Applied Intelligence (2022) 52:17387–17409
https://doi.org/10.1007/s10489-022-03314-9

A multiobjective prediction model with incremental learning ability by developing a multi-source filter neural network for the electrolytic aluminium process

Lizhong Yao[1,2] · Wei Ding[3] · Tiantian He[4] · Shouxin Liu[5] · Ling Nie[1]

Accepted: 27 January 2022 / Published online: 1 April 2022

Abstract

Improving current efficiency and reducing energy consumption are two important technical goals of the electrolytic aluminum process (EAP). However, because the process involves complex noise characteristics (i.e., unknown types, redundant distributions and variable forms), it is very difficult to accurately develop a multiobjective prediction model. To overcome this problem, in this paper, a novel framework of multiobjective incremental learning based on a multi-source filter neural network (MSFNN) is presented. The proposed framework first presents a "multi-source filter" (MSF) technique that utilizes the mean and variance in the unscented Kalman filter (UKF) to guide the importance function of the particle filter (PF) based on a density kernel estimation method. Then, the MSF is embedded in the mutated neural network to adjust weights in real time. Third, weights are calculated and normalized by a modified importance function, which is the basis for further optimizing a secondary sampling based on sampling importance resampling (SIR). Finally, the incremental learning model with two objectives (i.e., process power consumption and current efficiency) based on the MSFNN in the EAP is established. The presented framework has been verified by the real-world EAP and some closely related methods. All test results indicate that the MSFNN's relative prediction errors of the above two objectives are controlled within 0.51% and 0.38%, respectively and prove that MSFNN has significant competitive advantages over other recent filtering network models. Successfully establishment of the proposed framework provides a model foundation for multiobjective optimization problems in the EAP.

Keywords Multi-source filter · Unscented Kalman filter · Multiobjective problem · Neural network · Electrolytic aluminium process

Nomenclature and Abbreviations

NN	Neural network
KF	Kalman filter
PF	Particle filter
EKF	Extended Kalman filter
UKF	Unscented Kalman filter
MSF	Multi-source filter
MOP	Multiobjective problem
UT	Unscented transformation
EAP	Electrolytic aluminium process
SIR	Sampling Importance Resampling
PFNN	Particle filter neural network
BPNN	Back-propagation neural network
UKFNN	Extended Kalman filter neural network
EPFNN	Extended particle filter neural network
UPFNN	Unscented particle filter neural network
MSFNN	Multi-source filter neural network

1 Introduction

It is well known that the electrolytic aluminium industry holds an important strategic position worldwide [1]. However, the industry is characterized by high power consumption and high pollution. Investigations show that producing one ton of electrolytic aluminium will emit nearly 1500 m^3 of polluting gas into the atmosphere and consume approximately 500 kg of carbon anodes [2]. Research on energy-saving and emission-reduction technology in the electrolytic aluminium process (EAP) has significant engineering application value.

✉ Wei Ding
3120215245@bit.edu.cn

Extended author information available on the last page of the article.

Currently, on the premise of ensuring the stable production of electrolytic aluminium cells, improving current efficiency and reducing energy consumption have become important goals for electrolytic aluminium enterprises. To achieve the above goals, the main research in this field has been focused on the following two aspects: (1) Improving process equipment, such as shaped cathodes and perforated anodes. For example, Peng et al. [3] analyzed the method of improving the current efficiency of a Hall-Heroult cell by using a novel rectangular protruding cathode and process parameters in the EAP. (2) Establishing system models based on data mining technology and using a reasonable and effective filtering method to improve the model accuracy. For instance, Yao et al. [4] used the Kalman filter to establish the dynamic evolution model of the EAP in a Gaussian noise environment. Because the existing industrial aluminium electrolysis cell superheat identification mainly depends on manual experience, the accuracy is far from satisfactory, and a deep soft sensor method for superheat degree detection was proposed by Lei et al. [5].

The above two aspects are helpful for realizing the energy savings and emission reduction of the EAP. However, it should be noted that the first method is meant to improve equipment that is closely related to EAP data, such as measurement and transmission equipment. The implementation of these objectives is difficult and often consumes substantial financial and material resources, which are more applicable for new production systems. The second method predicts the state of process parameters based on the system model and filtering algorithm, which does not need to change the existing production equipment. Moreover, the internal information about real data that has been covered by complex noise can be further explored. Furthermore, since neural networks (NNs) still have an excellent nonlinear mapping ability to fit a large amount of data under the condition of an unknown system modeling mechanism [6], the application of NN modeling is an effective method. It does not need to understand the internal mechanism of EAP, and the mapping relationship between decision variables and industrial indicators can be obtained by learning and training a series of process data. However, once the traditional NN is trained, its model parameters cannot be further dynamically updated. The combination of an NN and new filtering algorithms is expected to enhance the ability to optimize the process model online. Yi et al. [7] proposed a dynamic prediction model based on false nearest neighbors and an UKFNN to determine the alumina concentration. Li et al. [8] presented a method that uses an improved UKFNN and NSGA-II algorithm to obtain the optimal output of stable operating variables in the EAP. A modular integrated fuzzy neural network was developed for predicting multiple fault diagnoses of the EAP by Li et al. [9]. The above studies have established a single objective prediction model in the EAP. However, current efficiency and DC power consumption are two main technical and economic indicators in a real electrolytic aluminium equipment process system. The NN modeling of two objectives provides a model foundation for realizing collaborative optimization.

Based on the above analysis, one of the main directions of energy savings and emission reduction is to establish a multiobjective prediction model that not only has high prediction accuracy but also can minimize unit power consumption and maximize current efficiency in the EAP. However, the EAP contains a series of physical and chemical reactions, and there are various internal and external parameters that present a complex coupling interaction, so that the EAP involves some complex noise characteristics (i.e., unknown types, redundant distributions and variable forms). The above problems make it difficult to establish a multiobjective prediction model for the EAP. Moreover, the EAP is extremely susceptible to interference from uncertain factors such as Gaussian noise or non-Gaussian noise while collecting a series of decision parameters such as series current, cell voltage, cell temperature, etc., which seriously affects the accuracy of the prediction model. Therefore, minimizing the noise interference in the model algorithm has become a feasible breakthrough for further improving model prediction accuracy [2].

However, the aforementioned studies did not consider the characteristics of mixed noise in the EAP, and usually only used a single filtering method to estimate the parameters of the NN's weights and thresholds. These investigations lack a discussion on the filtering prediction problem with complex and unknown system noise, and thus are not conducive to mining the model's prediction potential in depth. For instance, previous research studies have shown that combining the Kalman filter (or improved Kalman filter) and an NN may cause modeling failure because the noise is not limited to linear or Gaussian characteristics [10, 11]. Additionally, it has been demonstrated that a combination of a particle filter (PF) [12, 13] and NN can solve model problems with non-linear and non-Gaussian noise [14–16]. However, particle degradation in the PF may lead to algorithm divergence after several iterations. Therefore, the required state estimation cannot be obtained. To solve the above problems, in this paper, a "multi-source filter" technique is proposed, which uses the mean and variance in the UKF to adjust the PF's importance function based on the density kernel estimation method. Then, the NN's model parameters (i.e., weights) can be viewed as state variables of the filtering algorithm, and its outputs can be viewed as measurement variables, which give the above strategy a significant advantage of adaptively adjusting the state estimation under various mixed noise interferences. Finally, a multiobjective incremental learning prediction model that meets the production requirements of the EAP is established, which helps to significantly reduce power consumption and improve current efficiency in the EAP.

Through the aforementioned comprehensive analysis, some important contributions of this study can be summarized as follows:

(1) To solve the interference problem of mixed noise on the model accuracy, a "multi-source filter" technique that can be applied to the model's parameter estimation under various noise characteristics is proposed.

(2) Considering the dynamic performance of the model, this paper combines the "multi-source filter" and a NN to establish an incremental learning prediction model.

(3) To reduce particle degradation in the algorithm, this paper adopts the mean and variance of the UKF to optimize the PF's importance function based on a density kernel estimation.

(4) On the basis of the above research findings, in this article, a multi-source filter neural network (MSFNN) framework is developed, and its corresponding construction process is provided.

(5) The new framework is applied to the modeling of the EAP. The experimental results show that the MSFNN can accurately predict the current efficiency and power consumption data in real time.

The remainder of this paper is organized as follows: Section 2 gives a clear problem description encountered in the modeling process of electrolytic aluminium. Based on the NN's state-space model, Section 3 presents the "multi-source filter" technique and states the process design, theoretical analysis and implementation steps of the new framework (MSFNN) in detail. In Section 4, the framework developed in this paper is applied and verified in the EAP. Section 5 provides a summary.

2 Problem description

In the process manufacturing industry [17], system models are often required to demonstrate accurate prediction performance and an excellent incremental learning capability. However, an industrial process system usually has many characteristics, such as complex and changing environments, multiple alternating processes and strong coupling among parameters. In addition, the production mechanism is often vague and difficult to quantify. Facing the above-mentioned complex conditions, although supervised machine learning algorithms are popular for establishing process models to predict technical indicators of a real technological process [18, 19], the established process system models still have great development potential and can be developed further.

For ease of description, an industrial process system is defined as follows:

$$\begin{cases} x_k = f(x_{k-1}, u_k) + \theta_k \\ y_k = h(x_k, u_k) + \nu_k \end{cases} \tag{1}$$

where x_k represents variables (decision variables) of the industrial process system to be estimated at moment k; u_k and y_k respectively represent input variables and output variables in the process system at moment k; θ_k and ν_k represent the process noise and measurement noise (not necessarily consistent with Gaussian noise), respectively. The functions f and h represent the relationship of the effective variables with time change.

Because the process noise θ_k and measurement noise ν_k in the above-mentioned process system often have some characteristics, such as unknown types and redundant distributions, the Kalman filter (KF) and particle filter (PF) [20] are mostly used to estimate the state of decision variables directly in order to establish an accurate prediction model.

Since the traditional KF algorithm can only be applied to linear systems, research scholars have expanded its application scope and proposed two improved filtering technologies, such as the extended Kalman filter (EKF) [21, 22] and unscented Kalman filter (UKF) [23, 24]. However, the above two strategies are restricted by the condition of a non-linear normal distribution. It can be seen from the non-Gaussian distribution system model in Fig. 1 that the probability distribution is represented as a complex curve, which is composed of multiple Gaussian curve segments with multiple peaks and valleys. In terms of properties, it is not just a fusion of several similar Gaussian distributions, which cannot be characterized by simple means and variances. In related studies [12, 13] on the above issues, the effect of the PF algorithm depends on the establishment of the importance function and the choice of the resampling method. Because the PF algorithm has the advantage of not requiring mandatory constraints on system state variables, it is an "approximately optimal" tool used to solve the problem of state parameter estimation for non-linear non-Gaussian manufacturing systems. This shows that manufacturing systems with different characteristics need to adopt different filtering methods. If the industrial process system meets the operating characteristics of linear Gaussian white noise, then the KF algorithm is preferred. If the industrial process system belongs to the category of a non-linear Gaussian distribution, then it is necessary to comprehensively choose a method with better performance according to the calculation size of different filtering algorithms and the accuracy of state estimation. If the process system conforms to the non-linear and non-Gaussian properties, then the particle filter technique is preferred.

