Yuancheng Li [1,2], Ruixian Yang [2]

State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources(North China Electric Power University) (1), North China Electric Power University (2)

A hybrid algorithm combining auto-encoder network with Sparse Bayesian Regression optimized by Artificial Bee Colony for short-term Wind Power Forecasting

Abstract. *To forecast the short-term wind power precisely, this paper proposes a hybrid strategy which consists of a nonlinear dimensionality reduction component by auto-encoder network and a forecasting component based on Sparse Bayesian Regression optimized by Artificial Bee Colony Optimization. The proposed model can predict wind power curve per hour with a lead time of 3hours. Finally, an experiment is conducted to test the effectiveness of the forecasting model based on the detailed data from a wind farm in China.*

Streszczenie. *W artykule zaproponowano hybrydową metodę przewidywania krzywej prędkości wiatru w okresie kolejnej godziny. Algorytm bazuje na nieliniowej redukcji wymiarowości przez sieć auto-enkoderową (sztuczną sieć neuronową) oraz na elemencie przewidującym, opartym na rzadkiej regresji Bayesa (ang. Sparse bayesian Regression) zoptymalizowanej metodą sztucznej koloni pszczół. (**Krótkoterminowe przewidywanie energii wiatru przez algorytm hybrydowy – sieć auto-enkoderowa oraz regresja Bayesa SBR zoptymalizowana metodą sztucznej kolonii pszczół**).*

Keywords: Short-term wind power prediction; Auto-encoder network (AEN); Sparse Bayesian Regression (SBR); Artificial Bee Colony (ABC).
Słowa kluczowe: krótkoterminowa predykcja watru, sieć auto-enkoderowa, SBR, sztuczna kolonia pszczół

1. Introduction

Currently, since the energy and environmental problem has become so urgent to influence the development of human survival and social economy, it is of great importance to guarantee the realization of social and economic sustainable development by supplying secure and stable energy and utilizing the high efficiency and clean energy [1]. Wind power, as one kind of clean and renewable energy sources, has been gradually concerned around the world. As a major energy consumer, China has paid cosmopolitan attention to adjusting the energy structure and alleviating the environmental pollution. Besides, it makes great efforts to the utility of renewable energy, including wind power, in particular of its development and utilization [2].

With characters like strong randomness and long-term inaccurate prediction, wind power confronts special difficulty in power quality and power system operation. Good prediction methods are urgently required to resolve the relevant problems when wind energy is integrated into the power system.

Many researchers have taken the factors influencing wind power into consideration, and classified prediction methods into two groups. One is based on analysis of historical wind power data and the other is on the basis of numerical weather prediction (NWP) data [3]. The prediction method has been enhanced from traditional statistical methods to the artificial intelligence methods, especially the hybrid methods, which attract more and more researchers' attention [4, 5]. Taking advantage of the simplest statistical models, the persistence approach has been proven to be a useful approximation for short-term wind power forecasting. They surpass many other models and have been widely used in practice despite the unstable forecasting efficiency [5]. For the recent years, the artificial intelligence methodologies have appropriately been applied to many areas, such as artificial neural networks (ANN) [6-7], support vector machine [8] and some hybrid algorithms [9-11]. A multi-layer feed-forward neural network (MFNN) is proposed to forecast wind power and speed in time-scales which can vary from a few minutes to an hour and is trained by simultaneous perturbation stochastic approximation (SPSA) algorithm [12]. By considering more factors as the inputs for the model, it would get more accurate results. Paper [13] proposes a support vector machine (SVM)-based model for wind power forecasting, which firstly predicts the wind speed, and then predicts the wind power through using the power-wind speed characteristics of the wind turbine generators. They exert exhaustive searches to find the optimal parameters of SVM. The search process is time-consuming and can be improved by using some algorithms. For the past few years, there are some optimal algorithms which are used to search the best parameters of a model, such as genetic algorithm (GA) [14], differential evolution (DE) algorithm [15], Particle Swarm Optimization (PSO) [16] and so on. In this paper, artificial bee colony (ABC) algorithm is selected to find the optimal parameters of the model based on foraging behavior of honey bees. The selected algorithm is more effective compared with other existing algorithms including GA, PSO, differential evolution algorithm (DE) on many benchmark functions [17-19].

