Enhanced Prediction Accuracy in Complex Systems: An Approach Integrating Fuzzy K-Clustering and Fuzzy Neural Network

Tichun Wang[*], Xianwei Wang, Hao Li

College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, 210000 Nanjing, China

[*] Correspondence: Tichun Wang (wangtichun2010@nuaa.edu.cn)

Abstract: The quest for heightened precision in fuzzy system predictions has culminated in the development of an innovative model that integrates a Fuzzy K-Clustering (FKC) algorithm with a fuzzy neural network (FNN). In this approach, the novel FKC algorithm, herein introduced, undertakes the clustering of sample data. Subsequently, the clustering outcomes inform the configuration of the FNN, specifically guiding the determination of node quantities across its layers and the initial network parameters. A distinctive hybrid learning algorithm, designated as the Conjugate Recursive Least Squares (CRLS), facilitates the optimization of network parameters via distinct methods tailored to parameter types. This model underwent empirical validation using 2-minute interval average wind speed data from surface meteorological stations in China. Analytical comparisons between model predictions and actual wind speed data revealed an average absolute error of 0.2764m/s, an average absolute percentage error of 2.33%, and a maximum error of 0.6035m/s. The findings substantiate the model's superior predictive capability. This study thus presents a significant advancement in fuzzy system prediction methodologies, underscoring the potential of the FKC and FNN in complex data analysis.

Keywords: Prediction model; Fuzzy K-Clustering; Fuzzy neural network; Hybrid learning algorithm

1 Introduction

In the pursuit of robust decision-making tools across various sectors, predictive models are paramount for reducing uncertainties and optimizing resource utilization. Traditional precise modeling and decision-making approaches encounter significant challenges in complex, uncertain domains, spurred by the explosive growth of data accompanying rapid advancements in information technology [1]. In such scenarios, fuzzy systems have been extensively utilized as a powerful mathematical instrument, particularly for tackling issues replete with fuzziness, uncertainty, and complexity [2]. Confronted with systems involving uncertainty, multiple variables, and intricate interactions, conventional methodologies often fail to model accurately. Fuzzy systems, celebrated for their flexibility, adeptly navigate such complexities, effectively managing uncertainties and ambiguities to furnish enhanced decision-making capabilities.

For accurate predictions in fuzzy systems, it is essential that inherent patterns and features within the data are identified, thereby facilitating the aggregation of pertinent data points. Two main strategies for data partitioning have been delineated [3]. The first is characterized by the uniform division of initial fuzzy subsets based on the data's domain, yet this technique is limited by an assumption of uniform data distribution—an assumption often not aligned with real-world data—potentially leading to subsets that do not truly reflect the actual distribution characteristics of the data. The second strategy, more commonly utilized, leverages clustering algorithms to cluster sample data and then projects this onto various dimensions to determine the initial fuzzy subsets [4]. A variety of clustering algorithms are available for this purpose, including nearest-neighbor clustering, k-means, hierarchical clustering, and Fuzzy C-Means (FCM) [5]. Bryant and Cios [6] introduced a method using reverse nearest-neighbor counts to estimate observation density, thus obviating the necessity of manually setting neighbor parameters, which in turn, has been shown to increase the accuracy of clustering. In the context of k-means clustering, Xia et al. [7] described an approach using spheres to represent each cluster, efficiently identifying neighboring clusters and reducing the need

for distance calculations, demonstrating superior performance and computational efficiency compared to traditional k-means methods. Chu and Xun [8] put forth a hierarchical clustering algorithm for categorical data based on the average of similarity measures, aiming to overcome the dependency on arbitrary similarity thresholds. Concerning FCM clustering, Lei et al. [9] addressed the algorithm's noise sensitivity and the added computational complexity from local spatial information by introducing morphological reconstruction operations to mitigate noise impact. While these clustering methods have improved outcomes to a degree, the choice of initial values still significantly impacts the results. Moreover, certain methods incorporate additional operations such as spherical descriptions and morphological reconstructions, potentially leading to higher computational costs, particularly when processing large-scale data.

Common prediction models that are currently utilized include time series methods, Support Vector Machine (SVM) approaches, and neural network techniques [10]. It has been identified that time series methods may fall short in their focus on data fitting without sufficiently considering factors that affect predictive accuracy [11]. The performance of SVM is highly contingent upon the selection of parameters; inappropriate parameter choices are known to compromise model efficiency [12]. FNN, on the other hand, possess robust adaptability, learning from data and autonomously adjusting weights and parameters to accommodate various types of problems, thereby achieving high-accuracy predictions. Numerous scholars have proposed methods for the learning of FNN parameters. Wang et al. [13] established a multi-layer feed-forward neural network endowed with fuzzy weights and thresholds. Hong et al. [14] utilized a back-propagation (BP) neural network to adjust membership functions. To address the slow convergence rates typical of classical BP algorithms, Gao [15] introduced an adaptive interval conjugate gradient method, which has been applied to the learning process of FNN parameters. While these methods exhibit certain advantages in handling specific types of network parameters, challenges persist in actual optimization processes, especially when networks contain a mix of linear and non-linear parameters, potentially leading to reduced computational speed and, consequently, affecting the predictive outcomes.

In response to the shortcomings identified in the aforementioned studies, a predictive model based on the FKC algorithm and FNN is introduced. Initially, the proposed FKC algorithm is employed to effectively partition the dataset, followed by the construction of a FNN structure based on the clustering outcomes. A hybrid learning algorithm, also introduced herein, is utilized to optimize network parameters, culminating in the model's precise prediction capabilities. Empirical evidence has demonstrated that the predictive model proposed herein exhibits a marked superiority over several alternative prediction models, achieving smaller errors and higher accuracy.

2 Cluster Analysis

2.1 Preliminary Knowledge

The K-means algorithm, a commonly employed hard clustering method, is centered around the iterative search for the minimal distance between data points and cluster centers, thereby effectuating data segmentation. Specifically, given a dataset $x = [x_1, \cdots, x_n]$, the algorithm's objective is to identify k cluster centers that minimize the Euclidean distance between data points and the said centers. Its advantages include high computational efficiency and distinct clustering outcomes. However, its sensitivity to the initial selection of cluster centers is a notable limitation [16]; different initial points can lead to disparate results.

In contrast to the K-means algorithm, the Fuzzy C-Means (FCM) algorithm represents a quintessential soft partitioning method. Here, the categorical membership of a data point does not adhere to the dichotomous hard partitioning rules but is determined by membership degrees that sum to one, indicating the likelihood of belonging to various categories. This algorithm incorporates a membership degree matrix to express the fuzzy relationships between data points and categories, allowing a data point to concurrently belong to multiple categories, thereby offering a more flexible classification. Although FCM holds certain advantages over traditional hard clustering methods, especially when dealing with diverse and overlapping data, it is characterized by high computational complexity due to the continuous updates required for membership degrees and cluster centers [17], with results prone to convergence at local minima.

