IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 31, 2019, accepted January 13, 2020, date of publication January 21, 2020, date of current version January 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968508

An Improved Parallel MDBN With AVMD for Nonlinear System Modeling

QIBING JIN, HENGYU ZHANG, YUMING ZHANG, WU CAI, AND MEIXUAN CHI
College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

Corresponding author: Hengyu Zhang (zhanghybuct@163.com)

This work was supported in part by the National Natural Science Foundation of China (Researches on robust identification of system models in the presence of complex heavy-tailed noises) under Grant 61673004, and in part by the Fundamental Research Funds for the Central Universities of China under Grant XK1802-4.

ABSTRACT Nonlinear system modeling using Deep Belief Network (DBN) is currently a research hotspot. However, the training process of DBN needs large amount of data to guarantee accuracy, and the traditional DBN may not meet the requirement of high-precision modeling. In this paper, we first improve the original DBN and Variational Mode Decomposition (VMD) algorithms, and on this basis, we then proposed a parallel Momentum Deep Belief Networks (MDBN) with Adaptive Variational Mode Decomposition (AVMD). Parallel AVMD-MDBN is an improved modeling method based on the deep learning model DBN. Firstly, a single raw dataset is decomposed into a specific number of sub-datasets using AVMD. Then these sub-datasets are distributed among a number of improved MDBNs. A single raw dataset learning model and algorithm is extended to multiple feature extraction nodes to learn the characteristics of multiple sub-datasets in parallel. Finally, the results of the multiple nodes are transmitted to the main feature extraction node to complete the regression calculation. In order to verify the effectiveness of the model, the proposed parallel AVMD-MDBN model is tested on a nonlinear dynamic system modeling, a Mackey-Glass time-series prediction and a financial stock prediction. Our experimental results show that the proposed parallel AVMD-MDBN has better performances in terms of improving feature learning ability than that of other methods.

INDEX TERMS Parallel momentum deep belief networks, adaptive variational mode decomposition, contrast divergence algorithm, nonlinear system modeling, financial stock prediction.

I. INTRODUCTION

In the field of process control, the research of control theory is relatively perfect [1], [2]. However, most control theories require highly accurate model of practical processes that are complex and nonlinear, which brings a great challenge to system modeling [3]. What's more, in financial field, nonlinear systems are hard to be modeled due to their unknown structures and parameters [4]. Therefore, nonlinear system modeling is an important and challenging task, which has attracted extensive attention in many fields.

It has been proved that neural networks can approximate any nonlinear systems with high precision [5]. The study of artificial neural networks is extensively ranged [6]. Wu et al. proposed a novel learning algorithm for dynamic fuzzy neural network (D-FNN) based on extended radial basis function neural networks. The algorithm adopts the hierarchical online self-organizing learning paradigm, which is superior in terms of modeling complexity and modeling efficiency [7]. In order to improve the modeling ability of neural network for nonlinear systems, Li et al. proposed a self-organizing cascade neural network (SCNN) with random weights to construct the optimal network [8]. Han et al. investigated an automatic axon neural network (AANN) that can perform self-organizing architectures and weights while improving the network performance of nonlinear system modeling [9]. For the modeling of nonlinear systems in financial field, Zhang et al. proposed a quantile regression-radial basis function (QR-RBF) neural network model to predict soybean prices [10]. These above-mentioned algorithms only considered the single hidden layer architecture. To significantly improve the modeling accuracy, there should be enough (even equal to the number of training samples) hidden units. However, a neural network with that number of hidden units is impractical, especially

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

under the condition of numerous training samples. As a result, we cannot implement a neural network with unlimited hidden units to increase the modeling accuracy. Therefore, it is still a tough issue to break through the limitation of the accuracy of nonlinear system modeling.

In recent years, the deep belief network (DBN) based on restricted Boltzmann machine (RBM) can achieve desired modeling accuracy with less hidden neurons [11]. DBN is the superposition of successive restricted Boltzmann machines, which is commonly recognized as a deep neural network with multiple hidden layers, and the depth of DBN is reflected in its number of hidden layers. When the network is forward-operated, the output of the pre-RBM is treated as the input of the post-RBM. The DBN learning process consists of two parts: unsupervised learning and supervised learning. The unsupervised learning is implemented by Contrast Divergence (CD) algorithm to initialize the weights of DBN with a goal of minimizing the network energy of the RBM [12], which is superior to the random weight initialization in ANNs [13]. The supervised learning uses the error-propagation algorithm to fine-tune the initial weights produced by unsupervised learning. Due to the efficient ability of extracting features from sample data, DBN is applied to nonlinear system modeling [14]. Based on the strong forecasting ability of ARIMA model and the powerful expression ability of DBN on nonlinear relationships, Qin et al. proposed a hybrid model combining ARIMA and DBN for red tide forecasting, which shows good results [15]. However, the unsupervised learning of DBN starts from the bottom layer (input layer) to the top layer (target output), which is always time-consuming and easily causes local minimum or even training failure [16]. It should be pointed out that the local minimum poses a huge threat to the learning ability, which causes not only low efficiency but also low accuracy for nonlinear system modeling. Zhao et al. proposed a master-slave parallel computing method for the DBN learning process, to reduce the time consumption of pre-training and fine-tuning [17]. Ahn et al. proposed a virtual shared memory framework for the DNN learning process, called Soft Memory Box (SMB), which can share the memory of remote node among distributed processes in the nodes, thereby improving communication efficiency by parameter sharing [18]. Wei et al. proposed a medium-term load forecasting model for power supply units based on DBN and Apache Spark parallel processing platform, which improves the prediction speed [19]. However, the above algorithms mainly focused on the modification of structures and improvement of efficiency, but not the improvement of modeling accuracy. Zhou et al. designed a new generator and discriminator of Generative Adversarial Network (GAN) to generate more discriminant fault samples using a scheme of global optimization. The effectiveness of the algorithm is proved in rolling bearing experiments [20]. Because the accuracy of the prediction is not only related to data size and the architecture of the model, but also closely related to the quality of the data to be analyzed, data preprocessing before establishing a new model is also an important research direction [21].

In order to improve the quality and efficiency of DBN nonlinear system modeling, this paper proposes a parallelized structure DBN with variational mode decomposition (VMD). The procedure of data parallel computing studied in this paper is to firstly use VMD to decompose the raw dataset into several sub-datasets and then assign these sub-datasets to several computing nodes. The computing nodes execute their respective algorithms to process the corresponding sub-datasets. We use ‘‘coarse-grained’’ parallelization for data parallel computing, which can mine the deep features of the data and improve the effectiveness of the algorithm. In order to further optimize the proposed model, the DBN and VMD algorithms are improved respectively. Finally, a parallel computing model with adaptive variational mode decomposition (AVMD) and momentum deep belief network (MDBN), which can be summarized as AVMD-MDBN, is proposed for nonlinear system modeling. The AVMD-MDBN parallel computing model proposed in this paper is established as follows: First, the VMD based on the conservation law of frequency energy can determine the optimal number of decompositions and adaptively decompose the raw dataset into multiple sub-datasets with different characteristics. Then feature extraction is performed separately by distributing each sub-dataset to a DBN with an energy factor. Finally, the distributed features extracted by multiple parallel MDBNs are mapped to the sample mark space, thus constructing a nonlinear system model with higher precision. In this paper, a nonlinear dynamic system modeling, a Mackey-Glass time-series prediction and a financial stock prediction are carried out by the proposed AVMD-MDBN parallel computing model, and the superiority is verified.

The rest of this paper is organized as follows: The second part briefly introduces the DBN. The third part presents the details of the structure, learning process and complexity analysis of the proposed AVMD-MDBN parallel computing model. The fourth part presents the experimental results and discussions, demonstrating the performance of the proposed AVMD-MDBN compared with other existing similar methods. Finally, the fifth part is devoted to some conclusions and future work.

II. DBN

The deep belief network is a probability generation model that is composed of multiple RBMs [22]. The bottom layer of the deep belief network receives the input data vector and maps the input data to the hidden layer through RBM. Overall, the input of the higher layer RBM is from the output of the lower layer RBM. Below we will introduce the structure and principle of RBM, and then give the structure and training process of the deep belief network.