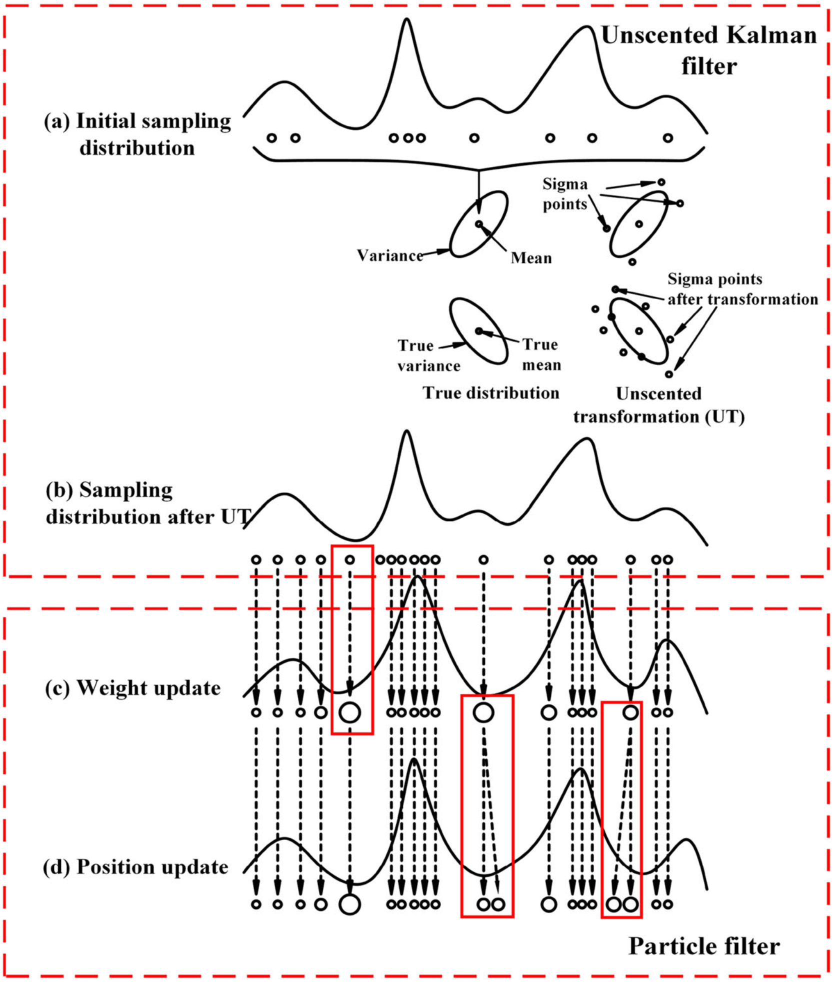


Fig. 1 Process design of the MSF technology

Table 1 lists the applicable range of various filtering technologies, from which we can see that the PF algorithm has the widest range of applicability. However, with the gradual development of the PF field, researchers have found that the PF may not be the best filtering technique when using it to approximately estimate all state parameters in industrial manufacturing systems. As the particles degenerate, the weights of most particles will decrease during the process of particle updating. The above phenomenon indicates that if the iterative update is continued, the weight updating exhibits no obvious improvement in the final filtering accuracy. Instead, the filter resources are exhausted to deal with the negligible particle calculation update. There are two methods to solve particle degradation: one is to increase the number of sampling particles, which may lead to the divergence of the PF algorithm itself. The other is to optimize the importance function to make it closer to the real distribution function.

To solve the above problems, inspired by previous studies [25–27], this paper proposes the "multi-source

Table 1 Applicability range of various filtering technologies

State equation	Observation equation			
	Linear Gaussian	Linear Non-Gaussian	Non-Linear Gaussian	Non-Linear Non-Gaussian
Linear Gaussian	KF	PF	EKF/UKF/PF	PF
Linear Non-Gaussian	PF	PF	PF	PF
Non-Linear Gaussian	EKF/UKF/PF	PF	EKF/UKF/PF	PF
Non-Linear Non-Gaussian	PF	PF	PF	PF

filter" (MSF) technique, which utilizes the UKF's mean and variance to guide the PF's importance function based on the density kernel estimation method. This method not only inherits the characteristics and application range of the two filtering methods, but also can solve the problem of particle degradation in the PF. Therefore, the MSF technique ignores the influence of noise properties and overcomes the disadvantages of the PF algorithm, so it can be widely used in the estimation of state parameters with various single noise or mixed noise interferences in process manufacturing systems.

To clearly describe this method, Fig. 1 shows the basic process of MSF parameter estimation. This process includes the UKF segment and PF segment, which interact with each other through the adjustment of particles. From Fig. 1, the particles are processed by the unscented transformation (UT) method, and then the sampling distribution in step(b) is obtained by the density kernel estimation method after UKF optimization. Next, the PF method is used to update the particles on this basis. Figure 1(a) represents the initial sampling distribution. Figure 1(b) represents the sampling distribution after the UT method. Figure 1(c) represents the probability distribution after the particle weight is updated, and Fig. 1(d) is the probability distribution after the particle position is updated.

The process from Fig. 1(a) to Fig. 1(b) is mainly based on the UKF segment. First, a set of sample points (represented by the circle in the figure) are randomly generated from the prior distribution of the initial state space, and then "Sigma points" are calculated from the above sample points by using the UT method in the UKF. Finally, the mean and variance of these "Sigma points" are substituted for the real mean and variance to continuously adjust the sampling distribution. The following process from Fig. 1(b) to Fig. 1(c) shows that the PF's importance function is adjusted by the mean and variance. The particles are sampled from the optimized importance function, and the weights of particles are constantly adjusted (shown as the change in the circle size in the figure) by using the measured data to modify the distribution. Finally, the weights are mapped to the probability distribution curve. Fig. 1(c) and 1(d) show that the particles in Fig. 1(c) are split to change the particles' position (indicated as a circle from one to two in Fig. 1(d)), so as to obtain the final modified probability distribution.

Considering that the real-time internal and external data in the EAP are frequently exchanged and constantly changed [28], in order to ensure that the system exhibits good dynamic performance, the above theory is combined with an NN to predict the current efficiency and unit power consumption. The NN's model parameters (i.e., weights) can be viewed as the state variables of the MSF, and its outputs can be viewed as measurement variables of the MSF. Then, in turn, the performance of the incremental learning model can be tested by the above NN. Finally, the perfect incremental learning prediction model for the Multiobjective problem (MOP) [29, 30] (i.e., unit power consumption and current efficiency in an EAP) is established.

In Fig. 2, we present the dynamic evolutionary process of the probability density distribution for two objectives in the incremental learning prediction model. Fig. 2(a) shows the process of updating the probability density distribution of unit power consumption with time in the EAP; Fig. 2(b) shows the process of updating the probability density distribution of current efficiency with time in the EAP. The above two figures reflect that the MOP prediction model established by MSF theory has an incremental learning ability, so that the model can evolve dynamically to predict the technical power consumption and current efficiency with time and sample changes in the EAP.

Therefore, in order to fully tap the complementary advantages of the MSF and NN in the MOP, in this paper, a MSFNN is designed that can typically solve adaptive modeling problems with unknown mixed noise by deeply merging the MSF and NN. The process design, theoretical analysis and implementation steps of the new framework (MSFNN) will be presented in the next section.

3 Design of the MSFNN algorithm

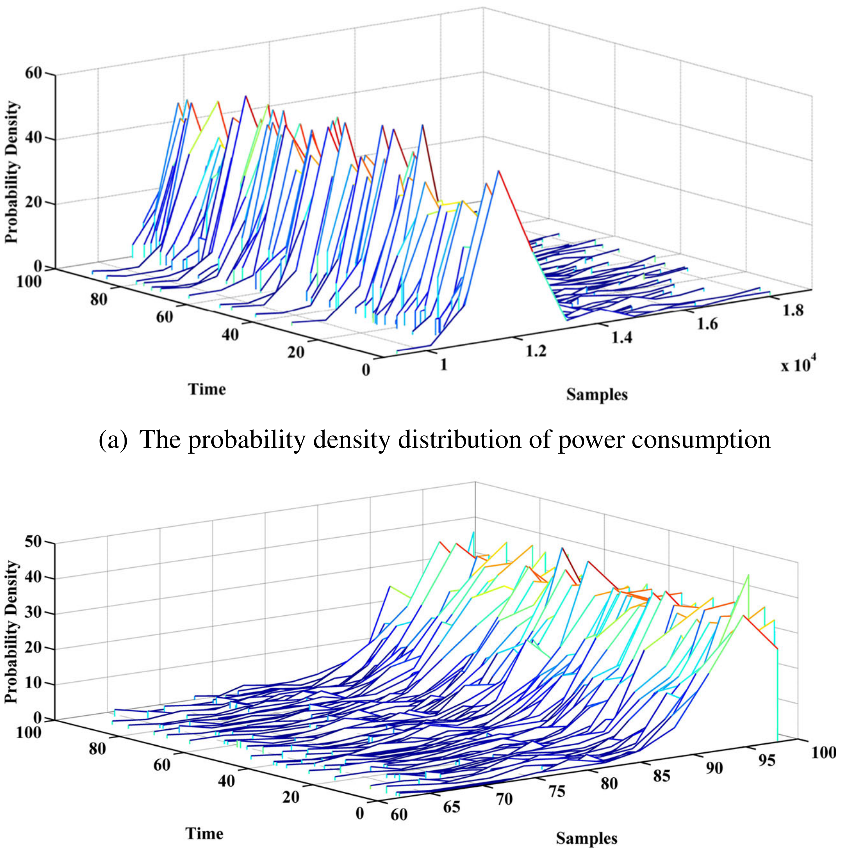
3.1 State-space model of the neural network

The state-space representation based on the NN describes the updating process of the back-propagation neural network's (BPNN's) weights and thresholds with time [31]. The above process includes using both a state equation to describe the change in the weights and thresholds and a measurement equation to describe the non-linear relationship between the inputs and outputs of the model. The specific equation is as follows.

$$\begin{cases} \omega_k = \omega_{k-1} + \theta_k \\ y_k = h(\omega_k, u_k) + \nu_k \end{cases} \tag{2}$$

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 EAP at moment k; y_k represents the measurement variables at moment k (i.e., the output variables to evaluate the advantages and disadvantages of the industrial process system). Assume that the system measurement noise ν_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

Fig. 2 The dynamic evolutionary process of the MSF-MOP prediction model



(a) The probability density distribution of power consumption

(b) The probability density distribution of current efficiency

the measurement noise v_k mainly describes the modeling error caused by sensors and other devices in the system.

The non-linear measurement function $h(\cdot)$ is approximated by using a multilayer perceptron.

$$h_l(\omega, u) = \sum_{j=1}^{9} \frac{\omega_{jk}^{**}}{1 + exp\left[-\left(\sum_{i=1}^{9} \omega_{ij}^{*} u_i + a_j\right)\right]} + b_l \qquad (3)$$

where 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_l represents the thresholds of output layer neurons; u_i is the input variable.

3.2 Multi-source filter technique

The existing filtering theory takes the state-space model of any system as the research object. Under the premise of the known measurement value, the parameter estimation of the state variable is carried out by rigorous mathematical derivation, and the error between the state value of the estimated system and the real value of the corresponding system is finally within the allowable range [32]. However, because the types of noise are unknown, the distribution is complicated, and the forms are variable in the actual process, the existing single filtering methods [33] have limited the applicability and lowered the accuracy, which cannot solve the problem of mixed noise.

To solve the modeling problem of process manufacturing systems in an environment with mixed noise, this paper proposes the MSF technique. The MSF utilizes the mean and variance in the UKF to guide the importance function of the PF based on the density kernel estimation method [34, 35], and it can be embedded in any state model to perform a probability estimation of state variables. Then, the “Sigma points” in the UKF are employed to update the model at every moment. Finally, the weights are calculated and normalized by the modified importance function, and whether to perform sampling importance resampling (SIR) is judged by the number of effective particles, so as to achieve an accurate estimation of the state parameters (decision parameters) of the process system.

The above theory can be applied to the estimation of state parameters under various noise interference conditions and improve the filtering accuracy. The main advantages are as follows:

(1) The method extracts particles from the probability distribution established by the initial values, so it is applicable to the different characteristics of initial states.

$$x \sim q(x_k|x_{0:k-1}) \tag{4}$$

where x represents particles; $q(x_k|x_{0:k-1})$ represents the probability of the state variable at moment k under the premise that the state data are known at moment k-1.

(2) The method of constructing a distribution function based on density kernel estimation is used to expand the application range of the filtering algorithm. This method gets rid of the previous filtering problem that the importance function is almost always represented by the Gaussian distribution $N(\bar{x}, \sigma)$ established by the mean $\bar{x}$ and variance σ of samples. In the proposed MSF, it is only necessary to simulate the desired distribution as the optimal importance function through a set of random particles carrying weights.

$$x \sim q(x_k|x_{0:k-1}, z_{0:k}) = N(\bar{x}, \sigma) \tag{5}$$

$$x \sim q(x_k|x_{0:k-1}, z_{0:k}) = F(\boldsymbol{x}) \tag{6}$$

where x represents the particles after sampling; $\boldsymbol{x}$ represents the particle set before sampling; $q(x_k|x_{0:k-1}, z_{0:k})$ represents the posterior probability of the state variable at moment k under the premise that the state data are known from moment 0 to k-1 and observation data are known from moment 0 to k. F is the distribution function constructed based on density kernel estimation.

Equation (5) shows that the importance function $q(x_k| x_{0:k-1}, z_{0:k})$ was approximated by the mean $\bar{x}$ and variance σ, but the posterior distribution cannot often be represented by only a Gaussian curve. In this paper, the distribution function F in (6) is used to replace the normal distribution N in (5) as the importance function.

(3) To improve the algorithm accuracy, this method optimizes the importance function by utilizing the mean and variance obtained in the UKF, as shown in Fig. 3.

