Expert Systems With Applications 238 (2024) 121860

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Review

# Density peak clustering algorithms: A review on the decade 2014–2023

Yizhang Wang [a,b,*], Jiaxin Qian [a], Muhammad Hassan [c,d], Xinyu Zhang [a], Tao Zhang [a], Chao Yang [a], Xingxing Zhou [a], Fengjin Jia [a]

[a] *College of Information Engineering, Yangzhou University, Yangzhou, China*
[b] *Institute of Scientific and Technical Information of China, Beijing, China*
[c] *Institute of Biopharmaceutical and Health Engineering, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China*
[d] *Department of Radiology, Shenzhen Children Hospital, Guangdong, China*

ARTICLE INFO

*Keywords:*
Density peak clustering
Clustering algorithms
Challenges

ABSTRACT

Density peak clustering (DPC) algorithm has become a well-known clustering method during the last decade, The research communities believe that DPC is a powerful tool applied to various fields underlying distinct contemporary issues and future prospects, it is time to summarize the research progress of DPC and help them quickly know what issues have been resolved, what issues remain open, and what to do in the future. In this survey, we first describe several frequently used synthetic, UCI, and image datasets followed by the reviewing of all the DPC-related works as categorized into: finding clusters with different densities, optimizing parameter values, preventing domino effects, clustering large datasets, implementing parameter-less DPC, clustering mixed data, and clustering imbalanced data. Then, we compare the recently and widely used extensions of DPC based on the 26 synthetic and UCI datasets. Finally, according to the above analysis, the survey concludes with the improvement of DPC on synthetic and UCI datasets, revisiting large-scale data clustering, parameter-less clustering, privacy-protecting based clustering like challenges, proposing solutions on the deployment of DPC in spark, introducing deep clustering to DPC, and finally federating DPC clustering. To the best of our knowledge, this is the first review that summarizes the progress of DPC in the last decade.

## 1. Introduction

Clustering is an important technology in machine learning, which divides data into different groups given certain assumptions including grid-based, density-based, and graph-based. In 2014, Alex Rodriguez and Alessandro Laio published a novel density-based clustering algorithm, namely density peak clustering (DPC) (Rodriguez & Laio, 2014). DPC assumes that cluster centers have higher local densities and that the centers of different clusters are far from each other. DPC is based on sophisticated theoretical background with a wide range of application. In the last decade (2014–2023), DPC has been applied to defect diagnosis (Sharma, Seal, Yazidi, & Krejcar, 2022; Shi, Deng and Wang, 2020), image segmentation (Jing, Jin, & Xiang, 2021; Shi, Chen, Qi, Meng, & Cui, 2017), social communities detection (Bai, Yang, & Shi, 2017; Li, Chen, Li and Yang, 2022; Lu, Shen, Sang, Zhao and Lu, 2020; Wang, Zuo and Wang, 2016; Xu, Li, Li, Zou, & Gu, 2019), anomaly detection (Tu, Yang, Li, Zhou, & He, 2020), scheduling (Pourbahrami, Khanli, & Azimpour, 2020), image classification (Tu, Zhang, Kang, Wang, & Benediktsson, 2019), target extraction (Shang, Yang, Han, Song, & Xue, 2021), and consumption behavior analysis (Wang, Chen, Kang and Xia, 2016). DPC has also been used to improve other algorithms, such as I-nice clustering (He, Wu, Qin, Huang, & Jin, 2021), active learning (Shi, Yu et al., 2020; Wang, Min, Zhang and Wu, 2017), spectral clustering (Cheng, Huang, Zhang, Zhang, & Luo, 2021), *etc.* DPC can be applied to a wide range of applications with promising results and improvements and the core idea of density peak is also a powerful tool that brings a lot of inspiration to clustering algorithm communities.

In DPC, cluster centers (*i.e.,* density peaks) can be manually selected from a 2D decision graph, which consists of two main properties: local density ($\rho$) as abscissa and relative distance ($\delta$) as ordinate. On the decision graph, the data points with higher values of local density and relative distance are selected as cluster centers following the density peak assumption. Undoubtedly, the decision graph is a novel way of selecting cluster centers and locating outliers.

Besides, its data points assignment strategy is based on single-chain label propagation, a data point is assigned to another data point with a higher local density. Intuitively, the final clusters generated by DPC are like mountains, and cluster centers are like mountain peaks, which is why the authors refer to cluster centers as density peaks. Obviously, this

[*] Corresponding author at: College of Information Engineering, Yangzhou University, Yangzhou, China.
*E-mail address:* wyizhang@yzu.edu.cn (Y. Wang).

https://doi.org/10.1016/j.eswa.2023.121860
Received 19 April 2023; Received in revised form 20 September 2023; Accepted 25 September 2023
Available online 29 September 2023

mechanism has a huge advantage, the outliers and boundary points are naturally merged into the nearest data points with higher local density.

The two aforementioned characteristics attract much interest to DPC. There are over 110 articles on DPC published in major journals and conferences. There are still a lot of academics studying this algorithm at present. This survey provides the basics of DPC, contemporary works, and future perspectives, addressing challenges in DPC to solve critical problems. We aim to help researchers who pursue this topic further find out which problems have been addressed and which are still open in DPC, and what needs to be done going forward.

The existing surveys cover a limited number of literature work: Li et al. reviewed 30 papers (Li, Sun, Tang and You, 2022), while Wei et al. reviewed about 70 papers (Wei, Peng, Huang, & Zhou, 2023). On the other hand, we review over 110 DPC-related papers, and the majority of these articles are published in credible journals and conferences. In particular, this study includes a large number of experiments to deeply analyze the research progress of the DPC algorithm. Thus, we believe that this review is an excellent resource to lay the foundation of DPC concept for beginners and provide innovative ideas for advanced learners. Reading this review will enable the researchers to access the necessary data and related codes, and allow them to concentrate on the most crucial future work of DPC.

In this survey, we provide a thorough analysis of the DPC algorithm and associated techniques. The review is divided into several parts according to the research fields of DPC including finding clusters with various densities, enhancing parameter values, clustering enormous datasets, parameter-free DPC, clustering mixed data, and clustering unbalanced data. Each section includes a thorough analysis of the DPC-related methodologies. According to the above analysis, our contributions are as follows:

(i) improvements in DPC (VDPC, SNNDPC, 3W-DPET, etc.) have been achieved with promising results on synthetic and UCI datasets. Continuing to improve DPC for higher accuracy on synthetic and UCI datasets is difficult.

(ii) Large-scale data clustering, parameter-less clustering, privacy-protecting based clustering, *etc* are challenging directions, focusing on the above topic can better promote the development of DPC.

(iii) We propose possible solutions for the deployment of DPC in Spark, introducing deep clustering to DPC, federated DPC clustering, *etc*.

The rest of the paper is organized as follows. Section 3 introduces DP process, Section 4 describes datasets, such as synthetic, UCI, and image datasets, Section 5 examines the DPC-related techniques, Section 6 presents experiments conducted on 15 DPC-related algorithms and their findings, and finally Section 7 concludes the manuscript with future works.

## 2. Research methodology

### 2.1. Data search strategy

Google Scholar was used as a search engine with DPC as term. The period of these papers in this survey is from 2014 to 2023 (actually before April 2023). The used keywords in Google Scholar are: *density peak clustering*, *density peak* or *DPC* indexed by Web of Science and Google Scholar, anyone can download them from the corresponding websites.

### 2.2. Inclusion and exclusion criteria

The number of citations of DPC is too much, we set some inclusion and exclusion criteria to select papers. The inclusion criteria are as follows: (1) peer-review published papers that are written in English. (2) focus on density peak clustering algorithms. (3) we only select papers from important journals and conferences standardized by the China Computer Federation (CCF)[1] (we name it CCF list later). All the collected papers are listed in CCF. For example, Expert Systems With Applications is a CCF-recommended journal, the comment of the China Computer Federation is "an internationally important journal with important international academic influence, and Chinese scholars are encouraged to submit papers". The exclusion criteria are as follows: (1) preprints papers. (2) the medium of writing is not English.

According to the above inclusion and exclusion criteria, all the co-authors are directed to select over 110 selected papers in DPC research. By selecting these papers, we may efficiently summarize and examine the research situation of DPC.

### 2.3. Data analysis method

We use both qualitative and quantitative methods to study the related work of DPC. From the qualitative perspective, we divided all references into several categories based on the shortcomings of DPC and evaluate the research status of each category separately. By the above method, we can clearly know what problems have been solved or not solved. From a quantitative perspective, we compare the clustering results of highly cited papers with access to their code for evaluation. If the scholars develop a new extent of DPC, they can know the performance level of the proposed algorithm.

## 3. Density Peak Clustering (DPC) algorithm

DPC uses two parameters to construct a decision graph, namely $\rho$ and $\delta$. The parameter $\rho$ denotes local density, and the parameter $\delta$ represents the minimum distance between a given data point and another point with higher density (Rodriguez & Laio, 2014). For a given dataset $D = (x_1, x_2, \dots, x_N)^T$ containing $N$ data points, the local density of a specific $x_i$ can be defined as follows:

$$\rho_i = \sum_j \chi(d_{ij} - d_c), \tag{1}$$

where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, $d_{ij} = \left\|x_i - x_j\right\|$, and $d_c$ is cutoff distance. The value of $d_c$ is often selected from the Euclidean distance matrix of all data points according to a percentage in percent. Another common-used method of calculating $\rho_i$ is as follows:

$$\rho_i = \sum_j e^{-(\frac{d_{ij}}{d_c})^2}, \tag{2}$$

Subsequently, parameter $\delta$ is defined as

$$\delta_i = \min_{j:\rho_j > \rho_i} d_{ij}. \tag{3}$$

By using $\rho$ and $\delta$, DPC constructs a decision graph as shown in Fig. 1. A rectangle can be selected underlying some data points as cluster centers, the remaining data points are assigned to the nearest data point with higher local density. Finally, the corresponding clustering results are obtained (see Fig. 1).

In the process of DPC, choosing the optimal value for $d_c$ is inevitable as cluster centers. Usually, the data points with higher values of $\rho$ and $\delta$ are suitable to be selected as centers. The advantages of DPC are as follows:

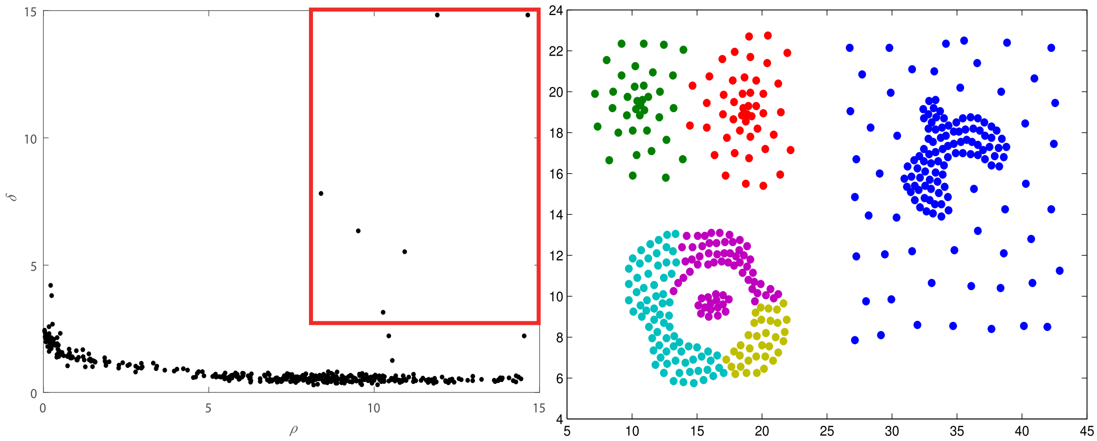
(i) DPC does not need iteration.

(ii) DPC can identify clusters with a single peak in datasets including Flame, Spiral, Jain, and Aggregation. However, it is difficult for K-means and DBSCAN to deal with these four datasets.

(iii) DPC is good at identifying outliers and boundary points.

Based on these advantages, DPC receives widespread attention. However, DPC also has some limitations as follows:

[1] https://www.ccf.org.cn.



(a) decision graph (b) clustering results generated by DPC

**Fig. 1.** We apply the DPC algorithm to the dataset Compound. (a) decision graph with a red rectangle highlighting a group of data points that are selected as cluster centers. (b) The clustering results were obtained by DPC using these data points in the red rectangle as cluster centers.

(i) DPC is not good at identifying clusters with different densities (see Fig. 1).

(ii) The values of $\rho$ and $\delta$ are global, which affects the cluster centers' determination and data point assignments.

(iii) For data point assignments in DPC, it exists domino effects, *i.e.,* once a data point is wrongly assigned, another data point will follow similar fashion of incorrect assignment.

(iv) It is difficult for DPC to deal with large datasets, and the time complexity of DPC is $O(n^2)$.

(v) DPC has one parameter and the selection of cluster centers needs human intervention. Clustering is an unsupervised learning algorithm, parameters adjustments violate the essence of unsupervised learning. The parameter-less DPC is needed.

(vi) the performances of DPC on mixed data are not good.

(vii) the performances of DPC on imbalanced data are not good.

Most DPC-related papers focus on the aforementioned problems, where the attempted solutions have been outlined in this study. In general, more than ten papers focus on the same problem based on our review. (i), (ii), and (iii) are only DPC's limitations, while (iv), (v), (vi), and (vii) are not only DPC's limitations but also common problems of all clustering algorithms. To some extent, (i), (ii), and (iii) have been solved in the past ten years. (iv), (v), (vi), and (vii) are still open problem. In the next section, we will discuss this perspective in more depth in Section 5.

## 4. Datasets

The design of clustering algorithms is often associated with data distribution. The authors of the DPC-related studies frequently employ synthetic, UCI, and even image datasets. The following are the popular datasets.

### 4.1. Synthetic datasets

Synthetic datasets typically have two or three dimensions, which makes them easy to visualize. In Table 1, we list 48 commonly used synthetic datasets, and their shapes and ground-truths can be observed in Fig. 2. These datasets can be downloaded from the provided link.[2] Different synthetic datasets possess different properties, which are summarized as follows:

(i) *Uniform density datasets*: different natural clusters have close densities in a dataset. Density, in cluster analysis, refers to the number of data points in a local space. Uniform density is observed in datasets such as 2piral, Pearl, Spiral, T8-8k, T5-8k, T7-10k, T4-8k, Diamond9, Zelink1, Cassini, Banana, Dountcurves, Dount3, and Smile2. Among density-based clustering algorithms, DBSCAN is good at identifying clusters with uniform density.