A. RESTRICTED BOLTZMANN MACHINE

Statistics show that any probability distribution can be transformed into an energy-based model, so RBM can provide

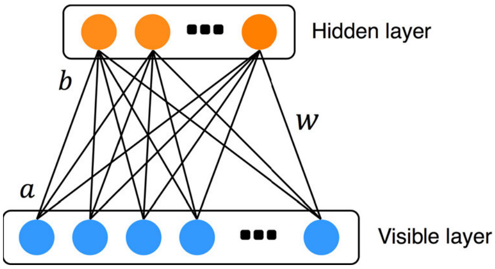


FIGURE 1. The basic structure of the RBM model.

a learning model for data that does not know the internal distribution [23], [24]. Each RBM contains a visible layer and a hidden layer. Only the connections between the visible layer and the hidden layer have bidirectional weights. There are no connections between the same layer units. Fig. 1 shows the basic structure of the RBM model [25]. Given the visible layer unit vector $v = \{v_1, v_2, v_3, \ldots v_m\} \in (0, 1)$, the hidden layer unit vector $h = \{h_1, h_2, h_3, \ldots h_n\} \in (0, 1)$, the weight matrix w, the threshold of the visible layer unit a, and the threshold of the hidden layer unit b, we can describe the energy function of the joint states $E(v, h)$ of all visible and hidden units as:

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

where m is the number of visible units and n is the number of hidden units. According to Eq. (1), the joint probability distribution between the hidden layer and the visible layer can be described as follows:

$$P(v, h) = \frac{e^{-E(v,h)}}{Z} \tag{2}$$

$$Z = \sum_v \sum_h e^{-E(v,h)} \tag{3}$$

where Z is a normalized constant that simulates a physical system, which is obtained by adding the energy values between all visible and hidden layer units [26]. Through the joint probability distribution of Eq. (2), the independent distribution of the visible layer vector v can be obtained as:

$$P(v) = \sum_h P(v, h) = \frac{\sum_h e^{-E(v,h)}}{\sum_v \sum_h e^{-E(v,h)}} \tag{4}$$

Since there is no connection between any two units of a same layer in RBM, when a random input visible layer vector v is given, all hidden layer units are independent of each other. Therefore, according to the joint probability distribution of Eq. (2), the probability of the hidden layer vector h is given by Eq. (5) under the condition that a visible layer vector v is given. Similarly, given a random input hidden layer vector h, the probability of the visible layer vector v is shown in Eq. (6):

$$P(^{h}/_{v}) = \prod_j P(h_j = ^{1}/_{v}) \tag{5}$$

$$P(^{v}/_{h}) = \prod_i P(v_i = ^{1}/_{h}) \tag{6}$$

Considering that the structural unit of RBM is a binary state, under the premise of defining the logical sigmoid function $(x) = 1/(1+e^{-x})$, the activation probability can be obtained as follows:

$$P\left(h_j = 1\,|v\right) = sig\left(b_j + \sum_{i=1}^{m} w_{ji} v_i\right) \tag{7}$$

$$P(v_i = 1\,|h) = sig\left(a_i + \sum_{j=1}^{n} w_{ji} h_j\right) \tag{8}$$

According to Eq. (7) and Eq. (8), the state of the hidden layer unit can be calculated by $P\left(h_j = 1/v\right)$ after the visible layer vector h is given. The state of reconstructing the visible layer unit $v_{Refactoring}$ is calculated by $P\left(v_i = 1/h\right)$. By a certain rule, the difference between the visible layer unit and the reconstructed visible layer unit is minimized, and the hidden layer unit is considered to be another expression of the visible layer unit. Therefore, the hidden layer unit can be used as the feature extraction result of the visual layer input unit, thereby achieving the purpose of feature extraction [27].

The essence of RBM is that the learned RBM model has the highest probability of coincident with the input sample distribution. That is, given the training data, the value of the probability $P(v)$ of Eq. (4) can be maximized by adjusting the corresponding parameters [28]. It can be known from Eq. (4) that in order to maximize the value of the probability $P(v)$, we can reduce the energy function value by adjusting the weight matrix w the threshold a of the visible layer unit, and the threshold b of the hidden layer unit, thereby indirectly increasing the probability value of $P(v)$ [29]. By using the maximum likelihood estimation, we can learn the parameter $\theta = \left\{a_i, b_j, w_{ji}\right\}$ of the RBM model from the training samples to maximize the value of probability $P(v)$ [30]. As shown in the following Eq. (9).

$$\begin{aligned}\frac{\partial \log P(v)}{\partial \theta} &= \frac{\partial \log\left(\frac{\sum_h e^{-E(v,h)}}{\sum_v \sum_h e^{-E(v,h)}}\right)}{\partial \theta} \\ &= -\sum_h P\left(^{h}/_{v}\right)\frac{\partial E(v,h)}{\partial \theta} + \sum_v \sum_h P(v,h)\frac{\partial E(v,h)}{\partial \theta}\end{aligned} \tag{9}$$

In the above Eq. (9), $\frac{\partial E(v,h)}{\partial \theta}$ can be calculated directly. However, if $P(h/v)$ and $P(v/h)$ are directly calculated, it will take a lot of time [31], [32]. This is obviously not advisable. According to Alternating Gibbs Sampling (AGS), also known as Markov Chain Monte Carlo (MCMC), we can extract samples that match the probability distributions of $P(h/v)$ and $P(v, h)$ from the training data. Perform an unbiased log-likelihood estimation for Eq. (9). In each iteration of alternate Gibbs sampling, all hidden layer units are updated by Eq. (7).

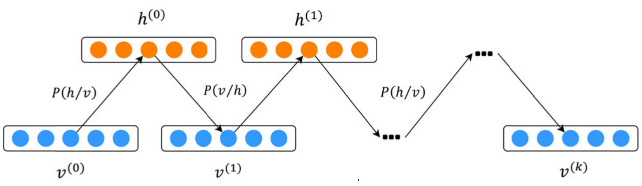


FIGURE 2. Alternating Gibbs sampling of the RBM.

All visual units are updated by Eq. (8) based on the updated hidden layer units. The detailed process is shown in Fig. 2.

According to the process described in Fig. 2, Eq. (9) can be updated as follows:

$$\Delta w_{ij} = \frac{\partial \log P(v)}{\partial \theta} = -E_{data}\left(v_i h_j\right) + E_{\bmod el}\left(v_i h_j\right) \quad (10)$$

In Eq. (10), $E_{data}(v_i h_j)$ represents the expectation of the data distribution, and $E_{model}(v_i h_j)$ represents the expectation of the model distribution. By using the contrast divergence algorithm, the infinity in $E_{model}(v_i h_j)$ can be replaced by n times. The algorithm(CD) shows good results when $k = 1$ (CD-1) [33].

B. DBN STRUCTURE

As mentioned above, the deep belief network can be seen as the superposition of multiple RBMs. By stacking multiple RBMs, deep features can be extracted from complex data. However, stacking RBM can only obtain some high-level features from complex raw data, and it is not able to directly classify the data. To get a complete DBN model, we also need to add a traditional supervised classifier to the top of the stacked RBM [34]. The number of nodes in the input layer is determined by the dimension of the input data, and the number of nodes in the output layer is determined by the number of categories of input data.

C. DBN TRAINING

The training of the deep belief network [35] consists of two processes: unsupervised layer-by-layer pre-training and supervised fine-tuning. The main difference between the deep belief network model and other models is that deep belief network consists an unsupervised layer-by-layer pre-training, which can learn nonlinear complex functions by directly mapping data from input to output. This is also the key to its powerful feature extraction capabilities. First, a vector is generated in the visible layer of the first RBM, and then the value is passed to the hidden layer through the RBM network, after which the visible layer is used to reconstruct the visible layer. The weight between the hidden layer and the visible layer is updated according to the difference between the reconstructed layer and the visible layer, until the maximum number of iterations is reached [36]. After the unsupervised training is completed, the deep belief network is supervised by adding tag data at the top of the deep belief network. That is, the back propagation (BP) is used to fine-tuning the relevant parameters of the deep belief network. Supervised training of the deep belief network will further reduce training errors and improve the accuracy of the deep belief network model. Compared to the one layer of training in unsupervised training, the inverse fine-tuning of supervised training is to update the parameters of all layers simultaneously. The process of unsupervised learning and supervised learning in the deep belief network is shown in Fig. 3.