It is assumed that the curve's expression in Fig. 3(a) is $f(x) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(x-\mu)^2}{2\sigma^2})$. Where, μ is the mean and σ is the variance; x is the sampled particle and $f_{max}(x)$ is the maximum probability density. The Gaussian model in Fig. 3(a) is established based on the mean μ and variance σ, which are updated by the UKF. The red vertical dotted line in Fig. 3 represents the symmetry axis of the Gaussian model. The green curve in Fig. 3(b) is the importance function established by the density kernel estimation method, from which we can see that it is a non-linear non-Gaussian curve. The green vertical dotted line, which represents the expectation of the importance function, divides the area enclosed by the green curve and the black line into two equal parts. In Fig. 3(c), the red vertical dotted line is on the left side of the green vertical dotted line, so the green curve should move toward the red curve to meet the requirements of the average expected value. Fig. 3(d) shows the importance function updated by the mean. Since the variance σ represents the distribution degree of all sampled particles, the importance function in Fig. 3(d) can be further optimized. The smaller the variance is, the more concentrated the distribution is, which makes the original importance function (red curve) move to the position of the yellow curve.

3.3 Design and analysis of the MSFNN algorithm

While modeling a process system with an unknown mechanism, the NN still has the ability to fit a large amount of non-linear process data, and further approximate the operation mode of a real process system. However, when a conventional NN constructs the process operation model of the industrial system, it is often assumed that the internal states of the process system and the interference of the external production environment are stable. In fact, the process system continuously exchanges materials, energy, and information with the external environment, making it difficult for the static NN to adapt to environmental change when modeling the process system.

To make full use of the complementarity between the MSF and NN, this study proposes a MSFNN framework. This MSFNN integrates the powerful non-linear fitting ability of the NN while using MSF theory to forecast the NN's model parameters. Specifically, the model parameters (i.e., weights) act as state variables of the MSF. Furthermore, the predicted outputs of the process model act as the measurement variables of the MSF.

Taking the neural network state-space model established by (2) as the research object, the main steps of the MSFNN algorithm are as follows:

(1) Initialization.

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

$$\bar{\omega}_0^{i(a)} = E(\omega_0^{i(a)}) \tag{7}$$

$$P_0^{i(a)} = E[(\omega_0^{i(a)} - \bar{\omega}_0^{i(a)})(\omega_0^{i(a)} - \bar{\omega}_0^{i(a)})^T] \tag{8}$$

where $\bar{\omega}_0^{i(a)}$ represents the mathematical expectation (mean) of particles; $P_0^{i(a)}$ represents the variance matrix of particles. The superscript number represents the particle sequence, and the subscript number represents the time sequence.

(2) Update each particle with the UKF at each moment as follows:

a. Calculate the Sigma points of each particle.

$$\chi_{k-1}^{i(a)} = [\bar{\omega}_{k-1}^{i(a)}, \bar{\omega}_{k-1}^{i(a)} \pm \sqrt{(n_a + \lambda)P_{k-1}^{i(a)}}] \tag{9}$$

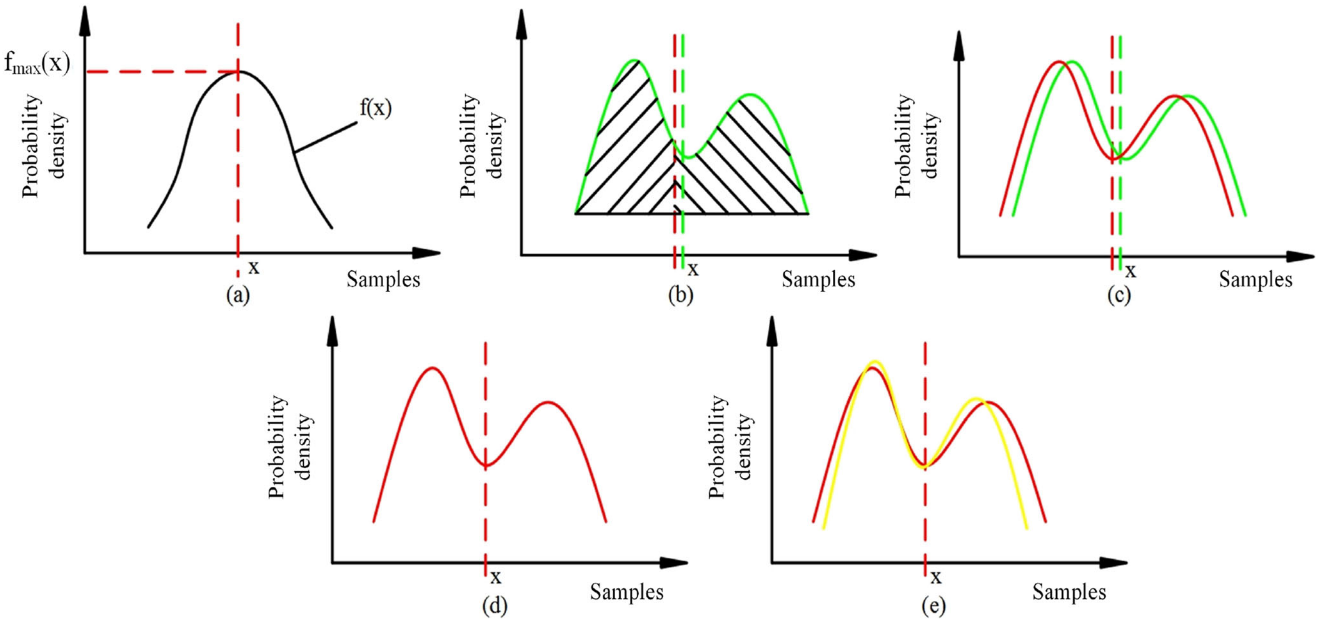


Fig. 3 The optimization process of the importance function in the MSF algorithm

where $\lambda = \alpha^2(n_x + \kappa) - n_x$ is the proportional coefficient, and the size of α determines the distribution of the selected sample points around the mean $\bar{\omega}$. In particular, lowering α can reduce higher-order effects to a greater extent; κ, n_x and n_a are the setting parameters in the UKF.

b. Introduce particle recursion (time update).

$$\chi^{i(x)}_{k|k-1} = f(\chi^{i(a)}_{k-1}) \tag{10}$$

$$\bar{\omega}^{i(x)}_{k|k-1} = \sum_{j=1}^{2n_a} w^m_j \chi^{i(x)}_{j,k|k-1} \tag{11}$$

$$P^i_{k|k-1} = \sum_{j=1}^{2n_a} w^c_j (\chi^{i(x)}_{j,k|k-1} - \bar{\omega}^{i(x)}_{k|k-1}) \times (\chi^{i(x)}_{j,k|k-1} - \bar{\omega}^{i(x)}_{k|k-1})^T \tag{12}$$

$$y^i_{k|k-1} = h(\chi^{i(x)}_{j,k|k-1}) \tag{13}$$

where χ is the sampling point obtained by UT method; $\chi^{i(a)}_{k-1}$ is the original sampling point; $\chi^{i(x)}_{k|k-1}$ is the sampling point obtained by symmetrically distributed sampling; w^m_j and w^c_j are the weights corresponding to the j-th sampling point, respectively. According to (1) and (2), it can be known that $f(\omega) = \omega$ and $h(\omega) = \sum_{j=1}^{9} \frac{\omega^{**}_{jk}}{1+exp\left[-\left(\sum_{i=1}^{9} \omega^*_{ij} u_i\right)+a_j\right]} + b_l$.

c. Calculate new measurement values (measurement update).

The mean $\bar{\omega}^i_k$ and variance $\hat{P}^i_k$ of the statistics y are calculated as follows.

$$\bar{y}^i_{k|k-1} = \sum_{j=1}^{2n_a} w^m_j y^i_{j,k|k-1} \tag{14}$$

$$P_{z_k z_k} = \sum_{j=1}^{2n_a} w^c_j (y^i_{j,k|k-1} - \bar{y}^i_{k|k-1}) \times (y^i_{j,k|k-1} - \bar{y}^i_{k|k-1})^T \tag{15}$$

$$P_{x_k z_k} = \sum_{j=1}^{2n_a} w^c_j (\chi^i_{j,k|k-1} - \bar{\omega}^i_{k|k-1}) \times (y^i_{j,k|k-1} - \bar{y}^i_{k|k-1})^T \tag{16}$$

$$K_k = P_{x_k z_k} P^{-1}_{z_k z_k} \tag{17}$$

$$\bar{\omega}^i_k = \bar{\omega}^i_{k|k-1} + K_k(y_k - \bar{y}^i_{k|k-1}) \tag{18}$$

$$\hat{P}^i_k = P^i_{k|k-1} - K_k P_{z_k z_k} K^T_k \tag{19}$$

d. Use a method based on density kernel estimation to construct the important function $q(\omega^i_k | x^i_{0:k-1}, y_{1:k}) = F(\cdot)$, and then utilize the mean $\bar{\omega}^i_k$ and variance $\hat{P}^i_k$ in the UKF to optimize the importance function of the PF $q(\omega^i_k | x^i_{0:k-1}, y_{1:k}) = \hat{F}(\cdot)$.

e. Calculate weights and normalize.

$$w_k^i = w_{k-1}^i \frac{p(y_k|\bar{\omega}_k^i)p(\bar{\omega}_k^i|\omega_{k-1}^i)}{q(\bar{\omega}_k^i|\omega_{0:k-1}^i, y_{1:k})} \tag{20}$$

$$\tilde{w}_k^i = w_k^i / \sum_{j=1}^{N} w_k^j \tag{21}$$

(3) SIR secondary resampling.

$$N_{eff} = 1 / \sum_{j=1}^{N} \tilde{w}_k^j \tag{22}$$

If $N_{eff} \leq N_{th}$ (N_{th} is a set threshold, generally taken as $N/3$), it means that the weights of the particles have been seriously degraded, so the residual resampling [36, 37] is needed; otherwise, it goes directly to the next step.

(4) $k = k + 1$, go to step (2).

The above steps are the process of the MSFNN algorithm. (The pseudocode of the MSFNN is given in Appendix A.) It can be seen that the MSFNN establishes an important function $F(\cdot)$, which gets rid of the limitation of the Gaussian model and reduces the algorithm sensitivity to mixed noise. The accurate establishment of the important function $F(\cdot)$ is a critical part of the incremental learning model applied to non-linear non-Gaussian systems.

On the convergence of the research framework, the MSFNN takes a NN as the basic model and adopts the UKF's mean and variance to optimize the PF's importance function based on the density kernel estimation method. Therefore, the NN's convergence performance is not changed [38]. Moreover, the convergence characteristic of the MSF algorithm depends on the UKF and PF. To simplify and clarify the discussion, the convergence analysis of the MSFNN can be found in Appendix B.

This section systematically presents the framework of the MSFNN incremental learning model through an in-depth analysis of important links in the model construction process and integrates MSF theory, a NN model, and density kernel estimation.

Figure 4 graphically shows the flow of the MSFNN incremental learning algorithm. First, the MSFNN algorithm needs to initialize the model parameters. Second, the UT method is performed near the estimated points, and the Sigma point sets are calculated. Third, these Sigma points are updated with time and measurement values. Then, the density function F is obtained through the density kernel estimation method, which is modified and optimized by means of the mean and variance in the UKF. Finally, the parameter estimation value is imported into the NN to test the performance of the model. If the system's sample increases or decreases (i.e., the inputs or outputs are changed), the model can adaptively update the NN's model parameters to achieve a new dynamic balance.

In Fig. 4, the red font represents the main contribution and innovation of this paper, and the blue virtual boxes represent important modules of this proposed method. Among these modules, i, ii, iii and iv respectively represent updating sigma points by UKF, constructing importance function F, updating model parameters by PF and testing incremental learning model performance. The MSFNN algorithm proposed in this paper performs deep optimization of the traditional BPNN model construction algorithm, mainly including:

(1) The traditional BPNN belongs to the category of static modeling. In contrast, the MSFNN uses a dynamic modeling mechanism, which can adjust the model parameters adaptively as the external or internal environment changes.

(2) To make the model suitable for parameter estimation under mixed noise characteristics, the technique of the MSF is proposed and combined with the NN first.

(3) To reduce the influence of various noise on the algorithm for improving the estimation accuracy of NN's parameters, this paper adopts a method utilizing the UKF's mean and variance to guide the PF's importance function based on the density kernel estimation method, so that the probability density function obtained from the state estimation can better tend to the real density function.

4 Multiobjective incremental learning model based on the MSFNN in the electrolytic aluminium equipment process system

To ensure that the industrial process system has an accurate prediction performance and a good incremental learning ability, the above algorithm can be applied to an EAP system [39]. The main steps in establishing a multiobjective incremental learning model of the EAP based on the MSFNN algorithm are as follows:

Step. 1: Import the input and output data of the EAP into the BPNN model to obtain the initial model parameters;

Step. 2: Build a basic process model, as described in (2), based on the NN principle;

Step. 3: The NN's model parameters (i.e., weights and thresholds) are taken as particles, and then an iterative loop is performed according to the MSFNN algorithm ((7) to (22)) to obtain a new round of model parameters;

Step. 4: The newly obtained model parameters are imported into the BPNN model to test whether it meets the expected prediction result. If not, the above weights and thresholds are regarded as the particles in the new round of the MSFNN algorithm to continue to iteratively update until the expectation is met.

