membranes

MDPI

Article

Membrane Fouling Diagnosis of Membrane Components Based on MOJS-ADBN

Yaoke Shi [1], Zhiwen Wang [1,2,3,*], Xianjun Du [1,2,3], Bin Gong [1], Yanrong Lu [1,2,3], Long Li [1,4] and Guobi Ling [1]

[1] College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China
[2] Key Laboratory of Gansu Advanced Control for Industrial Processes, Lanzhou University of Technology, Lanzhou 730050, China
[3] National Demonstration Center for Experimental Electrical and Control Engineering Education, Lanzhou University of Technology, Lanzhou 730050, China
[4] GS-Unis Intelligent Transportation System & Control Technology Co., Ltd., Lanzhou 730050, China
* Correspondence: wwwangzhiwen@163.com

Abstract: Given the strong nonlinearity and large time-varying characteristics of membrane component fouling in the membrane water treatment process, a membrane component-membrane fouling diagnosis method based on the multi-objective jellyfish search adaptive deep belief network (MOJS-ADBN) is proposed. Firstly, the adaptive learning rate is introduced into the unsupervised pre-training phase of DBN to improve the convergence speed of the network. Secondly, the MOJS method is used to replace the gradient-based layer-by-layer weight fine-tuning method in traditional DBN to improve the ability of network feature extraction. At the same time, the convergence of the MOJS-ADBN learning process is proven by constructing the Lyapunov function. Finally, MOJS-ADBN is used in the membrane packaging diagnosis to verify the performance of the model diagnosis. The experimental results show that MOJS-ADBN has a fast convergence speed and a high diagnostic accuracy, and can provide a theoretical basis for membrane fouling diagnosis in the actual operation of membrane water treatment.

Keywords: adaptive learning rate; MOJS; DBN; stability proof; membrane fouling diagnosis

check for updates

Citation: Shi, Y.; Wang, Z.; Du, X.; Gong, B.; Lu, Y.; Li, L.; Ling, G. Membrane Fouling Diagnosis of Membrane Components Based on MOJS-ADBN. *Membranes* **2022**, *12*, 843. https://doi.org/10.3390/membranes12090843

Academic Editors: Negin Koutahzadeh, Milad Rabbani Esfahani and Shamas Tabraiz

Received: 30 July 2022
Accepted: 26 August 2022
Published: 29 August 2022

1. Introduction

Membrane bioreactor (MBR) technology, as an important means in sewage treatment engineering, is a new wastewater treatment process that combines membrane technology and biological treatment technology and is mainly composed of membrane components and the bioreactor [1–3]. It has been recognized as one of the most promising new technologies in the field of water treatment in the 21st century due to its excellent comprehensive performance. However, membrane fouling of membrane components will increase the operating cost of the MBR, becoming a bottleneck problem that restricts its wide application [4,5]. Therefore, researchers are gradually focusing on membrane component–membrane fouling diagnosis technology in the field of water treatment. The traditional fault diagnosis method is divided into three steps. Firstly, the signal is preprocessed by denoising and decomposing. Secondly, the preprocessed signal can obtain its time domain, frequency domain, or other features through certain feature extraction methods. The feature extraction methods include wavelet transform [6], synchronous extraction [7], empirical wavelet transform [8], and so on. These methods filter the useless features of the signal, making the desired fault features more obvious. Finally, the extracted features are input into the classifier based on machine learning for training; the classification of faults can be recognized by the training classifier. The backpropagation neural network (BP-NN) [9] and support vector machine (SVM) [10] have been applied to fault classification. The above methods have the characteristics of simple feature extraction and easy adjustment of classifier parameters, and the final diagnostic recognition rate can meet most requirements. However, the above methods still separate fault feature extraction and diagnosis and recognition, which require a lot of expert experience in signal feature processing and rely on the ability of manual feature extraction, which is limited [11–13]. Similarly, the traditional fault diagnosis method based on signal processing adopts the manual extraction of features and input of the classification model for fault identification [14–16]. The process relies heavily on manual experience and prior knowledge, which are insufficient in large data scales and fast acquisition speeds.

In view of the dynamic and nonlinear characteristics of the membrane water treatment system, the traditional diagnostic model is inefficient, and the potential valuable features are ignored in the offline modeling stage, resulting in false alarms and inaccurate interpolation [17]. As a breakthrough in the field of modern artificial intelligence, deep learning can automatically learn valuable features from original feature sets and even original data, which means that deep learning can largely get rid of the dependence on advanced signal processing technology, artificial feature extraction, and cumbersome feature selection technology. Therefore, deep learning is widely used in the field of fault diagnosis with its powerful learning ability and feature extraction ability [18–20]. Ba-Alawi et al. proposed an inclusive framework for missing data interpolation and sensor self-verification based on the variational automatic encoder and deep residual network structure integration [21]. By learning the potential probability distribution of input data, complex features are automatically extracted to reduce the risk of gradient disappearance. By inputting missing data, detecting anomalies, identifying fault sources, and reconstructing fault data to a normal state, the reliability of fault sensors is improved. In recent years, a series of deep learning fault diagnosis models based on the convolutional neural network (CNN) have been greatly improved in diagnosis efficiency and accuracy. Shi et al. used attention mechanisms and improved convolutional neural networks to diagnose membrane pollution, which improved diagnostic accuracy and efficiency [22,23]. However, the deep learning model requires a large number of data to optimize parameters and is prone to over-fitting [24,25]. More researchers have studied the application of deep belief networks (DBNs) in the field of fault diagnosis. DBNs have strong feature extraction abilities, which can automatically extract features from a large number of data, reduce the dependence on expert fault diagnosis experience and signal processing technology, and reduce the uncertainty of feature extraction and fault diagnosis caused by manual participation in traditional methods [26–28]. A DBN characterizes the complex mapping relationship between signals and the health status by establishing a deep model, which is suitable for the diagnosis and analysis of diverse, nonlinear, high-dimensional health monitoring data in the context of big data [29]. Therefore, applying a DBN to the field of fault diagnosis has certain timeliness, practicality, and versatility. Zhao et al. proposed a fault diagnosis method based on a DBN, which adaptively extracted features from the original time series signals, increasing flexibility [30]. Simulation results show the effectiveness of this method in fault diagnosis. The structural parameters of a typical DBN model are determined by the learning rate [31]. Therefore, Liu et al. applied an optimized DBN to improve the accuracy of fault diagnosis [32]. Zhang et al. proposed a fault diagnosis model of complex chemical processes based on an extensible DBN [33]. With the help of mutual information technology, a DB subnetwork is used to extract individual fault features in the space–time domain. A global two-layer backpropagation network has been trained and used for fault classification, and the effect of fault diagnosis of this method was verified. Dai proposed a DBN fault diagnosis model with an improved model structure, which adopted multi-layer and multi-dimensional mapping to extract more detailed fault type differences and accurately diagnose faults [34]. Zhu et al. introduced a DBN network into a multi-sensor information fusion model to identify uncertain, unknown, and changing fault modes [35]. Compared with the traditional artificial neural network information fusion diagnosis method, this method has higher recognition accuracy. Su et al. used the model after GWO optimized the parameters of the support vector machine to diagnose the signal features extracted by DBN, realizing the online detection of equipment faults, and improving the diagnostic accuracy [36]. Zhu proposed an intelligent fault diagnosis method based on PCA and

DBN [37]. The PCA method is used to reduce the dimension of the original signal, to extract fault eigenvalues and eigenvectors. The modified samples are then trained and tested by DBN for fault classification and diagnosis. This method does not need complex signal processing of the original data, so it is easy to implement and has wide applicability. Due to the uncertainty of the dynamic system model of membrane water treatment, the nonlinearity of data signals, and the uncertainty of the membrane fouling state, the extraction of membrane fouling characteristics from membrane components is in trouble. In addition, with the increase in the scale and complexity of industrial control systems, membrane fouling data signals are often composed of a large number of high-dimensional data, which makes the processing of original membrane fouling data more complex.

Based on the above problems, this article proposes a membrane-packing diagnosis method based on MOJS-ADBN to optimize the DBN from the perspective of unsupervised learning and supervised learning: we used an adaptive learning rate to accelerate network convergence, and prove that the unsupervised part optimized by the adaptive learning rate is stable. The supervised part uses the MOJS algorithm optimization to fine-tune the weight and proves that MOJS optimization has global convergence and stability in the Lyapunov meaning. We used the MOJS-ADBN model as an example of the membrane fouling diagnosis of the parallel ultrafiltration membrane component and verified the comprehensive performance of the MOJS-ADBN model through a number of comparative tests.

2. Traditional DBN Model

2.1. Subsection

In 2006, Hinton proposed a DBN, which is a probability generation model composed of multiple restricted Boltzmann machine (RBM) stacks.

As a two-layer network, a RBM is bidirectionally connected by the visible layer and the hidden layer, and the neurons of the same layer network are independent of each other. The visual layer is used to input training data, while the hidden layer is used to extract features. The structure diagram of the RBM is shown in Figure 1. In the formula, w_1^R represents the connection weight, b is the bias coefficient of the hidden layer, and a is the bias coefficient of the visible layer.

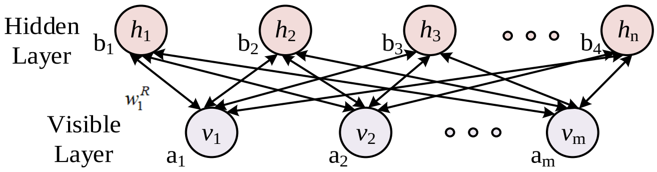


Figure 1. Structure of the RBM.

The feature extraction process of the DBN is classified into two stages: the pre-training stage and fine-tuning stage. In the pre-training stage, all RBMs are first pre-trained layer-by-layer, unsupervised, to form a feature model of unsupervised learning. Next, the supervised algorithm is used for reverse training, and all the initial connection weights of RBM are fine-tuned, to reduce the error caused by training, which is conducive to the DBN to extract the essential characteristics of the input data. The structure is shown in Figure 2.

2.2. Unsupervised Learning

To determine the initial weight of the network, Hinton used an unsupervised training method to learn the parameters. One RBM includes a visible layer and a hidden layer, which are represented by v and h, respectively. Given the model parameter $\theta = \{w^R, a, b\}$, the joint probability distributions $P(v, h; \theta)$ of the visible layer and the hidden layer are defined by the energy function $E(v, h; \theta)$ as:

$$P(v,h;\theta) = \frac{1}{Z} e^{-E(v,h;\theta)} \tag{1}$$

$$P(v;\theta) = \frac{1}{Z}\sum\nolimits_h e^{-E(v,h;\theta)} \tag{2}$$

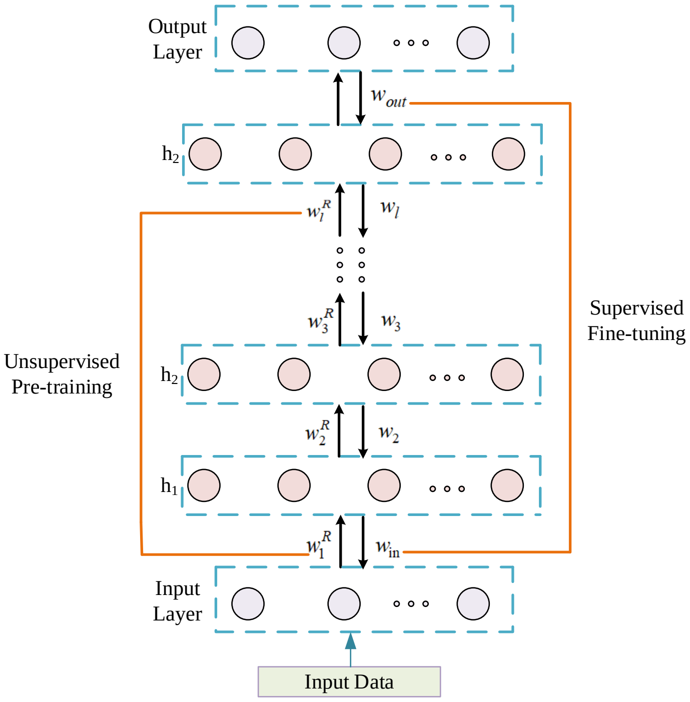


Figure 2. Structure of DBN.

