



A hybrid modelling method for time series forecasting based on a linear regression model and deep learning

Wenquan Xu^{1,2} · Hui Peng¹ · Xiaoyong Zeng¹ · Feng Zhou³ · Xiaoying Tian¹ · Xiaoyan Peng⁴

Published online: 20 February 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Time series forecasting has important theoretical significance and engineering application value. A number of studies have shown that hybrid modelling is very successful in various modelling applications, and both theoretical and empirical findings have shown that hybrid modelling is an effective method to improve the accuracy of time series models. This paper proposes a hybrid model that combines a linear regression (LR) model and deep belief network (DBN) model for the prediction of time series data. In the hybrid model, the linear AR (auto-regression) LR model or ARIMA (auto-regressive integrated moving average) model and the nonlinear DBN model are explored to capture the linear and nonlinear behaviours of a time series, respectively. We first use an LR model to fit the original data and obtain the LR model residuals between the original data and the predicted data of the LR model. Then, the residuals are regarded as the nonlinear component and are used as inputs into the DBN model. The LR model prediction and the output of the DBN model are the final forecasting value for the time series, which takes full advantage of the two models for predicting time series. The proposed hybrid model and other existing models are applied to four well-known time series for comparison, and the results show that the proposed hybrid model has a high prediction accuracy and may be a useful tool for time series forecasting.

Keywords Hybrid model · Time series forecasting · Linear regression · Deep learning

1 Introduction

Time series forecasting has been a very popular research topic because it has wide applications in many different areas. For example, it is very convenient for people to travel when they are provided a traffic forecasting service. Forecasting climate change provides great help in transportation and agricultural production departments. Forecasting financial data is very helpful for business investments. In time series modelling, some data can be characterized by linear models, and some data must be characterized by nonlinear models. Systems with linear and/or nonlinear behaviours can be modelled using

different methods. For time series data, various prediction models have been proposed in the literature [1], in which the typical linear regression (LR) model is either a linear auto-regression (AR) model or linear auto-regressive integrated moving average (ARIMA) model, and a promising nonlinear model is the deep belief network (DBN) model [2–4]. These three models are used in this paper.

The LR model is widely used in different fields, such as forecasting short-term electricity demands [5], stock indexes [6], and wind speeds [7]. In this paper, the linear AR and ARIMA models are used as predictive models for the linear behaviours of a time series. For nonlinear time series, however, LR modelling may not be a good choice. The artificial neural network (ANN) is an effective nonlinear modelling technique for predicting nonlinear time series [1]. It has been confirmed that ANNs could achieve a desired accuracy in nonlinear time series prediction [8–11]. Over the past several decades, ANN modelling techniques have developed substantially. There were more than 5000 publications using ANNs for time series forecasting by 2007 [12]. Some famous ANN structures are listed as follows: the radial basis function (RBF) NN, error back-propagation (BP) NN, and so on. With a great deal of research on ANNs, some problems have also arisen. The main problem is difficulty in determining the values of

✉ Hui Peng
huipeng@mail.csu.edu.cn

¹ School of Information Science and Engineering, Central South University, Changsha 410083, Hunan, China

² School of Physics and Electronic Engineering, Anqing Normal University, Anqing 246001, Anhui, China

³ College of Electronic Information and Electrical Engineering, Changsha University, Changsha 410003, Hunan, China

⁴ College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, Hunan, China

node connection weights and other parameters in an ANN; if one cannot obtain suitable values, the modelling accuracy decreases. Another problem is that the model parameter search in an optimization process is easy to stop at local optima. For these problems, Hinton and Osindero proposed the deep learning algorithm (DLA) [2], which first uses unsupervised learning to train the connection weights of a deep neural network. The DLA is a good method to extract intricate structures in high-dimensional data, and it has been successfully applied in many fields, such as image compression [13], forecasting exchange rates [14], the evaluation of vehicle interior sound quality [15], electricity load forecasting [3], breast cancer classification [16], magnetic resonance imaging [10], and time series modelling [3, 4, 12]. Currently, the DLA is one of the most effective algorithms for time series forecasting [2, 12] and is also commonly used for nonlinear behaviour modelling [17, 18]. The DLA may avoid falling into the local optimum and prevent over-fitting; these problems are often encountered in ANN training. The long short term memory (LSTM) model [19] is also widely applied in time series forecasting, which is a special type of recurrent neural network (RNN) architecture [19, 20], and RNNs a type of deep learning model. It has been shown that LSTM outperforms traditional RNNs in many temporal processing tasks [19].

In this paper, we use the DLA to train a DBN, which is used for the nonlinear behaviour prediction of a time series. The DBN has been proven to be very effective in learning representative features from observation data without prior knowledge. The DBN is a stack of restricted Boltzmann machines (RBMs), where the RBM is a basic and powerful neural network in which the connection between each neuron is a bipartite graph [21]. As mentioned in the literature [2, 13], the RBM is widely used in classic machine learning tasks, such as image or voice recognition, and it has shown impressive performance [22]. The RBM is a network of neurons composed of two layers: a hidden layer and a visible layer. The visible layer corresponds to the components of an observation, and the hidden layer is used to extract features from the visible layer [23]. The feature of the RBM is that the same layer variables are not connected to each other. The connection between the hidden layer and the visible layer is bidirectional and symmetric. Therefore, an RBM forms a Markov random field [21]. Hinton and Osindero proposed a fast, greedy learning method for the DBN, which learns one layer at a time [2]. After unsupervised training, a regression layer can be added at the top of the network for supervised training, and labelled data are then used for supervised fine-tuning to adjust the features for better prediction. Compared with the traditional ANN, if the model has fewer neurons, the traditional ANN has better advantages than the DBN. However, the DBN is much better than the traditional ANN when the number of nodes is very large [3].

Although the single model mentioned above may obtain good prediction results in many cases, a hybrid model, such as one combining a linear model and nonlinear model, may give better modelling results than a single model. This is because a hybrid model may absorb the qualities of the two models. Thus, using hybrid models has become a common practice to overcome the limitations of single models and improve prediction accuracy [24]. In the literature [1, 7, 8, 11, 24–29], some hybrid models have been proposed to combine the advantages of two or more individual models. For instance, a hybrid ARIMA-ANN model [25] was proposed, which combines a linear ARIMA model and a nonlinear ANN model for predicting sunspot time series, Lynx and exchange rate data, and it was shown that the prediction accuracy of the hybrid model was higher than that of a single model. Babu and Reddy also proposed a hybrid ARIMA-ANN model [1]; they used kurtosis to distinguish linear and nonlinear parts, and the proposed hybrid model achieved a higher prediction accuracy for sunspot data and electricity price data. Akouemo and Povinelli combined auto-regression with exogenous (ARX) processes and the ANN to identify anomalous data points, and the mean absolute percentage errors decreased [8]. Zhu and Wei [28] proposed a hybrid model using the ARIMA model and least squares support vector machine for carbon price forecasting, and the forecasting accuracy of the model was also better than that of a single model. Nourani et al. [29] used hybrid wavelet-artificial intelligence models in hydrology. Shukur and Lee [30] proposed a hybrid Kalman filter and ANN model to improve the accuracy of daily wind speed forecasting. Barak and Sadegh [31] proposed an ARIMA-ANFIS (adaptive network fuzzy inference systems) hybrid algorithm to forecast energy consumption. Qiu et al. [32] represent the empirical mode decomposition (EMD) and DBN-based hybrid model, called EMD-DBN, normally outperforms the corresponding single structure models for time series forecasting, and nine benchmark methods were compared to verify the effectiveness of the EMD-DBN method (i.e., the persistence method [32], ensemble DBN (EDBN) [33], support vector regression (SVR) [34], ANN [35], DBN [13], random forest (RF) [36], EMD-SVR [37], EMD-ANN [38] and EMD-RF [32]).

