Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 1083961, 9 pages
https://doi.org/10.1155/2022/1083961

*Research Article*

# An RBF Neural Network Clustering Algorithm Based on K-Nearest Neighbor

**Jitao Li,[1] Chugui Xu,[1] Yongquan Liang,[2] Gengkun Wu,[2] and Zhao Liang[1]**

[1]*Department of Information Science and Technology, Taishan University, Taian 271000, China*
[2]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*

Correspondence should be addressed to Jitao Li; kisstr@126.com

Received 1 July 2022; Accepted 1 August 2022; Published 24 August 2022

Academic Editor: Wei Liu

Neural network is a supervised classification algorithm which can deal with high complexity and nonlinear data analysis. Supervised algorithm needs some known labels in the training process, and then corrects parameters through backpropagation method. However, due to the lack of marked labels, existing literature mostly uses Auto-Encoder to reduce the dimension of data when facing of clustering problems. This paper proposes an RBF (Radial Basis Function) neural network clustering algorithm based on K-nearest neighbors theory, which first uses K-means algorithm for preclassification, and then constructs self-supervised labels based on K-nearest neighbors theory for backpropagation. The algorithm in this paper belongs to a self-supervised neural network clustering algorithm, and it also makes the neural network truly have the ability of self-decision-making and self-optimization. From the experimental results of the artificial data sets and the UCI data sets, it can be proved that the proposed algorithm has excellent adaptability and robustness.

## 1. Introduction

Cluster analysis is an important method of data mining, whose core idea is to gather high similarity points in the data set into clusters while ensuring that different clusters have significant differences. Clustering can explore hidden patterns and rules in data, which is the embodiment of the decision-making ability of artificial intelligence algorithms, and is now widely used in computer science, information security, and image processing.

In recent years, the sudden development of neural network has shown people its powerful function, which can deal well with the processing of high-dimensional and complex data, and it has achieved successful applications in the fields of image clustering [1, 2], facial recognition [3–5], image segmentation [6, 7], and so on. However, the shortcoming of neural network is also very obvious, the limitation of manual label annotation restrict its self-decision ability. Existing unsupervised learning methods include clustering and dimension reduction, and clustering algorithms are complex and diverse, which can be divided into clustering methods based on prototype-based, density, hierarchy, and dimension reduction includes Auto-encoders and PCA (Principal Component Analysis) method, whose major for data preprocessing.

From the existing literature, it can be seen that the more common idea is to migrate the loss function of the traditional clustering algorithm to the neural network structure and achieve clustering through global optimization. Yang et al. [8] propose the DCN model, which migrates the loss function of K-means to the feature space of the Auto-encoder, and realize feature learning and clustering through the alternating optimization of network parameters and cluster centers. Yang et al. [9] combine hierarchical clustering algorithm with CNN (Convolutional Neural Networks) and realize feature clustering through global optimization of cluster merging and feature learning. SpectralNet [10] introduces the idea of spectral clustering into deep learning, which first learns the similarity matrix between features through a Siamese Network, then obtain a new feature measure based on the spectral clustering objective function, and finally performs K-means in the feature space to obtain cluster assignments. VaDE (Variational Deep Embedding) [11] introduces an idea of migrate clustering of GMM (Gaussian Mixture Models) into the Variational Auto-encoder (VAE), which realizes the optimization of feature learning and cluster allocation through the distribution constraints of feature space. Another idea is to directly design a specific cluster loss function based on the desired clustering assumptions [12]. DAC (Deep-adaptive Image Clustering) [13] converts multiclassification problems into binary classification problems instead, continuously generates positive and negative sample pairs with high confidence based on the idea of self-paced learning, which are used as supervised information to guide the training of the model, and finally outputs the clustering result. IMSAT (Information Maximizing Self-Augmented Training) [14] and IIC (Invariant Information Clustering) [15] are based on the same assumption that simple transformations of data do not alter their intrinsic semantic information, and clustering is achieved by maximizing the information entropy of the original sample and its enhanced sample.

In terms of the combination of traditional clustering algorithms and Auto-encoders, Ren et al. [15] propose a deep density clustering framework by combining density clustering with Auto-encoder, which uses the t-SNE (t-distributed stochastic neighbor embedding algorithm) [16] and DPC (density peaking algorithm) [17], which completes the training by alternating or optimization of cluster pseudolabels and feature representations. Mrabah et al. [18] propose a model training method. In the pretraining phase, reliable feature representations are learned in a self-supervised manner by introducing data augmentation and adversarial interpolation techniques [19]. Due to the high dimension of the image, the combination of dimension reduction and traditional clustering algorithms can indeed effectively improve the accuracy of the algorithm, but it does not really use the neural network to classify the data, and it does not make the neural network have the ability of self-decisions.