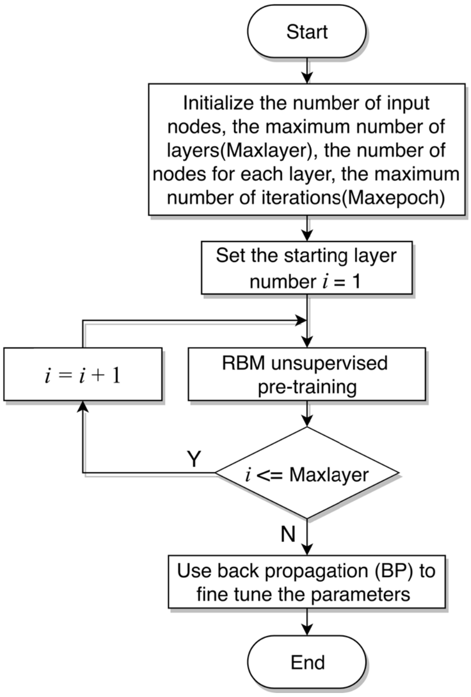

Start
Initialize the number of input nodes, the maximum number of layers(Maxlayer), the number of nodes for each layer, the maximum number of iterations(Maxepoch)
Set the starting layer number i = 1
$i = i + 1$
RBM unsupervised pre-training
Y
i <= Maxlayer
N
Use back propagation (BP) to fine tune the parameters
End

FIGURE 3. The process of unsupervised learning and supervised learning.

III. AVMD-MDBN

In our work, we first introduce the DBN with momentum factor and the adaptive VMD with frequency energy conservation. Then, the AVMD-MDBN and its learning algorithm are introduced in detail.

A. MOMENTUM DBN

It is not difficult to see from Section 2.1 that the core of the CD-k algorithm for the DBN unsupervised learning process is Gibbs sampling, and Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm. In the case of direct sampling difficulties, it can be used to generate a Gibbs chain to approximate a given multivariate distribution of samples [37]. The process of Gibbs sampling mainly includes the following two steps.

(1) Initialize the Gibbs chain with the data v of the observed sample to obtain the original input vector $v^{(0)}$ of the visible layer.

(2) According to Eq. (7) and Eq. (8), $h^{(t)}$ is sequentially sampled from $P\left(h^{(t)}/v^{(t)}\right)$ by iteration, and $v^{(t+1)}$ is sampled from $v^{(t+1)}$. t is the number of sampling steps, and the larger $t+1$, the better the effect.

Since a one-step Gibbs sampling can be used to get a good enough approximation, we make $k = 1$ in the CD-k algorithm.$(v^{(1)}, h^{(1)})$ is sampled from the model and is a rough estimate of $E_{model}(v_i h_j)$. Using $(v^{(1)}, h^{(1)})$ to estimate $E_{model}(v_i h_j)$, we can get the CD-1 algorithm. In the CD-1 algorithm, since each RBM requires multiple iterations, and the parameter update direction after each iteration is not the same, the fixed learning rate may cause the algorithm to be "premature" or difficult to converge. Therefore, it is critical to make the algorithm adaptively control the learning speed according to the way the parameters are updated [38]. This section gives a method for updating the learning rate based on the amount of change in the parameters of the two iterations during the RBM training. The improved step strategy is to add momentum factors ρ and ε based on the relationship between the two variations described above, which is used to accelerate the RBM training process and avoid falling into local optimum. The learning rate update mechanism for adding the momentum factor is given by:

When $\left|\mathrm{V}w_{ij}^{(\tau+1)}\right| \geq \left|\mathrm{V}w_{ij}^{(\tau)}\right|$:

$$lr = \begin{cases} \rho \times lr\Delta \mathrm{w}_{ij}^{(\tau+1)} \times \Delta \mathrm{w}_{ij}^{(\tau)} \geq 0 \\ \varepsilon \times lr\Delta \mathrm{w}_{ij}^{(\tau+1)} \times \Delta \mathrm{w}_{ij}^{(\tau)} < 0 \end{cases} \tag{11}$$

When $\left|\mathrm{V}w_{ij}^{(\tau+1)}\right| < \left|\mathrm{V}w_{ij}^{(\tau)}\right|$:

$$lr = \begin{cases} \rho \times lr & \Delta \mathrm{w}_{ij}^{(\tau+1)} \times \Delta \mathrm{w}_{ij}^{(\tau)} < 0 \\ \varepsilon \times lr & \Delta \mathrm{w}_{ij}^{(\tau+1)} \times \Delta \mathrm{w}_{ij}^{(\tau)} \geq 0 \end{cases} \tag{12}$$

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

$$\left(\Delta w_{ij}^{R}\right)^{(i+1)} = v_i^{(t+1)} h_i^{(t+1)} - v_i^{(t+2)} h_i^{(t+2)} \tag{14}$$

where τ is the number of iterations, ρ and ε are the momentum factors and $0 < \varepsilon < 1 < \rho$. The last learning rate is adjusted by multiplying a momentum factor. If the momentum factor is larger than 1, the convergence speed can be accelerated. If the momentum factor is less than 1, the convergence speed can be stabilized. Whether to accelerate or stabilize is determined by the step size of the previous step and the model update amount of two adjacent iterations. Unlike the original step strategy, the step size in the improved step strategy is the product of the previous step size and the momentum factor determined by the model variation of two successive iterations. The principle of adaptive learning rate with momentum factor added is shown in Eq. (11-14). Next, we illustrate the role of the introduction of the momentum term. After the training samples are sequentially added, the parameter update amount Δw_{ij} of two consecutive iterations is calculated. If the Δw_{ij} of the current and the next iterations are both positive or negative and the amount of change becomes larger, the step size multiplies the amount ρ, thereby accelerating the convergence speed. If the Δw_{ij} of the current and the next iterations are both positive or negative and the amount of change becomes smaller, the step size multiplies the amount ε, thereby stabilizing the convergence state and improving the convergence accuracy. It can be known from the CD algorithm that the weight is updated once in a Gibbs sampling period, the binary sampling in the intermediate state is performed twice and the weight of each update is proportional to the state sampling. Therefore ρ is approximately equal to twice the ε. The RBM with the improved step strategy of the momentum factor can effectively avoid the risk of falling into local optimum and speeds up the training. In particular, the DBN is stacked by a plurality of RBMs, so the DBN superimposes the above-described improved effects.

B. ADAPTIVE VMD

The variational mode decomposition (VMD) algorithm searches for the optimal solution of the variational mode model by iteratively calculating the center frequency and bandwidth of each eigenmode component, and then adaptively splitting the frequency of the signal and effectively separates the components [39].

In theory, each eigenmode component of the VMD decomposition is a finite bandwidth with a center frequency. Under this precondition, when we use VMD to decompose the signal into P eigenmode functions $u_p(t)$, two conditions need to be satisfied: First, the decomposition should minimize the sum of the estimated bandwidths of the modal components; Second, the decomposition is such that the sum of the modal components is equal to the input signal. The specific process is as follows:

(1) The eigenmode functions $u_p(t)$ can be regarded as a modulated signal, and Hilbert transformation is performed on $u_p(t)$ to obtain an analytical signal. The analytical signal is multiplied by $e^{-j\omega_p t}$ to modulate the spectrum of each $u_p(t)$ to the corresponding baseband.

$$\left[\left(\delta(t) + \frac{j}{\pi t}\right) u_p(t)\right] e^{-j\omega_p t} \tag{15}$$

(2) By calculating the norm of the square L^2 of the demodulated signal gradient after translation, the bandwidth of each modal signal is estimated, and the constrained variational problem is obtained as follows:

$$\min_{\{u_p\},\{\omega_p\}} \left\{ \sum_p \left\| \partial(t) \left[\left(\delta(t) + \frac{j}{\pi t}\right) u_p(t)\right] e^{-j\omega_p t} \right\|^2 \right\} \tag{16}$$

where $\{u_p\}$ is the modal component of the VMD decomposition, and $\{\omega_p\}$ is the center frequency of each modal component. In the variational problem, the quadratic penalty factor α and the Lagrangian penalty factor are introduced to obtain the optimal solution [40].

$$\begin{aligned} &L\left(\{u_p\}, \{\omega_p\}, \lambda\right) \\ &= \sum_p \left\| \partial(t) \left[\left(\delta(t) + \frac{j}{\pi t}\right) u_p(t)\right] e^{-j\omega_p t} \right\|^2 \end{aligned}$$

$$+\left\|f\left(t\right)-\sum_{p}u_p\left(t\right)\right\|_2^2$$
$$+\left[\lambda\left(t\right),f\left(t\right)-\sum_{p}u_p\left(t\right)\right] \tag{17}$$

The Alternating Direction Method of Multipliers (ADMM) is used to obtain the saddle point of the extended Lagrange. The above non-binding variational problem is solved, and the signal is decomposed into p eigenmode components [41].