Based on the aforementioned studies [1, 9, 23, 25], in this paper, we present a hybrid modelling approach that combines a linear AR or ARIMA model with a DBN model for nonlinear time series prediction because the best LR model may be different for different types of data. We call the proposed models “AR-DBN” and “ARIMA-DBN” and conduct thorough experimental studies. As seen in [3, 12, 14], it was shown that compared with the Kalman smoothing model, ARIMA model, multi-layer perceptron (MLP), self-organizing fuzzy neural network (SOFNN), error BP, ARMA and feed forward neural network (FFNN) model, the DBN model is applicable to the prediction of time series and

works better than other traditional methods or models. Therefore, the DBN is chosen in this hybrid modelling method for time series prediction in this paper. This new hybrid modelling method is used to overcome the limitation of a single model, as mentioned above, for achieving more accurate prediction results. For the selection of the LR model in our hybrid model, according to the prediction results of the AR-DBN and ARIMA-DBN models, the model with the minimum MSE (mean square error) is used as the final prediction model. To use the proposed hybrid modelling method, we first use a linear AR or ARIMA model to fit a time series, and the residuals of the AR or ARIMA models are then the nonlinear component of the time series. Next, a DBN model is used to model the nonlinear part. We use a fast, greedy learning method to train the DBN first in every layer [2]; then, according to the target values, the BP algorithm is used for fine-tuning all of the connection weights. By combining the advantages of the LR model and DBN, the hybrid model not only can model time series but can also extract different features of time series. In addition, the AR-DBN and ARIMA-DBN models are trained in a greedy manner, which permits the training of deep layer networks and alleviates trapping into local minimums. Therefore, benefiting from DBN, the proposed hybrid model has desirable stability and learning ability. The proposed hybrid model approach is applied to the prediction of four time series, and the results show that the prediction accuracy of the proposed hybrid model is better than that of some models for the studied time series.

The rest of the paper is organized as follows. The proposed hybrid modelling method is presented in Section 2. The forecasting evaluation criteria used and the results and analysis of the experiments are described in Section 3. Finally, we conclude this paper in Section 4.

2 Hybrid LR-DBN model

A novel hybrid LR-DBN modelling approach to time series prediction is proposed in this section, which first uses linear AR and ARIMA models to fit a time series and then uses two DBN models to fit the two residual series of the linear AR and ARIMA models. The final selected model (i.e., AR-DBN or ARIMA-DBN) is determined by the statistical properties of the final modelling residuals from the two LR-DBN models.

2.1 Linear AR modelling

In the proposed hybrid LR-DBN modelling approach, we first use a linear AR model to represent the linear behaviour of a time series. A linear AR(p) model of order p is a linear function of the relation between the present value of a variable and its past p observations. Given a time series $\{y(t) \in R, t = 1, 2, 3, \dots, N\}$, a linear AR(p) model is defined as follows:

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-p)) + e(t) = \alpha_0 + \sum_{i=1}^p \alpha_i y(t-i) + e(t) \tag{1}$$

where N represents the amount of data, $f(\bullet)$ represents linear AR mapping, $\alpha_i (i = 0, 1, \dots, p)$ represent the linear regressive coefficients of model (1), $e(t)$ represents the modelling error, and p represents the order of the model. Model (1) can be used for one-step- or multi-step-ahead prediction, and $\hat{y}(t) = f(\bullet)$ represents the one-step-ahead forecast result.

For model (1), we use the least squares method to estimate the AR coefficients $\alpha_i (i = 0, 1, \dots, p)$ by making the square of the error reach a minimum, and the obtained AR coefficients are given by

$$\alpha = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \tag{2}$$

where.

$$\mathbf{X} = \begin{bmatrix} 1 & y(p) & y(p-1) & \dots & y(1) \\ 1 & y(p+1) & y(p) & \dots & y(2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & y(N-1) & y(N-2) & \dots & y(N-p) \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} y(p+1) \\ y(p+2) \\ \vdots \\ y(N) \end{bmatrix}, \alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix},$$

and N represents the length of the time series.

2.2 Linear ARIMA modelling

The linear AR model is easily estimated and suitable for use in the proposed hybrid modelling method. However, if the time series to be modelled is nonstationary, the ARIMA model may be better for extracting the linear part from the data in the proposed LR-DBN modelling method. When using the ARIMA model, the given original data are first checked for stationarity. If they are not stationary, a differencing operation needs to be carried out. If the processed data are still nonstationary, the differencing operation is again carried out until the data are made stationary [1]. If differencing is carried out d times, the integration order of the ARIMA model is defined as d , and the resultant data are then fitted by an auto-regressive moving average (ARMA) model as follows:

$$\tilde{y}(t) = \beta_0 + \varphi_1 \tilde{y}(t-1) + \varphi_2 \tilde{y}(t-2) + \dots + \varphi_p \tilde{y}(t-p) + \xi(t) + \beta_1 \xi(t-1) + \beta_2 \xi(t-2) + \dots + \beta_\lambda \xi(t-\lambda) \tag{3}$$

where $\tilde{y}(t) = (1-z^{-1})^d y(t)$, $z^{-1}y(t) = y(t-1)$, $\xi(t)$ represents the modelling error, $\varphi_i (i = 1, 2, \dots, p)$ and $\beta_j (j = 0, 1, \dots, \lambda)$ represent the model parameters, and p and λ indicate the

orders of the model. For the estimation method of the ARIMA model (3), one can refer to references [1, 9, 11, 24, 25, 27]. Model (3) can be used to compute the one-step-ahead forecasting, $\hat{y}(t)$, as follows: first, from (3), we obtain the prediction $\bar{y}(t) = \beta_0 + \varphi_1\bar{y}(t-1) + \varphi_2\bar{y}(t-2) + \dots + \varphi_p\bar{y}(t-p) + \beta_1\xi(t-1) + \beta_2\xi(t-2) + \dots + \beta_\lambda\xi(t-\lambda)$ and then compute $\hat{y}(t) = \bar{y}(t) - (1-z^{-1})^d y(t) + y(t)$.

2.3 Hybrid modelling

Many time series data contain both linear and nonlinear characteristics. This subsection presents a hybrid LR-DBN modelling approach that combines the linear AR model (1) or ARIMA model (3) with a DBN model for this type of time series modelling. Time series $\{y(t) \in R, t = 1, 2, 3, \dots, N\}$ is decomposed into a linear component and a nonlinear component as follows:

$$y(t) = y_L(t) + y_N(t) \tag{4}$$

where $y_L(t)$ represents the linear component and $y_N(t)$ represents the nonlinear component. We first use the AR model (1) or ARIMA model (3) to fit the time series, and the residual $e(t)$ is given by

$$e(t) = y(t) - \hat{y}_L(t) \tag{5}$$

where $\hat{y}_L(t)$ represents the predicted value using the AR model (1) or ARIMA model (3) at time t and $e(t)$ only contains the nonlinear element. Next, we design a DBN model to fit the nonlinear component $e(t)$ as follows:

$$e(t) = g(e(t-1), e(t-2), \dots, e(t-q)) + \varepsilon(t) \tag{6}$$

where $g(\bullet)$ represents a nonlinear function approximated by the designed DBN, q represents the order of the model, $\varepsilon(t)$ represents the final modelling error, and the prediction of $e(t)$ is denoted as $\hat{e}(t) = g(e(t-1), e(t-2), \dots, e(t-q))$.

