Hindawi
Security and Communication Networks
Volume 2023, Article ID 9806317, 1 page
https://doi.org/10.1155/2023/9806317

Retraction

Retracted: Cotraining Algorithm Based on Weighted Principal Component Analysis and Improved Density Peak Clustering

Security and Communication Networks

Received 13 September 2023; Accepted 13 September 2023; Published 14 September 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] T. Wang, "Cotraining Algorithm Based on Weighted Principal Component Analysis and Improved Density Peak Clustering," *Security and Communication Networks*, vol. 2022, Article ID 8353697, 6 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 8353697, 6 pages
https://doi.org/10.1155/2022/8353697

Research Article

Cotraining Algorithm Based on Weighted Principal Component Analysis and Improved Density Peak Clustering

Tao Wang

School of Software, Changsha Social Work College, Changsha, Hunan 410004, China

Correspondence should be addressed to Tao Wang; 201903517@stu.ncwu.edu.cn

Received 9 July 2022; Revised 9 August 2022; Accepted 17 August 2022; Published 5 September 2022

Academic Editor: C. Venkatesan

In order to solve the problem of insufficient useful information of unlabeled samples added in the iterative process and the accumulation of classification errors caused by inconsistent labeling of samples by multiple classifiers, a cotraining algorithm based on weighted principal component analysis and improved density peak clustering is proposed. This paper firstly introduces the density peak clustering algorithm and the density peak clustering algorithm based on weighted voting consistency. In terms of experiments, the DPC-VM algorithm will be tested on the real datasets Seed, Haberman, and Vertebral, and the accuracy performance of the DPC-VM algorithm in clustering will be compared with the DPC algorithm. DPC-VM's dataset seed is 89.99, dataset Haberman is 55.69, and dataset Vertebral is 75.77. The dataset seed of DPC is 88.61, the dataset Haberman is 53.62, and the dataset Vertebral is 56.25. The dataset seed for E-FDPC is 40.38 and the dataset Haberman is 17.42. The dataset seed for K-means is 89.25 and the dataset Haberman is 51.36. The dataset seed for FCM is 89.49 and the dataset Haberman is 50.89. The performance of the DPC-VM algorithm on Acc is basically better than other algorithms.

1. Introduction

In unattended machine learning, integration algorithms are an important part of data exploration. It analyzes data according to the structure of data structures and divides it into subgroups based on characteristics such as "homogeneous groups," so that the properties of similar groups and the properties of different groups have different degrees of similarity [1]. In recent years, with the rise of big data, group algorithms have been widely used in new functions such as modeling, diagnostics, research scientific knowledge, big data processing such as biomedicine, and virtual reality. Currently, the most common applications can be subdivided into clusters, hierarchy, density, structure-based clusters, and network-based diagrams [2].

The K-means algorithm is a classic split-group algorithm. It starts with the first center K and then returns everything to the "closest" group as the mission is completed. The algorithm is reliable, simple, and intuitive, and has good performance on big data. Figure 1 shows the multipurpose internal control technology process based on a speed-based fast group search algorithm [3]. However, the K-identification algorithm provides sample data for nearest groups, so the algorithm can only be used for spherical groups and cannot capture nonspherical groups. Fast-based spatial clustering can detect irregularly shaped clusters [4]. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a rate-based clustering algorithm [5]. Compared to the K-identification algorithm, DBSCAN does not need to know the required group code first. However, the algorithm must first determine the consequences of two rates community radius Eps and minimum community number MinPts. The group's results show that the selection of the measurement area around the Eps radius is well understood, and DBSCAN does not affect the quality of high-dimensional data [6].

2. Literature Review

For this problem, MD Parmar, et al. propose CFSFDP (clustering by fast search and find of density peaks), similar to the DBSCAN and Mean-shift algorithm, it can detect