Even though the existing methods have gotten definitely improved over the years, more accurate forecast methods are still under great demand. In this paper, a new wind power forecasting strategy is proposed and its efficiency is exhibited by several experiments composed of auto-encoder network, Artificial Bee Colony and Sparse Bayesian Regression. Various factors which affect the wind power and the relevant historical data are taken as the inputs to make predictable process more precise. As a result, the auto-encoder network solves many issues and reduces the input dimensions. It can filter out its irrelevant and redundant features and select the most prominent candidate inputs for the proposed forecasting model. Besides, we utilize the Sparse Bayesian Regression (SBR) model whose parameters are optimized by the artificial bee colony to predict the power. By incorporating Gaussian process, Bayesian-theorem and automatic relevance determination prior, the method can achieve sparsity in a probabilistic Bayesian learning framework. The SBR algorithm can achieve an accurate prediction which utilizes much fewer functions than the comparable SVM algorithm and offer a number of additional advantages [20-22]. The model usually selects parameters randomly in a given range by empirical/common experience, but it is difficult to

find the optimal parameters. In this paper, artificial bee colony optimization is applied to find the optimal parameters. It has superior learning capability and can avoid the over-fitting and trapping in local minima problem.

The rest of this paper is organized as follows. The proposed prediction model is introduced in Section 2. In this section, the auto-encoder network, Artificial Bee Colony, Sparse Bayesian Regression, and the process of the model are presented respectively. Simulations and results are discussed in Section 3. Finally, conclusions are given in Section 4.

2. The proposed prediction strategy

The structure of the proposed wind power prediction strategy is shown in Fig. 1. Briefly, the proposed model is composed of a reducing dimension component and an optimized forecasting model, which will be introduced in Sections 2.1 and 2.2, respectively.

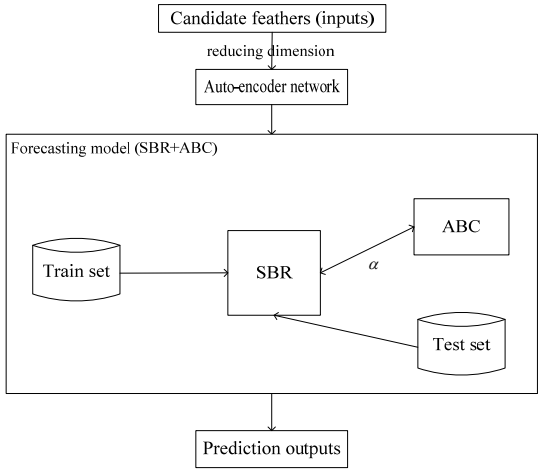

Fig 1. Structure of the proposed wind power forecast strategy

2.1 The reducing dimension section

Reducing dimension, i.e. features extraction, which aims at extracting certain characteristics from the original data, plays a key role in determining the performance of the predict strategy. Improper feature extraction approach will lead to poor regression in the model.

Wind power outputs can be seen as a nonlinear mapping function of several exogenous meteorological variables and its past values, such as wind speed, wind direction, temperature, humidity and so on. To predict the future power, denoted as P(t), we need to consider the past related factors. The output power and the model inputs can be constructed as follows:

$$P(t)=\{V(t),V(t-1),...,V(t-n),D(t),D(t-1),...,D(t-n),\\ T(t),T(t-1),...,T(t-n),P(t-1),...,P(t-n)\} \tag{1}$$

Where $V(t)$, $D(t)$, $T(t)$, $P(t)$ represent wind speed，wind direction, temperature and wind power at time t. In equation (1), $P(t-1),...,P(t-n)$ are the historical values of wind power and the same goes for the historical wind speed, wind direction and temperature. In addition, n indicates the order of back shift for the candidate features. These features should be considered as many as possible, but a compromise is always necessary to avoid too many candidate features. Because not all the candidate features have the same effect on the output, we take a method called auto-encoder network to select the most important factors.

The auto-encoder network is proposed by G.E.Hinton and R.R.Salakhutdinov in 2006 and they have used the method in handling images. The process of transforming the high-dimensional data into a low-dimensional involves an adaptive, multilayer encoder network and a corresponding decoder network [23]. It is not so easy to optimize the weights in nonlinear auto-encoders which contain more than one hidden layers. If the initial weights are large, auto-encoders maybe only find weak local minima while it is infeasible to train auto-encoders with many hidden layers of the small initial weights. So, to get the low-dimension data, we need pre-training an unrolling and fine-tuning network. The pre-training process consists of an independent restricted Boltzmann machine (RBM) which is an especial connection of Boltzmann machine (BM), and the RBM has one hidden and one visible layer [24].

The weight adjustment formula for BM is presented as follows:

$$w_{ij}(t+1)=w_{ij}(t)+\Delta w_{ij}=w_{ij}(t)+\frac{\eta}{T}(\langle v_i h_j\rangle^{+}-\langle v_i h_j\rangle^{-}) \tag{2}$$

Where $w_{ij}(t)$ represents the connection weight which is between neuron *i* and *j* in step t, η is a learning rate, T is a network temperature, $\langle v_i h_j\rangle^{+}$ is a positive average association, and $\langle v_i h_j\rangle^{-}$ is a reverse average association. In RBM, average association is the multiply of visible unites outputs and hidden units outputs. Let η and T integrate into coefficient $\mathcal{E}$, then the weight adjustment formula is given below:

$$w_{ij}(t+1)=w_{ij}(t)+\Delta w_{ij}=w_{ij}(t)+\varepsilon(\langle v_i h_j\rangle^{+}-\langle v_i h_j\rangle^{-}) \tag{3}$$

Where $\mathcal{E}$ denotes the iterations step size.