In this paper, we use the DLA to train the DBN model to calculate the predictive value $\hat{e}(t)$. The DBN model is composed of several RBMs, and the structure of the designed hybrid model is shown in Fig. 1, where N_r represents the total number of hidden layers, $\mathbf{h}^{(k)} (k = 0, 1, \dots, N_r)$ represent the output values in the k -th hidden layer, and $\mathbf{v}^{(k)} (k = 0, 1, \dots, N_r)$ represent the input values in the k -th visible layer. The LR modelling residuals are used as input data for the first RBM of the DBN. When a unit in the visible layer or hidden layer in the DBN is activated, the probabilities of the layer are described by [4]:

$$p(h_j = 1|\mathbf{v}) = \varphi\left(b_j + \sum_{i=1}^m v_i w_{ij}\right) \tag{7}$$

$$p(v_i = 1|\mathbf{h}) = \varphi\left(a_i + \sum_{j=1}^n h_j w_{ij}\right) \tag{8}$$

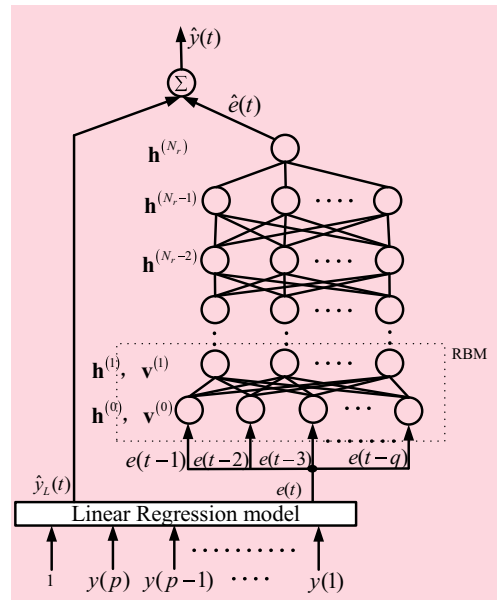


Fig. 1 Structure of the LR-DBN model

where $\varphi(x) = 1/(1 + e^{-x})$ is the sigmoid function that is obtained from the probability distribution of the visible layer or hidden layer, with a varying range of $[0, 1]$; w_{ij} represents the bi-directional weight between the visible unit i and hidden unit j ; m represents the number of neurons in the visible layer; $\mathbf{v} = (v_1, v_2, \dots, v_m)^T$ is the input vector; n represents the number of neurons in the hidden layer; $\mathbf{h} = (h_1, h_2, \dots, h_n)^T$ is the output vector; and a_i and b_j represent the bias in the input variables and hidden variables, respectively.

The RBM in Fig. 1 is an energy-based model, whose energy function is defined as follows [13, 39].

$$E(\mathbf{v}, \mathbf{h}) = \sum_{i=1}^m \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{i=1}^m \sum_{j=1}^n \frac{v_i h_j w_{ij}}{\sigma_i} - \sum_{j=1}^n b_j h_j \tag{9}$$

where σ_i represents the variance in the input variable v_i . The marginal probability distribution over the visible vector is defined by [13, 39]:

$$p(\mathbf{v}) = \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{\int_{\mathbf{h}} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h})) d\mathbf{v}} \tag{10}$$

According to the energy function shown in (9), we can define the probabilities for the visible and hidden units as follows:

$$P(v_i|\mathbf{h}) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{\left(x - a_i - \sigma_i \sum_{j=1}^n h_j w_{ij}\right)^2}{2\sigma_i^2}\right) \tag{11}$$

$$P(h_j|\mathbf{v}) = \varphi\left(b_j + \sum_{i=1}^m \frac{v_i}{\sigma_i} w_{ij}\right) \tag{12}$$

This model is suitable for continuous data, but the regular binomial-Bernoulli RBM can also be used if the data are normalized to $[0, 1]$. This method is used in our paper.

The RBMs are used as building blocks in the DBN. To minimize the deviation between the actual value and prediction value, we use the contrastive divergence (CD) algorithm, which is a good stochastic approximation approach, and its performance is better than that of some other algorithms [19]. According to the log-likelihood gradient, $\log p(\mathbf{v})$, which is obtained from (9) and (10), we use one step CD update rules to update the weight w_{ij} and bias a_i and b_j , which can be updated by.

$$\Delta w_{ij} = \varepsilon_1 \frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \varepsilon_1 \left(\frac{p(h_j = 1|\mathbf{v})v_i}{\sigma_i^2} - \frac{\sum_{i=1}^m p(\mathbf{v})p(h_j = 1|\mathbf{v})v_i}{\sigma_i^2} \right) \quad (13)$$

$$= \varepsilon_1 \left(\left\langle \frac{v_i h_j}{\sigma_i^2} \right\rangle_{data} - \left\langle \frac{v_i h_j}{\sigma_i^2} \right\rangle_{recon} \right)$$

$$\Delta a_i = \varepsilon_1 \frac{\partial \log p(\mathbf{v})}{\partial a_i} = \varepsilon_1 \left(\frac{v_i}{\sigma_i^2} - \frac{\sum_{i=1}^m p(\mathbf{v})v_i}{\sigma_i^2} \right) \quad (14)$$

$$= \varepsilon_1 \left(\left\langle \frac{v_i}{\sigma_i^2} \right\rangle_{data} - \left\langle \frac{v_i}{\sigma_i^2} \right\rangle_{recon} \right)$$

$$\Delta b_j = \varepsilon_1 \frac{\partial \log p(\mathbf{v})}{\partial b_j} = \varepsilon_1 \left(p(h_j = 1|\mathbf{v}) - \sum_{i=1}^m p(\mathbf{v})p(h_j = 1|\mathbf{v}) \right) \quad (15)$$

$$= \varepsilon_1 \left(\langle h_j \rangle_{data} - \langle h_j \rangle_{recon} \right)$$

where ε_1 represents the learning rate, $\langle \bullet \rangle_{data}$ denotes an expectation with respect to the data distribution and $\langle \bullet \rangle_{recon}$ denotes the reconstructed state.

The output value of the first RBM is used as the input data of the second RBM, and the next RBM is trained in the same manner. The output value of the DBN model is the prediction $\hat{e}(t)$. Next, using the difference between the actual output value $e(t)$ and predicted output value $\hat{e}(t)$, the BP algorithm is executed to fine tune the parameters of each RBM again.

The pseudo-codes for training the first RBM and fine-tuning the DBN are presented in algorithms 1 and 2, respectively. Let $\mathbf{h}^{(s)}$ ($s = 1, \dots, N_r$) represent the output of the s -th hidden layer, where N_r represents the total number of hidden layers. In Algorithm 1, $\mathbf{a}^{(s)}$ represents the bias of the s -th visible layer, $\mathbf{b}^{(s)}$ represents the bias of the s -th hidden layer, and $\mathbf{w}^{(s)}$ represents the weight of the pairwise interaction between the s -th layer and the $(s-1)$ -th layer. $\hat{e}(t)$ represents the output value of the DBN model, which is the predictive output of the residual $e(t)$ in (5).

Algorithm 1: Pseudo code for training the first RBM

Input values of the DBN: $\mathbf{e}_m(t) = (e(t-1), e(t-2), \dots, e(t-q))^T$, ε_1 represents the learning rate.