For an RBM with Bernoulli (visible layer) distribution–Bernoulli (hidden layer) distribution, the energy function of the unit joint configuration is defined as:

$$E(v,h) = -\sum_{i=1}^{m}\sum_{j=1}^{n} v_i w_{ij}^R h_j - \sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j \tag{3}$$

In the formula, w_{ij}^R is the connection weight of RBM, a_i and b_j are the offsets of the visible layer cells and hidden layer cells, respectively.

The conditional distributions of v and h are:

$$P(h_j = 1/v;\theta) = \sigma\left(b_j + \sum_{i=1}^{m} v_i w_{ij}^R\right) \tag{4}$$

$$P(v_i = 1/h;\theta) = \sigma\left(a_i + \sum_{j=1}^{n} w_{ij}^R h_j\right) \tag{5}$$

In the formula, σ is the activation function.

The probability value standard of the visible layer and the hidden layer is usually achieved by setting a threshold, because the visible layer and the hidden layer are Bernoulli binary states. Taking the hidden layer as an example, it can be expressed as:

$$h_j\begin{cases} 0\,if & p(h_j = 1/v) < \delta \\ 1\,if & p(h_j = 1/v) > \delta \end{cases} \tag{6}$$

In the formula, δ is a constant between 0.5 and 1.

We calculate the gradient of the log-likelihood function $\lg P(v;\theta)$, and the RBM weight update formula can be obtained as:

$$w_{ij}^R = w_{ij}^R + \eta \Delta w_{ij}^R \tag{7}$$

$$\Delta w_{ij}^{R} = E_{\text{data}}\left(v_i h_j\right) - E_{\text{model}}\left(v_i h_j\right) \tag{8}$$

In the formula, η represents the learning rate, $E_{\text{data}}\left(v_i h_j\right)$ are the data expectations observed in the training set, $E_{\text{model}}\left(v_i h_j\right)$ is the expectation on the distribution determined by the model, and $E_{\text{model}}\left(v_i h_j\right)$ can be obtained by the Gibbs approximation.

2.3. Supervised Learning

Supervised learning involves fine-tuning the weight w^R obtained by unsupervised learning. Taking the output layer and the last hidden layer of Figure 2 as examples, let F be the expected output of the model and define the cross-entropy function as the loss function:

$$F = -\frac{1}{n}\sum_{i}\left[y_i \ln y_i' + (1 - y_i)\ln\left(1 - y_i'\right)\right] \tag{9}$$

In the formula, y_i represents the output of the target output after SoftMax, y_i' represents the output of the expected output after SoftMax, and n represents the number of categories.

The weight update formula can be expressed as:

$$w_{out}(\tau + 1) = w_{out}(\tau) - \eta\frac{\partial F(\tau)}{\partial w_{out}(\tau)} \tag{10}$$

Using this method, the weight $w = (w_{out}, w_l, w_{l-1}, \cdots, w_2, w_{in})$ of the whole DBN network can be obtained by fine-tuning from the top output layer to the bottom input layer.

3. MOJS-ADBN Learning Algorithm

3.1. Adaptive Learning Rate CD Algorithm

In the unsupervised learning process of the DBN, Gibbs sampling, as the core of the contrastive divergence (CD) algorithm, is a Markov chain Monte Carlo (MCMC) algorithm. When it is difficult to directly sample the joint distribution, it is used to generate a set of approximate observations of a specific multi-parameter probability distribution. Gibbs sampling mainly consists of three steps.

(1) The Gibbs chain is initialized with sample V to obtain the visual layer input $v^{(0)}$.

(2) According to Formulas (4)–(6), sampling is carried out, respectively. In the formula, $h^{(\mathrm{t})}$ is obtained by sampling $P\left(h^{(t)}/v^{(t)};\theta\right)$, $v^{(\mathrm{t}+1)}$ is obtained by sampling $P\left(v^{(t+1)}/h^{(t)};\theta\right)$, t is the number of sampling steps.

(3) We repeat the second stage.

Because each RBM requires multiple iterations, the fixed learning rate η is prone to convergence difficulties. Therefore, the adaptive learning rate is used to determine the learning rate η according to the updates direction of the parameters. The principle of the adaptive learning rate is that the learning rate will increase if the parameter update direction is the same after two consecutive iterations, and the learning rate will decrease if the parameter update direction is opposite after two consecutive iterations. The update mechanism of the adaptive learning rate η is as follows:

$$\eta = \begin{cases} B\eta & \left(\Delta w_{ij}^{R}\right)^{(t)} + \left(\Delta w_{ij}^{R}\right)^{(t+1)} = \left|\left(\Delta w_{ij}^{R}\right)^{(t)}\right| + \left|\left(\Delta w_{ij}^{R}\right)^{(t+1)}\right| \\ b\eta & \left(\Delta w_{ij}^{R}\right)^{(t)} + \left(\Delta w_{ij}^{R}\right)^{(t+1)} < \left|\left(\Delta w_{ij}^{R}\right)^{(t)}\right| + \left|\left(\Delta w_{ij}^{R}\right)^{(t+1)}\right| \end{cases} \tag{11}$$

$$\left(\Delta w_{ij}^{R}\right)^{(t)} = v_i^{(t)} h_i^{(t)} - v_i^{(t+1)} h_i^{(t+1)} \tag{12}$$

$$\left(\Delta w_{ij}^{R}\right)^{(t+1)} = v_i^{(t+1)} h_i^{(t+1)} - v_i^{(t+2)} h_i^{(t+2)} \tag{13}$$

In the formula, B = 1.4, b = 0.7.

3.2. Supervised Fine Adjustment Based on MOJS

The three main features of MOJS are as follows: (1) Archiving is integrated into the jellyfish search to save and retrieve Pareto optimal solutions. (2) The crowding distance and roulette selection are used to effectively manage the archive population, including the optimal non-dominated solution in the spatial search process. (3) To alleviate local optimization, Lévy flight, an elite group, is added to MOJS based on opposite jumping. The weights obtained from the unsupervised process are fine-adjusted by MOJS.

3.2.1. Time Control Function

Jellyfish are attracted by nutrients in the ocean current; they gather in the ocean current (and thus form jellyfish groups). There are also movements in jellyfish groups, namely passive movement (A-type movement) and active movement (B-type movement). The transformation of jellyfish (from A-type movement to B-type movement) is affected by the time control function c(t), and its expression is as follows:

$$\mathrm{c}(t) = \left|\left(1 - \frac{t}{Max_{iter}}\right) \times (2 \times \mathrm{rand}(0,1) - 1)\right| \quad (14)$$

In the formula, c_0 = 0.5.

3.2.2. Elite Choice

We added a file to the MOJS algorithm to store and retrieve the best approximation of the real Pareto optimal solution in the optimization process. The selection of elite targets was set in the area with the least jellyfish in the Pareto optimal frontier. The recognition method of this region involved dividing the search space by finding the best elite target and the worst target of the obtained Pareto optimal solution, defining a hypersphere and n grid elements covering all solutions, and dividing the hypersphere into equal sub-hyperspheres in each iteration; the roulette mechanism was used to select. The roulette mechanism can improve the distribution of the whole Pareto optimal frontier. When there are more Pareto optimal numbers, the probability of being selected is smaller, as shown in the following formula:

$$\mathrm{P}_i = \frac{C}{\mathrm{N_i}} \quad (15)$$

In the formula, C = 10, $\mathrm{N_i}$ is the number of Pareto optimal solutions obtained in segment i.

3.2.3. Lévy Flight

The behaviors of most flying animals can be described by Lévy flight when the spatial dimension of a random walk is higher than one dimension and the step size distribution of Lévy flight is isotropic; we used the Mantegna algorithm to generate a stable step size:

$$\mathrm{Lvy(s)} \sim \mathrm{s} = \frac{\mathrm{u}}{|\mathrm{v}|^{\frac{1}{\tau}}},\ 0 < \tau \leq 2 \quad (16)$$

In the formula, u and v obey normal distributions: $\mathrm{u} \sim \mathrm{N}\left(0, \sigma_{\mathrm{u}}^2\right)$, $\mathrm{v} \sim \mathrm{N}\left(0, \sigma_{\mathrm{v}}^2\right)$.

$$\sigma_{\mathrm{u}} = \left\{\frac{\Gamma(1+\tau)\sin\left(\frac{\pi\tau}{2}\right)}{\Gamma\left[\frac{1+\tau}{2}\right]\tau 2^{(\tau-1)/2}}\right\}^{\frac{1}{\tau}},\ \sigma_{\mathrm{v}} = 1,\ \tau = 1.5 \quad (17)$$

In the formula, $\Gamma(\mathrm{z})$ is Gamma distribution: $\Gamma(\mathrm{z}) = \int\limits_0^\infty \mathrm{t}^{\mathrm{z}-1}\mathrm{e}^{-\mathrm{t}}\mathrm{dt}$.

3.2.4. Update and Archive

In the iteration process, the archived file will be updated every time, and it may reach the upper limit of the total number in the optimization process. We used the management mechanism to filter the archived files, and the specific contents are as follows:

(1) If there is a solution that can play a leading role in the Pareto optimal solution in the original archive, we store the solution and delete the dominant solution in the original archive.

(2) If there is a solution A, and there is no dominant relationship between the original archived solution, the solution in the original archive will be retained. If the number of archives does not reach the upper limit, solution A will be added to the archive file.

(3) If the number of archives reaches the upper limit, the solution will be deleted from the stage with the most filling, and solution A will be added to the archive file.

(4) If there is a solution A that can be dominated by the original archiving solution, then we eliminate solution A.

To effectively select solutions to be deleted from the archive, the worst (the most jellyfish) hypersphere should be selected to prevent jellyfish from searching in crowded areas without food. The selection method is realized through the roulette wheel mechanism, and the probability of each segment is:

$$P_{i'} = \frac{N_i}{C} \tag{18}$$

In the formula, C = 10, N_i is the number of Pareto optimal solutions obtained in segment i.

