Int. J. Nonlinear Anal. Appl. 16 (2025) 9, 7–13
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2024.33345.4962

A density-based clustering method with calculating the Eps parameter

Mahshid Asghari Sorkhi[a], Mohsen Rabbani[b,*], Ebrahim Akbari[a], Homayun Motameni[a]

[a]Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari,Iran

[b]Department of Applied Mathematics, Sari Branch, Islamic Azad University, Sari, Iran

(Communicated by Seyed Hossein Siadati)

Abstract

With regard to the non-linear nature of real-life data, their clusters' shapes are non-convex and unfortunately, some clustering methods cannot identify non-convex clusters and this is a challenge. Density-based clustering methods could be a solution to this problem. Among all methods of this type, the DBSCAN algorithm can cluster data with different shapes, sizes, and densities and also identify noise points. However, owing to the use of static input parameters-the neighbourhood radius (Eps) and the minimum value for cluster formation ($MinPts$)- this algorithm has some problems such as the difficulty in accurately determining these parameters in high-dimensional data sets and not recognizing clusters with different densities. Accordingly, this paper presents a density clustering algorithm, which requires minimal input parameters and one of its main parameters is Eps, which is automatically calculated based on the k-nearest neighbours of points and its value is different for each cluster. To evaluate the effectiveness of the proposed algorithm, some experiments were conducted. The obtained results showed the effectiveness and efficiency of the presented algorithm regarding the correct identification of clusters with the desired shape, size, and density. In addition, the proposed algorithm was found effective in estimating the number of clusters in most of the data sets considered in this study.

Keywords: Clustering, Density-based clustering, DBSCAN algorithm, k nearest neighbors
2020 MSC: 62H30

1 Introduction

Clustering, as an unsupervised learning method, is one of the main data mining techniques. It refers to the process of grouping a set of data and putting them into classes of similar samples. A cluster is a set of data that are most similar to each other but different from the data of other clusters [1]. One of the clustering approaches in data mining is the use of density-based clustering algorithms [2, 13, 4, 22]. These algorithms have three advantages: they do not limit themselves to the form of clusters, they are easy to understand, and they do not need to determine the number of clusters in advance. The relevant literature consists of many density-based clustering algorithms [20]. One of the famous algorithms in this field is DBSCAN, which was introduced by Ester in 1996 [7, 6]. Density-based clustering algorithms generally need input parameters to start their work. It is very difficult for the user to determine these

*Corresponding author
Email addresses: m_a_sorkhi@yahoo.com (Mahshid Asghari Sorkhi), mrabbani@iausari.ac.ir (Mohsen Rabbani), akbari@iausari.ac.ir (Ebrahim Akbari), motameni@iausari.ac.ir (Homayun Motameni)

Received: February 2024 *Accepted:* May 2024

parameters for large data. Therefore, the automatic determination of these parameters is one of researchers' concerns in this field. On the other hand, there are only a few methods that automatically calculate the input parameters. One of the problems of the DBSCAN algorithm is its incapability of clustering high-density data sets, which is because the static Eps and $MinPts$ values cannot be suitable for all clusters. To solve this problem, this paper proposes a method in which the input parameter k is the number of nearest neighbours of points and Eps is the radius of the neighbourhood. In this method, Eps is determined automatically and the value of Eps changes for each cluster. Therefore, it applies to clustering data sets with high density. The following are the achievements of this paper:

1. The proposed method works on most data sets because it uses only the k-nearest neighbours.
2. The proposed method calculates the Eps parameter as radius neighbourhood automatically and its value is different for each cluster.

The next parts of this paper are structured as follows. Section 2 reviews the related work. Then, Section 3 presents a density-based clustering method. Afterwards, Section 4 examines several artificial and real data sets. Finally, Section 5 concludes the study and suggests directions for future work.

