IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 31, 2018, accepted March 12, 2018, date of publication March 22, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2818108

A Model Combining Stacked Auto Encoder and Back Propagation Algorithm for Short-Term Wind Power Forecasting

RUNHAI JIAO[1], XUJIAN HUANG[1], XUEHAI MA[1], LIYE HAN[1], AND WEI TIAN[2], (Member, IEEE)
[1]School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China
[2]Electrical and Computer Engineering Department, Illinois Institute of Technology, Chicago, IL 60616, USA

Corresponding author: Runhai Jiao (runhaijiao@ncepu.edu.cn)

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2018ZD06.

ABSTRACT Recently, many countries have spent great efforts on wind power generation. Although there have been many methods in the field of wind power forecasting, the persistence statistics model based on historical data is still being challenged due to the randomness and uncontrollability in wind power. Hence, a more accurate and effective wind power forecasting method is still required. In this paper, a new forecasting method is proposed by combining stacked auto-encoders (SAE) and the back propagation (BP) algorithm. First, an SAE with three hidden layers is designed to extract the characteristics from the reference data sequence, and the subsequent loss function is used in the pre-training process to obtain the optimal initial connection weights of the deep network. Second, after adding one output layer to the stacked auto encoders, the BP algorithm is used to fine tune the weights of the whole network. To achieve the best network architecture, the particle swarm optimization is adopted to decide the number of neurons of the hidden layer and the learning rate of each auto encoder. Experimental results show that, for short-term wind power forecasting, the proposed method achieves more stable and effective performance than the existing BP neural network and support vector machines. The improvement in accuracy is 12% on average under different time steps.

INDEX TERMS Machine learning, particle swarm optimization, stacked auto-encoders, wind energy, wind power forecasting.

I. INTRODUCTION

In the last decade, wind power has been rapidly developed worldwide. Given that there are great volatility and intermittence in wind power, a large-scale wind farm connected to the electricity grid brings significant changes to the power grid's planning and operations. Thus, accurate wind power forecasting is critical to the establishment of a reasonable scheduling plan and to ensure that the power grid can operate safely and economically [1]–[4]. In particular, short-term wind power forecasting is extremely important since the system operators must address a large amount of fluctuating power generated from the increasing wind power. Short-term wind power forecasting could predict the power generation for the coming one or two days, and the time interval between two predicted values could range from minutes to hours.

Wind power forecasting method could be categorized into two types, the physical method and the statistical method. The former method mainly utilizes the laws of physics and atmospheric behavior to achieve predictions. The second method focuses on exploiting the relationships among historical data, which are subsequently used to establish the prediction model. In general, the physical method has advantages in long-term forecasting while the statistical method is better in short-term forecasting [5].

The statistical relationship in the historical data could be modeled by various methods, including persistence methods [6], linear methods (ARMA, ARIMA) [7], [8] and nonlinear methods. The autoregressive model has been considered as the simplest linear model and could outperform many other persistence models in very short-term forecasting. The autoregressive model has been widely used in practice despite its forecasting instability. The methods based on Artificial intelligence are nonlinear modeling methods [9]–[13]. Fuzzy models [14], wavelet-based models [15] and artificial neural

network (ANN)-based models [16] have shown their superiority over linear models. In addition, as a powerful machine learning tool, Support Vector Machines (SVMs) have been successfully used for time-series forecasting with satisfactory results in various fields [17]–[19]. Its main feature is the use of a kernel function to apply linear classification techniques to nonlinear classification problems. SVMs have excellent generalization ability and can address high dimensional data even with relatively small training samples [20]. However, more accurate wind power forecasting methods are still required. Recently, hybrid models have been extensively developed for short-term wind speed forecasting. Basing on the theories of wavelets, wavelet packets, time series analysis and ANNs, three hybrid models (Wavelet Packet-BFGS, Wavelet Packet-ARIMA-BFGS and Wavelet-BFGS) have been proposed to forecast the wind speed values [21]. It shows that these hybrid models have obtained better performances. In addition, another hybrid model (W-GP) that combines the theories of wavelet and Gaussian process learning paradigm has also been developed to forecast multi-step wind speed value. The combined method shows higher forecasting accuracy compared with a single method [22].

In general, it is difficult to establish analytical equations with parameters in practice to demonstrate the relationship among complex, unlabeled and high dimensional time series data. It is usually expensive and time consuming to manually extract domain-specified features for the traditional shallow ANN model. In addition, when ANNs are used as predictors, improper initial weights may affect the learning convergence speed and make learning fall into local optima [23]. Hinton *et al.* [24] proposed deep learning and presented a new wave of neural network research. Recently, deep artificial neural networks have won numerous contests in pattern recognition and machine learning. In addition to the MNIST handwriting challenge [25], Deep learning has been used in a substantial number of fields, such as face detection [26], speech recognition and detection [27]. Deep learning has obvious advantages when dealing with a large number of samples and nonlinear data. As one of deep learning architectures, SAE is fundamental in unsupervised learning and other tasks [28]. Hinton used SAEs to conduct a layer-by-layer unsupervised learning instead of manual selection to study the features of a mass of unlabeled data. SAE obtains the initial weights of the networks after pre-training, and then fine-tunes the whole network with global supervised learning. Moreover, SAE is more efficient because its objective function can be solved by fast backward propagation.