(ii) *Multiple/varied density datasets*: different natural clusters have varied densities in a dataset. Example datasets with multiple densities include Pathbased, Compound, Jain, D1, and Zenlink6. It can be a challenging task for most density-based clustering algorithms to identify clusters with multiple densities.

(iii) *Convex datasets*: all links between data points of one cluster are contained within the area of the cluster. Some examples of convex datasets include 2d-4c-np9, 2d-3c-no123, 2d-4c-no4, 2d-10c, 2d-20c-no0, 2dnormals, twenty, D2, Aml28, R15, Diamond9, Blobs, Aggregation, S1, DS5, and D31.

(iv) *Non-convex datasets*: all links between any two data points of one cluster are partially included within the area of the cluster. Examples of non-convex datasets include 2circle2_noise, 2sp2glob, 2spiral, Pathbased, Pearl, Jain, D13, Spiral, T8-8k, T5-8k, T7-10k, T4-8k, Clusterincluster, Compound, Zelink1, Atom, Crossline, Circle, Banana, Chainlink, Donutcurves, Dount3, Smile2, Spiralsquare, Rings, 3MC, and Zelink6.

(v) *Single peak datasets*: by applying DPC, any one natural cluster has one density peak as a center. Datasets such as 2d-4c-np9, 2d-3c-no123, 2d-4c-no4, 2d-10c, 2d-20c-no0, 2dnormals, twenty, D2, Aml28, Banana, R15, Diamond9, Blobs, S1, DS5, and D31 are single peak datasets, where DPC performs best at identifying such clusters.

(vi) *Multiple peaks datasets*: by applying DPC, any one natural cluster has multiple centers with a higher local density as centers. Examples of multiple peaks datasets include 2circle2_noise, Pathbased, Pearl, D13, T8-8k, T7-10k, T4-8k, Clusterincluster, Compound, Zelink1, Atom, Circle, Banana, Chainlink, Donutcurves, Dount3, Smile2, Rings, 3MC, and Zelink6, which pose a challenge for DPC in identification. The identifications of single peak and multiple peaks are obviously from DPC.

(vii) *Cross-manifold datasets*: there is an intersection between two or more clusters for a dataset, which is called a cross-manifold dataset. Examples of such datasets include Five_affine_subspaces, Clusterincluster, Atom, Crossline, and Chainlink, and it is challenging for most classic clustering algorithms to identify cross-manifold clusters.

Clustering algorithms are often evaluated using synthetic datasets that demonstrate their ability to identify different types of clusters. This approach provides readers with a clear understanding of the proposed algorithm, which is particularly important in the field of clustering. However, most studies only use a selected subset of the available

[2] https://github.com/mlyizhang/Clustering-Datasets.

**Table 1**
The features of 48 common-used synthetic datasets, $N$ denotes the number of samples, $D$ denotes the number of dimensions, and $NC$ denotes the number of natural clusters based on ground-truths.

| Type | ID | Datasets | $N/D/NC$ | ID | Datasets | $N/D/NC$ |
|---|---|---|---|---|---|---|
| Synthetic | 1 | 2circles_noise | 634/2/3 | 2 | 2d-4c-no9 | 312/2/3 |
| Synthetic | 3 | 2d-3c-no123 | 715/2/3 | 4 | 2d-4c-no4 | 863/2/4 |
| Synthetic | 5 | 2d-10c | 2990/2/9 | 6 | 2d-20c-no0 | 1517/2/20 |
| Synthetic | 7 | 2dnormals | 1000/2/2 | 8 | 2sp2glob | 2000/2/4 |
| Synthetic | 9 | 2spiral | 1000/2/2 | 10 | Pathbased | 300/2/3 |
| Synthetic | 11 | Pearl | 1000/2/2 | 12 | Jain | 373/2/2 |
| Synthetic | 13 | Twenty | 1000/2/20 | 14 | Five_affine_subspaces | 700/2/5 |
| Synthetic | 15 | D13 | 588/2/13 | 16 | D2 | 85/2/4 |
| Synthetic | 17 | Spiral | 312/2/3 | 18 | Aml28 | 804/2/5 |
| Synthetic | 19 | R15 | 600/2/15 | 20 | T8-8k | 8000/2/9 |
| Synthetic | 21 | T5-8k | 8000/2/7 | 22 | T7-10k | 10 000/2/10 |
| Synthetic | 23 | T4-8k | 8000/2/7 | 24 | Diamond9 | 3000/2/9 |
| Synthetic | 25 | Clusterincluster | 1012/2/2 | 26 | Compound | 399/2/6 |
| Synthetic | 27 | T2-4k | 4200/2/7 | 28 | Blobs | 300/2/3 |
| Synthetic | 29 | Flame | 240/2/2 | 30 | Zelink1 | 299/2/3 |
| Synthetic | 31 | Aggregation | 788/2/7 | 32 | D1 | 87/2/3 |
| Synthetic | 33 | Atom | 800/3/2 | 34 | Cassini | 1000/2/3 |
| Synthetic | 35 | Crossline | 1000/2/2 | 36 | Circle | 1000/2/2 |
| Synthetic | 37 | Banana | 4811/2/2 | 38 | Chainlink | 1000/3/2 |
| Synthetic | 39 | Donutcurves | 1000/2/4 | 40 | Donut3 | 999/2/3 |
| Synthetic | 41 | S1 | 5000/2/15 | 42 | Smile2 | 1000/2/4 |
| Synthetic | 43 | DS5 | 500/2/5 | 44 | Spiralsquare | 1500/2/6 |
| Synthetic | 45 | Rings | 1000/2/3 | 46 | D31 | 3100/2/31 |
| Synthetic | 47 | 3MC | 400/2/3 | 48 | Zelink6 | 238/2/3 |

**Table 2**
The features of common-used UCI datasets, $N$ denotes the number of samples, $D$ denotes the number of dimensions, and $NC$ denotes the number of natural clusters based on ground-truths.

| Type | ID | Datasets | $N/D/NC$ | ID | Datasets | $N/D/NC$ |
|---|---|---|---|---|---|---|
| UCI | 1 | Abalone | 4177/8/3 | 2 | Anuran-Calls | 7195/19/60 |
| UCI | 3 | Balancescale | 625/4/3 | 4 | Breast | 277/9/3 |
| UCI | 5 | Ecoli | 336/7/8 | 6 | German | 1000/24/2 |
| UCI | 7 | Gesture | 1747/18/2 | 8 | Glass | 214/9/6 |
| UCI | 9 | Heart | 303/13/2 | 10 | Immunotherapy | 90/7/12 |
| UCI | 11 | Ionosphere | 351/34/2 | 12 | Iris | 150/4/3 |
| UCI | 13 | Landsat | 2000/36/6 | 14 | Leaf | 340/15/30 |
| UCI | 15 | Liver | 345/6/2 | 16 | Movement_libras | 360/90/15 |
| UCI | 17 | Msplice | 3175/240/3 | 18 | Musk | 6598/166/2 |
| UCI | 19 | Pima | 768/8/2 | 20 | Seeds | 210/7/3 |
| UCI | 21 | Image Segmentation | 210/19/7 | 22 | Semeion | 1593/256/9 |
| UCI | 23 | Sonar | 208/60/2 | 24 | Spambase | 4601/57/2 |
| UCI | 25 | Vehicle | 846/18/4 | 26 | Vote | 435/16/2 |
| UCI | 27 | Wine | 178/13/3 | 28 | wpbc | 198/33/2 |
| UCI | 29 | Yeast | 1484/8/8 | 30 | Zoo | 101/16/7 |

48 synthetic datasets for experimentation. To date, no research has utilized all 48 datasets concurrently, it presents a significant challenge for clustering algorithms attempting to identify all shown datasets simultaneously.

### 4.2. UCI datasets

UCI datasets are often used to evaluate the performance of clustering methods. We introduce the well-known real-world UCI[3] datasets in this part for clustering techniques. Table 2 lists the 30 most popular UCI datasets.

The dimensions of all the UCI datasets are greater than three, which is impossible to visualize. Therefore, their internal structure cannot be observed. Clustering high-dimensional datasets is a challenging task owing to factors of capturing the intrinsic structure of high-dimensional datasets, despite the existence of several dimensionality reduction technologies. Similarly, the published paper only selects a part of 30 UCI datasets to validate the proposed clustering algorithm.

### 4.3. Image datasets

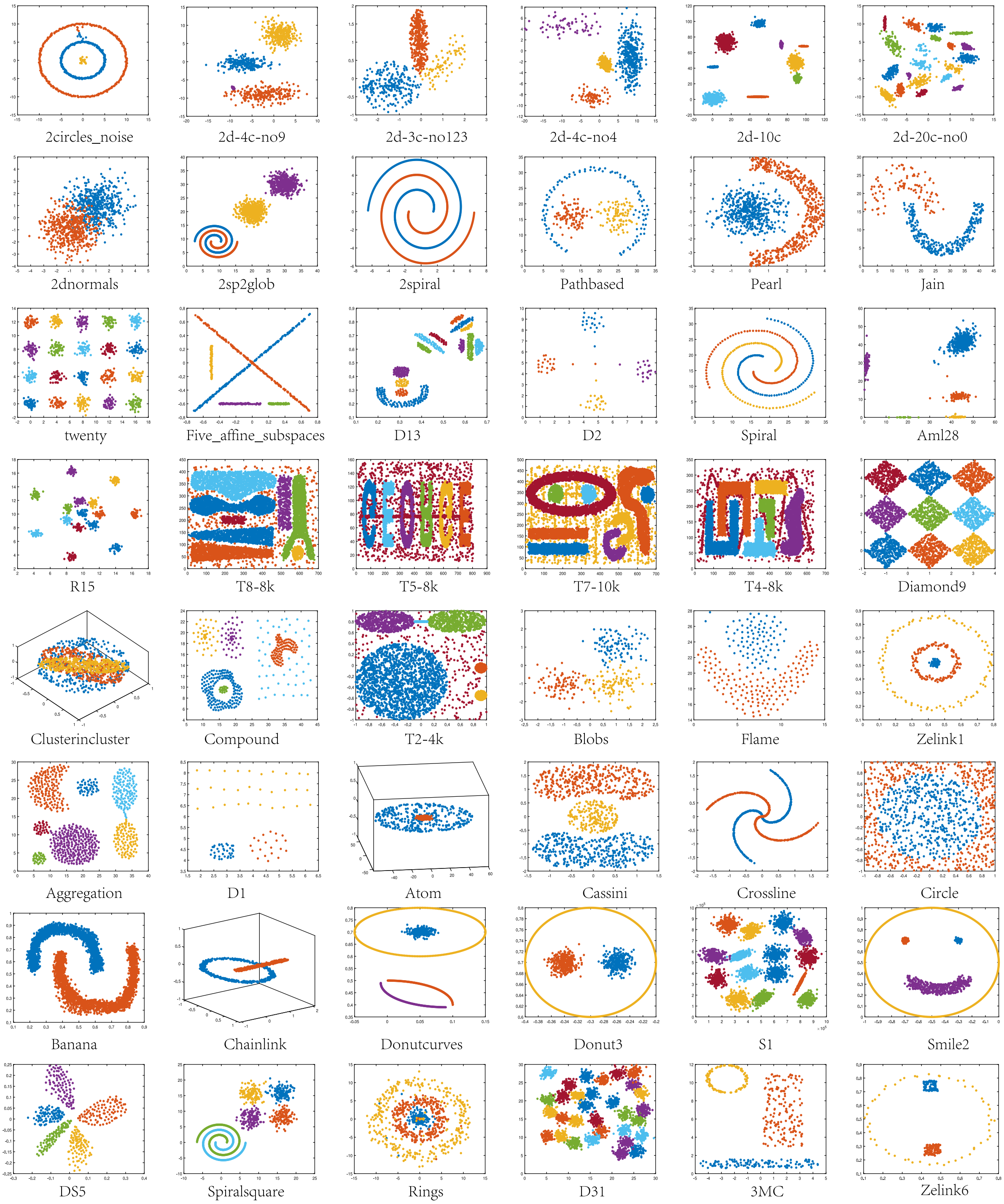
Images datasets are also important to verify the performance of clustering algorithms. The common-used image datasets are as follows:

(1) *STL-10* (Coates, Ng, & Lee, 2011): STL-10 images are from ImageNet, it has a total of 113 000 RGB images with 96 × 96 pixels including 5000 training images, 8000 testing images, and the rest 100 000 unlabeled images.

(2) *CIFAR-10* (Krizhevsky, Hinton, et al., 2009): The CIFAR-10 dataset is a small collection of RGB color images used for object recognition tasks. The dataset consists of 10 categories, which include aircraft, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each color image in CIFAR-10 has a size of 32 × 32 pixels. The dataset comprises 50,000 training images and 10,000 testing images, which makes it a valuable resource for evaluating the performance of machine learning models.

(3) *CIFAR-100* (Krizhevsky et al., 2009): The CIFAR-100 dataset is similar to CIFAR-10, but instead consists of 100 object categories, with each category containing 600 images. For each category, there are 500 training images and 100 test images. The categories are further divided into 20 superclasses, each image has a "fine" label indicating its specific category and a "coarse" label indicating the superclass to which it belongs. This dataset provides a challenging task for machine learning models due to its increased number of classes and subclasses.

[3] https://archive.ics.uci.edu/ml/index.php.



2circles_noise 2d-4c-no9 2d-3c-no123 2d-4c-no4 2d-10c 2d-20c-no0

2dnormals 2sp2glob 2spiral Pathbased Pearl Jain

twenty Five_affine_subspaces D13 D2 Spiral Aml28

R15 T8-8k T5-8k T7-10k T4-8k Diamond9

Clusterincluster Compound T2-4k Blobs Flame Zelink1

Aggregation D1 Atom Cassini Crossline Circle

Banana Chainlink Donutcurves Donut3 S1 Smile2

DS5 Spiralsquare Rings D31 3MC Zelink6

**Fig. 2.** The shapes and ground-truths of 48 frequently-used synthetic datasets, and note that different colors represent different ground-truth labels.