2 Related work

DBSCAN [7] is a basic method in density-based clustering, which uses two parameters: Eps and $MinPts$. This method has some disadvantages, for example, its input parameters are static, which makes the difficult to be determined manually in large data sets. In addition, when this method is used in clusters that are close to each other, the boundary data cannot be identified correctly. To improve the DBSCAN method, some other density-based clustering methods have been presented in the literature, which have succeeded in eliminating some of its disadvantages. Among them, some methods use only the k-nearest neighbours of points for data clustering [12, 17]. For example, Jiaxin Qian et al. [15] proposed a multi-density DBSCAN (MDBSCAN) algorithm based on the relative density, which extracts the low-density points in the data set, then detects the real clusters from the points with low density, and finally uses DBSCAN to cluster the rest of the points. One of the disadvantages of this method is that when the algorithm is faced with large data sets, the setting range of parameter k is large. Xiaogang Huang et al. [10] proposed the Grit-DBSCAN algorithm, which uses a grid tree to form the grids, and then suggested a method to reduce the distance calculations. Xiang Zhang et al. [23] developed an algorithm, called the Whale Optimization Algorithm-DBSCAN (WOA-DBSCAN), which uses WOA to find the input parameters of DBSCAN. Bing Ma et al. [14] proposed the K-DBSCAN algorithm, which detects input parameters of DBSCAN algorithm and finds the core points. In another study, Ziqing Wang et al. [21] proposed the Adaptive Multi-density DBSCAN (AMD-DBSCAN) algorithm, which uses a method to conform the input parameters of DBSCAN, i.e., Eps and $MinPts$. This method was applied to multi-density data sets. Moreover, the variance of the number of neighbors (VNN) traversed the density difference among clusters. Avory Bryant et al. [3] proposed the RNN-DBSCAN algorithm, which uses the reverse nearest neighbors to measure the density of points. K. .Ahmed Fahim [8] developed the E-DBSCAN algorithm using a dynamic radius to find clusters of any density. This method considers a density value for each data, then checks the data that have similar densities in a neighborhood radius regarding the cluster existence. Zeinab Falahiazar et al. [9] proposed the Dynamic Multi-Objective Genetic Algorithm-DBSCAN (DMOGA-DBSCAN) algorithm. It detects the parameters of DBSCAN automatically and dynamically. In another research, Igor de Moura Ventorim et al. [5] introduced the BIRCHSCAN algorithm, which reduces the set of points to cluster the data set. As the above review reveals, many methods have been proposed to improve the DBSCAN method. The uniqueness of the method proposed in the present paper is that, in this method, the radius of the neighbourhood is defined dynamically and its value is different for each cluster. Therefore, it is easy to determine it in data sets with large dimensions. In addition, unlike other methods, the value of neighbourhood radius is determined statically.

3 The proposed algorithm

Given that the basis of the proposed method is the DBSCAN algorithm, first, the distance between two points x_i and x_j were calculated as follows [16]:

$$D(x_i, x_j) = \sqrt{\sum_{t=1}^{d} (x_{it} - x_{jt})^2} \tag{3.1}$$

where (X=$\{x_1, x_2, ..., x_n\}$), x_i, x_j are two points of X data set, x_i=$(x_{i1}, x_{i2}, ..., x_{id})$, x_j=$(x_{j1}, x_{j2}, ..., x_{jd})$, and d is the dimensionality of points. In this method, a criterion was defined to measure the density of points as follows:

$$density(x) = \sum_{p \in KNN(x)} D(x, p). \tag{3.2}$$

Equation (3.2) is calculated as the total distance of each point from its k-nearest neighbor. In addition, the neighborhood radius was defined as follows:

$$Eps = \frac{\sum_{q \in F_p} dist\,(p, q)}{K}, \tag{3.3}$$

where F_p is the group of K nearest neighbors of p. In Equation (3.3), *Eps* was calculated as the average distance of the dense point from its k-nearest neighbor. To allow the border points to enter the cluster, the neighborhood radius *Eps* was reduced and a new neighborhood radius, namely *boundaryEps*, was obtained as follows:

$$boundaryEps = \frac{Eps}{2^i} \tag{3.4}$$

where the value of i starts from 1; then, in the process of the algorithm, every time the radius of the neighborhood decreases, its value increases by one unit. A threshold was additionally defined for the neighborhood radius value as follows:

$$\text{threshold} = \frac{\sum_{x \in H_p} dist\,(p, x)}{[0.5 \times K]}, \tag{3.5}$$

where H_p is the group of $[0.5 \times K]$ nearest neighbors of p. In Equation (3.5), the *threshold* was calculated as the average distance of the dense point from its $[0.5 \times K]$ nearest neighbor.