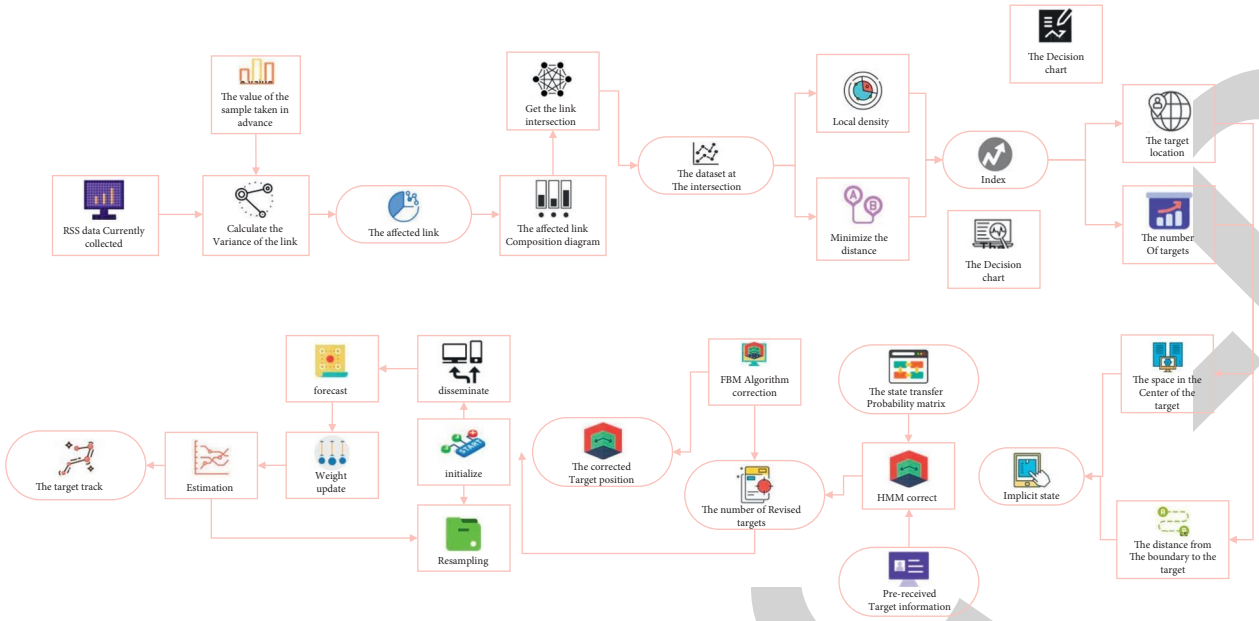

Figure 1: Indoor multitarget tracking technology based on the density-based fast search clustering algorithm.

arbitrary shape clusters without setting the number of clusters in advance [7]. Liang et al. define the local density of samples by combining Euclidean distance and shared nearest neighbor similarity, which reduces the dependence of the algorithm on the parameter cutoff distance [8]. Yang and Nataliani compare the distance parameters of the sample points to make the cluster center points have a higher degree of discrimination in the decision graph. However, when calculating the local density of sample points, the above method only considers the overall distribution of the data set and does not consider the distribution of various clusters, and there is still the problem of subjective selection of parameters [9]. The Fuzzy-CFSFDP algorithm proposed by Silva et al. uses fuzzy rules to determine the cluster center of the CFSFDP algorithm. However, it also lacks the comparison of distance parameters in the generation process of density peak points, resulting in multiple "similar" cluster center points in the decision diagram, which has a great impact on the selection of cluster center [10].

Quintian and Corchado propose a method to determine the cutoff distance and boundary correction through the thermal diffusion of the data set. Although it improves the selection of cluster center points and the accuracy of clustering results, the model is relatively complicated [11]. Geng et al. introduce KNN into local density calculation and designed a new allocation strategy. Although it can improve the quality of clustering, it also introduces new parameter correlation coefficients, increases the complexity of the algorithm, and its setting lacks a corresponding basis [12]. Yan et al. propose that the Euclidean distance between any two sample points needs to be calculated due to the density peak clustering, which will cause a lot of time overhead and reduce the computational efficiency. The grid is introduced to improve it so that it is not necessary to calculate the Euclidean distance between all sample points. At the same time, it reduces the subjective setting of parameters and improves the operation efficiency of density peak clustering. The original density peak clustering method is calculated by CPU [13].

Kulkarni and others use GPU to improve the operating efficiency of the density peak clustering method, which is a parallelized method. The improved clustering method is 45 times that of the original clustering method. Its main idea is to share a memory, so this requires a corresponding conversion of the data structure of the program, to make it applied to the combined access mechanism of the new architecture [14]. Wang and others analyze the operating efficiency of the density peak clustering algorithm and propose to use the local sensitive hash method for distributed computing, which improves the operating efficiency of the density peak clustering method. The improved clustering method is 1.7–70 times higher than the original clustering method. Its main idea is to regulate the parameters, so as to improve the running time of the algorithm. The clustering optimization process of hierarchical partition is shown in Figure 2 [15].