In this paper, we propose a novel wind power forecasting method based on SAEs that exploits the statistical relationship among the historical data through a deep neural architecture. The key is to establish a forecasting model through training on historical data, which is divided into two phases: the pre-training process and the fine-tuning process. Considering the complexity, the neural network in pre-training only consists of three stacked AEs including one visible layer, one hidden layer and one output layer. In the fine-tuning process, one more layer is added at the end of the pre-trained network, and then BP is adopted to fine-tune the weights of the whole network. Furthermore, to decide the proper neuron number of each hidden layer and the learning rate of each AE, we adopt Particle Swarm Optimization (PSO) [29] to optimize the parameters of the SAE and the whole network. The remaining part of the paper is organized as follows. In Section 2, the establishment and optimization of a general forecasting model is illustrated after a preliminary introduction of SAEs, BP and PSO. In Section 3, the forecasting method is applied to the wind power forecasting application, and the forecasting accuracy of the proposed model is evaluated through a substantial number of experiments and comparisons. Then, conclusions are given in Section 4.

II. THE PROPOSED FORECASTING MODEL BASED ON SAE AND BP

When machine learning (such as a neural network) is applied into regression problem, the key is properly constructing the training sample, which has not been well solved. The essence is to exploit the correlations between the historical time series data. In this paper, an SAE with sparse constraints and random noise is used to extract features from a mass of historical samples; then, the features are put into a BP network for regression analysis. To further improve the forecasting accuracy, the number of neurons in each hidden layer and the suitable learning rate for each basic AE are optimized by PSO.

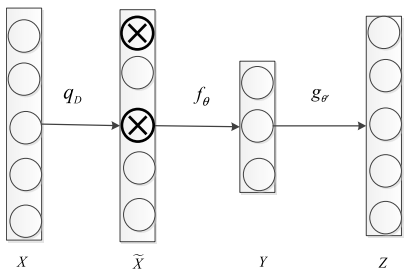


FIGURE 1. The basic architecture of the AE with random noise.

A. ESTABLISHMENT OF THE SAE_BP FORECASTING MODEL

An Auto-Encoder (AE) is a nonlinear feature extraction method mainly used to equate the inputs with the outputs. Usually, several AEs are stacked together to form an SAE, which typically has an input layer representing the original data and several hidden layers. The output values in the hidden layers represent the transformed features that can reconstruct the original data using the decoder. AEs have many variants; in our method, AEs with random noise (as shown in Figure 1) are adopted to guarantee the network's robustness. $\widetilde{X}$ is the transformation of X after adding noise. Y is the output of encoder, and Z is the output of the decoder.

In this paper, we propose a forecasting model based on SAEs that consist of three AEs in terms of spare constraints

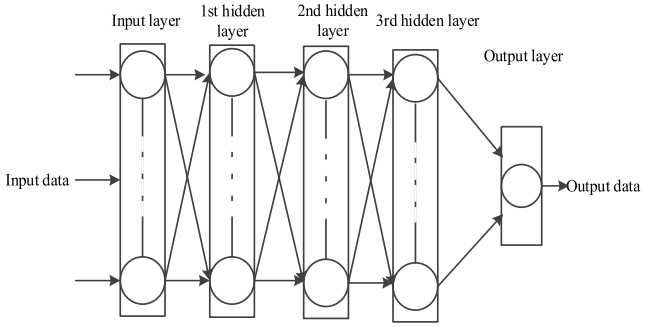


FIGURE 2. The structure of SAE_BP neural network.

and random noise, as shown in Figure 2. We use the features obtained by the last hidden layer and label data to train the whole network. In general, the proposed forecasting model is established through the pre-training process and the fine-tuning process. In the multi-layer network, the initial weights directly affect the training efficiency and the model's accuracy. If the initial weight is too large, it may be difficult to find a local minimum. Otherwise, the gradient of the front layers is too small. Pre-training can process extracted features and obtain more appropriate initial weights that are subsequently used in the fine-tuning process. The BP algorithm is adopted in both the pre-training process and the fine-tuning process.

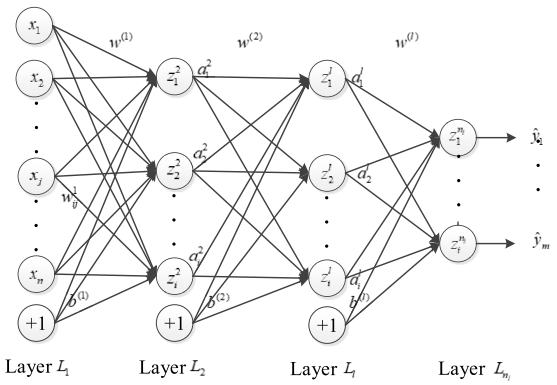


FIGURE 3. The structure of multi-layer neural network.