(4) *MNIST* (LeCun, Bottou, Bengio, & Haffner, 1998): The MNIST dataset is widely used in image recognition and machine learning, it consists of 70,000 grayscale images of handwritten digits. It is commonly used as a benchmark dataset for evaluating image processing techniques and consists of 60,000 training images and 10,000 test images. The images are of size 28 × 28 and are pre-processed to be centered and normalized. The simplicity of the dataset makes it an ideal starting point for those new to machine learning and image recognition techniques, as it requires minimal preprocessing or formatting.

(5) *Fashion-MNIST* (Xiao, Rasul, & Vollgraf, 2017): Fashion-MNIST is a dataset of 70,000 images of fashion products, which includes 60,000 training images and 10,000 testing images. Each image in the dataset is a 28 × 28 grayscale image, and it is associated with one of ten different classes that include t-shirts, dresses, sandals, sneakers, and more. The dataset is often used as a benchmark for evaluating image recognition and machine learning algorithms, and it provides a challenging yet realistic problem for researchers and developers to work on.

(6) *Caltech101* (Fei-Fei, Fergus, & Perona, 2004): The Caltech101 dataset is a collection of images that features 101 different object categories such as "helicopter", "elephant", and "chair", in addition to a background category containing images that do not belong to these categories. Each object category in the dataset has approximately 40 to

800 images, with most categories having around 50 images. The images have a resolution of approximately 300 × 200 pixels.

(7) *Olivetti* (Gao et al., 2021): The Olivetti dataset consists of a collection of 400 face images from 40 different individuals, which has 10 different face images. It is often used as a benchmark dataset for machine learning algorithms and can help evaluate the performance of different techniques. The images are grayscale and have a resolution of 64 × 64 pixels.

(8) *COIL20*: The Columbia University Image Library (COIL-20) database is a collection of images of 20 different objects. Each object has 72 images, which are taken from different angles in 5-degree increments along a horizontal rotation of 360 degrees. Each image has a resolution of 64 × 64 pixels.

(9) *Imagenet32* (Fei-Fei et al., 2004): Imagenet32 is a dataset that contains 1,281,167 images for training and 50,000 images for testing. These images are classified into 1000 different labeled categories, which can be used for tasks such as image classification, object detection, and semantic segmentation. The images in Imagenet32 have a size of 32 × 32 pixels and are a smaller version of the original Imagenet dataset.

(10) *ORL* (Samaria & Harter, 1994): The ORL database is a collection of images that includes 40 different individuals. Each individual has 10 different images, resulting in a total of 400 images in the dataset.

(11) *UMIST* (Wechsler, Phillips, Bruce, Soulie, & Huang, 2012): The UMIST face image database is a dataset that includes 20 subjects. Each subject has a wide range of multiview face images

In many research papers about DPC, the authors often rely on synthetic and UCI datasets. While these datasets may be small and easy to work with, they do not necessarily reflect all the complexities of real-world scenarios. For instance, the well-known Iris dataset from UCI contains only shape information (sepal length, sepal width, petal length, and petal width) for three types of iris plants, the visual features like color and texture are missing. Relying solely on UCI datasets to evaluate the performance of clustering algorithms is therefore insufficient.

Recent developments in classification algorithms and generative models have shown that using realistic data is essential for achieving real-world applications. Take ChatGPT, for example, which has revolutionized the potential of artificial intelligence for everyday use. Thus, to further advance DPC and other clustering algorithms, we must move beyond synthetic and UCI datasets and work with more realistic scenarios. By doing so, we can promote the development and application of clustering algorithms in various domains.

## 5. DPC-related work

We review over 110 DPC-related approaches in this survey. As of May 2023, among them, the citations of 36 papers are greater than 35 in Google Scholar (see Fig. 3), *i.e.,* the H-index of DPC is 35. Specifically, the citations of DPC-KNN-PCA (Du, Ding, & Jia, 2016), FKNNDPC (Xie, Gao, Xie, Liu, & Grant, 2016), SNNDPC (Liu, Wang, & Yu, 2018), CFSFDP-HD (Mehmood, Zhang, Bie, Dawood, & Ahmad, 2016), ADPC-KNN (Yaohui, Zhengming, & Fang, 2017), DenPEHC (Xu, Wang, & Deng, 2016), ALEC (Wang, Min et al., 2017), and Fast-DPeak (Chen et al., 2020) are more than 100, they are important baselines for researches of DPC, most of them are published before the year of 2018. Among the highly cited DPC-related papers, we only find six open-source codes, the number is very small.

In this section, we divide all DPC-related literature work based on the essence of the problems: (1) identifying clusters with different densities, (2) parameters values improvement, (3) avoiding domino effects, (4) clustering large datasets, (5) parameter-less DPC, (6) clustering mixed data, (7) clustering imbalanced data, and (8) applications. Then, we review the related work based on the above eight categories.

### 5.1. Identifying clusters with different densities

A typical issue in density clustering is the identification of clusters having various densities. Many clustering techniques have been presented to handle multiple density classical datasets such as Pathbased, Compound, Jain, D1, and Zenlink6 (see Fig. 2). Several methods utilize neighbor concepts like k-nearest neighbor (KNN), natural nearest neighbor (NNN), shared nearest neighbor (SNN), mutual k-nearest neighbor (MKNN), and reverse nearest neighbor (RNN). For example, Fan et al. incorporated the mutual k-nearest-neighbor graph into DPC, and the clustering results on datasets Pathbased, Atom, and Zelink6 are 1.0 (Fan, Jia, & Ge, 2020). Ding et al. used natural neighbors to recalculate $\rho$ and $\delta$ of DPC, generated sub-clusters, and merged all the sub-clusters by new similarities (Ding, Du et al., 2023). Sun et al. merged microclusters based on kNN and self-recommendation (Sun, Qin, Ding, Xu, & Zhang, 2021), while Wu et al. constructed a sparse graph by using KNN, identified pairs of representatives of initial clusters, and merged them based on nearest neighbors graph (Wu, Peng, Lee, Leibnitz, & Xia, 2021). Abbas et al. used MKNN to improve DPC to identify clusters with different densities (Abbas, El-Zoghabi, & Shoukry, 2021).

Some methods consider metric learning. For example, Rasool et al. proposed an effective method for DPC following data-dependent similarity measures based on probability mass and applied it to DPC (Rasool, Aryal, Bouadjenek, & Dazeley, 2023). Zhang et al. proposed a novel DPC algorithm based on balance density and connectivity (Zhang, Dai, & Wang, 2023). Pizzagalli et al. proposed a trainable DPC according to the properties of paths between points rather than point-to-point similarity (Pizzagalli, Gonzalez, & Krause, 2019). Du et al. introduced geodesic distances into DPC (Du, Ding, Xu and Xue, 2018), while Wang et al. proposed a relative density-based DPC (Wang & Yang, 2021). Tao et al. proposed to use global and local consistency adjustable manifold distance to improve DPC (Tao, Guo et al., 2021). Hou et al. introduced relative density relationship into DPC (Hou, Zhang, & Qi, 2020), while Yan et al. utilized statistical outlier detection method and probability density functions to deal with datasets of various distributions and dimensionalities (Yan, Wang, & Lu, 2019). Guan et al. realized some clusters have multi-peaks, so they proposed a graph structure for non-peak and density peak allocation (Guan, Li, He, & Chen, 2023).

Systematic ways are often used to identify clusters with varied densities: Wang et al. proposed a new method based on density levels and proposed three kinds of DPC (Wang, Wang, Pang et al., 2020; Wang, Wang, Zhang et al., 2020; Wang, Wang, Zhou, Zhang, & Quek, 2023). Zhu et al. improved hierarchical clustering by using the joint version of DPC and DBSCAN (Zhu, Ting, Jin, & Angelova, 2022). Li et al. proposed local gap density to identify clusters with different densities (Li, Yang, Qin, & Zhu, 2019).

Besides, Xu et al. presented a fat node leading tree for data stream clustering with density peaks (Xu, Wang, Li, Deng, & Gou, 2017). Bian et al. proposed fuzzy DPC to provide flexible adaptability for tackling ambiguity and uncertainty in clustering (Bian, Chung, & Wang, 2020). Fang et al. proposed a new adaptive core fusion-based DPC (Fang, Qiu, & Yuan, 2020). Parmar et al. improved DPC to handle datasets comprising various data distribution patterns based on residual error (Parmar et al., 2019).

Many studies claim that their proposed algorithms significantly identify clusters with different densities very well, but they are lacking to verify their proposed methods via multiple densities based datasets such as Pathbased, Compound, Jain, D1, and Zenlink6, which have multiple densities. If the research goal of multiple density clustering is to identify the above synthetic datasets with the accuracy of 100%, APC (Wang, Wang, Pang et al., 2020)and VDPC (Wang et al., 2023) can achieve this goal. All in all, there is no need to continue to address the problem of multiple-density cluster identification for synthetic datasets. Noteably, the identification of synthetic datasets containing multiple densities is less challenging compared to the identification of real-world

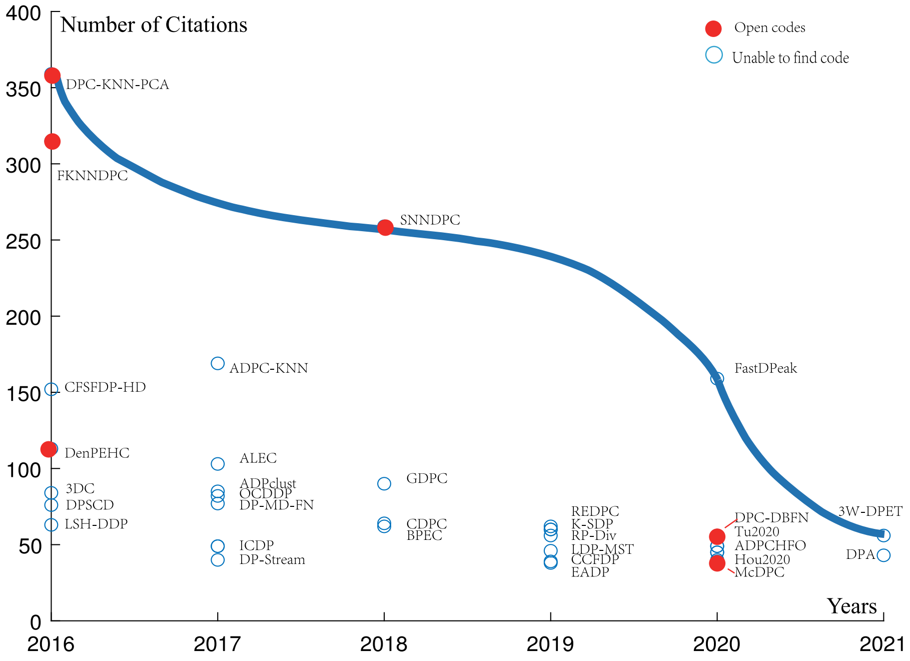


**Fig. 3.** The highly cited DPC-related papers (the number of citations is greater than 35) from 2014 to 2023.

datasets. However, there is currently no theoretical evidence to support the existence of various densities in real-world datasets. As a result, it is challenging to determine if a clustering algorithm can identify clusters with different densities using real-world datasets. However, the aforementioned multiple density algorithms perform better on real-world datasets with supporting evidence of multiple-density assumption via experimental findings. The latecomers are expected to concentrate on clustering data in realistic scenarios for this topic.

### 5.2. Improving the values of $\rho$ and $\delta$

The computing way of $\rho$ and $\delta$ is global, this feature makes it difficult for DPC to cope with different distributed data.

Density metric is a common-used strategy. Du et al. recalculated the values of $\rho$ and $\delta$ based on the sensitivities of local density and density-adaptive metric (Du, Ding, Xue, & Shi, 2019). Liang et al. used the concept of density-reachable in the DBSCAN and divide-and-conquer strategy to autonomously obtain the cluster centers in DPC (Liang & Chen, 2016). Zhang et al. improved DPC in density measurement and cluster center identification (Zhang, Miao, Tian, & Wang, 2022). Tong et al. designed a density metric by using a continuous function (Tong, Liu and Gao, 2021). Hou et al. used density normalization to improve DPC (Hou & Zhang, 2019). Guo et al. designed a new measure and proposed that a cluster center has a relatively high local density whether in dense clusters or sparse clusters (Guo, Huang, Cai, & Zhu, 2018). Other distance learning methods are also used: Qin et al. proposed the Jaccard coefficient to measure the similarity between points (Qin et al., 2021). Yang et al. utilized divergence distance and adjusted boxplot to transform the original $\rho$ and $\delta$ into new divergence distance space, and then find reasonable centers in the new distance space (Yang, Cai, Yang, & Zhao, 2022).

The fuzzy theory is an effective method to improve local density. Li et al. introduced a fuzzy semantic cell model to computing local density to improve the performance of DPC (Li, Sun and Tang, 2022). Su et al. introduced belief peaks into DPC and improved the values of $\rho$ and $\delta$ to significant level (Su & Denoeux, 2018). Du et al. used fuzzy neighborhood relation to redefine the local density (Du, Ding and Xue, 2018). The concept of nearest neighbors is also used to improve $\rho$ and $\delta$: Sun et al. proposed a novel nearest neighbors-based adaptive DPC algorithm with an optimized allocation strategy (Sun, Qin, Ding, & Xu, 2022). Liu et al. proposed an SNN-based DPC, which enhances the local density of data points with low density and makes the center selection reasonable. This is an important extent of DPC (Liu et al., 2018).

In addition, Cheng et al. identified local cores of dense members and used DPC to generate final clusters (Cheng, Zhang, & Huang, 2020). Ni et al. divided points into multiple potential clusters and then merged clusters whose density peaks are not prominent to obtain accurate clustering results (Ni, Luo, Zhu, & Liu, 2019). Li et al. taken the concept of comparison into DPC (Li & Tang, 2018).

SNNDPC is an important extension of DPC, it introduces SNN similarity to improve $\rho$ and $\delta$ and achieves very impressive performance on UCI datasets (Liu et al., 2018). SNN similarity makes the $\delta$ of data points in the low-density area higher, and cluster center selection becomes effective. SNNDPC is an important baseline in DPC-related research. New improvements of DPC in the future cannot ignore this algorithm, if some methods after SNNDPC do not compare them with SNNDPC, it is not fair. The belief peaks-based DPC is another important work of introducing fuzzy set theory into DPC (Su & Denoeux, 2018), and new extends of fuzzy DPC in the future are supposed to be compared with it.