Initialization:

Weights: $\mathbf{w}^{(s)}$

Bias: $\mathbf{a}^{(s)} = \mathbf{0}$, $\mathbf{b}^{(s)} = \mathbf{0}$

$\mathbf{w} \leftarrow \mathbf{w}^{(s)}$

$\mathbf{b} \leftarrow \mathbf{b}^{(s)}$

$\mathbf{a} \leftarrow \mathbf{a}^{(s)}$

for $i=1:1:\text{numepochs}$

Positive phase:

$\mathbf{v}^{(0)} \leftarrow \mathbf{e}_m(t)$

$\mathbf{h}^{(1)} \leftarrow \varphi(\mathbf{b} + \mathbf{w}\mathbf{v}^{(0)})$

Negative phase: according to Eq. (11) and Eq. (12)

Update parameters: according to Eqs. (13-15)

$\mathbf{w}^{(s)} \leftarrow \mathbf{w}^{(s)} + \varepsilon_1 (\mathbf{v}^{(0)}\mathbf{h}^{(0)} - \mathbf{v}^{(1)}\mathbf{h}^{(1)})$

$\mathbf{a}^{(s)} \leftarrow \mathbf{a}^{(s)} + \varepsilon_1 (\mathbf{v}^{(0)} - \mathbf{v}^{(1)})$

$\mathbf{b}^{(s)} \leftarrow \mathbf{b}^{(s)} + \varepsilon_1 (\mathbf{h}^{(0)} - \mathbf{h}^{(1)})$

end

Algorithm 2: Pseudo code for fine-tuning

$T_d = \{\mathbf{e}_m(t), e(t)\}$, $\mathbf{e}_m(t)$ represents the input data, $e(t)$ represents the target value at time t , ε_f represents the fine-tuned learning rate.

While error does not meet the criteria

Forward propagation:

$$\mathbf{h}^{(0)}(\mathbf{e}_m(t)) \leftarrow \mathbf{e}_m(t)$$

For $s \in \{1, \dots, N_r\}$

$$\mathbf{h}^{(s)}(\mathbf{e}_m(t)) = \varphi(\mathbf{b}^{(s)} + \mathbf{w}^{(s)}\mathbf{h}^{(s-1)}(\mathbf{e}_m(t)))$$

End for

$$\hat{e}(t) = \varphi(\mathbf{b}^{(N_r)} + \mathbf{w}^{(N_r)}\mathbf{h}^{(N_r-1)}(\mathbf{e}_m(t)))$$

% $\hat{e}(t)$ represents the predictive value at time t .

% Backward gradient propagation and parameter update:

The error is calculated by $\zeta(t) = \frac{1}{2}(e(t) - \hat{e}(t))^2$;

$$\mathbf{w}^{(s)} \leftarrow \mathbf{w}^{(s)} + \varepsilon_f \frac{\partial \zeta(t)}{\partial \mathbf{w}^{(s)}}$$

$$\mathbf{b}^{(s)} \leftarrow \mathbf{b}^{(s)} + \varepsilon_f \frac{\partial \zeta(t)}{\partial \mathbf{b}^{(s)}}$$

End while

Finally, the output value of the DBN model is $\hat{e}(t)$, which is the predicted value of the residual $e(t)$, where $\hat{e}(t)$ can be calculated as follows.

$$\begin{cases} \hat{e}(t) = \varphi(\mathbf{w}_1^{(N_r)}\mathbf{h}^{(N_r-1)}(t) + b_1^{(N_r)}) \\ \mathbf{h}^{(\ell)}(t) = (h_1^{(\ell)}(t), h_2^{(\ell)}(t), \dots, h_{Q_\ell}^{(\ell)}(t))^T, \ell \in \{1, 2, \dots, N_r-1\} \\ h_{n_\ell}^{(\ell)}(t) = \varphi(\mathbf{w}_{n_\ell}^{(\ell)}\mathbf{h}^{(\ell-1)}(t) + b_{n_\ell}^{(\ell)}), n_\ell \in \{1, 2, \dots, Q_\ell\} \\ \mathbf{w}_{n_\ell}^{(\ell)} = (w_{n_\ell,1}^{(\ell)}, w_{n_\ell,2}^{(\ell)}, \dots, w_{n_\ell,Q_{\ell-1}}^{(\ell)}), Q_0 = q \\ \mathbf{h}^{(0)}(t) = (e(t-1), e(t-2), \dots, e(t-q))^T \end{cases} \quad (16)$$

where $\mathbf{w}_{n_\ell}^{(\ell)}$ denotes the weight matrix between the layer ℓ and layer $\ell-1$, $(b_1^{(\ell)}, b_2^{(\ell)}, \dots, b_{Q_\ell}^{(\ell)})$ represents the bias of layer ℓ , Q_ℓ represents the number of nodes in layer ℓ , N_r represents the total number of layers, and $\mathbf{h}^{(\ell)}(t)$ denotes the output values of layer ℓ .

Finally, the final forecasting value of the time series using the AR-DBN model or ARIMA-DBN model is

$$\hat{y}(t) = \hat{y}_L(t) + \hat{e}(t) \quad (17)$$

The pseudo-code for the hybrid LR-DBN model is presented in Algorithm 3, and the modelling procedure of the proposed method is as follows:

- Stage 1: **Linear modelling.** The AR model (1) and ARIMA model (3) are used to estimate linear information from the observations. Then, the residuals $e(t)$ are obtained from this stage. The residuals are used as input data for the next stage.
- Stage 2: **Nonlinear modelling.** The DBN models are trained using the residuals of the AR model and ARIMA model. The coefficients of the two DBN models are adjusted.
- Stage 3: **Combining.** The predictive results of the first stage and second stage are combined, which results in the final predicted values of the hybrid LR-DBN

Algorithm 3: Hybrid LR-DBN model algorithm**Begin**

Initialization

Pick the input example from the training set

for the entire training data

use the AR model and ARIMA model to fit the data

 obtain the predicted value $\hat{y}_L(t)$ according to model (1) and model (3) obtain the residuals of LR modelling $e(t)$ according to Eq. (5)**end****repeat** **for** every input $e(t)$ in the training data **for** all hidden units and visible units update h_i and v_i according to Eqs. (7-8) **end** update weights w , bias a and b according to Eqs. (13-15) **end****until**

Use the BP algorithm to fine-tune the weights of each RBM.

return DBN output $\hat{e}(t)$ return the final predicted value $\hat{y}(t)$ according to Eq. (17)

According to the prediction results of the AR-DBN model and ARIMA-DBN model, the model with the minimum MSE is selected as the final prediction model.

end

models. According to the prediction results of the AR-DBN model and ARIMA-DBN model, the LR-DBN model with the minimum MSE is used as the final prediction model.

3 Application of the hybrid model to time series

In this section, experimental studies are presented to demonstrate the effectiveness and superiority of the proposed hybrid LR-DBN model. Four time series (i.e., the

