



ARTICLE

Application of FCM Algorithm Combined with Artificial Neural Network in TBM Operation Data

Jingyi Fang¹, Xueguan Song², Nianmin Yao¹ and Maolin Shi^{2,*}

¹School of Computer Science and Technology, Dalian University of Technology, Dalian, 116024, China

²School of Mechanical Engineering, Dalian University of Technology, Dalian, 116024, China

*Corresponding Author: Maolin Shi. Email: sh15985336@126.com

Received: 16 July 2020 Accepted: 11 September 2020

ABSTRACT

Fuzzy clustering theory is widely used in data mining of full-face tunnel boring machine. However, the traditional fuzzy clustering algorithm based on objective function is difficult to effectively cluster functional data. We propose a new Fuzzy clustering algorithm, namely FCM-ANN algorithm. The algorithm replaces the clustering prototype of the FCM algorithm with the predicted value of the artificial neural network. This makes the algorithm not only satisfy the clustering based on the traditional similarity criterion, but also can effectively cluster the functional data. In this paper, we first use the t -test as an evaluation index and apply the FCM-ANN algorithm to the synthetic datasets for validity testing. Then the algorithm is applied to TBM operation data and combined with the cross-validation method to predict the tunneling speed. The predicted results are evaluated by $RMSE$ and R^2 . According to the experimental results on the synthetic datasets, we obtain the relationship among the membership threshold, the number of samples, the number of attributes and the noise. Accordingly, the datasets can be effectively adjusted. Applying the FCM-ANN algorithm to the TBM operation data can accurately predict the tunneling speed. The FCM-ANN algorithm has improved the traditional fuzzy clustering algorithm, which can be used not only for the prediction of tunneling speed of TBM but also for clustering or prediction of other functional data.

KEYWORDS

Data clustering; FCM; artificial neural network; functional data; TBM

1 Introduction

Cluster analysis belongs to unsupervised pattern recognition and is a multivariate statistical analysis method. It divides an initial sample set into several subsets according to a certain criterion, so as to achieve clustering of sample sets and analyze clustering results. Because in the actual engineering application, the research objects without clear classification boundaries occupy the main position, the fuzzy clustering is mainly used to cluster such objects, that is, the objects can belong to two or more categories at the same time [1,2]. In practical applications, the more general method is the fuzzy clustering method based on the objective function. This method transforms the cluster into a nonlinear programming problem with constraints,



and then obtains the fuzzy partitioning and the clustering of the datasets through optimization. In the clustering algorithm based on objective function, Fuzzy c-Means (FCM) algorithm has the most complete theory and the most widely used. The algorithm was originally proposed and developed by Bezdek and Hathaway [3–5]. It introduces the concept of membership degree based on the hard c-means (HCM) algorithm and implements fuzzy clustering by means of alternating optimization.

There have been many researches on FCM algorithms. Yu et al. [6] proposed the generalized fuzzy clustering regularization (GFCR) model based on various fuzzy clustering algorithms such as FCM algorithm, and verified the validity of the model. Zhang et al. [7] applied an improved weighted fuzzy c-means (WFCM) model and introduced an interval number, making it easier to obtain appropriate weights. Askari et al. [8] proposed a Generalized Entropy based Possibilistic Fuzzy C-Means (GEPFCM) algorithm for noise data clustering, which is more accurate than the Possibilistic Fuzzy C-Means (PFCM) algorithm. Li et al. [9] improved the FCM algorithm for the clustering problem of data with missing attribute values, and realized the clustering of incomplete datasets.

In the research of FCM type clustering algorithm, the research of clustering prototype has always been an important direction. The initial clustering prototype is a “point” in space, which is only suitable for the detection of hypersphere clustering structures. In order to detect the non-hypersphere clustering structure, Bzedek et al. [10] proposed a new clustering prototype, which is a multi-dimensional linear cluster over a certain point. In addition, according to different clustering structures, the researchers also developed a variety of clustering prototypes, such as the spherical shell [11,12] and the ellipsoid shell [13,14] two clustering prototypes. With the deepening of research and the expansion of application requirements, clustering prototypes have been extended to more forms. For example, Suh et al. [15] proposed a fuzzy clustering algorithm based on the polyhedral shell as a clustering prototype. Although these clustering prototypes can detect the detection of one or more clustering structures, they need to use prior knowledge to select prototypes before clustering. This does not enable efficient clustering when encountering functional data with high complexity and inconspicuous clustering structure.

Functional data refers to data obeying a function, and can also be regarded as random observation data of a function in an interval. With the development of data acquisition and storage capabilities, data can be collected in many fields with functional features. Functional data has been applied to many fields such as economics, medicine, meteorology and neuroscience [16–19]. Functional data clustering analysis is a research hotspot in recent years. Researchers have proposed a variety of clustering methods for different objects. From the data itself, Zambom et al. [20] proposed a new method based on functional data clustering combining parallel hypothesis testing and mean testing. Delaigle et al. [21] first project data onto a finite-dimensional space, and then use the K-means algorithm to achieve clustering. Bruckers et al. [22] used data interpolation with multiple imputations for functional data with missing values, and then clustered the imputed datasets.

Although the above method can achieve clustering of some functional data, it lacks universality in application. Because some of these methods require high accuracy of data, they cannot be implemented with high data noise. Others will ignore certain parameter information, resulting in low classification accuracy.

In the recent period, machine learning methods represented by deep learning have been applied to many fields, such as computer vision, natural language processing, and data mining.

Since deep learning has good feature learning capabilities, it is also used as an alternative method for engineering problems. Samaniego et al. [23] combined DNN in the configuration method and the deep energy method to provide a novel idea for solving partial differential equations. Guo et al. [24] applied a deep learning method to the thin plate bending problem, which is suitable for the bending analysis of Kirchhoff plates with various geometric shapes. Vien et al. [25] proposed a deep energy method for processing nonlinear large deformation hyperelasticity based on DNN, which can quickly and efficiently obtain numerical solutions. Anitescu et al. [26] proposed an adaptive configuration strategy using artificial neural networks to solve partial differential equations, and obtained the desired results in solving boundary value problems.

In the field of fuzzy clustering, there are also many studies that combine neural network methods. ANN simulates a biological neural network and builds a training model from multiple nodes to achieve regression and approximation of complex functions. Xu et al. [27] proposed a WLAN hybrid indoor positioning method based on FCM and ANN, which reduces the positioning error while ensuring efficiency. Karlik et al. [28] proposed a new fuzzy clustering neural network (FCNN) algorithm as a pattern classifier for real-time odor recognition systems. The FCNN algorithm uses FCM clustering to reduce the number of data points before inputting to the neural network system, thereby shortening the training cycle of the neural network. In addition, Mohd-Safar et al. [29] and Moradi et al. [30] also combined FCM and ANN, proposed different models, and applied them to engineering problems. However, in these methods, FCM and ANN are independent of each other, which will not only increase the computational burden but also fail to achieve effective clustering on complex problems such as the clustering of functional data. In addition, in these studies, FCM is used more as a data preprocessing method and serves ANN. The algorithm we proposed takes FCM as the overall framework and ANN as a way to describe the functional relationship. The structure achieves the integration of the two, and at the same time carries out end-to-end training, which can realize the effective clustering of complex functional data.

For the clustering problem of functional data, this paper proposes the FCM–ANN algorithm. It obtains an approximate functional model through the ANN training dataset. Then the function model is used as a new clustering prototype to replace the clustering prototype of the traditional FCM algorithm, and participates in the algorithm's alternating optimization. Finally, the algorithm will obtain the clustering result of the dataset and the corresponding functional data ANN model.

