Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06530-5

ORIGINAL ARTICLE

Incremental learning model based on an improved CKS-PFNN for aluminium electrolysis manufacturing

Wei Ding[1] · Lizhong Yao[1,2] · Yanyan Li[1] · Wei Long[1] · Jun Yi[2]

Received: 2 August 2020 / Accepted: 8 September 2021

Abstract

Filtering neural networks (FNNs) are popular computing frameworks for process system modeling. However, they are vulnerable to non-Gaussian noise and consequently may suffer from low filtering accuracy. To overcome the problem, in this paper, a novel model construction algorithm by combining the improved clustering kernel function smoothing technique and the particle filter neural network (ICKS-PFNN) is proposed. Specifically, ICKS-PFNN firstly presents a construction framework for particle filter neural network (PFNN), which utilizes the dynamic approximation of particles to adjust the NN's weights and thresholds in real time. Then, the proposed model uses kernel fuzzy C-means algorithm to uncover clusters in the particles of PFNN. A novel proportional distribution sampling strategy is adopted to maintain the diversity in particle clusters, through merging the inferior and superior particles to generate new particles based on the set proportional factors, rather than directly eliminating particles. At last, the estimation of the PFNN model is achieved by utilizing a kernel function smoothing method to update the particles in each cluster. The proposed model has been tested on the real-world system for aluminium electrolysis manufacturing and compared with several closely related frameworks. The experimental results show ICKS-PFNN obtains a superb performance when compared with other baselines. ICKS-PFNN is able to tackle noise and improve the prediction accuracy when dealing with non-Gaussian systems. Successfully applying the proposed framework in aluminium electrolysis manufacturing broadens the practical impact of FNN systems.

Keywords Aluminium electrolysis · Particle filter · Proportional sampling · Kernel fuzzy C-means

Abbreviations

NN	Neural network
PF	Particle filter
EKF	Extended Kalman filter
UKF	Unscented Kalman filter
FNN	Filtering neural network
FCM	Fuzzy C-mean clustering
MLR	Multiple linear regression
SSE	Sum squared error
MAE	Mean absolute error
RMSE	Root mean squared error
AEM	Aluminium electrolysis manufacturing
ICKS	Improved clustering kernel function smoothing
CKS-PFNN	Particle filter neural network model based on the clustering kernel function smoothing method
ICS-PFNN	Particle filter neural network model based on the improved clustering smoothing method
ICKS-PFNN	Particle filter neural network model based on the improved clustering kernel function smoothing method
EKFNN	Extended Kalman filter neural network
UKFNN	Unscented Kalman filter neural network
BPNN	Back-propagation neural network

✉ Lizhong Yao
cqust-ylz@cqust.edu.cn

Wei Ding
dingwei1995@stu.scu.edu.cn

Yanyan Li
liyanyan@scu.edu.cn

Wei Long
long_wei@scu.edu.cn

Jun Yi
2010027@cqust.edu.cn

[1] School of Mechanical Engineering, Sichuan University, Chengdu 610065, People's Republic of China

[2] School of Electrical Engineering, Chongqing University of Science and Technology, Chongqing 401331, People's Republic of China

Published online: 23 September 2021

PFNN	Particle filter neural network
PDF	Probability density function
KFCM	Fuzzy kernel C-mean clustering
NLMR	Multiple nonlinear regression
MSE	Mean squared error
MRE	Mean relative error
R	Correlation coefficient

1 Introduction

Aluminium products have been widely used in various links including infrastructure constructions and core industries due to their good physical properties, stable chemical properties and mature manufacturing processes [1]. At present, most enterprises use the electrolytic method to smelt aluminium at home and abroad. After long-term development and improvement, the scale and technology of the aluminium electrolysis industries have become more and more comprehensive.

The physical and structural diagram of aluminium electrolysis process equipment is shown in Figs. 1 and 2, respectively. It can be seen from Fig. 2 that the large pre-baking aluminium electrolysis cell is generally divided into anode area, cathode area, sidewall part and guide bar bus. In addition, Fig. 3 vividly shows the basic process of aluminium electrolysis manufacturing. In this process, the large-scale pre-baking aluminium electrolysis cell in Fig. 2 is used as the main production equipment. The anode carbon rod and aluminium oxide are used as the raw materials to form the electrolytic system together with the cryolite and fluoride salt [2]. The catholyte is transported to the foundry for casting. The anode exhaust gas is vented after purification treatment. The residual anode is sent back to the production line for reprocessing. The production monitoring platform carries out real-time information exchange with the aluminium electrolysis cell to ensure the smooth operation of the aluminium electrolysis manufacturing [3]. Therefore, the real-world aluminium electrolysis

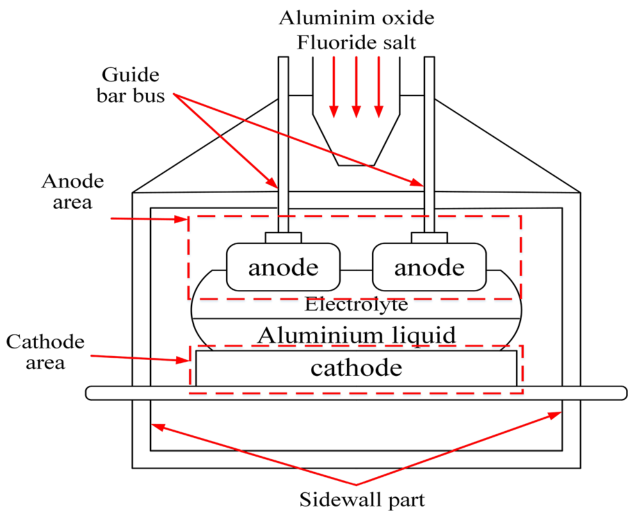


Fig. 2 The structural diagram of large pre-baking aluminium electrolysis cell

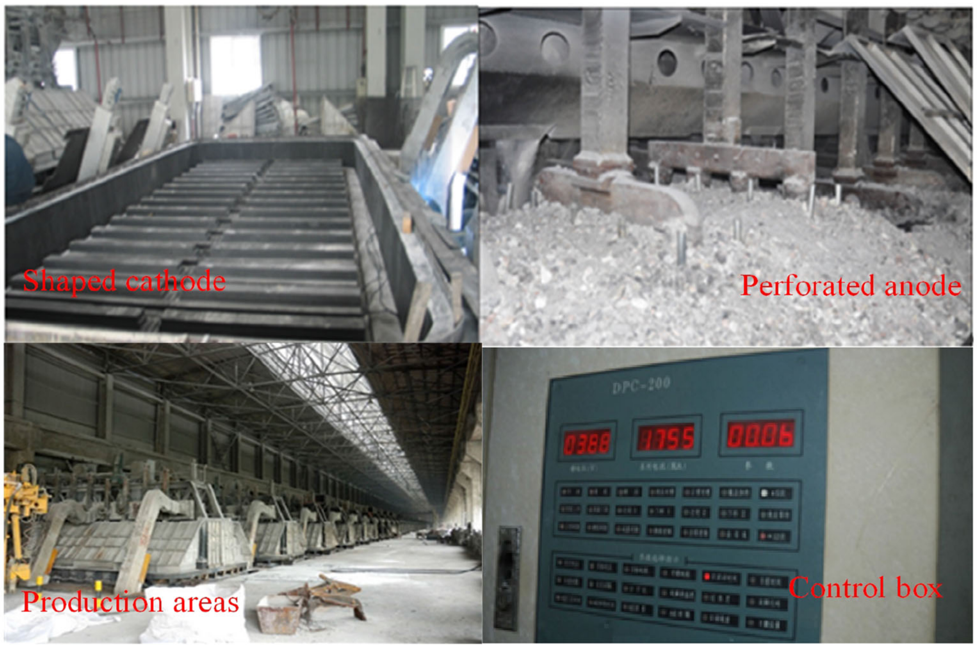


Fig. 1 The physical diagram of large pre-baking electrolytic equipment in real-world factory

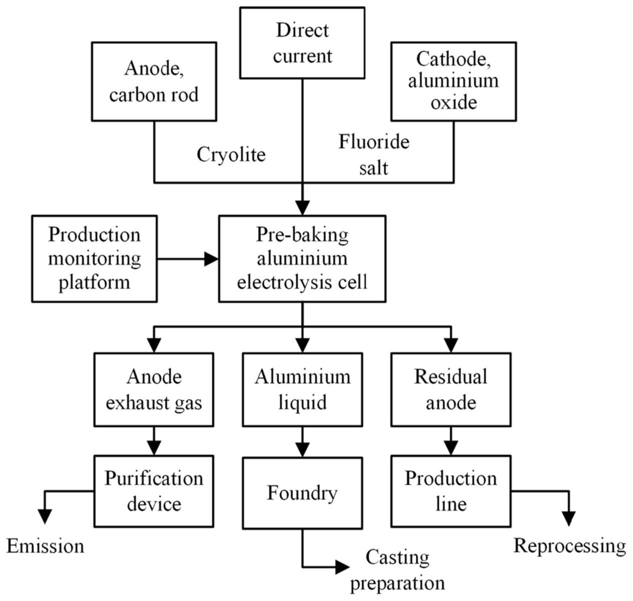


Fig. 3 The basic process of aluminium electrolysis manufacturing

manufacturing is a comprehensive and complicated processing system in which the states of material balance and energy balance change continuously and interact with each other. Meanwhile, the external processes (such as anode changing and shell breaking) are carried out alternately.

Normally, producing one ton of electrolytic aluminium requires more than 13,000 kW h DC power consumption by using current aluminium electrolysis technology. In fact, the theoretical power consumption is about 6320 kW h/t-Al, so the energy utilization rate does not exceed 50% [1]. Therefore, the research on energy-saving and consumption-reducing technology in the aluminium electrolysis manufacturing (AEM) has significant theoretical and engineering application value for improving the production efficiency and reducing the energy consumption.

Exploring the operation rules of the AEM to construct the system model can not only effectively monitor its own process status, but also can carry out intelligent decision optimization based on this system model. The above strategy is an important way to achieve energy saving and consumption reduction [4]. In recent years, many methods have been proposed to predict the energy consumption of aluminium electrolysis, which can be generally divided into two categories [5]: Physical model-based methods and Fully data-driven methods. The physical model-based methods, modeling the AEM through a mathematical model and corresponding decision parameters [6], are mainly suitable for the occasion where the system mechanism is clear. However, because the mechanism of aluminium electrolysis manufacturing process is extremely complex, it is difficult to express it clearly in a concise form. Therefore, many hypothetical conditions are often present in the physical model. If the model is performed under this condition, the resulting model may deviate significantly from the actual process, thereby limiting the generality of the above method.

The data-driven approaches are based on the historical data samples to train the system model. Bastillo et al. [7] proposed a smart manufacturing strategy for friction-drilling process based on boosting ensembles. In this strategy, the most accurate data-driven model is explored to deal with small-scale experimental datasets. These methods are quite popular in the engineering application of electrolysis system because it does not need to know the complicated mechanism process of manufacturing system and can obtain the complex mapping law between decision variables and performance indexes only by learning and training series of process data. For example, Yi et al. [8] proposed an improved quantum-behaved particle swarm algorithm to optimize the operating parameters of the aluminium electrolysis process. Huang et al. [9] proposed a nonlinear process monitoring system based on kernel dictionary learning and applied it to aluminium electrolysis process manufacturing. A semantic network based on intuitionistic fuzzy directed hyper-graphs was proposed by Chen et al. [10] for the state recognition of aluminium electrolysis cells. In [11], a knowledge reasoning fuzzy Bayesian network for root cause analysis of abnormal aluminium electrolysis cell condition was presented. Huang et al. [12] used the data structure dictionary learning-based method to carry out multimode process monitoring and applied it to the aluminium electrolysis process.