When the VMD decomposes the signal, it is necessary to set the decomposed number p value in advance. However, due to the complexity of the actual signal and the high noise in general, it is difficult to determine the optimal number of decompositions when we use VMD to decompose the signal [42]–[44]. This paper proposes an adaptive VMD (AVMD) based on frequency energy conservation. It determines the optimal number of decompositions based on the energy difference between the original signal and the eigenmodes. The frequency energy formula is $E_p = \int \left|u_p\left(t\right)\right|^2 dt$. In order to ensure the integrity of the signal decomposition, the value with the smallest energy difference is taken as the optimal decomposition number p. The specific steps and frequency energy formula are shown in Fig. 4.

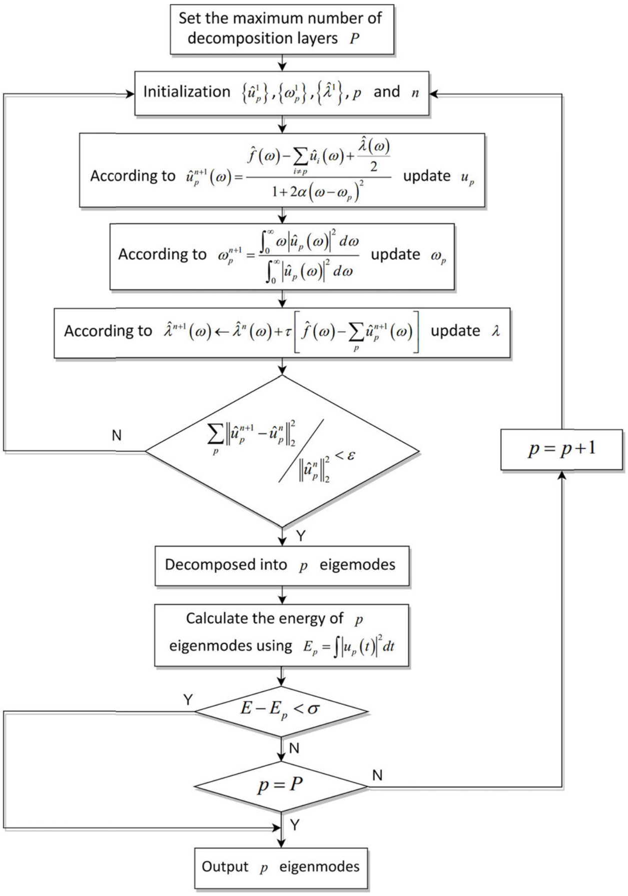

Set the maximum number of decomposition layers P
Initialization $\{\hat{u}_p^1\},\{\omega_p^1\},\{\hat{\lambda}^1\},p$ and n
According to $\hat{u}_p^{n+1}(\omega)=\frac{\hat{f}(\omega)-\sum_{i\neq p}\hat{u}_i(\omega)+\frac{\hat{\lambda}(\omega)}{2}}{1+2\alpha(\omega-\omega_p)^2}$ update u_p
According to $\omega_p^{n+1}=\frac{\int_0^\infty \omega|\hat{u}_p(\omega)|^2 d\omega}{\int_0^\infty |\hat{u}_p(\omega)|^2 d\omega}$ update ω_p
According to $\hat{\lambda}^{n+1}(\omega)\leftarrow\hat{\lambda}^n(\omega)+\tau\left[\hat{f}(\omega)-\sum_p\hat{u}_p^{n+1}(\omega)\right]$ update λ
N
$\sum_p\|\hat{u}_p^{n+1}-\hat{u}_p^n\|_2^2 / \|\hat{u}_p^n\|_2^2<\varepsilon$
$p=p+1$
Y
Decomposed into p eigemodes
Calculate the energy of p eigenmodes using $E_p=\int|u_p(t)|^2dt$
Y
$E-E_p<\sigma$
N
$p=P$
N
Y
Output p eigenmodes

FIGURE 4. The AVMD algorithm steps.

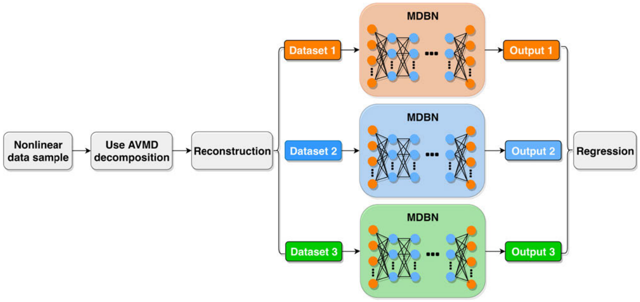


FIGURE 5. The parallel computing structure of AVMD-MDBN model.

C. AVMD-MDBN LEARNING PROCESS

The first two sections describe the AVMD and MDBN, respectively. This section presents the AVMD-MDBN parallel computing model to build a more accurate and efficient nonlinear system model. The AVMD-MDBN parallel computing model is shown in Fig. 5. The algorithm execution steps are detailed as follows:

Step 1: Decompose the original dataset using the AVMD method. The improved variational mode decomposition can divide each data item of the original dataset into several different sub-data items according to the frequency energy conservation formula, each of which contains partial information of the original data.

Step 2: Build a parallel data structure. In order to maintain the effectiveness of variational modal decomposition to reduce or eliminate uncertainties in the data sequence, we use parallel structures to perform feature extraction on the data. Each data item decomposed by AVMD has a different decomposition layer value. The decomposed sub-data items are divided into three sub-datasets according to the amplitude value, and the three sub-datasets are distributed to different feature extractors, each of which is an MDBN. The main task of this step is to complete the assignment of sub-datasets to the AVMD-MDBN parallel computing model.

Step 3: Parallel feature extraction. After feeding the sub-dataset decomposed by the AVMD to the MDBN, multiple parallel MDBNs begin data feature extraction.

Step 4: Regression prediction. A master node is added after three parallel MDBNs, the master node broadcasts the structure and parameters of a network and distributes the training datasets to all computing nodes. Each computing node reads its own local data, calculates the weight and bias variations, and transmits the calculation results to the master node. The master node synthetically processes the results transmitted from the respective computing nodes, and subsequently broadcasts the structure and updated parameters to each computing node. Such procedure is repeated until the supervised learning is to minimize the error functions, such that for any input $x(t-1)$ at any instant $t-1$, an appropriate output $y(t)$ can be predicted.

D. TIME COMPLEXITY ANALYSIS

The AVMD-MDBN model first uses the variational mode decomposition to decompose the nonlinear data. The time complexity is $O(NlogN)$, and the decomposition results in p different components.$q\,(q < p)$ subsequences are obtained by integrating the components of the same amplitude, and the integration process time complexity is a constant order $O(1)$. The time complexity of the DBN training process is $t_{dbn} = O(N * (I * b + E * D))$. Where N represents the size of the training sample, q represents the number of recombinant subsequences. I represents the number of RBM pre-trainings, b represents the number of network nodes, E represents the number of RBM supervision trainings, and D represents the dimension of the weights. Therefore, the time complexity of the AVMD-MDBN model can be expressed as:

$$T(N) = O(N \log N) + O(N * (I * b + E * D)) \tag{18}$$

IV. EXPERIMENT

In this section, we will conduct three experimental studies to demonstrate the effectiveness and superiority of the proposed AVMD-MDBN in modeling nonlinear systems, including nonlinear dynamic system modeling, Mackey-Glass time-series prediction and financial stock prediction. All simulations were performed on a hardware configuration with a 2.4GHz quad-core Intel Core i5 processor, and the programming environment was Python. Further, 4 nodes can be used in the parallel computing platform. This experiment requires 3 nodes, and one of these nodes is selected as the master node. The proposed parallel method uses 3 computing nodes and 3 data sets in the pre-training and fine-tuning stages, and each computing node corresponds to a data set. The master node also acts as a slave node and is responsible for the same computing tasks as the slave nodes, in addition to the tasks of broadcasting, synchronization, and synthesis. The performance of the proposed AVMD-MDBN is measured by Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-Squared, which are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{n}^{i=1}\left(Y_i - \hat{Y}_l\right)^2}{n}} \tag{19}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|Y_i - \hat{Y}_l\right|}{Y_i} \tag{20}$$

$$MAE = \frac{\sum_{n}^{i=1}\left|Y_i - \hat{Y}_l\right|}{n} \tag{21}$$

$$R^2 = 1 - \frac{\sum_{n}^{i=1}\left(Y_i - \hat{Y}_l\right)^2}{\sum_{n}^{i=1}\left(Y_i - \bar{Y}\right)^2} \tag{22}$$

Among them, the value range of R^2 is [0,1], and the larger R^2, the higher the accuracy of prediction.