2.2 Novel Clustering Algorithm Model and Procedure

A new clustering algorithm, denominated FKC, is introduced in this study. This algorithm synthesizes characteristics such as inter-category similarity and intra-category compactness, and devises a new clustering discriminant criterion named Similarity and Density (SDS). Diverging from traditional methodologies, a hybrid strategy is adopted that amalgamates the strengths of K-means and FCM. An initial clustering is conducted using K-means to determine preliminary cluster centers. Subsequently, a secondary fuzzy classification via FCM is applied to further refine the clustering outcomes.

2.2.1 Mathematical model of the FKC algorithm

The mathematical expression for the FKC algorithm is represented as an optimization problem seeking extreme values of the following objective function:

$$\text{Min } J(u,c) = \sum_{j=1}^{p} \sum_{i=1}^{k} u_{ij}^m d_{ij}(x_j, c_i) \tag{1}$$

where, p denotes the number of given sample data, k represents the number of categories after clustering by the FKC algorithm; $x = \{x_1, x_2, \cdots, x_p\}$ is a set containing p groups of data, signifying the input sample data; $c = \{c_1, c_2, \ldots, c_k\}$ designates the set of cluster centers, encompassing k cluster centers; $d_{ij}(x_j, c_i) = \|x_j - c_i\|^2$ signifies the Euclidean distance between data point x_j and cluster center c_i. u_{ij} represents the degree of membership of data point x_j to cluster center c_i; m is the fuzziness parameter, employed to control the level of fuzziness in the membership matrix u_{ij} [18]. A higher value of m increases the fuzziness of the categorization; when $m \to \infty, u_{ij} \to 1/k$, classification information can no longer be provided.

2.2.2 New criterion for clustering discrimination

An exemplary criterion for clustering discrimination should take into account the compactness within a category and the separation between different categories [19] to ensure that classification results distinctly delineate each category while maintaining cohesion within them.

The expression for the clustering discrimination criterion, Similarity and Density (SDS), is as follows:

$$SDS = \frac{1}{k} \sum_{j=1}^{k} \max(u_{ij}) - \frac{1}{G} \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \left[\frac{1}{k} \sum_{c=1}^{k} \min(u_{ic}, u_{jc}) \right] \tag{2}$$

where the value of G is given by:

$$G = \sum_{i=1}^{p-1} i \tag{3}$$

The clustering discrimination criterion is composed of two parts.

The first part reflects the degree of compactness within the same category. For each data point x_i, the closer the maximum membership degree $\max(u_{ij})$ is to 1, the closer the sample is to the center of its respective fuzzy category. Hence, a fuzzy set $\max(u_{ij})$ is considered a good classification indicator, with higher values denoting greater compactness within a category.

The second part reflects the degree of separation between categories. The separation between c_i and c_j is assessed using the intersection of two fuzzy sets. If a data point x_i is close to a category center c_i, then $\min(u_{ic}, u_{jc})$ will approach zero, indicating a clear separation between the categories c_i and c_j. Conversely, if $\min(u_{ic}, u_{jc})$ approaches $1/p$, it implies that the data point x_i has an equal degree of membership across all categories, which in turn denotes the most ambiguous category division.

2.2.3 Flow of the FKC algorithm

The flow of the FKC algorithm is shown in Figure 1.

Step 1. Set the maximum number of clusters Z=10

Step 2. Divide the given sample data $x = [x_1, \cdots, x_p]$ into k categories and randomly select k samples as the initial cluster centers

Step 3. Calculate the Euclidean distance between all data points and the cluster centers, and assign each data point to the class of the nearest cluster center

Step 4. Update the cluster center of each category, that is, take the average value of all data points in the category as the new cluster center

Step 5. If the difference between the two cluster center vectors is less than the set threshold or the maximum number of iterations has been reached, proceed to Step 6; otherwise, return to Step 3

Step 6. Take the cluster centers c_j obtained in Step 5 as the initial cluster centers for Step 7

Step 7. Set the fuzziness parameter m and the permissible error of the system ε. The parameter m determines the fuzziness of the membership degree and is usually set to 2

Step 8. Calculate the membership matrix u_{ij} between data points x_i and cluster centers c_j

Step 9. Update the cluster centers c_j according to the membership matrix u_{ij}
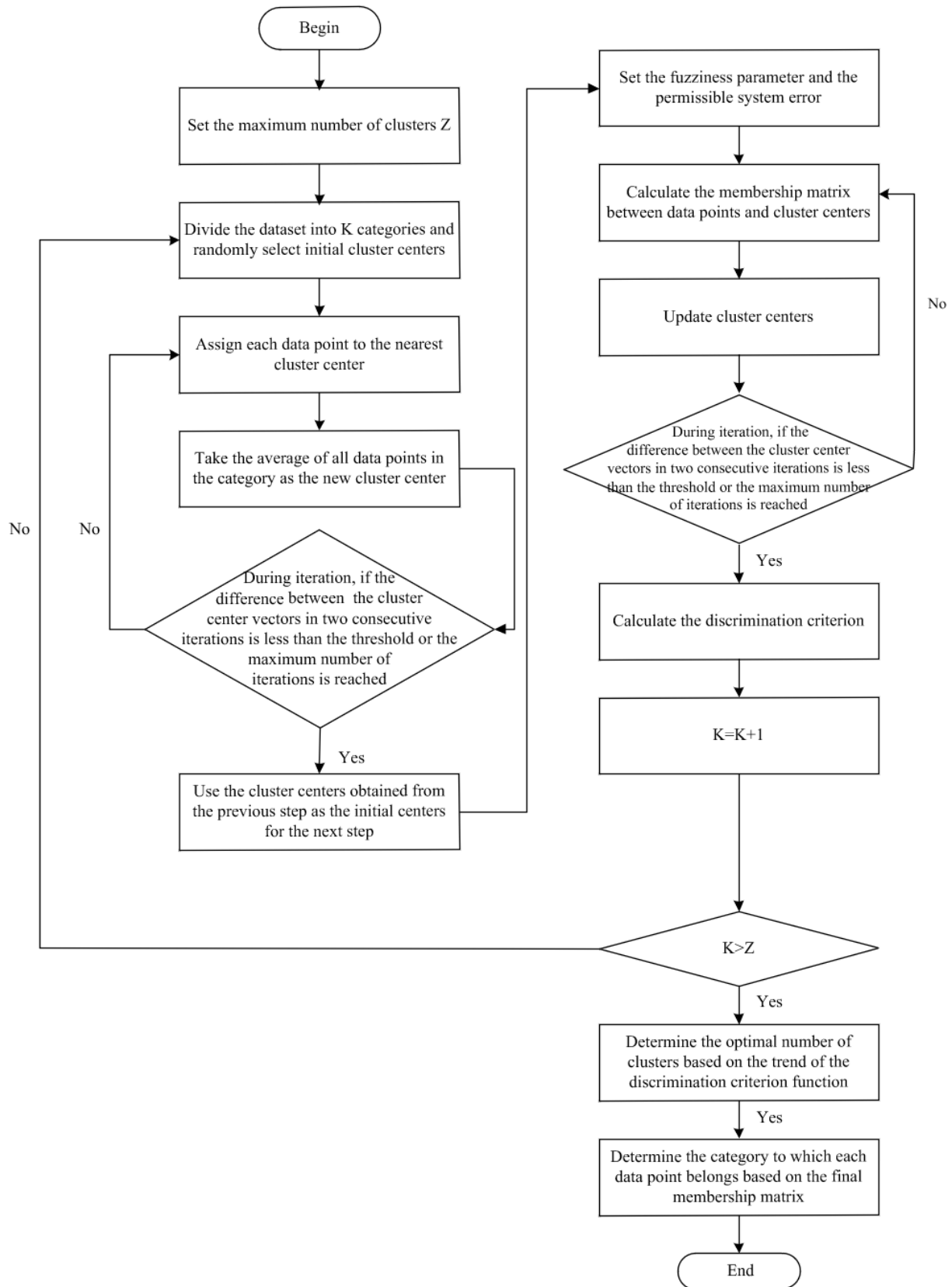
Figure 1. Flow of the FKC algorithm