Through the above analysis, it can be seen that the choice and optimization of machine learning methods are crucial to accurately establish the data-driven prediction models. The neural networks have been successfully and widely applied as the master model in complex process systems because of their strong generalization ability and good nonlinear fitting ability. Khera et al. [13] used artificial neural networks to monitor the status of online aluminium electrolytic capacitors. Xu et al. [14] proposed a novel method based on neural network genetic algorithm (NNGA) for the optimization of cell voltage. A multi-fault diagnosis of aluminium electrolysis based on modular fuzzy neural networks was proposed in [15]. Zeng et al. [16] developed a genetic neural network model based on cell resistance signals for the diagnosis of anode anomaly and metal wave. Bak et al. [17] used shallow neural network and data feature selection technology to predict the quality of aluminium castings. Ding et al. [18] established a dynamic evolutionary model of aluminium electrolysis manufacturing system based on multi-sampling inherited HAPFNN method.

The above researches promote the development of energy saving and consumption reduction technology in the AEM. However, the traditional neural network is usually static once the model training is completed and its model parameters cannot be dynamically updated with the changes of manufacturing systems, so it is difficult to truly reflect the complex mapping law between process parameters and product performance.

To increase the model's incremental learning ability on the operation rules of the AEM, introducing the filtering technology to refresh the model's parameters in real time is expected to overcome the defects of the above model. The previous literatures [19–22] proposed an extended Kalman filter neural network (EKFNN) and an unscented Kalman filter neural network (UKFNN) based on Kalman filtering theory and a NN algorithm. The above methods essentially make use of the extended Kalman filter (EKF) and unscented Kalman filter (UKF) to dynamically adjust the NN's weights and thresholds and establish a dynamic evolutionary model that changes with the production conditions in real-time to realize the optimal design of the process conditions in the AEM. Yao et al. [23] combined unscented Kalman filtering and neural networks in a Gaussian noise environment to establish a dynamic evolutionary model for aluminium electrolysis manufacturing. Such methods are usually more suitable for the process data whose noise meets Gaussian distribution or approximate Gaussian distribution.

In fact, there are a series of physical and chemical reactions inside the aluminium electrolysis process. Meanwhile, there are multiple operations such as changing the anode, lifting the bus and extracting the aluminium in the external environment. These factors make the noise characteristics redundant and complex. However, the existing research lacks an exploration of the incremental learning algorithm in the presence of dense noise. Inspired by the particle filtering (PF) theory, which has the typical advantages in dealing with non-Gaussian and nonlinear systems by using a set of random particles with importance weights to estimate the probability density function (PDF) of system states, a new notion employing the PF technology to update the NN's parameters is proposed. Studying the particle filter neural network (PFNN) is expected to break the category of Gaussian hypothetical learning models. Meanwhile, to further improve the prediction accuracy of learning models (i.e., PFNN), the problems of particle degradation and particle scarcity in the learning model should also be addressed.

Based on the above two research motivations, this paper presents a particle filter neural network with the improved clustering kernel function smoothing (ICKS-PFNN). The competitive advantages of the ICKS-PFNN are to use the dynamic approximation characteristics of particles to adjust the NN's weights and thresholds in real time and solve the problems of particle degradation and particle scarcity to improve the prediction accuracy by employing the KFCM clustering algorithm and a novel proportional distribution sampling strategy for important sampling.

Based on the above analysis, the main contributions of this paper can be summarized as follows:

1. Considering the dynamic performance of the AEM, this paper adopts the neural network as the main model and uses the optimization strategy of updating the NN's weights and thresholds by PF technology to establish a novel PFNN incremental learning model for the aluminium electrolysis process.
2. To maintain the particles' diversity in the algorithm, this paper presents a new proportional distribution sampling strategy to improve the stability of the algorithm and the accuracy of parameter estimation.
3. Based on the above findings, this paper systematically proposes a novel particle filter neural network algorithm based on improved clustering kernel function smoothing (ICKS-PFNN) and gives a detailed algorithm design process.
4. Apply the above algorithm to the industrial modeling problems of aluminium electrolysis manufacturing. Experimental results show that compared to related methods, ICKS-PFNN can accurately predict process energy consumption.

The rest of this paper is organized as follows. Section 2 gives a clear problem description encountered in the modeling process of aluminium electrolysis. Section 3 firstly proposes the particle filter neural network (PFNN), then introduces KFCM algorithm and kernel function smoothing method, presents a new proportional distribution sampling strategy, and, finally, systematically presents the novel particle filter neural network algorithm based on the improved clustering kernel function smoothing (ICKS-PFNN). In Sect. 4, the algorithm proposed in this paper is applied and verified in AEM. Section 5 provides a summary.

2 Problem description

According to the analysis of literature [19–22], Kalman filter neural network systems are suitable for solving the state estimation problems of Gaussian system. However, there are many nonlinear and non-Gaussian complex dynamic systems in the process industrial manufacturing, which limits the application of the above Kalman filter network systems. Facing this shortcoming of theory and application, it is necessary to further broaden the construction framework of the filtering neural network systems. This paper establishes a filtering neural network system framework and mainly studies the particle filter neural networks, as shown in Fig. 4.

Particle filter (PF) [24–26] is a statistical filtering method based on Monte Carlo theory and Bayesian principle in Fig. 4. Its essence is to extract particles (samples) from the posterior probability and give particles corresponding weights to represent probability distribution. Compared with Kalman filter, particle filter is more flexible. It not only gets rid of tedious numerical calculations, but also can be theoretically applied to any system models. Meanwhile, considering that the neural network is a typical representative of the data-driven model, its internal unit can be regarded as a black box, so the nonlinear relationship between input variables and output variables can be directly established only through a large amount of training data. Based on the above characteristics, the existing literatures [27–29] effectively combined the two theory and used neural network features to update particles. Inspired by the above literatures, this paper proposes a research approach that uses the dynamic characteristics of particle filter to optimize the neural network model based on the principle of reverse thinking and presents a novel particle filter neural network (PFNN) model.

In order to intuitively compare the two ideas, the following are the overall flows of the above two modeling methods.

Algorithm 1: Existing algorithms

```
Step 1: Generate particles from the prior distribution $p(x_0)$ in the
    specific process system model;
Step 2: Calculate the weights of particles;
Step 3: Take particles' weights as NN's weights;
Step 4: Use the state value of the specific process system model as the
    NN's inputs, and the measured
value as the NN's outputs;
Step 5: Obtain the particles' weights optimized by the neural network;
Step 6: Perform weight normalization;
Step 7: Fuse state estimation $x_k$ and output measurement value $y_k$
    using measurement equation;
```

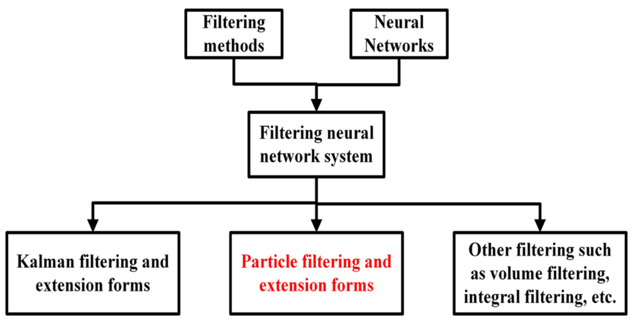

Filtering methods
Neural Networks
Filtering neural network system
Kalman filtering and extension forms
Particle filtering and extension forms
Other filtering such as volume filtering, integral filtering, etc.

Fig. 4 The broadened filtering neural network systems

Algorithm 1: Existing algorithms

```
Step 8: Use the state equation of the system to predict the unknown
    state parameter $x_{k+1}$;
Step 9: At moment $k = k + 1$, go to Step 2.
```

Algorithm 2: New algorithm

```
Step 1: Initialize the neural network model;
Step 2: Take the weights of the neural network as the state variables
    of the particle filter, and the outputs
of the neural network as the measurement variables of the particle
    filter;
Step 3: Generate particles from the prior distribution $p(x_0)$ in the state
    space model of the neural network;
Step 4: Run the particle filter algorithm to update the weights of
    particles;
Step 5: Obtain optimized particles (that is, neural network weights);
Step 6: Perform weight normalization;
Step 7: Fuse state estimation to get the optimal NN's weights and
    thresholds;
Step 8: Import the weights and thresholds into the BPNN model for
    performance test to determine
whether the prediction accuracy is met;
Step 9: At moment $k = k + 1$, go to Step 4.
```

It can be seen from the above algorithm flows that when new data are obtained, two algorithms both can dynamically approach the best model. In terms of the modeling methods, the existing algorithms update the particles' state through the measured data based on particle filter as the main model. In addition, the state space model used for parameter estimation needs to be established based on the specific process mechanism [27], which makes the modeling process more complicated. The significant difference is that the PFNN proposed in this paper takes the neural network model as the main model. The PFNN starts from the data of the research problem itself and does not need to fully understand the mechanism of the system. In the PFNN, the characteristics of the particle filter are used to update the neural network. Specifically, the NN's weights and thresholds are used as the state variable of particle filter, and the NN's output is used as the measurement variable of particle filter. Although the proposed algorithm has obvious advantages compared with existing algorithms, the PFNN still has not got rid of the loss of particle diversity [30] and is sensitive to the influence of isolated particle distribution and particle intensive degree. In order to describe the above problems, Table 1 simulates the

Table 1 Simulating particle updating process

Resampling number	Particles	Particles' mean	Particles' variance	Particles' type
I	2 4 9 2 3 7 1 8 3 6	4.5	2	8
II	9 9 2 4 2 3 7 8 3 6	4.9	2.67	7
III	9 9 9 9 4 3 7 8 3 6	6.7	2.54	6
IV	9 9 9 9 9 9 9 9 7 8	8.7	0.70	3
V	9 9 9 9 9 9 9 9 9 9	9.0	0	1

updating process of 10 particles based on the resampling strategy which copies particles with larger weights and eliminates particles with smaller weights.

As can be seen from Table 1, although the particles' variance is continuously reduced, the particles' type also gradually decreases in the iterative update process of particle weights using resampling method. Thus, the resampling technique solves the particles scarcity problem and reduces the overall variance, but it leads to the loss of particle diversity and samples dilution.

Besides the above-mentioned problem of the loss of particle diversity, the algorithm is also susceptible to isolated particle distribution and particle intensive degree.

The essence of particle filter is to approximate the posterior estimation results by using a set of sample points (particle set) with weights. The horizontal and vertical coordinates in Fig. 5 represent the normalized sampling range. The black "+" indicates normal sampling points, and the red "+" indicates abnormal sampling points (isolated particles). The middle black dot in the position of (0.5,0.5) represents the mean value of real particles. The black circle indicates the sampling range of the filtering algorithm, and its size can be controlled by adjusting the variance. (The small circle is the sampling range after setting a small variance, and the large circle is the sampling range after setting a large variance.) The red arrow indicates the offset direction of sampling range. Figure 5a shows that the overall sampling range is shifted to the lower left corner due to the dense particles in the red circle; Fig. 5b shows that the overall sampling range is shifted to the upper left corner due to the appearance of the red isolated particle "+". The above phenomena indicate that the isolated particle distribution and particle intensive degree are easy to change the original particles' sampling range, resulting in an anomaly of the whole sampling direction during the particles' updating process, which affects the accuracy of parameter estimation.