Fig.2. shows the weights training of RBM. Let t=0, update the hidden neurons state, subsequently update the visible neurons state and get the reconstruction data, then make t=1update the hidden units state with reconstruction and complete one RBM training session. Finally, repeat the above-mentioned training again from t=0, we can get the corresponding weights after training a RBM. Hence, we have:

$$\Delta w_{ij}=\varepsilon(\langle v_i h_j\rangle^{0}-\langle v_i h_j\rangle^{1}) \tag{4}$$

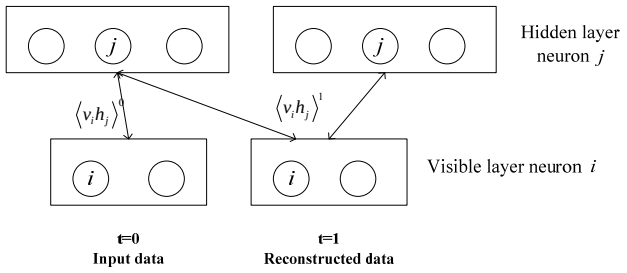

Fig. 2.The weights training of Restricted Boltzmann Machine

According to the above process several times, we can get the latest positive training output as the input of the next RBM for training. After pre-training, the model is unfolded to get encoder and decoder networks which are initialized by the equivalent weights obtained from the above pre-training. Then in the fine-tune stage, take back-propagation algorithm that mainly uses cross entropy as an objective function to fine-tune the auto-encoder weights based on the pre-training obtained weights. Cross entropy measures the difference between two sorts of probability distribution, and it is not negative. The more similar of the two distributions

are, the smaller the value is. The original cross entropy is defined as:

$$D(p\,||\,q)=\sum_{x} p(x)\log\frac{p(x)}{q(x)} \tag{5}$$

Where *x* is an extraneous variable, *q(x)* is known as probability distribution, *p(x)* is estimated as probability distribution.

When estimate *p(x)* with *q(x)*, minimize cross entropy *D(p||q)* by adjusting *p(x)*. Back-propagation algorithm cross entropy function, which is used to adjust auto-encoder networks weights, shown as follows:

$$H_m=-\sum_{i=1}^{m}[t_i\log y_i+(1-t_i)\log(1-y_i)] \tag{6}$$

Where t_i is the objective probability distribution, y_i is the real probability distribution.

The whole network training purpose is to adjust relevance weights to obtain the minimal cross entropy function value. We have the following weight adjusting formula.

$$\Delta w_{ij}=-\alpha\frac{\partial H_m}{\partial w_{ij}} \tag{7}$$

$$\frac{\partial H_m}{\partial w_{ij}}=\frac{\partial H_m}{\partial net_i}\frac{\partial net_i}{\partial w_{ij}} \tag{8}$$

$$\frac{\partial net_i}{\partial w_{ij}}=O_j \tag{9}$$

$$\frac{\partial H_m}{\partial net_i}=\frac{\partial H_m}{\partial O_i}\frac{\partial O_i}{\partial net_i}=\frac{\partial H_m}{\partial O_i}O_i(1-O_i) \tag{10}$$

Set the output layers $O_i=y_i$, and then we have another weight adjusting formula as follows:

$$\Delta w_{ij}=-\alpha\frac{\partial H_m}{\partial w_{ij}}=\alpha(t_i-y_i)O_j \tag{11}$$

We can finish the networks fine-tune training according to the above weight adjusting formula.

2.2 The optimized forecasting model

From the above reducing dimension, we take SBR model whose parameters are optimized by ABC to predict the wind power.

2.2.1 The Sparse Bayesian regression model

Given a training set $\{\mathbf{x}_n,t_n\}_{n-1}^{N}$, we hope to learn a model of the targets relied on the inputs, with the objective to make accurate predictions of t for previously unseen values of **X**. Considering the functions are scalar, we follow the standard probabilistic formulation and assume that the targets are samples from the model with additive noise [25]:

$$t_n=y(\mathbf{x}_n;\mathbf{w})+\varepsilon_n \tag{12}$$

Where $\{x_n\}_{n-1}^{N}$ is an input vector, $\{t_n\}_{n-1}^{N}$ is the corresponding target, and ε_n is an independent sample from some noise process which is further assumed to be mean-zero Gaussian with variance σ^2. **W** is the adjustable parameter (or 'weight'), and the objective is to estimate good values for those parameters. Thus, $p(t_n\,|\,\mathbf{x})=N(t_n\,|\,y(\mathbf{x}_n),\sigma^2)$, where the notation specifies a Gaussian distribution over t_n with mean $y(\mathbf{x}_n)$ and variance σ^2. The function $y(\mathbf{x}_n)$ is the basis function with the kernel parameterized by the training vectors: $\phi_i(\mathbf{x})=K(\mathbf{x},\mathbf{x}_i)$. Due to the assumption of independence of t_n, the likelihood of the complete data set can be written as

$$p(\mathbf{t}\,|\,\mathbf{w},\sigma^2)=(2\pi\sigma^2)^{-N/2}\exp\{-\frac{1}{2\sigma^2}\|t-\mathbf{\Phi}\mathbf{w}\|^2\} \tag{13}$$

Where $\mathbf{t}=(t_1...t_N)^T$, $\mathbf{w}=(w_0...w_N)^T$ and $\mathbf{\Phi}$ is the $N\times(N+1)$ matrix with $\mathbf{\Phi}=[\phi(x_1),\phi(x_2),...,\phi(x_N)]^T$, where $\phi(x_n)=[1,K(x_n,x_1),K(x_n,x_2),...,K(x_n,x_N)]^T$.

A prior probability distribution is defined for simpler functions by making the popular choice of a zero-mean Gaussian prior distribution over $\mathbf{w}$:

$$p(\mathbf{w}\,|\,\alpha)=\prod_{i=0}^{N}N(w_i\,|\,0,\alpha_i^{-1}) \tag{14}$$

Where α are the vector of N + 1 hyper-parameters. Importantly, there is an individual hyper-parameter independently associated with each weight, moderating the strength of the prior thereon.

Having defined the prior, Bayesian inference proceeds by computing, by Bayesian rule, the posterior over all unknowns given the data:

$$p(\mathbf{w},\alpha,\sigma^2\,|\,\mathbf{t})=\frac{p(\mathbf{t}\,|\,\mathbf{w},\alpha,\sigma^2)\,p(\mathbf{w},\alpha,\sigma^2)}{p(\mathbf{t})} \tag{15}$$

Then, given a new test point x_*, predictions are made for the corresponding targets t_*, in terms of the predictive distribution:

(16)

$$p(t_*\,|\,\mathbf{t})=\int p(t_*\,|\,\mathbf{w},\alpha,\sigma^2)p(\mathbf{w},\alpha,\sigma^2\,|\,\mathbf{t})d\mathbf{w}\,d\alpha\,d\sigma^2$$

The posterior distribution over the weights is thus given by

(17)

$$p(\mathbf{w}\,|\,\mathbf{t},\alpha,\sigma^2)=\frac{p(\mathbf{t}\,|\,\mathbf{w},\sigma^2)\,p(\mathbf{w}\,|\,\alpha)}{p(\mathbf{t}\,|\,\alpha,\sigma^2)}$$
$$=(2\pi)^{-(N+1)/2}\,|\textstyle\sum|^{-1/2}\exp\{-\frac{1}{2}(\mathbf{w}-\mu)^T\textstyle\sum^{-1}(\mathbf{w}-\mu)\}$$

At the convergence of the hyper-parameter estimation procedure, we make predictions based on the posterior distribution over the weights, conditioned on the maximizing values α_{MP} and σ^2_{MP}. Here, ABC is used to find the optimal hyper-parameter α_{MP}, the concrete process will be stated in the next session. Then, we can calculate the predictive distribution, from (16), for a new datum x_* using (17):

(18)

$$p(t_*\,|\,\mathbf{t},\alpha_{MP},\sigma^2_{MP})=\int p(t_*\,|\,\mathbf{w},\sigma^2_{MP})p(\mathbf{w}\,|\,\mathbf{t},\alpha_{MP},\sigma^2_{MP})d\mathbf{w}$$

Where the posterior covariance and mean are respectively:

$$\sum=(\sigma^{-2}\Phi^T\Phi+\mathbf{A})^{-1} \tag{19}$$

$$\mu=\sigma^{-2}\sum\Phi^T\mathbf{t} \tag{20}$$

With $A=diag(\alpha_0,\alpha_1,...,\alpha_N)$.

Since both terms in the integrand are Gaussian, this is readily calculated, giving:

(21) $$p(t_* \mid \mathbf{t}, \alpha_{MP}, \sigma^2_{MP}) = N(t_* \mid y_*, \sigma^2_*)$$

With $y_* = \mu^T \phi(x_*)$ and $\sigma_* = \sigma^2_{MP} + \phi(x_*)^T \sum \phi(x_*)$.