The rest of this paper is organized as follows. In Section 2, we first introduce the FCM algorithm and the artificial neural network algorithm, and then propose the FCM–ANN algorithm. Section 3 introduces the clustering results of applying the algorithm to synthetic dataset experiments and compares it with the traditional FCM algorithm. Section 4 introduces the experimental results of applying the FCM–ANN algorithm to the operation data of the tunnel boring machine (TBM) and compares it with the method of not performing classification modeling. In Section 5 we make some conclusions on this paper.

2 Proposed Algorithm (FCM–ANN)

2.1 Fuzzy *c*-Means Algorithm

The FCM algorithm is a fuzzy clustering algorithm based on objective function. For a given dataset $X = \{x_1, x_2, x_3, \dots, x_n\} \subset \mathbb{R}^s$, the FCM algorithm divides X into c ($2 \leq c \leq n$) clusters by

minimizing the clustering objective function. The clustering objective function is as follows:

$$J(U, P) = \sum_{i=1}^c \sum_{k=1}^n \mu_{ik}^m \|x_k - p_i\| \quad (1)$$

where $x_k = [x_{1k}, x_{2k}, x_{3k}, \dots, x_{sk}]^T$ is the target data, x_{jk} is the j -th attribute value of the data; p_i ($i = 1, 2, \dots, c$) represents the clustering prototype vector of the i -th cluster, and the clustering prototype matrix is represented as $P = [p_1, p_2, p_3, \dots, p_c] \in \mathbb{R}^{s \times c}$; μ_{ik} is the membership degree, which indicates the degree to which the data x_k belongs to the i -th cluster. For $\forall i, k$, there is $\mu_{ik} \in [0, 1]$, and it satisfies the following relationship:

$$\sum_{i=1}^c \mu_{ik} = 1 \quad \text{for } k = 1, 2, \dots, n \quad (2)$$

The partition matrix is expressed as $U = [\mu_{ik}] \in \mathbb{R}^{c \times n}$; m is a weighting parameter, also called a smoothing parameter, $m \in (1, \infty)$; $\|\cdot\|$ is the Euclidean distance in the s -dimensional space.

Combined with the constraint Eq. (2), the Lagrange multiplier method can be used to minimize the objective function. The final formula for the partition matrix and clustering prototype is as follows:

$$\mu_{ik} = \left[\sum_{l=1}^c \left(\frac{\|x_k - p_l\|}{\|x_k - p\|} \right)^{\frac{2}{m-1}} \right]^{-1} \quad \text{for } i = 1, 2, \dots, c \text{ and } k = 1, 2, \dots, n \quad (3)$$

$$p_i = \frac{1}{\sum_{k=1}^n \mu_{ik}^m} \sum_{k=1}^n \mu_{ik}^m x_k \quad \text{for } i = 1, 2, \dots, c \quad (4)$$

The specific steps of the FCM algorithm are as follows:

Step (i) Set the number of subclusters of fuzzy clustering c , for $2 \leq c \leq n$, n is the number of data of the target dataset; set the iteration stop threshold ε ; initialize the cluster prototype $P^{(0)}$; set the iteration counter $b = 0$.

Step (ii) Calculate (or update) the partition matrix $U^{(b)}$ using Eq. (3) and $P^{(b)}$.

Step (iii) Update clustering prototypes $P^{(b+1)}$ using Eq. (4) and $U^{(b)}$.

Step (iv) Calculate the discriminant Eq. (5):

$$\|P^{(b+1)} - P^{(b)}\| < \varepsilon \quad (5)$$

If the Eq. (5) is established, the algorithm stops and outputs the partition matrix U and the cluster prototype matrix P ; otherwise set $b = b + 1$ and re-execute Step (ii). In Eq. (5), $\|\cdot\|$ is a suitable matrix norm.

The algorithm can also start by initializing the fuzzy partition matrix U , then calculate (update) the clustering prototype matrix with the Eq. (4), and then update the fuzzy partition matrix with the Eq. (3), until the stopping criterion is met.

2.2 Artificial Neural Network (ANN)

As a hotspot in recent years, artificial neural network is a multi-disciplinary subject area with a wide range and depth. A neural network is a broad and interconnected network of adaptive simple units whose organization mimics the interactions of the biological nervous system with the real world. The most basic building block in a neural network is a neuron. Each neuron is connected to other neurons, and each neuron transmits information through signals.

In this paper, we use the BP (error BackPropagation) algorithm to construct a neural network, which is the most common neural network learning algorithm by far. For a given training dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ for $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}^l$, that is, the input data contains d attribute values, and the output data contains l attribute values. We abstract the algorithm model into a feedforward neural network structure as shown in Fig. 1.

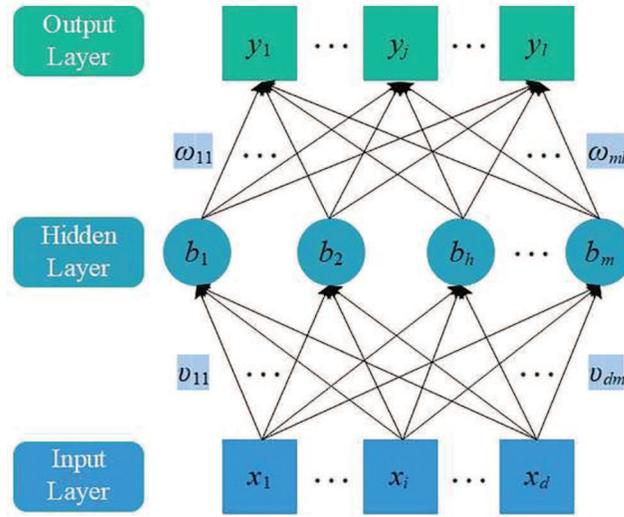


Figure 1: Feedforward neural networks with one hidden layer

The structure has d input neurons and l output neurons. The hidden layer neurons can be assumed to be m . The threshold of the j -th neuron in the output layer is represented by θ_j , and the threshold of the h -th neuron in the hidden layer is represented by γ_h . The connection weight between the h -th neuron in the hidden layer and the j -th neuron in the output layer is ω_{hj} , and the connection weight between the i -th neuron in the input layer and the h -th neuron in the hidden layer is v_{ih} .

For a d -dimensional input $\mathbf{x} = [x_1, x_2, x_3, \dots, x_d]^T \in \mathbb{R}^d$, the output value $y = [y_1, y_2, y_3, \dots, y_l]^T \in \mathbb{R}^l$ can be obtained according to the neural network structure. The final calculation result of y_j is as follows:

$$y_j = \sigma \left(\sum_{h=1}^m \omega_{hj} \cdot b_h - \theta_j \right) = \sigma \left[\sum_{h=1}^m \omega_{hj} \cdot \sigma \left(\sum_{i=1}^d v_{ih} \cdot x_i - \gamma_h \right) - \theta_j \right] \quad (6)$$

where σ is a sigmoid function, its equation is:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

2.3 FCM-ANN Algorithm

In view of the fact that the traditional FCM algorithm can not effectively solve the fuzzy clustering problem of functional data, we propose the FCM-ANN algorithm. In the FCM-ANN algorithm, we replace the clustering prototype in the traditional FCM algorithm with the neural network prediction value. This makes the clustering prototype change to conform to a certain data partitioning function, thus achieving accurate clustering of data.