In order to solve the above problems, this paper further explores the fusion strategy of particle filter and neural network. Based on the proposed particle filter neural network (PFNN) that can typically deal with nonlinear non-Gaussian problems, a new clustering kernel function smoothing method is presented to avoid the adverse effects caused by the problems of the loss of particle diversity and particle state heterogeneity. Finally, this paper systematically develops the construction framework of particle filter neural network based on the improved clustering kernel function smoothing (ICKS-PFNN). The detailed design process of the theory and algorithm will be introduced in turn below.

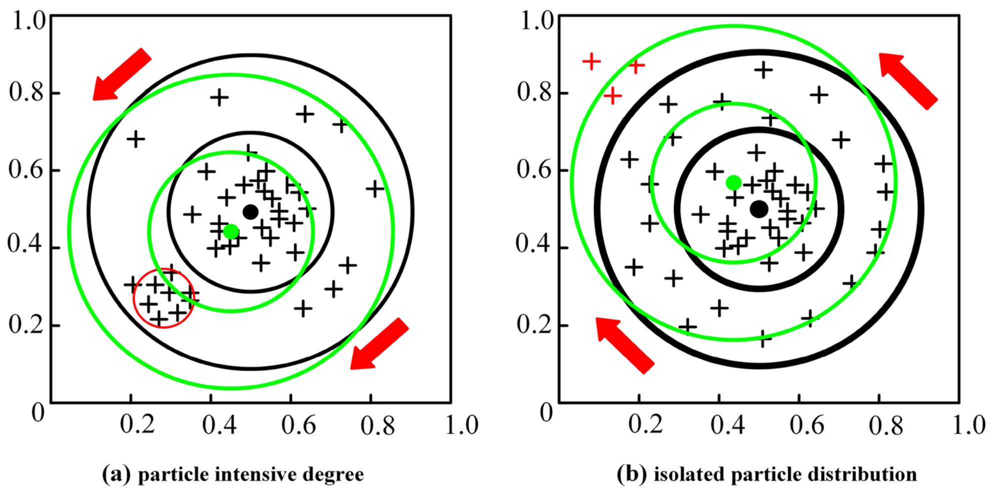

1.0
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0
(a) particle intensive degree
(b) isolated particle distribution

Fig. 5 The effects of isolated particle distribution and particle intensive degree

3 Particle filter neural network model based on the improved clustering kernel function smoothing method (ICKS-PFNN)

3.1 Particle filter neural network (PFNN)

In order to give full play to the dual advantages of particle filter method and neural network, this paper establishes a new particle filter neural network (PFNN) algorithm which expands the theoretical scope of filtering neural network family. This method adopts PF theory to estimate the parameters of NN's weights and thresholds. Essentially, the NN's weights and thresholds are used as the state variable x of particle filter and the NN's output is used as the measurement variable y of particle filter. Then, the optimal weights and thresholds are updated iteratively with the continuous operation of the manufacturing system.

The PFNN combines the technical characteristics of particle filter and neural network and can fully exert the advantages of filtering network when the manufacturing system is nonlinear and non-Gaussian. In this paper, the state space representation method of neural network is used to establish the filtering equations [31, 32] as shown in Eq. (1).

$$\begin{cases} \omega_k = \omega_{k-1} + \theta_k \\ y_k = h(\omega_k, u_k) + v_k \end{cases} \tag{1}$$

$$h(\omega, u) = \sum_{j=1}^{9} \frac{\omega_{jk}^{**}}{1 + \exp\left[-\left(\sum_{i=1}^{9} \omega_{ij}^{*} u_i + a_j\right)\right]} + b \tag{2}$$

where ω_k represents the state variables at moment k (i.e., the BPNN's weights and thresholds to be estimated); u_k represents the input variables of the industrial manufacturing system at moment k; y_k represents the measurement variables at moment k (i.e., the output variables for evaluating the advantages and disadvantages of the industrial manufacturing system). Assume that the system measurement noise v_k is Gaussian noise with mean 0 and variance R; the system process noise θ_k is Gaussian noise with mean 0 and variance Q. The NN's weights at moment k depend on the NN's weights at moment $k-1$ and the random system process noise θ_k, and the measurement noise v_k mainly describes the modeling error caused by sensors and other devices in the system. The nonlinear measurement function $h(\cdot)$ is approximated by using a multilayer perceptron in Eq. (2). ω_{ij}^{*} represents the connection weights between the i-th input layer and the j-th hidden layer; a_j represents the thresholds of hidden layer neurons; ω_{jk}^{**} represents the connection weights between the j-th hidden layer and the k-th output layer; b represents the thresholds of output layer neurons; u_i is the input variable.

Taking Eqs. (1) and (2) as the research object, the specific process of the PFNN algorithm is as follows:

1. Initialization.

Extract N particles $\omega_0^i \sim p(\omega_0)$, $i = 1, 2, \ldots, N$ from the prior distribution $p(\omega_0)$ established by the NN's weights and thresholds.

2. Importance sampling stage.

Use Bayes' conditional probability from Eq. (3) to calculate the probability distribution $q(\omega_k|\omega_{0:k-1}^{(i)})$, and then sample the particles according to Eq. (4):

$$q(\omega_k|\omega_{0:k-1}^{(i)}) = \frac{q(\omega_{0:k}^{(i)})}{q(\omega_{0:k-1}^{(i)})} \tag{3}$$

where $q(\omega_{0:k-1}^{(i)})$ represents the cumulative distribution probability of the i-th particle ω from the beginning to the moment of k-1; $q(\omega_{0:k}^{(i)})$ represents the cumulative distribution probability of the i-th particle ω from the beginning to the moment of k; $q(\omega_k|\omega_{0:k-1}^{(i)})$ represents the probability of the i-th particle ω at moment k under the premise that the state data are known at moment k-1.

$$\hat{\omega}_k^{(i)} \sim q(\omega_k|\omega_{0:k-1}^{(i)}, y_{1:k}) \approx q(\omega_k|\omega_{0:k-1}^{(i)}) \tag{4}$$

Update the weights of each particle:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k|\omega_k^{(i)}) p(\omega_k^{(i)}|\omega_{k-1}^{(i)})}{q(\omega_k^{(i)}|\omega_{0:k-1}^{(i)}, y_{1:k})} \tag{5}$$

where y represents the measurement variables (that is, the NN's output); ω represents the state variables (that is, the NN's weights and thresholds); $p(\cdot)$ represents the probability. The subscript indicates the moment; the superscript indicates the particle.

Weights normalization:

$$\tilde{w}_k^{(i)} = w_k^{(i)} / \sum_{j=1}^{N} w_k^{(j)} \tag{6}$$

where w is the particles' weights before normalization; N is the particles' number; $\tilde{w}$ is the particles' weights after normalization.

3. Selection process (resampling)

Use the resampling algorithm; the particle sets $\hat{\omega}_{0:k}^{(i)}$ are copied and eliminated according to the normalized weights $\tilde{w}_k^{(i)}$.

Reset the weights $w_k^{(i)} = \tilde{w}_k^{(i)} = 1/N$.

4. Continuously looping steps (2) and (3), and the final state estimation is the optimal NN's weights and thresholds ω.

This paper proposes a PFNN algorithm that can dynamically deal with nonlinear non-Gaussian system through the deep integration of PF and BPNN theory. However, to avoid the particle scarcity and particle degradation in the PFNN algorithm, the improved clustering kernel function smoothing (ICKS) is presented in the next section for further improving the accuracy of parameter estimation.

3.2 Improved clustering kernel function smoothing (ICKS)

3.2.1 Fuzzy kernel C-mean clustering (KFCM)

In order to further mine the estimation accuracy of PFNN algorithm, we hope to use some clustering algorithms to divide the particles into different "classes". Among them, the clustering algorithm with C-mean as the core is most widely used, such as FCM [33], KFCM [34, 35] and improved form mentioned in literature [36]. After comprehensive comparison, this paper chooses KFCM algorithm for particle clustering. This KFCM method first maps the data sets into a high-dimensional space through the kernel function and then clusters the high-dimensional space data through fuzzy theory. In this way, valuable feature information can be extracted to a greater extent, making the clustering effect more obvious.

The KFCM algorithm overcomes the following shortcomings of FCM: (1) FCM algorithm has a poor ability to adjust isolated data and mutation data; (2) FCM clustering is sensitive to the choice of initial clustering center. Because the KFCM algorithm can improve the data clustering effect as a whole and has good robustness to isolated and mutated data, the KFCM is used to cluster the weights and thresholds (i.e., particles) after optimizing them by the PFNN algorithm. The specific process is as follows:

The objective function of KFCM algorithm in Eq. (7) is built firstly based on the PFNN.

$$J_m = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^{m}\|\Phi(\omega_k) - \Phi(\omega_i^{'})\|^2 \tag{7}$$

where C $(2 \le C \le N)$ is the number to be grouped; $\omega_k = \{\omega_1, \omega_2, \ldots, \omega_N\}$ are the data sets to be classified (that is, the weights and thresholds obtained by PFNN); $\omega_i^{'} = \{\omega_1^{'}, \omega_2^{'}, \ldots, \omega_C^{'}\}$ are the clustering centers of the data set to be classified; N is the number of clustering samples; $\Phi(\cdot)$ is nonlinear transformation function; $\Phi(\omega_k)$ is the mapping of data in high-dimensional space; $\Phi(\omega_i^{'})$ is the mapping of cluster centers in high-dimensional space; u_{ik}^{m} is the membership matrix; m is the fuzzy weighting factor.

Based on the maximum value of the Silhouette coefficient [37], we finally determine the number of clusters C to be 20. Referring to literatures [38–40], it is pointed out that m is generally greater than 1 and is generally taken as 2 or 3.5.

Compared with the FCM algorithm, it can be seen that the KFCM algorithm using $\|\Phi(\omega_k) - \Phi(\omega_i^{'})\|^2$ replaces the original Euclidean distance $\|\omega_k - \omega_i^{'}\|^2$.

$$\|\Phi(\omega_k) - \Phi(\omega_i^{'})\|^2 = K(\omega_k, \omega_k)K(\omega_i^{'}, \omega_i^{'}) - 2K(\omega_k, \omega_i^{'}) \tag{8}$$

Because the Gaussian kernel function is recognized as having excellent generalization ability, and its range is between (0,1), in order to make the calculation process simple, this paper selects to use the Gaussian kernel that meets the Mercer condition, namely

$$K(x, y) = \exp\left(\frac{-\|x - y\|^2}{2\sigma^2}\right) \tag{9}$$

where, σ is the scale parameter. In literature [41], as $\sigma \to \infty$, the KFCM reduces to the classical FCM; if $\sigma \to 0$, it is just the strategy used in NPS algorithm. Therefore, σ is generally taken as 0.2 or 2.