3.2.5. MOJS

We used Lévy flight to speed up the local search along the ocean current; the formula of the ocean current motion is:

$$X_i(t+1) = EL\_X_i(t) + \overrightarrow{trend} \otimes Lvy(s) \tag{19}$$

$$\overrightarrow{trend} = X^*(t) - 3 \times rand(0,1) \times \frac{\sum EL\_X}{n_{pop}} \tag{20}$$

In the formula, $EL\_X_i(t)$ is the elite member in $X_i(t)$, $\sum EL\_X$ is the elite group, n_{pop} is the group size, $X^*(t)$ is the elite solution with time t selected in the archive.

Similarly, we used elite solutions to replace the current best solutions of active movements and passive movements in jellyfish groups.

Passive movement:

$$X_i(t+1) = X^*(t) + (EL\_X_i(t) - X^*(t)) \otimes Lvy(s) \tag{21}$$

Active movement:

$$X_i(t+1) = X^*(t) + \overrightarrow{Step} \tag{22}$$

In the formula:

$$\overrightarrow{Step} = rand(0,1) \times \overrightarrow{Direction} \tag{23}$$

$$\overrightarrow{Direction} = \begin{cases} EL\_X_j(t) - EL\_X_i(t) \text{ if } EL\_X_i(t) \succ EL\_X_j(t) \\ EL\_X_i(t) - EL\_X_j(t) \text{ if } EL\_X_j(t) \succ EL\_X_i(t) \end{cases} \tag{24}$$

3.2.6. Population Initialization

Logistic mapping, compared with the random initialization, is not easy to produce premature convergence and ensures population diversity. The formula is as follows:

$$X_{i+1} = \eta X_i(1 - X_i),\ 0 \le X_0 \le 1 \tag{25}$$

In the formula, X_i is the logistic value of the *i*-th jellyfish position, X_0 is used to generate the initial population of jellyfish, η is equal to 4, $X_0 \in (0,1), X_0 \notin \{0.0,\ 0.25,\ 0.75,\ 1.0\}$.

3.2.7. Increase Diversity through Opposition-Based Jumping

This mechanism is effective when the population is approximately transformed into the optimal solution. If the jump condition $\mathrm{rand}(0,1) < \frac{\mathrm{t}}{\mathrm{Max_{iter}}}$ is satisfied, the corresponding population based on opposition $\mathrm{X\prime_i(t)}$ is calculated and $\mathrm{n_{pop}}$ is calculated. After generating a new population through evolution, the most suitable individual is selected from the current population and the opposite population. In the formula, based on the opposite $\mathrm{X\prime_i(t)}$ population, the calculation formula is:

$$\mathrm{X\prime_i(t)} = (\mathrm{Lb_i}+\mathrm{Ub_i})-\mathrm{X_i(t)} \tag{26}$$

We extracted the hidden layer states obtained by unsupervised learning, and then carried out MOJS fine adjustment in sequence, according to the above steps, $w = (w_{out}, w_l, w_{l-1}, \cdots, w_2, w_2, w_{in})$.

So far, the supervised fine adjustment based on MOJS is complete. Firstly, the adaptive learning rate is used to accelerate the unsupervised training process and obtain the initial weight. Secondly, the MOJS algorithm is used to fine-tune the initial weight obtained from the unsupervised process to complete the MOJS-ADBN algorithm process.

4. Algorithm and Convergence Analysis

4.1. Adaptive Learning Rate CD Algorithm Analysis

(1) Convergence rate refers to the time taken by RBM to use Gibbs sampling many times in order to achieve the expected reconstruction error. The shorter the training time is, the faster the convergence speed is. As a probability model, the unsupervised learning of RBM is mainly used to learn features, which is called the coding adaptive learning rate, which automatically adjusts the learning factor by changing the step size. By comparing the sampling states of the visible layer and the hidden layer every two times, the efficiency of Gibbs sampling improves, and the convergence of the CD algorithm accelerates. Professor Hinton pointed out that hierarchical dimensionality reduction can achieve the effect of exponential reduction in the dimension of high-dimensional data. Similarly, since MOJS-ADBN is a hierarchical representation of multiple RBMs when a single RBM can accelerate convergence through the adaptive learning rate, the convergence speed of DBN will increase exponentially.

(2) The learning process of RBM weights is different from that of traditional BP networks. RBM is unsupervised learning, while BP is supervised learning; therefore, similar conclusions of the BP algorithm cannot measure RBM. In the unsupervised training stage, the adaptive learning rate algorithm adaptively increases or decreases the learning rate according to the parameter update direction. In addition, in the supervised fine-tuning stage, the algorithm can avoid being in cyclic fluctuations and falling into local optimization in the optimization process.

At the same time, the adaptive learning rate involves regularly increasing or decreasing the learning intensity of the algorithm on the internal correlation of data in the way of the variable step size, and converging in the shortest time.

4.2. Unsupervised Training Phase

In the unsupervised training phase of DBN, to quickly converge, RBM is trained in turn by using the adaptive learning rate. To avoid particularity, in Formulas (4) and (5), the upper and lower asymptotes of the sigmoid function are represented by A_H and A_L, and the input information of the RBM visual layer and the reconstruction state obtained after t samplings are represented by f_i^0 and f_j^t, respectively. Then, the visual layer and hidden layer are expressed as follows in a Gibbs sampling process:

$$f_i^0 \in [A_L, A_H] \tag{27}$$

$$f_j^0 = A_L + (A_H - A_L)\sigma\left(b_j + \sum_{i=1}^{m} f_i^0 W_{ij}\right) \tag{28}$$

$$f_i^1 = A_L + (A_H - A_L)\sigma\left(a_i + \sum_{j=1}^{n} W_{ij} f_j^0\right) \tag{29}$$

$$f_j^1 = A_L + (A_H - A_L)\sigma\left(b_j + \sum_{i=1}^{m} f_i^1 W_{ij}\right) \tag{30}$$

It can be concluded that, after t Gibbs sampling

$$f_i^t = A_L + (A_H - A_L)\sigma\left(a_i + \sum_{j=1}^{n} W_{ij}^R f_j^{t-1}\right) \tag{31}$$

$$f_j^t = A_L + (A_H - A_L)\sigma\left(b_j + \sum_{i=1}^{m} f_i^t W_{ij}^R\right) \tag{32}$$

From the formula, the network output is related to the intermediate state of the sampling process. At the same time, the convergence speed and accuracy of the algorithm are related to the adaptive learning rate. Too large or too small adaptive learning rates will affect the convergence speed and even make the network unstable. From the above, we can obtain the following performance analysis:

(1) Proof of sufficiency.

If $f_j^0, f_i^1 \in [A_L, A_H]$, according to (27) to (30), then $f_j^1 \in [A_L, A_H]$.

(2) Proof of necessity.

On the one hand, if the whole network is stable and the input state of the first RBM satisfies $f_i^0 \in [A_L, A_H]$, then the output state range of the top RBM satisfies $f_j^1 \in [A_L, A_H]$, and then it must satisfy $f_j^0, f_i^1 \in [A_L, A_H]$.

Proof:

If the network is stable, the visual and hidden layers of each RBM layer meet the input–output boundedness. Because the sigmoid function is monotonically increasing, and the number of open neurons is also increasing, we can obtain:

$$f_j^1 > f_i^1 \tag{33}$$

$$f_i^1 > f_j^0 \tag{34}$$

Then

$$f_j^0, f_i^1, f_j^1 \in [A_L, A_H] \tag{35}$$

So

$$f_j^t > f_i^t \tag{36}$$

$$f_i^t > f_j^0 \tag{37}$$

$$f_j^0, f_i^t, f_j^t \in [A_L, A_H] \tag{38}$$

Furthermore, we know:

Assume that f_j^0, f_i^t, f_j^t represent the input state, intermediate state, and output state of RBM, respectively, the sufficient and necessary condition for network stability is:$f_j^0, f_i^t, f_j^t \in [A_L, A_H]$.

According to Formula (6), the greater the δ, the smaller the probability that the neuron takes 1, resulting in the increased sparsity of neurons in the visible layer and hidden layer in the Gibbs sampling process, and the possibility that the weight update direction is the same in the Gibbs sampling iteration process for two consecutive times will increase.

$$P\left\{ \left(\Delta w_{ij}^{R}\right)^{(t)} + \left(\Delta w_{ij}^{R}\right)^{(t+1)} = \left|\left(\Delta w_{ij}^{R}\right)^{(t)}\right| + \left|\left(\Delta w_{ij}^{R}\right)^{(t+1)}\right| \right\} \propto \delta \tag{39}$$

According to (8)–(12), if the error fluctuation is not obvious in the process of adjusting the weight, the increase in the learning rate can accelerate the convergence of the weighted network.

Then there is:

$$B \propto P\left\{ \left(\Delta w_{ij}^{R}\right)^{(t)} + \left(\Delta w_{ij}^{R}\right)^{(t+1)} = \left|\left(\Delta w_{ij}^{R}\right)^{(t)}\right| + \left|\left(\Delta w_{ij}^{R}\right)^{(t+1)}\right| \right\} \tag{40}$$

According to Gibbs sampling, every time the weight is updated once, the intermediate state is accompanied by two binarization samples, and the updated weight is proportional to the state sampling, so the relationship between δ and the learning rate coefficients B and b can be obtained:

$$\begin{cases} B \approx 2\delta \\ b \approx \delta \end{cases} \tag{41}$$

The purpose of δ is to judge the state of binary neurons, which is generally 0.7.

4.3. Supervised Training Phase

4.3.1. Multi-Objective Jellyfish Behavior Process

For the optimization problem, the calculation formula is as follows:

$$\begin{array}{c} \max f(X) \\ s.t.\ g_i(X) \le 0 \quad i = 0, 1, 2, \cdots M \quad X \in Z \end{array} \tag{42}$$

In the formula, $f(X)$ is the objective function, $g_i(X)$ is the i-th constraint, M is the total number of constraints, X is the n-dimensional unknown variable, and Z is the search space. The position state of jellyfish is equivalent to the Pareto optimal solution, and its set represents the Pareto solution set, which is expressed as follows:

$$X = [X_1, X_2, \cdots X_n] \tag{43}$$

Assuming that the search space Z is a continuous state space, the interval $\left[X_i^l, X_i^h\right]$ where X is located can be decomposed into h-l discrete values. Then the accuracy can be expressed as $\varepsilon = \frac{X_i^h - X_i^l}{h-l}$, in the formula, ε is the accuracy of the optimal solution. Z is a discrete space, and its state size is:

$$|Z| = \frac{\prod_{i=1}^{n}\left(X_i^h - X_i^l\right)}{\varepsilon} \tag{44}$$

The position state $X \in Z$ of each jellyfish, and its food energy, is defined as:

$$F = \{f(X) | X \in Z\} \tag{45}$$

Then $|F| < |Z|$ is obtained, so:

$$F = \{F_1, F_2, \cdots, F_{|F|}\}, F_1 > F_2 > \cdots > F_{|F|} \tag{46}$$

According to the difference of energy, the search space set Z can be classified into several non-empty subsets $\{Z^i\}$, in the formula:

$$Z^i = \{X | X \in Z, f(X) = F_i\}\ i = 1, 2, \cdots, |F| \tag{47}$$

So, $\sum\limits_{i=1}^{|F|}\left|Z^i\right| = |Z|, \forall i \in \{1, 2, \cdots, |F|\}$, $Z^i = \phi$, and $\forall i \neq j, Z^i \cap Z^j = \phi$, which satisfy $\bigcup\limits_{i=1}^{|F|} Z^i = Z$.