Unsupervised clustering methods based on deep learning use VAE (Variational Auto-encoder) and GANs (Generative Adversarial Networks) [20, 21], which use existing data to generate data that does not exist in reality, mainly for image generation and sharpening. VaDE is a generative clustering model based on VAE; this algorithm models the process of data generation by introducing a Gaussian hybrid model. GMVAE [22] adopts a strategy similar to VaDE, which imposes a Gaussian mixture distribution constraint on the feature space. By minimizing the information constraints, the model avoids falling into a local solution at the beginning of training. Mukherjee et al. [23] propose a clustering method-based generate adversarial networks. The algorithm utilizes mixed discrete and continuous latent variables to construct new spaces for clustering by interpolating methods.

The existing cluster analysis has less robustness and the self-decision classification ability of neural networks is low. In order to make full use of the advantages of neural network, this paper proposes an RBF neural network clustering algorithm based on KNN graph (RBF-KNN). The core idea of the algorithm in this paper is to use the K-means algorithm to generate the initial pseudolabels, by using the overall and local information retention ability of the RBF neural network to train the classification network under the pseudolabel, and then use the self-supervision method based on the neighbor to continuously generate the corrected class label, optimize the generated neural network and obtain the clustering results, and finally achieve the purpose of self-optimization and self-decision of the neural network results. The algorithm process in this paper is simple, and while have fewer parameters, it can effectively handle irregular data sets and unbalanced data sets. From the experimental results of the artificial data set and the UCI data set, it can be seen that the proposed algorithm has good robustness and adaptability.

## 2. Related Works

*2.1. RBF Neural Network.* In 1985, Powell proposed a Radial Basis Function (RBF) method for multivariate interpolation, which uses a Gaussian kernel function in most cases, and the RBF neural network is a typical three-layer neural network that includes an input layer, a hidden layer, and an output layer. The transformation from input space to hidden space is nonlinear, while the transformation from hidden space to output layer space is linear. The network structure is as Figure 1.

The hidden nodes of BP (Back Propagation) neural network use the input pattern and the inner product of the weight vector as the arguments of the activation function, while the activation function uses the "Sigmoid" function. The parameters have an equally important effect on the output of the BP network, so the BP neural network is a global approximation of the nonlinear map.

Compared to traditional BP neural network, the hidden nodes of RBF neural network use the similarity between the input mode and the central vector (such as Euclidean distance) as the argument of the function, and the radial basis function as the activation function. Farther the input of a neuron is from the center of the radial basis function, the less activated the neuron becomes (Gaussian function). RBF neural network thus carries more local information and have a "local mapping" feature.

*2.2. K-Nearest Neighbor and K-Means Algorithms.* K-nearest neighbor algorithm is one of the most commonly used algorithms in supervised classification algorithms, which selects $K$ closest sample points to obtain the corresponding class labels through the election method, which have low complexity and high accuracy. The KNN graph is an undirected graph formed by connecting the sample to the adjacent $K$ sample based on the K-neighbor principle.

K-means algorithm is the most commonly used traditional clustering algorithm, which is based on the greedy principle, and selects the global optimal through iterative mode, and its optimization function is

$$y = \min \sum_{i=1}^{N} \operatorname{dist}(x_i, c_i). \tag{1}$$

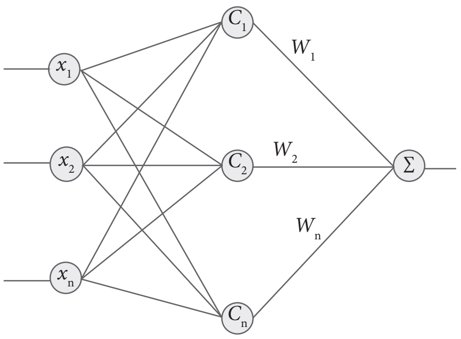


Figure 1: Structure of RBF neural network.

The K-means method is simple and practical which can meet most needs, but its problem of sensitivity to noise data and to spherical data has always been a problem that scholars solving, and the existing DPC is a density-based clustering method, but it needs to find a central point, which can also be seen as an improvement of the K-means algorithm.

## 3. Introduction of Algorithms in This Paper

In this paper, the algorithms need to perform two times K-means clustering and multiple iteration of RBF neural network backpropagation.

*3.1. Generate Pre-Trained Pseudo-Labels.* Since the training of neural network requires the assistance of class labels, the proposed algorithm first uses the K-means algorithm to cluster the datasets (random labels can also be used, but more iterations are required thus), and the resulting pseudolabels are transformed, as shown in Figure 2.

The resulting pseudolabels are used for the next step of neural network training.

*3.2. Full RBF Neural Network Training.* The traditional RBF neural network is a locally weighted network, which selects part of the sample set as the center point for the training of the neural network, and its training process is to use RBF as the "base" of the hidden and the input vector is directly mapped to the hidden space. The RBF neural network establishes a mapping relationship around the center point, and the mapping from the implicit layer space to the output space is linear, which means the output of the network is the linear weighted sum of the output of the hidden unit. Among them, the role of the hidden layer is to map the vector from low dimensions to high dimensions through the kernel function, and the situation of linear indivisibility of low dimensions can be mapped to high dimensions to become linearly separable.