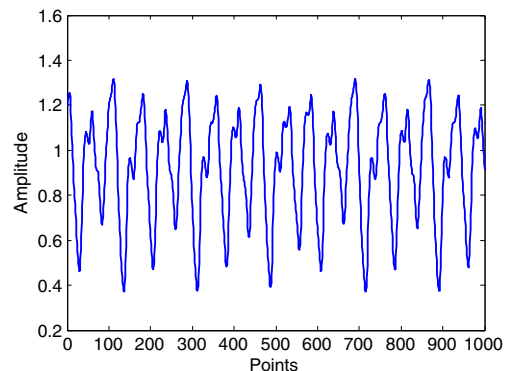


Fig. 2 Time series generated from the Mackey-Glass equation

Table 1 Comparison results for the Mackey-Glass time series

Model	Estimation method	Training set		Testing set	
		MSE	AIC	MSE	AIC
FIS (5 lags) [40]	ANFIS	3.42×10^{-7}	-6999.6	3.83×10^{-7}	-6944.0
RBF(5,5) [41]	EPA	2.92×10^{-4}	-4007.4	3.50×10^{-4}	-3916.8
RBF(5,5) [42]	SNPOM	1.16×10^{-7}	-7921.5	1.38×10^{-7}	-7835.7
RBF-AR(5,3,2) [41]	EPA	1.23×10^{-7}	-7895.5	1.31×10^{-7}	-7864.0
RBF-AR(5,3,2) [42]	SNPOM	1.08×10^{-7}	-7960.0	1.26×10^{-7}	-7880.3
LLRBF-AR(5,3,2) [26]	SNPOM	7.09×10^{-8}	-8134.5	9.25×10^{-8}	-8001.8
AR(5)		2.64×10^{-7}	-7484.1	2.61×10^{-7}	-7491.4
DBN(5,7,19,35,1) [13]		3.37×10^{-6}	-4671.6	3.55×10^{-6}	-4633.6
LSTM(5,20,1)		8.92×10^{-3}	-4721.1	8.81×10^{-3}	-4621.2
ANN(5,15,1)		1.63×10^{-6}	-6384.0	4.77×10^{-5}	-5036.7
ARIMA(5,1,4)		1.70×10^{-7}	-7839.1	1.54×10^{-7}	-7865.3
ARIMA(5,1,4)-ANN(5,15,1)	HYBRID	1.39×10^{-7}	-7677.1	2.72×10^{-7}	-7905.1
AR(5)-DBN(5,4,9,1)	HYBRID	1.67×10^{-8}	-8618.5	1.99×10^{-8}	-8531.6
EMD-DBN(5,9,15,1) [32]	HYBRID	4.90×10^{-6}	-4228.2	6.17×10^{-6}	-3978.8
ARIMA(5,1,4)-DBN(5,4,9,1)	HYBRID	1.01×10^{-8}	-8899.2	1.71×10^{-8}	-8621.6

Mackey-Glass, sunspot, Individual Household Electric Power Consumption (IHEPC) and electricity load demand data sets from the Australian Energy Market Operator (AEMO) are used to evaluate the proposed hybrid LR-DBN model. The modelling results of the proposed method are compared with the modelling results reported in some studies. To verify the model prediction accuracy, three criteria (i.e., the mean square error (MSE), normalized mean square error (NMSE) and root mean square error (RMSE)) are introduced to measure the performance of the proposed hybrid LR-DBN model as follows:

$$MSE = \frac{1}{N-p-q} \sum_{t=p+q+1}^N (y(t) - \hat{y}(t))^2 \tag{18}$$

$$RMSE = \sqrt{\frac{1}{N-p-q} \sum_{t=p+q+1}^N (y(t) - \hat{y}(t))^2} \tag{19}$$

$$NMSE = \left(\frac{\sum_{t=p+q+1}^N (y(t) - \hat{y}(t))^2}{\sum_{t=p+q+1}^N (y(t) - \bar{y})^2} \right) \tag{20}$$

where \bar{y} represents the mean value of the observation data.

When using the DBN model, we normalize all of the training and testing data by the following computation:

$$y^*(t) = \frac{y(t) - \min(y(t))}{\max(y(t)) - \min(y(t))} \tag{21}$$

where $y^*(t)$ represents the normalized value of $y(t)$.

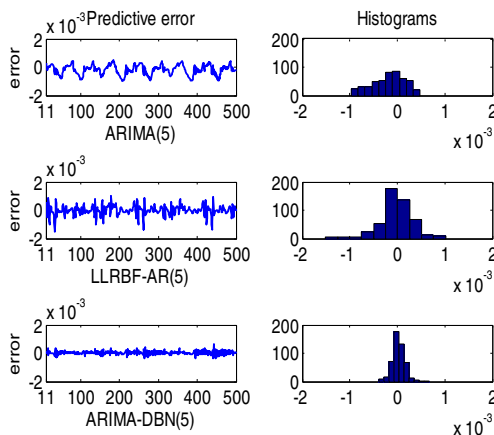


Fig. 3 Predictive errors and histograms by the three models for the testing data

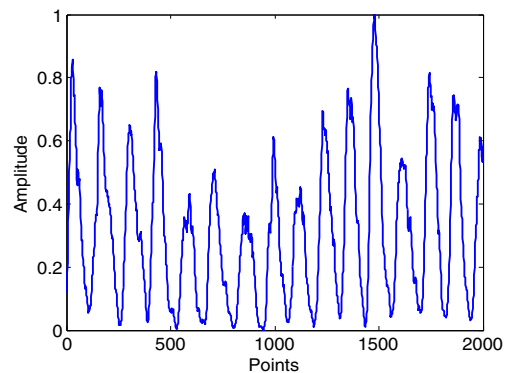


Fig. 4 Sunspot time series

Table 2 Modelling performance comparison for the test data in a sunspot series

Prediction method	MSE	RMSE	NMSE
WP-MLP [49]	Not provided	Not provided	1.25×10^{-1}
McNish-Lincoln [50]	Not provided	Not provided	8×10^{-2}
Sello-nonlinear method [51]	Not provided	Not provided	3.4×10^{-1}
Waldmeier [51]	Not provided	Not provided	5.6×10^{-1}
Denkmayr [52]	Not provided	Not provided	1.85
RBF-OLS [46]	Not provided	Not provided	4.6×10^{-2}
LLNF-LoLiMot [46]	Not provided	Not provided	3.2×10^{-2}
ERNN [47]	Not provided	1.29×10^{-2}	2.8×10^{-3}
MLP [53]	Not provided	Not provided	9.79×10^{-2}
Hybrid Elman-NARX with residual [45]	1.41×10^{-4}	1.19×10^{-2}	5.9×10^{-4}
SL-CCRNN [48]	Not provided	1.66×10^{-2}	1.47×10^{-3}
AR(5)	3.26×10^{-5}	5.7×10^{-3}	5.33×10^{-4}
ARIMA(5,1,4)	7.56×10^{-4}	2.75×10^{-2}	1.24×10^{-2}
DBN(5,6,1)	8.41×10^{-5}	9.2×10^{-3}	1.4×10^{-3}
LSTM(5,30,1)	1.81×10^{-3}	4.29×10^{-2}	3.2×10^{-2}
EMD-DBN(5,9,15,25,1) [32]	6.56×10^{-5}	8.1×10^{-3}	9.24×10^{-4}
ARIMA(5,1,4)-ANN(5,15,1)	2.09×10^{-4}	1.45×10^{-2}	3.4×10^{-3}
ARIMA(5,1,4)-DBN(5,6,1)	3.72×10^{-5}	6.1×10^{-3}	6.06×10^{-4}
AR(5)-DBN(5,6,1)	3.11×10^{-5}	5.6×10^{-3}	5.04×10^{-4}

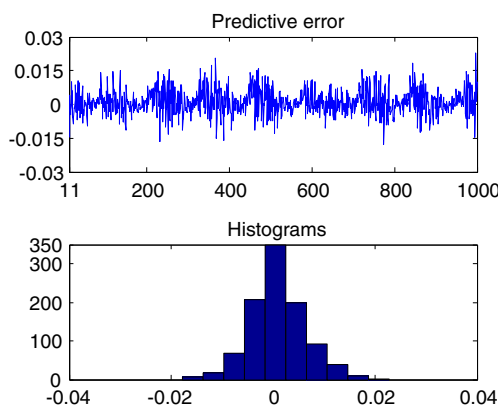
3.1 Mackey-glass time series modelling

Here, we use the famous Mackey-Glass Eq. (22) to obtain the time series and set $a = 0.2$, $b = 0.1$, and $c = 10$. Different values of τ produce various degrees of chaos.

$$\frac{dy}{dt} = \frac{ay^{(t-\tau)}}{1 + y^c(t-\tau)} - by(t) \quad (22)$$

For fair comparison, we select $\tau = 20$, as used in Gan et al. [26]. This chaotic time series model was also studied by Jang and Gulley [40] and Shi and Tamura [41]. We use 1000 data points, in which the first 500 observations are used to train the model, and the last 500 observations are used to test the modelling performance. The Mackey-Glass time series is shown in Fig. 2.