1) THE BASIC THEORY OF BACK PROPAGATION

A simple example can best illustrate the back propagation algorithm. In Figure 3, there are $n+1$ inputs $x_0, x_1, x_2, \ldots, x_n$ and a bias that are put into the neurons of the first hidden layer. $y_1, y_2, \ldots y_m$ are the expected output data, and $\hat{y}_1, \hat{y}_2, \ldots \hat{y}_m$ are the real output data of the network. We denote n_l as the number of layers in our network. We label layer l as L_l, which makes layer L_1 the input layer and layer L_{n_l} the output layer. $b^{(1)}, b^{(2)} \cdots b^{(l)}$ are the biases, which are set to $+1$. z_i^l is the weighted sum of the inputs to neuron i in layer $l$in formula (1), and a_i^l is the activated value of the ith neuron of the lth layer in formula (2). w_{ij}^l is the weight associated with the connection between the jth neuron in the lth layer and the ith neuron in the $(l+1)$th layer. f is a sigmoid activation function in formula (3).

$$z_i^l = \sum_{j=0}^{n} w_{ij}^{l-1} x_j^{l-1} + b_j^{l-1} \tag{1}$$

The activation value of lth layer is computed as follows:

$$a^l = f(z^{(l)}) \tag{2}$$

$$f(s) = \frac{1}{1+e^{-\beta s}} \tag{3}$$

Back propagation (referred as propagation) adopts the δ learning rule to adjust the weight between layers. The output error of the AE is $J(W, b; x, y)$, as shown in formula (4). An "error term" $\delta_i^{(l)}$ measures how much that neuron is "responsible" for any errors in the output. Connection weights of neurons and bias terms are adjusted according to the changes in (7) and (8).

$$J(W, b; x, y) = \frac{1}{2}||\hat{y} - y||^2 \tag{4}$$

For the output layer $l = n_l$, set

$$\delta^{(n_l)} = -(y - a^{(n_l)}) \bullet f'(z^{(n_l)}) \tag{5}$$

For other layers $l = n_l - 1, n_l - 2, n_l - 3, \ldots 2$, set

$$\delta^{(l)} = ((W^{(l)})^T \delta^{(l+1)}) \bullet f'(z^{(l)}) \tag{6}$$

Compute the partial derivatives:

$$\nabla_{W^l} J(W, b; x, y) = \delta^{l+1} (a^{(l)})^T \tag{7}$$

$$\nabla_{b^l} J(W, b; x, y) = \delta^{l+1} \tag{8}$$

2) TRAINING PROCESS OF THE PROPOSED SAE_BP FORECASTING MODEL

Since the proposed model has multiple hidden layers ($\geq$2), it is difficult to use the BP algorithm to optimize the weights of the entire network. In this regard, pre-training is adopted to train the stacked auto encoder network.

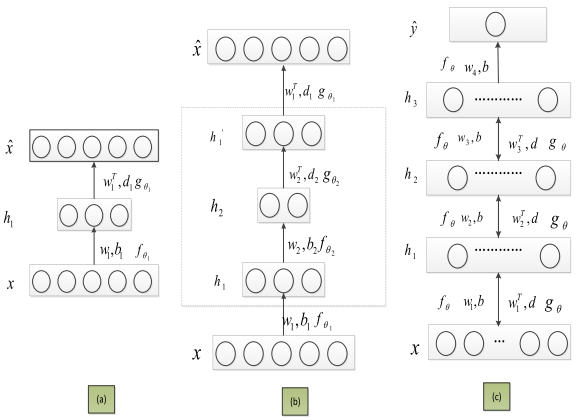


FIGURE 4. The pre-training and fine-tuning process of SAE_BP.

In the first stage, the SAE is assumed to have only three layers. They are input layer x, output layer $\hat{x}$ and output layer h_1 (which is the 1st hidden layer, as shown in Figure 4(a)).

The input is transformed into a low or a high dimensional code h_1 through the encoder function f_θ, and then h_1 reconstructs the original data x through the decoder function g_θ. The optimal reconstruction $\hat{x}$ is obtained by minimizing the reconstruction error in this stage using back propagation, and the minimizing reconstruction error function is shown as formula (9); f_{θ_i}, g_{θ_i} denotes a non-linear mapping, which here is an element-wise sigmoid function $1/(1+e^{-\beta x})$. The set of parameters $\theta_i = \{w_i, b_i, w_i^T, d_i\}, i = 1, 2, \cdots l-1$ (l denotes the number of layers in the SAE) is obtained through back propagation. b_i and d_i are the respective biases of the encoder and the decoder, and w_i and w_i^T are the respective weight matrices of the encoder and the decoder.

$$L(x, \hat{x}) = \frac{1}{2}||x - \hat{x}||^2 \tag{9}$$

In the second stage, a new hidden layer h_2 and output layer h_1' are stacked into the existing AE of Figure 4(a); they are subsequently combined with h_1 as the input layer to form a new AE, as illustrated in Figure 4(b). The second AE can similarly obtain a set of parameters using back propagation. By removing the last layer h_1' and using a similar process as above, we can stack many auto encoders. To reduce the computational complexity, three auto coders are stacked together in this paper. The combined first stage and second stage are called the pre-training process. It builds three stacked auto encoders containing three hidden layers h_1, h_2 and h_3 and trains the initial weights of the network.