This paper uses the MSFNN as the theoretical framework to establish a multiobjective incremental learning prediction

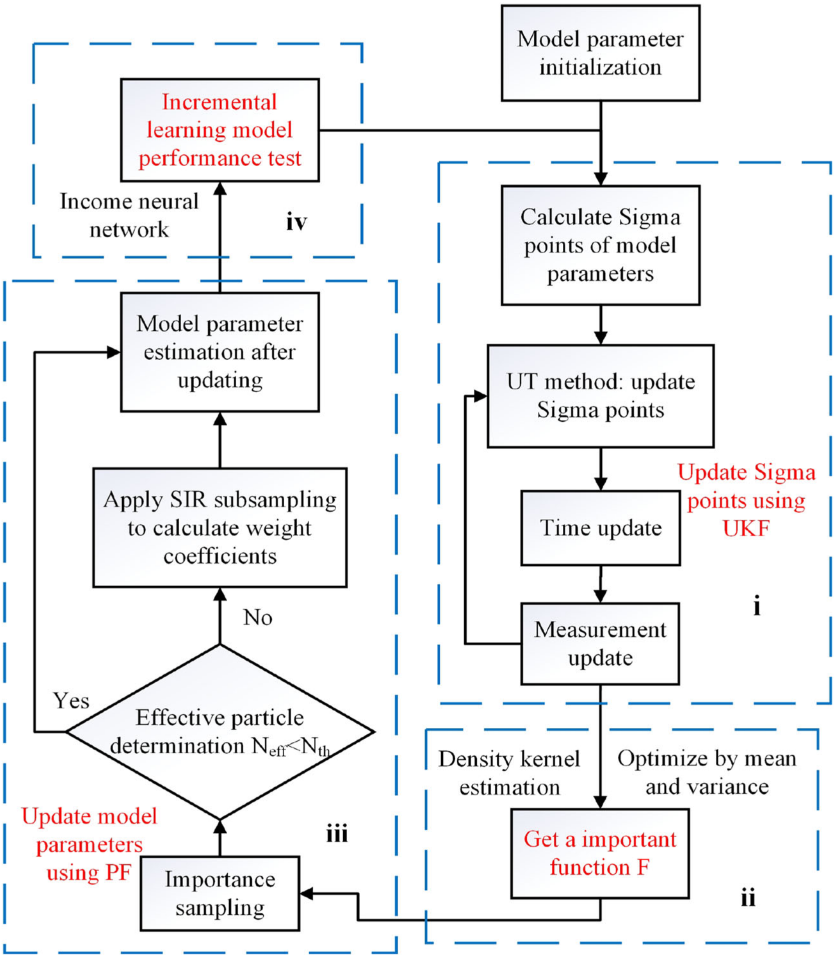

Model parameter initialization
Incremental learning model performance test
Income neural network
iv
Calculate Sigma points of model parameters
UT method: update Sigma points
Update Sigma points using UKF
Time update
Measurement update
i
Model parameter estimation after updating
Apply SIR subsampling to calculate weight coefficients
No
Yes
Effective particle determination N_eff<N_th
Update model parameters using PF
iii
Importance sampling
Density kernel estimation
Optimize by mean and variance
Get a important function F
ii

Fig. 4 Algorithm flow of the MSFNN incremental learning model

model for the unit power consumption and current efficiency of the EAP. The BPNN's weights and thresholds are estimated by the MSF, which enables the production model of the EAP to have good adaptability, accurate predictive ability and a wide application range.

4.1 Experiment object and model parameters

In this paper, industrial experiments based on an electrolytic aluminium cell combining a shaped cathode and perforated anode are carried out [4], as shown in Fig. 5. In the figure, f1 represents the current efficiency, and f2 represents the power consumption of electrolytic aluminium. Ideally, the power consumption should be as low as possible, and the current efficiency should be as high as possible.

However, the process system of electrolytic aluminium equipment is complex and has the following characteristics: nonlinearity, multiple parameters, strong coupling and noise redundancy. Moreover, it is accompanied by several operation links, such as anode changing, bus lifting, shell punching and aluminium discharging [40, 41]. It is difficult to obtain an accurate multiobjective incremental learning model using traditional modeling methods in the EAP. Fortunately, the proposed MSFNN algorithm can be applied to an environment with various complex noise, and it can update and track the real-time status of multiple targets in the EAP, which has the potential to obtain a high-precision process model.

By analyzing the operating variables related to the two goals (i.e., current efficiency and power consumption) in the EAP, leveraging expert knowledge and a data acquisition system, nine main operating variables and two predicted objectives are acquired and are listed in Table 2. To facilitate k-fold cross validation, all 780 groups of sample data were collected from device No. 160 in 170kA series electrolytic aluminium equipment. To verify the effectiveness of the

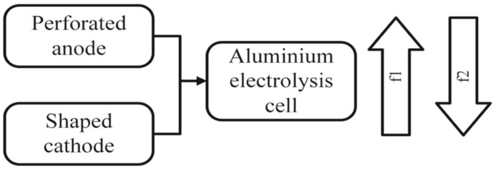


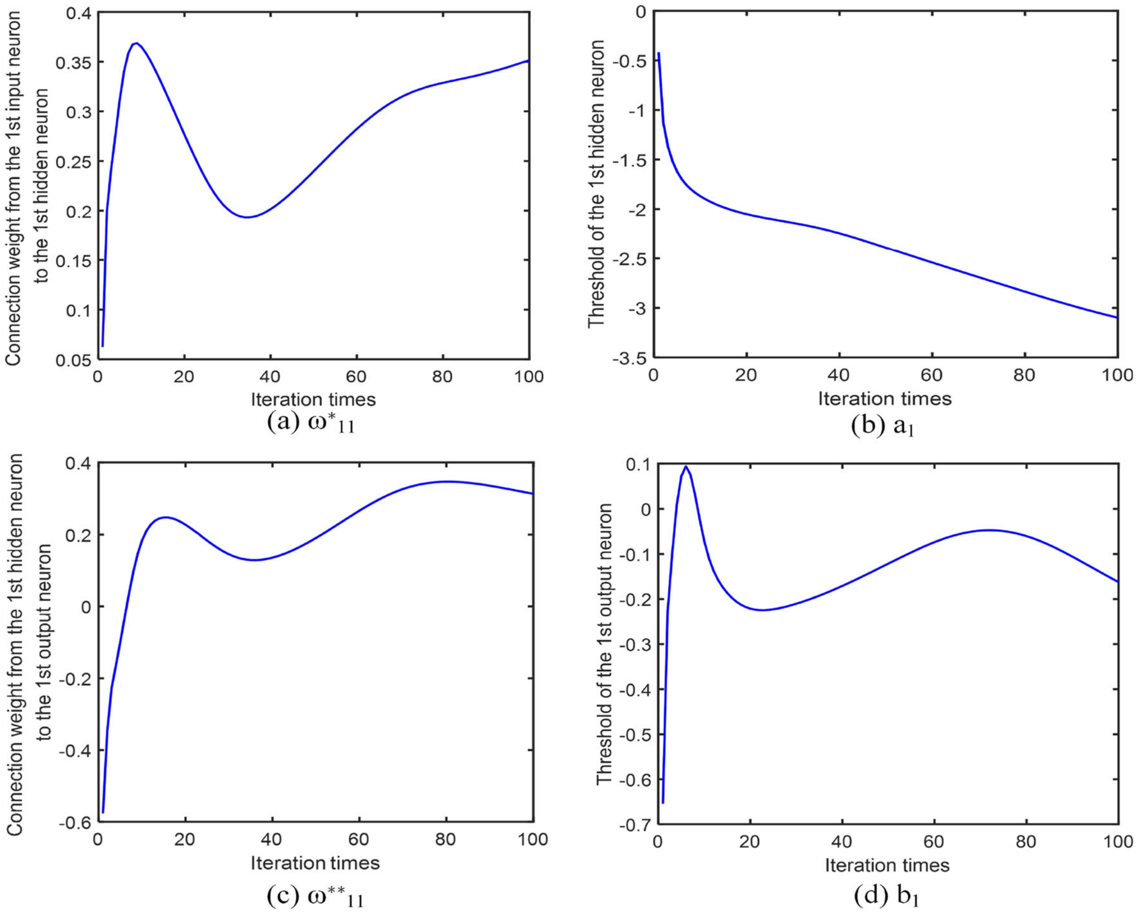
Fig. 5 The core components of electrolytic aluminium cell

Table 2 The sample data from No. 160 electrolytic aluminium equipment

Parameters	Sample				
	1	2	3	···	780
Series current (A)	1679	1679	1679	···	1761
Molecular ratio	2.31	2.31	2.31	···	2.75
Aluminium level (cm)	21.5	21.5	22	···	16
Electrolyte level (cm)	17	17	17	···	8
Cell temperature $^\circ C$	952	942	947	···	950
Aluminium output (kg)	1260	1240	1260	···	1345
Daily consumption of fluoride salt (kg)	29	23.4	21.6	···	19.7
NB times	646	707	671	···	798
Cell voltage (mV)	3712	3723	3718	···	3633
DC power consumption per ton of aluminium (kW.h/t-Al)	11852	12130	11889	···	12045
Current efficiency (%)	96.58	94.85	96.43	···	97.78

presented framework, we divided all 780 samples into 10 disjoint subsets on average. On this basis, 78 samples of one subset were selected as a testing set, and the other nine subsets were selected as a training set.

The MSFNN presented in the study was employed to build a 3-layer feedforward NN, which has 9 decision parameters and 2 outputs. The transfer functions of the second and third layers are Sigmoid and Purelin, respectively. The



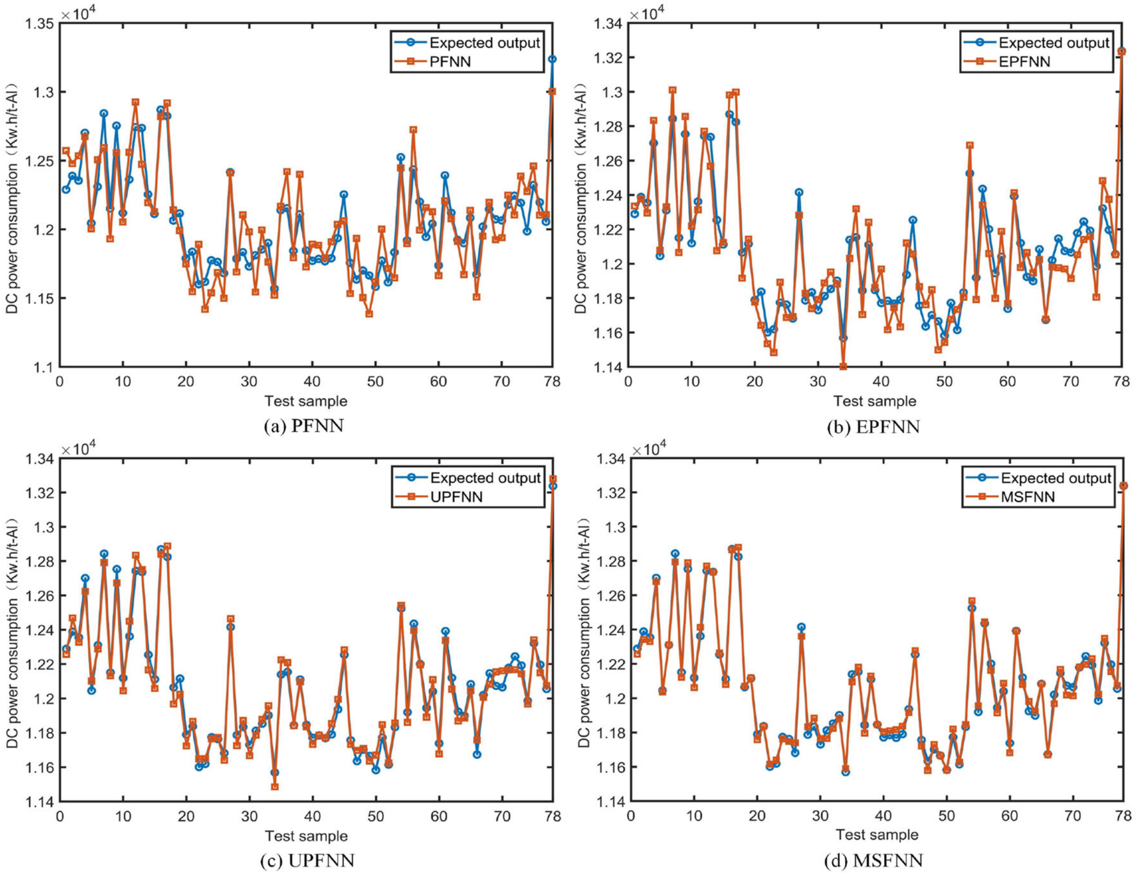
(a) ω^*_{11}

(b) a_1

(c) ω^{**}_{11}

(d) b_1

Fig. 6 The evolution of some weights and thresholds during the learning process



(a) PFNN

(b) EPFNN

(c) UPFNN

(d) MSFNN

Fig. 7 Prediction output of DC power consumption from the PFNN, EPFNN, UPFNN and MSFNN

number of NN's training iterations is set to 100. To maintain a certain precision and calculation speed, the second layer uses 9 neurons to form a 9-9-2 neural network structure in the experiment. In a consistent experimental object and environment, different models among PFNN, EPFNN, UPFNN and MSFNN are performed to predict technical indicators of the real-world EAP.