### 5.3. Avoiding domino effect

The domino effects are caused by the assignment's mechanism of the DPC itself: once a data point is incorrectly assigned, the other data points follow the same manner. There are many ideas for improving methods, which are heuristic algorithms.

Qin et al. developed a two-step allocation strategy based on label propagation and Jaccard coefficients (Qin et al., 2021). Yang et al. designed an order similarity, *i.e.*, order rank of Euclidean distance between two samples, and proposed a two-step assignment method including graph-based assignment and kNN (Yang, Cai, Li and Zhu, 2021). Zhou et al. proposed a KNN-based local density estimation, core points identification and integration, and the classification of the non-core objects to avoid domino effects (Zhou, Si, Sun, Qu, & Hou, 2022). Zhang et al. proposed a new concept of density decay graph and used it to avoid domino effects (Zhang, Zhu et al., 2021). Xu et al. used turning angle and the graph connectivity to automatically select cluster centers (Xu & Jiang, 2022). Guan et al. proposed an association degree transfer method and used it to improve DPC (Guan, Li, He, Zhu, & Chen, 2021). Seyedi et al. used a graph-based label propagation to assign labels to remaining points and form final clusters (Seyedi, Lotfi, Moradi, & Qader, 2019). Guo et al. proposed a new DPC with connectivity estimation (Guo et al., 2022). Lotf et al. used a fuzzy kernel to exclude the noises and constructed a density-based kNN graph for clusters backbones (Lotfi, Moradi, & Beigy, 2020). Du et al. introduced KNN into DPC (Du et al., 2016). Fuzzy set theory (Mahmood & Ali, 2022) is often applied to clustering algorithms, Yu et al. used three-way decision and D–S evidence theory to optimize the assignments process (Yu, Chen, & Yao, 2021). Yu et al. proposed an improved DPC by introducing weighted local density sequence and two-stage assignment strategies (Yu, Liu, Guo, Liu, & Yao, 2019).

The above research fields in Sections 5.1, 5.2, and 5.3 are only DPC's problems and their relationships are very close, and even influence each other. Sometimes, an algorithm handles multiple problems (among the above three problems) simultaneously to achieve better performances.

### 5.4. Clustering large datasets

Large-scale datasets clustering is a very hot topic in the field of clustering algorithms. In big data environments, almost every clustering algorithm has this problem.

One way is to avoid quadratic computation overhead ($O(n^2)$) of DPC: Wu et al. used a grid to design an improved DPC with $O(np\#grids)$ ($\#grids$ denotes the number of grids) (Fang, Xu, Ji, Wang, & Huang, 2022). Lu et al. proposed distributed DPC with $O(NlogN)$ (Lu, Zhao, Tan and Wang, 2020). Cheng et al. used a minimum spanning tree to improve DPC called LDP-MST, which performs better on large datasets (Cheng, Zhu, Huang, Wu, & Yang, 2019). The time complexity of LDP-MST is $O(NlogN + M^2)$, Qiu et al. further reduced it and proposed fast LDP-MST with $O(NlogN)$ (Qiu & Li, 2022). Li et al. introduced the polar coordinates to DPC to reduce the higher time complexity (Li, Ding, Xu, Du and Shi, 2022).

Another way is to reduce the time of similarity calculations. Tree-based structures are often used: Liu et al. developed a new data structure called VP-tree similar to ball-tree and used GPU to accelerate DPC (Liu et al., 2023). Parallel computing and cloud computing is effective: Niu et al. proposed a parallel grid-based density peak clustering for grouping trajectory data (Niu, Zheng, Fournier-Viger, & Wang, 2021). Zhang et al. proposed an efficient distributed DPC for clustering large datasets in MapReduce (Zhang, Chen, & Yu, 2016). Yang et al. proposed cloud and privacy-protecting based DPC (Yang, Liang, Zhang and Li, 2021).

Approximate distances avoid unnecessary calculations: Xu et al. used sparse search to reduce the computing times of similarities in DPC (Xu, Ding, Wang, Wang, & Jia, 2021). Xu et al. proposed a grid granulation framework to enable DenPEHC to cluster large-scale datasets (Xu et al., 2016). Chen et al. replaced local density by kNN-density (Chen et al., 2020). Sieranoja et al. used approximate k-nearest neighbor graph both for density and for delta calculation (Sieranoja & Fränti, 2019).

Reducing the number of data points participating in clustering is also a very good idea: Ding et al. used a sampling method to reduce the distance calculation and used approximate representatives to further enhance the clustering efficiency (Ding, Li et al., 2023). Laohakiat et al. used a fuzzy local clustering algorithm to generate micro-clusters and then merged them modified valley seeking algorithm (Laohakiat & Sa-Ing, 2021). Niu et al. proposed a two-step clustering algorithm with graph-augmented DPC (Niu, Zheng, Liu, & Wu, 2022). There are other strategies to accelerate DPC: Rasool et al. proposed index-based solutions for DPC. Xu et al. proposed grid-division and circle-division and used them to find cluster centers quickly.

Most clustering algorithms are designed based on a single machine, as we know, the memory of a single machine is limited, if clustering GB-level or TB-level data, a single machine is useless. cloud-based or multiple machines-based large-scale clustering algorithms are more practical and promising.

### 5.5. Parameter-less DPC

Most extend of DPC have parameters, if clustering real-world data without ground-truth labels, it is very difficult and time-wasting to fine-tune the values of parameters. Thus, parameter-less DPC is needed.

The related work of parameter-less DPC relatively has a small number. García-García et al. taken cluster validity index as the objective function for automatic parameter/center selection of DPC (García-García & García-Ródenas, 2021). Wang et al. used pseudo labels to replace true labels, used clustering results to replace predicted labels and constructed a neural network-like objective function to achieve the best parameters values (Wang, Pang, & Zhou, 2022). Zhang used shared k-nearest neighbors and conflict game to achieve adaptive DPC (Zhang, Du et al., 2021). Chowdhury et al. used adaptive optimal neighborhood size to compute density and find noises, generate Representative in remaining data points and the corresponding sub-clusters, and merged the sub-clusters (Chowdhury, Bhattacharyya, & Kalita, 2021). Gao et al. used a pattern search algorithm and jointly ranked the local density and relative distance to achieve adaptive DPC (Gao et al., 2022). Liu et al. used structural information from constraints to automatically obtain several potential cluster centers (Liu, Huang, Fei, Wang, & Liang, 2019). Mehmood et al. proposed a parameter-less DPC based on heat diffusion in an infinite domain (Mehmood et al., 2016). Flores et al. improved DPC by gap-based automatic center detection (Flores & Garza, 2020). d'Errico et al. proposed a non-parametric DPC by introducing topography (d'Errico, Facco, Laio, & Rodriguez, 2021). Liu et al. used KNN and aggregating strategy to construct an adaptive DPC (Yaohui et al., 2017).

Most adaptive strategies are heuristic and tailored to a particular algorithm. If the adaptive strategy of a specific clustering algorithm is transferred to another algorithm, it is often ineffective.

### 5.6. Clustering mixed data

Clustering mixed data is a well-known problem in the field of clustering algorithms. A few DPC-based approaches have been proposed in recent years to tackle this issue.

Du et al. presented a novel similarity criterion for mixed data and utilized DPC to improve the clustering outcomes. Their approach had high clustering accuracy and scalability (Du, Ding, & Xue, 2017).

Ding et al. proposed an entropy-based DPC for mixed-type data using a fuzzy neighborhood. They used the entropy weight method to determine the weights of different distance measures to improve the clustering accuracy (Ding, Du, Sun, Xu, & Xue, 2017).

These two methods are effective in clustering mixed data. However, there are still some concerns that need to be addressed, such as the computational efficiency of these algorithms when dealing with large datasets. Therefore, further studies are needed to develop more efficient clustering algorithms for mixed-type data.

### 5.7. Clustering imbalanced data

To address the challenge of clustering imbalanced data, Tong et al. introduced a novel 3D decision graph that can efficiently detect initial subcluster centers and noise points, thereby enabling the design of a merging method to group all subclusters (Tong, Wang and Liu, 2021). Meanwhile, Tao et al. proposed an adaptive weighted oversampling approach for imbalanced datasets using DPC with heuristic filtering (Tao et al., 2020). Yan et al. found that imbalanced images have similar decision graphs obtained by a lightweight DPC and applied DPC to Image segmentation (Yan et al., 2022). Besides, RangeTree is applied to accelerate RDP for large images due to the high complexity of DPC. Tao et al. used the SVDD boundary and DPC clustering technique-based oversampling approach for handling imbalanced and overlapped data (Tao, Chen et al., 2021). Mostafaei et al. first proposed a new technique for under-sampling based on the DPC from the majority

**Table 3**
The applications of DPC.

| Application | Authors | Content |
|---|---|---|
| Defect diagnosis | Sharma et al. (2022) | Proposed an adaptive mixture distance-based DPC and applied it to gearbox defect diagnosis. |
| | Shi, Deng et al. (2020) | Used DPC to improve the probabilistic neural network for identifying circuit fault diagnoses in multi-conditions. |
| | Wang, Wei, and Yang (2018) | Proposed a three-stage method to diagnose intelligent faults in three specific industrial cases. |
| Community detection | Bai et al. (2017) | Proposed an improved DPC algorithm that detects overlapping communities by incorporating the strength of linkage between nodes. |
| | Wang, Zuo et al. (2016) | Used DPC to identify social circles in social networks. |
| | Xu et al. (2019) | Proposed an adaptive DPC algorithm and applied it to detect overlapping communities in social networks. |
| | Lu, Shen et al. (2020) | Used nonnegative matrix factorization to improve DPC for community detection . |
| | Li, Chen et al. (2022) | Developed a stable community detection algorithm by combining DPC with label propagation. |
| | Zheng, Wang, Li, and Zhang (2019) | Effectively recommended relevant and diverse interests for users using DPC. |
| Bioinformatics | Kuhrova et al. (2016) | Introduced DPC and proposed a new algorithm that identifies the molecular force fields that disrupt folding structures. |
| | Henninger et al. (2017) | Employed DPC to estimate the number of hematopoietic stem cells (HSCs) in natural environments. |
| | Chen et al. (2018) | Used DPC to recommend valuable disease diagnoses and treatment plans to doctors and patients. |
| Image process | Jing et al. (2021) | Applied DPC to synthetic aperture radar image segmentation. |
| | Shang et al. (2021) | Applied DPC to target extraction. |
| | Shi et al. (2017) | Developed a novel image segmentation algorithm based on DPC. |
| | Tu et al. | Applied DPC to hyperspectral image classification (Tu et al., 2019) and hyperspectral anomaly detection (Tu et al., 2020). |
| | Jia, Tang, Zhu, and Li (2015) | Introduced improvements to the DPC algorithm for hyperspectral band selection. |
| | Sun, Geng, and Ji (2014) | Proposed a new band selection method called ECA, which filters out noisy data and improves accuracy, while also reducing computational complexity through the use of a ranking method. |
| | Mai et al. (2017) | Introduced a coherent optical receiver technology using an improved DPC algorithm, which achieved better classification performance. |
| Consumer behavior analysis | Wang, Chen et al. (2016) | Used DPC to obtain typical dynamics of consumer behavior. |
| Active learning | Wang, Min et al. (2017) | Used DPC to improve active learning. |
| | Shi, Yu et al. (2020) | Proposed an active version of DPC that considers both representativeness and informativeness for clustering. |
| Patient stratification | Li and Wong (2018) | Proposed a novel multi-objective algorithm for patient stratification using the DPC algorithm. |
| Indoor positioning | Meng, Yuan, Yan, and Zeng (2018) | Developed a wireless indoor positioning algorithm that utilizes neural network models and a novel clustering method. The authors proposed an RBF model by using a combination of the DPC algorithm and the LM algorithm. |
| Natural language processing (NLP) | Heimerl, John, Han, Koch, and Ertl (2016) | Utilized the DPC algorithm to identify the optimal number of clusters for a targeted set of documents in high-dimensional space. |
| | Wang, Zhang, Ding and Zou (2017) | Introduced DPC to evaluate the relevance and diversity of sentences. |

class on imbalanced datasets and combined it with SMOTE for clustering (Mostafaei & Tanha, 2023). Jiang et al. used DPC to select representative samples of the safe area, which is constructed by fuzzy entropy and fuzzy support (Jiang, Yang, & Qiu, 2022).

Although the first three issues of DPC have been effectively tackled, the remaining four persist as major challenges. To that end, we must make considerable efforts in these research fields, especially for large-scale data clustering and parameter-less clustering.

### 5.8. Applications

The DPC algorithm has been widely used in various fields, as exemplified by the following studies (see Table 3):

These applications demonstrate the effectiveness of DPC and its versatility in solving different data analysis problems.

## 6. Performances comparisons of several DPC-related algorithms

In this paper, although we review more than 110 papers, the authors rarely open source codes. We finally find 15 algorithms codes in GITHUB: DPC (Rodriguez & Laio, 2014), DPC-KNN[4] (Du et al., 2016), McDPC[5] (Wang, Wang, Zhang et al., 2020), FKNN-DPC[6] (Xie et al., 2016), SNNDPC[7] (Liu et al., 2018), DPC-DBFN[8] (Lotfi et al., 2020), FHC-DPC[9] (Guan et al., 2021), ADPC (Yan et al., 2019), DPCSA[10] (Yu et al., 2019), PLDPC (Wang et al., 2022), DPC-DLP[11] (Seyedi et al., 2019), Comparative DPC[12] (Li & Tang, 2018), DPC-CE[13] (Guo et al., 2022), VDPC[14] (Wang et al., 2023), and APC[15] (Wang, Wang, Pang et al., 2020). We also use 12 synthetic and 16 UCI datasets to evaluate their performances as shown in Tables 4 and 5. We fine-tune the parameters of all the used clustering algorithms and achieve the best results according to ARI and NMI.[16]

[4] https://github.com/mlyizhang/DPC-KNN-PCA.

[5] https://github.com/mlyizhang/Multi-center-DPC.

[6] https://github.com/liurui39660/SNNDPC.

[7] https://github.com/liurui39660/SNNDPC.

[8] https://github.com/abdulrahmanlotfi/DPC-DBFN.

[9] https://github.com/Guanjunyi/FHC-LDP-a-variant-of-density-peak-clustering-DPC.