Step 10. If the difference between the two cluster center vectors is less than the set threshold or the maximum number of iterations has been reached, proceed to Step 11; otherwise, return to Step 8

Step 11. Calculate the clustering discrimination criterion

Step 12. Increment the number of categories k=k+1

Step 13. If the number of categories k exceeds the maximum number of clusters Z, proceed to Step 14; otherwise, return to Step 2

Step 14. Assume that as the number of categories increases from 1 to K, the discrimination criterion function approaches 1, and when the number of categories exceeds K, the discrimination criterion does not increase or even shows a decreasing trend, indicating that the optimal number of clusters K has been found

Step 15. After the iteration is complete, determine the category of each data point according to the final membership matrix

3 Establishment and Parameter Optimization of FNN

3.1 Determination of Initial Structure

The FKC algorithm was employed to cluster the dataset, resulting in k categories, each with its respective cluster centers, denoted by $c_j = \left(c_{1j}, c_{2j}, \ldots, c_{nj}, c_{(n+1)j}\right), j = 1, 2, \ldots, k$. Utilizing these cluster centers, an initial model of a FNN can be established.

The inference rules of the model are expressed as follows:

$$\text{IF } x_1 \text{ is } A_1^j(x_1) \text{ and } x_2 \text{ is } A_2^j(x_2) \text{ and } \ldots \text{ and } x_n \text{ is } A_n^j(x_n), \text{ Then } y \text{ is } w_j, j = 1, 2, \ldots, k \qquad (4)$$

where, $A_i^j (i = 1, 2, \ldots, n; j = 1, 2, \ldots, k)$ is the value set of variable x_i. The membership function defining the degree to which x_i belongs to A_i^j is defined as $u_i^j (i = 1, 2, \ldots, n; j = 1, 2, \ldots, k)$, which can be represented by a Gaussian function with a favorable fit:

$$u_i^j = \exp\left[-(x_i - m_{ij})^2 / 2\sigma_{ij}^2\right], i = 1, 2, \ldots, n, j = 1, 2, \cdots, k \qquad (5)$$

where, m_{ij} is the center of the j-th membership function of the i-th variable; σ_{ij} is the width of the j-th membership function of the i-th variable, indicating the deviation of the i-th variable from its cluster center.

$$m_{ij} = (m_{1j}, m_{2j}, \cdots, m_{nj}) = (c_{1j}, c_{2j}, \cdots, c_{nj}) \qquad (6)$$

$$\sigma_{ij} = (\sigma_{1j}, \sigma_{2j}, \cdots, \sigma_{nj}) \qquad (7)$$

The values of σ_{ij} are determined by:

$$\sigma_{ij} = |c_j - \tilde{c}_j| / \varsigma \qquad (8)$$

where, c_j is the j-th cluster center, and $\tilde{c}_j$ is the nearest cluster center to the j-th center. The value of ς ranges from $[2, 4]$. Consequently, the conclusion function in the rules is defined as:

$$w_j = p_{j0} + p_{j1}x(1) + p_{j2}x(2) + \cdots + p_{jn}x(n) \qquad (9)$$

where, the initial value of the constant term p_{j0} is $p_{j0} = c_{(n+1)j}$.

The final rule output is given by:

$$Y = \left(\sum_{j=1}^{k} h_j w_j\right) / \left(\sum_{j=1}^{k} h_j\right) \qquad (10)$$