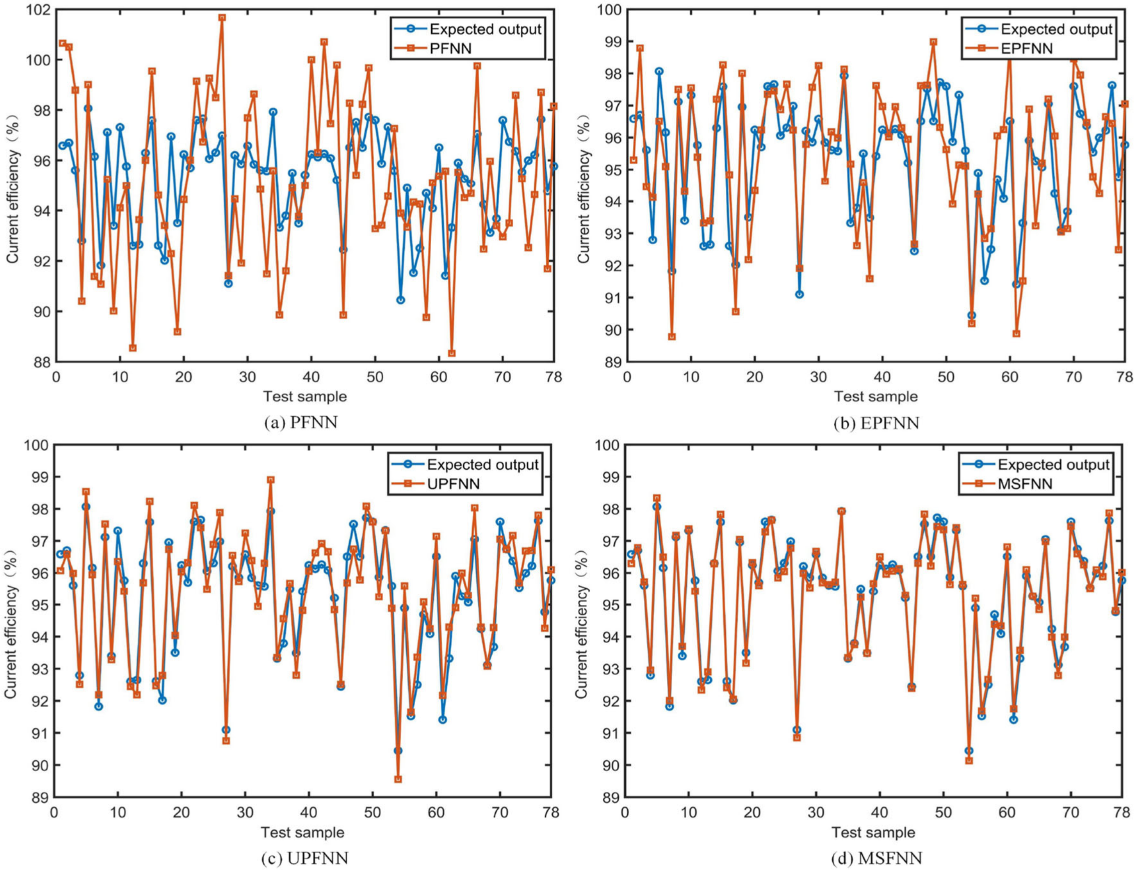
4.2 Analysis and discussion of experimental results

The following experimental results of the multiobjective prediction model based on the EAP all come from the PFNN, EPFNN, UPFNN and MSFNN algorithms. All process samples use the daily data shown in Table 2, and the experiment platform uses MATLAB R2014b (CPU: i7-9750H; RAM: 8.00GB; GPU: GTX 1660 Ti).

In Fig. 6, we select some weights and thresholds (such as ω_{11}^{*}, a_1, ω_{11}^{**} and b_1) represented in (3) to graphically show the evolution during the learning process. Figs. 7 and 8 show the fitting effects of DC power consumption and current efficiency by establishing the multiobjective prediction model for the EAP based on the above four algorithms. Figure 9 shows the comprehensive comparison effects of using the above four algorithms to predict the performance indicators (DC power consumption and current efficiency) of the EAP system.

Figure 10(a) intuitively shows the relative error percentage of the DC power consumption when using the four algorithms to predict the EAP model; Fig. 10(b) depicts the relative error percentage of the current efficiency when predicting the EAP system model based on the 4 algorithms. It can be seen that the relative error of the MSFNN algorithm is smaller than that of the other models, which demonstrates that the fitting effect of the MSFNN is better than that of the other three models. According to the experimental results, it has been verified that it is feasible to combine the MSF and the NN. Overall, the performance of the MSFNN model is more in line with the true characteristics of the EAP.

Table 3 compares different indicators from the multiobjective prediction models established by the above four algorithms, which shows six different evaluation criteria [42]. By analyzing Table 3, we can see that the predicted error in the PFNN model is larger than other models, and the predicted error value from the MSFNN process model is the smallest, indicating that the MSFNN's prediction accuracy is quite high. Meanwhile, it can also be confirmed from the



(a) PFNN
(b) EPFNN
(c) UPFNN
(d) MSFNN

Fig. 8 Prediction output of current efficiency from PFNN, EPFNN, UPFNN and MSFNN

side angle that the MSF technique plays a significant role in exploring the optimal model, which helps to further improve model performance and finally obtains the best parameter estimation values.

The significance nonparametric tests [47, 48] (i.e., Wilcoxon rank-sum test, Friedman test and Nemenyi test), which are an effective tool to verify the effectiveness of the developed framework, are adopted to analyze the significant difference of different algorithms. The test results with the MSFNN model as the comparison object have been shown in Table 4. It can be seen from the verification result that the developed framework has a significant difference compared with other algorithms. Furthermore, the time and space complexity of different algorithms are also analyzed and shown in Table 4. It indicates that although the MSFNN is obtained by constantly optimizing the PFNN, the corresponding complexity does not grow due to it. Therefore, the superiority of the proposed algorithm is reflected once again.

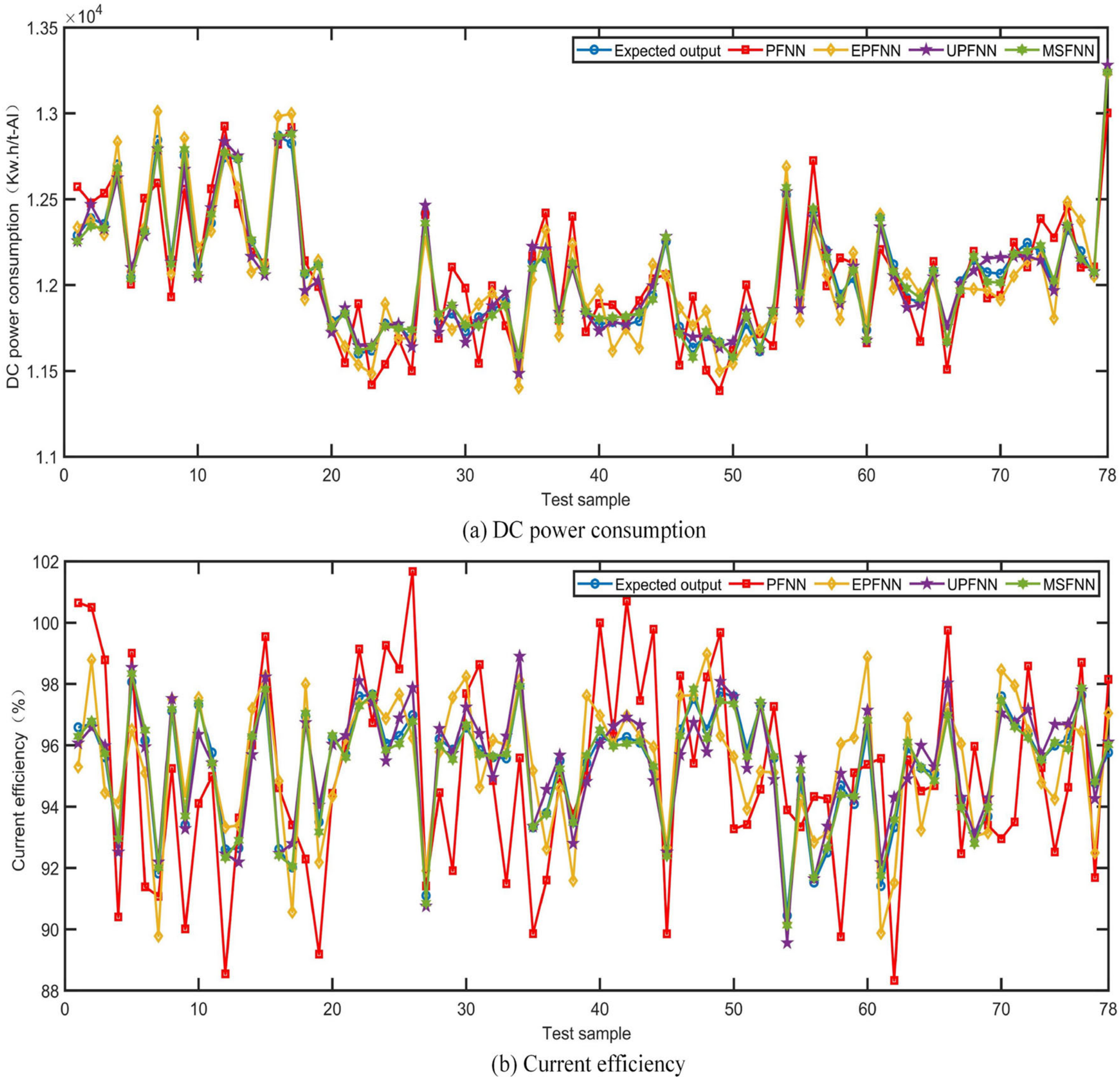
$$SSE = \sum_{i=1}^{T}(y_i - y)^2 \tag{23}$$

$$MSE = \frac{1}{T}\sum_{i=1}^{T}(y_i - y)^2 \tag{24}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(y_i - y)^2} \tag{25}$$

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

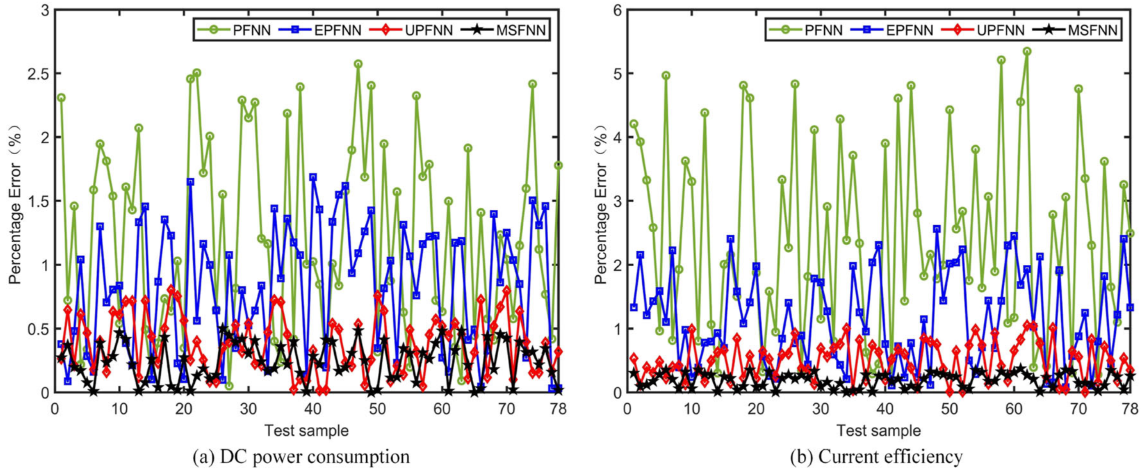
To better evaluate the prediction performance of different models and reduce the influence of overfitting on the proposed model, Tables 5 and 6 respectively give the statistical results of relevant performance indexes from DC power consumption and current efficiency based on k-fold cross-validation [49] with k = 10. The evaluation criteria include the mean absolute error (MAE), the mean relative error (MRE), and the correlation coefficient (R) [50]. Table 7 shows the statistical analysis results based on Tables 5 and 6, which better analyzes the 10 independent cross-validation tests of different algorithms. In Table 7, the comparison indicators include seven different levels. Based on the results, it can be seen that each indicator of



