Aiming at the problems that the initial cluster centers are randomly selected and the number of clusters is manually determined in traditional clustering algorithm, which results in unstable clustering results, we propose an improved clustering algorithm based on density peak and nearest neighbors. Firstly, an improved density peak clustering method is proposed to optimize the cutoff distance and local density of data points. It avoids that random selection of initial cluster centers is easy to fall into the local optimal solution. Furthermore, a K-value selection method is presented to choose the optimal number of clusters, which is determined by the sum of the squared errors within the clusters. Finally, we employ the idea of the K-nearest neighbors to carry out the assignment for outliers. Experiments on the UCI real data sets indicate that our proposed algorithm can achieve better clustering results compared with several known algorithms.
algorithm [23] uses heat diffusion to calculate the local density of data points to reduce the influence of cutoff distance on clustering result. The ADPC-KNN algorithm [24] is proposed based on the KNN to calculate the local density and automatically selects the initial cluster centers. The DPADN algorithm [25] uses a continuous function to redesign the local density, which can automatically determine the cluster centers. A multi-feature fusion with adaptive graph learning model [26] is developed, which is an unsupervised algorithm to address the issue of person re-identification. An adaptive approach is constructed [27] which is effective in simultaneously learning the affinity graph and feature fusion, resulting in better clustering results. The novel RCFE method [28] is proposed where the graph and feature fusion, resulting in better clustering re-
sults. IX_the novel RCFE method [28] is proposed where the number of clusters is guaranteed to converge to the ground truth via a rank constraint on the Laplacian matrix. Finally, there are some other recent work involving density peak and truth via a rank constraint on the Laplacian matrix. Finally, there are some other recent work involving density peak and

2 Mathematical Problems in Engineering

2.1. The DPC Algorithm. The DPC algorithm indicates that cluster centers are characterized by a higher density than their neighbors and a relatively large distance from data points with higher densities. In this approach, for each data point \( x_i \), two attributes need to be calculated: local density \( \rho_i \) and distance \( \sigma_i \).

There are two ways for calculating the local density \( \rho_i \) of \( x_i \), namely, cutoff kernel and Gaussian kernel. The cutoff kernel is often used when data size is large, and it is defined as follows:

\[
\rho_i = \sum_j \chi(d_{ij}/\sigma_i).
\]

(1)

where \( d_{ij} \) is the distance between data points \( x_i \) and \( x_j \), and the cutoff distance \( \sigma_i > 0 \) which needs to be set manually. Usually arranging all distance \( d_{ij} \), in ascending order, a value between the first 1% and 2% is selected as the cutoff distance. When the data size is small, the Gaussian kernel is used as

\[
\rho_i = \sum_j e^{-(d_{ij}^2/\sigma_i^2)}.
\]

(2)

Comparing the two local density calculation methods, it is found that local density obtained by the cutoff kernel is a discrete value, while the other is a continuous value. The use of the cutoff kernel may cause conflicts when calculating the local density of different data points; therefore, the Gaussian kernel is more generally used to calculate the local density.

The distance \( \sigma_i \) of data point \( x_i \) is measured by computing the minimum distance between the point \( x_i \) and any other point with higher density:

\[
\sigma_i = \min_{j : \rho_j \neq 0} (d_{ij}).
\]

(3)

For the point with highest density, it conventionally takes

\[
\sigma_i = \max_j (d_{ij}).
\]

(4)

Assume that the original data set is shown in Figure 1. The local density \( \rho_i \) and distance \( \sigma_i \) are calculated for each data point \( x_i \), and a decision graph is illustrated in Figure 2. According to the DPC algorithm, data points that have both large \( \rho \) and \( \sigma \) are most likely to be the cluster centers, which are located in the upper right in Figure 2.