where, $h_j = u_i^j(x_1) u_i^j(x_2) \ldots u_i^j(x_n) = \prod_{i=1}^{n} u_i^j(x_i)$
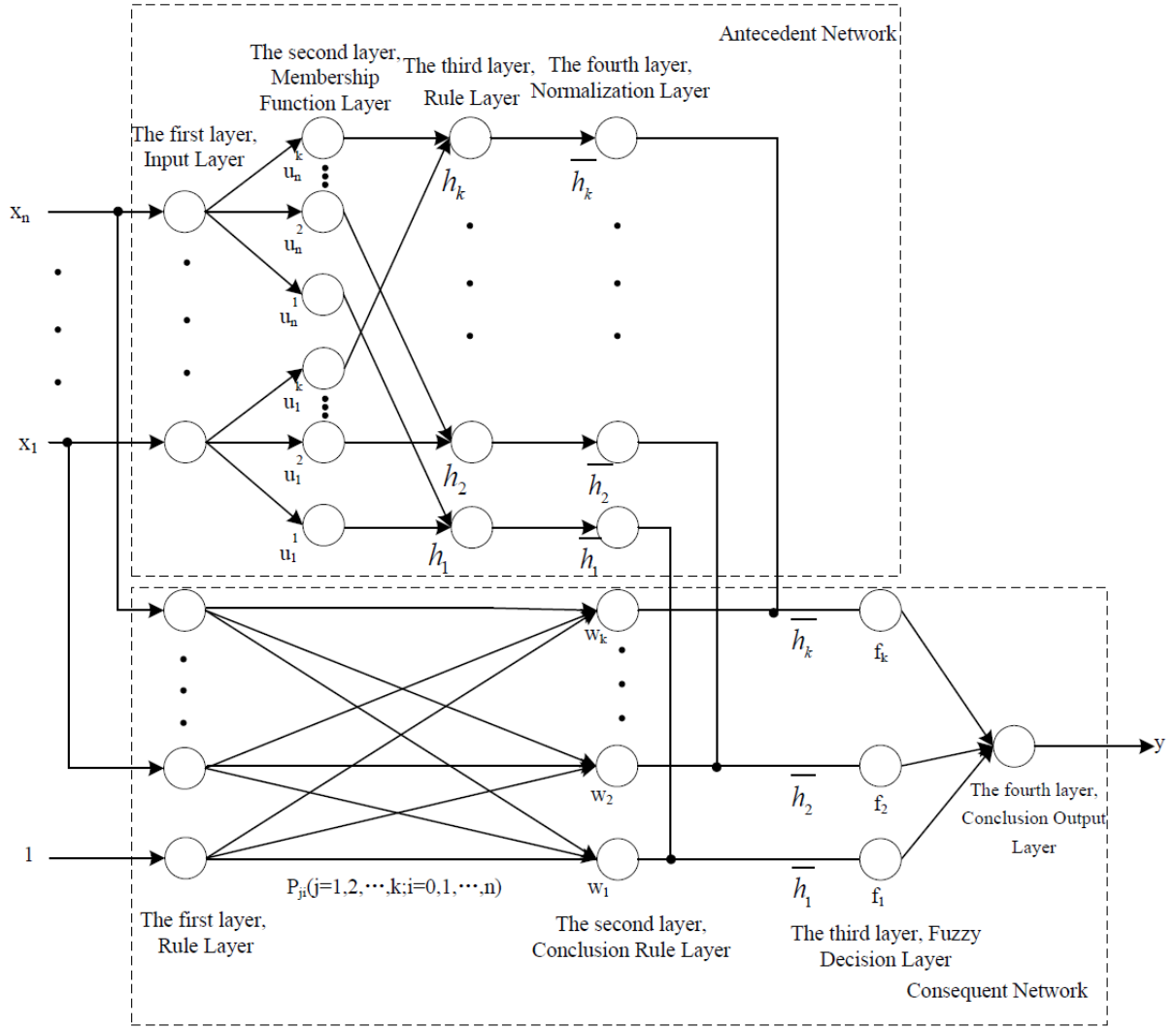
Figure 2. Structure of the FNN

Based on the number of input variables n, the number of clusters k, and the parameters of the inference rules u_i^j, h_j, and w_j, the number of nodes in each layer of the FNN and the initial values of the parameters are determined. The structure of the FNN is illustrated in Figure 2.

The specific functions of the antecedent network are as follows:

The first layer: Input Layer. Each input variable corresponds to one input node, making a total of n input nodes. Each input node transmits data to the next layer [20], and is connected to the k nodes of the second layer, where k is the number of clusters.

The second layer: Membership Function Layer. This layer receives data from the input layer and calculates the degree of membership of the input data using Gaussian functions as membership functions. There are n*k nodes in this layer, each representing a different fuzzy set, and their output reflects the degree of membership of the corresponding fuzzy set.

The third layer: Rule Layer. This layer contains k nodes, each representing a fuzzy rule, forming a fuzzy rule base consisting of k rules. Each node is connected only to the corresponding nodes in the membership function layer for the input variables.

The fourth layer: Normalization Layer. This layer performs normalization calculations on the output of the rule layer and contains k nodes. The normalization calculation formula is $\bar{h}_j = \frac{h_j}{\sum_{j=1}^{k} h_j}$.

The specific functions of the consequent network are as follows:

The first layer: Input Layer. This layer has n+1 nodes. Among them, the input of the 0-th node is 1, which provides a constant term for the generated rule consequent. The remaining nodes receive input variables from x_1 to x_n.

The second layer: Conclusion Rule Layer. This layer contains k nodes, each representing a rule.

The third layer: Fuzzy Decision Layer. This layer has k nodes, with the output of the j-th node being $f_j = w_j \bar{h}_j$.

The fourth layer: Conclusion Output Layer. Based on the output of the fuzzy decision layer, the final output of the model is calculated $y(x) = \sum_{j=1}^{k} f_j$.

3.2 Parameter Optimization

Upon the completion of the initial structure of the FNN, optimization of the centers m_{ij} and widths σ_{ij} of the membership functions in the antecedent network, as well as the conclusion parameter W of the consequent network, is required. This process is aimed at progressively approximating the expected output [21], thereby enhancing the precision of the network, wherein parameters m_{ij} and σ_{ij}, belonging to the premises of the fuzzy rules, exhibit non-linear characteristics; whereas, parameter W corresponding to the conclusion part of the fuzzy inference rules, demonstrates linear characteristics [22].

In light of this, a hybrid learning algorithm, termed CRLS, is proposed. This method employs a recursive least squares algorithm for the optimization of linear parameter W and a conjugate gradient algorithm for the optimization of m_{ij} and σ_{ij}.

3.2.1 Optimization of linear parameters

The expected output of the FNN can be represented as:

$$d(i) = \sum_{j=1}^{k} g_j(i)\theta_j + \varepsilon(i) \tag{11}$$

where, $d(i)$ is the desired output; $g_j(i)$ represents the regression coefficients; θ_j denotes the j-th linear parameter; and $\varepsilon(i)$ is the error.

When p sets of sample data are processed through the FNN, the expected output can be expressed as:

$$D(p) = G(p)\Theta(p) + E(p) \tag{12}$$

where,

$$D(p) = [d(1)d(2)\ldots d(p)]^T \tag{13}$$

$$G_p = H^T = \left[g^T(1)g^T(2)\ldots g^T(p)\right]^T \tag{14}$$

$$\Theta(p) = W^T = [\theta_1\theta_2\ldots\theta_p]^T \tag{15}$$

$$E(p) = [\varepsilon(1)\varepsilon(2)\ldots\varepsilon(p)]^T \tag{16}$$

By setting $Q(p) = \left(G^T(p)G(p)\right)^{-1}$, the linear parameter W can be derived:

$$W = \hat{\Theta}(p) = \hat{\Theta}(p-1) + L(p)\left(d(p) - G^T(p)\hat{\Theta}(p-1)\right) \tag{17}$$

where,

$$L(p) = Q(p)G(p) = Q(p-1)G(p)\left[1 + G^T(p)Q(p-1)G(p)\right]^{-1} \tag{18}$$

$$Q(p) = \left[I - L(p)G^T(p)\right]Q(p-1) \tag{19}$$

3.2.2 Optimization of non-linear parameters m_{ij} and σ_{ij}

The performance index of the FNN can be represented as:

$$E_i = \frac{1}{2}(d_i - y_i)^2, i = 1, 2, \ldots, p \tag{20}$$

where, for the i-th input vector, d_i denotes the desired output, and y_i represents the actual output; p indicates the number of samples.