For functional data, the iterative process of the FCM-ANN algorithm can be expressed as follows:

- Step (i) Set the number of subclusters of fuzzy clustering c , for $2 \leq c \leq n$, n is the number of data of the target data set; set the iteration stop threshold ε ; initialize the partition matrix $\mathbf{U}^{(0)}$; Set the membership threshold Ω ; set the iteration counter $b = 0$.
- Step (ii) The clustering result can be obtained according to the partition matrix $\mathbf{U}^{(b)}$. We use the clustering results as training data set to input artificial neural networks for training, and then we can obtain different neural network prediction models $ANN_i^{(b)}$, $i \in \{1, 2, \dots, c\}$.
- Step (iii) The prediction model is used for the independent variables $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n\} \subset \mathbb{R}^s$ in the dataset to obtain the corresponding predicted output $\hat{y}_{ik}^{(b)}$.
- Step (iv) Use the predicted output as a clustering prototype, the corresponding Euclidean distance $d_{ik}^{(b)}$ can be obtained according to the distance function:

$$d_{ik}^{(b)} = |y_k - \hat{y}_{ik}^{(b)}| \quad (8)$$

- Step (v) Update the partition matrix $\mathbf{U}^{(b+1)}$ by the Eq. (9).

$$\mu_{ik}^{(b+1)} = \left\{ \sum_{j=1}^c \left[\left(\frac{d_{ik}^{(b)}}{d_{jk}^{(b)}} \right)^{\frac{2}{m-1}} \right] \right\}^{-1} \quad (9)$$

- Step (vi) Calculate the discriminant:

$$\left\| \mathbf{U}^{(b+1)} - \mathbf{U}^{(b)} \right\| < \varepsilon \quad (10)$$

If the Eq. (10) is established, stop the algorithm and output the partition matrix \mathbf{U} and neural network prediction models ANN_i , $i \in \{1, 2, \dots, c\}$; otherwise set $b = b + 1$ and repeat Step (ii).

The FCM-ANN algorithm combines the advantages of both FCM and ANN. FCM makes the iteration always go in the direction of gradient descent. ANN provides accurate prediction results and can approximate any nonlinear function. In theory, the algorithm can achieve fuzzy clustering of arbitrary functional data.

Algorithm 1: FCM–ANN Algorithm**Require:** $X = \{x_1, x_2, x_3, \dots, x_n\} \subset \mathbb{R}^s, c, \varepsilon, \Omega$ **Ensure:** $U^{(0)}, b = 0$

```

1: for  $k = 1 \rightarrow n$  do
2:    $x'_k \leftarrow [x_{1k}, x_{2k}, \dots, x_{(s-1)k}]^T$ 
3:    $y_k \leftarrow x_{sk}$ 
4: end for
5: if  $\|U^{(b+1)} - U^{(b)}\| > \varepsilon$  then
6:   for  $i = 1 \rightarrow c$  do
7:     for  $k = 1 \rightarrow n$  do
8:       if  $\mu_{ik}^{(b)} > \Omega$  then
9:          $train_i.append(x'_k, y_k)$ 
10:      end if
11:    end for
12:  end for
13:  for  $i = 1 \rightarrow c$  do
14:     $ANN_i \leftarrow$  ANN train data sets  $train_i$ 
15:  end for
16:  for  $i = 1 \rightarrow c$  do
17:     $\hat{y}_{ik}^{(b)} \leftarrow ANN_i(x'_k)$ 
18:  end for
19:  for  $i = 1 \rightarrow c$  do
20:    for  $k = 1 \rightarrow n$  do
21:       $d_{ik}^{(b)} \leftarrow |y_k - \hat{y}_{ik}^{(b)}|$ 
22:    end for
23:  end for
24:  for  $i = 1 \rightarrow c$  do
25:    for  $k = 1 \rightarrow n$  do
26:       $\mu_{ik}^{(b+1)} = \left\{ \sum_{j=1}^c \left[ \left( \frac{d_{ik}^{(b)}}{d_{jk}^{(b)}} \right)^{\frac{2}{m-1}} \right] \right\}^{-1}$ 
27:    end for
28:  end for
29:   $b = b + 1$ 
30: end if
31: return  $U, ANN_1, \dots, ANN_i, \dots, ANN_c$ 

```

3 Experiments on Synthetic Datasets

3.1 Synthetic Datasets

In this section, we construct some synthetic datasets to study the validity of FCM–ANN algorithms and their effect on predictions with the different number of samples, number of attributes, and noise.

The synthetic datasets are created as follows. In each cluster of each dataset, the objects of i -th cluster except x_{obj} are first randomly sampled. The x_{obj} of each object is calculated according to the setted FRA. After that, the data of different cluster are combined as the obtained dataset. Each dataset is given a denomination by the number of object data, attributes, clusters and functional relationships among attributes. For instance, N400A2C2F1 denotes that the dataset contains 400 object data and can be divided evenly into two clusters, A2 denotes the dataset has two attributes. The clustering performance of the proposed clustering algorithm is compared with FCM.

Number of samples The synthetic datasets that discuss the effect of the number of samples size are divided into four groups. Each group of dataset consists of two clusters, each of which has 50, 100, 150, and 200 object data. And each cluster has two attributes, which is a functional relationship of $x_2 = f(x_1)$. The specific function relationship corresponding to the two types of data is as shown in Eqs. (11.1) and (11.2). At the same time, the noise is set to satisfy 3% of the standard functional relationship data and is subject to a Gaussian distribution.

Number of attributes The synthetic datasets that discuss the effect of the number of attributes are divided into five groups. Each dataset consists of two clusters, and the number of attributes of each cluster is 2–6, separately. They respectively satisfy the relationship as shown in [Tab. 1](#). Each dataset have 200 samples. The noise is also set to 3%, as well as the Gaussian distribution is satisfied.

Table 1: Functional equation corresponding to the number of attributes

Number of attributes	Cluster	Equation	
2	1	$x_2 = \sin^2(0.5x_1 - 6) \cdot (x_1/4)^2 + 10$	(11.1)
	2	$x'_2 = \sin^2(0.5x_1 - 6) \cdot (x_1/5)^2$	(11.2)
3	1	$x_3 = (4 + x_1 \sin x_1) [4 + \exp(-x_2^2)]$	(12.1)
	2	$x'_3 = (6 + x_1 \sin x_1) [4 + \exp(-x_2^2)]$	(12.2)
4	1	$x_4 = \sin(x_2 - x_1) + (x_3 - x_2)^2$	(13.1)
	2	$x'_4 = \cos(x_2 - x_1) + (x_3 - x_2)^2$	(13.2)
5	1	$x_5 = 1 + 2 \exp \left\{ -2 \left[(x_1 - 1)^2 \right] + x_2^2 - 0.5 (x_3^2 + x_4^2) \right\}$	(14.1)
	2	$x'_5 = 1 + \exp \left\{ -2 \left[(x_1 - 1)^2 \right] + x_2^2 - 0.5 (x_3^2 + x_4^2) \right\}$	(14.2)
6	1	$x'_6 = \sum_{i=1}^4 \left[(x_{i+1} - x_i)^2 + (x_i - 1)^2 \right]$	(15.1)
	2	$x'_6 = \sum_{i=1}^4 \left[(x_{i+1} - x_i)^2 + (x_i - 1)^2 \right] + 3$	(15.2)

Noise The synthetic datasets that discuss the effect of noise are divided into five groups. Each dataset consists of two clusters, each of which has 3–7% noise of the standard data. And the noise subject to Gaussian distribution. They have 200 samples per cluster and 6 attributes. The specific function relationships corresponding to the two clusters of dataset are shown in Eqs. (15.1) and (15.2).

3.2 Experimental Results

In order to test the clustering performance, we carried out a comparison experiment between the proposed algorithm and the FCM algorithm.

The clustering performance is evaluated in terms of misclassification (MS), which is calculated as follows:

$$MS = \frac{N_{error}}{N_{total}} \times 100\% \quad (16)$$

where N_{error} is the number of misclassified object data; N_{total} is the total number of object data.

We set the fuzzification parameter $m = 2$, convergence threshold $c = 10^{-4}$. The misclassification is discussed in each experiment by controlling the membership threshold. [Tabs. 2–4](#) present the average results over 30 experiments, with the optimal results being indicated in bold.