The outputs of the RBF network are as follows:

$$y = \sum_{i=1}^{n} W_i h_i (x), \tag{2}$$

where $y$ is the output of the RBF neural network, $n$ is the number of neurons in the hidden layer, $W_{\mathrm{i}}$ is the connection weight between the $i$ neuron of the hidden layer neuron and the neuron in the output layer, $h_i(x)$ is the activation function of the neurons of the hidden layer, the activation function usually takes the Gaussian function which is defined as follows:

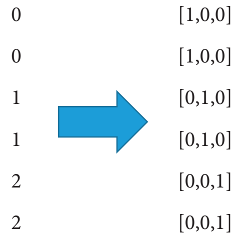


Figure 2: Transformation of pseudolabels.

$$h_i (x) = \exp\left( -\frac{1}{2\sigma_i^2} \|\zeta - \mu_i\|^2 \right). \tag{3}$$

$\zeta = (a_1, a_2, \ldots, a_m)^T$ is the input matrix, $\mu_i$ is the selected center point, $\sigma_i$ is the width of the neuron, and $\|\zeta - \mu_i\|$ is the Euclidean distance between the input matrix and the radial base center.

Clustering can be seen as a process of optimization of neural network output and weights, the output expression of the radial base network used in this algorithm is as follows, and its objective function can be seen as the process of minimization of the following formula:

$$y = \min \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \exp\left( -\frac{1}{2\sigma^2} \left\| x_i - x_j \right\| \right). \tag{4}$$

In this paper, the algorithm no longer selects the center point but retains all the samples and uses Gaussian kernel functions to map to high-dimensional space, so that the RBF neural network can retain the information between the data to the greatest extent. Using the pseudolabel to train RBF neural network, the algorithm selects the gradient descent algorithm as the backpropagation method, and finally obtains the appropriate network weights after several iterations of training.

*3.3. Generation of Self-Supervised Remediation Labels.* Due to the limitations of traditional partition-based clustering algorithms, the class labels obtained by preprocessing are not necessarily correct when facing nonspherical clusters and unbalanced datasets, so the network parameters need to be corrected.

According to the K-nearest neighbor principle, from the microscopic point of view, the data sample must have the same class label as the adjacent data point, so the resulting corrected label is the mean of the neighbor sample class label, the specific formula is as follows:

$$\text{label}_{\text{correction}} = \frac{1}{K} \sum_{n=1}^{K} \text{label}_{n-\text{nearest}}. \tag{5}$$

The algorithm loss function in this paper uses the MSE (Mean Square Error), and the backpropagation algorithm also uses the gradient descent algorithm, after multiple rounds of iteration, the new measure labels are finally obtained.

*3.4. Clustering Generating Class Labels.* After re-entering the data set $X$, the new measure labels are obtained, and the K-means clustering algorithm is used again for the resulting output to obtain the final results.

The pseudocode of algorithm in this paper is shown in Table 1.

Table 1: Algorithm process description.

```
Algorithm1 RBF-KNN clustering
Input: Dataset $X$, $k$, K-nearest neighbor-value
Output: Cluster labels $c_i$ of $X$
Run K-means to pre-divide $X$ int $k$ clusters
Transform pre-divide labels to pseudo-labels
Repeat
    Training RBF network by minimizing loss
Repeat
    Get correction labels by KNN graph
    Update network parameters by minimizing loss
Data_mark = RBF_KNN($X$)
Run K-means to divide Data_mark into $k$ clusters
```

## 4. Experimental Results and Analysis

*4.1. Artificial Data Set Verification.* Artificial data set is a human-made set data set with obvious characteristics, which can be easily artificially judged through experience. However, there is a high degree of complexity in artificial data and in terms of data set shape, density, imbalance, or other aspects problems for artificial intelligence algorithms, so it can well test the adaptability of a certain algorithm to a certain class or several types of complex properties.

The manual data sets used in this paper are all two-dimensional data (Table 2), and the used two-dimensional data sets can display more intuitively the quality of the algorithm clustering results and objectively evaluate the performance of the algorithm. The six datasets selected in this paper are typical artificial datasets, and there are problems of density, shape, or complex properties between clusters.

Table 2: Attribute description of each data set.

| ADS | $n$ | $d$ | $k$ |
|---|---|---|---|
| Aggregate | 788 | 2 | 7 |
| Jain | 373 | 2 | 2 |
| Flame | 1000 | 2 | 2 |
| Spiral | 1000 | 2 | 2 |
| Target | 770 | 2 | 6 |
| *UCI* | | | |
| Ecoli | 336 | 7 | 8 |
| Iris | 150 | 4 | 3 |
| Seeds | 210 | 7 | 3 |
| Soybean | 47 | 35 | 4 |
| Segment | 2310 | 11 | 7 |

The algorithm in this article first uses the K-means algorithm to do the preprocessing work, and the K-means run the result such as in the left side (Figures 3–8), and the result of the algorithm RBF-KNN in this article is as follows as the right side (Figures 3–8).

The proposed algorithm is a neural network self-supervised clustering method based on partition-based clustering as the basis for preprocessing, so the 6 artificial datasets (ADS) selected in this paper are all datasets that cannot be well processed by the K-means algorithm. It can be seen from the experimental results of RBF-KNN that the processing of the “aggregate,” “long,” “spiral”, and “target” datasets in this paper can achieve an accuracy rate of 100%, and there are still deficiencies in the details of the processing results of the “jain” and “flame” datasets, but the results have been greatly improved compared to the preprocessing results. Although there are still deficiencies in some data sets, it is limited by the functional limitations of neural networks.