To optimize the non-linear parameters m_{ij} and σ_{ij}, the centers m_{ij} and widths σ_{ij} of the membership functions of the FNN are combined into a set $M = \{m, \sigma\}$, with the performance index serving as the optimization objective function:

$$\min_M E_i = \frac{1}{2}(d_i - y_i)^2 \tag{21}$$

The optimization steps are as follows:

Step 1. Assign an initial point $M^{(0)}$ with precision $\varepsilon > 0$, where D is the number of elements in vector M.

Step 2. Set the initial value, $g_0 = \nabla E(M^0)$, n = 0; define $S^{(n)} = -g_n = -\nabla E(M^{(n)})$ as the search direction vector for a cycle of n times, with the gradient direction g for each parameter in M being:

$$\frac{\partial E}{\partial m} = (d_p - y_p)\frac{\partial f}{\partial \xi_j}\frac{\partial \xi}{\partial m_{ij}} \tag{22}$$

$$\frac{\partial E}{\partial \sigma} = (d_p - y_p)\frac{\partial f}{\partial \xi_j}\frac{\partial \xi}{\partial \sigma_{ij}} \tag{23}$$

where, $\frac{\partial f}{\partial \xi_j} = \frac{w_j(p)-y_p}{\sum_{j=1}^k \xi_j}$, $\frac{\partial \xi_j}{\partial m_{ij}} = \frac{(x_{ip}-m_{ij}(p-1))}{\sigma_{ij}^2(p-1)}\xi_j$, $\frac{\partial \xi_j}{\partial \sigma_{ij}} = \frac{(x_{ip}-m_{ij}(p-1))^2}{\sigma_{ij}^3(p-1)}\xi_j$, j = 1, 2, \ldots, k.

Step 3. Set the learning rates for the centers m_{ij} and widths σ_{ij} of the membership functions in M as η_1 and η_2, respectively, that is, $\eta = \{\eta_1, \eta_2\}$. To avoid system instability caused by improper selection of learning rates, an adaptive adjustment method for learning rates is utilized. Thus, the new vector $M^{(n+1)}$ in this direction is:

$$M^{(n+1)} = M^{(n)} + \eta s^{(n)} \tag{24}$$

where, when $\Delta E > 0$, $\eta = \eta\varphi, \varphi > 1$; when $\Delta E < 0, \eta = \eta\beta, \beta > 1$; φ, β are constants. The corresponding gradient is $g_{n+1} = \nabla E(M^{(n+1)})$.

Step 4. If $E_i < \varepsilon$, then the iteration concludes; otherwise, proceed to Step 5.

Step 5. If $n < D - 1$, compute the conjugate direction $s^{(n+1)} = -g_{n+1} + \frac{\|g_{n+1}\|^2}{\|g_n\|^2}s^{(n)}$, n = n + 1, and execute Step 3; otherwise, set $M^{(0)} = M^{(D)}$ and execute Step 2.

3.2.3 The process of parameter optimization using the hybrid learning algorithm

Figure 3 shows the flow of the CRLS algorithm.

Step 1. Initialize the iteration count n = 0, error precision $\varepsilon > 0$, and define learning rate η, etc.

Step 2. For the given set of p data groups, forward propagate through the FNN and acquire the conclusion parameter W of the consequent network using the recursive least squares method.

Step 3. Obtain the FNN output Y, and calculate the error E.

Step 4. Back propagate the error E through the FNN to obtain the gradient vector related to the membership function centers m_{ij} and widths σ_{ij} of the antecedent network.

Step 5. Adjust the values of m_{ij} and σ_{ij} using the conjugate gradient algorithm.

Step 6. Increment the iteration count k = k + 1.

Step 7. Check whether the set iteration number or error precision requirements are met. If not, return to Step 2; otherwise, the algorithm terminates, and the parameter optimization process for the FNN is complete.
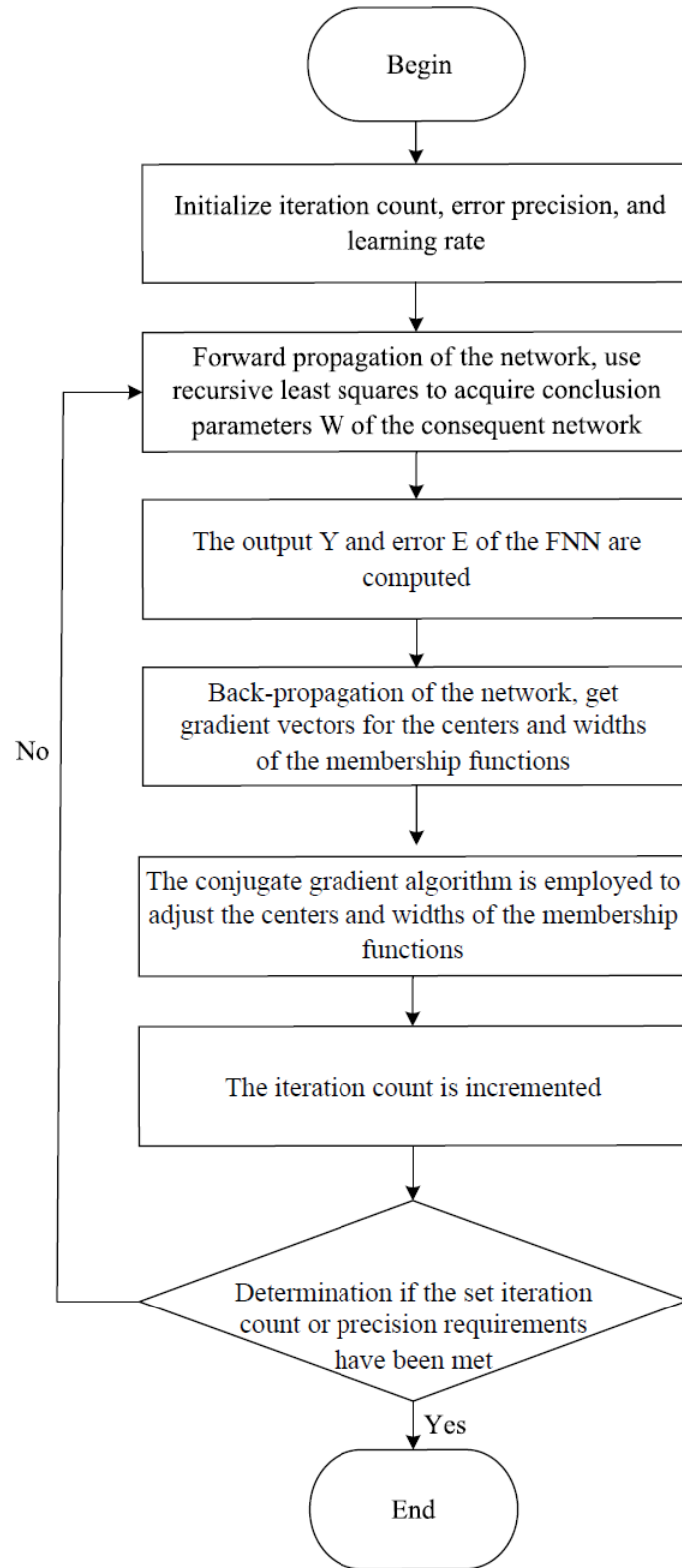
Figure 3. Flow of the CRLS algorithm

4 Experimental Verification

4.1 Evaluation Criteria

The ground meteorological observation data collected by the China Meteorological Administration are employed as a case study, with a data sampling interval of 2 minutes, documenting average wind speeds at different times.