Table 2: Effect of the number of samples on algorithm prediction

		Number of samples	Misclassification (MS)/%					FCM
			$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$	
\bar{x}	100	31.80	29.87	26.07	23.1	26.77	42.00	
Std.		9.92	10.87	10.22	10.36	11.82	0	
\bar{x}	200	29.33	25.63	23.93	20.85	18.23	35.50	
Std.		11.10	8.41	12.38	12.22	12.91	0	
\bar{x}	300	28.10	25.03	22.89	18.40	17.70	39.33	
Std.		9.94	9.39	11.59	13.13	13.41	0	
\bar{x}	400	27.91	23.69	22.31	16.80	16.58	38.50	
Std.		10.52	7.45	6.80	7.66	8.06	0	

Table 3: Effect of the number of attributes on the algorithm prediction

		Number of attributes	Misclassification (MS)/%					FCM
			$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$	
\bar{x}	2	31.33	27.63	23.93	20.85	18.23	34.50	
Std.		11.10	8.41	12.38	12.22	9.91	0	
\bar{x}	3	31.78	28.17	26.38	22.42	20.65	46.00	
Std.		13.40	12.08	11.15	12.75	12.23	0	
\bar{x}	4	21.13	18.63	15.60	12.63	13.20	47.00	
Std.		15.06	16.94	16.46	11.45	11.65	0	
\bar{x}	5	21.78	17.12	16.15	15.28	13.68	40.00	
Std.		12.76	13.43	12.17	13.48	11.85	0	
\bar{x}	6	30.17	28.42	25.25	23.93	22.35	36.00	
Std.		10.42	9.62	12.00	13.50	14.94	0	

Table 4: Effect of the noise on the algorithm prediction

Noise		Misclassification (<i>MS</i>)/%					FCM
		$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$	
\bar{x}	3%	28.47	27.26	26.59	22.50	21.81	34.25
Std.		11.88	13.07	12.70	13.41	14.38	0
\bar{x}	4%	29.98	27.47	28.65	23.09	22.22	36.25
Std.		10.73	10.27	11.37	11.06	13.87	0
\bar{x}	5%	30.12	28.53	26.00	24.25	23.38	30.50
Std.		10.24	11.23	13.01	12.21	13.78	0
\bar{x}	6%	30.94	28.08	26.34	25.48	23.37	33.50
Std.		11.55	13.17	12.82	12.82	13.65	0
\bar{x}	7%	31.53	28.65	28.28	27.07	24.93	33.00
Std.		11.95	12.32	12.69	16.23	15.21	0

We use the t -test as the measure to analyze the convergence of the experimental results. The t -test is mainly used for a normal distribution with a small sample size and an unknown population standard deviation σ . The t -test uses the t -distribution theory to infer the probability of occurrence of the difference, thereby comparing whether the difference between the two means is significant. The t -test can be divided into the one-sample t -test, the two-sample t -test and the paired t -test. The paper uses the one-sample t -test method to test the results of clustering.

The one-sample t -test is used to test whether the difference between the average of a sample set and the known population mean is significant. When the distribution of the population shows a normal distribution, the sample size is small and the standard deviation of the population is unknown, the dispersion statistics of the sample mean and the population mean are t -distributed.

The statistic for the one-sample t -test is:

$$t = \frac{\bar{X} - \mu}{S/\sqrt{N}} \quad (17)$$

where N is the total number of samples, \bar{X} is the mean number of samples, μ is the mean of the population, and S is the sample standard deviation. The t statistic obeys the t -distribution with a degree of freedom of $(n - 1)$ when $\mu = \mu_0$.

The t -test was performed using the data in [Tabs. 2–4](#). The hypothesis is as follows:

$$H_0 : \mu \leq 25, H_1 : \mu > 25$$

Under the condition of the significance level $\alpha = 0.05$, the rejection region of the hypothesis is $t > t_{0.05}(n - 1) = t_{0.05}(29)$. Checking out the table of t -distribution to get the upper quantile $t_{0.05}(29) = 1.6991$.

This hypothesis is tested by calculating the statistic t using the [Eq. \(17\)](#). The results are listed in [Tab. 5](#), where the results over the upper quantile are indicated in bold.

In order to evaluate whether there is a significant difference between the results obtained by the FCM–ANN algorithm and the FCM algorithm, we also used the SPSS software to conduct a Wilcoxon rank-sum test. We compare the results in [Tabs. 2–4](#) with the results of the FCM

algorithm according to different membership thresholds and calculate their rank-sum. The results of the rank-sum test are shown in Tab. 6. Given the significance level $\alpha = 0.05$, we can obtain the critical value of the rank sum of $166 \sim 240$. According to the calculated rank-sum, we can judge whether the difference between the results of the two methods is significant.

Table 5: The results of test statistics

Number of samples	$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$
100	3.6914	2.4127	0.5638	-0.9876	0.8064
200	2.1007	0.4034	-0.4654	-1.8288	-2.8240
300	1.6795	0.0002	-0.9804	-2.7069	-2.9315
400	1.4896	-0.9469	-2.1303	-5.7648	-5.6257
Number of attributes	$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$
2	3.0710	1.6841	-0.4654	-1.8288	-3.6789
3	2.7247	1.4132	0.6665	-1.0897	-1.9154
4	-1.3838	-2.0250	-3.0753	-5.8179	-5.4545
5	-1.3590	-3.1597	-3.9161	-3.8831	-5.1443
6	2.6719	1.9145	0.1122	-0.4268	-0.9552
Noise	$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$
3%	1.5729	0.9312	0.6742	-1.0040	-1.1946
4%	2.4994	1.2952	1.7287	-0.9300	-1.0794
5%	2.6926	1.6928	0.4139	-0.3308	-0.6331
6%	2.7695	1.2594	0.5629	0.2016	-0.6431
7%	2.9427	1.5954	1.3919	0.6868	-0.0248

Table 6: The result of the Wilcoxon rank-sum test

Rank sum	$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$
T	110	105	105	105	105

3.3 Discussion

1) By analyzing Tabs. 2–4, the mean of misclassifications obtained by using the FCM–ANN algorithm is significantly lower than that of the FCM algorithm, except for the experimental results corresponding to 3% noise in Tab. 4. In addition, from the Wilcoxon rank-sum test results in Tab. 6, it can be seen that the results obtained by the FCM–ANN algorithm are significantly different from those obtained by the FCM algorithm. Therefore, we can conclude that the clustering result of the FCM–ANN algorithm is better than the FCM algorithm. At the same time, according to the results in Tabs. 2–4, as the membership threshold increases, the misclassifications show a downward trend. The results show that the higher the membership threshold, the more the invalid data can be avoided to participate in the artificial neural network training, so the fitting result is more accurate and the prediction effect is better. However, as the membership threshold increases, the standard deviation of the misclassification increases, which results in a decrease in the stability of the results.

2) According to the results of [Tab. 2](#), in the case where the membership threshold is the same, the misclassifications will decrease as the number of samples increases. According to the results, as the number of samples increases, the training dataset provided by the experiment will also increase, which makes the prediction effect of the algorithm more accurate.

3) According to the vertical comparison [Tab. 3](#), the clustering results of the dataset are better when the different number of attributes are taken. Therefore, it can be concluded that the FCM–ANN algorithm can achieve effective clustering of functional data with the different number of attributes. This also shows that the algorithm has favorable stability.

4) According to the results of [Tab. 4](#), when the noise increases, the misclassifications of the experiment using the FCM algorithm has no significant change, which indicates that the increase of noise will have less influence on the clustering result of the FCM algorithm. However, in the test results using the FCM–ANN algorithm, misclassifications will increase as the noise increases. Therefore, improving the accuracy of datasets and the reduction of noise can improve the clustering effect of the FCM–ANN algorithm.