So the predictive mean is intuitively $y(x_*; \mu)$, or the basis functions weighted by the posterior mean weights, many of which will typically be zero. The predictive variance (or 'error-bars') comprises the sum of two variance components: the estimated noise on the data and that due to the uncertainty in the prediction of the weights. In practice, then, we may thus choose to set our parameters $\mathbf{w}$ to fixed values μ for the purposes of point prediction, and retain $\sum$ if required for computation of error bars.

2.2.2 Using Artificial Bee Colony (ABC) optimizing the hyper-parameter α of the SBR

Artificial Bee Colony (ABC) optimization algorithm simulates the intelligent foraging behavior of honey bee swarms based upon stochastic optimization algorithm [26]. The ABC algorithm includes three bee groups and many other food sources. The bees could be distinguished into three sorts in accordance with their responsibility such as onlookers, scouts, and employee. A bee waiting on the dance area for making decision to choose a food source is called onlooker and one going to the food source is named as employed bee. The other kind of bee is scout bee who carries out random searches for discovering new sources. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source stands for the quality (or fitness) of the associated solution. An onlooker bee selects one solution and tries to improve it. When the network consists of n cluster-head sensors, the bees fly in the search space with n dimensions. The ABC imitates a population of bees to find the cluster-heads. The detailed implementation for finding an optimal parameter of SBR procedures of the algorithm is given below:

Step1: Randomly generate an initial population of N food sources with a range of boundaries of the variables.

(22) $$x_{ij} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min})$$

Where *i*=1…*N*, *j*=1…*D*. *N* is the number of food source and *D* is the number of optimization variables.

Step2: Evaluate the fitness of each food source (i.e. calculate the nectar amount) according to Eq.(23). Here the food source *I* is applied to calculate the prediction value $\mathbf{y}'$ of solution x_i according to Eq.(16).

(23) $$fitness_i = \left| \frac{\mathbf{y} - \mathbf{y}'}{\mathbf{y}} \right|$$

Where $\mathbf{y}$ is the actual data and $\mathbf{y}'$ is the predicting data in SBR, $fitness_i$ is the cost value of solutions X_i .

Step3: Each employed bee searches a candidate food source V_i according to Eq. (24). Evaluate the candidate food source and apply greedy selections to select a better one as the new food source.

(24) $$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

Where *j* is a random integer in the range [1, *D*] and $k \in \{1, 2, \ldots, N\}$ is a randomly chosen index different from *i* . ϕ_{ij} is a uniformly distributed real random number in the range [-1, 1].

Step4: Calculate probability values based on the fitness values of the solutions in the population. Each onlooker selects a food source according to Eq. (24) by roulette wheel selection and generates a candidate solution according to Eq. (23).

(25) $$p_i = \frac{fitness_i}{\sum_{i=1}^{N} fitness_i}$$

Step5: Evaluate the candidate food source and select a better one as the new food source according to greedy selection.

Step6: Memorize the best food source position (solution) found so far.

Step7: If the position of a particular food source cannot be improved through the predetermined number of trials 'limit', then select it as an abandoned one. Replace the solutions by a different position that is randomly produced by a scout according to Eq. (22).

Step8: Repeat the procedure from step 3 until the termination criterion is met. When the algorithm is terminated, the position of optimal food source and its nectar amount are the optimal values of the decision variables and objective function for the considered problem.

2.3 The hybrid prediction strategy

We refer to the factors, including the past 24h wind speed, wind direction, temperature and the past wind power, as inputs to predict the future 3h wind power, in that way, the inputs of the model are so numerous, which makes the model very complex. Therefore, we take the auto-encoder network to extract the essential features. Auto-encoders give mappings in both directions between the data and code spaces, and they can be applied to large data sets because both the pre-training and the fine-tuning scale are linear in time and space with the number of training cases. The auto-encoder consists of an encoder with layer size of 103-80-40-20 and a symmetric decoder. The network is trained on 2000 data sets and tested on 148 new data sets. We originally use RBM to initialize the weights. Then we carried out back-propagation to obtain optimal weights. The auto-encoder can learn to encode the data sets that allow almost perfect reconstruction. Fig. 3 displays the entire auto-encoder network. The auto-encoder with three layers functions well in our experiments.