(a) DC power consumption

(b) Current efficiency

Fig. 9 Comparison of multi-objective prediction effects from the PFNN, EPFNN, UPFNN and MSFNN



(a) DC power consumption

(b) Current efficiency

Fig. 10 The relative error percentage of different multi-objective prediction models

Table 3 The comparison of related performance indicators from different models

	Model Category	Comparison of absolute error indicators in testing sets					
		Max	Min	Average	SSE	MSE	RMSE
MLR	Power consumption (Kw.h/t-Al)	1085.5	10.789	561.05	3.4098×10^7	4.3715×10^5	661.17
	Current efficiency (%)	17.529	0.3206	9.0028	5.6903×10^3	72.952	8.5412
NLMR	Power consumption (Kw.h/t-Al)	918.52	6.2576	448.01	1.8367×10^7	2.3548×10^5	485.26
	Current efficiency (%)	12.722	0.2799	6.9537	3.6040×10^3	46.205	6.7974
PFNN	Power consumption (Kw.h/t-Al)	604.15	1.5680	314.95	7.8429×10^6	1.0055×10^5	317.02
	Current efficiency (%)	8.8220	0.1588	4.4567	2.6931×10^3	34.527	5.8760
EPFNN	Power consumption (Kw.h/t-Al)	375.61	1.0267	186.05	3.7932×10^6	4.8631×10^4	220.52
	Current efficiency (%)	4.7600	0.1199	2.4582	761.01	9.7566	3.1236
UPFNN	Power consumption (Kw.h/t-Al)	180.44	2.5138	98.472	1.2670×10^6	1.6277×10^4	127.58
	Current efficiency (%)	2.1035	0.0429	1.0261	105.61	1.3540	1.1636
MSFNN	Power consumption (Kw.h/t-Al)	122.58	1.2663	68.520	3.5010×10^5	4.4885×10^3	66.996
	Current efficiency (%)	0.8410	7.8523×10^{-4}	0.4033	8.4786	0.1087	0.3297

The related performance indicators based on multiple linear regression (MLR, Fashoto et al, 2021) [43], multiple nonlinear regression (NLMR, Wen et al, 2021) [44], Particle filter neural network (PFNN, Qin et al, 2020) [14], Extended particle filter neural network (EPFNN, Jiang et al, 2021) [45] and Unscented particle filter neural network (UPFNN, Wang et al, 2020) [46] are added to Tables 3 and 4

the MSFNN algorithm is superior to other algorithms. The effectiveness of the proposed algorithm is proven again.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{26}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}|\frac{\hat{y}_i - y_i}{y_i}| \tag{27}$$

$$R = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - y_i)\sum_{i=1}^{n}(y_i - \bar{y})}} \tag{28}$$

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

Due to the complex production process of aluminum electrolysis, the cell condition information has dynamic and time-varying characteristics. To further verify the compensation ability of the proposed method for parameter variations and disturbance signals, different disturbances of 5%, 10% and 15% are artificially imposed on each model [4], as shown in Fig. 11(a)–(c). It is obvious from Fig. 11 that the DC energy consumption and current efficiency of the MSFNN can still resist the influence on the interference

Table 4 The comparison of related performance indicators from different models

Model Category	Significant nonparametric test ($p - value$)			Algorithm complexity	
	Wilcoxon rank-sum test	Friedman test	Nemenyi test	Time	Space
MLR	7.11×10^{-3}	6.16×10^{-4}	2.28×10^{-5}	$O(n^3)$	$O(1)$
NLMR	4.54×10^{-3}	4.89×10^{-4}	5.62×10^{-4}	$O(n^3)$	$O(1)$
PFNN	2.26×10^{-3}	1.57×10^{-4}	1.85×10^{-4}	$O(n^3)$	$O(n)$
EPFNN	9.33×10^{-2}	5.13×10^{-3}	8.46×10^{-3}	$O(n^3)$	$O(n)$
UPFNN	7.92×10^{-2}	3.49×10^{-3}	5.98×10^{-3}	$O(n^3)$	$O(n)$
MSFNN	\	\	\	$O(n^3)$	$O(n)$

Table 5 The statistical results of related performance indexes based on 10 cross-validation tests from PFNN, EPFNN, UPFNN and MSFNN models (DC aluminium power consumption)

	PFNN			EPFNN			UPFNN			MSFNN		
	MAE	MRE	R	MAE	MRE	R	MAE	MRE	R	MAE	MRE	R
1	3.8657	0.0257	0.9247	2.7536	0.0161	0.9539	1.4760	0.0088	0.9876	0.8074	0.0058	0.9949
2	3.3582	0.0198	0.9135	2.8995	0.0168	0.9673	1.5467	0.0089	0.9982	0.8049	0.0060	0.9967
3	4.0259	0.0260	0.8957	2.7465	0.0149	0.9438	1.3994	0.0094	0.9873	0.8956	0.0052	0.9982
4	4.6275	0.0278	0.9420	2.1346	0.0166	0.9666	1.4937	0.0089	0.9960	0.8732	0.0067	0.9926
5	4.5973	0.0271	0.9089	2.9465	0.0158	0.9439	1.5004	0.0090	0.9895	0.8349	0.0058	0.9893
6	4.1147	0.0248	0.8863	2.2337	0.0152	0.9712	1.4682	0.0085	0.9843	0.8766	0.0050	0.9924
7	4.0350	0.0230	0.8983	2.7469	0.0157	0.9420	1.4137	0.0095	0.9808	0.8438	0.0053	0.9907
8	3.8352	0.0285	0.9274	2.5354	0.0168	0.9167	1.4206	0.0087	0.9911	0.8467	0.0069	0.9958
9	4.3509	0.0264	0.9556	2.6568	0.0162	0.9650	1.3864	0.0089	0.9971	0.7928	0.0063	0.9990
10	3.0564	0.0277	0.8937	2.7699	0.0170	0.9417	1.6137	0.0103	0.9854	0.8650	0.0057	0.9977

Table 6 The statistical results of related performance indexes based on 10 cross-validation tests from PFNN, EPFNN, UPFNN and MSFNN models (Current efficiency)

	PFNN			EPFNN			UPFNN			MSFNN		
	MAE	MRE	R	MAE	MRE	R	MAE	MRE	R	MAE	MRE	R
1	0.0677	0.0548	0.6761	0.0378	0.0289	0.8858	0.0151	0.0125	0.9812	0.0062	0.0041	0.9970
2	0.0660	0.0527	0.6292	0.0382	0.0294	0.8703	0.0147	0.0108	0.9786	0.0064	0.0050	0.9975
3	0.0701	0.0511	0.6783	0.0391	0.0267	0.8628	0.0135	0.0134	0.9691	0.0069	0.0048	0.9954
4	0.0599	0.0460	0.7297	0.0355	0.0271	0.9073	0.0140	0.0148	0.9826	0.0060	0.0036	0.9960
5	0.0618	0.0578	0.7098	0.0337	0.0275	0.8906	0.0152	0.0092	0.9742	0.0054	0.0041	0.9987
6	0.0646	0.0636	0.6398	0.0336	0.0250	0.8934	0.0137	0.0105	0.9648	0.0056	0.0048	0.9978
7	0.0710	0.0555	0.6007	0.0354	0.0243	0.8349	0.0144	0.0109	0.9772	0.0058	0.0045	0.9990
8	0.0682	0.0592	0.5869	0.0364	0.0266	0.8439	0.0139	0.0111	0.9834	0.0061	0.0044	0.9985
9	0.0793	0.0453	0.6580	0.0411	0.0305	0.8754	0.0152	0.0107	0.9809	0.0064	0.0046	0.9988
10	0.0715	0.0624	0.5934	0.0382	0.0249	0.8931	0.0150	0.0099	0.9745	0.0059	0.0039	0.9971

Table 7 The statistical analysis of performance indexes from 10 independent cross-validation tests

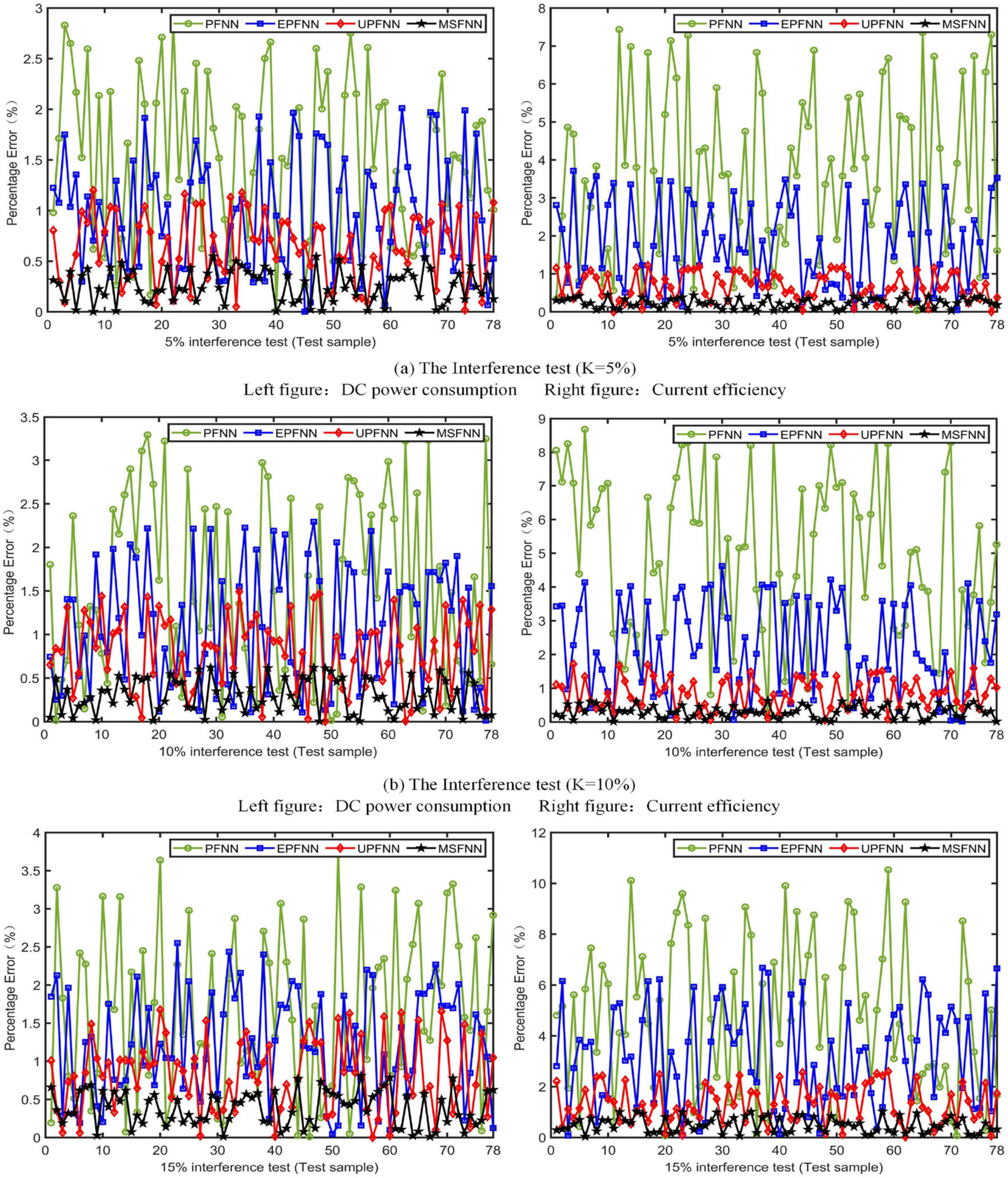
Evaluation Criteria	Model Category		Average	Min	Max	Std	Median	Lower quantile	Upper quantile
MAE	PFNN	Power consumption (Kw.h/t-Al)	3.9867	3.0564	4.6275	0.4727	4.0305	3.8428	4.2919
		Current efficiency (%)	0.0620	0.0599	0.0793	0.0052	0.0680	0.0650	0.0708
	EPFNN	Power consumption (Kw.h/t-Al)	2.6423	2.1346	2.9465	0.2541	2.7467	2.5658	2.7658
		Current efficiency (%)	0.0369	0.0336	0.0411	0.0023	0.0371	0.0354	0.0382
	UPFNN	Power consumption (Kw.h/t-Al)	1.4719	1.3864	1.6137	0.0675	1.4721	1.4154	1.4987
		Current efficiency (%)	0.0145	0.0135	0.0152	0.0006	0.0146	0.0139	0.0151
	MSFNN	Power consumption (Kw.h/t-Al)	0.8441	0.7928	0.8956	0.0326	0.8452	0.8143	0.8712
		Current efficiency (%)	0.0061	0.0054	0.0069	0.0004	0.0061	0.0058	0.0064
MRE	PFNN	Power consumption (Kw.h/t-Al)	0.0285	0.0198	0.0285	0.0025	0.0262	0.0250	0.0276
		Current efficiency (%)	0.0549	0.0453	0.0636	0.0059	0.0552	0.0515	0.0559
	EPFNN	Power consumption (Kw.h/t-Al)	0.0149	0.0149	0.0170	0.0007	0.0162	0.0157	0.0168
		Current efficiency (%)	0.0271	0.0243	0.0305	0.0019	0.0269	0.0254	0.0286
	UPFNN	Power consumption (Kw.h/t-Al)	0.0091	0.0085	0.0103	0.0005	0.0089	0.0088	0.0093
		Current efficiency (%)	0.0114	0.0092	0.0148	0.0016	0.0109	0.0106	0.0122
	MSFNN	Power consumption (Kw.h/t-Al)	0.0059	0.0050	0.0069	0.0006	0.0058	0.0054	0.0062
		Current efficiency (%)	0.0044	0.0036	0.0050	0.0004	0.0046	0.0041	0.0048
R	PFNN	Power consumption (Kw.h/t-Al)	0.9146	0.8863	0.9556	0.0214	0.9112	0.8964	0.9267
		Current efficiency (%)	0.6502	0.5869	0.7279	0.0465	0.6489	0.6078	0.6778
	EPFNN	Power consumption (Kw.h/t-Al)	0.9512	0.9163	0.9712	0.0160	0.9489	0.9425	0.9662
		Current efficiency (%)	0.8758	0.8349	0.9073	0.0219	0.8806	0.8647	0.8925
	UPFNN	Power consumption (Kw.h/t-Al)	0.9897	0.9808	0.9982	0.0055	0.9886	0.9859	0.9948
		Current efficiency (%)	0.9767	0.9648	0.9834	0.0058	0.9779	0.9742	0.9811
	MSFNN	Power consumption (Kw.h/t-Al)	0.9947	0.9893	0.9990	0.0032	0.9954	0.9925	0.9975
		Current efficiency (%)	0.9976	0.9954	0.9990	0.0012	0.9977	0.9970	0.9987