5) According to [Tab. 5](#), most of the results of the test are below the upper quantile. Therefore, it can be considered that the hypothesis H_0 is valid, and the FCM–ANN algorithm can effectively implement the clustering of functional data. However, the results in the reject domain appear when the membership threshold is small. Therefore, appropriately increasing the membership threshold can make the clustering result more accurate.

3.4 Computation Cost and Convergence Analysis

3.4.1 Computation Cost

We use the average of running time to measure the computation cost of the algorithm. The experimental environment is intel(R) Core (TM) i5-9300H 2.40 GHz CPU, 8 GB RAM, Windows 10 operating system, Python 3.7.0. [Tab. 7](#) show comparisons of running times for experiments on synthetic datasets.

According to the results in [Tab. 7](#), although the FCM–ANN algorithm runs less time than the FCM algorithm, the results are still within acceptable limits. Therefore, the FCM–ANN algorithm can be considered to have a lower computation cost. At the same time, in the results, the running time will increase with the increase of the membership threshold, so the computation cost should be considered to set a reasonable membership threshold.

3.4.2 Convergence Analysis

Then, we concentrate on the convergence analysis of FCM–ANN algorithm. In the Synthetic data sets, we selected data sets with the number of attributes of 2–5 for testing, as shown in [Fig. 2](#).

It can be seen that in the case of different number of attributes, the objective function value declines rapidly in the first 6 iterations and then tends to converge gradually. Therefore, it can be further proved that the proposed algorithm is effective.

4 Engineering Application on TBM Operation Data

4.1 Project Review

The tunnel project studied in this paper is located on a metro line in China, with a length of 2,000 meters and a diameter of 6.4 meters. The project adopted earth pressure balance (EPB) shield TBM. This system consists of a cutterhead, chamber, screw conveyor, tail skin and other

Table 7: Comparison of running times

Number of attributes	Running time/s					
	$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$	FCM
100	0.3271	0.3989	0.5709	0.5559	0.5527	0.0369
200	0.4092	0.4252	0.6779	0.6681	0.7563	0.1057
300	0.6002	0.5838	0.5714	0.4235	0.9569	0.0768
400	0.7380	0.8997	1.0559	1.1190	1.2350	0.1127
Number of attributes	Running time/s					
	$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$	FCM
2	0.4092	1.4252	1.6778	2.6681	2.7563	0.0439
3	0.7149	0.9637	0.7444	0.9320	1.6733	0.0469
4	0.8643	1.2287	1.2804	1.0578	1.1262	0.7982
5	1.1038	1.1927	1.2482	1.6605	1.7108	0.2075
5	0.9758	1.2253	1.4003	1.8045	2.3908	0.2558
Noise	Running time/s					
	$U > 0.5$	$U > 0.6$	$U > 0.7$	$U > 0.8$	$U > 0.9$	FCM
3%	1.4720	1.6738	1.6185	2.3198	2.0802	0.2922
4%	1.5481	2.2992	2.7570	2.9883	2.8625	0.2666
5%	1.3907	2.4404	2.6289	3.4468	3.7464	0.3583
6%	1.2716	1.8903	2.2232	2.9157	3.0527	0.2733
7%	1.5146	1.3511	2.3948	2.6059	2.8353	0.3480

auxiliary. The operation data adopted in this paper are collected from TBM construction projects, and there are 52 parameters, among which each parameter has different correlation with the tunneling speed. The statistical chart of the datasets used in this paper is listed in Appendix A.

4.2 Experiments and Results

The experiment is divided into two parts. Experiment 1 uses the neural network algorithm to directly train the TBM operation data to establish the prediction model of predicting the tunneling speed. Experiment 2 uses the FCM-ANN algorithm to cluster the TBM operation data firstly, aiming to obtain clustering results and prediction models for each cluster, and then use the model to predict each cluster of data. The two experimental results are compared to verify the practicability of the algorithm.

The same 1200 sets of TBM operation data were used in both experiments. We divide the datasets into three parts: Training set, validation set, and test set. Among them, the test set accounts for 20%, and the remaining 80% of the data is divided into training sets and validation sets. In addition, in order to prevent over-fitting problems, we perform 5-fold cross-validation on the training set and the validation set. The data are randomly divided into 5 equal parts, and each of the 5 equal parts is used as a separate test set, and the remaining 4 parts are used as the training set for building the model for validity verification. We use mean square error (MSE) as an indicator to evaluate the prediction accuracy of the validation set, which is calculated

by Eq. (18).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (18)$$

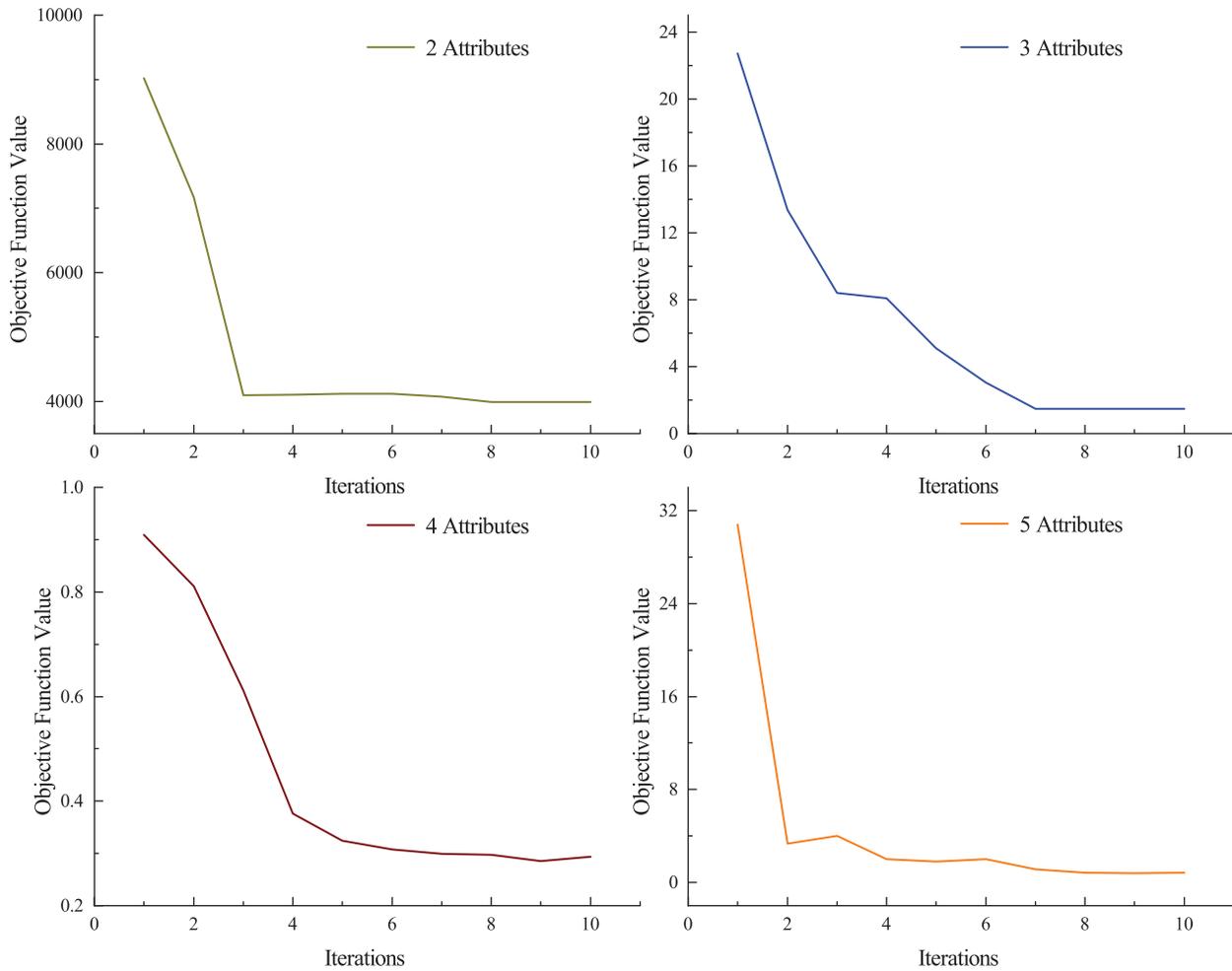


Figure 2: Objective function value in the iterative procedure

1) The steps of Experiment 1 are as follows:

- ① Use the neural network algorithm to train the training set and the validation set, and adopt the cross-validation method to obtain multiple neural network prediction models.
- ② Use MSE as an indicator to evaluate the predictive effect of each model, and select the optimal model.