The energy of jellyfish (that is food) is defined as:

$$E(X) = f(X) \tag{48}$$

Let X_s be a set of all jellyfish, X is n-vector variable, X satisfies $\forall X \in X_S$, and $\forall X \in X_S$, so $F_{|F|} \leq E(X) \leq F_1$, set X_s can be reduced to a non-empty subset, and the expression is shown as follows:

$$X_S^i = \{X|X \in X_S, E(X) = f(X) = F_i\}\ i = 1, 2, 3, \cdots, |F| \tag{49}$$

So, $\sum\limits_{i=1}^{|F|}\left|X_S^i\right| = |X_S|, \forall i \in \{1, 2, 3, \cdots, |F|\}, X_S^i \neq \phi$, and $\forall i \neq j, X_S^i \cap X_S^j = \phi$ satisfies $\bigcup\limits_{i=1}^{|F|} X_S^i = X_S$.

Let $X^{i,j}$ satisfy $i = 1, 2, \cdots, |F|, j = 1, 2, \cdots, \left|X_S^i\right|$. $X^{i,j}$ represents the position of the j-th jellyfish in X^i. Multi-mechanism jellyfish include ocean current movement, jellyfish A-type movement, and jellyfish B-type movement. Assume that the transition of j-th jellyfish from one motion state to another is represented by $X^{i,j} \to X^{m,n}$, and the probability of occurrence is $P_{ij,mn}$, assume that the transition of the j-th jellyfish from the i-th region to the m region in X^i represents $X^{i,j} \to X^m$, and the probability of occurrence is $P_{ij,m}$, and satisfies $P_{ij,m} = \sum\limits_{\mathrm{n}=1}^{|X_S^k|} P_{ij,mn}$, $\sum\limits_{k=1}^{|F|} P_{ij,m} = 1$. Assume that the jellyfish in X^i changes from the i-th region to the m-th region, indicating $X^i \to X^m$, and the probability of occurrence is $P_{i,m}$ and satisfies $P_{i,m} \geq P_{ij,m}$.

4.3.2. Stability of Reducible Random Matrix

Theorem 1: Let P be a reducible random matrix of order N, after the same row transformation and column transformation, $P = \begin{bmatrix} C \cdots 0 \\ R \cdots T \end{bmatrix}$, in the formula, C is a primitive random matrix of order M, R and T are matrices of order N-M, and neither R nor T is a matrix of **0**. Therefore,

$$P^\infty = \lim_{k\to\infty} P^k = \lim_{k\to\infty}\begin{bmatrix} C^k & \cdots & 0 \\ \sum\limits_{i=1}^{k-1} T^i R C^{k-i} & \cdots & T^k \end{bmatrix} = \begin{bmatrix} C^\infty \cdots 0 \\ R^\infty \cdots T \end{bmatrix} \tag{50}$$

In the formula, P^∞ is a stable random matrix, and $P^\infty = 1'P^\infty$, $P^\infty = P^0 P^\infty$ are uniquely determined and independent of the initial distribution, P^∞ satisfies the condition:

$$P^\infty = \left[P_{ij}\right]_{N\times N} = \begin{cases} P_{ij} > 0 & 1 \leq i \leq N, \quad 1 \leq j \leq M \\ P_{ij} = 0 & 1 \leq i \leq N, \quad M < j \leq N \end{cases} \tag{51}$$

4.3.3. Proof of Global Convergence

Lemma: in the multi-mechanism jellyfish algorithm, $\forall X^{i,j} \in X_S^i$, $i = 1, 2, \cdots |F|$, $j = 1, 2, \cdots, \left|X_S^i\right|$ satisfy:

$$\forall m > i, P_{i,m} = 0 \tag{52}$$

$$\exists m < i, P_{i,m} > 0 \tag{53}$$

here is the proof of Formula (52).

Let $X^{i,j}$ be the artificial jellyfish after t iterations, and record it as $X(t)$, the jellyfish with the highest energy in $X(t)$ is X_{Best}. In the formula, and X_{Best} is the n-dimensional vector, then there is $E(X_{\text{Best}}) = F_i$. According to the definition of the update archive in multi-mechanism jellyfish, the highest energy jellyfish update in the iteration process can be known as:

$$E(X(t+1)) \geq E(X(t)) \tag{54}$$

Then

$$\forall m > i, P_{ij,mn} = 0 \tag{55}$$

$$\forall m > i, P_{ij,m} = \sum_{n=1}^{|X_S^k|} P_{ij,mn} = 0 \tag{56}$$

So

$$\forall m > i, P_{i,m} = 0 \tag{57}$$

here is the proof of Formula (53).

According to the change of time state and environmental state, there will be ocean current movement, jellyfish A-type movement, and jellyfish B-type movement. If $X(t + 1)$ is the best jellyfish and $B(t+1) = X(t+1)$, the following three phenomena will occur.

Phenomenon 1. Let the jellyfish carry out ocean current movement, and let the probability of producing the ocean current movement be $P_{Ocean} \geq 0$, then the jellyfish group will be attracted by the nutrients in the ocean current to update its position. Then the food concentration at the position before moving is lower than that at the position after moving; that is, $E(X(t+1)) > E(X(t))$, which proves that $\exists m < i, P_{i,m} > 0$.

Phenomenon 2. If the jellyfish carries out jellyfish A-type movement, set the probability of generating the jellyfish A-type movement as $P_A \geq 0$, and it will move around its own position. Then two situations will occur.

Situation 1. The food concentration at the position after moving is higher than that at the position before moving. Let the probability of this phenomenon be P_{A1}, which proves to be the same as Phenomenon 1.

Situation 2. The food concentration at the current location of the jellyfish is higher than that at the surrounding location. If the probability of this situation is $P_{A2} = 1 - P_{A1}$, the surrounding location needs to be re-selected. Assume t attempts, the probability is P_{A2}^t. If the food concentration in the position after moving is higher than that in the position before moving, it is the same as that in situation 1. Therefore, if it is still not satisfied after t iterations, according to the time control function $c(t)$, the jellyfish movement gradually changes from the A-type movement to B-type movement with the increase of times, as shown in Phenomenon 3.

Phenomenon 3. If jellyfish carries out jellyfish B-type movement, it is caused by two conditions.

Situation 1. Jellyfish are produced at the beginning. Let the probability of producing jellyfish B-type movement be $P_{B1} \geq 0$ and $P_{B2} = 1 - P_{Ocean} - P_A$, if the food concentration at the location of a jellyfish in the neighborhood is higher than the food concentration at the current set location, so $E(X(t+1)) > E(X(t))$, which means $\exists m < i, P_{i,m}$.

Situation 2. The jellyfish gradually evolves from A-type movement to B-type movement with the time control function $c(t)$; assume that the probability of occurrence is $P_{B2} \geq 0$, if the food concentration at the location of a jellyfish in the neighborhood is higher than the food concentration at the current set location, then $E(X(t+1)) > E(X(t))$, which means $\exists m < i, P_{i,m}$.

With the increase of t, the use of the jump based on opposites can effectively prevent local optimization.

According to the multi-mechanism jellyfish algorithm, the three movements of jellyfish meet $P_{B2} + P_{Ocean} + P_A = 1$, and $\exists m < i, P_{i,m} > 0$ are proved in each case.

Theorem: 2—the multi-mechanism jellyfish algorithm has convergence.

Proof: X_S^i, $i = 1, 2, \cdots, |F|$ is only related to current changes and has nothing to do with history, and the sample space is limited, so it can be regarded as a finite Markov Chain. According to Lemma (1) in Section 4.3.3, the transfer matrix of Markov Chain is:

$$P = \begin{bmatrix} P_{1,1} & 0 & \cdots & 0 \\ P_{2,1} & P_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ P_{|F|,1} & P_{|F|,2} & \cdots & P_{|F|,|F|} \end{bmatrix} = \begin{bmatrix} C & 0 \\ R & T \end{bmatrix} \tag{58}$$

According to Lemma (2) in Section 4.3.3:

$$P_{2,1} > 0,\ R = \left(P_{2,1}, P_{3,1}, \cdots, P_{|F|,1}\right)^T \tag{59}$$

$$T = \begin{bmatrix} P_{2,2} & \cdots & 0 \\ \vdots & \ldots & \vdots \\ P_{|F|,2} & \cdots & P_{|F|,|F|} \end{bmatrix} \neq 0,\ C = P_{1,1} = 1 \tag{60}$$

If P is a reducible random matrix of order N, then
$P^\infty = \lim\limits_{k\to\infty} P^k = \lim\limits_{k\to\infty} \begin{bmatrix} C^k & \cdots & 0 \\ \sum\limits_{i=1}^{k-1} T^i R C^{k-i} & \cdots & T^k \end{bmatrix} = \begin{bmatrix} C^\infty & \cdots & 0 \\ R^\infty & \cdots & T \end{bmatrix}$ and $C^\infty\ = 1$, $R^\infty = [1, 1, , \cdots, 1]^T$.

Therefore, $P^\infty = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}$ is a stable random matrix, which leads to:

$$\lim_{\mathrm{t}\to\infty} P\{E(X(t)) = F_B\} = 1 \tag{61}$$

In the formula, F_B is the optimal objective function, so the multi-mechanism jellyfish algorithm has global convergence.

4.3.4. Global Stability Proof

From Section 4.3.3, it can be seen that the multi-mechanism jellyfish algorithm finally converges to the global best, so the initial position of X_0 will eventually converge to the global best x_{max}. x_{max} is assumed to be the equilibrium point under the Lyapunov meaning.

Proof: Assume the objective function of the multi-mechanism jellyfish algorithm is $f(X)$, then the dynamic formula is:

$$\dot{X} = f(X, t) \tag{62}$$

Let the x axis translate $f(x_{max})$ upward, then the dynamic formula is updated as:

$$\dot{X} = f(X, t) - f(x_{max}) \tag{63}$$

According to the convergence of algorithm, when $t \to \infty$, the position state X of jellyfish tends to the global best x_{max}:

$$\lim_{\mathrm{t}\to\infty} \|X(t:\ X_0, t_0) - X_{\mathrm{e}}\| = 0 \tag{64}$$

So, for all t, the equilibrium state is satisfied.

$$\dot{X}_{\mathrm{e}} = f(X_{\mathrm{e}}, t) - f(x_{max}) = 0 \tag{65}$$

In the formula, x_{max} is the equilibrium point in the MOJS algorithm, and $\dot{X}_{\mathrm{e}} = f(x_{max}, t)$ is the equilibrium state. Therefore, there are equilibrium points and equilibrium states in the MOJS algorithm.

4.3.5. Stability of the MOJS Algorithm in the Lyapunov Meaning

Assume that the initial condition state of the MOJS algorithm is within the hypersphere $S(\delta)$ with the equilibrium point x_{max} as the center and δ as the radius, then $X \in S(\delta)$ can represent $S(\delta) = \{X | \|X - x_{max}\| \le \delta\}$; that is:

$$\|X - x_{max}\| \le \delta \tag{66}$$

As shown in Figure 3, $S(\gamma)$ is a circle with a center radius of γ and a circle $S(\delta)$ with a center radius of δ, the points on both sides of the side are set as x_1, x_2, the circle $S(\gamma)$ and $f(x)$ intersect with x_3, x_4, assume $f(x)$ is the objective function graph, $f(x_{max})$ is the maximum value of the function, $f(x_{max1})$ is the next largest value of the function, and S_{max} is the region between the maximum value and the next largest value of the objective function, which is called the optimal region.