The algorithm structure of the RBF-KNN algorithm is simple and has fewer parameters, that is only two parameters (the value of K-nearest neighbor-value is basically 2 or 3), and compared with the traditional clustering algorithm based on division, the results of this algorithm can basically meet the needs of cluster robustness.

*4.2. UCI Data and Validation, and Evaluation Indicators.* In order to verify that the proposed algorithm can achieve a good clustering effect when dealing with practical problems, several UCI datasets are selected to verify the performance of the proposed algorithm (Table 2), all of which are derived from the UCI machine learning library.

UCI data sets come from different types of industries. “Ecoli” is a data set on molecular research and cell biology for the determination and prediction of yeast data. “Iris” is a data set of classification information about iris plants, which is used to distinguish between three types of plants. “Seeds” data set is mainly a coefficient obtained by X-ray technology for three wheat varieties, and a test data set for wheat classification. The “Soybean” data set is Michalski’s famous soybean disease database for predicting the diseases yielded in soybeans, while the “Segment” data set is a data set that classifies image data on higher numerical attributes, which can better test the performance of this algorithm.

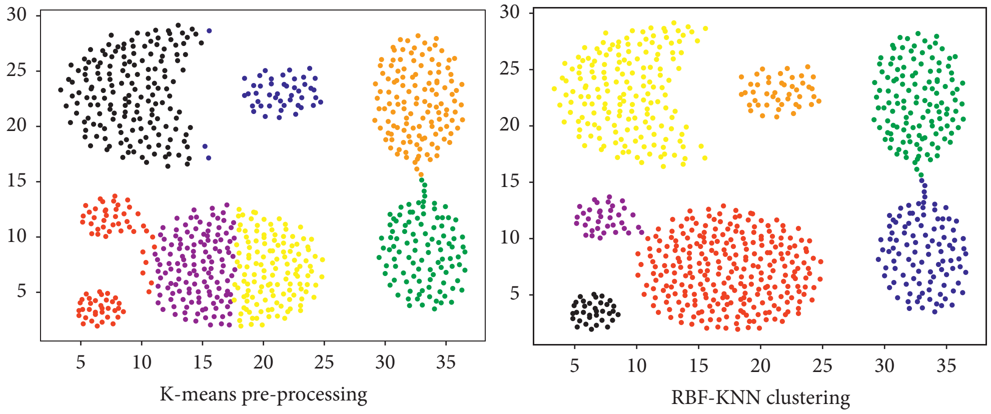
The evaluation criteria for the algorithms selected in this paper include the V-measure coefficient, the adjusted rand index (ARI), and the normalized mutual information (NMI), among which ARI requires the use of the Rand index (RI). Because the use of a single evaluation index will lead to the evaluation results being too one-sided, this paper selects multiple evaluation indicators for cross-validation as formulas (6)–(8).

$$v - \text{mesure} = \frac{2 * (h * c)}{h + c}, \tag{6}$$

$$RI = \frac{a + b}{C_2^{nsamples}}, \tag{7}$$

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]}, \tag{8}$$

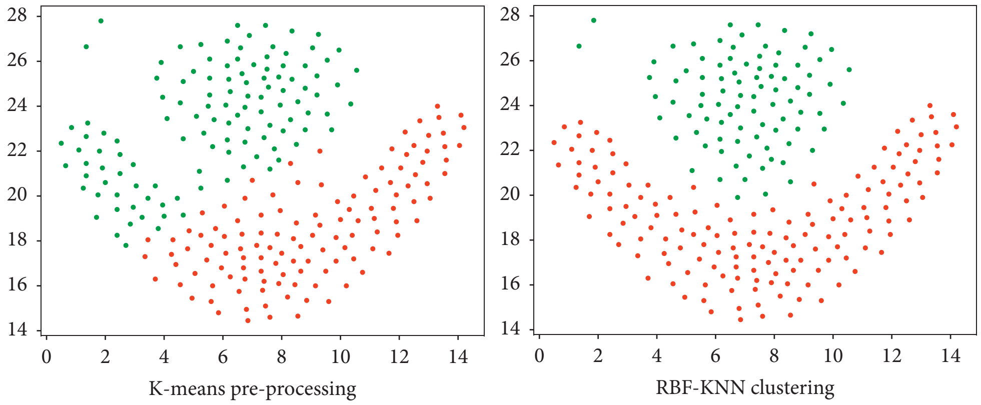
where $h$ denotes homogeneity; $c$ indicates completeness; $a$ and $b$ are the selected categories; $C^2 n$ is the probability of picking 2 classes from all $n$ classes. As an adjustment function of RI, ARI is adjusted using the resulting RI.