The proposed algorithm operates through the following steps. First, the density of points in the data set is measured using Equation (3.2). After that, the data with the lowest density is identified as the dense point in the data set. Then, the cluster is created and the dense point enters the cluster. In this step, the neighbourhood radius is calculated for this cluster using Equation (3.3). The direct density reachable points from the dense point to the neighbourhood radius *Eps* are retrieved. If there is at least one point in the neighbouring radius of *Eps*, then the cluster is formed and the dense point is entered into the cluster; otherwise, that point is recognized as noise. In this step, the density reachable from the dense point with the new neighbourhood radius *boundaryEps* is also recovered. This is the shrinking of the neighbourhood radius to a size that does not come down less than the threshold value defined for *Eps* in Equation (3.5). Otherwise, the threshold value is considered to be the new neighbourhood radius value and the process continues. That way, all border points enter the cluster and the cluster is completed. Then, the data of the completed cluster is removed from the data set. These steps are repeated for the remaining data set until the whole data is checked and the final clusters are created. Algorithm 1 describes the process.

Algorithm 1 proposed algorithm

Require: X (Dataset), K (The number of neighbors of each point)
Ensure: Final Clusters
Initialize c=0 (c is the number of clusters)
while X is not empty **do**
  c=c+1
  Calculate the value of *density* for all points by means of Equation (3.2) using the k-nearest neighbors of points.
  Find a point that has the minimum value of *density* (the dense point)
  Calculate the *Eps* value using the Equation (3.3)
  Create the new cluster
  Create all the direct density reachable points from the dense point with radius neighborhood *Eps*
  Create all the density reachable points from the dense point with radius neighborhood *boundryEps*
  Remove the points included in the cluster
end while

Regarding the analysis of the time complexity of the method, since the algorithm detects the k-nearest neighbours of points, it takes o($Knlogn$) [19]. The calculation of *density* can be performed in o(Kn). The process of finding direct density reachable and density reachable points takes o(n). Therefore, the time complexity of the proposed method is o(Kn^2logn).

4 Experimental results

In this section, the results of the proposed algorithm, the DBSCAN, and the RNN-DBSCAN algorithm are evaluated on five artificial and five real data sets regarding two cluster quality evaluation criteria, ARI [11] and NMI [18]. Table 1 shows the characteristics of these data sets. The data sets used in this paper are available on the GitHub repository (https://github.com/mlyizhang/Clustering-Datasets). ARI and NMI are two criteria for evaluating the quality of clustering, whose values are between zero and one; closer values to one offer more accurate clustering.

The parameters in the proposed algorithm, in the DBSCAN algorithm, and in the RNN-DBSCAN algorithm are determined as follows. In DBSCAN, the value of $MinPts$ is determined based on the K-distance graph; it is set to (2*data set dimension), and the Eps values are detected where the graph has elbows. On the other hand, in the proposed algorithm and in the RNN-DBSCAN algorithm, the value of parameter K, as the number of nearest neighbours, is selected experimentally.

Table 1: Distribution of data sets

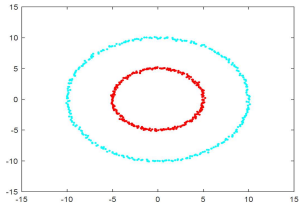
Data set	Data size (n)	Dimensionality (d)	Number of clusters (K)
2circles	600	2	2
D31	3100	2	31
Aggregation	788	2	7
D1	87	2	3
zelink6	1238	2	3
Iris	150	4	3
Seeds	210	7	3
vote	435	16	2
Zoo	101	16	7
Landsat	2000	36	6