In the third stage, we add an output layer and initialize the set of parameters w_4, b_4 between the last hidden layer and the output layer to form the whole SAE_BP neural network. All the weights and biases $w_i, b_i, i = 1, 2, \cdots l$ of the whole networks are trained together using the BP algorithm, which is called the fine-tuning process. Through the above three stages, a deep network with more than one single hidden layer has been trained to converge to a global minimum.

3) SAE WITH SPARSE CONSTRAINTS

When the number of neurons in the hidden layer is larger than that of the input layer, the minimization of the loss function is likely to train the SAE as an identity function. To solve this problem, the SAE imposes a sparsity constraint on the hidden layers to assess the input data structure. Therefore, for the input data, there are only a few hidden neurons activated in the whole network. Informally, we treat a neuron as active if its output value is close to 1, and it is inactive if its output value is close to 0. Set $\hat{\rho}_j$ to be the mean activation of the hidden neuron j, as shown in formula (10), where $a_j^{(2)}(x)$ is the activation of neuron j when the network is given a specific input x.

$$\hat{\rho}_j = \frac{1}{n}\sum_{i=1}^{n}[a_j^{(2)}(x^{(i)})] \tag{10}$$

Sparsity is imposed on the network by using a parameter ρ whose value is close to zero. We enforce the constraint $\hat{\rho}_j = \rho$. Here, KL divergence (as shown in formula (11)) is used to measure the similarity between the desired distribution and the actual distribution. It is called the penalty term since it penalizes $\hat{\rho}_j$ when it significantly deviates from ρ. The pre-training process is realized by minimizing the objective function shown in formula (12). The first term is the reconstruction costs, and the second term is the sparse mapping from the input layer to the hidden layer. β controls the weight of the sparsity penalty term.

$$KL(\rho||\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \tag{11}$$

$$\min_{w,b}[\sum_{i=1}^{m}(\hat{x}-x)^2 + \beta\sum_{j=1}^{k}KL(\rho||\hat{\rho}_j)] \tag{12}$$

TABLE 1. Model parameters optimized by PSO.

model	parameter	description
BP	n_1, n_2, n_3	The number of neurons in the first, second and third hidden layers
	ε	The Learning Rate of the BP network
SVM	c	Punishment coefficient of the SVM
	g	Radius of the kernel function
SAE_BP	n_1, n_2, n_3	The number of neurons in the first, second and third hidden layers
	$\varepsilon_1, \varepsilon_2, \varepsilon_3$	Learning Rate of first, second and third AEs
	ε_4	Learning Rate of the predictor
	b	The sample number of each batch

B. OPTIMIZATION OF THE PROPOSED MODEL

For a neural network, the number of neurons in the hidden layer and the learning rate of the network are the important parameters that remarkably affect the final model performance. In this paper, we establish an SAE_BP model that consists of three hidden layers and one output layer to forecast wind power. To improve the model's performance, we use PSO to determine the values of these important parameters. A particle is designed to be a multi-dimensional vector $x(n_d, \varepsilon_q)$. $n_d (d = 1, 2, 3)$ is considered as the number of neurons in the three hidden layers, and $\varepsilon_q (q = 1, 2, 3)$ are the learning rates of the three respective AEs. Then, we choose MAPE as the adaptive value of fitness function. The population is initialized with p particles $x_k (k = 1, 2 \ldots p)$. After a number of iterations, the optimal values of the parameters are determined and adopted in the final forecasting model. To make a fair comparison, the parameters of the BP model and the SVM model are also decided by the PSO algorithm. Table 1 shows three models' parameters that should be optimized by PSO, and the parameters settings for the PSO are shown in Table 2.

The flowchart of this optimization process is shown in Fig. 5, which is detailed as follows:

TABLE 2. The parameter settings of PSO for each model.

parameter	description	Value
p	Population of the PSO	15
I	PSO stop condition	50
c_1,c_2	Velocity coefficient	2
w	Damping coefficient	Decided by (15)
$w\_\max$	The maximum value of the Damping coefficient	0.9
$w\_\min$	The minimum value of the Damping coefficient	0.4
d_1	Number of particle dimensions in the BP	4
d_2	Number of particle dimensions in the SVM	2
d_3	Number of particle dimensions in the SAE_BP	8

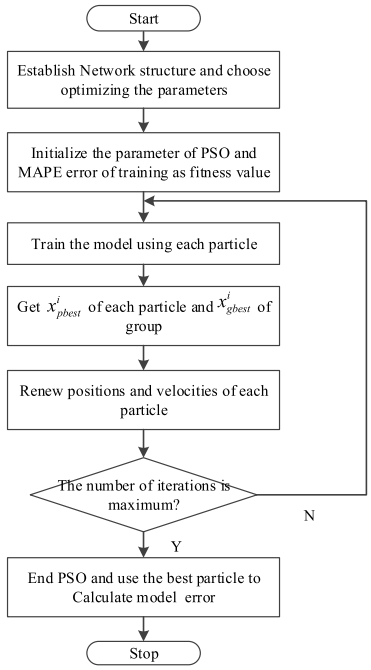


FIGURE 5. The optimization process of the proposed SAE_BP model.