K-means pre-processing

RBF-KNN clustering

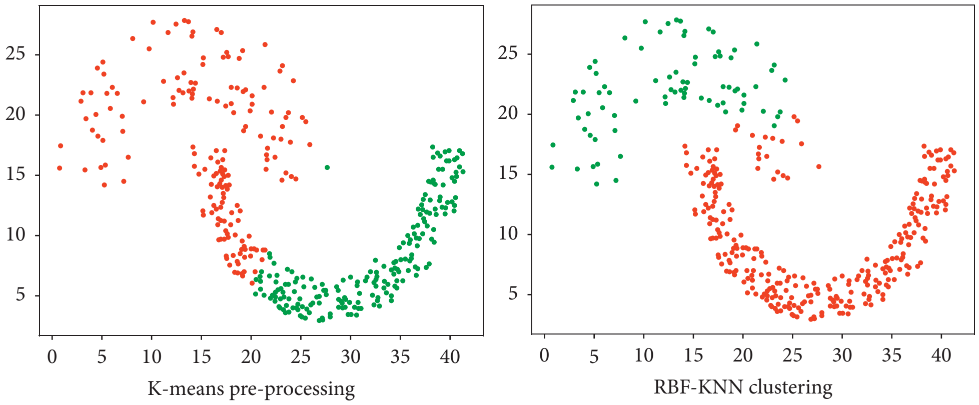
Figure 3: Aggregate (K-nearest neighbor-value = 5).



K-means pre-processing

RBF-KNN clustering

Figure 4: Flame (K-nearest neighbor-value = 3).



K-means pre-processing

RBF-KNN clustering

Figure 5: Jain (K-nearest neighbor-value = 2).

V-measure represents the harmonized mean of homogeneity and integrity, the value range is [0, 1], and the larger the value, the better the clustering effect. The ARI indicates the degree to which the resulting category information matches the expected category, and the ARI range is [−1, 1], and the larger the value, the higher the coincidence of the clustering result with the real situation.

NMI is defined as the formulas (9) and (10).

$$NMI = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}}, \tag{9}$$

$$I(X,Y) = \sum_{h=1}^{k(a)} \sum_{l=1}^{k(b)} n_{h,l} \log\left(\frac{n \cdot n_{h,l}}{n_h^{(a)} n_l^{(b)}}\right). \tag{10}$$

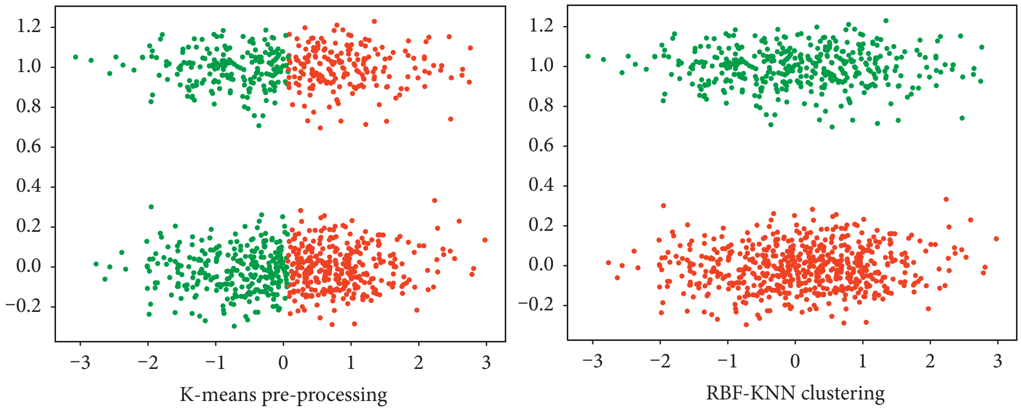


Figure 6: Long (K-nearest neighbor-value = 2).

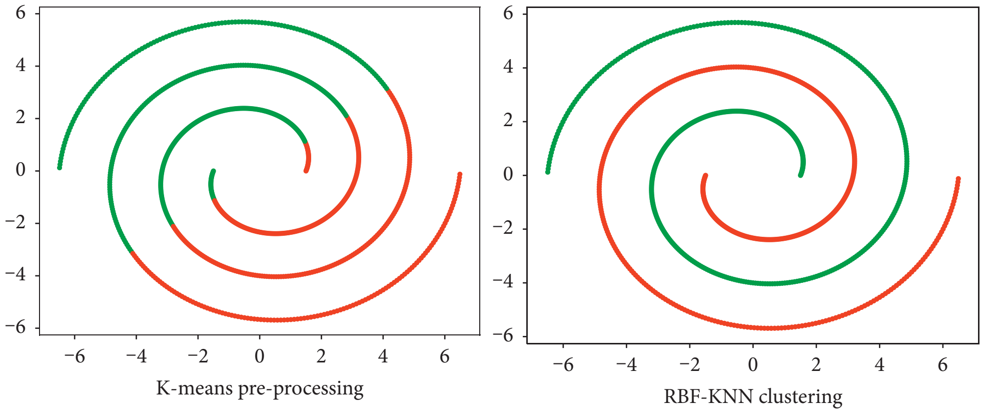


Figure 7: Sprial (K-nearest neighbor-value = 2).