Wang et al. apply the density peak clustering to the image field, which is aimed at the scene and the face [16]. Dafu et al. use density peak clustering in the field of network communities. They have different clustering purposes, so they are improved for specific problems [17]. Based on the current research, this paper firstly introduces the density peak clustering algorithm, and density peak clustering based on weighted voting consistency. For experiments, the DPC-VM algorithm will be tested on the real data sets Seed, Haberman, and Vertebral, and the accuracy performance of the DPC-VM algorithm in clustering will be compared with the DPC algorithm. The performance of the DPC-VM algorithm on Acc is basically better than other algorithms. Even for the data set Wine, the Acc using the DPC-VM

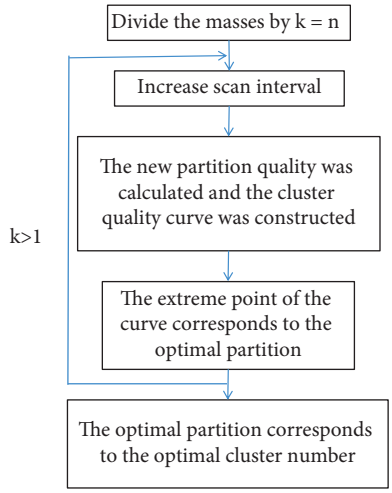

Figure 2: Density optimized clustering algorithm based on hierarchical partition.

algorithm test is no lower than the DPC algorithm. Seed, Haberman, and Vertebral are all data sets with intersections and overlaps of edge-like points, which shows that the DPC-VM algorithm performs better on data sets with intersections and overlaps in point distribution.

3. Methods d_c

3.1. Density Peak Clustering Algorithm. The DPC algorithm determines the characteristics of the center point: the higher the local size, the further away from the center point [18]. In this section, algorithms are used to determine the diagram based on the values of the main attributes of the complex sites. In order to interpret the image, two things should be considered for each parameter data: the speed of a point and the distance from a point to a certain point. There are two ways to speed up local content: a truncated kernel function and a Gaussian kernel function. Assuming that there is a data set $X = \{x_1, \ldots, x_i, \ldots x_n\}$, for each data point, its local density ρ_i can be expressed as follows:

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

where $x < 0$ then $\chi(x) = 1$, otherwise $\chi(x) = 0$. Euclidean distance between data points x_i and x_j. d_c stands for space.

$$\begin{aligned} d_c &= d_{\lceil N \times q \rceil} \in D \\ &= \{d_1, d_2, \ldots, d_N\}, \end{aligned} \tag{2}$$

where D is the distance between two points, and the distance in D is in ascending order. N is the value of the data contained in D. q is the percentage modified manually. $\lceil . \rceil$ means rounding up.

Equation (1) uses the kernel truncation function to calculate the local distribution of the data content. When using the Gaussian kernel, the density of the local content is reported accordingly

$$\rho_i = \sum_j \exp\left(-\frac{d_{ij}^2}{d_c^2}\right). \tag{3}$$

For a data point x_i, δ_i is the minimum distance from x_i to any point x_i with a higher density, which is defined as follows:

$$\delta_i = \min_{j:\, \rho_j > \rho_i} \left(d_{ij}\right). \tag{4}$$

For the point x_k with the highest density, δ_k is expressed as follows:

$$\delta_k = \max_j \left(d_{kj}\right). \tag{5}$$

After calculating the velocities and distances of each point, they can be set up according to a diagram determined with abscissa ρ and ordinate δ. The characteristics of the joints and other points can be seen at the end of the figure. Figure 3 shows the main points of the R15 configuration file, and Figure 4 shows the main points of the R15 section after grouping with the DPC algorithm. The high points δ and high ρ are the mean, and the high points δ and the low ρ points can be considered as a group of different points, that is, the difference [19]. After finding the place, each part is grouped by the density of its neighbors. Teamwork is just one step and does not need to be repeated.