Step 1 (Definition of the Solution Space and Fitness Function): Select the parameters that need to be optimized. Put these parameters into a reasonable range that includes the optimal solution. We then set a minimum and maximum value for each dimension in a multi-dimensional optimization.

Step 2 (Initialization): Decide the size of the population and the maximum number of iterations. Initialize the start position and the start velocity of each particle. The error between the prediction value and real data for each particle is evaluated by MAPE.

Step 3 (Optimization process): First, find the best position of a particle from its history, and the best particle position of the swarm. Second, the positions and velocities are updated by formulas (13) and (14), where is the inertia weight and is the learning factor.

$$x_k^{i+1} = x_k^i + v_k^{i+1} \tag{13}$$

$$v_k^{i+1} = wv_k^i + c_1(x^i_{pbest(k)} - x_k^i) + c_2(x^i_{gbest} - x_k^i) \tag{14}$$

$$w = w\_\max - (w\_\max - w\_\min) * i/I \tag{15}$$

Step 4 (Iteration End Judgment): If the evaluation function (predicted MAPE of training data) converges, then the optimization ends; otherwise, proceed to Step 3.

III. APPLICATION OF THE PROPOSED METHOD IN WIND POWER FORECASTING

To evaluate the accuracy of the proposed forecasting model, a real wind farm dataset is used to conduct the training and forecasting procedures. The data containing 6057 samples with an interval of 15 minutes are selected from the homepage of EirGrid [30]. The data samples from 1 May 2014 to 21 June 2014 are selected to form the training data and to establish the forecasting model from which we obtain a set of optimal parameters. The data samples from 22 June 2014 to 1 July 2014 are selected to form the verification data, where PSO is adopted to arrive at the best fitness value. At last, the established model is used to forecast the data in one day of July 2014, which contains 96 wind power data points in total.

A. WIND POWER FORECASTING

After obtaining the original time series of wind power, we get the training samples, verifying samples and testing samples that are all normalized. For a better comparison, we apply the BP model, the SVM model and the proposed model to the same dataset. For each training sample, we choose 12 historical wind power data points to form the reference data vector. The first to tenth elements of the reference vector are the wind powers of the time points immediately before the current time point, and the eleventh and twelfth elements are the wind powers at the same time point in the previous two days.

Suppose that the current time point is i and we would like to forecast the wind power of the next time point. The data in the reference vector is $X = \{x_i, x_{i-1}, x_{i-2}, \ldots x_{i-9}, x_{i-96}, x_{i-96\times2}\}$, where x_i are the wind power values of the i-th point in the current day, and x_{i-96} is the wind power of the i-th point of the prior day. This is called 1-step ahead forecasting, and the output of 1-step ahead forecasting is noted as $\hat{y}_{i+1}$. Based on the forecasting result of the $i+1$ time point, we can forecast the wind power of the $i+2$ time point, where the reference vector is $\{\hat{y}_{(i+1)}, x_i, x_{i-1}, x_{i-2}, \cdots x_{i-8}, x_{(i+1)-96}, x_{(i+1)-96\times2}\}$. This is called 2-step ahead forecasting. It is obvious that the previous forecasted wind power is included in the reference vector of the 2-step ahead forecasting. Similarly, we can realize multi-step ahead forecasting.

B. FORECASTING ACCURACY EVALUATION

To fully verify the proposed model's performance, three widely used forecast accuracy evaluation criteria are chosen

to compare the BP, the SVM, and the SAE_BP. These criteria are the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), and the Mean Absolute Percentage Error (MAPE) and are defined in formulas (17), (18) and (19)

$$e_i = x_i - \hat{x}_i \tag{16}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{17}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}e_i^2} \tag{18}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i} \times 100\right| \tag{19}$$

where x_i represents the actual wind power value at the ith time point, $\hat{x}_i$ represents the forecasted value for the same point, and n is the number of forecasted points.

TABLE 3. The optimal parameter values of the BP model for different steps.

	1-step	2-step	3-step	4-step	5-step	6-step	9-step
n_1	21	21	40	39	23	28	18
n_2	53	70	64	21	73	43	58
n_3	58	14	6	12	49	46	39
ε	7.8873	1.741	9.1673	0.4183	1.064	2.3071	0.8798

TABLE 4. The optimal parameter values of the SAE_BP model.

	1-step	2-step	3-step	4-step	5-step	6-step	9-step
n_1	46	18	47	69	50	96	28
n_2	63	53	38	31	27	65	29
n_3	56	75	51	17	29	65	3
ε_1	3.636	5.204	3.886	1.317	2.324	5.668	3.407
ε_2	2.984	0.978	9.677	2.075	6.135	3.821	2.115
ε_3	3.457	0.963	4.544	5.438	6.277	3.5324	2.829
ε_4	4.376	1.362	1.921	1.389	1.028	4.424	5.501
bs	17	463	593	610	676	208	795

TABLE 5. The optimal parameter values of the SVM model.