[10] https://github.com/Yu123456/DPCSA.

[11] https://github.com/amjadseyedi/DPC-DLP.

[12] https://github.com/ZejianLi/ComparativeDensityPeaks.

[13] https://github.com/WJ-Guo/DPC-CE.

[14] https://github.com/mlyizhang/VDPC.

[15] https://github.com/mlyizhang/APC.

[16] https://github.com/mlyizhang/Clustering-evaluation.

**Table 4**
A comparison of 15 clustering algorithms on 12 synthetic datasets.

| Algorithms | ARI | NMI | ARI | NMI |
|---|---|---|---|---|
| | Dataset R15 | | Dataset Spiral | |
| DPC (Rodriguez & Laio, 2014) | **0.9928** | **0.9942** | **1.0000** | **1.0000** |
| DPC-KNN (Du et al., 2016) | **0.9928** | **0.9942** | 0.2652 | 0.3367 |
| McDPC (Wang, Wang, Zhang et al., 2020) | **0.9928** | **0.9942** | **1.0000** | **1.0000** |
| FKNN-DPC (Xie et al., 2016) | **0.9928** | **0.9942** | **1.0000** | **1.0000** |
| SNNDPC (Liu et al., 2018) | **0.9928** | **0.9942** | **1.0000** | **1.0000** |
| DPC-DBFN (Lotfi et al., 2020) | 0.9093 | 0.9647 | 0.0257 | 0.1426 |
| FHC-DPC (Guan et al., 2021) | 0.9857 | 0.9893 | **1.0000** | **1.0000** |
| ADPC (Yan et al., 2019) | 0.9928 | 0.9942 | **1.0000** | **1.0000** |
| DPCSA (Yu et al., 2019) | 0.9857 | 0.9893 | **1.0000** | **1.0000** |
| PLDPC (Wang et al., 2022) | **0.9928** | **0.9942** | **1.0000** | **1.0000** |
| DPC-DLP (Seyedi et al., 2019) | **0.9928** | **0.9942** | 0.0447 | 0.1463 |
| Comparative DPC (Li & Tang, 2018) | 0.4806 | 0.7600 | 0.0175 | 0.0816 |
| DPC-CE (Guo et al., 2022) | **0.9928** | **0.9942** | 0.9427 | 0.9456 |
| VDPC (Wang et al., 2023) | **0.9928** | **0.9942** | **1.0000** | **1.0000** |
| APC (Wang, Wang, Pang et al., 2020) | **0.9928** | **0.9942** | **1.0000** | **1.0000** |
| | Dataset Pathbased | | Dataset Compound | |
| DPC (Rodriguez & Laio, 2014) | 0.6600 | 0.4572 | 0.6368 | 0.5263 |
| DPC-KNN (Du et al., 2016) | 0.5448 | 0.7423 | 0.5448 | 0.7423 |
| McDPC (Wang, Wang, Zhang et al., 2020) | **1.0000** | **1.0000** | 0.7665 | 0.8427 |
| FKNN-DPC (Xie et al., 2016) | 0.4942 | 0.6047 | 0.8541 | 0.8159 |
| SNNDPC (Liu et al., 2018) | 0.9294 | 0.9529 | 0.7942 | 0.8204 |
| DPC-DBFN (Lotfi et al., 2020) | 0.4026 | 0.4998 | 0.7672 | 0.8310 |
| FHC-DPC (Guan et al., 2021) | 0.5772 | 0.7245 | 0.8506 | 0.9064 |
| ADPC (Yan et al., 2019) | 0.7049 | 0.7287 | 0.7839 | 0.8472 |
| DPCSA (Yu et al., 2019) | 0.5617 | 0.7295 | 0.7801 | 0.8510 |
| PLDPC (Wang et al., 2022) | 0.9294 | 0.9013 | 0.8522 | 0.8948 |
| DPC-DLP (Seyedi et al., 2019) | 0.4669 | 0.5293 | 0.6764 | 0.7492 |
| Comparative DPC (Li & Tang, 2018) | 0.3088 | 0.3477 | 0.2545 | 0.4478 |
| DPC-CE (Guo et al., 2022) | 0.4623 | 0.5491 | NaN | NaN |
| VDPC (Wang et al., 2023) | **0.9928** | **0.9942** | **1.0000** | **1.0000** |
| APC (Wang, Wang, Pang et al., 2020) | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| | Dataset Jain | | Dataset Flame | |
| DPC (Rodriguez & Laio, 2014) | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| DPC-KNN (Du et al., 2016) | 0.5692 | 0.5420 | **1.0000** | **1.0000** |
| McDPC (Wang, Wang, Zhang et al., 2020) | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| FKNN-DPC (Xie et al., 2016) | 0.3863 | 0.3784 | **1.0000** | **1.0000** |
| SNNDPC (Liu et al., 2018) | **1.0000** | **1.0000** | 0.9502 | 0.9768 |
| DPC-DBFN (Lotfi et al., 2020) | 0.0099 | 0.0351 | 0.0063 | 0.0319 |
| FHC-DPC (Guan et al., 2021) | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| ADPC (Yan et al., 2019) | 0.4224 | 0.3614 | 0.9065 | 0.8224 |
| DPCSA (Yu et al., 2019) | 0.3821 | 0.5997 | **1.0000** | **1.0000** |
| PLDPC (Wang et al., 2022) | 0.6723 | 0.7620 | 0.9502 | 0.8994 |
| DPC-DLP (Seyedi et al., 2019) | 0.8701 | 0.7511 | 0.9014 | 0.8475 |
| Comparative DPC (Li & Tang, 2018) | 0.5146 | 0.5052 | 0.0251 | 0.2188 |
| DPC-CE (Guo et al., 2022) | **1.0000** | **1.0000** | 0.8398 | 0.8548 |
| VDPC (Wang et al., 2023) | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| APC (Wang, Wang, Pang et al., 2020) | **1.0000** | **1.0000** | **1.0000** | **1.0000** |
| | Dataset D31 | | Dataset aggregation | |
| DPC (Rodriguez & Laio, 2014) | **0.9509** | **0.9659** | **1.0000** | **1.0000** |
| DPC-KNN (Du et al., 2016) | 0.1721 | 0.4580 | 0.9957 | 0.9884 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.9370 | 0.9579 | **1.0000** | **1.0000** |
| FKNN-DPC (Xie et al., 2016) | 0.9239 | 0.9524 | 0.9855 | 0.9797 |
| SNNDPC (Liu et al., 2018) | 0.9509 | 0.9525 | 0.9594 | 0.9681 |
| DPC-DBFN (Lotfi et al., 2020) | 0.9093 | 0.9647 | 0.9033 | 0.9423 |
| FHC-DPC (Guan et al., 2021) | 0.9390 | 0.9595 | **1.0000** | **1.0000** |
| ADPC (Yan et al., 2019) | 0.5906 | 0.8488 | **1.0000** | **1.0000** |
| DPCSA (Yu et al., 2019) | 0.9353 | 0.9573 | 0.8765 | 0.9140 |
| PLDPC (Wang et al., 2022) | 0.6086 | 0.8770 | 0.9710 | 0.9651 |
| DPC-DLP (Seyedi et al., 2019) | 0.5835 | 0.8262 | 0.6881 | 0.7522 |
| Comparative DPC (Li & Tang, 2018) | 0.2954 | 0.6511 | 0.4280 | 0.5958 |
| DPC-CE (Guo et al., 2022) | 0.9377 | 0.9587 | 0.9978 | 0.9957 |
| VDPC (Wang et al., 2023) | **0.9509** | **0.9659** | **1.0000** | **1.0000** |
| APC (Wang, Wang, Pang et al., 2020) | 0.9370 | 0.9579 | **1.0000** | **1.0000** |

(*continued on next page*)

**Table 4** (*continued*).

| | Dataset T8-8k | | Dataset T5-8k | |
|---|---|---|---|---|
| DPC (Rodriguez & Laio, 2014) | 0.5551 | 0.6892 | 0.7765 | 0.8211 |
| DPC-KNN (Du et al., 2016) | 0.1695 | 0.5418 | 0.0653 | 0.4719 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.5551 | 0.6892 | 0.7765 | 0.8209 |
| FKNN-DPC (Xie et al., 2016) | 0.6589 | 0.6727 | 0.2951 | 0.4983 |
| SNNDPC (Liu et al., 2018) | 0.7380 | 0.7966 | 0.5335 | 0.6633 |
| DPC-DBFN (Lotfi et al., 2020) | 0.5754 | 0.6521 | 0.3416 | 0.5697 |
| FHC-DPC (Guan et al., 2021) | 0.6269 | 0.7484 | 0.5944 | 0.7016 |
| ADPC (Yan et al., 2019) | 0.5212 | 0.6753 | **0.7766** | **0.8218** |
| DPCSA (Yu et al., 2019) | 0.6319 | 0.7645 | 0.4040 | 0.6355 |
| PLDPC (Wang et al., 2022) | 0.6507 | 0.7724 | 0.4704 | 0.6320 |
| DPC-DLP (Seyedi et al., 2019) | 0.4012 | 0.5691 | 0.1852 | 0.3558 |
| Comparative DPC (Li & Tang, 2018) | 0.2082 | 0.2926 | 0.1404 | 0.2989 |
| DPC-CE (Guo et al., 2022) | 0.4728 | 0.6681 | 0.7765 | 0.8212 |
| VDPC (Wang et al., 2023) | 0.1759 | 0.4338 | 0.1729 | 0.4464 |
| APC (Wang, Wang, Pang et al., 2020) | **0.9265** | **0.9121** | 0.7765 | 0.8209 |
| | Dataset T7-10k | | Dataset T4-8k | |
| DPC (Rodriguez & Laio, 2014) | 0.3805 | 0.5704 | 0.6209 | 0.7372 |
| DPC-KNN (Du et al., 2016) | 0.2301 | 0.5863 | 0.1828 | 0.5449 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.3803 | 0.5918 | 0.6279 | 0.7528 |
| FKNN-DPC (Xie et al., 2016) | 0.6221 | 0.7116 | 0.7476 | 0.8161 |
| SNNDPC (Liu et al., 2018) | 0.5636 | 0.6923 | 0.6323 | 0.7370 |
| DPC-DBFN (Lotfi et al., 2020) | 0.4450 | 0.6216 | 0.8167 | 0.8023 |
| FHC-DPC (Guan et al., 2021) | 0.6214 | 0.7574 | 0.6779 | 0.7582 |
| ADPC (Yan et al., 2019) | 0.4397 | 0.6248 | 0.6380 | 0.7249 |
| DPCSA (Yu et al., 2019) | 0.4431 | 0.6815 | 0.5153 | 0.6996 |
| PLDPC (Wang et al., 2022) | 0.4820 | 0.6689 | 0.6144 | 0.7134 |
| DPC-DLP (Seyedi et al., 2019) | 0.3074 | 0.5410 | 0.4168 | 0.5475 |
| Comparative DPC (Li & Tang, 2018) | 0.1610 | 0.3246 | 0.0022 | 0.0153 |
| DPC-CE (Guo et al., 2022) | 0.4574 | 0.6506 | 0.5503 | 0.6530 |
| VDPC (Wang et al., 2023) | 0.8921 | **0.8869** | 0.4188 | 0.6090 |
| APC (Wang, Wang, Pang et al., 2020) | **0.8996** | 0.8661 | **0.8745** | **0.8472** |

The best results are highlighted in boldface.

In Table 4, we use 12 common-used synthetic datasets: R15, Spiral, Pathbased, Compound, Jain, Flame, D31, Aggregation, T8-8k, T5-8k, T7-10k, and T4-8k, which have different size, shapes and densities. Among 15 algorithms, APC and VDPC achieve the best performances (11 of 12), the problem of identifying clusters with different densities is well addressed for synthetic datasets.

Table 5 presents the clustering results of sixteen UCI datasets including Iris, Wine, Ecoli, Seeds, Abalone, Librs Movement, Segment, Balance Scale, Breast, Vote, Sonar, Vehicle, Zoo, Thyroid, Banknote, and Landsat. The evaluation of the clustering performance shows that, in general, SNNDPC and PLDPC methods outperform other methods in most cases.

Of course, many DPC-related works perform better than the above 15 algorithms, but we do not find their codes or their open source code do not work. Opening codes is helpful to promote DPC, the other researchers can obtain more inspiration. Most DPC-related work uses synthetic and UCI datasets to evaluate their performances. Synthetic dataset clustering has achieved very good results, and it is unnecessary to use the synthetic dataset to verify the effectiveness of the algorithm if no strong reasons. Promoting the DPC algorithm to deal with data in realistic scenarios is a crucial step.

If one wants to continue to research the DPC, one should compare with the SOTA methods (especially highly cited papers shown in Fig. 3) in each topic, some papers do not do it, and it is very detrimental to the development of DPC algorithms.

## 7. Conclusions and future work

DPC has become a classical clustering algorithm in machine learning and it has also promoted the development of many other fields through a decade of growth. As researchers in the field of DPC, they want to know what issues have been solved, what issues still need to be, and what more future work is required for DPC. In this paper, we review the most important work of DPC to answer these questions. Specifically, we qualitatively and quantitatively analyze DPC-related work, we use research fields of DPC as a framework to review all the related work, and we also compare the 15 DPC-related works based on open source codes, we can see the current development of clustering synthetic and UCI datasets. The conclusions are as follows:

**What problems have been solved?** The improved DPC algorithms have achieved promising results in clustering synthetic datasets, the results of most synthetic datasets (2circle_noise, 2d-4c-no9, 2d-3c-no123, 2d-4c-no4, 2d-10c, 3d-20c-no0, 2spiral, Pathbased, Peal, Jain, Twenty, D2, Spiral, Aml28, R15, Diamond9, Compound, Blobs, Flame, Zelink1, Aggregation, D1, Atom, Cassini, Banana, Chainlink, Donutcurves, Donut3, S1, Simle2, DS5, Rings, D31, 3MC, Zelink6) achieve the accuracy of 100% by using an improved DPC. It is unnecessary to spend any more time analyzing and clustering these datasets. In addition, the algorithms represented by SNNDPC have also achieved very good results in clustering commonly used UCI data (see Table 5). For clustering UCI datasets, the competition in this field is very fierce. For example, clustering results of Iris by SNNDPC are $ARI = 0.9222$, and $ARI = 0.9600$ based on 3W-DPET (Yu et al., 2021). If we continue to focus on UCI datasets clustering, the results will be meaningful if they perform better than SNNDPC and 3W-DPET, otherwise, it is unnecessary. To say the least, the dataset Iris only contains shape information, even though we design a model in which the clustering results of Iris is 1.0000, it may not work when we take it to group iris in reality.