In order to determine the cluster centers and its number clearly, the quantity \( \gamma \) in [13] which is a comprehensive consideration of \( \rho \) and \( \sigma \) is defined as

\[
\gamma_i = \rho_i \sigma_i.
\]

(5)

According to \( \gamma_i \) and number of data points, a new decision graph is shown in Figure 3. Obviously, the larger the value \( \gamma \), the more likely the corresponding data point to be the cluster center. It can be seen from Figure 3 that \( \gamma \) of noncluster center is relatively smooth, and there is an obvious jump of the value \( \gamma \) between the cluster center and the noncluster center. It is demonstrated that the data point above the dotted line is the cluster center. Consequently, the remaining data points are assigned to the cluster which has the nearest point with larger local density.

2.2. Set the Cutoff Distance. It is not advisable to manually set the cutoff distance \( d_c \) based on experiences for diverse data structures. It affects the local density of data points, which in turn leads to different clustering results. Assumption \( p \) is a percentage, we define

\[
d_c = \text{dis}(N \ast p).
\]

(6)

\( N \) is the number of distances between any two data points, and \( \text{dis} \) is an ascending order of these distances. Here, it takes the \( N \ast p \) position of \( \text{dis} \) sequence as the value of the cutoff distance \( d_c \). Figure 4 shows clustering results by running the DPC algorithm in terms of different percentage \( p \).

We can see that the cutoff distance \( d_c \) obtained at different percentage \( p \) has a huge impact on clustering results. Hence, the setting of the cutoff distance should be flexible, and an appropriate \( d_c \) should be selected to certain data structure.
In information theory, Shannon entropy is used to measure the uncertainty of a system. The greater the entropy, the greater the uncertainty [37]. Similarly, the uncertainty of data distribution can be expressed by entropy. Suppose that the local density is

\[ \rho_i = \sum_j e^{-\left(\frac{d_{ij}}{\sigma}\right)^2}, \]  

(7)

\( \sigma > 0 \) is an impact factor which is used to optimize the cutoff distance \( d_c \). For a data set \( X = \{x_1, x_2, \ldots, x_n\} \), the local density of each point is \( \{\rho_1, \rho_2, \ldots, \rho_n\} \). Considering the information entropy to evaluate the rationality of local density estimation [38],

\[ H = -\sum_{i=1}^{n} \frac{\rho_i}{Z} \log\left(\frac{\rho_i}{Z}\right), \]  

(8)

\[ Z = \sum_{i=1}^{n} \rho_i, \]  

(9)

\( Z \) is the normalized factor. The relationship between information entropy \( H \) and impact factor \( \sigma \) is shown in Figure 5. We can see that when \( \sigma \) starts to increase from 0, the information entropy \( H \) first decreases rapidly, then increases slowly, and finally tends to be stable. It follows that \( \sigma \) is the optimal value when the entropy is the lowest.

2.3. Optimize the Local Density. The DPC algorithm only considers the global structure of data set, and the clustering result is not good enough when it comes to unevenly distributed data sets [41]. When the data distribution is relatively concentrated and the density of each cluster is quite different, the change of local density will lead to the difficulty of selecting the correct cluster centers, thus affecting the final clustering results. Based on the idea of KNN algorithm, the nearest neighbors are introduced into local density calculation. The closer the data point is to the target point, the more contribution it makes to the local density of the target point. The new local density is defined as follows:

\[ \rho_i = \sum_{j=1}^{m} e^{-\left(\frac{d_{ij}}{\sigma}\right)^2} + \sum_{j=m+1}^{n} e^{-\left(\frac{d_{ij}}{(j-m)^2}\right)}, \]  

(11)

where \( m = 2 \times K \), \( d_{ij} \) represents the ascending order of the distances between data point \( i \) and other data points, \( n \) is the number of data points in data set, \( m \) is the number of nearest neighbors of data point \( i \), and \( K \) denotes the number of clusters.

3. The K-Value Selection Method

In the DPC algorithm, the boundary between cluster center point and noncluster center point may be unclear, which cause different people to choose different numbers of clusters according to their own experiences. Similarly, in the K-means algorithm, the number of clusters also needs to be preset by the user based on experiences. However, it is often difficult to determine the value of \( K \).