The wind speed at the next forecast time point $v(t+1)$ is defined as the decision attribute. Considering the high correlation between the wind speeds of the previous 15 minutes $v(t-14), v(t-13), \ldots, v(t)$ and the forecast time, these wind speeds are listed as influencing factors. Data from one month in the historical records are selected for FKC clustering analysis, and data at 30 time points from a subsequent day are chosen as test samples to assess the predictive accuracy of the model.

The model's predictive performance is measured using three indicators: mean absolute error (E_{MAE}), mean absolute percentage error (E_{MAPE}), and maximum error (E_{MAX}), whose expressions are as follows:

$$E_{MAE} = \frac{1}{n} \sum_{i=1}^{n} |v_i - v_i'| \tag{25}$$

$$E_{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|v_i - v_i'|}{v_i} \tag{26}$$

$$E_{MAX} = \max\left(|v_i - v_i'|\right) \tag{27}$$

where, v_i represents the actual wind speed values and v_i' denotes the model-predicted wind speed values.

4.2 Implementation and Analysis of the Prediction Model

The FKC algorithm was employed for data clustering. The parameters for the FKC algorithm are set as follows: the number of cluster categories k=1, with k initial cluster centers chosen at random, the maximum number of iterations $T_{\max} = 20$, and the fuzziness index m=2. When the number of cluster categories increased from 1 to 10, it's found that the growth of the criterion function value for cluster discrimination was rapid from 1 to 4 but almost negligible from 4 to 10. Consequently, the optimal number of clusters was determined to be 4 (Figure 4).
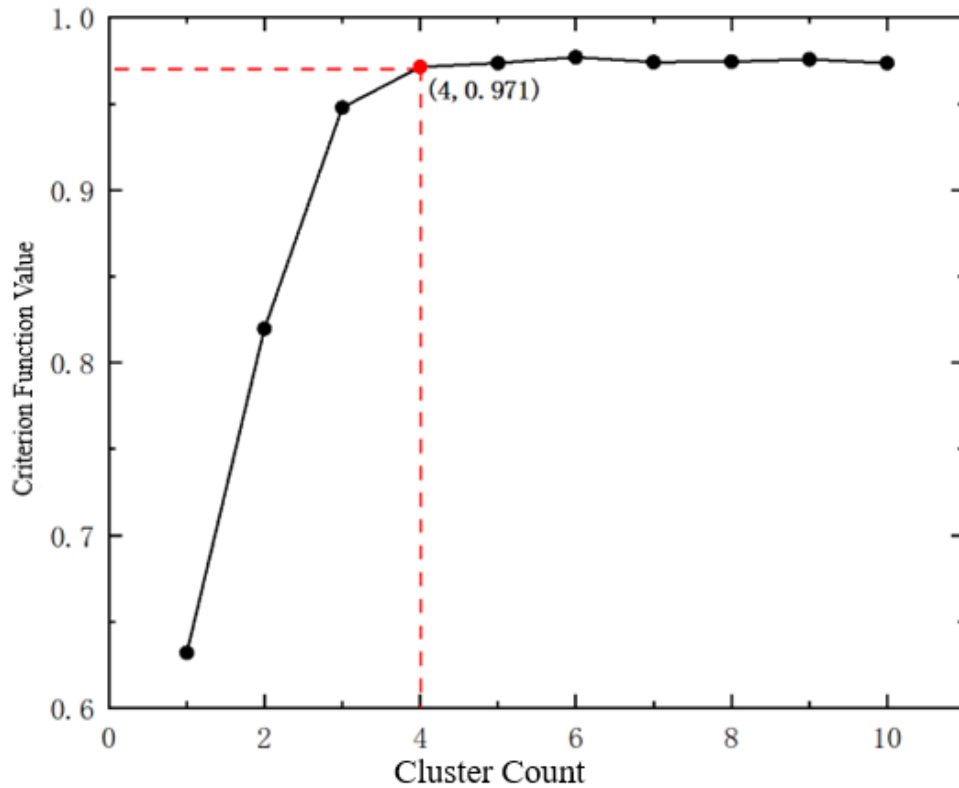

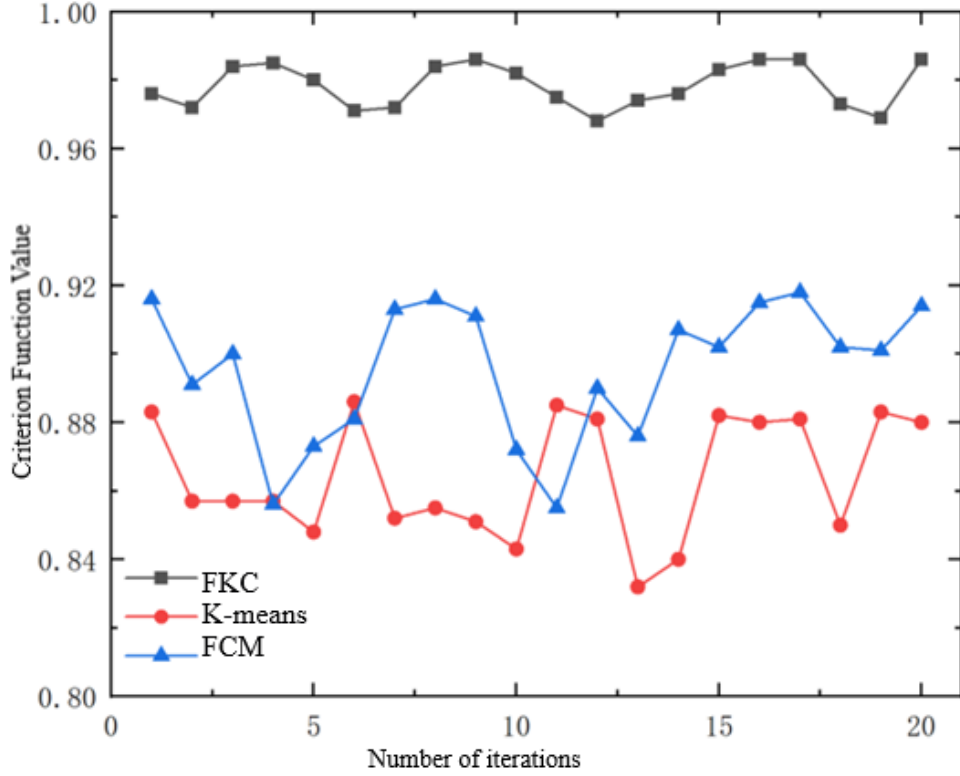**Figure 4.** Determination of the optimal number of clusters