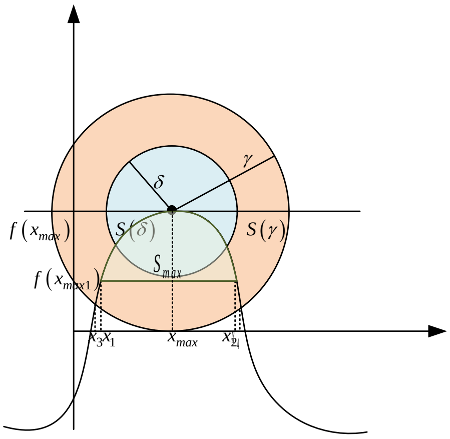


Figure 3. Stability of the MOJS algorithm in the Lyapunov meaning.

It is assumed that the MOJS algorithm satisfies the stability in the Lyapunov meaning, and the equilibrium state is uniformly asymptotically stable.

Proof: According to the global convergence of MOJS in Section 4.3.3, when X is in S_{max}, X will be attracted by food and move towards x_{max}, so the initial solution $X(t:\ X_0, t_0)$ of the equation is located in S_{max}, and S_{max} is included in the intersection region of $S(\delta)$ and $f(x)$, then X will not escape $S(\delta)$. Then δ satisfies:

$$\begin{cases} \delta \le \min(\|x_{max} - x_1\|, \|x_{max} - x_2\|) \\ \delta \le \min(f(x_{max}) - f(x_3), f(x_{max}) - f(x4)) \end{cases} \tag{67}$$

So

$$\|X(t:\ X_0, t_0) - X_{\mathrm{e}}\| \le \gamma,\ t \ge t_0 \tag{68}$$

Therefore, when $t \to \infty, \forall t,$ makes $X(t:\ X_0, t_0) \in S(\gamma)$, it satisfies the stability under the Lyapunov meaning.

If $\forall \gamma > 0$, $\exists \delta$ and δ satisfy formula (67), and the initial state x_0 satisfies $\|x_0 - x_{max}\| \le \delta$, then x_0 satisfies $\| X(t : X_0, t_0) - X_e \| \le \delta$. Therefore, it can be concluded that δ is independent of t_0, and the equilibrium state x_{max} of the MOJS algorithm is uniformly stable, which is proved.

So far, the stability proof of the membrane fouling fault diagnosis model based on MOJS-ADBN has been completed.

5. Simulation Experiment and Research Analysis

5.1. Membrane Fouling Data Acquisition

We used CFD software aimed at the problem that the membrane flux is easily affected by influent flow and temperature; this article used the parallel hollow fiber membrane device as the research object, and accurately classified the factors that cause membrane pollution. CFD software was used to simulate and calculate the water production in the MBR system to collect fault data.

Using the modeling process of the parallel hollow fiber membrane unit as an example, the Euler multiphase flow model was selected to simulate and build the MBR simulation system. The equation of mass and momentum conservation is as follows.

Mass-conservation equation:

$$\frac{\partial}{\partial \mathrm{t}}(\alpha_q \rho_q) + \nabla \cdot (\alpha_q \rho_q \mu_q) = 0 \tag{69}$$

In the formula, α_q is volume, ρ_q is density $(\mathrm{kg} \cdot \mathrm{m}^{-3})$, μ_q is he average velocity vector of q-th $(\mathrm{m} \cdot \mathrm{s}^{-1})$, and q is liquid s.

Momentum conservation equation:

$$\frac{\partial \alpha_q \rho_q \mu_q}{\partial \mathrm{t}} + \nabla g(\alpha_q \rho_q \mu_{qj} \mu_q) = -\alpha_q g \nabla p_q + \nabla g(\alpha_q \tau_q) + F_q + \alpha_q \rho_q g \tag{70}$$

In the formula, q represents the liquid phase, j represents x, y, z in three directions, α_q is volume fraction, μ_q is velocity $(\mathrm{m} \cdot \mathrm{s}^{-1})$, ρ_q is density $(\mathrm{kg} \cdot \mathrm{m}^{-3})$, P_q is pressure (Pa), τ_q is viscous stress tensor (Pa), F_q is interaction force $(\mathrm{N} \cdot \mathrm{m}^{-3})$, g is gravitational acceleration $(\mathrm{m} \cdot \mathrm{s}^{-2})$.

In the control of the solution, we set up the solution method at first. In the drop-down list of pressure–speed coupling, a phase-based coupling algorithm was selected to calculate the grid file. In the differential discrete format option, we set the gradient to cell-based least squares and the transient item format to first-order implicit. We set the monitoring window and convergence threshold. In the simulation data, we summarized nine types of membrane contamination data, such as too large, too small, and within the tolerance range; these data were collected for the main influencing factors of membrane contamination.

According to the analysis of the importance of membrane pollution factors, when the transmembrane pressure difference was constant, the above four influencing factors were selected as the research objects for analysis because the concentration difference of COD in and out water (*C*), BOD in and out water (*B*), solid concentration of mixed suspension (*X*) and hydraulic retention time (*H*) had obvious effects on membrane pollution. After testing and comparison, a tolerance of 5% was set for the COD concentration difference and BOD concentration difference of the inlet and outlet water in the series tubular membrane device, and a tolerance of 7% was set for the mixed suspension solid concentration and hydraulic retention time. A tolerance of 5% was set for the values of the membrane fouling factors in the parallel hollow fiber membrane device. When the membrane fouling factor value was within the set tolerance range, it indicated that there was no pollution in the series tubular membrane device. When the values of the membrane pollution factors exceeded the set tolerance, it meant that membrane pollution factors, such as COD concentration difference, BOD concentration difference, mixed suspension solid concentration, and hydraulic retention time were too large, resulting in membrane pollution. The types of membrane pollution are f2, f4, f6, and f8, respectively When the value of the membrane pollution factor was lower than the set tolerance, it indicated that the COD concentration difference, BOD concentration difference, mixed suspension solid concentration, and hydraulic retention time of the inlet and outlet water were too small, resulting in membrane pollution. The categories of membrane pollution are f3, f5, f7, and f9. Membrane pollution codes f1–f9 correspond to different membrane pollution types caused by "normal", "too large", and "too small" membrane pollution factors of the parallel hollow fiber membrane device in the actual operation of the membrane water treatment; see Table 1.

Table 1. Membrane fouling mode of the membrane device.

Fault Code	Fault Type	Tolerance
f1	No fouling	—
f2	C too large	5%
f3	C too small	5%
f4	B too large	5%
f5	B too small	5%
f6	X too large	7%
f7	X too small	7%
f8	H too large	7%
f9	H too small	7%

To better speed up the training of the network model, we made the data easy to calculate, obtained more generalized results, and the input data were standardized; the mathematical expression is:

$$X = \frac{X - X_{\min}}{X_{max} - X_{\min}} \tag{71}$$

5.2. Experimental Process

The experimental processes of this article are fault data collection, fault classification and coding, data pre-processing, data analysis and division, MOJS-ADBN model construction, prediction coding, and result analysis. The specific steps are as follows:

(1) Take membrane fouling data.

(2) Encode the data classification of membrane fouling.

(3) Classify the data into a training set and test set according to the ratio of 7:3.

(4) Build the MOJS-ADBN model, retain the weight in the unsupervised learning process, and use the adaptive learning rate to accelerate the training process. In the process of supervised learning, MOJS is used to optimize the algorithm and fine-tune the weight. The training set is used to adjust the network model to make the model optimal.

(5) Compare the actual code of the test set with the prediction code generated by the model. If the prediction code is consistent with the real coding result, the classification is correct; if the prediction code is inconsistent with the real coding result, the classification is wrong.

(6) Further analyze the model and judge the performance of the model from the perspective of average accuracy, average precision, average recall, and running time.

In this article, the MOJS-ADBN hidden layer was set as three layers, and the optimal number of hidden layer neurons was selected to determine the optimal number of hidden layer neurons based on the model error and running time. According to the experimental method, when the number of neurons in the hidden layer is 20, the performance effect is the best, as shown in the Figure 4. At this time, ape and MSE are 0.0618 and 0.0742, respectively. Figure 4 shows the relationship between the model error and the number of hidden layer neurons. In the formula, ape and MSE represent the absolute percentage error and mean square error, respectively.

$$\text{APE} = \frac{1}{N_t}\sum_{i=1}^{N_t}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{72}$$

$$\text{MSE} = \frac{1}{N_t}\sum_{i=1}^{N_t}(\hat{y}_i - y_i)^2 \tag{73}$$

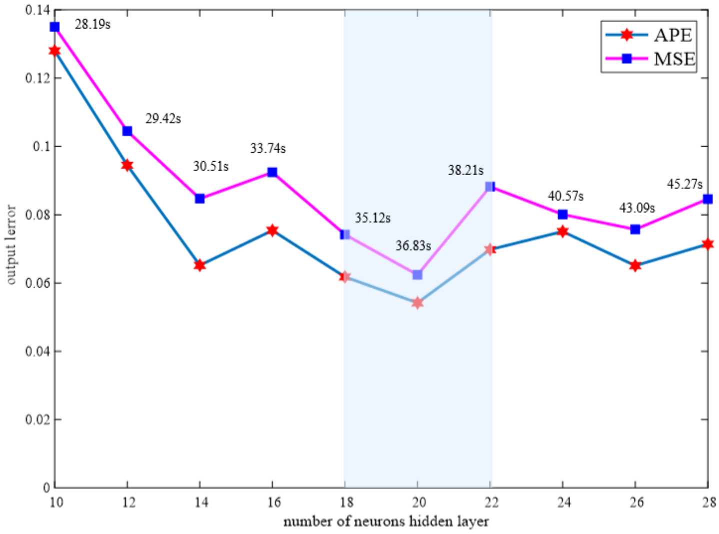


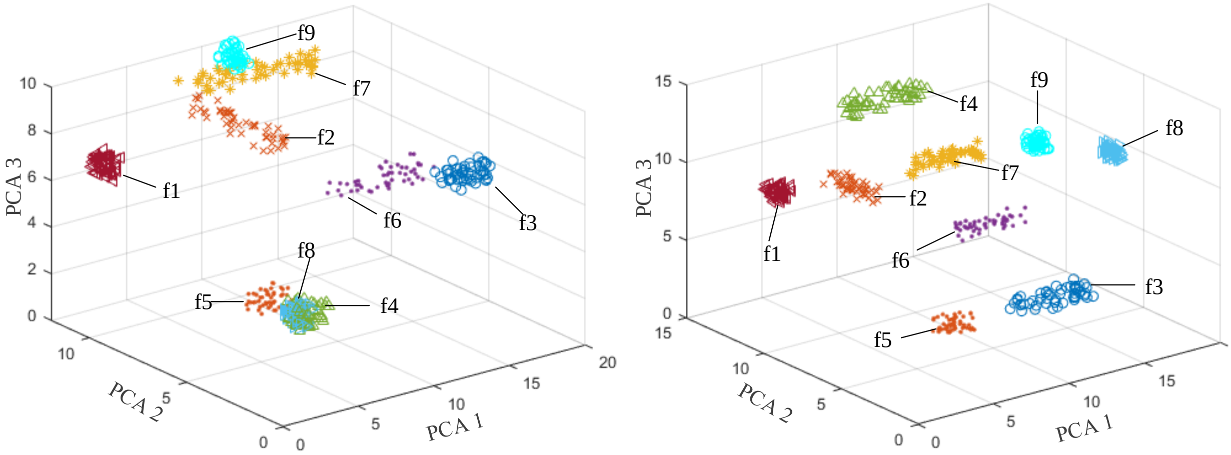
Figure 4. Relationship between the MOJS-ADBN model error and the number of hidden layer neurons.