To minimize the objective function J_m, use J_m to calculate partial derivatives for u_{ik} and $\Phi(\omega_i^{'})$, and make them equal to 0. So, Eqs. (10) and (11) can be obtained.

$$\frac{\partial J_m}{\partial u_{ik}} = m \cdot u_{ik}^{-m}\|\Phi(\omega_k) - \Phi(\omega_i^{'})\|^2 = 0 \tag{10}$$

$$\frac{\partial J_m}{\partial \Phi(\omega_i)} = -2\sum_{k=1}^{N} u_{ik}^{m}\|\Phi(\omega_k) - \Phi(\omega_i^{'})\|^2 = 0 \tag{11}$$

From Eqs. (8) to (11), we can get:

$$\|\Phi(\omega_k) - \Phi(\omega_i^{'})\|^2 = 2 - 2K(\omega_k, \omega_i^{'}) \tag{12}$$

Initializing the center matrix $\omega_i^{'(0)}$, Eqs. (13) and (14) are obtained.

$$U^{(s)} = [u_{ik}]^{(s)} = \frac{[1 - K(\omega_k, \omega_i^{'(s)})]^{-\frac{1}{m-1}}}{\sum_{i=1}^{C}[1 - K(\omega_k, \omega_i^{'(s)})]^{-\frac{1}{m-1}}} \tag{13}$$

$$\omega_i^{'(s+1)} = \frac{\sum_{j=1}^{C} u_{ik}^{(s)} K(\omega_k, \omega_i^{'(s)})\omega_k}{\sum_{k=1}^{C} u_{ik}^{(s)} K(\omega_k, \omega_i^{'(s)})} \tag{14}$$

where s is the number of iterations; ε is the cutoff error value of iteration process. If s is too small or ε is too large, clustering effect is not ideal; if s is too large or ε is too small, the clustering time will increase. Therefore, s and ε are set to 10,000 and 0.0001, respectively, in this paper.

Calculate iteratively Eqs. (13) and (14). If $\|U^{(s+1)} - U^{(s)}\| < \varepsilon$ is met, then terminate the iteration to obtain the required clustering center ω_i' and membership matrix U; otherwise, continue to update the membership matrix.

We have done the cross validation of m and σ values. It can be seen from Table 2 that the clustering effect is better when m and σ are equal to 2 and 0.2.

3.2.2 Clustering kernel function smoothing based on new proportional strategy

Because the PFNN algorithm cannot get rid of the inherent defects of particle filter, this paper proposes a clustering kernel function smoothing method based on new proportional strategy. This method can not only effectively solve the problem of the loss of particle diversity, but also particles' uneven density in each group after KFCM clustering, thereby further improving the estimation accuracy of neural network parameters.

The specific implementation steps of the clustering kernel function smoothing algorithm based on new proportional strategy are as follows:

1. Select the particle with the largest weight in the PFNN algorithm (that is, the NN's weights and thresholds to be estimated) as the clustering center, and calculate the distance to the remaining particles. If the distance is less than the set threshold ε, the particles are classified into same category;
2. Similarly, select the particle with the largest weight from the remaining particles as the clustering center of the new round, and perform step (1) recurrently until the cluster is completed;
3. After all particles have been clustered by KFCM, if the cluster number is less than the original set cluster number C, end the cluster process; otherwise, reduce the cluster number C or increase the threshold ε, and then, restart the cluster until the set condition is satisfied;

Table 2 Performance index of clustering effect under different m and σ

m	σ	RS	Γ	DB	I	CH	S
2	0.2	**0.9845**	4025	**0.1547**	**9652**	**4389**	**0.8941**
	2	0.9527	**4140**	0.1755	9042	3678	0.8410
3.5	0.2	0.9078	3841	0.2476	7410	3702	0.8327
	2	0.8853	3912	0.2950	6894	3362	0.7944

Bold indicates the optimal values of the relevant indicators

The related clustering indicators of RS, Γ, DB, I, CH, S can refer to literature [42]

4. After the clustering is completed, the proportional distribution sampling method is applied to each class for particle sampling.

   In the proportional distribution sampling method, all particles in each class are arranged from large to small by weights. Particles with small weight are not directly excluded, but are combined with particles with large weight in a certain proportion to generate new particles to replace the excluded particles of traditional sampling methods. The relevant equations are as follows.

$$N_i = N \times \alpha_i \tag{15}$$

$$N_\lambda = N_i \times \lambda \tag{16}$$

$$\omega_n' = L \times (\omega_q + \omega_p) \tag{17}$$

   where α_i is the ratio of the particles' weight accumulation in the i-th cluster to the weight accumulation of all particles; N is the total number of particles; N_i is the number of particles in the i-th cluster; $\omega_k, k = 1, 2, \ldots, N$ are a set of particles arranged from large to small by weights in the i-th cluster; λ is the distribution factor, which represents the proportion of particles to be combined from each cluster. The larger the value is, the more the diversity of particles is, but the convergence time of particles will increase. N_λ is the number of particles to be combined in N_i; $\omega_q, q = 1, 2, \ldots, N_\lambda/2$ are the particles with the smaller weight in ω_k; $\omega_p, p = N_i, N_{i-1}, \ldots, N_i - N_\lambda/2 + 1$ are the particles with the greater weight in ω_k; ω_n' are the newly generated particles.

   Let L be taken as 0.5, then Eq. (17) is expressed as:

$$\omega_n' = \frac{\omega_q + \omega_p}{2} \tag{18}$$

   The above equation indicates that the newly generated particles are the average of larger and smaller particles' weights.

   In Fig. 6, the yellow circles are the particles with larger weight; the green circles are the particles with smaller weight; the blue circles are the particles without proportional distribution sampling; the red circles are the particles obtained by proportional distribution sampling of particles with larger weight and smaller weight.
5. Kernel function smoothing is performed on the particles obtained in step (4) according to their respective groups. In this step, the Gaussian kernel function is selected for calculation to obtain the final state estimation of the target.

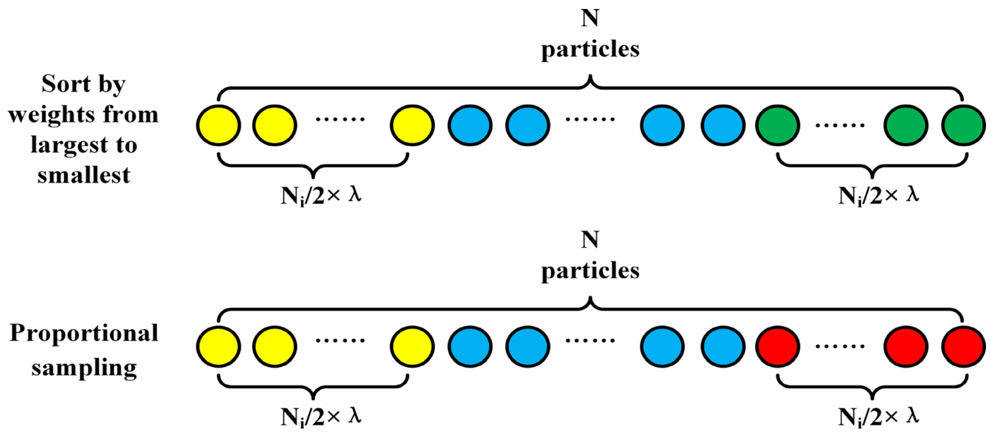


Fig. 6 Particles' change before and after proportional sampling

$$\hat{\omega}_j^{(i)} = \frac{c_j}{N}\omega_j^{(i)} k\| \frac{\omega_j^{(i)} - \omega_j^{(0)}}{R} \| \tag{19}$$

Fusion output:

$$\hat{\omega}^{(i)} = \sum_{j=1}^{C} \beta_j \hat{\omega}_j^{(i)} \tag{20}$$

where $c_j = 1/\sum_{j=1}^{C} k\| \frac{\omega_j^{(i)} - \omega_j^{(0)}}{R} \|$ is the normalization parameter; $\omega_j^{(i)}$ and $\omega_j^{(0)}$ represent the i-th particle and particle mean in the j-th cluster; $k(\cdot)$ is the kernel function centering around $\omega_j^{(0)}$; $\hat{\omega}_j^{(i)}$ represents the particle smoothed by the i-th particle kernel function in the j-th cluster; β_j is the distribution factor, $\sum_{j=1}^{C} \beta_j = 1$.

The parameter R determines the smoothness of the kernel function. The larger the parameter R is, the better the smoothness is. By adjusting the parameter R, a trade-off can be made between over smoothing and under smoothing caused by noise. Literature [43] points out that the scale parameter R is often selected artificially. From $R = 0.1$ to $R = 10$, we perform the traversal experiment with the step size of 0.1 and finally get the best result with $R = 0.5$.

It can be seen from the above algorithm that kernel function smoothing is introduced after sampling in order to avoid excessive concentration of particles in each clustering group, so that the state estimation of manufacturing system parameters tends to the posterior probability distribution, greatly improving the prediction ability of the algorithm.

3.2.3 ICKS-PFNN algorithm

Compared with the traditional BPNN algorithm, the ICKS-PFNN algorithm proposed in this paper has the following advantages:

1. Traditional BPNN belongs to the category of static models; while ICKS-PFNN is a dynamic model, it can constantly adjust itself. When it gets new data, ICKS-PFNN can actively approach the best model;
2. To make the model suitable for parameter estimation in nonlinear non-Gaussian systems, the particle filter is combined with neural network;
3. In order to maintain particle diversity, this paper proposes a novel proportional distribution sampling strategy for the importance sampling;
4. In order to prevent the negative impact of isolated particles and intensive particles, this paper introduces KFCM algorithm and kernel function smoothing technology to cluster and fuse the output of particles, so that the state estimation tends to be near the posterior peak, which helps to improve the estimation accuracy of neural network parameters.

Through analyzing the important links of the model construction process, this paper systematically proposes a novel clustering kernel function smoothing PFNN incremental learning model framework by integrating PFNN, KFCM clustering algorithm, new proportional distribution sampling and kernel function smoothing technology. The ICKS-PFNN algorithm framework is shown in Fig. 7.

It can be seen from Fig. 7 that the ICKS-PFNN algorithm firstly initializes the model and takes the NN's weights and thresholds as the estimation object. Secondly, the weights and thresholds that initially meet the expected accuracy are obtained by using PFNN algorithm to update parameter estimation. Thirdly, the weights and thresholds (i.e., particles) in PFNN algorithm are clustered by KFCM,

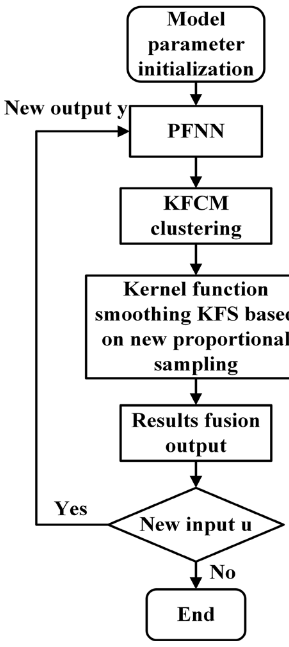

Model parameter initialization
New output y
PFNN
KFCM clustering
Kernel function smoothing KFS based on new proportional sampling
Results fusion output
Yes
New input u
No
End

Fig. 7 The flowchart of ICKS-PFNN algorithm

and then each class is randomly sampled by proportional distribution method. Fourthly, the kernel smoothing is performed on the sampled particles according to their respective distribution. Finally, the results are fused to obtain the final state estimation of the target. If the system sample increases or decreases (that is, when the input and output change), the model can update the NN's weights and thresholds ω to achieve a new dynamic balance.

4 Incremental learning model of aluminium electrolysis equipment process system based on ICKS-PFNN

4.1 Experimental object and model parameters

The process system of aluminium electrolysis equipment is complex, with various physical and chemical changes inside, and frequent exchange of materials, energy and information with the outside. It is difficult to obtain accurate the incremental learning model of process system based on traditional modeling methods. By using the ICKS-PFNN algorithm for adaptive dynamic system modeling, the aluminium electrolysis process can be updated and tracked in real time under various complex noises, which can effectively ensure the prediction accuracy and reliability of the model.