Figure 5. Iteration results of the clustering algorithms

When the number of clusters was 4, FKC was compared experimentally with K-means and FCM algorithms. The clustering effectiveness of these three algorithms was compared and analyzed using the cluster discrimination criterion proposed in this paper. The closer the criterion function value is to 1, the better the clustering effect. The iterative results of the three algorithms are shown in Figure 5.

Analysis of Figure 5 reveals that, during 20 iterations, the criterion function value of the FKC algorithm ranged within [0.96,0.98], while the function value for the K-means algorithm was between [0.832,0.886], and for the FCM algorithm, it was between [0.855,0.918]. Therefore, the clustering result of the FKC algorithm is superior to the other two algorithms. Furthermore, the fluctuation amplitude of the FKC algorithm did not show as drastic changes as the other two algorithms with the increase of iterations, indicating that its stability is also superior to the others.

Since the wind speed of the previous 15 moments $v(t-14), v(t-13), \ldots, v(t)$ can affect the wind speed of the next forecast moment $v(t+1)$, it is necessary to construct a fuzzy neural network with 15 inputs and 1 output, where the input variables are the wind speeds of the previous 15 moments, and the output variable is the wind speed of the next forecast moment. Through the FKC algorithm, the data was divided into 4 categories, thereby determining the structure of the fuzzy neural network. The number of nodes in each layer of the fuzzy neural network is shown in Table 1.

Table 1. Number of nodes in each layer of the FNN

No.	FNN Layer	Associated Network	Number of Nodes
1	Input Layer	Antecedent network	15
2	Membership Function Layer	Antecedent network	60
3	Rule Layer	Antecedent network	4
4	Normalization Layer	Antecedent network	4
5	Rule Layer	Consequent network	16
6	Conclusion Rule Layer	Consequent network	4
7	Fuzzy Decision Layer	Consequent network	4
8	Conclusion Output Layer	Consequent network	1

To verify the effectiveness of the proposed method, the particle swarm optimization (PSO) algorithm, FNN algorithm, the neural network algorithm based on FCM, and the proposed method were adopted to predict the data

of 30 time points on a certain day. The prediction results are shown in Figure 6, and the values of the three evaluation indicators are shown in Table 2.
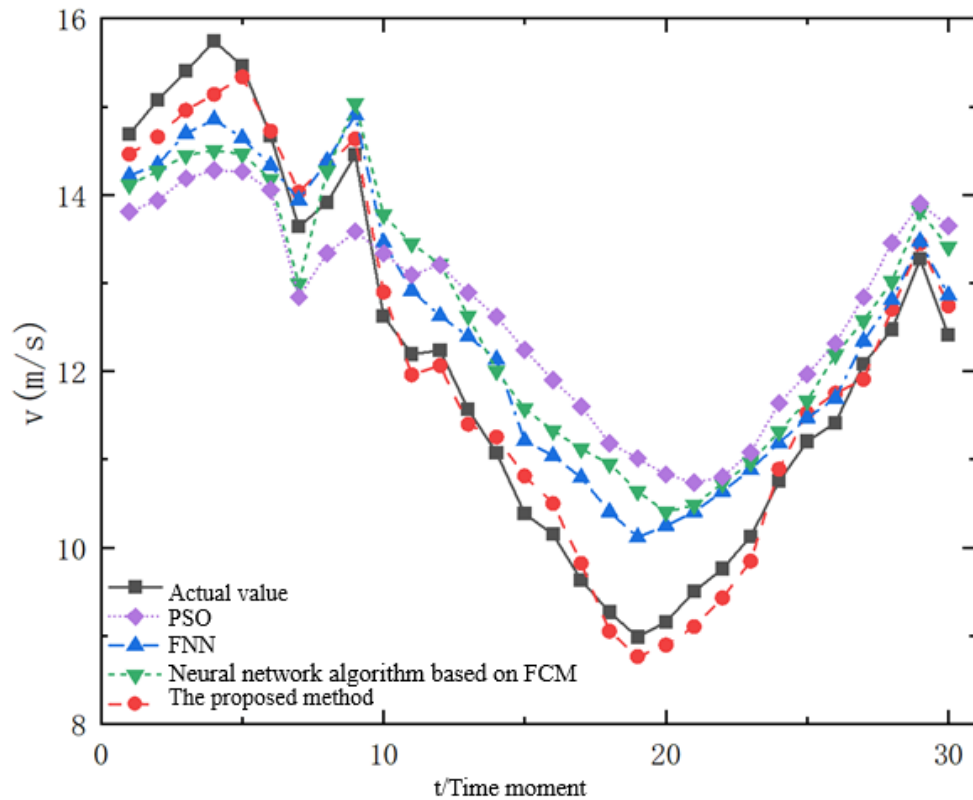

Figure 6. Prediction effect

Table 2. Comparison of prediction results

Method	E_{MAE}	E_{MAPE}	E_{MAX}
PSO	1.1597	0.1002	2.0268
FNN	0.9207	0.0811	1.6837
Neural network algorithm based on FCM	0.6677	0.0592	1.1606
The proposed method	0.2764	0.0233	0.6035

Table 2 shows that compared to the first method, the other three methods have significantly improved across all indicators, indicating that the FNN algorithm outperforms the particle swarm optimization in terms of forecasting performance. The neural network algorithm based on FCM, compared to the FNN algorithm, has seen reductions in EMAE and EMAPE by 27.48% and 27% respectively, with EMAX decreasing from 1.6837 to 1.1606. This suggests that the fuzzy clustering method can provide the model with training samples of higher similarity, effectively enhancing the accuracy of predictions. This paper introduces a new fuzzy clustering algorithm, FKC, which, compared to the neural network algorithm based on FCM, shows decreases in EMAPE by 58.6% and 60.64%, with EMAX falling from 1.1606 to 0.6035. Therefore, the method proposed in this study significantly improves the performance of the forecast.