3.2. Density Peak Clustering Based on Weighted Voting Consistency. In the DPC algorithm, each point belongs to the same class as its nearest neighbor, and the distribution of the target point is determined by its occupancy. When the distribution of content in the configuration file is discontinuous and overlapping, it is not necessary to allocate the closest objects with the DPC algorithm, which requires a lot of close content to determine the result of the target point [20]. The algorithms described in this paper are based on the K-En Nearest Neighbor (KNN) concept. The main idea of the KNN algorithm is that if the location feature includes most of the proximal models of the model, then the model also belongs to this class. Because KNN's classification criteria are based on environmental standards rather than individual classes, models need to be divided into more classes or overlapped. The simplest approach to the KNN algorithm is to find the k closest to the sample points of the training configuration based on distance measurements to a given example. Then make predictions based on data from nearest neighbors. In general, the "voting method" can be used for the allocation of labor, that is, the group of symbols that appear in most of the k examples is selected according to the hypothesis. Based on the concept of the KNN algorithm, this paper proposes a new group DPC-VM19 algorithm based on the DPC algorithm. In other words, when segmenting content, refer to information from multiple people around you rather than the environment alone [21].

Assuming that there is a data set $X = \{x_1, \ldots, x_i, \ldots x_n\}$, the neighbor points of each point x_i are screened to see whether the three conditions required for voting are met: first, the density of neighbor points is greater than this point.

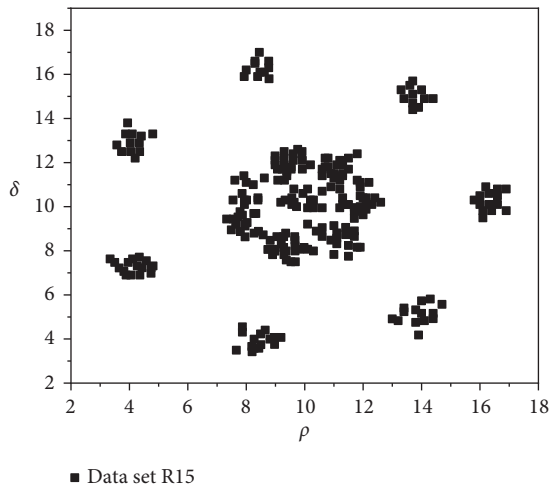

Figure 3: The original point distribution of the data set R15.

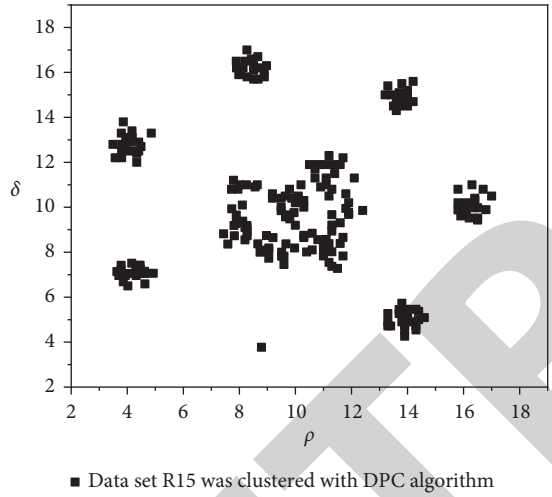

Figure 4: Point distribution diagram of data set R15 after clustering with DPC algorithm.

Second, the distance to the point is the closest. Third, the distance to this point is less than the cutoff distance (the cutoff distance is n times the cutoff distance d_c in the second part). The points that meet these conditions are the points around the target point that have a greater impact on the classification of points [22].

For the neighbor point x_j that meets the conditions around the point x_i, it is stipulated to select at most k_m neighbor points from them. Then, randomly select k_i points from the k_m neighbor point to vote, and get a label, which indicates the category to which the neighbor point belongs. Repeat the previous step once, each time the points and number of voting are different, and t tags can be obtained and used as a total tag set. Finally, vote again on the total tag set, and the resulting tag is used as the final classification of the point. The pseudocode for the implementation of the DPC-VM algorithm is shown below.

Require: Data set $X = \{x_1, \ldots, x_i, \ldots x_n\}$, cutoff distance d_c, number of neighbor points k_m , and number of repeated votes t.