	1-step	2-step	3-step	4-step	5-step	6-step	9-step
c	9.4817	11.887	4.3905	4.801	7.4056	2.1594	2.5047
g	0.0716	0.001	0.0003	0.0001	0.0002	0.0003	0.0885

C. EXPERIMENTAL RESULTS AND ANALYSIS

To make a fair comparison of the different models, the key parameters of the PSO remain the same in each model. Table 3, Table 4 and Table 5 show the optimal parameter values obtained through PSO optimization for the different forecasted steps (1-step to 9-step), respectively.

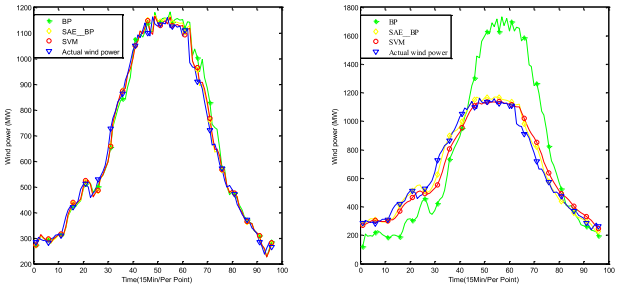


FIGURE 6. The 1-step ahead and 2-step ahead forecasting curves.

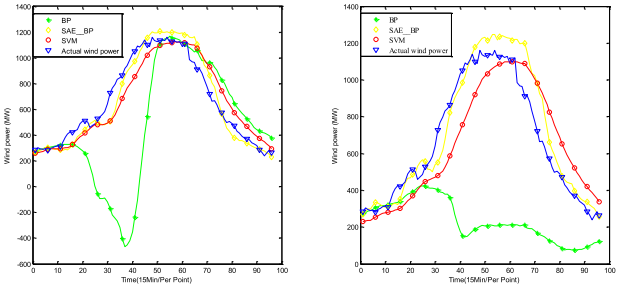


FIGURE 7. The 3-step ahead and 4-step ahead forecasting curves.

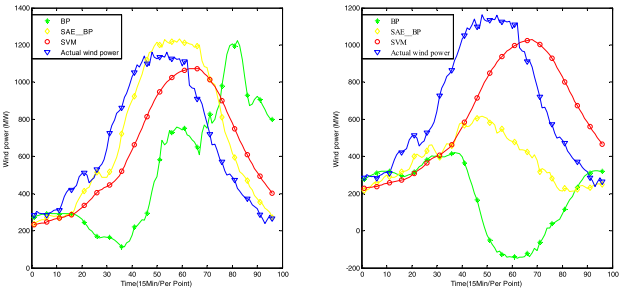


FIGURE 8. The 5-step ahead and 6-step ahead forecasting curves.

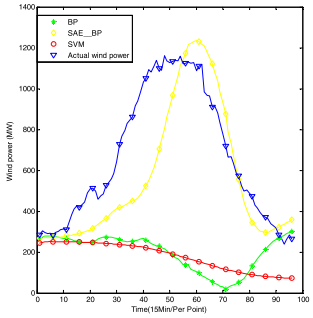


FIGURE 9. The 9-step ahead forecasting curve.

Figures 6-9 show the forecasting results for the 5962th-6057th data points under several steps using the three models. The forecasting errors are listed in Table 6. From Figures 6-9, the following can be observed; (1) the proposed method is more effective in forecasting the non-stationary wind power, and the model achieves a more accurate wind power forecasting performance than other models for most of

TABLE 6. Comparison of forecasting performance for multi-step wind power forecasting.

	1-step	2-step	3-step	4-step	5-step	6-step	9-step	average
RMSE (MW)								
BP	42.5541	318.43	509.5931	552.618	499.7061	664.5697	591.7251	454.17
SVM	28.1785	66.3899	120.554	171.7443	224.7633	270.0486	584.321	209.43
SAE_BP	28.3197	44.9789	88.0719	108.9562	143.1312	353.7017	220.7056	141.12
MAE (MW)								
BP	30.3281	246.016	322.2614	435.8386	421.0345	495.5854	472.6068	346.24
SVM	19.887	52.0355	98.4002	147.3712	197.3102	239.0523	490.0291	177.73
SAE_BP	20.5644	33.322	70.5791	81.9918	116.3902	286.1584	166.1055	110.73
MAPE（%）								
BP	4.654	34.3	46.696	56.067	71.512	59.189	58.873	47.33
SVM	3.39	8.768	16.5	25.102	33.925	41.145	66.36	27.88
SAE_BP	3.472	5.639	10.94	12.164	19.008	36.859	23.604	15.96

TABLE 7. Forecasting performance for three-day multi-step wind power forecasting (MAPE).

	1-step	2-step	3-step	4-step	5-step	6-step	9-step	average
SAE_BP	4.81	8.14	9.10	14.20	16.98	19.35	30.45	14.71

the steps. (2) Along with the increase of time steps, the forecasting accuracy for all models decreases. For example, the approximating ability for real wind power of the 3-step ahead forecasting curve is lower than that of the 1-step ahead forecasting curve. This phenomenon (generally referred as error transmission) is completely natural because there are more forecasting wind powers in the reference vector of the 3-step ahead forecasting; (3) for each concave and convex point of the real power curve, each model has an unsatisfactory performance.