In the first modelling stage, the order p of the LR model is five, which is the same as that set in the literature [26, 40–42]. The AR model and ARIMA model are used to fit the original

**Fig. 5** Prediction error analysis of the AR-DBN model

data. In the process of dealing with the LR modelling residuals, the structure of the DBN is $N_r = 3$, $Q_0 = 5$, $Q_1 = 4$, $Q_2 = 9$, and $Q_3 = 1$. The nonlinear DBN model in LR-DBN is fine-tuned by the BP algorithm, and the number of fine-tuning occurrences is 1000 in all cases. According to the prediction results of the estimated AR-DBN and ARIMA-DBN models, the model with the minimum MSE is selected as the final prediction model. As seen in Table 1, the ARIMA-DBN model is selected as the prediction model, and the orders of the ARIMA are $p = 5$, $d = 1$, $\lambda = 4$. The modelling results of the proposed hybrid LR-DBN model are shown in Table 1. For comparison, the modelling results of the LLRBF-AR [26], RBF-AR [42], FIS [40], RBF [42], linear AR, ARIMA-ANN, LSTM [19], DBN [13] and EMD-DBN [32] models are also listed in Table 1, from which it can be seen that the ARIMA-DBN model yields the smallest MSE and AIC, thus

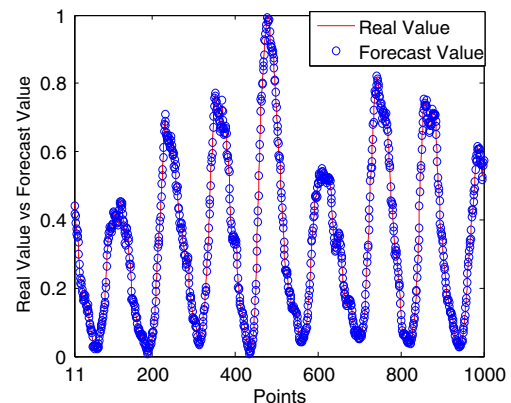
**Fig. 6** Comparison between the original sunspot time series and the final predicted values

Table 3 Modelling results comparison for IHEPC-GAP data

Dataset	Model	Training set				Testing set	
		Adjustable parameters	Time	MSE	AIC	MSE	AIC
IHEPC-GAP	AR(20)	21	95.63 s	1.51×10^{-3}	-194,580	1.52×10^{-3}	-25,813
	DBN(20,35,65,1)	3141	871 min	8.14×10^{-6}	-345,520	2.03×10^{-7}	-55,829
	ARIMA(20,1,20)	41	157.37 s	1.22×10^{-3}	-201,020	1.19×10^{-3}	-26,702
	EMD-DBN(20,35,65,1) [32]	31,410	1650 min	1.37×10^{-6}	-346,000	9.51×10^{-7}	5156.2
	ARIMA(20,1,20)-ANN(20,55,1)	1251	601 min	1.13×10^{-4}	-279,349	6.30×10^{-6}	-45,218
	LSTM (20,30,1)	661	30 min	7.67×10^{-5}	-307,210	3.01×10^{-5}	-41,966
	ARIMA(20,1,20)-DBN(20,35,65,1)	3182	817 min	4.41×10^{-6}	-385,610	3.51×10^{-6}	-49,528
	AR(20)-DBN(20,35,65,1)	3162	866 min	1.51×10^{-7}	-469,840	7.47×10^{-8}	-61,717

providing the best modelling result from an Akaike information criterion (AIC) point of view compared to other models.

For further comparison, we depict the predictive errors and their histograms by the ARIMA model, the LLRBF-AR model and ARIMA-DBN model for the testing data of the Mackey-Glass series in Fig. 3, from which it is seen that the prediction accuracy of the ARIMA-DBN model is better than that of the other two models.

3.2 Sunspot data modelling

A sunspot series is the most basic parameter used to describe the level of solar activity. Studying sunspot data models plays an important role in protecting the environment [43]. The smoothed, monthly sunspot time series from November 1834 to June 2001 (2000 points) in Fig. 4 is obtained from the SIDC (World Data Center for the Sunspot Index) [44]. The data are the same as those used in the literature [45–53], where the original data are scaled between [0, 1]. The data contain the linear part and nonlinear part, which are widely used in hybrid modelling [1, 25]. To evaluate the advantages of the proposed model, a fair comparison is required. Therefore, the sunspot data are also divided into two parts, as in the literature. The first 1000 observations are used to train the model, and the last 1000 observations are used to test the modelling performance.

In this case, the LR model order p is selected as 5, as in [45–53]. After obtaining the residuals of the LR model (i.e., the nonlinear part), we then use the DBN to fit the residuals. The structure parameters of the DBN model used here are $N_r = 2$, $Q_0 = 5$, $Q_1 = 6$, and $Q_2 = 1$, and the number of fine-tuning occurrences is 400. The model performance measures of the proposed AR-DBN model and other models are given in Table 2. It can be seen from Table 2 that the AR-DBN model gives the best modelling result compared to other models for the testing data.

To ensure that the problem can be explained clearly, we analyse the predictive errors of the proposed AR-DBN model for the testing data, which is plotted in Fig. 5. It can be seen from Fig. 5 that the residuals are small, and their histograms have a reasonably symmetric shape around zero and a Gaussian appearance. The original values of the sunspot time series and their predicted values are compared in Fig. 6, which shows that the AR-DBN model achieves a good prediction accuracy near the peaks and valleys. These results show the good statistical properties of the estimated hybrid model and prove that the estimated hybrid model can represent the dynamic behaviour of the original data well.

3.3 University of California Irvine repository data modelling

In general, deep learning algorithms are more advantageous for large-scale data. To use a larger data set to further validate

Table 4 The results of the one-day ahead load forecasting for testing data of the AEMO time series

Data	Month	Metrics	Prediction model											
			Persistence [32]	SVR [34]	ANN [35]	DBN [13]	RF [36]	EDBN [33]	EMD-SVR [37]	EMD-ANN [38]	EMD-RF [32]	EMD-DBN [32]	LSTM [19]	AR-DBN
TAS	Jan	RMSE	89.82	60.97	69.92	63.96	65.90	60.68	61.73	63.38	58.51	56.10	118.02	53.21
	Apr	RMSE	157.73	111.89	94.40	93.81	92.64	109.78	104.59	87.41	86.61	85.13	190.61	83.60
	Jul	RMSE	120.47	90.99	89.17	87.30	90.48	85.19	92.54	82.92	81.34	73.91	126.02	80.35
	Oct	RMSE	109.46	79.45	72.86	75.73	69.80	80.81	82.85	80.85	73.86	68.26	128.89	67.51

the effectiveness of the proposed hybrid modelling method, in this subsection, experiments are carried out on UCI (University of California, Irvine) benchmark data sets [54]. The modelling data used, with 34,000 data points, are the Global Active Power data, which are extracted from the Individual Household Electric Power Consumption (IHEPC) data set, and we use the first 30,000 data points as training data and the following 4000 data points as testing data. We denote the data as IHEPC-GAP.