In this paper, through analyzing the influencing factors of the unit DC energy consumption of the aluminium electrolysis cell, the effective decision parameters are selected as series current, molecular ratio, aluminium level, electrolyte level, cell temperature, aluminium output, daily consumption of fluoride salt, blanking interval and cell voltage by combining the expert experience and considering the actual difficulty of on-site data acquisition. The aluminium electrolysis cell [23] in the 170kA series of an aluminium plant was sampled, and 773 groups daily data from 9 decision parameters were obtained, as shown in Table 3. The detailed parametric study on those parameters used in the developed model has been shown around the corresponding Equations for better understanding in Sect. 3.2.

4.2 The analysis and discussion of experimental results

In order to verify the effectiveness and superiority of the ICKS-PFNN algorithm, this paper compares the prediction errors of ICKS-PFNN with EKFNN, UKFNN, PFNN, CKS-PFNN and ICS-PFNN under the same data samples and MATLAB R2014b (CPU: i7-9750H; RAM: 8.00GB; GPU: GTX 1660 Ti) to measure the algorithms' performance. In all experiments, the 773 groups of aluminium electrolysis data in Table 3 are divided into 700 groups of training set samples and 73 groups of test set samples. The inputs are 9 decision parameters and the output is the unit DC power consumption. The network structure of BPNN is 9–9–1, which means that the number of nodes in input layer, hidden layer and output layer is 9, 9 and 1, respectively; Levenberg–Marquardt algorithm is selected as the training algorithm, and the learning rate is 0.1. The types of transfer functions in hidden layer and output layer are Sigmoid function and Purelin function, respectively.

The fitting effects of DC power consumption in Fig. 8 are from the incremental learning prediction models of the aluminium electrolysis process system established by using the above six algorithms. From Fig. 8a and b, we can see that the fitting ability of EKFNN and UKFNN is not very ideal, because they cannot guarantee excellent fitting results in the case of nonlinear and non-Gaussian. In addition, we perform a thorough ablation study on different components in the model through the experimental comparison from Fig. 8c–f. The fitting ability of the model from high to low is as follows: ICKS-PFNN > ICS-PFNN ≈ CKS-PFNN > PFNN as shown in Fig. 8c–f. Comparing Fig. 8d, f, the fitting ability of ICKS-PFNN is better than CKS-PFNN, because the proportional distribution sampling method is introduced to maintain the diversity of particles. Moreover, by analyzing Fig. 8e, f, the use of KFCM clustering improves the fitting ability of ICKS-PFNN to a certain extent.

Figure 9a shows the radar chart of absolute error based on prediction outputs among EKFNN, UKFNN and PFNN. It can be seen that PFNN algorithm has more scattered

Table 3 Data samples of aluminium electrolysis cell

Parameters	Range	Sample				
		1	2	3	...	773
Series current (A)	1600–1800	1679	1679	1679	...	1752
Cell voltage (mV)	3500–3800	3712	3723	3718	...	3617
NB times	600–800	646	707	671	...	781
Molecular ratio	2.20–2.90	2.31	2.31	2.31	...	2.72
Aluminium output (kg)	1200–1400	1260	1240	1260	...	1340
Aluminium level (cm)	15.0–25.0	21.5	21.5	22	...	16
Electrolyte level (cm)	5–20	17	17	17	...	8
Cell temperature °C	940–960	952	942	947	...	948
Daily consumption of Fluoride salt (kg)	15.0 ~ 30.0	29.0	23.4	21.6	...	19.8
DC power consumption per ton of Aluminium (kW.h/t-Al)	–	12350	12130	11889	...	11702

Series Current refers to the current in each electrolytic device, reflecting the electrolytic capacity of manufacturing system. *Cell Voltage* refers to the potential difference between cathode bus and anode bus, reflecting the operation status of electrolytic device. *NB times* refers to the normal interval of shelling and blanking process. *Molecular ratio* refers to the ratio of NaF to AlF3 in electrolyte. *Aluminium Level* is adjusted and matched according to different technical conditions such as specific cell structure design, electrolysis process and cell age. *Electrolyte Level* refers to the height of molten liquid inside the electrolytic device. This parameter directly affects the contact area between the surface of carbon anode and molten liquid in the tank. *Cell Temperature* refers to the temperature of the electrolytic material during the normal operation. *Aluminium Output* and *Daily Consumption of Fluoride Salt* are key factors to maintain energy and material balance in electrolysis process

points near the outer ring, so the fitting error is small. As shown in Fig. 9b, we see that the fitting accuracy of ICKS-PFNN is higher, even more than 10 times higher than that of PFNN. The reason is that the ICKS optimization method solves the inherent problem of particle diversity loss in PFNN algorithm, which makes the probability density distribution reflect the overall particle state in the sampling process.

Figure 10 visually presents the radar chart of the relative error percentage for the DC power consumption by the six schemes. It can be seen that the relative errors of EKFNN, UKFNN and PFNN belong to the same order of magnitude, while the relative errors of CKS-PFNN, ICS-PFNN and ICKS-PFNN belong to the same order of magnitude. As shown in Fig. 10, we can see that the relative error percentage of ICKS-PFNN is smaller than other models, indicating that the fitting effect of ICKS-PFNN is significantly better than other models. According to the experimental results, the particle filter neural network and the new clustering kernel function smoothing method proposed in this paper are feasible and effective. Overall, the performance of the ICKS-PFNN model is in good agreement with the actual characteristics of the aluminium electrolysis cell.

For Fig. 11, we see clearly that the proportions of relative error (RE) in the interval $|RE| < 0.2\%$ from EKFNN, UKFNN, PFNN, CKS-PFNN, ICS-PFNN and ICKS-PFNN, respectively, are 1, 1, 7, 26, 28 and 37%, in which ICKS-PFNN model has the maximum proportion. The proportions of relative error (RE) in the interval $|RE| > 5\%$ from EKFNN, UKFNN, PFNN, CKS-PFNN, ICS-PFNN and ICKS-PFNN, respectively, are 40, 26, 1, 0, 0 and 0%, in which EKFNN model has the worst prediction performance. These phenomena indicate that ICKS-PFNN captures excellent prediction performance.

Table 4 gives the comparison of the performance indicator of energy consumption prediction models established by six algorithms. It lists 6 evaluation indexes for different model performances, including Max, Min, Average, SSE, MSE and RMSE [48]. It can be seen that the prediction error of the ICKS-PFNN algorithm is the smallest, indicating that the prediction accuracy of the algorithm is quite high. In addition, the accuracy of CKS-PFNN and ICS-PFNN are almost the same, but the latter is slightly better than the former. It is proved that the idea of replacing ordinary sampling with proportional sampling is more reliable than that of replacing FCM with KFCM. Based on the results of SSE and MSE in Table 4, it can be seen that the new clustering kernel function smoothing theory has a more obvious effect to avoid the particle diversity loss, which further helps to improve the model performance and finally obtain the best parameter estimation.

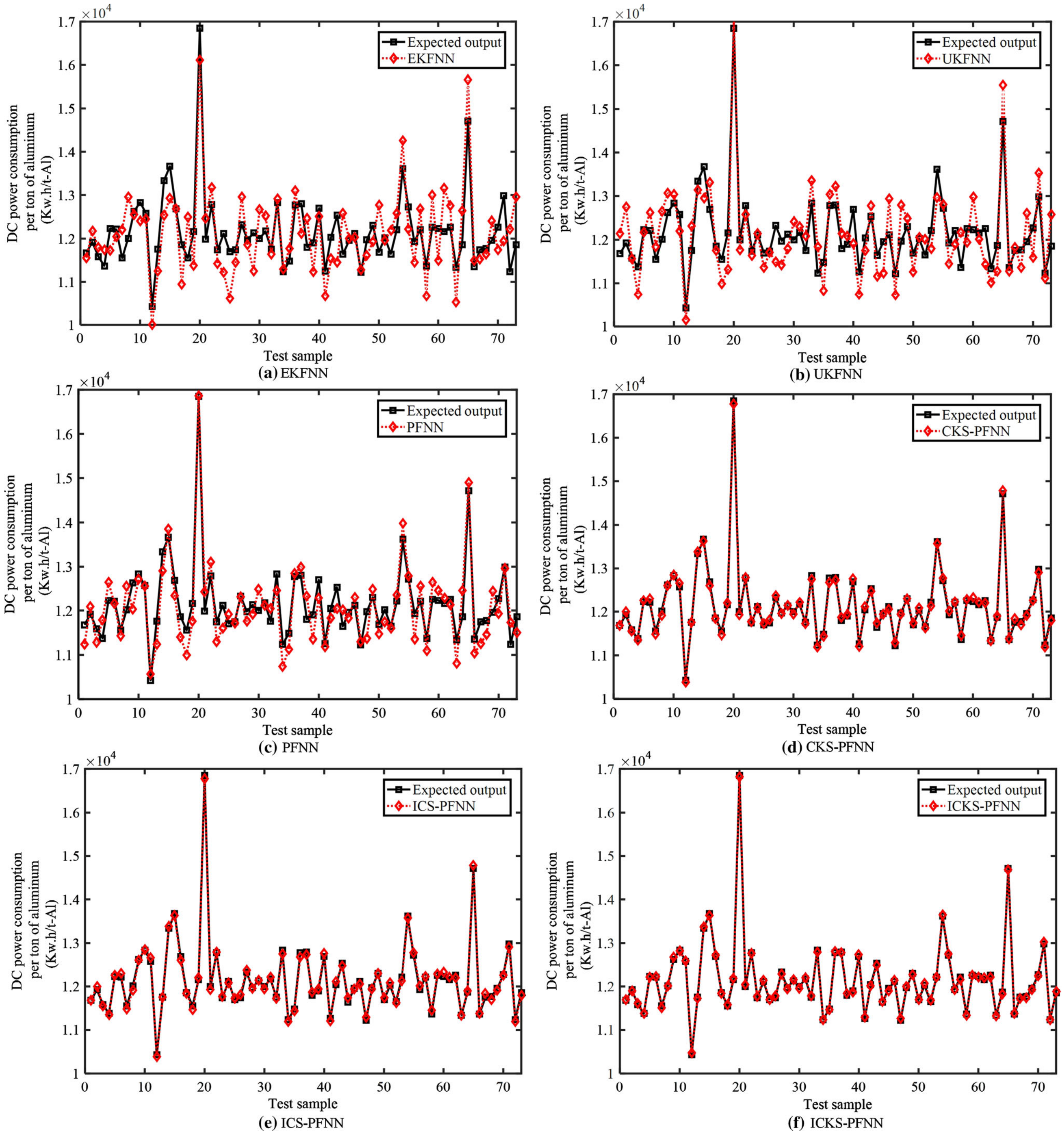


Fig. 8 The prediction outputs of DC power consumption based on EKFNN, UKFNN, PFNN, CKS-PFNN, ICS-PFNN and ICKS-PFNN