In order to solve the problem that it is difficult to choose the optimal number of clusters, we proposed a \( K \)-value selection method based on the SSE and the ET-SSE algorithm in [42].
Figure 4: Clustering results of the DPC algorithm under different percentage (p): (a) $p = 1\%$, (b) $p = 2\%$, (c) $p = 4\%$, and (d) $p = 6\%$.

Figure 5: The relationship between information entropy $H$ and impact factor $\sigma$. 
Ordinarily, the $K$-means algorithm employs SSE to measure clustering quality as

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} (d(x, c_i))^2,$$  \hspace{1cm} (12)$$

where $d(x, c)$ represents the Euclidean distance from data points $x$ and $c$, and $K$ is the number of clusters. Let $C_i$ denote the set of all data points in the $i$-th cluster, and $c_i$ represents the cluster center of $i$-th cluster. Usually, the relationship between SSE and $K$-value is demonstrated in Figure 6(a). It is obvious that with the increase of $K$, the SSE will decrease and eventually stabilize. Therefore, the $K$-value corresponding to the obvious “elbow point” in $K$ can be used as the optimal number of clusters. Nevertheless, sometimes there will be situation as shown in Figure 6(b), where there is no obvious “elbow point” in the figure. In this case, it will be more difficult to select the optimal $K$-value, which will affect the final clustering results.

To solve the problem that the “elbow point” in Figure 6(b) is not clear, the exponential function $e^x$ is introduced into the SSE formula. Using the property that the $e^x$ function is sensitive to positive number, the SSE values of different clusters are scaled down to further improve the difference degree of SSE values, when $K$-value is not equal to the actual number of clusters. Meanwhile, in order to prevent the exponential explosion, an adjustment factor $\delta$ is added to update the weight of the SSE value. A new calculation formula of the SSE is defined as follows:

$$SSE_K = \sum_{i=1}^{K} e^\delta \sum_{x \in C_i} (d(x, c_i))^2 + \max_{i=1}^{K} \sum_{x \in C_i} (d(x, c_i))^2 \delta,$$  \hspace{1cm} (13)$$

where max means to find the largest SSE value in $K$ clusters. In order to reduce the influence of manual parameters on the clustering result, based on a large number of experiments on different data sets, it is found that when

$$\delta = \max_{i=1}^{K} \sum_{x \in C_i} (d(x, c_i))^2,$$  \hspace{1cm} (14)$$

the “elbow point” in Figure 6 is the most obvious. Then, the predicted number of clusters is closest to the actual number of clusters, and the clustering effect is the best.

4. The Improved Clustering Algorithm

Aiming at the problem of unstable clustering results caused by randomly selecting initial cluster centers in traditional $K$-means algorithm, this paper proposed an improved clustering algorithm based on the density peak and nearest neighbors. Firstly, using information entropy to improve the DPC algorithm, we find the optimal cutoff distance of data set and then calculate the local density of the data points; additionally, the optimal number of clusters is obtained by the $K$-value selection method proposed in Section 3. Finally, according to the descending order of values in (5), the top $K$ corresponding data points are selected as the initial cluster centers. The $K$-means algorithm is used for iterative clustering.

4.1. Weighted Euclidean Distance. Let set $X = \{x_1, x_2, \ldots, x_n\}$ be a data set containing $n$ data points, and each data point contains $m$-dimensional attribute characteristics, where $x_{ip}$ ($i = 1, \ldots, n; p = 1, \ldots, m$) represents the $p$-th attribute characteristic of the $i$-th data point.

In order to remove the unit restrictions of different attributes in the original data and avoid its impact on the clustering results, the original data need to be normalized and converted into pure numerical data. After the normalization, each attribute is in the same order of magnitude, which is suitable for comprehensive comparative evaluation. The normalization formula is as follows:

$$x'_{ip} = \frac{x_{ip} - \min(x_{ip})}{\max(x_{ip}) - \min(x_{ip})},$$ \hspace{1cm} (15)$$

where $\max(x_{ip})$ and $\min(x_{ip})$, respectively, represent the maximum and minimum of the $p$ dimension attributes in a data point.