A. NONLINEAR DYNAMIC SYSTEM MODELING

When evaluating the performance of a neural network, we typically use nonlinear dynamic system modeling. In particular, the nonlinear dynamic system described in Eq. (23) is a classical benchmark problem used to evaluate neural networks [45], [46].

$$y(t+1) = \frac{y(t)\,y(t-1)\,[y(t)+2.5]}{1+y^2(t)+y^2(t-1)} + u(t) \tag{23}$$

where $y(1) = 0, y(2) = 0, u(t) = sin(2\pi t/25)$. The prediction model is given by

$$y(t+1) = f\,[y(t-1), y(t), u(t)] \tag{24}$$

The structure of AVMD-MDBN is selected as 3-32-32-1. That is, there are 3 neurons in the input layer, 32 neurons in each of the two hidden layers and 1 neuron in the output layer. Data samples are generated using Eq. (23), with the first 1000 data points used to train the AVMD-MDBN model and the last 200 data points used to test the model. In the unsupervised pre-training phase, the number of training sessions per RBM is 50. We chose to iterate 500 times during the supervisory training phase of the AVMD-MDBN model.

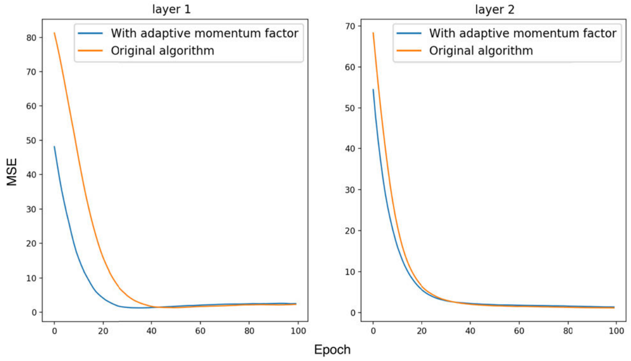


FIGURE 6. Unsupervised training error curve.

As shown in Fig. 6, the deep belief network based on the momentum adaptive factor can effectively improve the training effect and training efficiency of the pre-training process. The momentum adaptive factors ρ and ε are set to 1.4 and 0.7. The variational mode decomposition algorithm can decompose the original complex signal into p sub-signals with different characteristics. As shown in Fig. 7, the variable energy mode decomposition (AVMD) based on frequency energy adaptive can adaptively decompose the original complex signal into p sub-signals with different characteristics, so that the decomposed sub-signals can fully express the characteristics of the original signal. In the experiment of classical nonlinear system modeling, $p_{y(t-1)} = 9, p_{y(t)} = 1, p_{u(t)} = 1$.

In order to effectively prove the superiority of AVMD-MDBN modeling, 50 independent comparison experiments are conducted. The comparison models include AVMD-DBN, MDBN, DBN, CDBN, DFNN, AANN and SCNN. As can be seen from Table 1, Fig. 8 and Fig. 9, the AVMD-MDBN model has the minimum RMSE, MAE and MAPE, which is mainly due to the addition of momentum factor to the optimization of unsupervised learning, the improvement

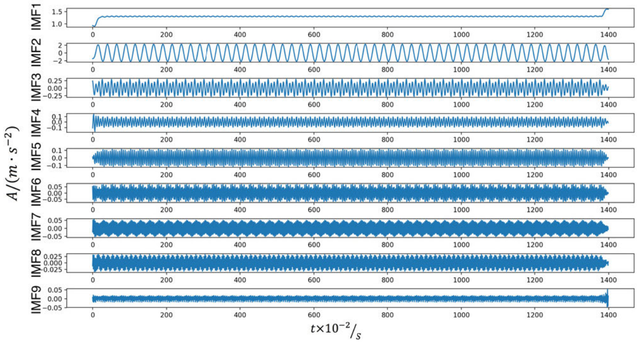


FIGURE 7. AVMD decomposition results.

TABLE 1. Comparison of experimental results of different methods.

Method	Neurons number	MAE	MAPE	RMSE	R^2
AVMD-MDBN	**11-32-32-1**	**0.000512**	**0.15%**	**0.000612**	**0.999**
AVMD-DBN[a]	11-32-32-1	0.001744	0.48%	0.002017	0.999
MDBN[a]	3-32-32-1	0.007015	2.09%	0.008853	0.999
DBN[a]	3-32-32-1	0.009497	2.85%	0.011305	0.999
CDBN[a]	3-32-32-1	0.004761	1.23%	0.005891	0.999
DFNN[b]	—	—	—	0.025962	—
AANN[b]	—	—	—	0.042426	—
SCNN[b]	—	—	—	0.003111	—

The best experimental data is shown in bold.

a

This method is a traditional DBN and variants proposed herein, where CDBN is a variant proposed in the cited literature. Experimental data was calculated by the author of this paper.

b

The experimental data is the same as the reference paper.

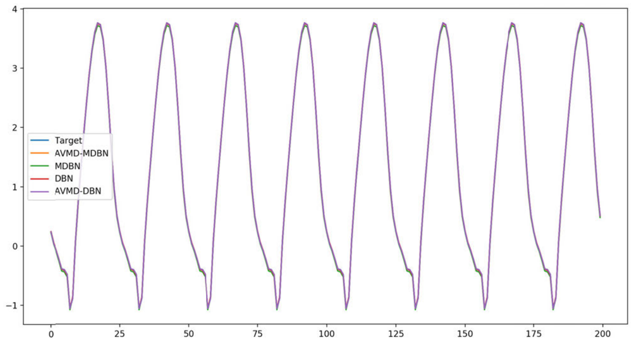


FIGURE 8. Modeling effects of different models.

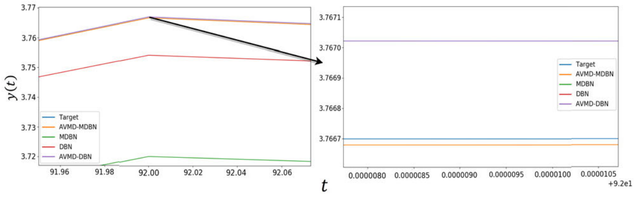


FIGURE 9. Comparison of modeling accuracy of different models.

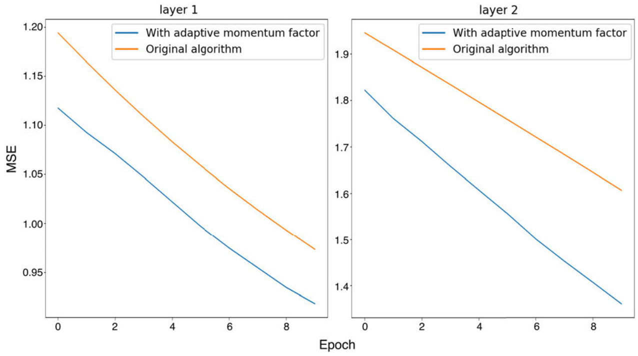


FIGURE 10. Unsupervised training error curve.

of AVMD in feature decomposition and the AVMD-MDBN parallel computing structure proposed in this paper.

It can be observed from Table 1 that the MAE of the AVMD-MDBN model is reduced by 70.6%, 92.7%, 94.6%, and 89.2%, respectively, compared with AVMD-DBN, MDBN, DBN, and CDBN. The MAPE of the AVMD-MDBN model is reduced by 68.8%, 92.8%, 94.7% and 87.8%, respectively. The RMSE of the AVMD-MDBN model decreases by 69.7%, 93.1%, 94.6%, 89.6%, 97.6%, 98.6% and 80.3%, compared with AVMD-DBN, MDBN, DBN, CDBN, DFNN, AANN and SCNN, respectively. At the same time, the proposed algorithm has the highest R-Squared and the prediction accuracy is improved. In general, the AVMD-MDBN algorithm can better find the inherent trends and laws of nonlinear systems.

B. MACKEY-GLASS TIME-SERIES PREDICTION

In this example, the proposed AVMD-MDBN is applied to predict the Mackey-Glass time series.

$$x(t+1) = (1-a)\,x(t) + \frac{bx(t-\tau)}{1+x^{10}(t-\tau)} \tag{25}$$

where $a = 0.1$, $b = 0.2$, $\tau = 17$ and the initial state is $x(0) = 1.2$. The prediction model is given by

$$x(t+1) = f[x(t), x(t-\tau), x(t-2\tau), x(t-3\tau)] \tag{26}$$

In this example, 600 patterns between $t = 1$ and 600 are chosen as the training samples, other 400 patterns between $t = 601$ and 1000 are used as testing samples. The structure of AVMD-MDBN is selected as 4-32-32-1. That is, there are 4 neurons in the input layer, 32 neurons in each of the two hidden layers and 1 neuron in the output layer. In the unsupervised pre-training phase, the number of training sessions per RBM is 50. We chose to iterate 500 times during the supervisory training phase of the AVMD-MDBN model.