5 Conclusion

In the realm of meteorological prediction, advancements have been achieved through the introduction of a novel model employing the FKC algorithm in tandem with FNN. Validation of this model has been performed utilizing data pertaining to the average wind speed, recorded at two-minute intervals from terrestrial meteorological observations across China. Findings from this research have demonstrated a substantial enhancement in predictive accuracy over four comparative methods, as evidenced by improvements across three critical evaluative measures: mean absolute error, mean absolute percentage error, and maximum error.

It has been observed that the predictive capability of the proposed model excels, suggesting a considerable step forward in the predictive modelling domain. Such an enhancement in performance underscores the robustness and potential applicability of the model for real-world scenarios, where precision in meteorological forecasting is paramount.

Looking ahead, the focus will be directed towards the fortification of the theoretical underpinnings of the utilized methodologies. Concurrently, the development of an intuitive system interface is anticipated, with the aim of broadening the reach and operational application of the model. This initiative is expected to yield a more user-centered tool, facilitating the model's integration into service-oriented sectors, thereby augmenting the quality and efficiency of meteorological services provided to society.

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] B. Li, Z. Xu, and Y. Qin, "The development track and research characteristics in the field of automatic control and decision making: A comprehensive bibliometric analysis of control and decision," *Control and Decision*, vol. 37, no. 6, pp. 1583–1590, 2022. https://doi.org/10.13195/j.kzyjc.2021.0267

[2] B. Zhang, "Research on data-driven crowd evacuation decision rule mining method," Ph.D. dissertation, Xidian University, 2023. https://doi.org/10.27389/dcnki.gxadu.2021.003529

[3] B. Shen, "Data-driven intelligent warning method and application for groundwater risk," Ph.D. dissertation, North China University of Water Resources and Electric Power, 2023. https://doi.org/10.27144/d.cnki.ghbsc.2022.000181

[4] L. Bin, M. Shahzad, H. Khan, M. M. Bashir, A. Ullah, and M. Siddique, "Sustainable smart agriculture farming for cotton crop: A fuzzy logic rule based methodology," *Sustainability*, vol. 15, no. 18, p. 13874, 2023. https://doi.org/10.3390/su151813874

[5] Z. Yu, L. Li, Y. Gao, J. You, J. Liu, H. S. Wong, and G. Han, "Hybrid clustering solution selection strategy," *Pattern Recogn.*, vol. 47, no. 10, pp. 3362–3375, 2014. https://doi.org/10.1016/j.patcog.2014.04.005

[6] A. Bryant and K. Cios, "RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, 2018. https://doi.org/10.1109/tkde.2017.2787640

[7] S. Xia, D. Peng, D. Meng, C. Zhang, G. Wang, E. Giem, W. Wei, and Z. Chen, "Ball k-means: Fast adaptive clustering with no bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 87–99, 2020. https://doi.org/10.1109/TPAMI.2020.3008694

[8] K. Chu and Y. Xun, "A hierarchical clustering analysis algorithm for classification data based on mean similarity," *Comput. Technol. Dev.*, vol. 32, no. 11, pp. 154–163, 2022. https://doi.org/10.3969/j.issn.1673-629X.2022.11.023

[9] T. Lei, X. Jia, Y. Zhang, L. He, H. Meng, and A. K. Nandi, "Significantly fast and robust fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3027–3041, 2018. https://doi.org/10.1109/TFUZZ.2018.2796074

[10] X. Niu and J. Wang, "A combined model based on data preprocessing strategy and multi-objective optimization algorithm for short-term wind speed forecasting," *Appl. Energy*, vol. 241, pp. 519–539, 2019. https://doi.org/10.1016/j.apenergy.2019.03.097

[11] H. Sharadga, S. Hajimirza, and R. Balog, "Time series forecasting of solar power generation for large-scale photovoltaic plants," *Renew. Energy*, vol. 150, pp. 797–807, 2020. https://doi.org/10.1016/j.renene.2019.12.131

[12] R. M. Adnan, O. Kisi, R. R. Mostafa, A. N. Ahmed, and A. El-Shafie, "The potential of a novel support vector machine trained with modified mayfly optimization algorithm for streamflow prediction," *Hydrol. Sci. J.*, vol. 67, no. 2, pp. 161–174, 2022. https://doi.org/10.1080/02626667.2021.2012182

[13] Y. Wang, H. Ishibuchi, M. J. Er, and J. Zhu, "Unsupervised multilayer fuzzy neural networks for image clustering," *Inf. Sci.*, vol. 622, pp. 682–709, 2023. https://doi.org/10.1016/J.INS.2022.11.149

[14] C. S. Hong, J. Huang, Y. Y. Guan, and X. Q. Ma, "BP/RBF neural network coupled with fuzzy rules for combustion control of power plant boilers," *J. Eng. Thermal Energy Power*, vol. 36, no. 4, pp. 142–148, 2021. https://doi.org/10.16146/j.cnki.rndlgc.2021.04.021

[15] T. Gao, "Research on fuzzy neural network parameter learning strategy and a generalized recursive network structure," Ph.D. dissertation, China University of Petroleum (East China), 2020. https://doi.org/10.27644/d.cnki.gsydu.2020.000090

[16] J. Luo, Z. Suo, and Q. Guo, "A k-nearest neighbor based density peak clustering algorithm," *Software*, vol. 41, no. 7, pp. 185–188, 2020. https://doi.org/10.3969/j.issn.1003-6970.2020.07.037

[17] K. Sri, R. Linda, and G. Elyza, "Fuzzy linear regression based on a hybrid of fuzzy C-means and the fuzzy inference system for predicting serum iron levels in patients with chronic kidney disease," *Expert Syst. with Appl.*, vol. 227, 2023. https://doi.org/10.1016/j.eswa.2023.120314

[18] D. Zhang, "Generation and optimization of fuzzy rule library," Ph.D. dissertation, Yanshan University, 2006. https://doi.org/10.7666/d.y867215

[19] F. Wang, Q. Wang, F. Nie, Z. H. Li, W. Z. Yu, and R. Wang, "Unsupervised linear discriminant analysis for jointly clustering and subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1276–1290, 2019. https://doi.org/10.1109/TKDE.2019.2939524

[20] P. Souza, "Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature," *Appl. Soft Comput.*, vol. 92, 2020. https://doi.org/10.1016/j.asoc.2020.106275

[21] C. Wang, "Research on collaborative estimation strategy for soc and soh of lithium battery based on bp neural network correction and optimization," Ph.D. dissertation, Southwest University of Science and Technology, 2023. https://doi.org/10.27415/d.cnki.gxngc.2023.000047

[22] L. Meng, W. Xie, L. Li, Y. Li, and Z. Liu, "Hesitant hierarchical T–S fuzzy system with fuzzily weighted recursive least square," *Eng. Appl. Artif. Intell.*, vol. 126, p. 106812, 2023. https://doi.org/10.1016/j.engappai.2023.106812