Table 6 presents the values for the criteria that evaluate the accuracy of the SAE_BP and other models under several time steps. The first column indicates the three models and the remaining columns represent the error value under each time step. It is obvious that the forecasting performance decreases when the number of forecasting steps increases. The improvements for the proposed SAE_BP are more obvious than BP over the entire forecast horizon. The result illustrates that the SAE is more effective in feature extraction and is more efficient in regressions.

From Table 6, the MAPE of the proposed approach has an average value of 15.96%, while the averages for the BP and the SVM are 47.33% and 27.88%, respectively. The improvement in the average MAPE of the SAE_BP method with respect to the other two models is 66.27% and 42.75% as calculated by formula (20), which are significant under the MAE and RMSE criteria.

$$I_{improvement} = \frac{|e_{other\ model} - e_{proposed}|}{e_{other\ model}} \times 100\% \qquad (20)$$

Furthermore, although the SAE_BP model is slightly worse than the SVM models in 1-step ahead forecasting, the results with more forecasting steps strongly demonstrate the efficiency of the proposed SAE_BP model. For example, the MAPE of the SAE_BP (5.639%) for the 2-step ahead forecasting is lower than that of the BP (28.66%) and SVM (3.129%). Additionally, the MAPE of the SAE_BP (23.604%) for the 9-step ahead forecasting is significantly lower than that of the BP (32.27%) and the SVM (42.76%).

Table 3-7 shows the forecast results of the SAE_BP model forecasting wind power values for the three days from July 12, 2014 to July 14, 2014. From the results, it can be seen that the three-day average accuracy is approximately equal to one-day average prediction accuracy, thus the accuracy of the model in responding to various weather events is still proved.

In general, the SAE_BP method enhances forecasting accuracy. Moreover, the error of the proposed method has been relatively stable as forecasting step grows. Hence, the SAE_BP model is effective and meaningful for short-term wind power forecasting.

IV. CONCLUSION

A high accuracy forecasting method is critical for wind energy generation and integration. In this paper, a multi-layer deep neural network for wind power forecasting is established based on Stacked Auto Encoders. The network parameters are gradually trained by the pre-training process and the fine-tuning process using the back propagation algorithm. To further improve the forecast performance, the network architecture is optimized by the PSO algorithm. The simulation results convincingly illustrate the efficiency of the proposed SAE_BP method in short-term multi-step ahead wind power forecasting. Compared with the BP and SVM methods, the forecasting error of the proposed method in most cases achieves the highest accuracy and the performance is relatively stable with the increase of forecasting steps.

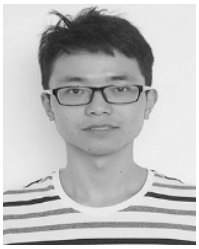
Deep learning is a new research topic, and limited research has focused on its application in regressions. Although the proposed method achieves superior performance than existing methods, our work could still be considered as a preliminary attempt of applying deep learning in regressions. It may be noted that our method can be improved, which will be addressed in our future work.