**What problems have not been solved?** Large-scale data clustering and parameter-less clustering are still unsolved issues, and particular types of data clustering, such as mixed-type data, unbalanced data, stream data, and time series data, still need to be addressed. In addition, privacy-protecting based clustering algorithms are a pressing issue to be resolved, trainable clustering algorithms such as deep clustering and adversarial clustering receive greater attention, and federated clustering algorithms are a hot topic.

Pre-training + Prompt + Fine-tuning models are effective in various NLP tasks. However, it is still unclear whether a similar approach can be applied to clustering algorithms. One challenge is that clustering

**Table 5**
A comparison of 15 clustering algorithms on 16 UCI datasets.

| Algorithms | ARI | NMI | ARI | NMI |
|---|---|---|---|---|
| | Dataset Iris | | Dataset Wine | |
| DPC (Rodriguez & Laio, 2014) | 0.6314 | 0.7112 | 0.3910 | 0.4308 |
| DPC-KNN (Du et al., 2016) | 0.7243 | 0.7747 | 0.2614 | 0.3336 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.8858 | 0.8705 | 0.5432 | 0.6812 |
| FKNN-DPC (Xie et al., 2016) | 0.9038 | 0.8851 | 0.7990 | 0.8074 |
| SNNDPC (Liu et al., 2018) | **0.9222** | **0.9144** | **0.8992** | **0.8782** |
| DPC-DBFN (Lotfi et al., 2020) | 0.5681 | 0.7612 | 0.4328 | 0.5761 |
| FHC-DPC (Guan et al., 2021) | 0.8177 | 0.8212 | 0.6855 | 0.7181 |
| ADPC (Yan et al., 2019) | 0.7613 | 0.7694 | 0.3203 | 0.4211 |
| DPCSA (Yu et al., 2019) | 0.9038 | 0.8851 | 0.7414 | 0.7528 |
| PLDPC (Wang et al., 2022) | **0.9222** | **0.9144** | 0.8685 | 0.8529 |
| DPC-DLP (Seyedi et al., 2019) | 0.9125 | 0.9037 | 0.3077 | 0.4594 |
| Comparative DPC (Li & Tang, 2018) | 0.6205 | 0.6891 | −0.0070 | 0.0917 |
| DPC-CE (Guo et al., 2022) | 0.6175 | 0.7049 | 0.2926 | 0.4049 |
| VDPC (Wang et al., 2023) | 0.8349 | 0.8513 | 0.3180 | 0.4226 |
| APC (Wang, Wang, Pang et al., 2020) | 0.8766 | 0.8581 | 0.3910 | 0.4308 |
| | Dataset Ecoli | | Dataset Seeds | |
| DPC (Rodriguez & Laio, 2014) | 0.4476 | 0.5507 | 0.7170 | 0.6744 |
| DPC-KNN (Du et al., 2016) | 0.1721 | 0.4580 | 0.6455 | 0.6381 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.7408 | 0.6978 | 0.7027 | 0.6982 |
| FKNN-DPC (Xie et al., 2016) | 0.5535 | 0.5630 | 0.7422 | 0.7008 |
| SNNDPC (Liu et al., 2018) | **0.7547** | **0.6979** | 0.7776 | 0.7423 |
| DPC-DBFN (Lotfi et al., 2020) | 0.6514 | 0.6379 | 0.4974 | 0.6022 |
| FHC-DPC (Guan et al., 2021) | 0.6974 | 0.6518 | 0.7289 | 0.6946 |
| ADPC (Yan et al., 2019) | 0.4398 | 0.5411 | 0.6989 | 0.6557 |
| DPCSA (Yu et al., 2019) | 0.4782 | 0.5189 | 0.7236 | 0.7151 |
| PLDPC (Wang et al., 2022) | **0.7547** | **0.6979** | **0.7890** | **0.7543** |
| DPC-DLP (Seyedi et al., 2019) | 0.2816 | 0.4399 | 0.4308 | 0.4652 |
| Comparative DPC (Li & Tang, 2018) | 0.5320 | 0.5758 | 0.0004 | 0.0269 |
| DPC-CE (Guo et al., 2022) | 0.7389 | 0.6917 | 0.4402 | 0.5372 |
| VDPC (Wang et al., 2023) | 0.4260 | 0.4600 | 0.5851 | 0.5688 |
| APC (Wang, Wang, Pang et al., 2020) | 0.4531 | 0.4630 | 0.7027 | 0.6982 |
| | Dataset Abalone | | Dataset Librs Movement | |
| DPC (Rodriguez & Laio, 2014) | 0.1507 | 0.1312 | 0.3193 | 0.5358 |
| DPC-KNN (Du et al., 2016) | 0.1311 | 0.1228 | 0.2596 | 0.5689 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.0551 | 0.0436 | 0.0211 | 0.0265 |
| FKNN-DPC (Xie et al., 2016) | 0.1050 | 0.1141 | 0.2533 | 0.5242 |
| SNNDPC (Liu et al., 2018) | 0.1490 | **0.1794** | 0.3927 | 0.6605 |
| DPC-DBFN (Lotfi et al., 2020) | **0.1732** | 0.1585 | 0.1159 | 0.3463 |
| FHC-DPC (Guan et al., 2021) | 0.0473 | 0.1023 | 0.2499 | 0.5444 |
| ADPC (Yan et al., 2019) | 0.0385 | 0.0914 | NaN | NaN |
| DPCSA (Yu et al., 2019) | 0.1033 | 0.1464 | 0.2486 | 0.5461 |
| PLDPC (Wang et al., 2022) | 0.1490 | **0.1794** | **0.4079** | **0.6754** |
| DPC-DLP (Seyedi et al., 2019) | 0.1027 | 0.1166 | 0.0829 | 0.3913 |
| Comparative DPC (Li & Tang, 2018) | 0.0001 | 0.0015 | 0.2879 | 0.5852 |
| DPC-CE (Guo et al., 2022) | 0.1014 | 0.1090 | NaN | NaN |
| VDPC (Wang et al., 2023) | 0.1046 | 0.1186 | 0.0930 | 0.6676 |
| APC (Wang, Wang, Pang et al., 2020) | 0.1507 | 0.1312 | 0.0476 | 0.1692 |
| | Dataset Segmentation | | Dataset Balance Scale | |
| DPC (Rodriguez & Laio, 2014) | 0.4218 | 0.6186 | 0.0845 | 0.2126 |
| DPC-KNN (Du et al., 2016) | 0.3751 | 0.5790 | 0.0000 | 0.2465 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.0551 | 0.0436 | 0.0211 | 0.0265 |
| FKNN-DPC (Xie et al., 2016) | 0.0285 | 0.1360 | 0.0236 | 0.0351 |
| SNNDPC (Liu et al., 2018) | 0.3670 | 0.6297 | 0.0000 | 0.0000 |
| DPC-DBFN (Lotfi et al., 2020) | 0.1947 | 0.4836 | 0.0872 | 0.1501 |
| FHC-DPC (Guan et al., 2021) | 0.5083 | 0.6656 | NaN | NaN |
| ADPC (Yan et al., 2019) | 0.0001 | 0.0188 | 0.2856 | 0.3699 |
| DPCSA (Yu et al., 2019) | 0.3667 | 0.5748 | 0.0000 | 0.0000 |
| PLDPC (Wang et al., 2022) | **0.5229** | **0.6937** | **0.5161** | **0.4332** |
| DPC-DLP (Seyedi et al., 2019) | 0.0000 | 0.0000 | 0.0331 | 0.0315 |
| Comparative DPC (Li & Tang, 2018) | 0.3028 | 0.4621 | 0.0018 | 0.0513 |
| DPC-CE (Guo et al., 2022) | 0.3093 | 0.5179 | NaN | NaN |
| VDPC (Wang et al., 2023) | 0.1922 | 0.5721 | 0.0000 | 0.2486 |
| APC (Wang, Wang, Pang et al., 2020) | 0.0011 | 0.5346 | −0.0091 | 0.0015 |

(*continued on next page*)

**Table 5** (*continued*).

| | Dataset Breast | | Dataset Vote | |
|---|---|---|---|---|
| DPC (Rodriguez & Laio, 2014) | 0.0358 | 0.0409 | 0.4698 | 0.3999 |
| DPC-KNN (Du et al., 2016) | −0.0468 | 0.0693 | 0.5210 | 0.4003 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.0551 | 0.0436 | 0.0551 | 0.0436 |
| FKNN-DPC (Xie et al., 2016) | 0.1284 | 0.0960 | 0.4905 | 0.4318 |
| SNNDPC (Liu et al., 2018) | 0.1535 | 0.0720 | 0.5368 | 0.4866 |
| DPC-DBFN (Lotfi et al., 2020) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| FHC-DPC (Guan et al., 2021) | 0.0472 | 0.0179 | 0.5367 | 0.4643 |
| ADPC (Yan et al., 2019) | 0.0102 | 0.0142 | 0.0000 | 0.0000 |
| DPCSA (Yu et al., 2019) | 0.0069 | 0.0040 | 0.5368 | 0.4866 |
| PLDPC (Wang et al., 2022) | **0.3327** | **0.5862** | **0.7776** | **0.7423** |
| DPC-DLP (Seyedi et al., 2019) | 0.0000 | 0.0000 | 0.0998 | 0.0689 |
| Comparative DPC (Li & Tang, 2018) | −0.0110 | 0.0122 | 0.6132 | 0.5026 |
| DPC-CE (Guo et al., 2022) | 0.1200 | 0.0636 | NaN | NaN |
| VDPC (Wang et al., 2023) | 0.0001 | 0.1721 | 0.0014 | 0.2024 |
| APC (Wang, Wang, Pang et al., 2020) | 0.1341 | 0.0721 | 0.2998 | 0.4005 |
| | Dataset Sonar | | Dataset Vehicle | |
| DPC (Rodriguez & Laio, 2014) | 0.0085 | 0.0105 | 0.0901 | 0.1335 |
| DPC-KNN (Du et al., 2016) | −0.0046 | 0.0049 | 0.0298 | 0.3381 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.0551 | 0.0436 | 0.0211 | 0.0265 |
| FKNN-DPC (Xie et al., 2016) | 0.0580 | 0.0436 | 0.1411 | 0.1380 |
| SNNDPC (Liu et al., 2018) | −0.0014 | 0.0143 | 0.1161 | 0.1526 |
| DPC-DBFN (Lotfi et al., 2020) | 0.0000 | 0.0000 | 0.0142 | 0.0616 |
| FHC-DPC (Guan et al., 2021) | 0.0443 | 0.0379 | 0.0798 | 0.1244 |
| ADPC (Yan et al., 2019) | 0.0000 | 0.0000 | 0.0006 | 0.0188 |
| DPCSA (Yu et al., 2019) | −0.0031 | 0.0027 | 0.1083 | 0.1929 |
| PLDPC (Wang et al., 2022) | 0.0671 | 0.1347 | **0.1649** | 0.2632 |
| DPC-DLP (Seyedi et al., 2019) | 0.0220 | 0.0228 | 0.1398 | 0.2294 |
| Comparative DPC (Li & Tang, 2018) | **0.0789** | 0.0787 | 0.1035 | 0.1389 |
| DPC-CE (Guo et al., 2022) | 0.0191 | 0.1424 | 0.1041 | 0.1899 |
| VDPC (Wang et al., 2023) | 0.0000 | 0.2292 | 0.0490 | **0.3290** |
| APC (Wang, Wang, Pang et al., 2020) | 0.0381 | **0.2329** | 0.1140 | 0.2443 |
| | Dataset Zoo | | Dataset Thyroid | |
| DPC (Rodriguez & Laio, 2014) | 0.4809 | 0.7133 | 0.0453 | 0.1353 |
| DPC-KNN (Du et al., 2016) | 0.0542 | 0.0133 | 0.2435 | 0.2930 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.0551 | 0.0436 | 0.2000 | 0.3225 |
| FKNN-DPC (Xie et al., 2016) | 0.7147 | 0.7410 | 0.0000 | 0.0000 |
| SNNDPC (Liu et al., 2018) | 0.8470 | 0.8158 | 0.7726 | 0.6598 |
| DPC-DBFN (Lotfi et al., 2020) | 0.3805 | 0.4770 | 0.3570 | 0.4578 |
| FHC-DPC (Guan et al., 2021) | 0.7122 | 0.8001 | 0.3701 | 0.3932 |
| ADPC (Yan et al., 2019) | 0.4743 | 0.5848 | 0.1984 | 0.2158 |
| DPCSA (Yu et al., 2019) | 0.8374 | 0.8197 | 0.3185 | 0.3283 |
| PLDPC (Wang et al., 2022) | **0.9113** | **0.8546** | **0.8152** | **0.6697** |
| DPC-DLP (Seyedi et al., 2019) | 0.0345 | 0.3265 | 0.6972 | 0.6303 |
| Comparative DPC (Li & Tang, 2018) | 0.8118 | 0.8059 | 0.2302 | 0.2677 |
| DPC-CE (Guo et al., 2022) | 0.7537 | 0.7472 | 0.6541 | 0.4857 |
| VDPC (Wang et al., 2023) | 0.1347 | 0.2939 | 0.0005 | 0.2673 |
| APC (Wang, Wang, Pang et al., 2020) | −0.0528 | 0.1581 | 0.2234 | 0.2644 |
| | Dataset Banknote | | Dataset Landsat | |
| DPC (Rodriguez & Laio, 2014) | 0.6200 | 0.6023 | 0.4727 | 0.5554 |
| DPC-KNN (Du et al., 2016) | 0.0586 | 0.1086 | 0.4159 | 0.5488 |
| McDPC (Wang, Wang, Zhang et al., 2020) | 0.2000 | 0.3225 | 0.2785 | 0.4042 |
| FKNN-DPC (Xie et al., 2016) | 0.0268 | 0.2970 | 0.3129 | 0.3833 |
| SNNDPC (Liu et al., 2018) | 0.6056 | 0.5600 | **0.6570** | **0.6741** |
| DPC-DBFN (Lotfi et al., 2020) | 0.0000 | 0.0000 | 0.0257 | 0.1426 |
| FHC-DPC (Guan et al., 2021) | 0.9624 | 0.9317 | 0.6368 | 0.6522 |
| ADPC (Yan et al., 2019) | 0.2885 | 0.3972 | 0.3384 | 0.4577 |
| DPCSA (Yu et al., 2019) | **0.9653** | **0.9359** | 0.4860 | 0.6058 |
| PLDPC (Wang et al., 2022) | 0.7484 | 0.7191 | **0.6570** | **0.6741** |
| DPC-DLP (Seyedi et al., 2019) | 0.3075 | 0.3119 | 0.5220 | 0.6287 |
| Comparative DPC (Li & Tang, 2018) | 0.0735 | 0.0485 | 0.0013 | 0.0079 |
| DPC-CE (Guo et al., 2022) | 0.5754 | 0.6423 | 0.5224 | 0.6239 |
| VDPC (Wang et al., 2023) | 0.0796 | 0.2440 | 0.0000 | 0.3716 |
| APC (Wang, Wang, Pang et al., 2020) | 0.5597 | 0.5305 | 0.0000 | 0.3716 |

The best results are highlighted in boldface.

algorithms often have specific requirements based on the data type and distribution of the dataset, which is different from NLP tasks where the pre-training models can work on various types of texts.