The traditional $K$-means algorithm employs Euclidean distance to measure the similarity between data points. It is applicable to the uniform measurement of each attribute in the data point, and it treats the difference between different attributes equally. But in practice, the contribution of different attributes to the clustering results is quite different. To solve this problem, a weighted Euclidean distance is used to measure the similarity between data points. Let the weight of the $l$-th dimension attribute of the data set be

$$w_l = \frac{\sum_{i=1}^{n} (x_{ip} - \bar{x}_p)^2}{\sum_{p=1}^{m} \sum_{i=1}^{n} (x_{ip} - \bar{x}_p)^2}, \hspace{1cm} l = 1, 2, \ldots, m,$$ \hspace{1cm} (16)$$

where $\bar{x}_p$ denotes the average value of the $p$-th dimension attribute. Then, the weighted Euclidean distance between the data points $x_1$ and $x_2$ is

$$d(x_1, x_2) = \sum_{l=1}^{m} w_l (x_{1l} - x_{2l})^2.$$ \hspace{1cm} (17)$$

4.2. The Framework of the Proposed Algorithm. The $K$-means algorithm is sensitive to outliers [43], and the improved DPC algorithm can exclude the influence of outliers on the selection of initial cluster centers. However, $K$-means is an iterative clustering algorithm, and each iteration will generate new cluster centers. The outliers will have an impact on the generation of new cluster centers. Hence, it is necessary to distinguish the outliers in each cluster.

Let $C = (C_1, C_2, \ldots, C_k)$ be the $K$ clusters after first iteration, and the cluster centers are $c = (c_1, c_2, \ldots, c_k)$. The average distance of the $i$-th cluster is

$$\text{MeanDist}_i = \frac{\sum_{j=1}^{n} \sum_{x \in C_i} d(x_j, c_i)}{\sum (C_i)}.$$ \hspace{1cm} (18)$$
where \( \text{sum}(C_i) \) is the number of data points in the \( i \)-th cluster. According to the average distance \( \text{MeanDist} \) of each cluster, the data points in the cluster are divided into core points and outliers. If the distance between the data point \( x_j \) and its cluster center is less than \( \text{MeanDist} \), \( x_j \) is regarded as a core point; otherwise it is an outlier. The average value of the data points marked as core points in the cluster is calculated as the new cluster center of the next iteration, and the outliers do not participate in the calculation of the new cluster center.

To ensure that the assignment of outliers is closer to real situation and improve the clustering accuracy, the idea of KNN is introduced into the assignment of outliers [44–46]. Suppose that \( x_i \) is an outlier, calculate the distance sequence \( \text{dist} = \{\text{dist}_1, \text{dist}_2, \ldots, \text{dist}_r\} \) from \( x_i \) to each core point by (17). Then, \( \text{dist}' = \{\text{dist}_1', \text{dist}_2', \ldots, \text{dist}_r'\} \) is an ascending order, and we select the first \( 2^K \) corresponding core points as the core neighbor point \( \text{core} = \{\text{core}_1, \text{core}_2, \ldots, \text{core}_{2^K}\} \). Since the core points have completed the assignment, the cluster category corresponding to each data point is counted from the sequence \( \text{Core} \), and the cluster with the largest number of statistics is regarded as the cluster to which the outlier \( x_i \) belongs. If there is more than one cluster category with the largest number of statistics, the cluster belonging to the nearest center point is marked as the cluster category of \( x_i \), according to the distance from \( x_i \) to the center point.