As shown in Fig. 10, the deep belief network based on the momentum adaptive factor can effectively improve the training effect and training efficiency of the pre-training process.

The momentum adaptive factors ρ and ε are set to 1.3 and 0.8. The variational mode decomposition algorithm can decompose the original complex signal into p sub-signals with different characteristics. As shown in Fig. 11, the variable energy mode decomposition (AVMD) based on frequency energy adaptive can adaptively decompose the original complex signal into p sub-signals with different characteristics, so that the decomposed sub-signals can fully express the characteristics of the original signal. In the experiment of classical nonlinear system modeling, $p_{x(t)} = 1, p_{x(t-\tau)} = 1, p_{x(t-2\tau)} = 7, p_{x(t-3\tau)} = 1$.

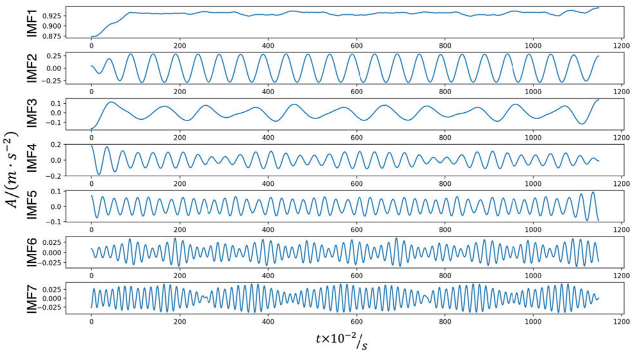


FIGURE 11. **AVMD decomposition results.**

In order to effectively prove the superiority of AVMD-MDBN modeling, 50 independent comparison experiments are conducted. The comparison models include AVMD-DBN, MDBN, DBN, CDBN, DFNN and AANN. As can be seen from Table 2, Fig. 12 and Fig. 13, the AVMD-MDBN model has the minimum RMSE, MAE and MAPE, which is mainly owing to the addition of momentum factor to the optimization of unsupervised learning, the improvement of AVMD in feature decomposition and the AVMD-MDBN parallel computing structure proposed in this paper.

TABLE 2. **Comparison of experimental results of different methods.**

Method	Neurons number	MAE	MAPE	RMSE	R^2
AVMD-MDBN	**10-32-32-1**	**0.004833**	**0.59%**	**0.006165**	**0.999**
AVMD-DBN	10-32-32-1	0.006794	0.79%	0.008440	0.998
MDBN	4-32-32-1	0.010145	1.16%	0.011528	0.997
DBN	4-32-32-1	0.017759	2.10%	0.019053	0.992
CDBN	4-32-32-1	0.019743	2.32%	0.020757	0.991
DFNN	4-40-1	0.010918	1.40%	0.012061	0.996
AANN	4-40-1	0.013871	1.58%	0.014069	0.994

It can be calculated from Table 2 that the MAE of the AVMD-MDBN model is reduced by 28.9%, 52.4%, 72.8%, 75.5%, 55.7% and 65.2%, respectively, compared with AVMD-DBN, MDBN, DBN, CDBN, DFNN and AANN. The MAPE of the AVMD-MDBN model is reduced by 25.4%, 49.4%, 72.0%, 74.7%, 58.0% and 62.8%, respectively. The RMSE of the AVMD-MDBN model decreases by 27.0%, 46.5%, 67.6%, 70.3%, 48.9% and 56.2%, respectively. At the same time, the proposed algorithm has the highest R-Squared and the prediction accuracy is improved. In general, the AVMD-MDBN algorithm can better find the inherent trends and laws of nonlinear systems.

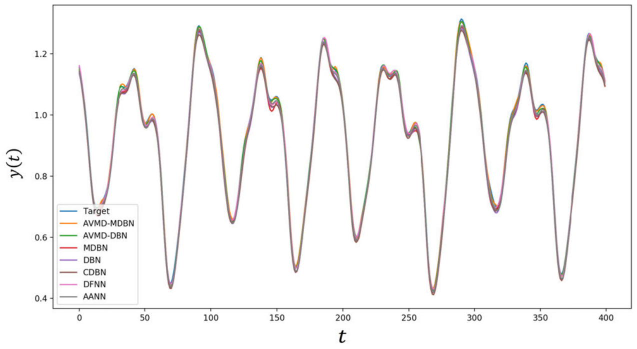


FIGURE 12. **Modeling effects of different models.**

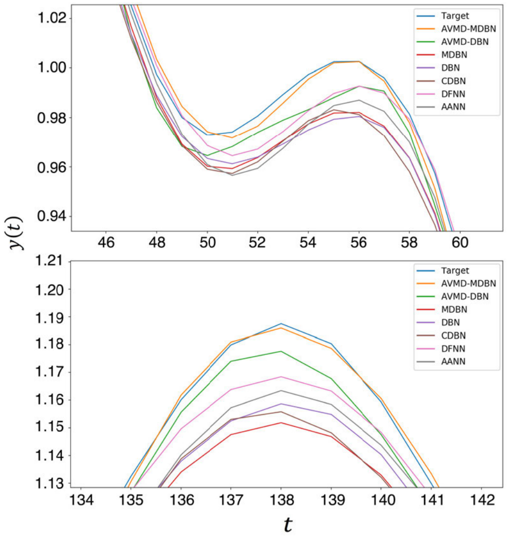


FIGURE 13. **Comparison of modeling accuracy of different models.**

C. FINANCIAL STOCK PREDICTION

There are often some complex nonlinear systems in the analysis of financial time series data. It is difficult to accurately describe these complex system state equations by analytical mathematical equation. Aiming at the complexity and uncertainty of data analysis of current financial time series, the simulation of complex nonlinear systems is transformed into the pattern recognition of financial time series data curves, thus verifying the effectiveness of the proposed modeling method.

In this section, the main purpose is to use the proposed AVMD-MDBN model to predict the closing price in stock financial data. The data used for verification in this experiment comes from the CSI 300 Index. The CSI 300 Index is an index jointly published by the Shanghai and Shenzhen

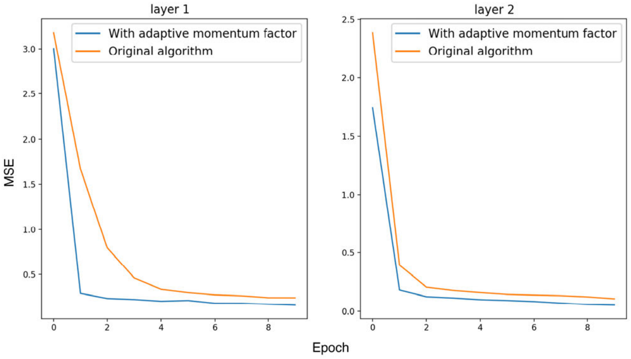


FIGURE 14. Unsupervised training error curve.

Stock Exchanges on April 8, 2005. The sample covers about 60% of the market value of the Shanghai and Shenzhen markets, and has a good market representation, so it can basically represent the high-quality stocks in the A-share market. We downloaded the data of the CSI 300 Index from the WIND database, which is a widely used financial database in mainland China. Then we extracted data from the downloaded dataset for 3,000 trading days from October 25, 2005 to February 23, 2018. The dataset includes 7 dimensions of "date, lowest price, highest price, opening price, volume, transaction amount and closing price". In particular, 2000 samples are randomly selected from 3000 samples as training samples, and the remaining 1000 samples are used as test samples. Prior to the experiment, the input samples of the AVMD-MDBN model are normalized to a closed interval (0, 1). The architecture of AVMD-MDBN is 6-64-64-1, with no supervised pre-training process, and the number of training sessions per RBM is 10. We chose to iterate 500 times during the supervisory training phase of the AVMD-MDBN model. The prediction result of AVMD-MDBN is better than DBN. Therefore, the AVMD-MDBN model can fully understand the nonlinear dynamic characteristics of the closing price in stock financial data.