In the formula, y_i and $\hat{y}_i$ represent the real value and predicted value, respectively, and N_t represents the number of test samples.

To objectively prove that the best model structure of MOJS-ADBN is 18-20-20-20-9, 300 data were collected for each membrane pollution category of the parallel hollow fiber membrane device, with a total of 2700 experimental data. A total of 1890 samples were randomly selected as training samples, and the remaining 810 samples were used as test samples.

In the unsupervised training phase, each RBM was set to iterate 378 times, and the learning rate coefficients were set to B = 1.4 and b = 0.7, respectively; the kernel principal component analysis (KPCA) was used to extract the three principal components of the first RBM output feature and the three principal components of the final DBN output feature, which are represented in Figure 5a,b respectively. It can be seen from Figure 5a that only f1 does not overlap with other faults; f2, f7, and f9 overlap, and the distribution of similar faults in f2 and f7 is relatively scattered. Moreover, f4, f5, and f8 overlap seriously, and the fault types cannot be classified correctly. Although f3 and f6 can be classified, there is still a small amount of overlap. It can be seen from Figure 5b that all kinds of faults do not overlap and can be classified better. Therefore, the DBN model can accurately distinguish other fault categories, and the distribution of similar faults is more compact than that in Figure 5a because the input data will undergo (four times) nonlinear mapping and the data will be reconstructed after passing through four RBMs, which can more accurately and abstractly express the input data.

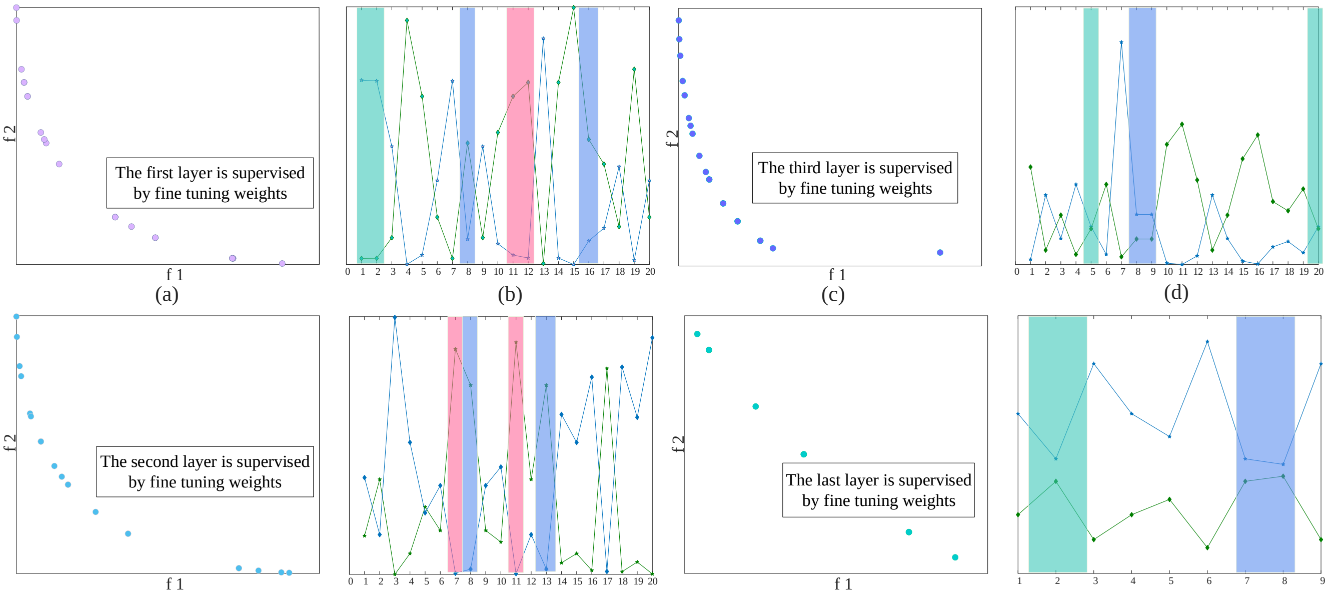
We used the MOJS algorithm for supervised fine adjustment. We set the three layers as in [9, 20, 20, 20], respectively, to establish the MOJS-ADBN model. Figure 6a,c,e,g represent the Pareto front scatter diagram, in the formula, and the abscissa and ordinate represent the objective function of the Pareto optimal solution respectively; while Figure 6b,d,f,h represent the Pareto frontier broken line graph, in the formula, the abscissa represents the number of Pareto optimal solutions, the two broken lines represent the objective functions of the Pareto optimal solutions, respectively, and the color block in the graph represents the overlapping part of the Pareto frontier scatter diagram. It can be seen from the graph that the weight can be improved after four times the MOJS algorithm optimization and supervised fine-tuning, make the weight distribution more reasonable.



(a) The first RBM extracts the principal component distribution of features

(b) Principal component distribution of DBN extracted features

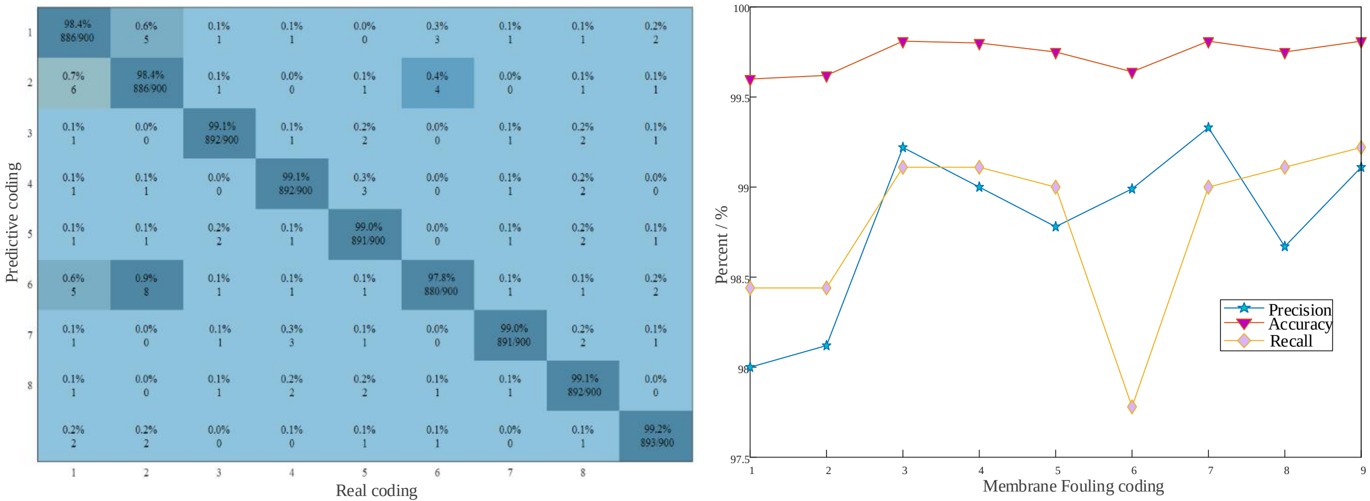
Figure 5. Principal component distribution of feature extraction.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6. Pareto frontier analysis of the MOJS-ADBN model.

To reduce the influence of experimental randomness on the evaluation of the model diagnostic performance, 10 independent diagnostic experiments were carried out on the parallel hollow fiber membrane device. Figure 7a presents the average confusion matrix of 10 diagnostic faults of the MOJS-ADBN model. From the figure, it can be seen that there are 9 fault codes from f1 to f9, and each fault is counted 900 times in total. In the formula, the total number of f1 misclassifications is 14. In the formula, misclassification is: f2—five times, f6—three times, f9—two times, and f3, f4, f7, and f8 are misclassified once each; the total number of false divisions is 14; f1 is classified six times, f6 is classified four times, and f3, f5, f8, and f9 are classified once each. The total number of f3 misclassifications is eight; f1, f4, f7, f9 are misclassified once each, and f5 and f8 are misclassified twice each. The total number of f4 misclassifications is 8; f1, f2, and f7 are misclassified once, f5 is misclassified three times, and f8—twice. The total number of misclassifications is 20. In the formula, misclassification is f1—five times, f2—eight times, exception misclassifications of f3, f4, f5, f7, and f8—once each, misclassification of f9—twice. The total number of f7 misclassifications is 9; f1, f3, f5, and f9 are misclassified once each, f4 is misclassified three times, f8 is misclassified two times. The total number of f8 misclassifications is 8; f1, f3, f6, and f7 are misclassified once each, f4 and f5 are misclassified twice. The total number of f9 misclassifications is eight. In the formula, misclassifications of f5, f6, and f8 are once each, and misclassifications of f1 and f2 are two times each. From the figure, it can be seen that f1, f2, and f6 are easy to be confused compared with the other faults. Figure 7b shows the curve of the accuracy, accuracy, and recall of all kinds of faults. From the figure, it can be seen that the accuracy, accuracy, and recall of all kinds of faults is above 97%; therefore, the MOJS-ADBN proposed in this article still has strong robustness.



	1	2	3	4	5	6	7	8	
1	98.4% 886/900	0.6% 5	0.1% 1	0.1% 1	0.0% 0	0.3% 3	0.1% 1	0.1% 1	0.2% 2
2	0.7% 6	98.4% 886/900	0.1% 1	0.0% 0	0.1% 1	0.4% 4	0.0% 0	0.1% 1	0.1% 1
3	0.1% 1	0.0% 0	99.1% 892/900	0.1% 1	0.2% 2	0.0% 0	0.1% 1	0.2% 2	0.1% 1
4	0.1% 1	0.1% 1	0.0% 0	99.1% 892/900	0.3% 3	0.0% 0	0.1% 1	0.2% 2	0.0% 0
5	0.1% 1	0.1% 1	0.2% 2	0.1% 1	99.0% 891/900	0.0% 0	0.1% 1	0.2% 2	0.1% 1
6	0.6% 5	0.9% 8	0.1% 1	0.1% 1	0.1% 1	97.8% 880/900	0.1% 1	0.1% 1	0.2% 2
7	0.1% 1	0.0% 0	0.1% 1	0.3% 3	0.1% 1	0.0% 0	99.0% 891/900	0.2% 2	0.1% 1
8	0.1% 1	0.0% 0	0.1% 1	0.2% 2	0.2% 2	0.1% 1	0.1% 1	99.1% 892/900	0.0% 0
	0.2% 2	0.2% 2	0.0% 0	0.1% 0	0.1% 1	0.1% 1	0.0% 0	0.1% 1	99.2% 893/900

(a) MOJS-ADBN model confusion matrix for diagnosis of membrane contamination

(b) Evaluation of diagnostic performance of MOJS-ADBN model for membrane fouling

Figure 7. Performance of the MOJS-ADBN model in membrane fouling diagnosis.