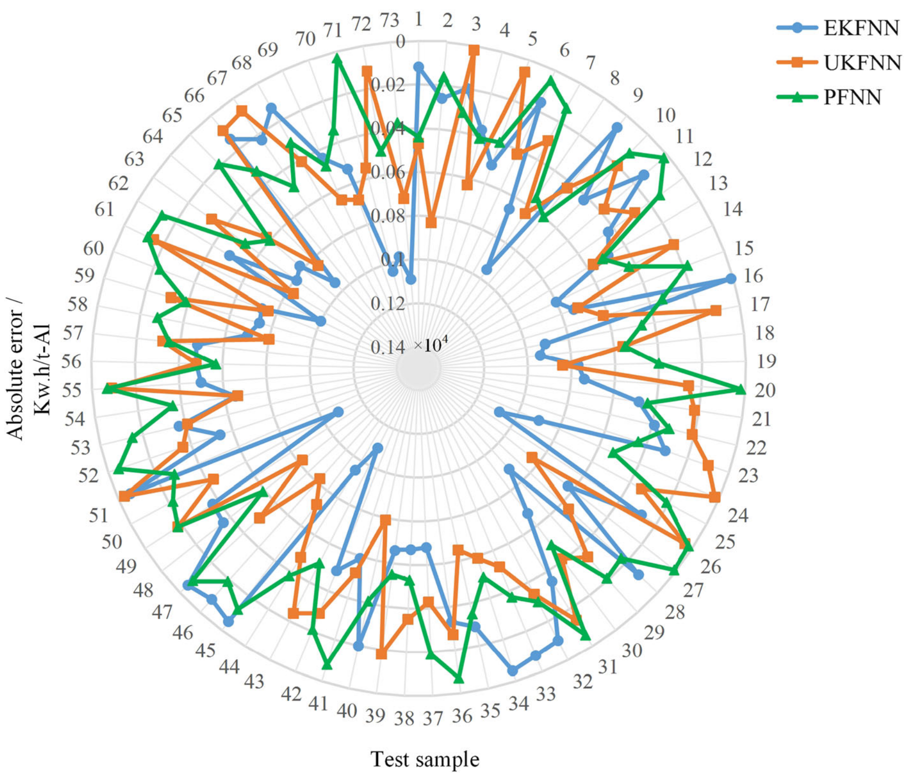
$$\mathrm{SSE} = \sum_{i=1}^{T}(y_i - y)^2 \tag{21}$$

$$\mathrm{MSE} = \frac{1}{T}\sum_{i=1}^{T}(y_i - y)^2 \tag{22}$$

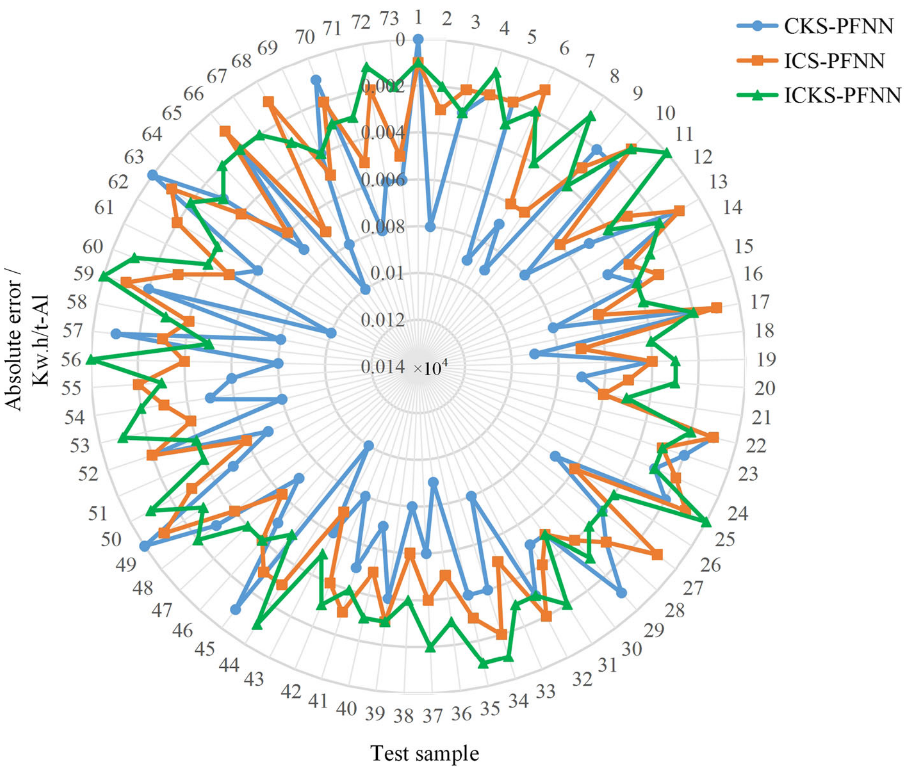
$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(y_i - y)^2} \tag{23}$$

where y_i is the predicted value of the test sample; y is the true value of the test sample; and T is the number of test sample groups.

Fig. 9 The radar chart of absolute error based on prediction outputs among EKFNN, UKFNN, PFNN, CKS-PFNN, ICS-PFNN and ICKS-PFNN



(a) Absolute error of prediction outputs among EKFNN, UKFNN and PFNN



(b) Absolute error of prediction outputs among CKS-PFNN, ICS-PFNN and ICKS-PFNN

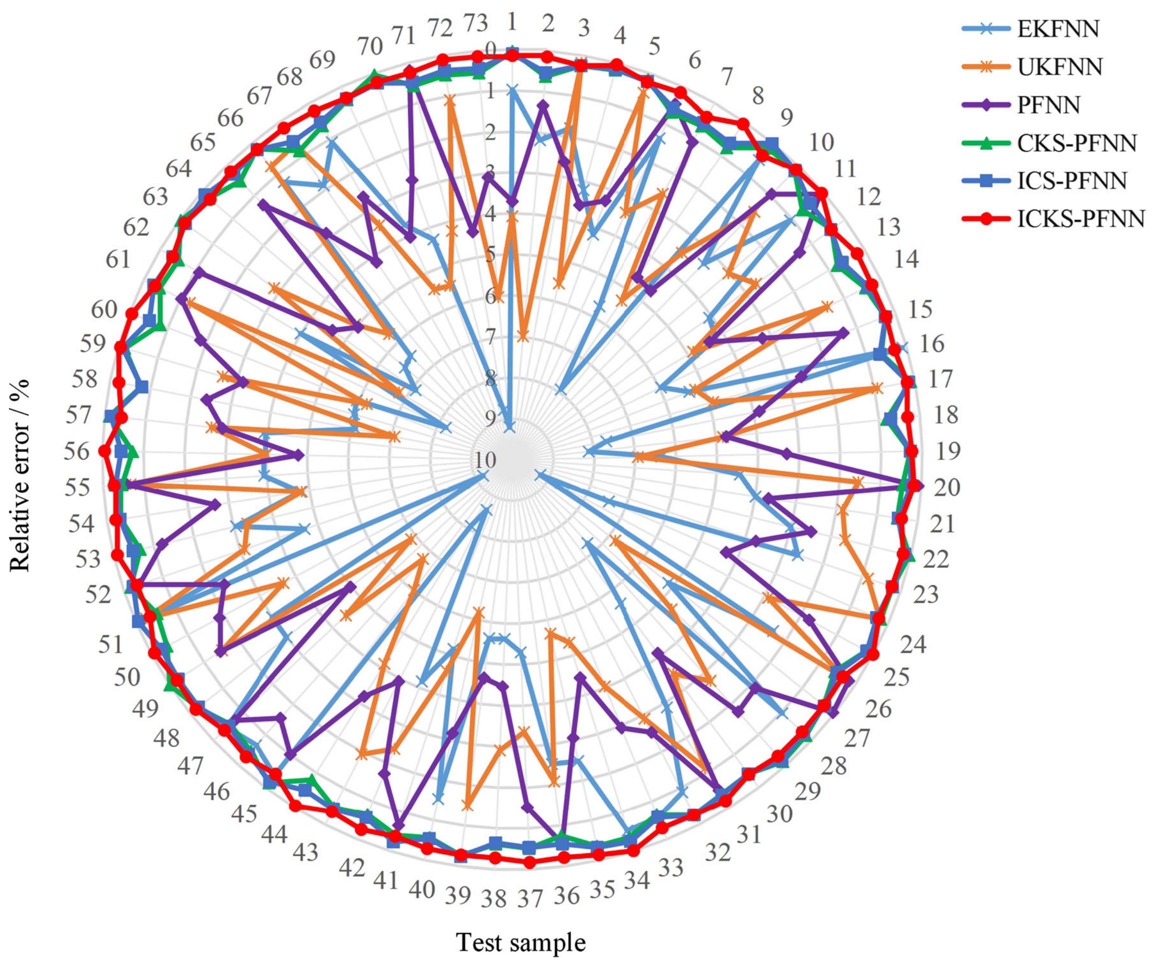


Fig. 10 The radar chart of relative error percentage based on prediction outputs among EKFNN, UKFNN, PFNN, CKS-PFNN, ICS-PFNN and ICKS-PFNN

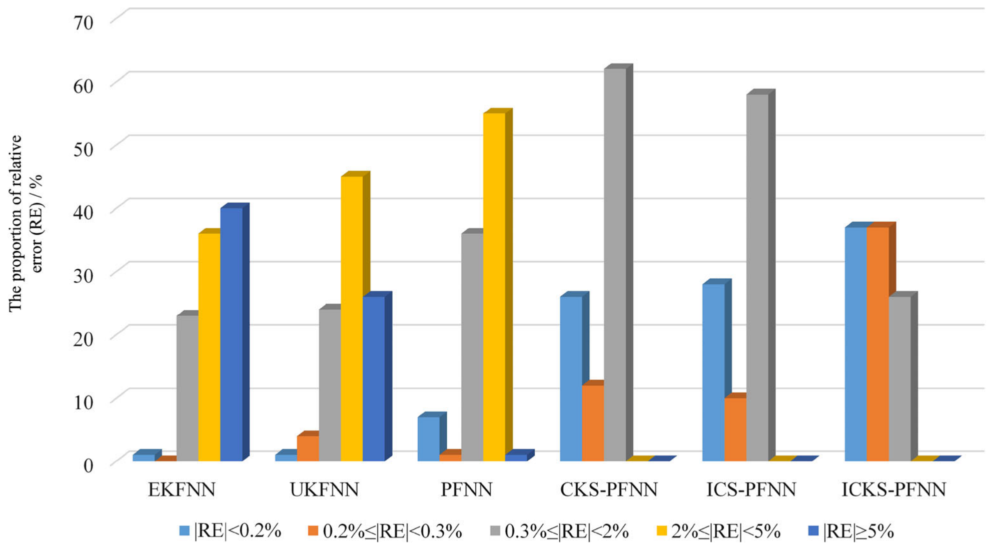


Fig. 11 The proportion of relative error (RE) in different intervals among EKFNN, UKFNN, PFNN, CKS-PFNN, ICS-PFNN and ICKS-PFNN