The order p of the linear regression model in the AR-DBN or ARIMA-DBN model is determined by the AIC values. For the IHEPC-GAP data, we finally select $p = 20$. The structure parameters of the DBN model used here are $N_r = 3$, $Q_0 = 20$, $Q_1 = 35$, $Q_2 = 65$, and $Q_3 = 1$, and number of fine-tuning occurrences is 5000. The modelling results of the different models are given in Table 3, which shows that the AR-DBN model with the smallest MSE and AIC gives the best modelling performance for the IHEPC-GAP data compared with the other models.

Table 3 also gives the training time and the number of adjustable parameters for each model, which shows that the number of training times for the DBN-type models is approximate (excluding the EMD-DBN) and longer than that of the other models. This is because of the complexity of training the DBN model and hybrid model. In general, the training time or adjustable parameters of the LR-DBN model are longer or greater than those of other single models when the DBN module has many hidden layers, respectively. However, training of the LR-DBN model performs offline; therefore, in most cases, the use of the LR-DBN model in practice may not be affected by long-term offline training. In this way, we computed the results using a PC with an Intel i7-3770 CPU (3.4 GHz and 8 GB-RAM).

3.4 Prediction of the electric load time series from AEMO

In this subsection, using the electricity load time series from the Australian Energy Market Operator (AEMO) [55], the performance of the proposed hybrid modelling method is evaluated by comparing it with other eleven benchmark modelling methods (i.e., the persistence method [32], artificial neural network (ANN) [35], DBN [13], support vector regression (SVR) [34], ensemble DBN (EDBN) [33], empirical mode decomposition (EMD)-based SVR model (EMD-SVR) [37], EMD-based ANN (EMD-ANN) [38], EMD-based random forest (RF [36]) (EMD-RF) [32], EMD-based DBN (EMD-DBN) [32] and LSTM [19]).

For fair comparison, data sets from the year 2013 for Tasmania (TAS) are chosen to train and test the proposed hybrid model and other compared models. For TAS, January, April, July and October data are used to reflect the different seasons. In the experiment, the first 3 weeks of data

are used to train the model, and data from the remaining week are used to test the model [32]. The electricity load demand data from AEMO is sampled every half hour, which means that there are 48 data points for 1 day [32, 56]. Therefore, there are 1008 data points for training and 336 data points for testing [32]. In this paper, for one-day ahead load demand forecasting (i.e., the input data are composed of the data points from $y(t-48)$ to $y(t-96)$), $y(t)$ represents the output of the hybrid modelling method, which is the same as that in [32].

According to the prediction results of the AR-DBN model and the ARIMA-DBN model, the AR-DBN model with the minimum MSE is selected as the final prediction model. The prediction results of the one-day ahead load forecasting are given in Table 4 using the estimated AR-DBN model and the other eleven benchmark methods for the testing data. Table 4 shows that the proposed hybrid modelling method gives better prediction results than the other methods in most cases. Thus, it verifies the advantage of the proposed hybrid model for the time series prediction.

4 Conclusion

In this paper, a novel hybrid model composed of the LR model and DBN model was proposed to overcome the deficiencies in a single LR model or a single DBN model. Using an LR model or nonlinear DBN model alone may be difficult in characterizing a time series accurately, while the proposed hybrid model, which combines the merits of the LR model and DBN model, could be better than a single model. After first using a linear AR model or ARIMA model to reveal the linear part of the time series, the residuals of the AR or ARIMA models only contain the nonlinear behaviour of the time series. Next, the DBN better models the nonlinear part of the time series. Case studies for four well-known time series indicate that the proposed hybrid model has better modelling accuracy than some single models and hybrid models. The main reason is that the DBN has the strong ability to extract features among layers and self-organization characteristics. According to the four experimental results, we observed that the more training data there are, the higher the prediction accuracy. This is determined by the characteristics of deep learning.

Acknowledgements The authors would like to thank the editors and referees for their valuable comments and suggestions, which substantially improved the original manuscript. This research was supported by the National Natural Science Foundation of China (61773402, 51575167, 61540037, and 71271215).

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

References

1. Babu CN, Reddy BE (2014) A moving-average filter based hybrid Arima-ann model for forecasting time series data. *Appl Soft Comput* 23(10):27–38
2. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. *Neural Comput* 18(7):1527–1554
3. Dedinec A, Filiposka S, Dedinec A, Kocarev L (2016) Deep belief network based electricity load forecasting: an analysis of macedonian case. *Energy* 115:1688–1700
4. Hrasko R, Pacheco AGC, Krohling RA (2015) Time series prediction using restricted boltzmann machines and backpropagation. *Procedia Comput Sci* 55:990–999
5. Vu DH, Muttaqi KM, Agalgaonkar AP, Bouzerdoum A, Vu DH, Muttaqi KM (2017) Short-term electricity demand forecasting using autoregressive based time varying model incorporating representative data adjustment. *Appl Energy* 205:790–801
6. Wang JJ, Wang JZ, Zhang ZG, Guo SP (2012) Stock index forecasting based on a hybrid model. *Omega* 40(6):758–766
7. Lydia M, Kumar SS, Selvakumar AI, Kumar GEP (2016) Linear and non-linear autoregressive models for short-term wind speed forecasting. *Energy Convers Manag* 112:115–124
8. Akouemo HN, Povinelli RJ (2017) Data improving in time series using ARX and ANN models. *IEEE Trans Power Syst* 32(5):3352–3359
9. Choubin B, Malekian A (2017) Combined gamma and m-test-based ANN and ARIMA models for groundwater fluctuation forecasting in semiarid regions. *Environ Earth Sci* 76(15):538
10. Suk HI, Wee CY, Lee SW, Shen D (2016) State-space model with deep learning for functional dynamics estimation in resting-state fmri. *Neuroimage* 129:292–307
11. Pektaş AO, Cigizoglu HK (2013) ANN hybrid model versus ARIMA and ARIMAX models of runoff coefficient. *J Hydrol* 500(11):21–36
12. Kuremoto T, Kimura S, Kobayashi K, Obayashi M (2014) Time series forecasting using a deep belief network with restricted boltzmann machines. *Neurocomputing* 137(15):47–56
13. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507
14. Shen F, Chao J, Zhao J (2015) Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing* 167:243–253
15. Huang HB, Li RX, Yang ML, Lim TC, Ding WP (2017) Evaluation of vehicle interior sound quality using a continuous restricted boltzmann machine-based dbn. *Mech Syst Signal Process* 84: 245–267
16. Abdel-Zaher AM, Eldeib AM (2016) Breast cancer classification using deep belief networks. *Expert Syst Appl* 46:139–144
17. Yousefi-Azar M, Hamey L (2017) Text summarization using unsupervised deep learning. *Expert Syst Appl* 68:93–105
18. Huang Z, Siniscalchi SM, Lee CH (2016) A unified approach to transfer learning of deep neural networks with applications to speaker adaptation in automatic speech recognition. *Neurocomputing* 218:448–459
19. Ma X, Tao Z, Wang Y, Yu H, Wang Y (2015) Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp Res C* 54:187–197
20. Zheng H, Yuan J, Chen L (2017) Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. *Energies* 10(8):1168
21. Probst M, Rothlauf F, Grahl J (2017) Scalability of using restricted boltzmann machines for combinatorial optimization. *Eur J Oper Res* 256(2):368–383
22. Dahl GE, Yu D, Deng L, Acero A (2012) Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans Audio Speech Lang Process* 20(1):30–42
23. Memisevic R, Hinton GE (2014) Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural Comput* 22(6):1473–1492
24. Khashei M, Bijari M (2011) A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Appl Soft Comput* 11(2):2664–2675
25. Zhang GP (2003) Time series forecasting using a hybrid ARIMA ARIMA and neural network model. *Neurocomputing* 50(1):159–175
26. Gan M, Peng H, Peng X, Chen X, Inoussa G (2010) A locally linear RBF network-based state-dependent AR model for nonlinear time series modeling. *Inf Sci* 180(22):4370–4383
27. Liu H, Tian HQ, Li YF (2012) Comparison of two new Arima ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction. *Appl Energy* 98(1):415–424
28. Zhu B, Chevallier J (2013) Carbon price forecasting with a hybrid Arima and least squares support vector machines methodology. *Omega* 41(3):517–524
29. Nourani V, Baghanam AH, Adamowski J, Kisi O (2014) Applications of hybrid wavelet-artificial intelligence models in hydrology: a review. *J Hydrol* 514:358–377
30. Shukur OB, Lee MH (2015) Daily wind speed forecasting through hybrid KF-ANN model based on ARIMA. *Renew Energy* 76:637–647
31. Barak S, Sadegh SS (2016) Forecasting energy consumption using ensemble ARIMA-ANFIS hybrid algorithm. *Int J Electr Power Energy Syst* 82:92–104
32. Qiu X, Ren Y, Suganthan PN, Amaratunga GAJ (2017) Empirical mode decomposition based ensemble deep learning for load demand time series forecasting. *Appl Soft Comput* 54(C):246–255
33. Qiu X, Zhang L, Ren Y, Suganthan PN, Amaratunga G (2015) Ensemble deep learning for regression and time series forecasting. *Comput Intell Ensemble Learn* 1–6
34. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
35. Haykin S (1999) *Neural networks: a comprehensive foundation*, International Edition, Prentice Hall
36. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
37. Ye L, Liu P (2011) Combined model based on EMD-SVM for short-term wind powerprediction. In: *Proc. Chinese Society for Electrical Engineering (CSEE)* 31:102–108
38. Liu H, Chen C, Tian H, Li Y (2012) A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks. *Renew Energy* 48:545–556
39. Salakhutdinov R, Hinton G (2007) Using deep belief nets to learn covariance kernels for Gaussian processes. *International Conference on Neural Information Processing Systems* 1249–1256
40. Jang RJS, Gulley N (2000) *Fuzzy logic toolbox user's guide*. The Math Works, Inc., Natick
41. Shi Z, Tamura Y, Ozaki T (1999) Nonlinear time series modelling with the radial basis function-based state-dependent autoregressive model. *Int J Syst Sci* 30(7):717–727
42. Peng H, Ozaki T, Haggan-Ozaki V, Toyoda Y (2003) A parameter optimization method for radial basis function type models. *IEEE Trans Neural Netw* 14(2):432–438
43. Hipel KW, McLeod TI (1994) Time series modelling of water resources and environmental systems. *J Hydrol* 167(1–4):399–400
44. SIDC (World Data Center for the Sunspot Index), <http://sidc.oma.be/index.php3>
45. Ardalani-Farsa M, Zolfaghari S (2010) Chaotic time series prediction with residual analysis method using hybrid Elman-NARX neural networks. *Neurocomputing* 73(13):2540–2553