In summary, the steps of our proposed algorithm are as follows:

1. Input the original data set \( X = \{x_1, x_2, \ldots, x_n\} \).
2. Normalize the data set \( X \) according to (15) to obtain the processed data set \( X^* \).
3. According to (17), the weighted Euclidean distance between each data point in data set \( X^* \) is obtained.
4. Calculate \( \rho_i \) and \( \sigma_i \) of each data point in data set \( X^* \) according to (3), (4), and (11) and determine the optimal cutoff distance \( d_i \) according to the information entropy in Section 2.2.
5. Calculate \( y_i = \rho_i\sigma_i \) and arrange the \( y \) value sequence in descending order to obtain the sequence \( y' \).
6. Determine the number of clusters \( K \) according to the \( K \)-value selection method presented in Section 3.
7. The first \( K \) corresponding data points are selected from the sequence \( y' \) as the initial cluster centers.
8. The distances from each data point to each cluster center are calculated, and a data point is classified into the cluster with the nearest cluster center.
9. Calculate the mean distance \( \text{MeanDist} \) of each cluster according to (18). The data points in each cluster are divided into core points and outliers.
10. Calculate the average value of the core points in each cluster as the new cluster centers of the next iteration. The idea of nearest neighbors in KNN algorithm is used to re-assign the outliers.
11. When the cluster centers no longer change, the algorithm terminates and outputs the clustering result. Otherwise, go to Step 8.

Furthermore, a flowchart for the overall process is provided in Figure 7.

4.3. The Time Complexity Analysis. The time complexity of the improved DPC algorithm mainly depends on the local density \( \rho \) and distance \( \sigma \), which is \( O(n^2) \). The time complexity of \( K \)-value selection method mainly comes from the sum of the squared error (SSE), which is \( O(Kn^2) \). The time complexity of the idea based on nearest neighbors in KNN algorithm is used to re-assign the outliers.

Therefore, the total time complexity of our proposed algorithm is \( O(n^2) \).

5. Experiments

5.1. Experimental Environment and the Data Sets. The hardware environment is based on Windows 10 Professional 64-bit, Inter Core i3-4000M CPU, 2.40 GHz, and 4 GB memory. The proposed algorithm is implemented in MATLAB R2011a. The data of Wine, Pima, WDBC, Iris, and
Parkinsons in the UCI real data sets [47] are used as the experimental data sets, as shown in Table 1.

5.2. The Evaluation Measures. This paper uses accuracy (ACC), adjusted Rand index (ARI), and adjusted mutual information (AMI) to evaluate the performance of the clustering algorithms. Assume that $P_j$ is a known manually labeled cluster, and $C_j$ is a cluster generated by the clustering algorithm. The ACC calculation formula is as follows:

$$\text{ACC}(P_j, C_j) = \frac{|P_j \cap C_j|}{|P_j|}$$

where the value range of ACC is $[0, 1]$ and the value range of ARI and AMI is $[-1, 1]$. The specific meanings and calculations of ARI and AMI are referred to [48, 49]. The three evaluation measures are positively correlated with clustering performance. The larger the value, the better the clustering performance of the algorithm.
5.3. Experimental Results and Discussion. In this paper, we compare our proposed algorithm with several existing algorithms: K-means, DPC [13], CNACS-K-means [50], and DCC-K-means [51]. Due to the influence of the initial cluster centers, the K-means algorithm takes the mean value of 20 times. The experimental results on the UCI data sets are shown in Tables 2–4. The data in bold are the best experimental results on this data set.

As shown in Tables 2–4, the algorithm we proposed is generally better than the other four comparison algorithms in terms of the three evaluation measures. It achieves the best experimental results especially on the Wine and Iris datasets.

In the comparison of ACC in Table 2, our algorithm is slightly worse than the K-means and DCC-K-means algorithms on the Pima and WDBC data sets, but is better than the other two comparison algorithms. However, the situation is just the opposite on the Parkinsons data set.