For $i = 1$: n do
For $j = 1$: n do
Calculate the Euclidean distance between the i-th point and the j-th point.
End for
For $i = 1$: n do
Calculate the density of the i-th point.
End for
For $i = 1$: n do
If the i-th point is not the point with the highest density, then, calculate the distance δ_i from the i-th point to its nearest neighbor point with a higher density;
End if
End for
For the point with the highest density, δ_k is the maximum distance. Make a decision diagram and select the point with high δ and high ρ as the cluster center.
For $i = 1$: n do
Calculate the set of eligible neighbor points of the i-th point.
End for
For $i = 1$: n do
If the i-th point has a qualified neighbor point, then.
For $j = 1$: n do
Randomly select any number of eligible neighbor points to vote, and record the most frequent label category.
End for
Points are assigned to the tag category that appears most frequently.
Else
Points are assigned to clusters of nearest neighbor points with higher density.
End if
End for
Ensure: clustering result

4. Results and Analysis

In this part, the DPC-VM algorithm will be tested on the real data set Seed, Haberman, and Vertebral, and the accuracy performance of the DPC-VM algorithm on clustering will be compared with the DPC algorithm [23]. In this paper, accuracy is measured by the Acc indicator:

$$\mathrm{Acc} = \frac{\sum_{i=1}^{k} a_i}{n}, \tag{6}$$

where a_i is the number of correctly classified points in the i-th cluster. k is the number of clusters. n is the number of points in the data set.

For Acc, a higher value means better clustering quality. When their value is 1, it means that the clustering result is completely correct. Figures 5–7, respectively, show the decision diagrams of the DPC-VM algorithm on these three commonly used data sets. At the same time, the accuracy performance of other clustering algorithms, the k-Means algorithm and fuzzy C-means algorithm on these real data sets are queried [24]. The accuracy comparison of the DPC-VM algorithm, DPC algorithm, and these three clustering algorithms is shown in Table 1.

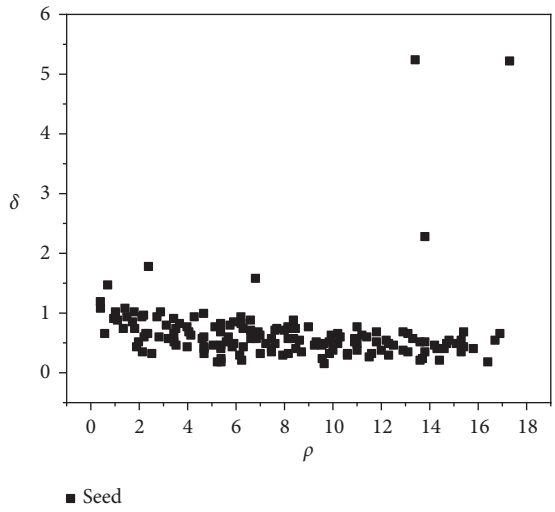

Figure 5: Decision diagram of the data set Seed.

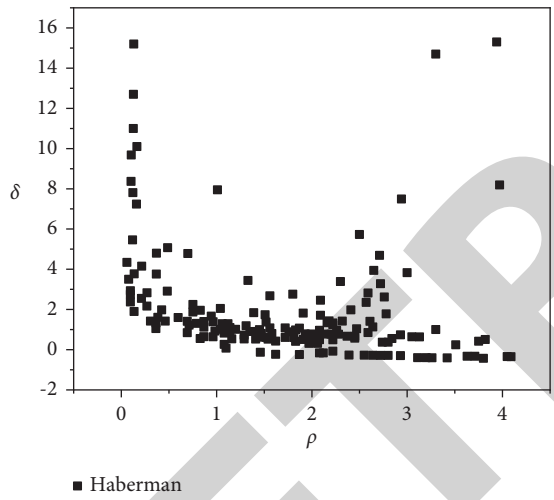

Figure 6: Decision diagram of the data set Haberman.