46. Gholipour A, Araabi BN, Lucas C (2006) Predicting chaotic time series using neural and neurofuzzy models: a comparative study. *Neural Process Lett* 24(3):217–239
47. Ma QL, Zheng QL, Peng H, Zhong TW, Xu LQ (2007) Chaotic time series prediction based on evolving recurrent neural networks. *Int Conf Mach Learn Cybern* 3496–3500
48. Chandra R, Zhang M (2012) Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction. *Neurocomputing* 86(12):116–123
49. Teo KK, Wang L, Lin Z (2001) Wavelet packet multi-layer perceptron for chaotic time series prediction: effects of weight initialization. *Comput Sci* 2074:310–317
50. Mcnish AG, Lincoln JV (1949) Prediction of sunspot numbers. *Eos Trans Am Geophys Union* 30(5):673–685
51. Sello S (2001) Solar cycle forecasting: a nonlinear dynamics approach. *Astron Astrophys* 377(1):312–320
52. Denkmayr K, Cugnon P (1997) About sunspot number medium-term predictions. In: Heckman G et al. (Ed.), *Solar-Terrestrial Prediction Workshop V*, Hiraiso Solar Terrestrial Research Center, Japan, 103
53. Koskela T, Lehtokangas M, Saarinen J, Kaski K (1996) Time series prediction with multilayer perceptron, FIR and Elman neural networks. In: *Proceedings of the World Congress on Neural Networks* 491–496
54. Lichman M (2013) UCI machine learning repository Irvine. University of California. Irvine, CA, USA. [Online]. <http://archive.ics.uci.edu/ml>
55. AEMO, Australian Energy Market Operator (2013). <http://www.aemo.com.au/>
56. Qiu X, Suganthan PN, Amaratunga GAJ (2018) Ensemble incremental learning random vector functional link network for short-term electric load forecasting. *Knowl-Based Syst* 145:182–196



Wenquan Xu received the M.S. degree in Control theory and control engineering from Hohai University, Nanjing, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Central South University, Changsha, China. He is a lecturer with the School of Physics and Electronic Engineering, Anqing Normal University, Anqing, China. His current research interests include deep learning, system identifica-

tion, and nonlinear time series analysis.

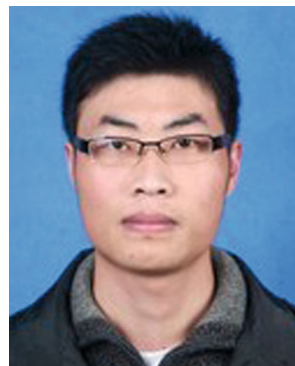


Hui Peng received the B. Eng. and M. Eng degrees in control science and engineering from the Central South University, Changsha, China, in 1983 and 1986, and the Ph.D. degree in Statistical Science from the Graduate University for Advanced Studies, Japan, in 2003. He has been a Professor at the Central South University, Changsha, China, since 1998. His research interests include complex systems modeling, control and optimization; advanced

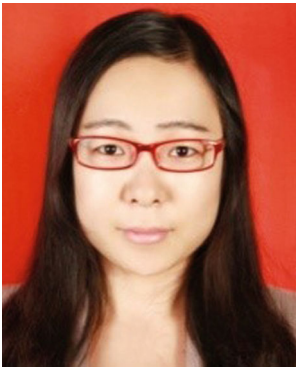
control theory and intelligent automation system; industrial process control system development. Corresponding author of this paper.



Xiaoyong Zeng obtained M. Eng degree in control science and engineering in 2006 in Wuhan University, Wuhan, China. He is currently working towards the Ph.D. degree in control science and engineering of the Central South University, Changsha, China. He is Lecturer of Changsha University of Science and Technology now. His current research interests are in complex systems modeling, optimization and control.



Feng Zhou received the B. Eng. and M. Engdegrees in control science and engineering from Central South University, Changsha, China, in 2005 and 2009, respectively. He is currently working towards the Ph.D. degree in control science and engineering of the Central South University, Changsha, China. His current research interests are in complex systems modeling, optimization and control.



Xiaoying Tian obtained M. Eng. degree in control science and engineering in 2015 in Henan Polytechnic University, Jiaozuo, China. She is currently working towards the Ph.D. degree in control science and engineering of the Central South University, Changsha, China. Her current research interests are in complex systems modeling, optimization and control.



Xiaoyan Peng received the B.S. and M.S. degrees in mechanical engineering and Ph.D. degree in automatic control from Hunan University, Changsha, China, in 1986, 1989, and 2013, respectively. She is currently a Professor in the College of Mechanical and Vehicle Engineering, Hunan University. Her research interests cover control of mechatronic systems and safety analysis of autonomous vehicles.