As for ARI comparison in Table 3, the proposed algorithm is lower than the CNACS-K-means on the Pima data set, whereas it is higher than the other three comparison algorithms. On the WDBC data set, it is only lower than K-means and DCC-K-means algorithms by 1.7%. Furthermore, it is lower than CNACS-K-means and 0.02% difference from DPC on the Parkinsons data set, but it is higher than the other two comparison algorithms.

In Table 4, the comparison of AMI shows that our algorithm is superior to the other four comparison algorithms on the Pima data set. On the WDBC data set, it is close to the K-means and DCC-K-means algorithms, and superior to the other two comparison algorithms. Although our algorithm outperforms the DPC and CNACS-K-means algorithms, it is indeed inferior to the other two comparison algorithms on the Parkinsons data set.

6. Conclusion

Focusing on the problems of randomly selecting initial cluster centers, manually determining the number of clusters, and not considering the influence of outliers on clustering process, this paper proposes an improved clustering
algorithm based on the density peak and the nearest neighbors. Our algorithm uses an improved DPC algorithm to determine the initial cluster centers and calculates the sum of the squared errors within the clusters to find the optimal cluster number K. Moreover, the average distance within the clusters and the nearest neighbor idea are combined to determine the outliers and its assignment. The experimental results show that the proposed algorithm can achieve better clustering results on the UCI real data sets. However, it is not sure that the algorithm can be applied to large-scale data sets. In future, how to improve the stability and running efficiency of our algorithm for large-scale data set will be the focus.

**Data Availability**

The data sets used to support the findings of this study are included within the article.

**Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this paper.

**Acknowledgments**

This work was supported by the National Natural Science Foundation of China (61902262) and Handan City Science and Technology Research and Development Program (19422031008-15).

**References**

[1] A. Saxena, M. Prasad, A. Gupta et al., “A review of clustering techniques and developments,” *Neurocomputing*, vol. 267, pp. 664–681, 2017.

[2] G. M. Mazzeo, E. Masciari, and C. Zaniolo, “A fast and accurate algorithm for unsupervised clustering around centroids,” *Information Sciences*, vol. 400-401, pp. 63–90, 2017.

[3] A. Fahad, N. Alshatri, Z. Tari et al., “A survey of clustering algorithms for big data: taxonomy and empirical analysis,” *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–279, 2014.

[4] A. K. Jain. “Data clustering: 50 years beyond K-means,” *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[5] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, California, Berkeley, June 21-July 18, 1965.

[6] J. C. Yang and C. Zhao, “Survey on K-Means clustering algorithm,” *Computer Engineering and Applications*, vol. 55, no. 23, pp. 7–14, 2019.

[7] H. Ismikhan and I. k-means, “A k-means algorithm for an enhanced version of the k-means algorithm,” *Pattern Recognition*, vol. 79, pp. 402–413, 2018.

[8] J. S. Lei, T. Jiang, K. Wu, H. Z. Du, G. K. Zhu, and Z. Q. Wang, “Robust K-means algorithm with automatically splitting and merging clusters and its applications for surveillance data,” *Multimedia Tools and Applications*, vol. 75, no. 19, pp. 12043–12059, 2016.

[9] G. Tzortzis and A. Likas, “The MinMax k-Means clustering algorithm,” *Pattern Recognition*, vol. 47, no. 7, pp. 2505–2516, 2014.

[10] J. Qi, Y. Yu, L. Wang, J. Liu, and Y. Wang, “An effective and efficient hierarchical K-means clustering algorithm,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 8, Article ID 155014771772862, 2017.

[11] G. Zhang, C. Zhang, and H. Zhang, “Improved K-means algorithm based on density canopy,” *Knowledge-Based Systems*, vol. 145, pp. 289–297, 2018.

[12] H. Xie, L. Zhang, C. P. Lim et al., “Improving K-means clustering with enhanced firefly algorithms,” *Applied Soft Computing*, vol. 84, Article ID 105763, 2019.

[13] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[14] B. Wu, H. L. Lu, and H. J. Jiang, “Adaptive density peak clustering algorithm,” *Computer Applications*, vol. 40, no. 06, pp. 1654–1661, 2020.