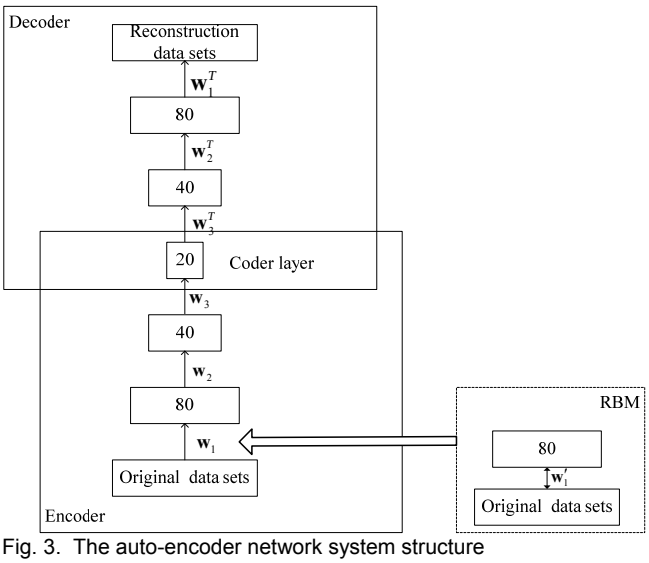

Fig. 3. The auto-encoder network system structure

The SBR model optimized by ABC gets well subsequent to the reducing dimension process. In the process of training model, we use ABC to attain optimal hyper-parameter α , which has stated in section 2.2.2. Then, we pass the obtained optimal hyper-parameters to the SBR for

calculating the output values. The model training phase is terminated under the stopping condition to avoid the over-fitting problem. In the algorithm, the convergence condition depends on whether the error is small enough or not, which is determined by the expected modeling and prediction accuracy. Then, we test the model to predict wind power.

3. Simulations and results discussion

The proposed wind power predicting model is tested on the real data from wind farm in western China. Wind blows randomly and it has an important effect on the power quality and the operation of the power system after the wind gains access to the power grid. So the accuracy of wind power prediction is particularly significant. For the reason that the time needed by the power grid scheduling and resource allocation mainly focus on 0 ~ 3h, we predict hourly wind power with a lead time of 3h.

We choose data of Jan, April, July and Nov in 2009 which typifies the four seasons of a year. The data are separated into two groups, of which one is used for estimation and the other is reserved for model validation. Three quarters of the data in every month are the train set and the rest is the test set, as a consequence, we have four test sets for four seasons. We refer to the recent 24h wind speed, wind direction, temperature, wind power to predict the next 3h wind power. First, we normalize these data into [0, 1] using the formula (26) and then utilize auto-encoder network works to reduce dimension. At last we take the proposed model to predict the wind power.

$$x' = \frac{x_{\max} - x}{x_{\max} - x_{\min}} \tag{26}$$

Where x' is the normalized data, $x_{\max}$ is the maximum among the dataset, $x_{\min}$ is the minimum among the dataset, x is the not normalized data.

The error of prediction is a crucial criterion judging the performance of a method. Here, the mean absolute percentage error (MAPE) is taken. Note that MAPE is a reference of accuracy in the time series model usually expressed as a percentage:

$$MAPE = \frac{1}{n}\sum_{j=1}^{n}\left|\frac{y_d(j) - y_f(j)}{y_d(j)}\right| \tag{27}$$

Where $y_d(j)$ is the actual data, $y_f(j)$ is the predicting data at the time step *j*, and *n* is the number of samples.

A comparative performance result with forecast models of support vector machine (SVR), core vectors machine (CVR) and a hybrid method namely SVR-PSO is depicted in table 1. For the sake of a fair comparison, all methods of Tables 1 use the same training set and test set. SVR and CVR are the two methods which are popular in artificial intelligent area. The results show that the proposed method outperforms other methods for the same time period in all the seasons. The fine-tine results in reducing dimension by auto-encoder networks are shown in fig.4.and the predictions by the proposed strategy and the comparative methods prediction results in winter, spring, summer, and autumn are shown respectively in fig 5-8.

Table 1. Comparative MAPE results

Test set	SVR	CVR	SVR-PSO	Hybrid Strategy
Winter	17.46836	8.37155	10.40742	5.329107
Spring	10.68621	4.512724	15.68683	3.116635
Summer	7.496231	4.458986	10.83847	3.864655
Autumn	50.3021	24.23551	48.06441	12.551755