4.1 Results on artificial datasets

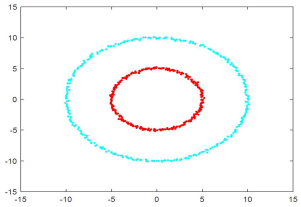
In this part, the proposed algorithm is compared with the DBSCAN and RNN-DBSCAN algorithms on five artificial data sets (2circles, D31, Aggregation, D1, and zelink6) regarding the NMI and ARI criteria. Table 2 shows the results of this evaluation. In this table, the values that have the highest accuracy are marked in bold. The defined parameter value of the proposed algorithm for 2circles, D31, Aggregation, D1, and zelink6 are respectively: $(K = 30)$,$(K = 32)$,$(K = 11)$,$(K = 7)$,$(K = 10)$; for DBSCAN, they are set to $(MinPts = 5,\ Eps = 1.8)$, $(MinPts = 20,\ Eps = 0.7)$,$(MinPts = 10,\ Eps = 1.6)$,$(MinPts = 10,\ Eps = 0.7)$,$(MinPts = 55,\ Eps = 0.05)$; and for RNN-DBSCAN, they are set to $(K = 11)$,$(K = 22)$,$(K = 7)$,$(K = 6)$,$(K = 13)$.The results showed that the proposed algorithm had higher accuracy than the DBSCAN algorithm in all four data sets. In general, the presented method showed an acceptable performance compared to the data sets studied in this research. In addition, in Fig. 1, the final clustering and the clusters created with the proposed algorithm and with the DBSCAN and RNN-DBSCAN algorithms were shown on each of the five artificial data sets, respectively. As can be seen, the proposed method was able to identify the correct clusters in all these data sets, but the RNN-DBSCAN algorithm did not perform the correct clustering in the zelink6 data set. The DBSCAN algorithm performed the correct clustering but in some data sets, such as the D31 data set, some border points were detected as noise.

Table 2: efficiency of different clustering algorithms on the five artificial data sets

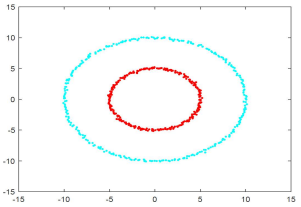
Data set	proposed algorithm		DBSCAN		RNN-DBSCAN	
	ARI	NMI	ARI	NMI	ARI	NMI
2circles	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
D31	**0.9427**	**0.9615**	0.6463	0.8612	0.9163	0.9567
Aggregation	**1.0000**	**1.0000**	0.9877	0.9841	0.9978	0.9957
D1	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
zelink6	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.7228	0.6888