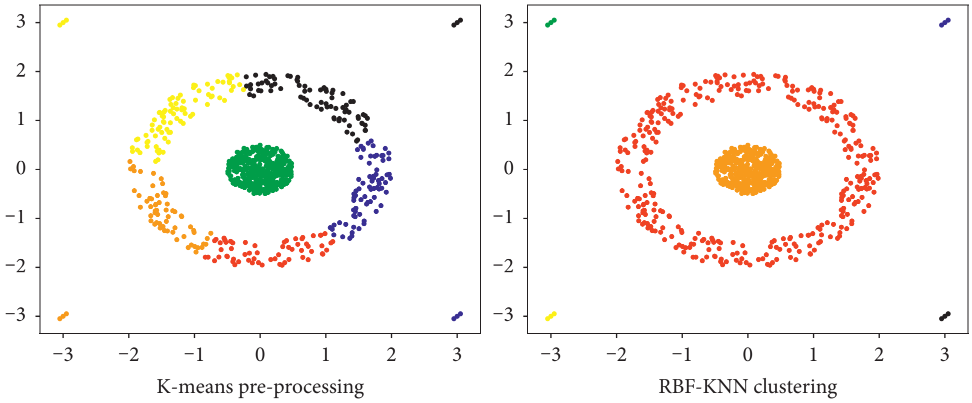


Figure 8: Target (K-nearest neighbor-value = 2).

$X$ and $Y$ represent variables, $I$ $(X, Y)$ represents the mutual information of two variables, $H(X)$ and $H(Y)$ represent entropy for the sum of variables.

NMI is a measure of the interdependencies between variables, indicating the strength of the relationship between two variables. The value range of the NMI index is [0, 1], and the closer the value is to 1, the better the clustering effect. On the contrary, the clustering effect is considered to be poor. Through the above evaluation methods, the processing effect of various algorithms on the data set can be compared more intuitively.

This section selects several types of unsupervised learning methods to compare with the proposed algorithm, including the K-means algorithm based on partitioning, the DPC density clustering algorithm based on K-nearest neighbor, the VaDE clustering algorithm based on VAE and

Table 3: Comparison of evaluation indicators V-measure of different algorithms.

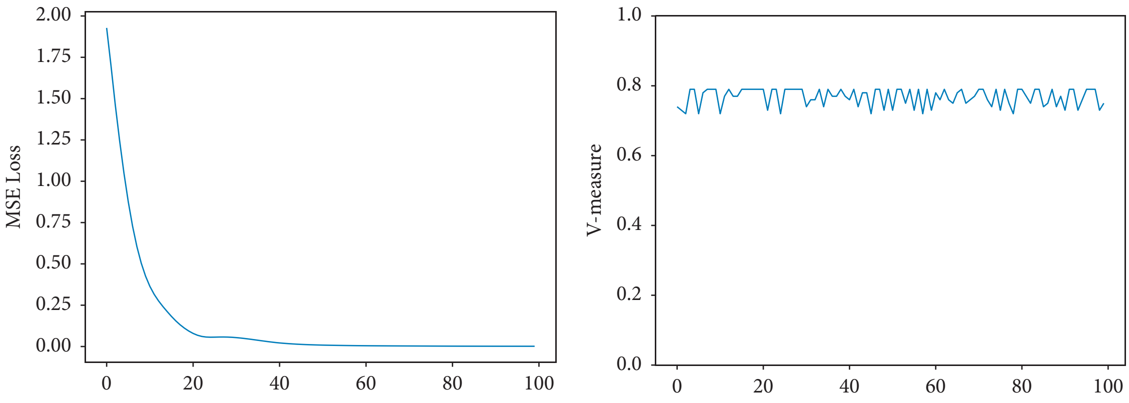
| | K-means | KNN + DPC | VaDE | DCN | SpectralNet | RBF-KNN |
|---|---|---|---|---|---|---|
| Aggregate | 0.4940 | **1.0** | 0.5261 | 0.5620 | 0.9497 | **1.0** |
| Jain | 0.3690 | 0.5335 | 0.2024 | 0.4587 | 0.4606 | **0.7075** |
| Flame | 0.3987 | 0.8865 | 0.3453 | 0.4025 | 0.4595 | **0.8993** |
| Spiral | 0.0268 | **1.0** | 0.0406 | 0.3872 | 0.3194 | **1.0** |
| Target | 0.2931 | 0.4655 | 0.7335 | 0.4982 | 0.7917 | **1.0** |
| Ecoli | 0.6409 | 0.5311 | 0.2664 | 0.6130 | 0.6241 | **0.6855** |
| Iris | 0.7581 | 0.6182 | 0.2645 | 0.7381 | 0.7660 | **0.7943** |
| Seeds | 0.6992 | 0.5884 | 0.2629 | 0.7200 | 0.5073 | **0.7165** |
| Soybean | 0.7157 | 0.7672 | 0.3872 | 0.6820 | 0.2992 | **0.9457** |
| Segment | 0.6040 | 0.3740 | 0.2024 | 0.6135 | 0.0995 | **0.6244** |

Table 4: Comparison of evaluation indicators ARI of different algorithms.