**What else need to do in the future?** For large-scale data clustering, most DPC-related algorithms are published in famous journals, but most of them work in a single machine. We know that the memory of a single machine is limited, so it is better to focus on cloud computing-based clustering algorithms, which can process GB-level or TB-level data. Spark provides a big data-based computing framework, when clustering algorithms meet Spark, the problem of large-scale data clustering may be well addressed, and how to deploy DPC to Spark needs to be analyzed. A possible solution is shown below, we can introduce the granular ball (Xia et al., 2020) into spark-based DPC, *i.e.,* we use the results of one standard k-means iteration to construct Resilient Distributed Datasets (RDD), and group all the data points into stable points and active points. At last, according to different kinds of data points, we take different operations to obtain final clustering results. This approach benefits from the scalability and fault-tolerance offered by Spark, which allows it to process and group massive amounts of data at scale. However, when deploying DPC algorithms on Spark, it is also important to consider factors such as load balancing, data shuffling, and network bandwidth to ensure efficient processing and accurate clustering results.

For parameter-less DPC, previous work often tailored a parameter-less strategy for a specific algorithm, if apply this strategy into another clustering algorithm, it may not work. We need to design a general parameter-less strategy for most clustering algorithms rather than a certain extent of DPC. Based on this idea, we can introduce deep clustering to DPC, which uses optimization theory to obtain the best parameter values. Utilizing the internal evaluation index is another option. An internal evaluation index that is consistent with DPC can be designed based on the local density and relative distance characteristics of the algorithm without significantly increasing complexity. Consequently, true parameter-less clustering algorithms can be achieved.

Combining federated learning with clustering algorithms to form federated clustering is a promising solution to achieve privacy-preserving clustering. Federated learning allows multiple parties to jointly learn a model without sharing their raw data. Federated clustering can be implemented by dividing the data into disjoint subsets and assigning them to different parties. Each party then performs clustering on its own data subset and sends a summary of its clustering results to a central aggregator node that aggregates the cluster summaries and updates the global clustering model. Federated cloud clustering can be achieved by deploying federated clustering on cloud computing infrastructure, which will enable the use of resources and technologies such as big data processing and distributed computing to handle larger and more complex datasets.

We can also use the framework of Pre-training + Prompt + Fine-tuning to improve DPC or other clustering algorithms, Ultimately, we need to establish a unified big model to group multi-sources, large-scale, and different types of data into different clusters. This approach requires careful consideration of how different types of data and distributions are integrated and represented in the model to capture the inherent structure and relationships between the data points. We think that a unified big model is an important topic for clustering algorithms in the future.

## CRediT authorship contribution statement

**Yizhang Wang:** Conceptualization, Methodology, Funding acquisition, Investigation, Project administration, Writing – original draft, Writing – review & editing. **Jiaxin Qian:** Investigation, Writing – review & editing. **Muhammad Hassan:** Writing – review & editing. **Xinyu Zhang:** Investigation. **Tao Zhang:** Investigation. **Chao Yang:** Investigation. **Xingxing Zhou:** Investigation. **Fengjin Jia:** Investigation.

## 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.

## Data availability

The data that has been used is confidential.

## Acknowledgments

This research is supported by the China Postdoctoral Science Foundation (Grant No. 2022M713064), Lvyangjinfeng Excellent Doctoral Program of Yangzhou (Grant No. YZLYJFJH2021YXBS105), Innovation and Entrepreneurship Program of Jiangsu Province (Grant No. JSSCBS20211048), and Jiangsu Provincial Universities of Natural Science General Program (Grant Nos. 21KJB520021, 20KJB520007).

## References

Abbas, M., El-Zoghabi, A., & Shoukry, A. (2021). DenMune: Density peak based clustering using mutual nearest neighbors. *Pattern Recognition*, *109*, Article 107589.

Bai, X., Yang, P., & Shi, X. (2017). An overlapping community detection algorithm based on density peaks. *Neurocomputing*, *226*, 7–15.

Bian, Z., Chung, F., & Wang, S. (2020). Fuzzy density peaks clustering. *IEEE Transactions on Fuzzy Systems*, *29*(7), 1725–1738.

Chen, Y., Hu, X., Fan, W., Shen, L., Zhang, Z., Liu, X., et al. (2020). Fast density peak clustering for large scale data based on kNN. *Knowledge-Based Systems*, *187*, Article 104824.

Chen, J., Li, K., Rong, H., Bilal, K., Yang, N., & Li, K. (2018). A disease diagnosis and treatment recommendation system based on big data mining and cloud computing. *Information Sciences*, *435*, 124–149.

Cheng, D., Huang, J., Zhang, S., Zhang, X., & Luo, X. (2021). A novel approximate spectral clustering algorithm with dense cores and density peaks. *IEEE transactions on systems, man, and cybernetics: systems*, *52*(4), 2348–2360.

Cheng, D., Zhang, S., & Huang, J. (2020). Dense members of local cores-based density peaks clustering algorithm. *Knowledge-Based Systems*, *193*, Article 105454.

Cheng, D., Zhu, Q., Huang, J., Wu, Q., & Yang, L. (2019). Clustering with local density peaks-based minimum spanning tree. *IEEE Transactions on Knowledge and Data Engineering*, *33*(2), 374–387.

Chowdhury, H. A., Bhattacharyya, D. K., & Kalita, J. K. (2021). UIFDBC: Effective density based clustering to find clusters of arbitrary shapes without user input. *Expert Systems with Applications*, *186*, Article 115746.

Coates, A., Ng, A., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *International conference on artificial intelligence and statistics* (pp. 215–223).

d'Errico, M., Facco, E., Laio, A., & Rodriguez, A. (2021). Automatic topography of high-dimensional data sets by non-parametric density peak clustering. *Information Sciences*, *560*, 476–492.

Ding, S., Du, M., Sun, T., Xu, X., & Xue, Y. (2017). An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood. *Knowledge-Based Systems*, *133*, 294–313.

Ding, S., Du, W., Xu, X., Shi, T., Wang, Y., & Li, C. (2023). An improved density peaks clustering algorithm based on natural neighbor with a merging strategy. *Information Sciences*, *624*, 252–276.

Ding, S., Li, C., Xu, X., Ding, L., Zhang, J., Guo, L., et al. (2023). A sampling-based density peaks clustering algorithm for large-scale data. *Pattern Recognition*, *136*, Article 109238.

Du, M., Ding, S., & Jia, H. (2016). Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems*, *99*, 135–145.

Du, M., Ding, S., Xu, X., & Xue, Y. (2018). Density peaks clustering using geodesic distances. *International Journal of Machine Learning and Cybernetics*, *9*, 1335–1349.

Du, M., Ding, S., & Xue, Y. (2017). A novel density peaks clustering algorithm for mixed data. *Pattern Recognition Letters*, *97*, 46–53.

Du, M., Ding, S., & Xue, Y. (2018). A robust density peaks clustering algorithm using fuzzy neighborhood. *International Journal of Machine Learning and Cybernetics*, *9*, 1131–1140.

Du, M., Ding, S., Xue, Y., & Shi, Z. (2019). A novel density peaks clustering with sensitivity of local density and density-adaptive metric. *Knowledge and Information Systems*, *59*, 285–309.

Fan, J., Jia, P., & Ge, L. (2020). Mk-NNG-dpc: density peaks clustering based on improved mutual K-nearest-neighbor graph. *International Journal of Machine Learning and Cybernetics*, *11*, 1179–1195.

Fang, F., Qiu, L., & Yuan, S. (2020). Adaptive core fusion-based density peak clustering for complex data with arbitrary shapes and densities. *Pattern Recognition*, *107*, Article 107452.

Fang, X., Xu, Z., Ji, H., Wang, B., & Huang, Z. (2022). A grid-based density peaks clustering algorithm. *IEEE Transactions on Industrial Informatics*.

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE conference on computer vision and pattern recognition workshop* (pp. 178–178).

Flores, K. G., & Garza, S. E. (2020). Density peaks clustering with gap-based automatic center detection. *Knowledge-Based Systems*, *206*, Article 106350.

Gao, T., Chen, D., Tang, Y., Du, B., Ranjan, R., Zomaya, A. Y., et al. (2022). Adaptive density peaks clustering: Towards exploratory EEG analysis. *Knowledge-Based Systems*, *240*, Article 108123.

Gao, Z., Lin, H., Tan, C., Wu, L., Li, S., et al. (2021). Git: Clustering based on graph of intensity topology. arXiv preprint arXiv:2110.01274.

García-García, J. C., & García-Ródenas, R. (2021). A methodology for automatic parameter-tuning and center selection in density-peak clustering methods. *Soft Computing*, *25*, 1543–1561.

Guan, J., Li, S., He, X., & Chen, J. (2023). Clustering by fast detection of main density peaks within a peak digraph. *Information Sciences*.

Guan, J., Li, S., He, X., Zhu, J., & Chen, J. (2021). Fast hierarchical clustering of local density peaks via an association degree transfer method. *Neurocomputing*, *455*, 401–418.

Guo, Z., Huang, T., Cai, Z., & Zhu, W. (2018). A new local density for density peak clustering. In *Advances in knowledge discovery and data mining* (pp. 426–438). Springer International Publishing.

Guo, W., Wang, W., Zhao, S., Niu, Y., Zhang, Z., & Liu, X. (2022). Density peak clustering with connectivity estimation. *Knowledge-Based Systems*, *243*, Article 108501.

He, Y., Wu, Y., Qin, H., Huang, J. Z., & Jin, Y. (2021). Improved I-nice clustering algorithm based on density peaks mechanism. *Information Sciences*, *548*, 177–190.

Heimerl, F., John, M., Han, Q., Koch, S., & Ertl, T. (2016). DocuCompass: Effective exploration of document landscapes. In *IEEE conference on visual analytics science and technology* (pp. 11–20).

Henninger, J., Santoso, B., Hans, S., Durand, E., Moore, J., Mosimann, C., et al. (2017). Clonal fate mapping quantifies the number of haematopoietic stem cells that arise during development. *Nature Cell Biology*, *19*(1), 17–27.

Hou, J., & Zhang, A. (2019). Enhancing density peak clustering via density normalization. *IEEE Transactions on Industrial Informatics*, *16*(4), 2477–2485.

Hou, J., Zhang, A., & Qi, N. (2020). Density peak clustering based on relative density relationship. *Pattern Recognition*, *108*, Article 107554.

Jia, S., Tang, G., Zhu, J., & Li, Q. (2015). A novel ranking-based clustering approach for hyperspectral band selection. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(1), 88–102.

Jiang, M., Yang, Y., & Qiu, H. (2022). Fuzzy entropy and fuzzy support-based boosting random forests for imbalanced data. *Applied Intelligence*, *52*(4), 4126–4143.

Jing, W., Jin, T., & Xiang, D. (2021). Fast superpixel-based clustering algorithm for SAR image segmentation. *IEEE Geoscience and Remote Sensing Letters*, *19*, 1–5.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Kuhrova, P., Best, R. B., Bottaro, S., Bussi, G., Sponer, J., Otyepka, M., et al. (2016). Computer folding of RNA tetraloops: identification of key force field deficiencies. *Journal of Chemical Theory and Computation*, *12*(9), 4534–4548.

Laohakiat, S., & Sa-Ing, V. (2021). An incremental density-based clustering framework using fuzzy local clustering. *Information Sciences*, *547*, 404–426.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, C., Chen, H., Li, T., & Yang, X. (2022). A stable community detection approach for complex network based on density peak clustering and label propagation. *Applied Intelligence*, *52*(2), 1188–1208.

Li, C., Ding, S., Xu, X., Du, S., & Shi, T. (2022). Fast density peaks clustering algorithm in polar coordinate system. *Applied Intelligence*, *52*(12), 14478–14490.

Li, Y., Sun, L., & Tang, Y. (2022). DPC-fsc: An approach of fuzzy semantic cells to density peaks clustering. *Information Sciences*, *616*, 88–107.

Li, Y., Sun, L., Tang, Y., & You, W. (2022). A review of related density peaks clustering approaches. In *International conference on intelligent human-machine systems and cybernetics* (pp. 145–149).

Li, Z., & Tang, Y. (2018). Comparative density peaks clustering. *Expert Systems with Applications*, *95*, 236–247.

Li, X., & Wong, K. (2018). Evolutionary multiobjective clustering and its applications to patient stratification. *IEEE transactions on cybernetics*, *49*(5), 1680–1693.

Li, R., Yang, X., Qin, X., & Zhu, W. (2019). Local gap density for clustering high-dimensional data with varying densities. *Knowledge-Based Systems*, *184*, Article 104905.

Liang, Z., & Chen, P. (2016). Delta-density based clustering with a divide-and-conquer strategy: 3Dc clustering. *Pattern Recognition Letters*, *73*, 52–59.

Liu, Z., Gong, S., Su, Y., Wan, C., Zhang, Y., & Yu, G. (2023). Improving density peaks clustering through GPU acceleration. *Future Generation Computer Systems*, *141*, 399–413.