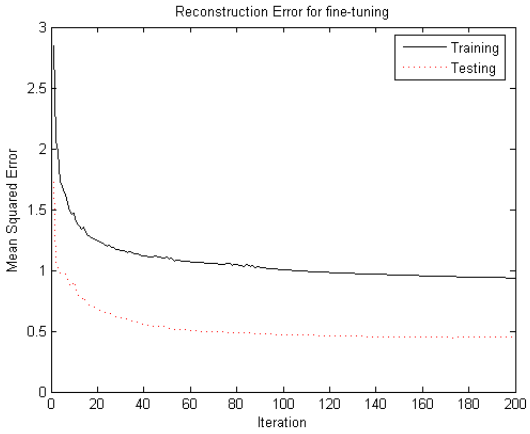

Fig.4. the fine-tine result

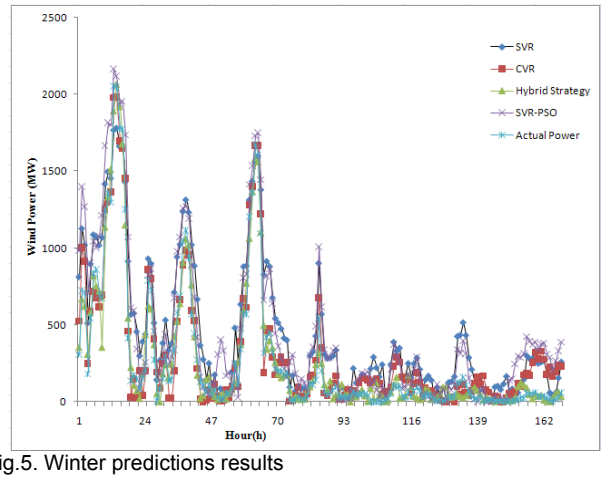

Fig.5. Winter predictions results

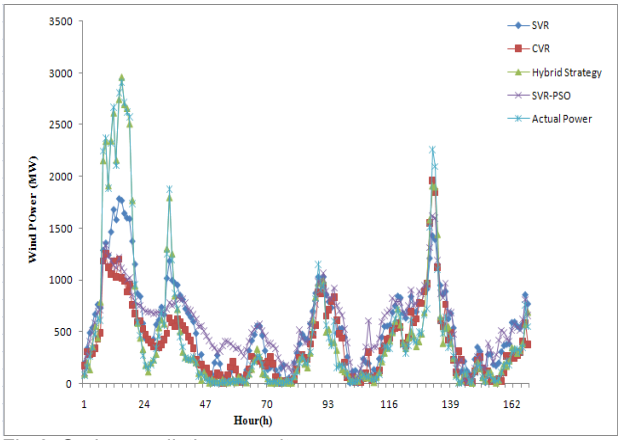

Fig.6. Spring predictions results

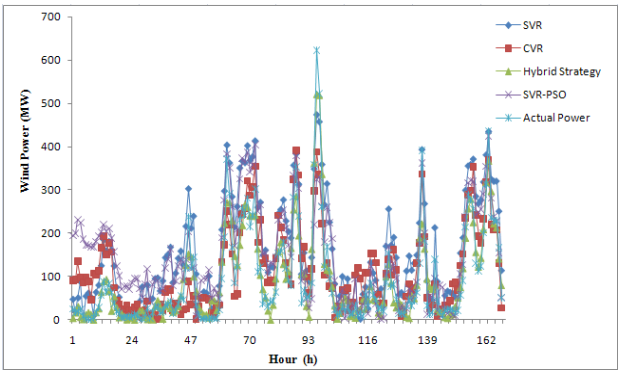

Fig.7. Summer predictions results

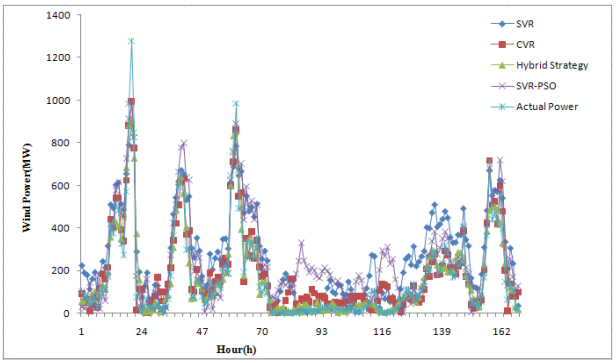

Fig.8. Autumn predictions results

From fig.4, we can see the reconstruction error is very slight by fine-tune, in this way, we select the most important feathers and reduce dimensions without loss of information. The different model prediction MAPE can be obtained from table 1, from which we can see that our proposed hybrid strategy has the best accurate results while SVR has the worse, SVR-PSO models has better prediction results than SVR owing to the PSO gets the optimal parameters of SVR, but it still performs worse than CVR model. From the prediction results we can see that the error in spring and summer are smaller than that in winter and autumn and the forecasting error is the largest in autumn. The reason of this phenomenon lies in the fact that the wind in winter and autumn gets more variables and has larger effect on the power outputs.

4. Conclusions

In this paper, a new wind power forecast strategy is proposed which is composed of an efficient auto-encoder network and an optimized forecasting model. The presented feature selection sections utilize a nonlinear generalization of principal component analysis. It uses an adaptive, multilayer “encoder” network to transform the high-dimensional data into a low-dimensional one and a similar “decoder” network to recover the data. The proposed forecasting model implements spare probability to compute the outputs and the parameters in the process are optimized by ABC. The experimental results show that the proposed hybrid strategy has better performance.