[15] Y. Zhang, S. Chen, and G. Yu, “Efficient distributed density peaks for clustering large data sets in MapReduce,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3218–3230, 2016.

[16] K. G. Flores and S. E. Garza, “Density peaks clustering with gap-based automatic center detection,” *Knowledge-Based Systems*, vol. 206, Article ID 106350, 2020.

[17] S. L. Wang, D. K. Wang, C. Y. Li, Y. Li, and G. Y. Ding, “Clustering by fast search and find of density peaks with data field,” *Chinese Journal of Electronics*, vol. 25, no. 3, pp. 397–402, 2016.

[18] H. B. Du, A. Z. Bai, and L. J. Zhu, “K-means algorithm based on improved density peak algorithm,” *Statistics & Decisions*, vol. 34, no. 18, pp. 20–24, 2018.

[19] D. H. Yu, G. J. Liu, M. Z. Guo, X. Y. Liu, and S. Yao, “Density peaks clustering based on weighted local density sequence and nearest neighbor assignment,” *IEEE Access*, vol. 7, pp. 34301–34317, 2019.

[20] J. Y. Xie, H. C. Gao, W. X. Xie, X. H. Liu, and P. W. Grant, “Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors,” *Information Sciences*, vol. 354, pp. 19–40, 2016.

[21] S. Seyedi, Lotfi, P. Moradi, and N. Qader, “Dynamic graph-based label propagation for density peaks clustering,” *Expert Systems with Applications*, vol. 115, pp. 314–328, 2019.

[22] H. Lu, Z. Shen, X. S. Sang, Q. H. Zhao, and J. F. Lu, “Community detection method using improved density peak clustering and nonnegative matrix factorization,” *Neurocomputing*, vol. 415, pp. 247–257, 2020.

[23] M. Mehmood, G. Z. Zhang, R. F. Bie, D. Dawood, and H. Ahmad, “Clustering by fast search and find of density peaks via heat diffusion,” *Neurocomputing*, vol. 208, pp. 210–217, 2016.

[24] Y. H. Liu, Z. G. Ma, and F. Yu, “Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy,” *Knowledge-Based Systems*, vol. 133, pp. 208–220, 2017.

[25] W. N. Tong, S. Liu, and X. Z. Gao, “A density-peak-based clustering algorithm of automatically determining the number of clusters,” *Neurocomputing*, vol. 458, pp. 655–666, 2021.

[26] R. W. Zhou, J. J. Chang, L. Shi, Y. D. Shen, Y. Yang, and F. P. Nie, “Person reidentification via multi-feature fusion with adaptive graph learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1592–1601, 2019.
[27] Z. H. Li, F. P. Nie, X. J. Chang, Y. Yang, C. Q. Zhang, and N. Sebe, "Dynamic affinity graph construction for spectral clustering using multiple features," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 12, pp. 6323–6332, 2018.

[28] Z. H. Li, F. P. Nie, X. J. Chang, F. P. Nie, H. X. Zhang, and Y. Yang, "Rank-constrained spectral clustering with flexible embedding," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 12, pp. 6073–6082, 2018.

[29] T. H. Fan, Z. F. Yao, L. Z. Han, B. H. Liu, and L. Lv, "Density peaks clustering based on k-nearest neighbors sharing," Concurrency and Computation: Practice and Experience, vol. 33, no. 5, Article ID e5993, 2021.

[30] R. Zhang, T. Du, S. N. Qu, and H. W. Sun, "Adaptive density-based clustering algorithm with shared KNN conflict game," Information Sciences, vol. 565, pp. 344–369, 2021.

[31] Z. Zhou, G. Q. Si, H. D. Sun, K. Qu, and W. C. Hou, "A robust clustering algorithm based on the identification of core points and KNN kernel density estimation," Expert Systems with Applications, vol. 195, Article ID 116573, 2022.