Table 4 The comparison of performance indicators from different models

Model	The number of relative error in different interval distribution						Statistical test	Complexity	
	Max	Min	Average	SSE	MSE	RMSE	p value	Time	Space
MLR	1176.8517	6.8744	571.6347	3.0178×10^{7}	4.2687×10^{5}	658.1246	8.77×10^{-4}	$O(n^3)$	O(1)
NLMR	997.5234	2.4826	443.9952	1.8206×10^{7}	2.4821×10^{5}	508.7492	5.91×10^{-4}	$O(n^3)$	O(1)
EKFNN	1075.8185	8.1645	547.0498	2.9213×10^{7}	4.0018×10^{5}	632.5993	8.56×10^{-4}	$O(n^3)$	O(n)
UKFNN	841.8672	2.7989	416.7612	1.6903×10^{7}	2.3154×10^{5}	481.1890	4.97×10^{-4}	$O(n^3)$	O(n)
BFNN	620.4753	3.5720	327.7918	9.8630×10^{6}	1.5778×10^{5}	435.1285	3.17×10^{-4}	$O(n^3)$	O(n)
PFNN	598.5619	1.9909	288.1628	8.1670×10^{6}	1.1188×10^{5}	334.4797	2.11×10^{-3}	$O(n^3)$	O(n)
CKS-PFNN	99.7162	4.1031	49.9339	2.5340×10^{5}	3.4712×10^{3}	58.9171	1.82×10^{-2}	$O(n^3)$	O(n)
ICS-PFNN	95.4715	1.2486	48.2590	2.2529×10^{5}	3.0571×10^{3}	52.8450	1.55×10^{-2}	$O(n^3)$	O(n)
ICKS-PFNN	49.9478	1.1100	26.0482	6.4987×10^{4}	8.9023×10^{2}	29.8368	–	$O(n^3)$	O(n)

The related performance indicators of energy consumption prediction models based on multiple linear regression (MLR, Fashoto et al. 2021) [44], multiple nonlinear regression (NLMR, Genidy et al. 2018) [45], extended Kalman filter neural network (EKFNN, Acosta et al. 2018) [19], unscented Kalman filter neural network (UKFNN, Yao et al. 2019) [23], Bayesian filter neural network (BFNN, Alshangiti et al. 2020 and Gu et al. 2019) [46, 47] and particle filter neural network (PFNN, Qin et al. 2019) [26] are added to Table 4

To further prove the effectiveness of the proposed method, Wilcoxon signed rank test is employed to perform a non-parametric test using the ICKS-PFNN as the reference objective. The corresponding results are summarized in Table 4. Based on the presented results, the proposed method is significantly better than other approaches. In addition, the algorithm complexity has also been added in Table 4. It can be seen that the space complexity of the PFNN algorithm is better than MLR and NLMR. Even if PFNN is continuously optimized to obtain ICKS-PFNN, the complexity of its algorithm has not increased.

In order to avoid the accidental effect of randomized initial values on the models, 20 independent tests were performed. The statistics of relevant performance indicators for DC power consumption after 20 tests from different models are shown in Table 5, including MAE, MRE and R [49]. In Table 5, the bold indicates the optimal values of the relevant indicators. So, we clearly see that the optimal MAE of EKFNN, UKFNN, PFNN, CKS-PFNN, ICS-PFNN and ICKS-PFNN is 7.3491, 5.3752, 3.6466, 0.6047, 0.5771 and 0.3278, and the optimal MSE of them is 0.0419, 0.0324, 0.0218, 0.0035, 0.0029 and 0.0020, respectively. The MAE and MSE of the ICKS-PFNN model are the smallest. In the six models, the correlation coefficient (R) of ICKS-PFNN is 0.9997, which is the largest, and the others are 0.7962, 0.8743, 0.9679, 0.9985 and 0.9988, respectively.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{24}$$

$$\mathrm{MRE} = \frac{1}{n}\sum_{i=1}^{n} |\frac{\hat{y}_i - y_i}{y_i}| \tag{25}$$

$$R = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (\hat{y}_i - y_i) \sum_{i=1}^{n} (y_i - \bar{y})}} \tag{26}$$

where y_i is the true value of the test sample; $\hat{y_i}$ is the predicted value of the test sample; $\bar{y_i}$ is the average value of the true value of the test sample; and n is the group number of test samples.

The statistical analysis results of Table 5 are given in Table 6 for better comparison between different models. In Table 6, the comparison indicators include Average, Max, Min, Std, Median, Lower Quantile and Upper Quantile. The bold in Table 6 indicates the optimal values of the relevant indicators. Compared with EKFNN and UKFNN, the ICKS-PFNN algorithm can solve nonlinear non-Gaussian system problems. Compared with PFNN, CKS-PFNN and ICS-PFNN, the ICKS-PFNN algorithm improves the particles' diversity and the ability to deal with the abnormal state of particles, due to its own new proportional distribution sampling for resampling and introducing KFCM clustering algorithm. Therefore, the prediction accuracy of ICKS-PFNN has a significant competitive advantage over other methods when constructing an incremental learning model.

In fact, it is time-consuming, laborious and inaccurate only through manual operation to adjust the decision-making parameters of aluminium electrolysis manufacturing. Using intelligent optimization theory to mine the optimal operating parameters is an important way to address the above issue. However, establishing an accurate

Table 5 Statistics of relevant performance indicators from 20 tests of different models

Test number	EKFNN			UKFNN			PFNN			CKS-PFNN			ICS-PFNN			ICKS-PFNN		
	MAE	MSE	*R*	MAE	MSE	*R*	MAE	MSE	*R*	MAE	MSE	*R*	MAE	MSE	*R*	MAE	MSE	*R*
1	7.4938	0.0459	0.7840	5.7091	0.0349	**0.8743**	3.9474	0.0241	0.9169	0.6840	0.0042	0.9974	0.6482	0.0037	0.9980	0.3568	0.0022	0.9993
2	7.4078	0.0449	0.7765	5.9010	0.0361	0.8492	4.0499	0.0248	0.9155	0.6144	0.0038	0.9975	0.6847	0.0040	0.9981	**0.3278**	**0.0020**	0.9993
3	7.9772	0.0479	0.7949	5.5711	0.0335	0.8512	**3.6466**	**0.0218**	**0.9679**	0.6084	0.0039	0.9981	0.6103	0.0041	0.9977	0.3617	0.0024	0.9991
4	7.5618	0.0428	0.7529	5.6751	0.0358	0.8671	3.7849	0.0232	0.9511	**0.6047**	0.0040	0.9984	0.5937	0.0038	0.9985	0.3607	0.0022	0.9994
5	7.6174	0.0465	0.7752	5.4790	0.0394	0.8705	3.8493	0.0259	0.9572	0.6240	0.0043	0.9975	0.6540	0.0035	0.9986	0.3741	0.0022	0.9995
6	7.4057	0.0428	0.7452	5.5781	0.0365	0.8481	4.0018	0.0245	0.9583	0.6157	0.0048	0.9981	**0.5771**	0.0038	0.9985	0.3684	0.0029	0.9989
7	7.5085	0.0429	0.7474	5.4994	0.0359	0.8664	3.9991	0.0228	0.9573	0.6427	0.0037	0.9980	0.6428	0.0040	0.9982	0.3579	0.0027	0.9992
8	7.5281	0.0438	0.7561	5.8117	0.0401	0.8728	3.8425	0.0224	0.9438	0.6281	0.0039	0.9971	0.6439	0.0032	0.9982	0.3889	0.0025	0.9990
9	7.5317	**0.0419**	0.7619	5.7567	0.0386	0.8549	3.9852	0.0237	0.9547	0.6348	0.0045	0.9975	0.6377	0.0035	0.9987	0.3718	0.0028	0.9987
10	7.6824	0.0488	0.7852	5.6236	**0.0324**	0.8428	4.0005	0.0225	0.9258	0.6234	0.0043	0.9969	0.6800	0.0036	**0.9988**	0.3762	0.0031	0.9988
11	7.6382	0.0453	0.7684	5.6276	0.0372	0.8658	3.7861	0.0226	0.9463	0.6576	**0.0035**	0.9979	0.5834	0.0031	0.9981	0.3789	0.0034	0.9995
12	7.6582	0.0458	0.7519	5.6147	0.0394	0.8539	3.8627	0.0230	0.9571	0.6185	0.0039	0.9981	0.6157	**0.0029**	0.9986	0.3628	0.0021	0.9994
13	7.6282	0.0514	0.7674	5.6845	0.0381	0.8674	3.9543	0.0241	0.9668	0.6258	0.0041	0.9976	0.6229	0.0032	0.9982	0.3784	0.0028	**0.9997**
14	7.7849	0.0472	0.7756	5.6889	0.0388	0.8546	3.8523	0.0250	0.9588	0.6278	**0.0035**	0.9972	0.6308	0.0038	0.9987	0.3517	0.0026	0.9991
15	7.8260	0.0583	0.7438	**5.3752**	0.0387	0.8713	3.9654	0.0247	0.9429	0.6127	0.0037	0.9975	0.6158	0.0035	0.9981	0.3699	0.0028	0.9990
16	7.3589	0.0591	0.7426	5.6112	0.0374	0.8628	4.0120	0.0239	0.9589	0.6477	0.0038	0.9974	0.6294	0.0034	0.9977	0.3785	0.0027	0.9992
17	7.7652	0.0692	0.7854	5.6839	0.0391	0.8527	4.0115	0.0237	0.9583	0.6384	0.0045	0.9981	0.6279	0.0034	0.9978	0.3825	0.0028	0.9989
18	7.8254	0.0583	0.7582	5.8570	0.0388	0.8699	4.3752	0.0255	0.9610	0.6410	0.0040	**0.9985**	0.6374	0.0036	0.9979	0.3749	0.0031	0.9996
19	7.4256	0.0672	**0.7962**	5.8429	0.0375	0.8475	4.3855	0.0265	0.9456	0.6389	0.0039	0.9974	0.6511	0.0037	0.9980	0.3581	0.0024	0.9994
20	**7.3491**	0.0583	0.7593	5.7528	0.0393	0.8517	4.3918	0.0244	0.9429	0.6376	0.0042	0.9979	0.6357	0.0038	0.9981	0.3912	0.0028	0.9995

Table 6 Comparing the MAE, MSE and *R* of different models

Indicators	Models	Average	Min	Max	Std	Median	Lower quantile	Upper quantile
MAE	EKFNN	7.5987	7.3491	7.9772	0.1740	7.5896	7.4768	7.7031
	UKFNN	5.6672	5.3752	5.9010	0.1130	5.6795	5.6029	5.7538
	PFNN	3.9852	3.6466	4.3918	0.1990	3.9753	3.8516	4.0116
	CKS-PFNN	0.6313	0.6047	0.6840	0.0186	0.6280	0.6178	0.6394
	ICS-PFNN	0.6311	0.5771	0.6847	0.2689	0.6333	0.6158	0.6450
	ICKS-PFNN	**0.3686**	**0.3278**	**0.3912**	**0.0145**	**0.3709**	**0.3601**	**0.3784**
MSE	EKFNN	0.0504	0.0419	0.0692	0.0083	0.0469	0.0446	0.0583
	UKFNN	0.0374	0.0324	0.0401	0.0021	0.0378	0.0361	0.0389
	PFNN	0.0240	0.0218	0.0265	0.0012	0.0240	0.0230	0.0247
	CKS-PFNN	0.0040	0.0035	0.0048	0.0003	0.0040	0.0038	0.0042
	ICS-PFNN	0.0036	0.0029	0.0041	0.0003	0.0036	0.0034	0.0038
	ICKS-PFNN	**0.0026**	**0.0020**	**0.0034**	**0.0004**	**0.0027**	**0.0024**	**0.0028**
R	EKFNN	0.7664	0.7426	0.7962	0.0170	0.7647	0.7527	0.7784
	UKFNN	0.8597	0.8428	0.8743	0.0100	0.8589	0.8516	0.8680
	PFNN	0.9494	0.9155	0.9679	0.0149	0.9559	0.9436	0.9584
	CKS-PFNN	0.9977	0.9969	0.9985	0.0004	0.9976	0.9974	0.9981
	ICS-PFNN	0.9982	0.9977	0.9988	0.0003	0.9981	0.9980	0.9985
	ICKS-PFNN	**0.9992**	**0.9987**	**0.9997**	**0.0003**	**0.9993**	**0.9990**	**0.9994**

process prediction model is an important prerequisite. If the deviation between the model and the real system is large, the optimal process point obtained by using the intelligent optimization theory will not be reliable. In the proposed work, the production process model between decision parameters and DC energy consumption is established based on incremental learning algorithm. So, the equipment is endowed with the ability of "self-learning", which is helpful to adjust the operation parameters to the optimal operation states of aluminium electrolysis manufacturing.