signal of electrolytic cell to a certain extent, indicating the advancement and effectiveness of this method.

By discussing the above experimental results, the fundamental reasons why the proposed method has better results than other methods are analyzed as follows:

(1) Although PFNN can handle nonlinear and non-Gaussian parameter estimation problems, the accuracy of the PFNN algorithm will gradually decrease with an increasing number of sampling particles. The main reason is the degradation of particles, that is, the weights of most particles decrease in the iterative process of particle updating. And MSFNN uses Sigma points obtained by UT method to guide the importance function, thus reducing the number of sampling particles and weakening the influence of particle degradation.

(2) The EPFNN uses the EKF to obtain sampling points for updating the importance function of the PF, while MSFNN proposes "multi-source filter" to update the



(a) The Interference test (K=5%)
Left figure：DC power consumption Right figure：Current efficiency

(b) The Interference test (K=10%)
Left figure：DC power consumption Right figure：Current efficiency

(c) The Interference test (K=15%)
Left figure：DC power consumption Right figure：Current efficiency

Fig. 11 The compensation ability of different methods in parameters variations and interference signals

importance function of the PF by using UT method to obtain Sigma points. Because EKF approximates linear estimation by discarding higher-order terms, its accuracy is poorer than that of UKF based on UT method.

(3) The UPFNN and MSFNN take into account the mean and variance of Sigma points obtained by UT method instead of the real mean and variance to achieve continuous sampling distribution. However, MSFNN constructs an important function $F(\boldsymbol{x})$ based on density kernel estimation, which gets rid of the previous filtering problem that the importance function is almost always represented by the Gaussian distribution $N(\bar{x}, \sigma)$ established by the mean $\bar{x}$

and variance σ of samples. It expands the application range of the filtering algorithm and reduces the algorithm sensitivity to mixed noise.

5 Conclusion

A multi-source filter neural network (MSFNN) algorithm is developed for exploring the system model's predictive potential. To apply the algorithm to an environment with mixed noise, the MSF technique is presented first. The MSF utilizes the mean and variance in the UKF to optimize the PF's importance function based on the density kernel estimation method. Then, the MSF employs the particles to evaluate the weights and thresholds of the NN. Finally, a multiobjective incremental learning prediction model based on the MSFNN for EAP systems is established. The performance comparison between the MSFNN and other electrolytic aluminium models established by the PFNN, EPFNN, and UPFNN algorithms shows that the multiobjective incremental learning model established by the MSFNN has high prediction accuracy and low sensitivity to noise interference, which greatly improves the adaptability of the EAP model.

However, this method is only applicable to the situation where the production data is available and the operation parameters are controllable. Moreover, although the MSFNN algorithm alleviates particle degradation, this problem still exists. In the future, the clustering kernel function smoothing method will be explored to overcome the problems of particle shortages in the MSFNN algorithm and the construction of deep filtering networks.

Appendix A: Pseudocode for MSFNN algorithm

Algorithm 1 Pseudocode for MSFNN algorithm.

```
Input: Initial network weight $W_1$, Network size (innum, midnum, outnum), Particles size $N$,
Discount factor $\gamma$, State $Q$ and Measurement $R$ noise, UT parameters ($\alpha$, β, $\kappa$)
Output: Particles' mean Xupf, Particles' variance Pupf
for t ← 2 to Particles size do
  $W1$ ← [ ]
  Network size ← [9,10,2]
  $N$ ← 200
  $\gamma$ ← $2 \times 10^{-4}$
  [$Q$, $R$] ← [$1 \times 10^{-5}$, $1 \times 10^{-5}$]
 [$\alpha$, β, $\kappa$] ← [1,0,2]
  processNoise(t) ← sqrt($Q$) * randn   // Generate process noise
  measureNoise(t) ← sqrt($R$) * randn   // Generate measurement noise
   $W_t$ ← $W_{t-1}$ + processNoise(t)   // Update system state
   $Y_t$ ← $W_t$ + measureNoise(t)   // Update system measurement
   $Z$ ← $Y_t$ - $\gamma$ + round()
   for i ← 1 to Particles size do   // Step 1: particle sampling
   vNoise ← [ ]
   wNoise ← [ ]
   noises ← vNoise + wNoise
   if (noises) then   // Using noises to extend state vector
   xQ ← Xukf
   PQ ← Pukf
   [xSigmaPts, wSigmaPts] ← SigmaPoints(xQ, PQ, alpha, beta, kappa)
   xPredSigmaPts ← xSigmaPts()   // Calculate Sigma point by UT method
   M ← size(xPredSigmaPts)
   for $l$ ← 1 to M do
   y($l$) ← reshape(xPredSigmaPts)
  zPredSigmaPts ← y($l$) + xSigmaPts()
  xPred ← [ ]
  zPred ← [ ]
  xPred ← xPred + xPredSigmaPts()   // Calculate the mean
```

```
zPred ← zPred + zPredSigmaPts()
exSigmaPt ← xPredSigmaPts() - xPred   // Calculate the variance
ezSigmaPt ← zPredSigmaPts() - zPred
PPred ← wSigmaPts() * exSigmaPt * exSigmaPt' // Unscented Transformation
PxzPred ← wSigmaPts() * exSigmaPt * ezSigmaPt'
S ← wSigmaPts() * ezSigmaPt * ezSigmaPt'
K ← PxzPred / S   // Calculation Kalman gain
inovation ← Z - zPred
Xukf ← xPred + K * inovation   // Further update state
Pukf ← PPred - K * S * K'   // Further update measurement
Xset ← Xukf() + mean(sqrt(Pukf()) * randn)   // Sampling particles
 for i ← 1 to Particles size do   // Step 2: Calculate the weight
Zpre ← $W_t$
 for i ← 1 to N do   // Calculate the weight and normalize
wi ← lik * prior / proposal
wi ← wi / sum(wi)
outIndex ← ResidualR(wi)   // Step 3: Resampling
 Xupfset ← Xset(outIndex)   // Step 4: Update sample sets
 Pupf ← mean(Xupfset()) * randn
 Xupf ← mean(Xupfset()) * randn   // Get the mean as final estimate
```

Appendix B: Robustness analysis of MSFNN algorithm

Taking (2) as the research object, inspired by the proof of UKF [51] and PF [52, 53] convergence, we further prove the robustness of the MSFNN algorithm. **Def. 1** The importance function constructed by density kernel estimation is $F(\cdot)$.

$$q(\omega_k^i|x_{k-1}^i, y_k) = F(\cdot) \tag{29}$$

Def. 2 The importance function $\hat{F}(\cdot)$ is obtained by optimizing $F(\cdot)$ with UKF's mean and variance.

$$q(\omega_k^i|x_{k-1}^i, y_k) = \hat{F}(\cdot) \tag{30}$$

Def. 3 The function φ in space $L_{k,p}$ can be expressed as

$$\|\varphi(\omega_k)\|_{k,p} = [\int_{L_{k,p}} |\varphi(\omega_k)|^p F_{k|k}(\omega_k)]^{\frac{1}{p}} \tag{31}$$

where ω represents the weights and thresholds of MSFNN algorithm and $\varphi(\cdot)$ is an arbitrary function.

Hypothesis 1 [54] Assuming that the measurement sequence y_k is known, and the parameter β_k in MSFNN satisfies the following formula.

$$(F_{k|k-1}(\omega_k), p(y_k|\omega_k)) > \beta_k > 0 \tag{32}$$

Hypothesis 2 [54] When $k > 0$

$$\|p(y_k|\omega_k)K(\omega_k|\omega_{k-1})/G(\omega_k|\omega_{k-1}, y_k)\| = \|\rho\| < \infty \tag{33}$$

$$\|p(y_k|\omega_k)\| = \|g\| < \infty \tag{34}$$

Lemma 1 When **Hypothesis 1-2** are satisfied, if $\varphi(\omega_k) \in L_{k,p}$, then

$$(K(\omega_k|\omega_{k-1}), g(y_k, \omega_k)|\varphi(\omega_k)|^p)^{1/p} \in L_{k-1,p} \tag{35}$$

The following Lemmas are drawn under **Hypothesis 1** and **Hypothesis 2**.

Lemma 2 When **Hypothesis 1-2** are satisfied, the following two formulas hold.

$$E[((F_{0|0}^N(\omega_0), \varphi(\omega_0)) - (F_{0|0}(\omega_0), \varphi(\omega_0)))^4] \le \frac{b_{0|0}^*(\varphi(\omega_0))}{N^2} \tag{36}$$

$$E[(F_{0|0}^N(\omega_0), \varphi(\omega_0))^4] \le m_{0|0}^*(\varphi(\omega_0)) \tag{37}$$

where b and m are arbitrary infinite decimals. Lemma 2 shows that the initialization phase of MSFNN before UKF optimization is convergent.

Lemma 3 When **Hypothesis 1-2** are satisfied, the following two formulas hold.

$$E[((F_{k-1|k-1}^N(\omega_{k-1}), \varphi(\omega_{k-1})) -(F_{k-1|k-1}(\omega_{k-1}), \varphi(\omega_{k-1})))^4] \le \frac{b_{k-1|k-1}^*(\varphi(\omega_{k-1}))}{N^2} \tag{38}$$

$$E[(F_{k-1|k-1}^N(\omega_{k-1}), \varphi(\omega_{k-1}))^4] \le m_{k-1|k-1}^*(\varphi(\omega_{k-1})) \tag{39}$$

After UKF optimization, the following two formulas hold.

$$E[((\hat{F}^N_{k|k-1}(\omega_{0:k}), \rho(\omega_k, y_k)\varphi(\omega_k)) \\ -(\hat{F}_{k|k-1}(\omega_k), g(y_k, \omega_k)\varphi(\omega_k)))^4] \\ \leq \frac{b_{k|k-1}(\varphi(\omega_k))}{N^2} \tag{40}$$

$$E[(\hat{F}^N_{k|k-1}(\omega_k), \rho(\omega_k, y_k)\varphi(\omega_k))^4] \leq m_{k|k-1}(\varphi(\omega_k)) \tag{41}$$

Lemma 4 When **Hypothesis 1-2** are satisfied, the following two formulas hold.

$$E[((\hat{F}^N_{k|k-1}(\omega_k), \rho(\omega_k, y_k)\varphi(\omega_k)) \\ -(\hat{F}_{k|k-1}(\omega_k), g(y_k, \omega_k)\varphi(\omega_k)))^4] \\ \leq \frac{b^{**}_{k|k-1}(\varphi(\omega_k))}{N^2} \tag{42}$$

$$E[(\hat{F}^N_{k|k-1}(\omega_k), \rho(\omega_k, y_k)\varphi(\omega_k))^4] \leq m^{**}_{k|k-1}(\varphi(\omega_k)) \tag{43}$$

Lemmas 3-4 indicate that MSFNN is convergent in the prediction phase from moment k-1 to k.