REFERENCES

[1] Costa,A.; Crespo, A. A review on the young history of the wind power short-term prediction [J]. Renewable and Sustainable Energy Reviews 2008, 12(6),1725–1744.
[2] Xie,L.; Liu,J.H; NipunLopli. Wind Integration in Power Systems: Operational Challenges and Possible Solutions. Proceedings of the IEEE, 2011, 214-232.
[3] Foley ,A.M.; Leahy ,P.G.; Marvuglia A.; McKeogh , E.J.Current methods and advances in forecasting of wind power generation [J]. Renewable Energy 2012,37, 1-8.
[4] Wu, Y.K; Hong J.S. A literature review of wind forecasting technology in the world[C]. IEEE Conference on Power Technology ,Lausanne ,2007,7,504-509.
[5] Ma, L.; Luan S.Y.; Jiang C.W. A review on the forecasting of wind speed and generated power [J]. Renewable and Sustainable Energy Reviews 2009, 13(4), 915–920.
[6] Bashir,Z. A. ; El-Hawary M. E. Applying Wavelets to Short-Term Load Forecasting Using PSO-Based Neural Networks. IEEE Transaction on Power System 2009, 1 (24), 20-27.
[7] Catalão ,J.P.S.; Pousinho,H.M.I.; Mendes,V.M.F. Short-term wind power forecasting in Portugal by neural networks and wavelet transform. Renewable Energy 2011, 36, 1245-1251.
[8] Sancho S.S.; Emilio G. O.G. Short term wind speed prediction based on evolutionary support vector regression algorithms. Expert System with Application 2011, 38,4052-4057.
[9] Ajay,S.P.; Devender, S.; Sunil K.S. Intelligent Hybrid Wavelet Models for Short-Term Load Forecasting. IEEE Transaction on Power System 2010, 25(3), 1266-1273.
[10] Amjady, N.; Keynia, F.; Zareipour, F. Wind Power Prediction by a New Forecast Engine Composed of Modified Hybrid Neural Network and Enhanced Particle Swarm Optimization. IEEE Transaction on Sustainable Energy 2011, 2(3), 265-276.
[11] Catalão ,J.P.S.; Pousinho,H.M.I.; Mendes,V.M.F. Hybrid Wavelet-PSO-ANFIS Approach for Short-Term Wind Power Forecasting in Portugal. IEEE Transaction on Sustainable Energy 2011,2(1), 50-59.
[12] Hong,Y.Y; Chang,H.L.; Chiu,C.S. Hour-ahead wind power and speed forecasting using simultaneous perturbation stochastic approximation (SPSA) algorithm and neural network with fuzzy inputs.Energy 2010,35,3870-3876.
[13] Zeng,J.W.; Qiao,W. Support Vector Machine-Based Short-Term Wind Power Forecasting. Power System Conference and Exposition (PSCE),2011,1-8.
[14] Holland ,J.H.. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.
[15] Price,K.V.; Storn,R.M.; Lampinen, J.A. (Eds.).Differential Evolution: A Practical Approach to Global Optimization. Springer Natural Computing Series, 2005.
[16] Liao,R.G.; Zheng,H.B.; Stanislaw Grzybowski,Yang,L.G. Particle swarm optimization-least squares support vector regression based forecasting model on dissolved gases in oil-filled power transformers. Electric Power System Research 2011, 12(81), 2074-2080.
[17] Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 2007, 39,459-471.
[18] Karaboga, D.; Basturk,B. On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing, 2011,8(1),687-697.
[19] Karaboga,D. ; Akay,B. A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation 2009, 214, 108-132.
[20] Yang ,D.P.; Xu L.; Gong,S.P.; Li, H.S.; Gregory D. Peterson. Joint Electrical Load Modeling and Forecasting Based on Sparse Bayesian Learning for the Smart Grid. Conference on Information Sciences and Systems (CISS), 2011, 1-6.
[21] Dimitris, G.; Tzika; Aristidis,C.; Likas. Sparse Bayesian Modeling With Adaptive Kernel Learning. IEEE TRANSACTION ON NEURAL NETWORKS 2009, 20(6) ,926-937.
[22] Qing, D.; Zhao, J.G.; Niu, L.; KeLuo.Regression Based on Sparse Bayesian Learning and the Applications in Electric Systems. Fourth International Conference on Natural Computation,2008, 106-110.
[23] Hinton, G. E.; Salakhutdinov ,R. R.Reducing the dimensionality of data with neural networks. Science 2006, 313, 504–507.
[24] Hopfield,J. J.; Proc. Natl. Acad. Sci. U.S.A. 79, 2554 ,1982.
[25] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. J. Mach. Learn. Res., 2001,1(1), 211–244.
[26] Karaboga, D. An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Computer Engineering Department, Erciyes University, Kayseri, Turkey, (2005).

***Authors**: prof. dr Yuancheng Li, North China Electric Power University, Beijing,102206,China, E-Mails: ycli@ncepu.edu.cn; Ruixian Yang, North China Electric Power University, Beijing,102206,China, E-mail: yrx361@163.com*