| | K-means | KNN + DPC | VaDE | DCN | SpectralNet | RBF-KNN |
|---|---|---|---|---|---|---|
| Aggregate | 0.3475 | **1.0** | 0.3289 | 0.3625 | 0.9231 | **1.0** |
| Jain | 0.3241 | 0.2784 | −0.0044 | 0.4214 | 0.2165 | **0.7900** |
| Flame | 0.4534 | 0.6742 | 0.2895 | 0.4629 | 0.2507 | **0.9502** |
| Spiral | 0.0359 | **1.0** | 0.0548 | 0.0115 | 0.3094 | **1.0** |
| Target | 0.2931 | 0.4358 | 0.2192 | 0.1568 | 0.8219 | **1.0** |
| Ecoli | 0.6720 | 0.3884 | 0.0193 | 0.6892 | 0.6552 | **0.6855** |
| Iris | 0.7302 | 0.5178 | 0.0192 | 0.7125 | 0.7437 | **0.7566** |
| Seeds | 0.7166 | 0.5521 | 0.1845 | 0.7083 | 0.4134 | **0.7285** |
| Soybean | 0.5452 | 0.7672 | 0.3262 | 0.6726 | 0.0756 | **0.9256** |
| Segment | 0.4373 | 0.0346 | 0.0342 | 0.3240 | 0.0012 | **0.5256** |

Table 5: Comparison of evaluation indicators NMI of different algorithms.

| | K-means | KNN + DPC | VaDE | DCN | SpectralNet | RBF-KNN |
|---|---|---|---|---|---|---|
| Aggregate | 0.4941 | **1.0** | 0.5282 | 0.3286 | 0.9498 | **1.0** |
| Jain | 0.3690 | 0.3034 | 0.2024 | 0.4265 | 0.4607 | **0.7075** |
| Flame | 0.3987 | 0.7628 | 0.3454 | 0.3255 | 0.4595 | **0.8993** |
| Spiral | 0.0267 | **1.0** | 0.0456 | 0.1153 | 0.2974 | **1.0** |
| Target | 0.2931 | 0.4726 | 0.2098 | 0.2656 | 0.8217 | **1.0** |
| Ecoli | 0.6440 | 0.5312 | 0.2663 | 0.5568 | 0.6307 | **0.6856** |
| Iris | 0.7582 | 0.6258 | 0.2657 | 0.7265 | 0.7661 | **0.7857** |
| Seeds | 0.6949 | 0.5884 | 0.2629 | 0.7024 | 0.5079 | **0.7124** |
| Soybean | 0.7158 | 0.8489 | 0.3581 | 0.6526 | 0.3057 | **0.9435** |
| Segment | 0.6048 | 0.4000 | 0.2831 | 0.5663 | 0.1427 | **0.6875** |



(a)

(b)

Figure 9: Performance of RBF-KNN. (a) Optimization processing. (b) Stability line chart.

GMM clustering algorithms, the DCN clustering algorithm combining Auto-encoder and K-means algorithms, and the semisupervised neural network. SpectralNet clustering algorithm is based on graph theory, of which the SpectalNet algorithm only selects $k$ labeling points in this paper.

It can be seen from the experimental results that compared with the traditional K-means and the DPC algorithm combined with KNN, the algorithm of this paper can obtain better clustering results on each data set, and the comprehensive comparison of the three indicators of V-measure, ARI, and NMI shows the result obtained by the proposed algorithm is more excellent (Tables 3–5).

Compared with the clustering algorithms VaDE and DCN that combine neural networks, both algorithms are clustering algorithms for image processing, and these two algorithms are inefficient in coping with low-dimensional datasets; SpectralNet as a semisupervised clustering algorithm also has poor clustering effect with less prior knowledge. As a self-supervising clustering algorithm, we get better results than the selected 5 algorithms from the three evaluation criteria, which show RBF-KNN has better stability performance and robustness.

*4.3. Discussion on RBF-KNN Algorithm Performance and Stability Study.* From the V-measure results of the artificial data set, it can be seen that the clustering accuracy of the RBF-KNN algorithm on the Jain data set is low, so the clustering process of the Jain data is selected for analysis in this section, and MSE loss in each iteration is shown in Figure 9(a), and the algorithm can quickly optimized in the 40th iteration. The main step of the algorithm is to map the data set to a Gaussian function and back-propagate, so the algorithm complexity of the RBF-KNN algorithm is $O(n^2 epochs)$.

In the absence of explicit class label correction, the neural network will cause the clustering process to greatly fluctuate due to initialization problems and missing label problems. In terms of stability, this section still uses the Jain data set and process 100 epochs. The V-measure result is shown in Figure 9(b), and it can be seen from the results that the clustering results of the proposed algorithm do not fluctuate significantly, and the proposed algorithm shows good stability.