- ③ Apply the optimal model to the test set for prediction to obtain prediction results of the tunneling speed.
- ④ Compare the prediction results of the tunneling speed with the target values to evaluate the prediction results of the neural network algorithm.

2) The steps of Experiment 2 are as follows:

- ① Use the FCM-ANN algorithm to train the training set, so as to obtain the clustering results of the training set and their corresponding neural network prediction models.
- ② Use the FCM-ANN algorithm to train the validation set, then the clustering results of the validation set can be obtained.
- ③ According to the clustering results of the validation set, the neural network prediction models obtained by the training set is used to predict each cluster of corresponding data, then the prediction results of the validation set can be obtained.
- ④ Use the cross-validation method to repeat Steps 1–3, and use MSE as an indicator to evaluate the effect of each prediction, and select the optimal model.
- ⑤ Use the FCM-ANN algorithm to train the test set, then the clustering results of the test set can be obtained.
- ⑥ According to the clustering results of the test set, the optimal neural network prediction models are used to predict each cluster of corresponding data, then the prediction results of the tunneling speed can be obtained.

Tab. 8 shows the comparison results of the MSE of the validation set obtained through cross-validation during the two experiments. According to the above steps, Experiment 1 and Experiment 2 are executed and we obtain 240 sets of data were respectively. Compare this with the target values and the results are shown in Figs. 3 and 4.

Table 8: Comparison of Validation MSE

Experiments	Validation MSE				
	1	2	3	4	5
1	319.22	255.40	230.16	355.52	193.76
	43.77	20.16	26.24	31.43	42.94

As can be seen from Fig. 3, the differences between the target values and the predicted values of the tunneling speed are obvious. This indicates that the neural network algorithm cannot accurately predict the tunneling speed without the cluster. But in Fig. 4, the target values of the tunneling speed show a good correlation with the predicted values.

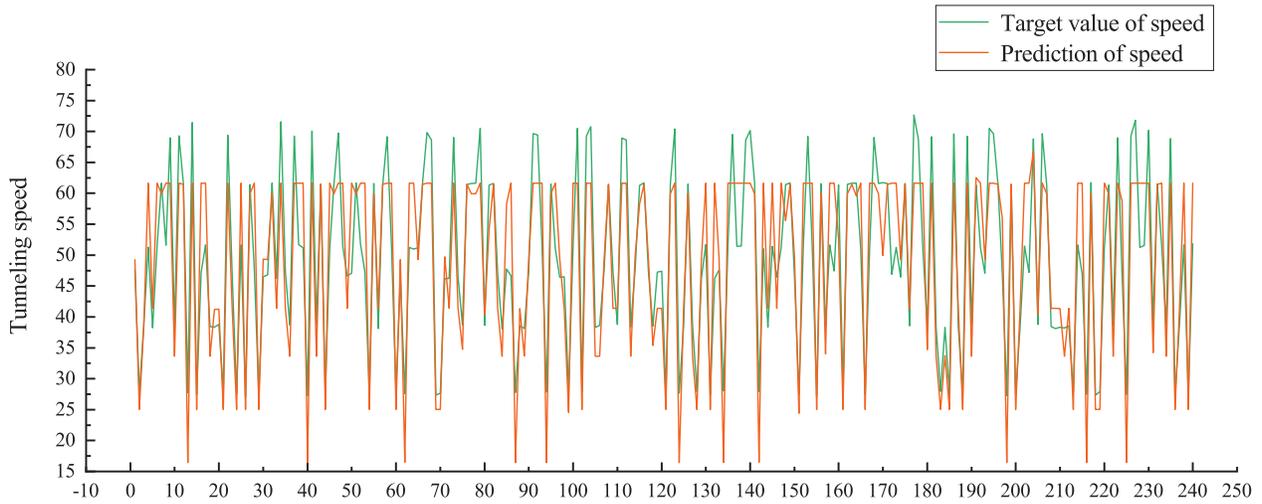


Figure 3: Correlation between predicted and target values (Experiment 1)

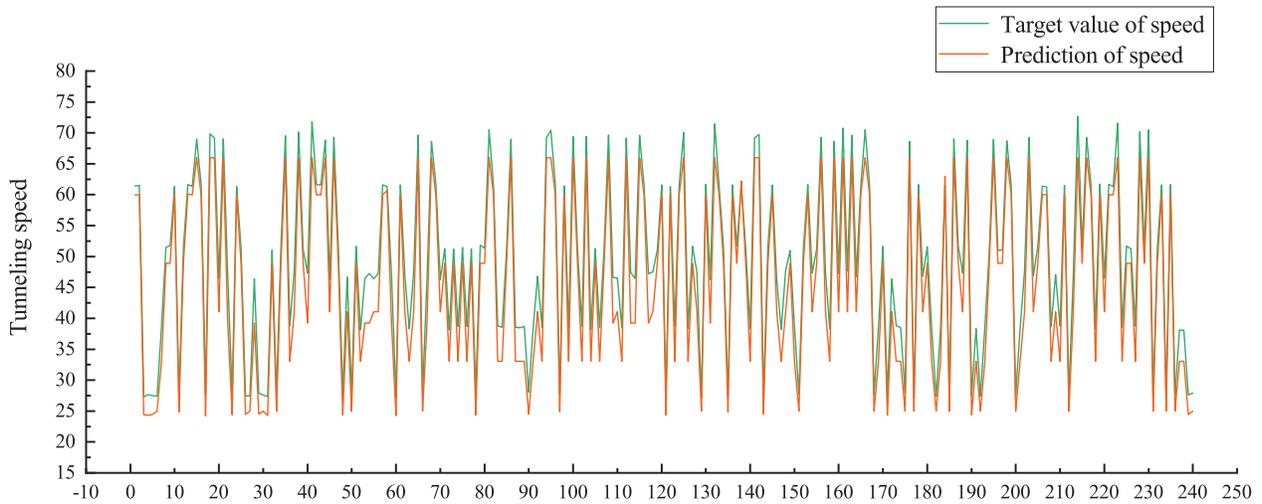


Figure 4: Correlation between predicted and target values (Experiment 2)

4.3 Comparison of Results

In this section, we use root mean square error (*RMSE*) and coefficient of determination R^2 as statistical criteria to evaluate the performance of the algorithm. The root mean square error (*RMSE*) is a measure of goodness-of-fit that best describes the average measure of error when predicting a dependent variable. For a set of data $y = (y_1, y_2, \dots, y_n)$, the result of regression prediction is $\hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$. The *RMSE* can be expressed as the form shown in Eq. (19).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (19)$$

The coefficient of determination R^2 is also a measure of the goodness-of-fit. It can be used to test the error of the predicted dependent and target values at the variable level. Before describing the goodness-of-fit, several additional indicators, namely the population squared TSS , the regression squared ESS , and the residual squared sum RSS , need to be introduced.

$$TSS = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (20)$$

$$ESS = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 \quad (21)$$

$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (22)$$

where $\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$. The equation for calculating the R^2 is:

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS} \quad (23)$$

The value range of R^2 is [0,1]. The closer R^2 is to 1, the higher the goodness-of-fit.