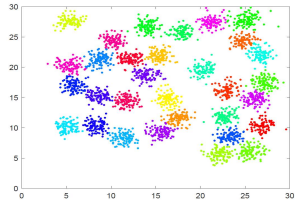
(a) ours-2circles



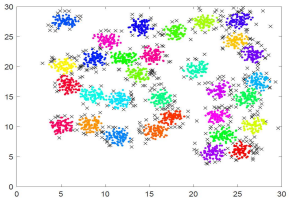
(b) DBSCAN-2circles



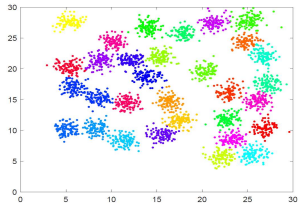
(c) RNN-DBSCAN-2circles



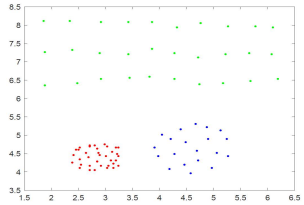
(d) ours-D31



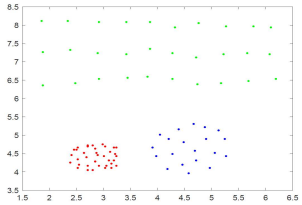
(e) DBSCAN-D31



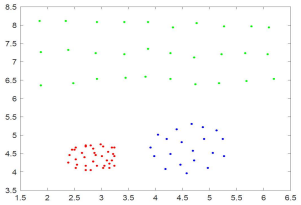
(f) RNN-DBSCAN-D31



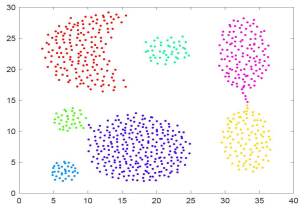
(g) ours-D1



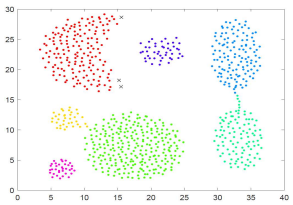
(h) DBSCAN-D1



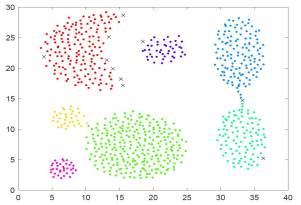
(i) RNN-DBSCAN-D1



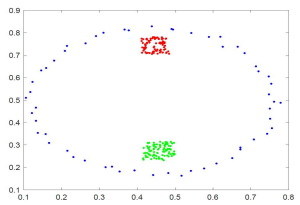
(j) ours-Aggregation



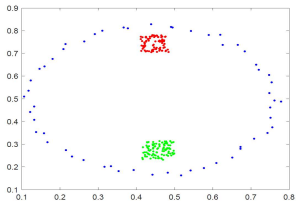
(k) DBSCAN-Aggregation



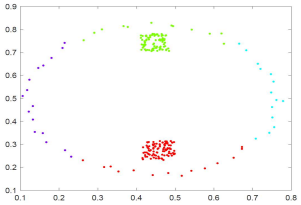
(l) RNN-DBSCAN-Aggregation



(m) ours-zelink6



(n) DBSCAN-zelink6



(o) RNN-DBSCAN-zelink6

Fig. 1: Display the results of the clustering algorithms on five artificial data sets

4.2 Results on real data sets

In this section, the proposed method was compared with the DBSCAN and RNN-DBSCAN algorithms on five real data sets, namely, Iris, Seeds, vote, Zoo, and Landsat regarding the NMI and ARI criteria.

Table 3 shows the results of this evaluation. The defined parameter values of the proposed algorithm for Iris, Seeds, vote, Zoo, and Landsat are respectively: $(K = 8)$,$(K = 6)$,$(K = 15)$,$(K = 15)$,$(K = 10)$; for DBSCAN, they are set to $(MinPts = 6, Eps = 1)$, $(MinPts = 2, Eps = 0.6)$,$(MinPts = 3, Eps = 1)$,$(MinPts = 10, Eps = 1.5)$,$(MinPts = 1,$

$Eps = 0.05$); and for RNN-DBSCAN, they are set to $(K = 6)$,$(K = 6)$,$(K = 2)$,$(K = 15)$,$(K = 10)$. Table 3 shows the higher accuracy of the proposed method than the DBSCAN algorithm in all five data sets. In other words, the findings confirmed the superiority of the proposed algorithm over DBSCAN and RNN-DBSCAN algorithms on the data sets studied in this research.

Table 3: efficiency of different clustering algorithms on the five real data sets

Data set	proposed algorithm		DBSCAN		RNN-DBSCAN	
	ARI	NMI	ARI	NMI	ARI	NMI
Iris	**0.9180**	**0.8851**	0.5681	0.7337	0.7504	0.7884
Seeds	**0.5790**	**0.6365**	0.3761	0.4786	0.5346	0.4123
vote	**0.5167**	**0.4605**	0.2999	0.3955	0.0223	0.1952
Zoo	**0.6996**	**0.7370**	0.5623	0.7331	0.5942	0.3831
Landsat	**0.4830**	**0.6090**	0.0000	0.3716	0.3421	0.5460

5 Conclusion

This paper proposed a density-based clustering method in which a criterion was defined for measuring the local density of points using the k-nearest neighbourhood of the points. Common density-based clustering algorithms have high thresholds that are selected by the user through a trial-and-error approach; for example, the DBSCAN method applies only one fixed threshold to all data sets. However, in the real world, there is neither a single *Eps* nor a single $MinPts$ applicable to all data sets. Therefore, the fewer the threshold, the better the result. In addition, in high-dimensional data sets, it is difficult for the user to determine these thresholds statically. This has resulted in some problems in DBSCAN, such as the incapability to determine correctly the clustering in high-dimensional data sets. Accordingly, the method developed in this study was set to use fewer thresholds compared to other density-based methods. One of the main parameters used in the proposed method is *Eps*, which is computed automatically. This parameter, which was formalized in this study, offers a different value for each cluster. This is the key superiority of the proposed algorithm over the other density-based algorithms. Also, we have used two indicators: NMI and ARI to analyze the results of the experiment on four artificial and four real data sets. The results showed that the proposed method is more accurate than the DBSCAN algorithm in the data sets studied in this research. In future work, the algorithm proposed in this study could be combined with hybrid clustering algorithms to improve the quality of the final clustering.

References

[1] K. Backhaus, B. Erichson, S. Gensler, R. Weiber, and T. Weiber, *Cluster analysis*, Multivariate Anal.: Application-Oriented Introduction, Springer, 2023, pp. 453–532.

[2] R. Bhuyan and S. Borah, *A survey of some density based clustering techniques*, arXiv preprint arXiv:2306.09256 (2023).

[3] A. Bryant and K. Cios, *Rnn-dbscan: A density-based clustering algorithm using reverse nearest neighbor density estimates*, IEEE Trans. Knowledge Data Engin. **30** (2017), no. 6, 1109–1121.