Lemma 5 When **Hypothesis 1-2** are satisfied, the probability satisfies the following formula.

$$P[\frac{1}{N}\sum_{i=1}^{N}\rho(\omega^{(i)}_{k|k-1}, y_k) < \gamma_k] \leq \frac{d_k}{N^2} \tag{44}$$

And when $N \geq [d_k^{0.5}] + 1$,

$$P[\frac{1}{N}\sum_{i=1}^{N}\rho(\omega^{(i)}_{k|k-1}, y_k) < \gamma_k] \leq \frac{d_k}{|[d_k^{0.5}] + 1|^2} < 1 \tag{45}$$

Lemma 6 When **Hypothesis 1-2** are satisfied, the following two formulas hold.

$$E[((\hat{F}^N_{k|k-1}(\omega_k), \rho(\omega_k, y_k)\varphi(\omega_k)) \\ -(\hat{F}_{k|k-1}(\omega_k), g(y_k, \omega_k)\varphi(\omega_k)))^4] \\ \leq \frac{b^{*}_{k|k-1}(\varphi(\omega_k))}{N^2} \tag{46}$$

$$E[(\hat{F}^N_{k|k-1}(\omega_k), \rho(\omega_k, y_k)\varphi(\omega_k))^4] \leq m^{*}_{k|k-1}(\varphi(\omega_k)) \tag{47}$$

Lemma 7 When **Hypothesis 1-2** are satisfied, the following two formulas hold.

$$E[((\hat{F}^N_{k|k}(\omega_k), \varphi(\omega_k)) - (\hat{F}_{k|k}(\omega_k), \varphi(\omega_k)))^4] \\ \leq \frac{b^{*}_{k|k}(\varphi(\omega_k))}{N^2} \tag{48}$$

$$E[(\hat{F}^N_{k|k}(\omega_k), \varphi(\omega_k))^4] \leq m^{*}_{k|k}(\varphi(\omega_k)) \tag{49}$$

Lemmas 6-7 indicate that MSFNN is still convergent after residual resampling.

Lemma 8 When **Hypothesis 1-2** are satisfied, the following two formulas hold.

$$E[((\hat{F}^N_{k|k}(\omega_k), \varphi(\omega_k)) - (\hat{F}_{k|k}(\omega_k), \varphi(\omega_k)))^4] \\ \leq \frac{b_{k|k}(\varphi(\omega_k))}{N^2} \tag{50}$$

$$E[(\hat{F}^N_{k|k}(\omega_k), \varphi(\omega_k))^4] \leq m_{k|k}(\varphi(\omega_k)) \tag{51}$$

According to Lemmas 2-8, the following conclusions can be drawn.

Conclusion 1 When **Hypothesis 1-2** are satisfied, the following formula holds for any $\varphi(\omega_k) \in L_{k,4}$:

$$e^N_{MSFNN}[\varphi(\omega_k)] \rightarrow e_{optimal}[\varphi(\omega_k)] \quad a.s. \tag{52}$$

To sum up, MSFNN obtained in this paper is convergent.

Acknowledgements This study was supported by the National Natural Science Foundation of China (No. 51805059 and 61802317), and Scientific and Technological Research Program of Chongqing Municipal Education Commission (KJQN2021033).

Declarations

Conflict of Interests 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 Automatica Sinica 44(11):1957–1970
2. 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(7):305–319
3. Peng J, Song Y, Di Y, Wang Y, Feng N (2019) Towards improved current efficiency of hall-heroult cells by using a novel cathode and process parameters. JOM 71(2):1–6
4. 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 & Applic 31(8):4271–4285
5. Lei Y, Chen X, Min M, Xie Y (2019) A semi-supervised laplacian extreme learning machine and feature fusion with cnn for industrial superheat identification. Neurocomputing, 381(3)

6. Miao R, Gao Y, Ge L, Jiang Z, Zhang J (2019) Online defect recognition of narrow overlap weld based on two-stage recognition model combining continuous wavelet transform and convolutional neural network. Comput Ind 112:103115
7. Yi J, Li TF, Hou J, Yao LZ, Tian YF (2013) Dynamic prediction model based on fnn-ukf neural networks for alumina concentration. Journal of Sichuan University 45(1):169–174
8. Li T, Yao L, Hou J (2015) Generic hybrid dynamic modeling and robust optimizing of industrial processes using improved ukfnn and nsga-2 for performance optimization. Application Research of Computers 32(9):2716–2719
9. 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
10. Van Quan D, Dinh M-C, Kim CS, Park M, Doh C-H, Bae JH, Lee M-K, Liu J, Bai Z (2021) Design of an effective state of charge estimation method for a lithium-ion battery pack using extended kalman filter and artificial neural network. Energies 14(9):1–20
11. Zhou S, Shen CY, Zhang L (2019) Dual-optimized adaptive kalman filtering algorithm based on bp neural network and variance compensation for laser absorption spectroscopy. Optics express 27(22):31874–31888
12. Barbieri M, Nguyen K, Diversi R, Medjaher K, Tilli A (2021) Rul prediction for automatic machines: a mixed edge-cloud solution based on model-of-signals and particle filtering techniques. J Intell Manuf 32:1421–1440
13. Cheng Q, Sun P, Yang C, Yu R, Liu PX (2021) Forecasting and simulation of cutting force in virtual surgery based on particle filtering. Appl Intell 51:1934–1946
14. Qin W, Lv H, Liu C, Nirmalya D, Jahanshahi P (2020) Remaining useful life prediction for lithium-ion batteries using particle filter and artificial neural network. Industrial Management & Data Systems 120(2):312–328
15. Lin C, Wang H, Fu M, Yuan J, Gu J (2021) A gated recurrent unit-based particle filter for unmanned underwater vehicle state estimation. IEEE Trans Instrum Meas 70:1–12
16. Zhou R, Tang M, Gong Z, Hao M (2020) Freetrack: Device-free human tracking with deep neural networks and particle filtering. IEEE Syst J 14(2):2990–3000
17. Morariu C, Morariu O, Rileanu S, Borangiu T (2020) Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. Comput Ind 120:103244
18. Aktepe A, Yank E, Ersz S (2021) Demand forecasting application with regression and artificial intelligence methods in a construction machinery company. J Intell Manuf 32(6):1587–1604
19. Zhao Q, Liu Y, Yao W, Yao Y (2022) Hourly rainfall forecast model using supervised learning algorithm. IEEE Trans Geosci Remote Sens 60:1–9
20. Zhang S, Zhang Y, Zhu J (2018) Residual life prediction based on dynamic weighted Markov model and particle filtering. J Intell Manuf 29(4):753–761
21. Jamei M, Olumegbon IA, Karbasi M, Ahmadianfar I, Asadi A, Mosharaf-Dehkordi M (2021) On the thermal conductivity assessment of oil-based hybrid nanofluids using extended kalman filter integrated with feed-forward neural network. Int J Heat Mass Transfer 172:121159
22. Pesce V LM (2020) Radial basis function neural network aided adaptive extended kalman filter for spacecraft relative navigation. Neural Comput & Applic 96:105527
23. Goleijani S AMT (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
24. 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
25. Zhang H, Zhou X, Wang Z, Yan H (2020) Maneuvering target tracking with event-based mixture kalman filter in mobile sensor networks. IEEE Transactions on Cybernetics 50(10):4346–4357
26. Zhang Q, Xu W, Zhang W, Feng J, Chen Z (2020) Multi-hypothesis square-root cubature kalman particle filter for speaker tracking in noisy and reverberant environments. IEEE/ACM Transactions on Audio, Speech, and Language Processing 28:1183–1197
27. Bae SJ (2016) Remaining useful life prediction of lithium-ion batteries based on spherical cubature particle filter. IEEE Transactions on Instrumentation & Measurement 65(6):1282–1291
28. Kumar A, Dimitrakopoulos R, Maulen M (2020) Adaptive self-learning mechanisms for updating short-term production decisions in an industrial mining complex. J Intell Manuf 31:1795–1811
29. Dong H, Sun J, Li T, Ding R, Sun X (2020) A multi-objective algorithm for multi-label filter feature selection problem. Appl Intell 50:3748–3774
30. Kubalik J, Derner E, Babuska R (2021) Multi-objective symbolic regression for physics-aware dynamic modeling. Expert Syst Appl 182:115210
31. Taifu LI, Hou J, Yao L, Jun YI, Xiaohua GU, You Y (2014) Kalman artificial neural network with measurable noise estimation by gamma test for dynamic industrial process modeling. Journal of Mechanical Engineering 50(18):29
32. 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
33. Cheng CA, A. RX, Rya B, Ws B, Fs A (2019) State-of-charge estimation of lithium-ion battery using an improved neural network model and extended kalman filter. J Clean Prod 234:1153–1164
34. Roy PT, Jofre L, Jouhaud J, Cuenot B (2020) Versatile sequential sampling algorithm using kernel density estimation. Eur J Oper Res 284(1):201–211
35. Zhu G, Dai Q (2021) Ensp & incl: a hybrid time series prediction algorithm integrating dynamic ensemble pruning, incremental learning, and kernel density estimation. Appl Intell 51:617–645
36. Pan C, Huang A, He Z, Lin C (2021) Prediction of remaining useful life for lithium-ion battery based on particle filter with residual resampling. Energy Science & Engineering 9(8):1115–1133
37. Zhang H, Qin S, Ma J, You H (2013) Using residual resampling and sensitivity analysis to improve particle filter data assimilation accuracy. IEEE Geoscience & Remote Sensing Letters 10(6):1404–1408
38. Breda M (1998) Appendix 1: Convergence aspects on back propagation neural networks. Subst Use Misuse 33(2):503–514
39. Yi J, Bai (2018) Recurrent neural network and preference information based aluminum electrolytic modeling and optimizing method. CN 109086469-A
40. 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 Transactions on Industrial Informatics 14(8):3405–3415
41. 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
42. 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

43. Ali M, Prasad R, Xiang Y, Deo RC (2020) Near real-time significant wave height forecasting with hybridized multiple linear regression algorithms. Renew Sust Energ Rev 132:110003
44. Rastgou M, Bayat H, Mansoorizadeh M, Gregory AS (2020) Estimating the soil water retention curve: Comparison of multiple nonlinear regression approach and random forest data mining technique. Comput Electron Agric 174:105502
45. Zhengxin J, Qin S, Yujiang W, Hanlin W, Bingzhao G, Lin H (2021) An Immune Genetic Extended Kalman Particle Filter approach on state of charge estimation for lithium-ion battery. Energy, 230(C)
46. Wang Y, Chen Z (2020) A framework for state-of-charge and remaining discharge time prediction using unscented particle filter. Appl Energy 260:114324
47. Barros RSMD, Hidalgo JIG, Cabral DRDL (2018) Wilcoxon rank sum test drift detector. Neurocomputing 275:1954–1963
48. Saji Y, Barkatou M (2021) A discrete bat algorithm based on levy flights for euclidean traveling salesman problem. Expert Syst Appl 172:114639
49. Wong T-T, Yeh P-Y (2020) Reliable accuracy estimates from k-fold cross validation. IEEE Trans Knowl Data Eng 32(8):1586–1594
50. Li W, Kong D, Wu J (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):694–710
51. Daid A, Busvelle E, Aidene M (2021) On the convergence of the unscented kalman filter. Eur J Control 57:125–134
52. Hu X, Schon TB, Ljung L (2011) A general convergence result for particle filtering. IEEE Transactions on Signal Processing 59(7):3424–3429
53. Hu X, Schon TB, Ljung L (2008) A basic convergence result for particle filtering. IEEE Transactions on Signal Processing 56(4):1337–1348
54. Ding W, Yao L, Li Y, Long W, He T (2020) Dynamic evolutionary model based on a multi-sampling inherited hapfnn for an aluminium electrolysis manufacturing system. Appl Soft Comput 99(11):106925

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Lizhong Yao[1,2] · **Wei Ding**[3] · **Tiantian He**[4] · **Shouxin Liu**[5] · **Ling Nie**[1]

Lizhong Yao
cqust-ylz@cqust.edu.cn

Tiantian He
tiantian.he@ntu.edu.sg

Shouxin Liu
2019223029160@stu.scu.edn.cn

Ling Nie
2005050@cqust.edu.cn

[1] School of Electrical Engineering, Chongqing University of Science and Technology, Chongqing 401331, Peoples Republic of China

[2] College of Physics and Electronic Engineering, Chongqing Normal University, Chongqing 401331, Peoples Republic of China

[3] School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, Peoples Republic of China

[4] School of Computer Science and Engineering, Nanyang Technological University, Nanyang, 639798, Singapore

[5] School of Mechanical Engineering, Sichuan University, Chengdu 610065, Peoples Republic of China