## 5. Conclusion

In this paper, an RBF neural network clustering algorithm based on the K-nearest neighbor principle (RBF-KNN) is proposed, which belongs to a self-supervising clustering algorithm. The central idea is to use the full RBF network to retain global information, and then do back-propagation to self-supervise the resulting neural network based on K-nearest neighbor principle, in order to solve the problem of poor adaptability and lack of robustness of traditional clustering algorithms. From the processing results of the artificial data set and the UCI data set, we can see that the performance of the proposed algorithm is excellent, and it can handle well with the multitype and unbalanced data.

Based on the above analysis, compared with the lack of versatility of traditional clustering methods and the complex priori conditions of semisupervised clustering, the proposed algorithm can ensure the accuracy of clustering while the process is simple and the simple parameters setting, which shows more obvious advantages than the proposed traditional clustering algorithms and semisupervised neural network clustering algorithms.

## Data Availability

The artificial data used to support the findings of this study are included within the article. The UCI data used to support the findings of this study have been deposited in the "https://archive.ics.uci.edu/ml/index.php" repository.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by Shandong Provincial Natural Science Foundation(ZR2021MF045).

## References

[1] J. L. Chang, Y. W. Guo, L. F. Wang, M. Gaofeng, X. Shiming, and P. Chunhong, "Deep discriminative clustering analysis," 2019, https://arxiv.org/abs/1905.01681.

[2] J. J. Zhao, D. H. Lu, K. Ma, Y. Zhang, and Y. Zheng, "Deep image clustering with category-style representation," in *Proceedings of the European Conference on Computer Vision*, pp. 54–70, Springer, Berlin, Germany, August 2020.

[3] M. Tapaswi, M. T. Law, and S. Fidler, "Video face clustering with unknown number of clusters," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5027–5036, IEEE, Piscataway, NJ, U.S.A, January 2019.

[4] M. Wang and W. H. Deng, "Deep face recognition with clustering based domain adaptation," *Neurocomputing*, vol. 393, pp. 1–14, 2020.

[5] W. A. Lin, J. C. Chen, C. D. Castillo, and R. Chellappa, "Deep density clustering of unconstrained faces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8128–8137, IEEE, Piscataway, NJ, U.S.A, June 2018.

[6] Z. Khan and J. Yang, "Bottom-up unsupervised image segmentation using FC-Dense u-net based deep representation clustering and multidimensional feature fusion based region merging," *Image and Vision Computing*, vol. 94, Article ID 103871, 2020.

[7] L. Zhou and W. Y. F. Wei, "DIC: deep image clustering for unsupervised image segmentation," *IEEE Access*, vol. 8, pp. 34481–34491, 2020.

[8] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: simultaneous deep learning and clustering," *Proceedings of Machine Learning Research*, vol. 70, pp. 3861–3870, 2017.

[9] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern*

*Recognition*, pp. 5147–5156, IEEE, Piscataway, NJ, U.S.A, June 2016.

[10] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger, “SpectralNet: spectral clustering using deep neural networks,” in *Proceedings of the International Conference on Learning Representations*, pp. 267–280, IEEE, Piscataway, NJ, U.S.A, January 2018.

[11] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, “Variational deep embedding: an unsupervised and generative approach to clustering,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1965–1972, Margan Kaufmann, San Francisco, CA, U.S.A, January 2017.

[12] J. Y. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in *Proceedings of the International Conference on Machine Learning*, pp. 478–487, ACM, New York, NY, U.S.A, June 2016.

[13] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, “Deep adaptive image clustering,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5879–5887, IEEE, Piscataway,NJ, U.S.A, December 2017.

[14] W. H. Hu, T. Miyato, S. Tokuis, E. Matsumoto, and M. Sugiyama, “Learning discrete representations via information maximizing self-augmented training,” in *Proceedings of the International Conference on Machine Learning*, ACM, New York, NY, U.S.A, July 2017.

[15] Y. Ren, N. Wang, M. Li, and Z. Xu, “Deep density-based image clustering,” *Knowledge-Based Systems*, vol. 197, Article ID 105841, 2020.

[16] V. D. M. Laurens and G. E. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.

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

[18] N. Mrabah, N. M. Khan, R. Ksantini, and Z. Lachiri, “Deep clustering with a Dynamic Autoencoder: from reconstruction towards centroids construction,” *Neural Networks*, vol. 130, pp. 206–228, 2020.

[19] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, “Understanding and improving interpolation in autoencoders via an adversarial regularizer,” in *Proceedings of the International Conference on Learning Representations*, pp. 272–284, IEEE, Piscataway, NJ, U.S.A, July 2019.

[20] K. L. Lim, X. D. Jiang, and C. Y. Yi, “Deep clustering with variational autoencoder,” *IEEE Signal Processing Letters*, vol. 27, pp. 231–235, 2020.

[21] I. J. Goodfellow, A. J. Pouget, M. Mirza et al., “Generative adversarial networks,” *Advances in Neural Information Processing Systems*, vol. 27, no. 3, pp. 2672–2680, 2014.

[22] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo et al., “Deep unsupervised clustering with Gaussian mixture variational autoencoders,” 2016, https://arxiv.org/abs/1611.02648.

[23] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, “ClusterGAN: latent space clustering in generative adversarial networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4610–4617, AAAI, Menlo Park, CA, U.S.A, July 2019.