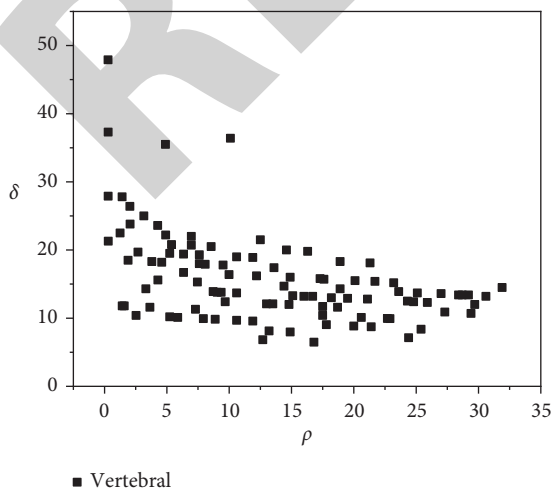

Figure 7: Decision diagram of the data set Vertebral.

Table 1: Results of the data set.

Algorithm	Seed	Haberman	Vertebral
DPC-VM	89.99	55.69	75.77
DPC	88.61	53.62	56.25
E-FDPC	40.38	17.42	—
K-means	89.25	51.36	—
FCM	89.49	50.89	—

In Table 1, by adjusting the total number of decision votes t, the accuracy of the test data set Seed, Haberman, Vertebral, Ecoli, Iris, and Wine can be stabilized by the DPC-VM algorithm [25]. The accuracy of the DPC-VM algorithm shown in Table 1 tested on different data sets is not necessarily the best performing value, but they are all relatively stable and significantly improved values (values with the most occurrences). In the table, the accuracy of the DPC-VM algorithm is improved differently compared with other algorithms on different data sets. Taking the DPC algorithm as the main comparison object, the Acc improvement of the DPC-VM algorithm on the Vertebral data set is more obvious. This is because the Vertebral data set has a lot of intersections between categories, and the DPC-VM algorithm has improved the classification problem of this part [26]. For the data set Wine, the DPC-VM algorithm can hardly improve the accuracy. This is because the Wine is a high-dimensional sparse data set. Because the points are very sparse, the neighbor points that meet the conditions are almost empty sets, which are equivalent to no reallocation, but are assigned to the nearest neighbor class according to the original algorithm.

5. Conclusion

The performance of the DPC-VM algorithm on Acc is basically better than other algorithms. Even for the data set Wine, the Acc when using the DPC-VM algorithm test is no lower than the DPC algorithm. Seed, Haberman, and Vertebral are all data sets with intersections and overlaps of edge-like points, which show that the DPC-VM algorithm performs better on data sets with intersections and overlaps in point distribution. The processing of high-dimensional data sets can be further studied in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Silva, F. Araujo, L. Santos, R. Veras, and F. Medeiros, "Optic disc detection in retinal images using algorithms committee with weighted voting," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2446–2454, 2016.

[2] J. Hou, A. Zhang, and N. Qi, "Density peak clustering based on relative density relationship," *Pattern Recognition*, vol. 108, no. 8, Article ID 107554, 2020.

[3] C. M. Own, Z. Meng, and K. Liu, "Handling neighbor discovery and rendezvous consistency with weighted quorum-based approach," *Sensors*, vol. 15, no. 9, Article ID 22377, 2015.

[4] G. Mawloud and M. Djamel, "Weighted sparse representation for human ear recognition based on local descriptor," *Journal of Electronic Imaging*, vol. 25, no. 1, Article ID 13036, 2016.

[5] S. Martin, J. L. Lerma, and H. Uzkeda, "Heuristic method based on voting for extrinsic orientation through image epipolarization," *Journal of Electronic Imaging*, vol. 26, no. 06, p. 1, 2017.

[6] L. Yaohui, M. Zhengming, and Y. Fang, "Adaptive density peak clustering based on k-nearest neighbors with aggregating strategy," *Knowledge-Based Systems*, vol. 133, no. 1, pp. 208–220, 2017.

[7] M. D. Parmar, W. Pang, D. Hao, J. Jiang, and Y. Zhou, "Fredpc: A Feasible Residual Error-Based Density Peak Clustering Algorithm with the Fragment Merging Strategy," *IEEE Access*, vol. 7, no. 99, p. 1, 2019.

[8] W. Liang, K. C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2020.

[9] M. S. Yang and Y. Nataliani, "A feature-reduction fuzzy clustering algorithm based on feature-weighted entropy," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 817–835, 2018.

[10] R. R. V. Silva, F. H. D. D. Araujo, L. M. R. dos Santos, R. M. S. Veras, and F. N. S. D. Medeiros, "Optic disc detection in retinal images using algorithms committee with weighted voting," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2446–2454, 2016.