Liu, R., Huang, W., Fei, Z., Wang, K., & Liang, J. (2019). Constraint-based clustering by fast search and find of density peaks. *Neurocomputing*, *330*, 223–237.

Liu, R., Wang, H., & Yu, X. (2018). Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Information Sciences*, *450*, 200–226.

Lotfi, A., Moradi, P., & Beigy, H. (2020). Density peaks clustering based on density backbone and fuzzy neighborhood. *Pattern Recognition*, *107*, Article 107449.

Lu, H., Shen, Z., Sang, X., Zhao, Q., & Lu, J. (2020). Community detection method using improved density peak clustering and nonnegative matrix factorization. *Neurocomputing*, *415*, 247–257.

Lu, J., Zhao, Y., Tan, K.-L., & Wang, Z. (2020). Distributed density peaks clustering revisited. *IEEE Transactions on Knowledge and Data Engineering*, *34*(8), 3714–3726.

Mahmood, T., & Ali, Z. (2022). Fuzzy superior mandelbrot sets. *Soft Computing*, *26*(18), 9011–9020.

Mai, X., Liu, J., Wu, X., Zhang, Q., Guo, C., Yang, Y., et al. (2017). Stokes space modulation format classification based on non-iterative clustering algorithm for coherent optical receivers. *Optics Express*, *25*(3), 2038–2050.

Mehmood, R., Zhang, G., Bie, R., Dawood, H., & Ahmad, H. (2016). Clustering by fast search and find of density peaks via heat diffusion. *Neurocomputing*, *208*, 210–217.

Meng, H., Yuan, F., Yan, T., & Zeng, M. (2018). Indoor positioning of RBF neural network based on improved fast clustering algorithm combined with LM algorithm. *IEEE Access*, *7*, 5932–5945.

Mostafaei, S. H., & Tanha, J. (2023). Ouboost: boosting based over and under sampling technique for handling imbalanced data. *International Journal of Machine Learning and Cybernetics*, 1–19.

Ni, L., Luo, W., Zhu, W., & Liu, W. (2019). Clustering by finding prominent peaks in density space. *Engineering Applications of Artificial Intelligence*, *85*, 727–739.

Niu, X., Zheng, Y., Fournier-Viger, P., & Wang, B. (2021). Parallel grid-based density peak clustering of big trajectory data. *Applied Intelligence*, 1–16.

Niu, X., Zheng, Y., Liu, W., & Wu, C. Q. (2022). On a two-stage progressive clustering algorithm with graph-augmented density peak clustering. *Engineering Applications of Artificial Intelligence*, *108*, Article 104566.

Parmar, M., Wang, D., Zhang, X., Tan, A.-H., Miao, C., Jiang, J., et al. (2019). REDPC: A residual error-based density peak clustering algorithm. *Neurocomputing*, *348*, 82–96.

Pizzagalli, D. U., Gonzalez, S. F., & Krause, R. (2019). A trainable clustering algorithm based on shortest paths from density peaks. *Science advances*, *5*(10), 1231–1248.

Pourbahrami, S., Khanli, L. M., & Azimpour, S. (2020). Improving neighborhood construction with apollonius region algorithm based on density for clustering. *Information Sciences*, *522*, 227–240.

Qin, X., Han, X., Chu, J., Zhang, Y., Xu, X., Xie, J., et al. (2021). Density peaks clustering based on jaccard similarity and label propagation. *Cognitive Computation*, *13*, 1609–1626.

Qiu, T., & Li, Y. (2022). Fast LDP-mst: an efficient density-peak-based clustering method for large-size datasets. *IEEE Transactions on Knowledge and Data Engineering*, *35*, 4767–4780.

Rasool, Z., Aryal, S., Bouadjenek, M. R., & Dazeley, R. (2023). Overcoming weaknesses of density peak clustering using a data-dependent similarity measure. *Pattern Recognition*, *137*, Article 109287.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *science*, *344*(6191), 1492–1496.

Samaria, F. S., & Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. In *IEEE workshop on applications of computer vision* (pp. 138–142).

Seyedi, S. A., Lotfi, A., Moradi, P., & Qader, N. N. (2019). Dynamic graph-based label propagation for density peaks clustering. *Expert Systems with Applications*, *115*, 314–328.

Shang, X., Yang, T., Han, S., Song, M., & Xue, B. (2021). Interference-suppressed and cluster-optimized hyperspectral target extraction based on density peak clustering. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 4999–5014.

Sharma, K. K., Seal, A., Yazidi, A., & Krejcar, O. (2022). A new adaptive mixture distance-based improved density peaks clustering for gearbox fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–16.

Shi, Y., Chen, Z., Qi, Z., Meng, F., & Cui, L. (2017). A novel clustering-based image segmentation via density peaks algorithm with mid-level feature. *Neural Computing and Applications*, *28*, 29–39.

Shi, J., Deng, Y., & Wang, Z. (2020). Analog circuit fault diagnosis based on density peaks clustering and dynamic weight probabilistic neural network. *Neurocomputing*, *407*, 354–365.

Shi, Y., Yu, Z., Cao, W., Chen, C. P., Wong, H.-S., & Han, G. (2020). Fast and effective active clustering ensemble based on density peak. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(8), 3593–3607.

Sieranoja, S., & Fränti, P. (2019). Fast and general density peaks clustering. *Pattern Recognition Letters*, *128*, 551–558.

Su, Z., & Denoeux, T. (2018). BPEC: Belief-peaks evidential clustering. *IEEE Transactions on Fuzzy Systems*, *27*(1), 111–123.

Sun, K., Geng, X., & Ji, L. (2014). Exemplar component analysis: A fast band selection method for hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, *12*(5), 998–1002.

Sun, L., Qin, X., Ding, W., & Xu, J. (2022). Nearest neighbors-based adaptive density peaks clustering with optimized allocation strategy. *Neurocomputing*, *473*, 159–181.

Sun, L., Qin, X., Ding, W., Xu, J., & Zhang, S. (2021). Density peaks clustering based on k-nearest neighbors and self-recommendation. *International Journal of Machine Learning and Cybernetics*, *12*, 1913–1938.

Tao, X., Chen, W., Zhang, X., Guo, W., Qi, L., & Fan, Z. (2021). SVDD boundary and DPC clustering technique-based oversampling approach for handling imbalanced and overlapped data. *Knowledge-Based Systems*, *234*, Article 107588.

Tao, X., Guo, W., Ren, C., Li, Q., He, Q., Liu, R., et al. (2021). Density peak clustering using global and local consistency adjustable manifold distance. *Information Sciences*, *577*, 769–804.

Tao, X., Li, Q., Guo, W., Ren, C., He, Q., Liu, R., et al. (2020). Adaptive weighted over-sampling for imbalanced datasets based on density peaks clustering with heuristic filtering. *Information Sciences*, *519*, 43–73.

Tong, W., Liu, S., & Gao, X.-Z. (2021). A density-peak-based clustering algorithm of automatically determining the number of clusters. *Neurocomputing*, *458*, 655–666.

Tong, W., Wang, Y., & Liu, D. (2021). An adaptive clustering algorithm based on local-density peaks for imbalanced data without parameters. *IEEE Transactions on Knowledge and Data Engineering*, *29*, 3419–3432.

Tu, B., Yang, X., Li, N., Zhou, C., & He, D. (2020). Hyperspectral anomaly detection via density peak clustering. *Pattern Recognition Letters*, *129*, 144–149.

Tu, B., Zhang, X., Kang, X., Wang, J., & Benediktsson, J. A. (2019). Spatial density peak clustering for hyperspectral image classification with noisy labels. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(7), 5085–5097.

Wang, Y., Chen, Q., Kang, C., & Xia, Q. (2016). Clustering of electricity consumption behavior dynamics toward big data applications. *IEEE transactions on smart grid*, *7*(5), 2437–2447.

Wang, M., Min, F., Zhang, Z.-H., & Wu, Y.-X. (2017). Active learning through density clustering. *Expert Systems with Applications*, *85*, 305–317.

Wang, Y., Pang, W., & Zhou, J. (2022). An improved density peak clustering algorithm guided by pseudo labels. *Knowledge-Based Systems*, *252*, Article 109374.

Wang, Y., Wang, D., Pang, W., Miao, C., Tan, A.-H., & Zhou, Y. (2020). A systematic density-based clustering method using anchor points. *Neurocomputing*, *400*, 352–370.

Wang, Y., Wang, D., Zhang, X., Pang, W., Miao, C., Tan, A.-H., et al. (2020). Mcdpc: multi-center density peak clustering. *Neural Computing and Applications*, *32*, 13465–13478.

Wang, Y., Wang, D., Zhou, Y., Zhang, X., & Quek, C. (2023). VDPC: Variational density peak clustering algorithm. *Information Sciences*, *621*, 627–651.

Wang, Y., Wei, Z., & Yang, J. (2018). Feature trend extraction and adaptive density peaks search for intelligent fault diagnosis of machines. *IEEE Transactions on Industrial Informatics*, *15*(1), 105–115.

Wang, Y., & Yang, Y. (2021). Relative density-based clustering algorithm for identifying diverse density clusters effectively. *Neural Computing and Applications*, *33*, 10141–10157.

Wang, B., Zhang, J., Ding, F., & Zou, Y. (2017). Multi-document news summarization via paragraph embedding and density peak clustering. In *International conference on asian language processing* (pp. 260–263).

Wang, M., Zuo, W., & Wang, Y. (2016). An improved density peaks-based clustering method for social circle discovery in social networks. *Neurocomputing*, *179*, 219–227.

Wechsler, H., Phillips, J. P., Bruce, V., Soulie, F. F., & Huang, T. S. (2012). Face recognition: From theory to applications. *vol. 163*, Springer Science & Business Media.

Wei, X., Peng, M., Huang, H., & Zhou, Y. (2023). An overview on density peaks clustering. *Neurocomputing*, *554*, Article 126633.

Wu, C., Peng, Q., Lee, J., Leibnitz, K., & Xia, Y. (2021). Effective hierarchical clustering based on structural similarities in nearest neighbor graphs. *Knowledge-Based Systems*, *228*, Article 107295.

Xia, S., Peng, D., Meng, D., Zhang, C., Wang, G., Giem, E., et al. (2020). Ball k-means: Fast adaptive clustering with no bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(1), 87–99.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Xie, J., Gao, H., Xie, W., Liu, X., & Grant, P. W. (2016). Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors. *Information Sciences*, *354*, 19–40.

Xu, X., Ding, S., Wang, Y., Wang, L., & Jia, W. (2021). A fast density peaks clustering algorithm with sparse search. *Information Sciences*, *554*, 61–83.

Xu, T., & Jiang, J. (2022). A graph adaptive density peaks clustering algorithm for automatic centroid selection and effective aggregation. *Expert Systems with Applications*, *195*, Article 116539.

Xu, M., Li, Y., Li, R., Zou, F., & Gu, X. (2019). EADP: An extended adaptive density peaks clustering for overlapping community detection in social networks. *Neurocomputing*, *337*, 287–302.

Xu, J., Wang, G., & Deng, W. (2016). Denpehc: Density peak based efficient hierarchical clustering. *Information Sciences*, *373*, 200–218.

Xu, J., Wang, G., Li, T., Deng, W., & Gou, G. (2017). Fat node leading tree for data stream clustering with density peaks. *Knowledge-Based Systems*, *120*, 99–117.

Yan, M., Chen, Y., Chen, Y., Zeng, G., Hu, X., & Du, J. (2022). A lightweight weakly supervised learning segmentation algorithm for imbalanced image based on rotation density peaks. *Knowledge-Based Systems*, *244*, Article 108513.

Yan, H., Wang, L., & Lu, Y. (2019). Identifying cluster centroids from decision graph automatically using a statistical outlier detection method. *Neurocomputing*, *329*, 348–358.

Yang, X., Cai, Z., Li, R., & Zhu, W. (2021). GDPC: Generalized density peaks clustering algorithm based on order similarity. *International Journal of Machine Learning and Cybernetics*, *12*, 719–731.

Yang, Y., Cai, J., Yang, H., & Zhao, X. (2022). Density clustering with divergence distance and automatic center selection. *Information Sciences*, *596*, 414–438.

Yang, H., Liang, S., Zhang, Y., & Li, X. (2021). Cloud-based privacy-and integrity-protecting density peaks clustering. *Future Generation Computer Systems*, *125*, 758–769.

Yaohui, L., Zhengming, M., & Fang, Y. (2017). Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy. *Knowledge-Based Systems*, *133*, 208–220.

Yu, H., Chen, L., & Yao, J. (2021). A three-way density peak clustering method based on evidence theory. *Knowledge-Based Systems*, *211*, Article 106532.

Yu, D., Liu, G., Guo, M., Liu, X., & Yao, S. (2019). Density peaks clustering based on weighted local density sequence and nearest neighbor assignment. *IEEE Access*, *7*, 34301–34317.

Zhang, Y., Chen, S., & Yu, G. (2016). Efficient distributed density peaks for clustering large data sets in mapreduce. *IEEE Transactions on Knowledge and Data Engineering*, *28*(12), 3218–3230.

Zhang, Q., Dai, Y., & Wang, G. (2023). Density peaks clustering based on balance density and connectivity. *Pattern Recognition*, *134*, Article 109052.

Zhang, R., Du, T., Qu, S., & Sun, H. (2021). Adaptive density-based clustering algorithm with shared KNN conflict game. *Information Sciences*, *565*, 344–369.

Zhang, R., Miao, Z., Tian, Y., & Wang, H. (2022). A novel density peaks clustering algorithm based on hopkins statistic. *Expert Systems with Applications*, *201*, Article 116892.

Zhang, Z., Zhu, Q., Zhu, F., Li, J., Cheng, D., Liu, Y., et al. (2021). Density decay graph-based density peak clustering. *Knowledge-Based Systems*, *224*, Article 107075.

Zheng, J., Wang, S., Li, D., & Zhang, B. (2019). Personalized recommendation based on hierarchical interest overlapping community. *Information Sciences*, *479*, 55–75.

Zhou, Z., Si, G., Sun, H., Qu, K., & Hou, W. (2022). A robust clustering algorithm based on the identification of core points and KNN kernel density estimation. *Expert Systems with Applications*, *195*, Article 116573.

Zhu, Y., Ting, K. M., Jin, Y., & Angelova, M. (2022). Hierarchical clustering that takes advantage of both density-peak and density-connectivity. *Information Systems*, *103*, Article 101871.