[32] M. Parmar, D. Wang, X. F. Zhang et al., "REDPC: a residual error-based density peak clustering algorithm," Neurocomputing, vol. 348, pp. 82–96, 2019.

[33] J. Y. Guan, S. Li, X. X. He, and J. J. Chen, "Peak-graph-based fast density peak clustering for image segmentation," IEEE Signal Processing Letters, vol. 28, pp. 897–901, 2021.

[34] M. D. Parmar, W. Pang, D. H. Hao et al., "FREDPC: a feasible residual error-based density peak clustering algorithm with the fragment merging strategy," IEEE Access, vol. 7, pp. 89789–89804, 2019.

[35] C. W. Li, H. M. Chen, T. R. Li, and X. L. Yang, "A stable community detection approach for complex network based on density peak clustering and label propagation," Applied Intelligence, vol. 52, no. 2, pp. 1188–1208, 2022.

[36] M. Parmar, D. Wang, A. H. Tan, C. Y. Miao, J. H. Jiang, and Y. Zhou, "A Novel Density Peak Clustering Algorithm Based on Squared Residual Error," in Proceedings of the 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), pp. 43–48, IEEE, Shenzhen, China, 15-17 December 2017.

[37] S. L. Wang, W. Y. Gan, D. Y. Li, and D. R. Li, "Data field for hierarchical clustering," International Journal of Data Warehousing and Mining, vol. 7, no. 4, pp. 43–63, 2011.

[38] W. Y. Gan and C. Liu, "An improved clustering algorithm for search density peaks," Journal of Intelligent Systems, vol. 12, no. 02, pp. 229–236, 2017.

[39] I. Bárány and V. H. Vu, "Central limit theorems for Gaussian polytopes," Annals of Probability, vol. 35, no. 4, pp. 1593–1621, 2006.

[40] S. L. Wang, D. K. Wang, C. Y. Li, and Y. Li, "Comment on "Clustering by fast search and find of density peaks," 2015, https://arxiv.org/abs/1501.04267.

[41] Z. Yang and H. J. Wang, "Improved density peak clustering algorithm based on weighted K-nearest neighbors," Application Research of Computers, vol. 37, no. 03, pp. 667–671, 2020.

[42] J. R. Wang, X. Ma, and G. L. Duan, "Improved K-means clustering k value selection algorithm," Computer Engineering and Applications, vol. 55, no. 08, pp. 27–33, 2019.

[43] G. J. Gan and M. K. P. Ng, "k-means clustering with outlier removal," Pattern Recognition Letters, vol. 90, pp. 8–14, 2017.

[44] J. Y. Xie, H. C. Gao, and W. X. Xie, "K-nearest neighbor optimization based density peak fast search clustering algorithm," Science China Information Sciences, vol. 46, no. 02, pp. 258–280, 2016.

[45] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance," Journal of Machine Learning Research, vol. 11, pp. 2837–2854, 2010.

[46] M. J. Du, S. F. Ding, and H. J. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," Knowledge-Based Systems, vol. 99, pp. 135–145, 2016.

[47] D. Dua and C. Graff, UCI Machine Learning Repository, Chennai, Tamil Nadu 600002, 2019.

[48] Y. H. Lu and C. Xia, "Optimal k-nearest neighbors and local density-based clustering algorithm for uncertain data," Control and Decision-making, vol. 31, no. 3, pp. 541–546, 2016.

[49] R. Liu, H. Wang, and X. M. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," Information Sciences, vol. 450, pp. 200–226, 2018.

[50] R. Y. Jia and Y. G. Li, "K-means algorithm for self-determining the number of clusters and initial center points," Computer Engineering and Applications, vol. 54, no. 07, pp. 152–158, 2018.

[51] S. X. Tian, L. Q. Ding, and J. Q. Zheng, "K-means text clustering algorithm based on density peak optimization," Computer Engineering and Design, vol. 38, no. 04, pp. 1019–1023, 2017.