5.3. Comparative Test

5.3.1. Comparative Test of Different Learning Rates

As a probability model, RBM is mainly affected by weight, so a reasonable weight is the premise to ensure accurate network classification. Figure 8 shows the weights obtained by using the adaptive learning rate and fixed learning rate, respectively. As can be seen from the figure below, the weight distribution obtained by using the adaptive learning rate is more compact than that obtained by the fixed learning rate, which can effectively avoid the problems of ignoring detailed features or gradient disappearance caused by too large or too small weights.

In the past, the learning rate of the DBN was determined by experience. To further prove the applicability of the adaptive learning rate, comparative experiments were used to verify 0.01, 0.05, 0.1, 0.5, and 1 as the learning rates of RBM, the supervised learning part was fixed, and 10 experiments were carried out on the parallel hollow fiber membrane device as the research object. The training and test data were classified for verification. Table 2 shows the diagnostic comparison experiment. It can be seen from the table that the diagnostic accuracies of learning rates 0.1 and 1 were higher than that of other learning rates, but the adaptive learning rate proposed in this article not only ensured the accuracy but also accelerated the network convergence. Therefore, the adaptive learning rate proposed in this article, based on the setting of the parameter update direction, progressed compared to the traditional empirical method.

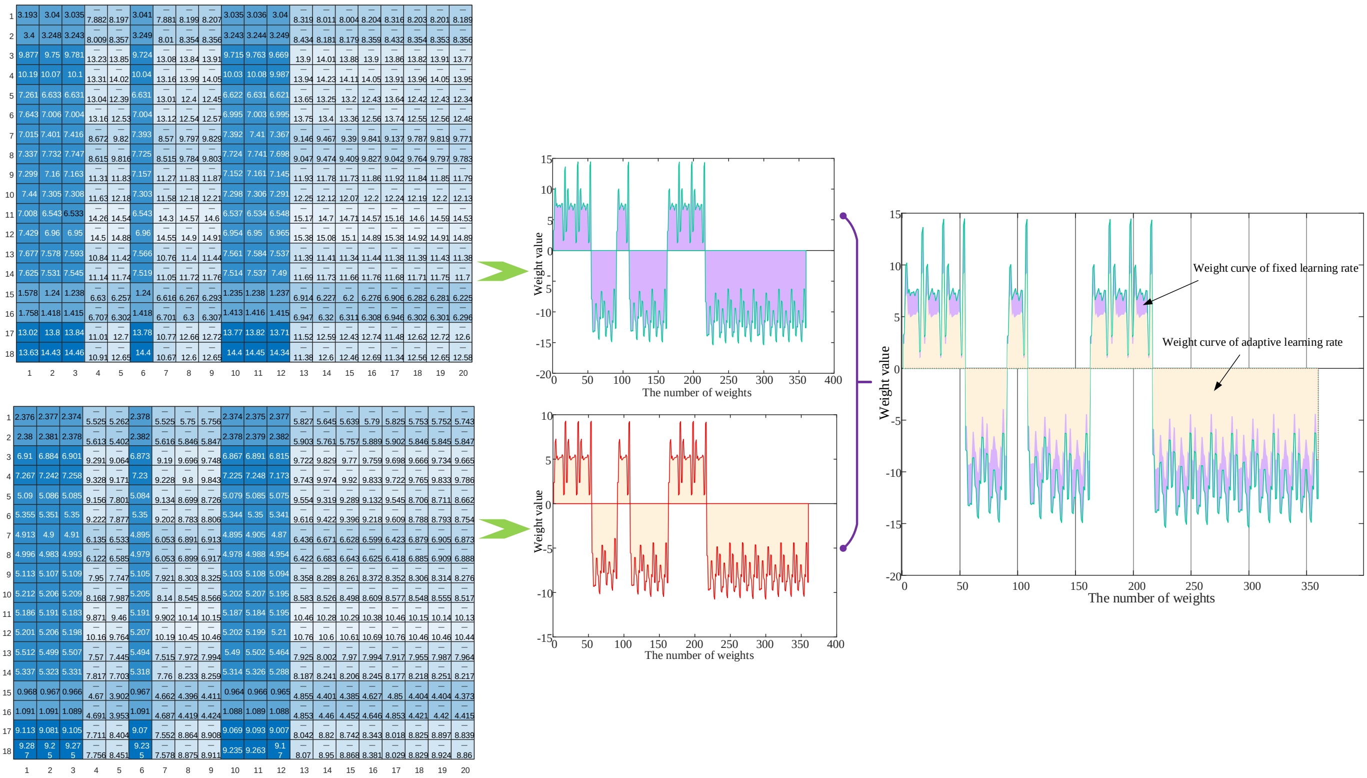


Figure 8. Comparison of optimization weights of different learning rates.

Table 2. Diagnostic accuracies of different fixed learning rates.

Learning Rate	Average Accuracy/%
0.01	95.26
0.05	93.73
0.1	96.21
0.5	94.57
1	96.75

5.3.2. Comparison of Ablation Experiments

To further prove the effectiveness and superiority of the MOJS-ADBN model for membrane device–membrane fouling diagnosis, this method is compared with some common fault diagnoses and classification methods. We combined wavelet transform with PCA to extract features. In the learning of the shallow neural network, BP, extreme learning machine (ELM), SVM, and least square support vector machines (LSSVM) were used for classification diagnosis. In deep learning, the traditional DBN and adaptive learning rate DBN (ALRDBN) were used, the data set was expanded by overlapping sampling, and then the convolutional neural network (CNN) was used for comparison. According to the method in this article, training data and test data were classified for 10 independent diagnosis experiments, and the comparison indicators included network structure, average time, mean value, and variance of the test MSE; the results are shown in Table 3. It can be seen from the table that compared with the shallow network, the DBN can effectively extract the essence and depth characteristics of faults. After optimization, the DBN improved both the accuracy and network performance to varying degrees, and the nonlinear mapping between the initial data and characteristics were more obvious. Compared with the deep network, although the CNN has a lower diagnosis time than MOJS-ADBN, the diagnosis rate of the improved CNN is lower than MOJS-ADBN, and the CNN needs a large number of data sets and reasonable division to ensure the rationality of the model, so the MOJS-ADBN proposed in this article is more conducive to the accurate identification of faults.

Table 3. Comparison of diagnostic performances of different models.

Diagnosis Method	Network Structure	Testing MSE		Average Time/s	Average Accuracy/%
		Mean	Variance		
BP	18-20-9	0.0294	0.0121	55.42	78.51
ELM	18-20-9	0.0313	0.0106	59.47	81.05
SVM	Gaussian Kernel Function	0.0251	0.0092	62.73	80.93
LSSVM	Gaussian Kernel Function	0.0247	0.0085	60.51	83.57
DBN	18-20-20-20-9	0.0218	0.0075	52.14	90.92
ALRDBN	18-20-20-20-9	0.0157	0.0053	34.91	93.75
Improved CNN	21 layers	0062	0.0035	20.97	95.72
MOJS-ADBN	18-20-20-20-9	0.0052	0.0027	35.12	98.79

The parallel hollow fiber membrane device membrane fouling simulation data set was used to carry out ablation experiments. Five performances, including average accuracy, average accuracy, average recall, average time, and average determination coefficient R^2 were used as the bases for the model judgment, and the performances of the DBN ALRDBN improved CNN, and MOJS-ADBN were verified, respectively.

According to the analysis in Figure 9, the performance of the improved model in this article improved to varying degrees. Although the reduction effect of the running time was not prominent, the accuracy significantly improved (besides the running time), while the other four performance effects of the MOJS-ADBN model were significantly better than the other three network models, which verifies the effectiveness and superiority of the MOJS-ADBN diagnostic model proposed in this article.

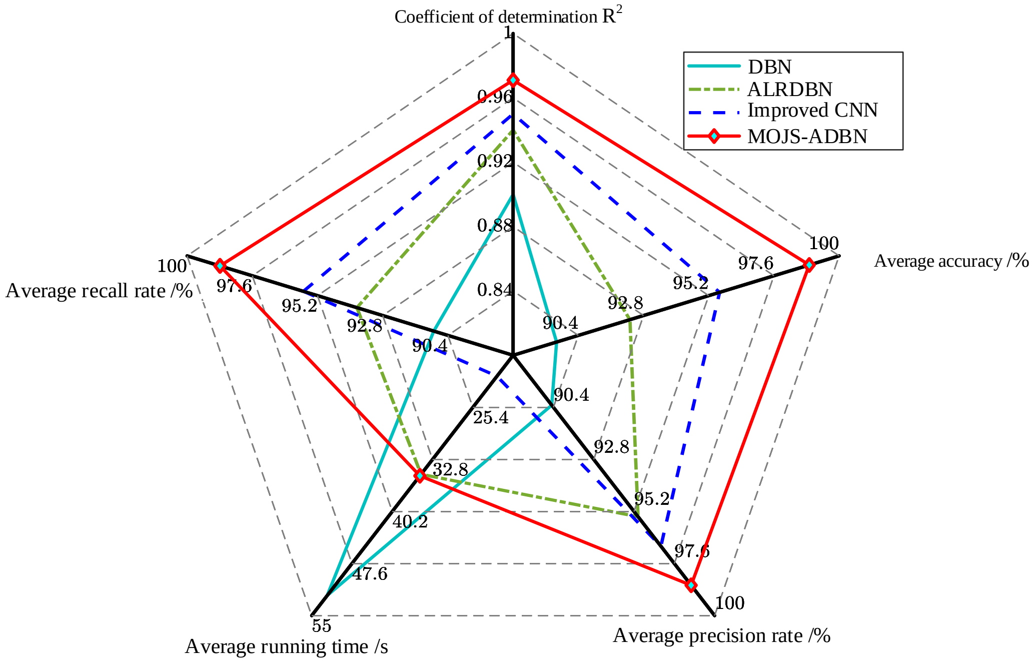


Figure 9. Performance comparison results of the ablation experiments.

5.3.3. Variable Noise Membrane Fouling Diagnosis Results of Different Diagnostic Methods

During the actual operation of the membrane bioreactor, there was environmental noise when the membrane component was treating sewage. At the same time, due to the characteristics of the membrane component itself, there was also noise, which produced unnecessary randomness in the collection of the membrane pollution data. At the same time, because the simulated data needed to be more consistent with the uncertainty of the operation of the membrane component under the actual working conditions, it was very important to add the variable noise experiment to the membrane fouling diagnosis experiment. To verify whether this method could obtain higher fault diagnosis accuracy and better generalization ability in the variable noise experiment, the experimental results of this article were compared with the experimental results of the methods proposed in references [34] and [36]. Reference [34] proposed a DBN fault diagnosis model with an improved model structure. The model uses multi-layer and multi-dimensional mapping to extract more detailed fault type differences and accurately diagnose faults. Reference [36] used the model after optimizing the parameters of a support vector machine to diagnose the signal features extracted by the DBN, realized the online detection of equipment faults, and improved the accuracy of the diagnosis. In this article, aimed at the membrane pollution data of the parallel hollow fiber membrane component as the training sample, Gaussian white noise (with SNRs of −2, 0, 2, and 4 dB) was added to the test sample, and the obtained membrane fouling diagnosis results were compared with other diagnostic methods. The experimental results are shown in Table 4

Table 4. Diagnosis accuracy rates of different methods under different noises.