As shown in Fig. 14, the deep belief network based on the momentum adaptive factor can effectively improve the training effect and training efficiency of the pre-training process. The momentum adaptive factors ρ and ε are set to 1.3 and 0.8. The variational mode decomposition algorithm can decompose the original complex signal into p sub-signals with different characteristics. As shown in Fig. 15, the variable energy mode decomposition (AVMD) based on frequency energy adaptive can adaptively decompose the original complex signal into p sub-signals with different characteristics, so that the decomposed sub-signals can fully express the characteristics of the original signal. In the experiment of classical nonlinear system modeling, $p_{Lowestprice} = 3$, $p_{Openingprice} = 3$, $p_{Highestprice} = 4$, $p_{Volume} = 4$, $p_{Transactionamount} = 1$, $p_{Data} = 1$.

In order to fully demonstrate the superiority of the proposed AVMD-MDBN model for financial data prediction, 50 independent comparison experiments are conducted. The comparison models include AVMD-DBN, MDBN, DBN, CDBN, DFNN, AANN and the corresponding results are shown in Table 3, Fig. 16 and Fig. 17. It can be seen from Table 3 that the AVMD-MDBN model has the best performance in the four evaluation indexes of RMSE, MAE, MAPE and R-Squared. Table 3, Fig. 16 and Fig. 17 detail the superiority of the AVMD-MDBN model and can be used to predict the closing price in stock financial data.

It can be observed from Fig. 16 and Fig. 17 that the difference between the prediction result of the AVMD-MDBN method and the actual result is small, and the accuracy between the prediction result and the actual result on the horizontal time axis is higher than other methods, that is, the proposed method improves the lag of stock forecasting.

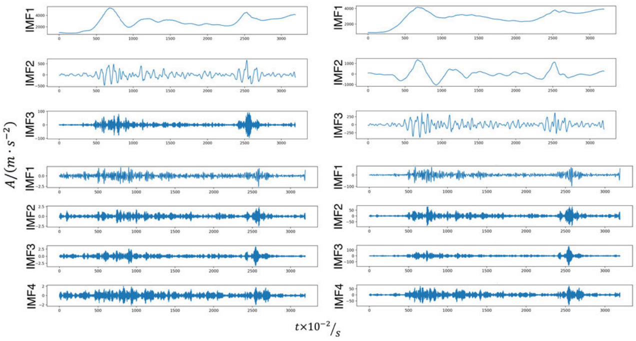


FIGURE 15. AVMD decomposition results.

TABLE 3. Comparison of experimental results of different methods.

Method	Neurons number	MAE	MAPE	RMSE	R^2
AVMD-MDBN	**16-64-64-1**	**27.06485**	**0.80%**	**42.94552**	**0.996**
AVMD-DBN	16-64-64-1	31.97048	0.91%	60.78569	0.992
MDBN	6-64-64-1	38.6817	1.15%	62.97621	0.992
DBN	6-64-64-1	41.16745	1.21%	69.30449	0.990
CDBN	6-64-64-1	44.5652	1.29%	75.21432	0.988
DFNN	6-80-1	30.21455	0.90%	59.75362	0.994
AANN	6-80-1	32.47421	0.93%	61.69421	0.993

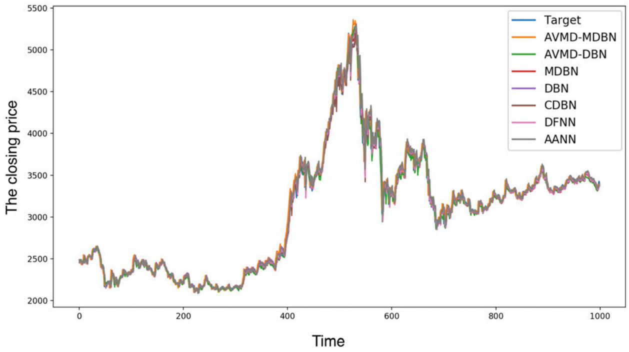


FIGURE 16. Modeling effects of different models.

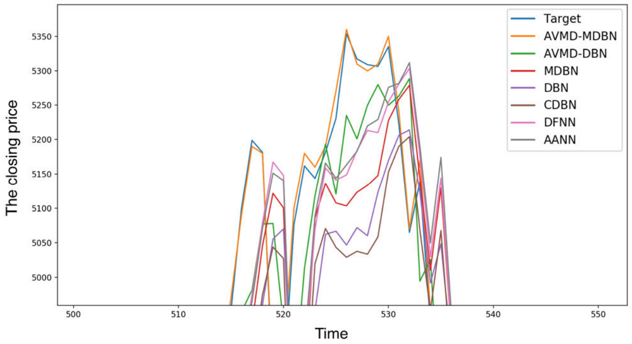


FIGURE 17. Comparison of modeling accuracy of different models.

It can be observed from Table 3 that the MAE of the AVMD-MDBN model is reduced by 15.3%, 30.0%, 34.3%, 39.3%, 10.4% and 16.7%, respectively, compared with AVMD-DBN, MDBN, DBN, CDBN, DFNN and AANN. The MAPE of the AVMD-MDBN model was reduced by 12.1%, 30.4%, 33.9%, 38.0%, 11.1% and 14.0%, respectively. The RMSE of the AVMD-MDBN model is reduced by 29.3%, 31.8%, 38.0%, 43.0%, 28.1% and 30.4% compared with other algorithms. The algorithm proposed in this paper has the highest R-Squared, and the prediction accuracy is greatly improved. In general, the AVMD-MDBN algorithm can better identify the inherent trends and patterns of data.

In addition, the experimental results of the comparative modeling methods listed in Table 1, Table 2 and Table 3 are based on the optimal parameters and structure of the respective methods (the optimal parameters of the modeling method are generated by trial and error). It should be noted that in order to make a fairer comparison, the proposed modeling method and the comparative modeling method must use the best comparison model.

V. CONCLUSION

This paper proposes a deep neural network model based on improved VMD and improved DBN (AVMD-MDBN) parallel structure. This model is suitable for nonlinear system modeling. The proposed AVMD-MDBN introduces the adaptive momentum factor into the unsupervised learning process, which effectively improves the modeling speed and accuracy to a certain extent. In order to improve the modeling accuracy, the nonlinear data based on frequency energy conservation is adaptively decomposed into multiple subsequences with different components. Each subsequence is then distributed into different MDBNs with adaptive momentum factors for efficient feature extraction. Finally, the characteristics of different DBN outputs are deeply integrated, and the nonlinear system model is constructed. The experimental results show that the proposed AVMD-MDBN parallel computing model has excellent nonlinear system modeling performance. At the same time, it is confirmed that selecting a suitable AVMD-MDBN structure is very important, especially when processing large samples. However, in the existing literature, little research has been done on the method of designing a suitable DBN architecture. For future work, our goal is to design a self-organizing architecture for AVMD-MDBN to improve the efficiency of deep neural networks from a compact architectural perspective.

REFERENCES

[1] J. Doyle, "Robust and optimal control," in *Proc. 35th IEEE Conf. Decis. Control*, vol. 2, Dec. 2002, pp. 1595–1598.

[2] A. Isidori, *Nonlinear Control Systems II* (Lecture Notes in Control & Information Sciences), London, U.K.: Springer, 1999. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4471-0549-7.

[3] H. Peng, Z.-J. Yang, W. Gui, M. Wu, H. Shioya, and K. Nakano, "Nonlinear system modeling and robust predictive control based on RBF-ARX model," *Eng. Appl. Artif. Intell.*, vol. 20, no. 1, pp. 1–9, Feb. 2007.

[4] S. Anderson and V. Kadirkamanathan, "Modelling and identification of non-linear deterministic systems in the delta-domain," *Automatica*, vol. 43, no. 11, pp. 1859–1868, Nov. 2007.

[5] F. Leung, H. Lam, S. Ling, and P. Tam, "Tuning of the structure and parameters of a neural network using an improved genetic algorithm," *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 79–88, Jan. 2003.

[6] M. Paliwal and U. A. Kumar, "Neural networks and statistical techniques: A review of applications," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 2–17, Jan. 2009.

[7] M. J. Er and S. Wu, "A fast learning algorithm for parsimonious fuzzy neural systems," *Fuzzy Sets Syst.*, vol. 126, no. 3, pp. 337–351, Mar. 2002.

[8] F. Li, J. Qiao, H. Han, and C. Yang, "A self-organizing cascade neural network with random weights for nonlinear system modeling," *Appl. Soft Comput.*, vol. 42, pp. 184–193, May 2016.

[9] H.-G. Han, L.-D. Wang, and J.-F. Qiao, "Efficient self-organizing multilayer neural network for nonlinear system modeling," *Neural Netw.*, vol. 43, pp. 22–32, Jul. 2013.