In the prediction experiment of the tunneling speed, 8 independent experiments were performed, and the $RMSE$ and R^2 indicators were calculated for each experiment. The results are shown in Figs. 5 and 6.

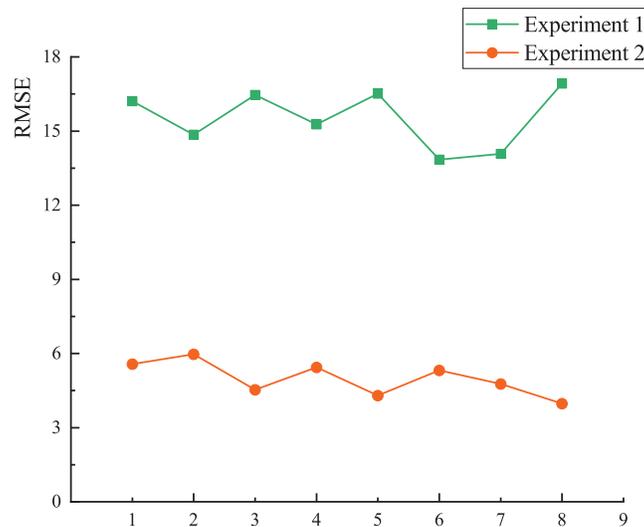


Figure 5: $RMSE$ indicator evaluation results

The results of Experiment 1 and Experiment 2 can be compared by the above figure. As can be seen from Fig. 5, most of the $RMSE$ values of Experiment 1 are much higher than Experiment 2, and the results of Experiment 1 varied so much that the maximum value reached 16.92. However, the results of Experiment 2 remain stable between 3.5 and 6.5. As

can be seen from Fig. 6, the R^2 of Experiment 1 vary so much that some negative values appear. But the R^2 of Experiment 2 are always maintained above 0.9. Therefore, according to the results of RMSE and R^2 , it can be concluded that the FCM-ANN algorithm can accurately predict the tunneling speed of TBM, and the effect is obviously better than the neural network algorithm.

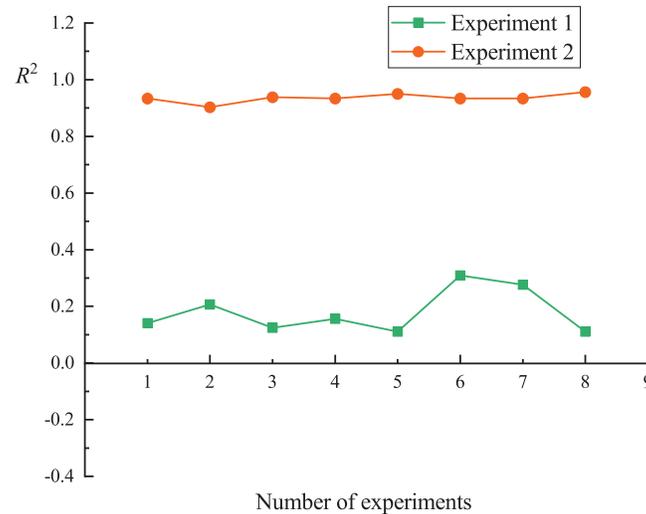


Figure 6: R^2 indicator evaluation results

5 Conclusions

In this paper, we propose the FCM-ANN algorithm for functional data that is difficult to cluster effectively in traditional methods. The FCM-ANN algorithm is based on the FCM algorithm and uses the predicted value of artificial neural network as the clustering prototype to perform the iterative update of the algorithm. We first apply the algorithm to the synthetic datasets and discuss the effects of different number of samples, different number of attributes, and different noise on the clustering results under different membership thresholds. Then the algorithm is applied to the TBM operation data and compared with the method of modeling without clustering. The results show that the FCM-ANN algorithm can accurately and effectively predict tunneling speed. The future work will be mainly focused on replacing existing ANN models with more sophisticated neural network models while improving algorithm theory to make the iterative process more complete.

Funding Statement: This research is supported by the National Key R&D Program of China (Grant Nos. 2018YFB1700704 and 2018YFB1702502) and the Study on the Key Management and Privacy Preservation in VANET, The Innovation Foundation of Science and Technology of Dalian (2018J12GX045).

Conflicts of Interest: We declare that we have no conflicts of interest to report regarding the present research.

References

1. Jain, A. K., Murty, M. N., Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3), 264–323. DOI 10.1145/331499.331504.
2. Hamdia, K. M., Arafa, M., Alqedra, M. (2018). Structural damage assessment criteria for reinforced concrete buildings by using a fuzzy analytic hierarchy process. *Underground Space*, 3(3), 243–249. DOI 10.1016/j.undsp.2018.04.002.
3. Bezdek, J. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York, NY, USA: Plenum.
4. Hathaway, R. J., Bezdek, J. C. (1995). Optimization of clustering criteria by reformulation. *IEEE Transactions on Fuzzy Systems*, 3(2), 241–245. DOI 10.1109/91.388178.
5. Hathaway, R. J., Bezdek, J. C., Hu, Y. K. (2000). Generalized fuzzy c-means clustering strategies using lp norm distances. *IEEE Transactions on Fuzzy Systems*, 8(5), 576–582. DOI 10.1109/91.873580.
6. Yu, J., Yang, M. S. (2007). A generalized fuzzy clustering regularization model with optimality tests and model complexity analysis. *IEEE Transactions on Fuzzy Systems*, 15(5), 904–915. DOI 10.1109/TFUZZ.2006.889957.
7. Zhang, L., Pedrycz, W., Lu, W., Liu, X., Zhang, L. (2014). An interval weighed fuzzy c-means clustering by genetically guided alternating optimization. *Expert Systems with Applications*, 41(13), 5960–5971. DOI 10.1016/j.eswa.2014.03.042.
8. Askari, S., Montazerin, N., Zarandi, M. H. F., Hakimi, E. (2017). Generalized entropy based possibilistic fuzzy c-means for clustering noisy data and its convergence proof. *Neurocomputing*, 219, 186–202. DOI 10.1016/j.neucom.2016.09.025.
9. Li, T., Zhang, L., Lu, W., Hou, H., Liu, X. et al. (2017). Interval kernel fuzzy c-means clustering of incomplete data. *Neurocomputing*, 237, 316–331. DOI 10.1016/j.neucom.2017.01.017.
10. Bezdek, J. C., Anderson, I. M. (1985). An application of the c-varieties clustering algorithms to polygonal curve fitting. *IEEE Transactions on Systems Man and Cybernetics*, 15(5), 637–641. DOI 10.1109/TSMC.1985.6313440.
11. Dave, R. N. (1990). Fuzzy shell-clustering and application to circle detection in digital images. *International Journal of General Systems*, 16(4), 343–355. DOI 10.1080/03081079008935087.
12. Mahdipour, H. H. A., Khademi, M., Sadoghi, H. Y. (2012). Model-based fuzzy c-shells clustering. *Neural Computing & Applications*, 21(S1), S29–S41. DOI 10.1007/s00521-011-0571-0.
13. Dave, R. N. (1992). Generalized fuzzy c-shells clustering and detection of circular and elliptic boundaries. *Pattern Recognition*, 25(7), 713–721. DOI 10.1016/0031-3203(92)90134-5.
14. Szilagy, L., Varga, Z. R., Szilagy, S. M. (2014). Application of the fuzzy-possibilistic product partition in elliptic shell Clustering. In: Torra, V., Narukawa, Y., Endo, Y., (Eds.), *Modeling decisions for artificial intelligence, Mdaï 2014*, vol. 8825, pp. 158–169. Lecture Notes in Artificial Intelligence.
15. Suh, I., Kim, J., Rhee, F. (1996). Fuzzy clustering involving convex polytopes. *5th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 96)*, September 08–11, New Orleans, LA.
16. Aue, A., Horvath, L., Pellatt, D. F. (2017). Functional generalized autoregressive conditional heteroskedasticity. *Journal of Time Series Analysis*, 38(1), 3–21. DOI 10.1111/jtsa.12192.
17. Shah, D. A., De Wolf, E. D., Paul, P. A., Madden, L. V. (2019). Functional data analysis of weather variables linked to fusarium head blight epidemics in the united states. *Phytopathology*, 109(1), 96–110. DOI 10.1094/PHYTO-11-17-0386-R.
18. Curceac, S., Ternynck, C., Ouarda, T. B. M. J., Chebana, F., Niang, S. D. (2019). Short-term air temperature forecasting using nonparametric functional data analysis and sarma models. *Environmental Modelling & Software*, 111, 394–408. DOI 10.1016/j.envsoft.2018.09.017.
19. Zipunnikov, V., Caffo, B., Yousem, D. M., Davatzikos, C., Schwartz, B. S. et al. (2011). Functional principal component model for high-dimensional brain imaging. *Neuroimage*, 58(3), 772–784. DOI 10.1016/j.neuroimage.2011.05.085.
20. Zambom, A. Z., Collazos, J. A. A., Dias, R. (2019). Functional data clustering via hypothesis testing k-means. *Computational Statistics*, 34(2), 527–549. DOI 10.1007/s00180-018-0808-9.