Diagnostic Method	**SNR/dB**			
	−2	**0**	**2**	**4**
DBN	87.17%	91.08%	89.38%	90.74%
Reference [34]	94.11%	96.20%	96.03%	95.77%
Reference [36]	94.21%	95.97%	96.12%	96.33%
MOJS-ADBN	96.42%	98.94%	98.16%	98.23%

From Table 4, it can be seen (from the comparative data of four) that in the experimental results of different SNRs, the accuracy of the membrane component-membrane fouling diagnosis based on MOJS-ADBN was higher than that of other methods, and its anti-noise performance was stronger than the first three diagnostic methods.

6. Conclusions

This article presents a method of membrane packaging diagnosis based on MOJS-ADBN to optimize the DBN from the perspectives of unsupervised learning and supervised learning:

(1) The adaptive learning rate was used to accelerate the convergence of the network and proved that the unsupervised part optimized by the adaptive learning rate was stable.

(2) The supervised part used the MOJS algorithm optimization to fine-tune the weight, proving that MOJS optimization has global convergence and stability in the Lyapunov meaning.

(3) MOJS-ADBN was verified by a simulation experiment with a parallel hollow fiber membrane component. The experimental results show that the MOJS-ADBN model can effectively classify and locate faults, and can be used as a new solution in the field of membrane fouling diagnosis for membrane water treatment.

Author Contributions: Investigation, methodology, writing—review and editing, Y.S.; funding acquisition, investigation, Z.W.; investigation, methodology, X.D. and Y.L.; methodology, B.G.; writing—review and editing, L.L. and G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (61863026), the Science and Technology Program of Gansu Province (21ZD4GA028), and the Science Technology Foundation for Young Scientists of Gansu Province (21JR7RA246).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, Y.; Zhou, Z.; Cheng, C.; Wang, Z.W.; Pang, H.J.; Jiang, L.Y.; Jiang, L.M. Effects of packing carriers and ultrasonication on membrane fouling and sludge properties of anaerobic side-stream reactor coupled membrane reactors for sludge reduction. *J. Membr. Sci.* **2019**, *581*, 312–320. [CrossRef]
2. Du, X.J.; Shi, Y.K.; Jegatheesan, V.; Haq, I.U. A review on the mechanism, impacts and control methods of membrane fouling in MBR system. *Membranes* **2020**, *10*, 24. [CrossRef]
3. Wang, C.S.; Ng, T.C.A.; Ding, M.Y.; Ng, H.Y. Insights on fouling develop-ment and characteristics during different fouling stages between a novel vibrating MBR and an air-sparging MBR for domestic wastewater treatment. *Water Res.* **2022**, *212*, 118098. [CrossRef] [PubMed]
4. Kim, M.J.; Sankararao, B.; Yoo, C.K. Determination of MBR fouling and chemical cleaning interval using statistical methods applied on dynamic index data. *J. Membr. Sci.* **2011**, *375*, 345–353. [CrossRef]
5. Farias, E.L.; Howe, K.J.; Thomson, B.M. Effect of membrane bioreactor solids retention time on reverse osmosis membrane fouling for wastewater reuse. *Water Res.* **2014**, *49*, 53–61. [CrossRef] [PubMed]
6. Saravanan, N.; Ramachandran, K.I. Fault diagnosis of spur bevel gear box using discrete wavelet features and decision tree classification. *Expert Syst. Appl.* **2009**, *36*, 9564–9573. [CrossRef]
7. Zhang, Z.Q.; Mei, J.M.; Zhao, H.M.; Chang, C.; Shen, H. Early bearing fault feature extraction based on CEMP time-frequency features. *J. Vib. Shock* **2020**, *39*, 168–173. [CrossRef]
8. Wang, D.; Tsui, K.L.; Qin, Y. Optimization of segmentation fragments in empirical wavelet transform and its applications to extracting industrial bearing fault features. *Measurement* **2019**, *133*, 328–340. [CrossRef]
9. Ling, G.B.; Wang, Z.W.; Shi, Y.K.; Wang, J.Y.; Lu, Y.R.; Li, L. Membrane Fouling Prediction Based on Tent-SSA-BP. *Membranes* **2022**, *12*, 691. [CrossRef]
10. Hu, H.J.; Li, Y.X.; Liu, M.F.; Liang, W.H. Classification of defects in steel strip surface based on multiclass support vector machine. *Multimed. Tools Appl.* **2014**, *69*, 199–216. [CrossRef]
11. Yu, J. A particle filter driven dynamic Gaussian mixture model approach for complex process monitoring and fault diagnosis. *J. Process Control* **2012**, *22*, 778–788. [CrossRef]
12. Zhou, J.; Huang, F.N.; Shen, W.H.; Liu, Z.; Corriou, J.P.; Seferlis, P. Sub-period division strategies combined with multiway principle component analysis for fault diagnosis on sequence batch reactor of wastewater treatment process in paper mill. *Process Saf. Environ.* **2021**, *146*, 9–19. [CrossRef]
13. Mid, E.C.; Dua, V. Model-based parameter estimation for fault detection using multiparametric programming. *Ind. Eng. Chem. Res.* **2017**, *56*, 8000–8015. [CrossRef]
14. Mid, E.C.; Dua, V. Fault detection in wastewater treatment systems using multiparametric programming. *Processes* **2018**, *6*, 231. [CrossRef]
15. Wu, W.Q.; Song, C.Y.; Liu, J.; Zhao, J. Data-knowledge-driven distributed monitoring for large-scale processes based on digraph. *J. Process Control* **2022**, *109*, 60–73. [CrossRef]
16. Han, H.G.; Liu, H.X.; Liu, Z.; Qiao, J.F. Fault detection of sludge bulking using a self-organizing type-2 fuzzy-neural-network. *Control Eng. Pract.* **2019**, *90*, 27–37. [CrossRef]
17. Shi, Y.K.; Wang, Z.W.; Du, X.J.; Gong, B.; Jegatheesan, V.; Haq, I.U. Recent advances in the prediction of fouling in membrane bioreactors. *Membranes* **2021**, *11*, 381. [CrossRef]
18. Hoang, D.T.; Kang, H.J. A survey on Deep Learning based bearing fault diagnosis. *Neurocomputing* **2019**, *335*, 327–335. [CrossRef]
19. Tang, S.; Yuan, S.; Zhu, Y. Deep learning-based intelligent fault diagnosis methods toward rotating machinery. *IEEE Access* **2020**, *9*, 9335–9346. [CrossRef]
20. Chang, P.; Li, Z.Y.; Wang, G.M.; Wang, P. An effective deep recurrent network with high-order statistic information for fault monitoring in wastewater treatment process. *Expert Syst. Appl.* **2021**, *167*, 114141. [CrossRef]
21. Ba-Alawi, A.H.; Loy-Benitez, J.; Kim, S.; Yoo, C. Missing data imputation and sensor self-validation towards a sustainable operation of wastewater treatment plants via deep variational residual autoencoders. *Chemosphere* **2022**, *288*, 132647. [CrossRef] [PubMed]
22. Shi, Y.K.; Wang, Z.W.; Du, X.J.; Gong, B.; Lu, Y.R.; Li, L. Membrane Fouling Diagnosis of Membrane Components Based on Multi-feature Information Fusion. *J. Membr. Sci.* **2022**, *657*, 120670. [CrossRef]
23. Shi, Y.K.; Wang, Z.W.; Du, X.J.; Ling, G.B.; Jia, W.C.; Lu, Y.R. Research on the membrane fouling diagnosis of MBR membrane module based on ECA-CNN. *J. Environ. Chem. Eng.* **2022**, *10*, 107649. [CrossRef]
24. Ji, D.X.; Yao, X.; Li, S.; Tang, Y.G.; Tian, Y. Model-free fault diagnosis for autonomous underwater vehicles using sequence convolutional neural network. *Ocean Eng.* **2021**, *232*, 108874. [CrossRef]
25. Miyata, S.; Lim, J.; Akashi, Y.; Kuwahara, Y.; Tanaka, K. Fault detection and diagnosis for heat source system using convolutional neural network with imaged faulty behavior data. *Sci. Technol. Built Environ.* **2019**, *26*, 1–9. [CrossRef]
26. Liu, H.; Zhang, H.; Zhang, Y.; Zhang, F.; Huang, M. Modeling of wastewater treatment processes using dynamic Bayesian networks based on fuzzy PLS. *IEEE Access* **2020**, *8*, 92129–92140. [CrossRef]
27. Tamilselvan, P.; Wang, P.F. Failure diagnosis using deep belief learning based health state classification. *Reliab. Eng. Syst. Safe* **2013**, *115*, 124–135. [CrossRef]

28. Deng, W.; Liu, H.; Xu, J.; Zhao, H.; Song, Y. An improved quantum-inspired differential evolution algorithm for deep belief network. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 7319–7327. [CrossRef]
29. Wang, Y.L.; Pan, Z.F.; Yuan, X.F.; Yang, C.H.; Gui, W.H. A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Trans.* **2020**, *96*, 457–467. [CrossRef]
30. Zhao, G.; Liu, X.; Zhang, B.; Liu, Y.; Niu, G.; Hu, C. A novel approach for analog circuit fault diagnosis based on deep belief network. *Measurement* **2018**, *121*, 170–178. [CrossRef]
31. Hinton, G.E.; Osindero, S.; The, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
32. Liu, Z.; Jia, Z.; Vong, C.M.; Bu, S.; Han, J.; Tang, X. Capturing high-discriminative fault features for electronics-rich analog system via deep learning. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1213–1226. [CrossRef]
33. Zhang, C.; He, Y.; Yuan, L.; Xiang, S. Analog circuit incipient fault diagnosis method using DBN based features extraction. *IEEE Access* **2018**, *6*, 23053–23064. [CrossRef]
34. Dai, J.; Song, H.; Sheng, G.; Jiang, X. Dissolved gas analysis of insulating oil for power transformer fault diagnosis with deep belief network. *IEEE Trans. Dielectr. Electr. Insul.* **2017**, *24*, 2828–2835. [CrossRef]
35. Zhu, D.; Cheng, X.; Yang, L.; Chen, Y.S.; Yang, S.X. Information fusion fault diagnosis method for deep-sea human occupied vehicle thruster based on deep belief network. *IEEE Trans. Cybern.* **2021**, *99*, 3055770. [CrossRef]
36. Su, X.; Cao, C.; Zeng, X.; Feng, Z.; Shen, J.; Yan, X.; Wu, Z. Application of DBN and GWO-SVM in analog circuit fault diagnosis. *Sci. Rep.* **2021**, *11*, 7969. [CrossRef] [PubMed]
37. Zhu, J.; Hu, T.; Jiang, B.; Yang, X. Intelligent bearing fault diagnosis using PCA–DBN framework. *Neural Comput. Appl.* **2020**, *32*, 10773–10781. [CrossRef]