[4] A.A. Bushra, D. Kim, Y. Kan, and G. Yi, *Autoscan: Automatic detection of dbscan parameters and efficient clustering of data in overlapping density regions*, Peer J. Comput. Sci. **10** (2024), e1921.

[5] I. de Moura Ventorim, D. Luchi, A.L. Rodrigues, and F.M. Varejão, *Birchscan: A sampling method for applying dbscan to large datasets*, Expert Syst. Appl. **184** (2021), 115518.

[6] S. Erich, S. Jörg, E. Martin, K.H. Peter, and X. Xiaowei, *Dbscan revisited, revisited: Why and how you should (still) use dbscan*, ACM Trans. Database Syst. **42** (2017), no. 3, 1–21.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*, KDD 96 (1996), no. 34, 226–231.

[8] A. Fahim, *An extended dbscan clustering algorithm*, Int. J. Adv. Comput. Sci. Appl. **13** (2022), no. 3.

[9] Z. Falahiazar, A.R. Bagheri, and M. Reshadi, *Determining parameters of dbscan algorithm in dynamic environments automatically using dynamic multi-objective genetic algorithm*, J. AI Data Min. **10** (2022), no. 3, 321–332.

[10] X. Huang, T. Ma, C. Liu, and S. Liu, *Grit-dbscan: A spatial clustering algorithm for very large databases*, Pattern Recog. **142** (2023), 109658.

[11] L. Hubert and P. Arabie, *Comparing partitions*, J. Class. **2** (1985), no. 1, 193–218.

[12] J.-Hun Kim, J.-H. Choi, Y.-H. Park, C. Kai-Sang Leung, and A. Nasridinov, *Knn-sc: Novel spectral clustering algorithm using k-nearest neighbors*, IEEE Access **9** (2021), 152616–152627.

[13] O. Kulkarni and A. Burhanpurwala, *A survey of advancements in dbscan clustering algorithms for big data*, 3rd Int. Conf. Power Electron. IoT Appl. Renew. Energy Control (PARC), IEEE, 2024, pp. 106–111.

[14] B. Ma, C. Yang, A. Li, Y. Chi, and L. Chen, *A faster dbscan algorithm based on self-adaptive determination of parameters*, Procedia Comput. Sci. **221** (2023), 113–120.

[15] J. Qian, Y. Zhou, X. Han, and Y. Wang, *Mdbscan: A multi-density dbscan based on relative density*, Neurocomputing **576** (2024): 127329.

[16] J. Ravi and S. Kulkarni, *Automatic generation of parameters in density-based spatial clustering.*, ICTACT J. Soft Comput. **12** (2022), no. 2.

[17] M.A. Sorkhi, E. Akbari, M. Rabbani, and H. Motameni, *A dynamic density-based clustering method based on k-nearest neighbor*, Knowledge Inf. Syst. **66** (2024), 3005–3031 .

[18] A. Strehl and J. Ghosh, *Cluster ensembles-A knowledge reuse framework for combining multiple partitions*, J. Machine Learn. Res. **3** (2003), 583–617.

[19] P.M. Vaidya, *An o(n logn) algorithm for the all-nearest-neighbors problem*, Discrete Comput. Geom. **4** (1989), no. 2, 101–115.

[20] Y. Wang, J. Qian, M. Hassan, X. Zhang, T. Zhang, C. Yang, X. Zhou, and F. Jia, *Density peak clustering algorithms: A review on the decade 2014–2023*, Expert Syst. Appl. **238** (2023), 121860.

[21] Z. Wang, Z. Ye, Y. Du, Y. Mao, Y. Liu, Z. Wu, and J. Wang, *Amd-dbscan: An adaptive multi-density dbscan for datasets of extremely variable density*, IEEE 9th Int. Conf. Data Sci. Adv. Anal., IEEE, 2022, pp. 1–10.

[22] H. Yin, A. Aryani, S. Petrie, A. Nambissan, A. Astudillo, and S. Cao, *A rapid review of clustering algorithms*, arXiv preprint arXiv:2401.07389 (2024).

[23] X. Zhang and S. Zhou, *Woa-dbscan: Application of whale optimization algorithm in dbscan parameter adaption*, IEEE Access **11** (2023), 91861–91878.