[11] H. Quintian and E. Corchado, "A Novel Ensemble Beta-Scale Invariant Map Algorithm," *IEEE Access*, vol. 8, Article ID 108884, 2020.

[12] R. Geng, I. Bose, and X. Chen, "Prediction of financial distress: an empirical study of listed Chinese companies using data mining," *European Journal of Operational Research*, vol. 241, no. 1, pp. 236–247, 2015.

[13] Y. T. Yan, Y. P. Zhang, J. Chen, and Y. W. Zhang, "Incomplete data classification with voting based extreme learning machine," *Neurocomputing*, vol. 193, no. Jun.12, pp. 167–175, 2016.

[14] V. Y. Kulkarni, P. K. Sinha, and M. C. Petare, "Weighted hybrid decision tree model for random forest classifier," *Journal of the Institution of Engineers: Serie Bibliographique*, vol. 97, no. 2, pp. 209–217, 2016.

[15] Z. M. Wang, G. H. Song, and C. Gao, "An Isolation-Based Distributed Outlier Detection Framework Using Nearest Neighbor Ensembles for Wireless Sensor Networks," *IEEE Access*, vol. 7, no. 99, Article ID 96333, 2019.

[16] T. Wang, X. Guan, X. Wan, H. Shen, and X. Zhu, "A Spectrum-Aware Clustering Algorithm Based on Weighted Clustering Metric in Cognitive Radio Sensor Networks," *IEEE Access*, vol. 7, no. 99, Article ID 109565, 2019.

[17] S. Dafu, Z. Leihong, L. Dong, L. Bei, and K. Yi, "Recovery of a spectrum based on a compressive-sensing algorithm with weighted principal component analysis," *Laser Physics*, vol. 27, no. 7, Article ID 075201, 2017.

[18] Y. Yang, D. Chen, and W. Hui, "Active sample selection based incremental algorithm for attribute reduction with rough sets," *IEEE Transactions on Fuzzy Systems*, vol. 25, pp. 825–838, 2016.

[19] N. Guo, Y. Fang, Z. Tian, and S. Cao, "Research on soc fuzzy weighted algorithm based on ga-bp neural network and ampere integral method," *Journal of Engineering*, no. 15, pp. 576–580, 2019.

[20] T. Xue, T. T. Li, and B. Sun, "Research on parallelization of knn locally weighted linear regression algorithm based on mapreduce," *Journal of communications*, vol. 10, no. 11, pp. 864–869, 2015.

[21] T. Feng, L. Yang, X. Zhao, H. Zhang, and J. Qiang, "Online identification of lithium-ion battery parameters based on an improved equivalent-circuit model and its implementation on battery state-of-power prediction," *Journal of Power Sources*, vol. 281, no. 1, pp. 192–203, 2015.

[22] X. Zhang, K. P. Rane, I. Kakaravada, and M. Shabaz, "Research on vibration monitoring and fault diagnosis of rotating machinery based on internet of things technology," *Nonlinear Engineering*, vol. 10, no. 1, pp. 245–254, 2021.

[23] J. Chen, J. Liu, X. Liu, X. Xu, and F. Zhong, "Decomposition of toluene with a combined plasma photolysis (CPP) reactor: influence of UV irradiation and byproduct analysis," *Plasma Chemistry and Plasma Processing*, vol. 41, no. 1, pp. 409–420, 2021.

[24] K. Sharma and B. K. Chaurasia, "Trust Based Location Finding Mechanism in VANET Using DST," in *Proceedings of the Fifth International Conference On Communication Systems & Network Technologies*, pp. 763–766, Gwalior, India, April 2015.

[25] P. Ajay, B. Nagaraj, R. A. Kumar, R. Huang, and P. Ananthi, "Unsupervised hyperspectral microscopic image segmentation using deep embedded clustering algorithm," *Scanning*, vol. 2022, Article ID 1200860, 9 pages, 2022.

[26] G. Veselov, A. Tselykh, A. Sharma, and R. Huang, "Special issue on applications of artificial intelligence in evolution of smart cities and societies," *Informatica*, vol. 45, no. 5, p. 603, 2016, http://www.informatica.si/index.php/informatica/article/view/3600.