[10] D. Zhang, G. Zang, J. Li, K. Ma, and H. Liu, "Prediction of soybean price in China using QR-RBF neural network model," *Comput. Electron. Agricult.*, vol. 154, pp. 10–17, Nov. 2018.

[11] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[12] Y. Bengio and J.-S. Senecal, "Adaptive importance sampling to accelerate training of a neural probabilistic language model," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 713–722, Apr. 2008.

[13] U. Yolcu, E. Egrioglu, and C. H. Aladag, "A new linear & nonlinear artificial neural network model for time series forecasting," *Decis. Support Syst.*, vol. 54, no. 3, pp. 1340–1347, Feb. 2013.

[14] J. Qiao, G. Wang, X. Li, and W. Li, "A self-organizing deep belief network for nonlinear system modeling," *Appl. Soft Comput.*, vol. 65, pp. 170–183, Apr. 2018.

[15] M. Qin, Z. Li, and Z. Du, "Red tide time series forecasting by combining ARIMA and deep belief network," *Knowl.-Based Syst.*, vol. 125, pp. 39–52, Jun. 2017.

[16] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Feb. 2010.

[17] L. Zhao, Y. Zhou, H. Lu, and H. Fujita, "Parallel computing method of deep belief networks and its application to traffic flow prediction," *Knowl.-Based Syst.*, vol. 163, pp. 972–987, Jan. 2019.

[18] S. Ahn, J. Kim, E. Lim, and S. Kang, "Soft memory box: A virtual shared memory framework for fast deep neural network training in distributed high performance computing," *IEEE Access*, vol. 6, pp. 26493–26504, 2018.

[19] W. Jiang, H. Tang, L. Wu, H. Huang, and H. Qi, "Parallel processing of probabilistic models-based power supply unit mid-term load forecasting with apache spark," *IEEE Access*, vol. 7, pp. 7588–7598, 2019.

[20] F. Zhou, S. Yang, H. Fujita, D. Chen, and C. Wen, "Deep learning fault diagnosis method based on global optimization GAN for unbalanced data," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104837.

[21] P. Jiang and M. X. Jiao, "A hybrid forecasting approach applied in the electrical power system based on data preprocessing, optimization and artificial intelligence algorithms," *Appl. Math. Model.*, vol. 40, no. 23, pp. 10631–10649, Dec. 2016.

[22] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[23] Y. Bengio, "Learning deep architectures for AI," *FNT Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[24] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[25] G. Hinton, "A practical guide to training restricted Boltzmann machines," *Momentum*, vol. 9, no. 1, pp. 926–947, Jan. 2010.

[26] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[27] S. Tang, C. Shen, D. Wang, S. Li, W. Huang, and Z. Zhu, "Adaptive deep feature learning network with Nesterov momentum and its application to rotating machinery fault diagnosis," *Neurocomputing*, vol. 305, pp. 1–14, Aug. 2018.

[28] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, Dec. 2013.

[29] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," *Neurocomputing*, vol. 137, pp. 47–56, Aug. 2014.

[30] N. Liu and J.-M. Kan, "Improved deep belief networks and multi-feature fusion for leaf identification," *Neurocomputing*, vol. 216, pp. 460–467, Dec. 2016.

[31] W. Zhang, H. Zou, L. Luo, Q. Liu, W. Wu, and W. Xiao, "Predicting potential side effects of drugs by recommender methods and ensemble learning," *Neurocomputing*, vol. 173, pp. 979–987, Jan. 2016.

[32] Y. Chen, "Mineral potential mapping with a restricted Boltzmann machine," *Ore Geol. Rev.*, vol. 71, pp. 749–760, Dec. 2015.

[33] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[34] H. Han, X.-L. Wu, and J.-F. Qiao, "Nonlinear systems modeling based on self-organizing fuzzy-neural-network with adaptive computation algorithm," *IEEE Trans. Cybern.*, vol. 44, no. 4, pp. 554–564, Apr. 2014.

[35] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[36] H. Ren, Y. Chai, J. Qu, X. Ye, and Q. Tang, "A novel adaptive fault detection methodology for complex system using deep belief networks and multiple models: A case study on cryogenic propellant loading system," *Neurocomputing*, vol. 275, pp. 2111–2125, Jan. 2018.

[37] R. Salakhutdinov, "Learning deep Boltzmann machines using adaptive MCMC," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2010, pp. 943–950.

[38] L.-W. Kim, "DeepX: Deep learning accelerator for restricted Boltzmann machine artificial neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1441–1453, May 2018.

[39] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 531–544, Feb. 2014.

[40] G. Choi, H.-S. Oh, and D. Kim, "Enhancement of variational mode decomposition with missing values," *Signal Process.*, vol. 142, pp. 75–86, Jan. 2018.

[41] Y. Liu, G. Yang, M. Li, and H. Yin, "Variational mode decomposition denoising combined the detrended fluctuation analysis," *Signal Process.*, vol. 125, pp. 349–364, Aug. 2016.

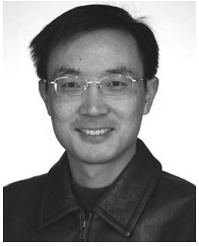
[42] Z. Li, Y. Jiang, Q. Guo, C. Hu, and Z. Peng, "Multi-dimensional variational mode decomposition for bearing-crack detection in wind turbines with large driving-speed variations," *Renew. Energy*, vol. 116, pp. 55–73, Feb. 2018.

[43] S. Liu, G. Tang, X. Wang, and Y. He, "Time-frequency analysis based on improved variational mode decomposition and Teager energy operator for rotor system fault diagnosis," *Math. Problems Eng.*, vol. 2016, pp. 1–9, Nov. 2016.

[44] K. He, Y. Chen, and G. K. Tso, "Forecasting exchange rate using Variational Mode Decomposition and entropy theory," *Phys. A, Stat. Mech. Appl.*, vol. 510, pp. 15–25, Nov. 2018.

[45] J. Qiao, G. Pan, and H. Han, "Design and application of continuous deep belief network," *Acta Automatica Sinica*, vol. 41, no. 12, pp. 2138–2146, Dec. 2015.

[46] J. Qiao, G. Wang, W. Li, and X. Li, "A deep belief network with PLSR for nonlinear system modeling," *Neural Netw.*, vol. 104, pp. 68–79, Aug. 2018.



QIBING JIN was born in Yichang, Hubei, China, in 1971. He received the Ph.D. degree in control theory and engineering from Northeastern University, Shenyang, Liaoning, China, in 1999. He joined the Beijing University of Chemical Technology, Beijing, China, in 2002. He is currently a Full Professor with the College of Information Science and Technology, the Director of the Institute of Automation with the Beijing University of Chemical Technology, Beijing, and the Executive Director of the China National Association for Automation in Petroleum and Chemical Industry. In recent years, he was awarded several Prizes for Science and Technology Progress. His main research interests include advanced control, intelligent instrument, system identification, and control theory. He has rich experience in control engineering, and his many research results have been applied in petroleum and chemical industry.

HENGYU ZHANG received the B.Sc. degree in communication engineering from the Beijing University of Chemical Technology, Beijing, China, in 2016, where he is currently pursuing the master's degree in control science and engineering.

His main research interests are process control, including improvement of deep neural networks, the classical model-based controller tuning, the data-driven performance optimization, and the control design and modeling for some specific chemical processes.

YUMING ZHANG received the B.Sc. degree in automation from the Tianjin University of Commerce, Tianjin, China, in 2011. In 2013, he entered the Beijing University of Chemical Technology, where he is currently pursuing the Ph.D. degree.

His main research interests are modern control and system identification, including active disturbance rejection control, disturbance observer, model predictive control, and intelligent modeling.

WU CAI received the B.Sc. degree in automation from the Beijing University of Chemical Technology, Beijing, China, in 2014, where she is currently pursuing the Ph.D. degree in control science and engineering.

Her main research interests are analysis and design of disturbance rejection of multiple-input-multiple-output (MIMO) systems; specifically, in the partially decentralized controller design and disturbance observer-based (DOB) design.

MEIXUAN CHI received the B.Sc. degree in computer science and technology from the Beijing University of Chemical Technology, Beijing, China, in 2016, where she is currently pursuing the master's degree in computer science and technology.

Her main research interests are intelligent optimization and machine learning, including the improvement of Group Intelligence Optimization Algorithm, and Classification of High-Dimensional Data Sets Based on Group Intelligence Optimization Algorithm.