5 Conclusion

The proposed ICKS-PFNN algorithm has been successfully applied to the energy consumption dynamic modeling of AEM. This algorithm presents a novel proportional sampling method and uses KFCM instead of FCM to complete the task of particle clustering. Through different models and comparative experiments, the main conclusions of this paper can be given as follows:

1. Aiming at the problem that Kalman filter neural network cannot solve the state estimation of non-Gaussian system, particle filter is used to optimize the neural network.
2. The proportional sampling and KFCM clustering are implemented to the PFNN algorithm, which greatly improves the fitting effect.
3. The ablation study shows that the method of proportional sampling instead of ordinary sampling is more reliable than that of KFCM instead of FCM.
4. The experimental result shows that the complexity of the algorithm does not increase with the improvement of the prediction accuracy.

In the future research, we will further explore the fusion strategy of other filtering theories and deep learning to build a deep network incremental learning model.

Acknowledgements This study was supported by the National Natural Science Foundation of China (No.518 05059), Chongqing Research Program of Basic Research and Frontier Technology under grant (cstc2018jcyjA X0350 and cstc2018jcyjA1663) and Special Project of Technological Innovation and Application Development in Chongqing (No. cstc2019jscx-msxmX0054).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Gui W, Yue W, Xie Y, Zhang H, Yang C (2018) A review of intelligent optimal manufacturing for aluminum reduction production. Zidonghua Xuebao/Acta Automat Sin 44(11):1957–1970
2. Yi J, Huang D, Siyao F, He H, Li T (2016) Multi-objective bacterial foraging optimization algorithm based on parallel cell

entropy for aluminum electrolysis production process. IEEE Trans Ind Electron 63(4):2488–2500
3. Zhou W, Shi J, Yin G, He W, Yi J (2020) Optimal control for aluminum electrolysis process using adaptive dynamic programming. IEEE Access 8(1–1):12
4. Yang C, Zhou L, Huang K, Ji H, Long C, Chen X, Xie Y (2019) Multimode process monitoring based on robust dictionary learning with application to aluminium electrolysis process. Neurocomputing 332:305–319
5. Yi W, Li W, Wang Y, Zhang K (2019) Remaining useful life prediction of lithium-ion batteries using neural network and bat-based particle filter. IEEE Access 7:54843–54854
6. Lebreux M, Desilets M, Allard F, Micheau P, Blais A (2020) An on-line estimation tool for predicting the time-varying ledge profile inside aluminum electrolysis cells. Numer Heat Transf Part A Appl 77(2):134–161
7. Bustillo A, Urbikain G, Perez JM, Pereira OM, Luis N, de Lacalle L (2018) Smart optimization of a friction-drilling process based on boosting ensembles. J Manuf Syst 48:108–121
8. Yi J, Bai J, Zhou W, He H, Yao L (2018) Operating parameters optimization for the aluminum electrolysis process using an improved quantum-behaved particle swarm algorithm. IEEE Trans Ind Inf 14(8):3405–3415
9. Huang K, Wen H, Ji H, Cen L, Chen X, Yang C (2019) Nonlinear process monitoring using kernel dictionary learning with application to aluminum electrolysis process. Control Eng Pract 89:94–102
10. Chen Z, Li Y, Chen X, Yang C, Gui W (2017) Semantic network based on intuitionistic fuzzy directed hyper-graphs and application to aluminum electrolysis cell condition identification. IEEE Access 5:20145–20156
11. Yue W, Chen X, Gui W, Xie Y, Zhang H (2017) A knowledge reasoning fuzzy-Bayesian network for root cause analysis of abnormal aluminum electrolysis cell condition. Front Chem Eng China 11(3):414–428
12. Huang K, Yiming W, Yang C, Peng G, Shen W (2020) Structure dictionary learning-based multimode process monitoring and its application to aluminum electrolysis process. IEEE Trans Autom Sci Eng 17(4):1989–2003
13. Khera N, Khan SA (2017) Prognostics of aluminum electrolytic capacitors using artificial neural network approach. Microelectron Reliab 81(81):328–336
14. Chenhua X, Wang L, Lin X, Li Z, Xin Yu (2016) Intelligent optimization of cell voltage for energy saving in process of electrolytic aluminum. J Adv Comput Intell Intell Inf 20(2):231–237
15. Li J, Zhou P, Pian J (2014) Multi-fault diagnosis of aluminum electrolysis based on modular fuzzy neural networks. Asian J Chem 26(11):3339–3343
16. Zeng S, Bing L (2016) Application of genetic neural network for diagnosis of anode anomaly and metal wave in aluminum electrolysis. Int Conf Artif Intell: Technol Appl 2016:325–328
17. Bak C, Roy AG, Son H (2021) Quality prediction for aluminum diecasting process based on shallow neural network and data feature selection technique. CIRP J Manuf Sci Technol 33:327–338
18. Ding W, Yao L, Li Y, Long W, Yi J, He T (2021) Dynamic evolutionary model based on a multi-sampling inherited hapfnn for an aluminium electrolysis manufacturing system. Appl Soft Comput 99:106925
19. Acosta M, Kanarachos S (2018) Tire lateral force estimation and grip potential identification using neural networks, extended kalman filter, and recursive least squares. Neural Comput Appl 30(11):3445–3465
20. Pesce V, So Silvestrini, Lavagna M (2020) Radial basis function neural network aided adaptive extended Kalman filter for spacecraft relative navigation. Aerosp Sci Technol 96:105527
21. Sassan Goleijani and Mohammad Taghi Ameli (2019) An agent-based approach to power system dynamic state estimation through dual unscented Kalman filter and artificial neural network. Soft Comput 23(23):12585–12606
22. Wang Y, Chai S, Nguyen HD (2019) Unscented Kalman filter trained neural network control design for ship autopilot with experimental and numerical approaches. Appl Ocean Res 85:162–172
23. Yao L, Li T, Li Y, Long W, Yi J (2019) An improved feed-forward neural network based on ukf and strong tracking filtering to establish energy consumption model for aluminum electrolysis process. Neural Comput Appl 31(8):4271–4285
24. Peng K, Jiao R, Dong J, Pi Y (2019) A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter. Neurocomputing 361:19–28
25. Zhang X, Liu D, Lei B, Liang J, Ji R (2021) An intelligent particle filter with resampling of multi-population cooperation. Digit Signal Process 115:103084
26. Qin W, Lv H, Liu C, Nirmalya D, Jahanshahi P (2019) Remaining useful life prediction for lithium-ion batteries using particle filter and artificial neural network. Ind Manag Data Syst 120(2):312–328
27. Cadini F, Sbarufatti C, Corbetta M, Cancelliere F, Giglio M (2019) Particle filtering-based adaptive training of neural networks for real-time structural damage diagnosis and prognosis. Struct Control Health Monitor 26:12
28. Kasantikul K, Yang D, Wang Q, Lwin A (2018) A novel wind speed estimation based on the integration of an artificial neural network and a particle filter using beidou geo reflectometry. Sensors 18(10):3350
29. Wang D, Yang F, Tsui K, Zhou Q, Bae SJ (2016) Remaining useful life prediction of lithium-ion batteries based on spherical cubature particle filter. IEEE Trans Instrum Meas 65(6):1282–1291
30. Wei W, Gao S, Zhong Y, Chengfan G, Gaoge H (2018) Adaptive square-root unscented particle filtering algorithm for dynamic navigation. Sensors 18(7):2337
31. Taifu LI (2014) Kalman artificial neural network with measurable noise estimation by gamma test for dynamic industrial process modeling. J Mech Eng 50(18):29
32. Tai-Fu LI, Yao LZ, Jun YI, Wen-Jin HU, Ying-Ying SU, Jia W (2014) An improved ukfnn based on square root filter and strong tracking filter for dynamic evolutionary modeling of aluminum reduction cell. Acta Autom Sin 40(3):522–530
33. Fazli S, Ghiri SF (2013) Robust fuzzy c-means clustering with spatial information for segmentation of brain magnetic resonance images. Int J Sci Eng Investig 2:12
34. Wang Y, Zhen W, Xia A, Guo C, Chen Y, Yang Y, Tang Z (2019) Energy management strategy for hev based on kfcm and neural network. Concurr Comput Pract Exp 31(10):e4838
35. Khanlari M, Ehsanian M (2017) An improved kfcm clustering method used for multiple fault diagnosis of analog circuits. Circuits Syst Signal Process 36(9):3491–3513
36. Zhang H, Wang S, Xu X, Chow TWS, Jonathan Wu QM (2018) Tree2vector: learning a vectorial representation for tree-structured data. IEEE Trans Neural Netw Learn Syst 29(11):5304–5318
37. Liu Y, Ma S, Du X (2020) A Novel Effective Distance Measure and a Relevant Algorithm for Optimizing the Initial Cluster Centroids of K-means. IEEE Access 2020:3044069
38. Yang X, Zhang G, Jie L, Ma J (2011) A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for

classification problems with outliers or noises. IEEE Trans Fuzzy Syst 19(1):105–115

39. Zhao F, Jiao L, Liu H (2013) Kernel generalized fuzzy c-means clustering with spatial information for image segmentation. Digital Signal Process 23(1):184–199
40. Lin HY (2020) Feature clustering and feature discretization assisting gene selection for molecular classification using fuzzy c-means and expectation maximization algorithm. J Supercomput 77:1–17
41. Zhang D, Chen S (2003) Clustering incomplete data using kernel-based fuzzy c-means algorithm. Neural Process Lett 18(3):155–162
42. Liu Y, Li Z, Xiong H, Gao X, Wu J (2010) Understanding of internal clustering validation measures, pp 911–916
43. Zhe Z, Xiyu L, Lin W (2020) Spectral clustering algorithm based on improved gaussian kernel function and beetle antennae search with damping factor. Comput Intell Neurosci. https://doi.org/10.1186/s12859-018-2402-0
44. Fashoto SG, Mbunge E, Ogunleye G, Burg J (2021) Implementation of machine learning for predicting maize crop yields using multiple linear regression and backward elimination. Malays J Comput 6(1):679–697
45. El Genidy M (2019) Multiple nonlinear regression of the Markovian arrival process for estimating the daily global solar radiation. Commun Stat Theory Methods 48(22):5427–5444
46. Moayad A, Weishi S, Xumin L, Qi Y (2020) A bayesian learning model for design-phase service mashup popularity prediction. Expert Syst Appl 149:113231
47. Gu Y, Lu W, Xu X, Qin L, Zhang H (2019) An improved Bayesian combination model for short-term traffic prediction with deep learning. IEEE Trans Intell Transp Syst 99:1–11
48. Aladag CH, Yolcu U, Egrioglu E (2013) A new multiplicative seasonal neural network model based on particle swarm optimization. Neural Process Lett 37(3):251–262
49. Li W, Kong D, Jinran W (2017) A novel hybrid model based on extreme learning machine, k-nearest neighbor regression and wavelet denoising applied to short-term electric load forecasting. Energies 10(5):694