). VDBSCAN: Varied density based spatial clustering of applications with noise. In *Proc. 2007 international conference on service systems and service management* (pp. 1–4). http://dx.doi.org/10.1109/ICSSSM.2007.4280175.

Luchi, D., Rodrigues, A. L., & Varejão, F. M. (2019). Sampling approaches for applying DBSCAN to large datasets. *Pattern Recognition Letters*, *117*, 90–96.

Pavlis, M., Dolega, L., & Singleton, A. (2017). A modified DBSCAN clustering method to estimate retail center extent: Clustering retail agglomerations. *Geographical Analysis*, *50*, http://dx.doi.org/10.1111/gean.12138.

Preparata, F. P., & Shamos, M. I. (1985). *Computational geometry: An introduction*. Berlin, Heidelberg: Springer-Verlag.

Ram, A., Sunita, J., Jalal, A., & Manoj, K. (2010). A density based algorithm for discovering density varied clusters in large spatial databases. *International Journal of Computer Applications*, *3*, http://dx.doi.org/10.5120/739-1038.

Ren, Y., Liu, X., & Liu, W. (2012). DBCAMM: A novel density based clustering algorithm via using the mahalanobis metric. *Applied Soft Computing*, *12*, 1542–1554. http://dx.doi.org/10.1016/j.asoc.2011.12.015.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. http://dx.doi.org/10.1016/0377-0427(87)90125-7.

Ruiz, C., Spiliopoulou, M., & Menasalvas, E. (2007). C-DBSCAN: Density-based clustering with constraints. In A. An, J. Stefanowski, S. Ramanna, C. J. Butz, W. Pedrycz, & G. Wang (Eds.), *Rough sets, fuzzy sets, data mining and granular computing* (pp. 216–223). Berlin, Heidelberg: Springer Berlin Heidelberg.

Sander, J., Ester, M., Kriegel, H., & Xu, X. (1998). Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery*, *2*, 169–194.

Thorndike, R. (1953). Who belongs in the family? *Psychometrika*, *18*(4), 267–276. http://dx.doi.org/10.1007/BF02289263.

Ultsch, A. (2005). Fundamental clustering problems suite (FCPS). http://dx.doi.org/10.13140/RG.2.1.2394.5446.

Voronoi, G. (0000). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. premier mémoire. sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, *1908* 97–102.