21. Delaigle, A., Hall, P., Tung, P. (2019). Clustering functional data into groups by using projections. *Journal of the Royal Statistical Society Series B—Statistical Methodology*, 81(2), 271–304. DOI 10.1111/rssb.12310.
22. Bruckers, L., Molenberghs, G., Dendale, P. (2017). Clustering multiply imputed multivariate high-dimensional longitudinal profiles. *Biometrical Journal*, 59(5), 998–1015. DOI 10.1002/bimj.201500027.
23. Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V. M., Guo, H. et al. (2020). An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 362(SI), 112790.1–112790.29.
24. Guo, H., Zhuang, X., Rabczuk, T. (2019). A deep collocation method for the bending analysis of kirchhoff plate. *Computers, Materials & Continua*, 59(2), 433–456. DOI 10.32604/cmc.2019.06660.
25. Nguyen-Thanh, V. M., Zhuang, X., Rabczuk, T. (2020). A deep energy method for finite deformation hyperelasticity. *European Journal of Mechanics A—Solids*, 80, 10384.
26. Anitescu, C., Atroshchenko, E., Alajlan, N., Rabczuk, T. (2019). Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials & Continua*, 59(1), 345–359. DOI 10.32604/cmc.2019.06641.
27. Yubin, X., Mu, Z., Lin, M. (2009). Hybrid FCM/ANN indoor location method in WLAN environment. *IEEE Youth Conference on Information, Computing and Telecommunication*, Beijing: Grad Univ.
28. Karlik, B., Yuksek, K. (2007). Fuzzy clustering neural networks for real-time odor recognition system. *Journal of Automated Methods & Management in Chemistry*, 2007(12), 38405.
29. Mohd-Safar, N. Z., Ndzi, D., Sanders, D., Noor, H. M., Kamarudin, L. M. (2018). Integration of fuzzy c-means and artificial neural network for short-term localized rainfall forecast in tropical climate. lecture notes in networks and systems. *SAI Annual Conference on Areas of Intelligent Systems and Artificial Intelligence and their Applications to the Real World (IntelliSys)*, September 21–22, Vol. 16. London, England.
30. Moradi, A. M., Dariane, A. B. (2017). Evolving neural networks and fuzzy clustering for multireservoir operations. *Neural Computing & Applications*, 28(5), 1149–1162. DOI 10.1007/s00521-015-2130-6.

Appendix A: Statistics of TBM Operation Data

Parameter	(X ₁)	(X ₂)	(X ₃)	(X ₄)	(X ₅)	(X ₆)	(X ₇)	(X ₈)	(X ₉)	(X ₁₀)
Min.	712.59	28.38	28.72	13.98	71.90	148.52	20.56	34.48	31.18	58.02
1st quartile	1242.31	33.40	29.74	212.00	107.84	171.89	115.02	64.93	49.24	114.25
Mean	1702.14	37.74	33.41	219.24	111.85	190.18	130.40	93.17	58.88	133.77
3rd quartile	2675.15	41.16	36.58	255.49	135.82	235.63	142.17	97.95	97.95	155.89
Max.	3058.41	51.39	38.10	322.31	197.64	290.47	160.32	115.13	119.14	202.20
Parameter	(X ₁₁)	(X ₁₂)	(X ₁₃)	(X ₁₄)	(X ₁₅)	(X ₁₆)	(X ₁₇)	(X ₁₈)	(X ₁₉)	(X ₂₀)
Min.	1.32	1.70	1.63	0.00	1.74	2.13	1.59	2.16	0.00	7.12
1st quartile	3.20	2.75	3.49	0.00	4.35	4.39	4.15	4.40	7.67	72.59
Mean	5.33	6.18	5.77	4.27	5.74	6.84	12.57	5.57	11.94	85.71
3rd quartile	14.97	14.50	14.99	12.17	14.62	14.82	15.63	15.45	16.15	112.34
Max.	57.19	57.29	65.54	63.21	46.43	54.31	68.20	57.57	21.61	171.73

Appendix A: (continued)

Parameter	(X_{21})	(X_{22})	(X_{23})	(X_{24})	(X_{25})	(X_{26})	(X_{27})	(X_{28})	(X_{29})	(X_{30})	
Min.	-4.31	0.00	1.47	1.30	1.59	1.51		1.37	0.22	30.53	0.00
1st quartile	-4.09	0.02	1.68	1.48	1.76	1.67		1.60	2.15	39.84	0.00
Mean	-4.03	0.04	1.92	1.54	2.21	2.10		1.70	2.57	41.20	0.00
3rd quartile	-3.97	0.10	2.00	1.61	2.29	2.17		1.78	3.53	43.00	1.48
Max.	-3.84	0.14	2.25	1.80	2.53	2.34		2.04	6.39	51.24	5.11
Parameter	(X_{31})	(X_{32})	(X_{33})	(X_{34})	(X_{35})	(X_{36})	(X_{37})	(X_{38})	(X_{39})	(X_{40})	
Min.	0.00	0.00	0.00	1.22	1.44	1.69		1.15	0.52	1.20	0.00
1st quartile	0.00	0.00	0.00	3.18	3.97	3.82		3.26	2.28	1.37	18.39
Mean	0.00	0.00	0.00	12.47	5.70	5.36		5.17	2.70	1.70	25.99
3rd quartile	0.94	0.00	0.00	18.35	15.02	15.09		15.01	3.67	1.80	31.85
Max.	5.12	0.00	1.49	74.67	61.05	48.54		43.46	5.89	2.04	53.75
Parameter	(X_{41})	(X_{42})	(X_{43})	(X_{44})	(X_{45})	(X_{46})	(X_{47})	(X_{48})	(X_{49})	(X_{50})	
Min.	0.00	314.98	335.83	306.87	291.67	0.36		22.79	4.53	0.19	10099.52
1st quartile	3.00	685.91	720.30	702.94	683.34	8.21		35.54	23.89	0.20	14692.63
Mean	3.00	1086.51	1073.90	1040.60	1050.74	10.17		49.59	45.41	7.37	15640.34
3rd quartile	3.00	1334.75	1383.33	1368.67	1334.47	19.48		64.42	48.42	11.75	16340.86
Max.	3.00	1805.12	1828.83	1795.89	1778.86	21.59		66.08	51.73	12.76	16967.07
Parameter	(X_{51})	(X_{52})									
Min.	0.50	0.55									
1st quartile	0.64	33.67									
Mean	1.90	48.02									
3rd quartile	2.06	60.31									
Max.	2.12	72.64									