REFERENCES

[1] A. D. Şahin, "Progress and recent trends in wind energy," *Prog. Energy Combustion Sci.*, vol. 30, no. 5, pp. 501–543, 2004.
[2] B. Ernst *et al.*, "Predicting the wind," *IEEE Power Energy Mag.*, vol. 5, no. 6, pp. 78–89, Nov./Dec. 2007.
[3] I. J. Ramirez-Rosado, L. A. Fernandez-Jimenez, C. Monteiro, J. Sousa, and R. Bessa, "Comparison of two new short-term wind-power forecasting systems," *Renew Energ*, vol. 34, no. 7, pp. 1848–1854, 2009.
[4] C. Monteiro, R. Bessa, V. Miranda, A. Botterud, J. Wang, and G. Conzelmann, "Wind power forecasting: State-of-the-art 2009," Argonne Nat. Lab., Lemont, IL, USA, Tech. Rep. ANL/DIS-10-1 (2009), 2009.
[5] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," *Renew. Sustain. Energy Rev.*, vol. 13, no. 4, pp. 915–920, 2009.
[6] I. G. Damousis, M. C. Alexiadis, J. B. Theocharis, and P. S. Dokopoulos, "A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 352–361, Jun. 2004.
[7] J. L. Torres, A. García, M. De Blas, and A. De Francisco, "Forecast of hourly average wind speed with ARMA models in Navarre (Spain)," *Sol Energy*, vol. 79, no. 1, pp. 65–77, 2005.
[8] L. Jian, Y. Zhao, Y.-P. Zhu, M.-B. Zhang, and D. Bertolatti, "An application of ARIMA model to predict submicron particle concentrations from meteorological factors at a busy roadside in Hangzhou, China," *Sci. Total Environ.*, vol. 426, pp. 336–345, Jun. 2012.
[9] M. C. Mabel and E. Fernandez, "Analysis of wind power generation and prediction using ANN: A case study," *Renew. Energy*, vol. 33, no. 5, pp. 986–992, 2008.
[10] S. Han, J. Li, and Y. Liu, "Tabu search algorithm optimized ANN model for wind power prediction with NWP," *Energy Procedia*, vol. 12, no. 39, pp. 733–740, 2011.
[11] P. Ramasamy, S. S. Chandel, and A. K. Yadav, "Wind speed prediction in the mountainous region of India using an artificial neural network model," *Renew. Energy*, vol. 80, pp. 338–347, Aug. 2015.
[12] J. P. S. Catalao, H. M. I. Pousinho, and V. M. F. Mendes, "An artificial neural network approach for short-term wind power forecasting in portugal," *Eng. Intell. Syst.*, vol. 17, no. 1, pp. 5–11, 2009.
[13] G. Li and J. Shi, "On comparing three artificial neural networks for wind speed forecasting," *Appl. Energy*, vol. 87, no. 7, pp. 2313–2320, 2010.
[14] R. Iqdour and A. Zeroual, "A rule based fuzzy model for the prediction of daily solar radiation," in *Proc. Int. Conf. Ind. Technol.*, Hammamet, Tunisia, Dec. 2004, pp. 1482–1487.
[15] G. Capizzi, F. Bonanno, and C. Napoli, "A wavelet based prediction of wind and solar energy for long-term simulation of integrated generation systems," in *Proc. Int. Symp. Power Electron., Electr. Drives, Autom. Motion*, Pisa, Italy, Jun. 2010, pp. 586–592.
[16] C. JPS, P. HMI, and M. VMF, "An artificial neural network approach for short-term wind power forecasting in Portugal," *Int. J. Eng. Intell. Syst. Elect. Eng. Commun.*, vol. 17, no. 1, pp. 5–11, 2009.
[17] M. A. Mohandes, T. O. Halawani, S. Rehman, and A. A. Hussain, "Support vector machines for wind speed prediction," *Renew. Energy*, vol. 29, no. 6, pp. 939–947, 2004.
[18] J. Zhou, J. Shi, and G. Li, "Fine tuning support vector machines for short-term wind speed forecasting," *Energy Convers. Manage.*, vol. 52, no. 4, pp. 1990–1998, 2011.
[19] K. Chen and J. Yu, "Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach," *Appl. Energy*, vol. 113, no. 6, pp. 690–705, 2014.
[20] A. I. Belousov, S. A. Verzakov, and J. von Frese, "A flexible classification approach with optimal generalisation performance: Support vector machines," *Chemometrics Intell. Lab. Syst.*, vol. 64, no. 1, pp. 15–25, 2002.
[21] H. Liu, H.-Q. Tian, D.-F. Pan, and Y.-F. Li, "Forecasting models for wind speed using wavelet, wavelet packet, time series and artificial neural networks," *Appl. Energy*, vol. 107, pp. 191–208, Jul. 2013.
[22] N. Chen, Z. Qian, and X. Meng, "Multistep wind speed forecasting based on wavelet and Gaussian processes," *Math. Problems Eng.*, vol. 2013, no. 26, pp. 589–606, 2013.
[23] 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.
[24] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning—ICANN*. Berlin, Germany: Springer, 2011, pp. 44–51.
[25] Y. Lecun and C. Cortes. (2010). *The MNIST Database of Handwritten Digits*. Accessed: Sep. 10, 2014. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[26] B. Kwolek, "Face detection using convolutional neural networks and gabor filters," in *Artificial Neural Networks: Biological Inspirations—ICANN*. Berlin, Germany: Springer, 2005, pp. 551–556.
[27] S. Sukittanon, A. C. Surendran, J. C. Platt, and C. J. C. Burges, "Convolutional networks for speech detection," in *Proc. Interspeech*, 2004, pp. 1077–1080.
[28] S. Rifai *et al.*, "Higher order contractive auto-encoder," in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Germany: Springer, 2011, pp. 645–660.
[29] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Nov./Dec. 1995, pp. 1942–1948.
[30] *All Island—System Demand*. Accessed: Sep. 10, 2014. [Online]. Available: http://http://www.eirgridgroup.com/how-the-grid-works/system-information/

RUNHAI JIAO received the Ph.D. degree in computer science from Beihang University in 2008. He is currently an Associate Professor with North China Electric Power University. He has also been a Visiting Scholar at the Galvin Center for Electricity Innovation, Illinois Institute of Technology. His research interest is data mining and its application into power system. He has published over 20 papers in the international journals and conferences.

XUJIAN HUANG is currently pursuing the master's degree in computer technology with North China Electric Power University. He joined the Research on Standard System of the Data Model for the End Equipment of the Sale Distribution Network Based on IEC-CIM Project supported by the State Grid Corporation of China. His current research interest includes deep learning in non-intrusive load monitoring.



XUEHAI MA is currently pursuing the master's degree with North China Electric Power University. His research interest includes data mining and load forecasting based on deep learning.

LIYE HAN received the master's degree in computer technology from North China Electric Power University in 2016. Her research interests include machine learning and load forecasting.

WEI TIAN received the B.S. degree in mathematics from Shandong University in 2001, the M.S. degree in systems engineering from Xi'an Jiaotong University, China, in 2004, and the Ph.D. degree from the Illinois Institute of Technology (IIT), Chicago, in 2011. He is currently a Visiting Scientist at the Robert W. Galvin Center for Electricity Initiative, IIT. He is a Senior Energy Development Engineer with Willdan Inc. His research interests include power system restructuring and long-range planning.