Deep Belief Networks with Genetic Algorithms in Forecasting Wind Speed

KUO-PING LIN 1,2, Senior Member, IEEE, PING-FENG PAI 3, Senior Member, IEEE AND YI-JU TING 3

1 Institute of Innovation and Circular Economy, Asia University, 500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
2 Department of Industrial Engineering and Enterprise Information, Tunghai University, No.1727, Sec.4, Taiwan Boulevard, Taichung 40704, Taiwan
3 Department of Information Management, National Chi Nan University, 1 University Rd., Puli, Nantou 54561, Taiwan

Corresponding author: Ping-Feng Pai (e-mail: paipf@ncnu.edu.tw).

This work was financially supported in part by the Ministry of Science and Technology of Taiwan, Republic of China, under Contract Numbers MOST 103-2410-H-260-020, MOST 104-2410-H-260-018, MOST 105-2410-H-260-017-MY2 and MOST 107-2410-H-260-012.

ABSTRACT Wind is one of the most essential sources of clean, renewable energy, and is therefore a critical element in responsible power consumption and production. The accurate prediction of wind speed plays a key role in decision-making and management in wind power generation. This study proposes a model using a deep belief network with genetic algorithms (DBNGA) for wind speed forecasting. The genetic algorithms are used to determine parameters for deep belief networks. Wind speed and weather-related data are collected from Taiwan’s central weather bureau for this purpose. This study uses both time series data and multivariate regression data to forecast wind speed. The seasonal autoregressive integrated moving average (SARIMA) method and the least squares support vector regression for time series with genetic algorithms (LSSVRTSGA) are used to forecast wind speed in a time series, and the least squares support vector regression with genetic algorithms (LSSVRGA) and DBNGA models are used to predict wind speed in a multivariate format. Empirical results show that forecasting wind speed by DBNGA models outperforms the other forecasting models in terms of forecasting accuracy. Thus, the DBNGA model is a feasible and effective approach for wind speed forecasting.

INDEX TERMS Forecast, wind speed, deep belief networks, genetic algorithms, multivariate regression, time series

I. INTRODUCTION

As global population growth, economic development and the progress of science and technology have all rapidly increased, so too has the global demand for energy. This increasing energy demand has resulted in energy shortages and global warming. Wind power, however, is free and clean, and has thus become one of the most popular sources of renewable energy. The accurate prediction of wind speed plays a crucial role in effectively generating, distributing and managing wind power, and, to date, a number of wind speed forecasting models have been proposed [1-5]. Deep learning is an emerging technology that has been applied in many fields, including wind speed prediction. Wang et al. [6] presented a hybrid model including wavelet transform, a deep belief network and spine quantile regression to forecast wind speed. Wind farm hourly data with high-level nonlinear and non-stationary characteristics were predicted by the proposed hybrid model. A trial-and-error method was employed to determine the number of hidden layers and hidden nodes required in a deep belief network for predicting wind speed. The numerical result indicated that the proposed hybrid model outperformed the auto-regressive and moving average models, back-propagation neural networks, and the Morlet wavelet neural network in terms of forecasting accuracy. Hu et al. [7] developed a shared-hidden-layer deep neural network model for predicting wind speed. The numerical outcomes demonstrated that the proposed shared-hidden-layer deep neural network model achieved more accurate forecasting results than models using support vector regression, deep neural networks, or extreme learning machines. Liu et al. [8] designed a hybrid deep-learning model consisting of empirical wavelet transform, long short term memory neural networks and Elman neural networks for forecasting wind speed. The empirical wavelet transform divided original wind speed data...
into low-frequency wind speed data to be processed by the long short term memory neural network, and high-frequency wind speed data to be processed by the Elman neural network. The experiment results showed that the proposed hybrid model outperformed other forecasting techniques in terms of forecasting accuracy. Liu et al. [9] proposed a multistep forecasting model including variational mode decomposition, singular spectrum analysis, long short term memory networks, and an extreme learning machine for predicting wind speed. The variational mode decomposition and the singular spectrum analysis were employed to decompose data and extract data trends; the long short term memory network and the extreme learning machine were used to deal with low-frequency and high-frequency data, respectively. The empirical results revealed that the presented hybrid model could generate more accurate results in forecasting wind speed than the other models. Chen et al. [10] developed a combined model by applying the ensemble learning technique to long short term memory neural networks and a support vector regression machine for wind speed forecasting. Simulation results indicated that the proposed model achieved a better forecasting performance than the autoregressive integrated moving average models, support vector regression, the k-nearest neighbor technique, artificial neural networks, and the gradient boosting regression tree for wind speed forecasting. Liu et al. [11] proposed a hybrid model using the wavelet packet decomposition technique, a convolutional neural network and a long short term memory network for predicting wind speed. The wavelet packet decomposition technique was used to decompose wind speed data into high-frequency and low-frequency data patterns. The convolutional neural network was employed to cope with high-frequency wind speed data, while the low-frequency wind speed data was processed by the convolutional long short term memory network. The numerical results demonstrated that the developed hybrid model outperformed other forecasting models in terms of wind speed prediction accuracy. Li et al. [12] developed a hybrid model for forecasting wind speed. The proposed model included empirical wavelet transform decomposition, a long short term memory network, a regularized extreme learning machine network and inverse empirical wavelet transform. The models performed raw wind speed data decomposition, time series wind speed data forecasting, forecasting error series modeling, final forecasting series integration, and outlier filtering. The numerical results showed that the performance of the presented hybrid model was superior to that of the other seven compared forecasting techniques in terms of forecasting accuracy. Nevertheless, the choice of parameters for deep belief networks significantly influences their forecasting accuracy. Experimental or trial-and-error methods have been adopted to cope with such parameter selection [6,13-16]. In addition, some metaheuristics, including the tabu search algorithm [17], particle swarm optimization [18-20] and genetic algorithms [21-24] have been used for deep learning network parameter selection. In this study, genetic algorithms [25] are used for deep belief network parameter selection for predicting wind speed by weather-related data. The remainder of this paper is arranged as follows. Section 2 presents the methodologies used in this study. The proposed wind speed forecasting framework and numerical results are described in Section 3. Finally, conclusions are drawn in Section 4.

II. METHODOLOGIES
A. THE SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE MODEL
Developed by Box and Jenkins [26] the autoregressive integrated moving average (ARIMA) model is a popular method in coping with time series data. In the ARIMA method, two components, the autoregressive and the moving average; are included; and a future variable value is determined by a linear combination of prior values and errors. Three parameters, the order of the autoregressive (p), the degree of differencing (d), and the order of the moving average model (q), are contained in the ARIMA model. The autoregressive model, AR(p), represents the association between present and past values. For a sequence with variables Y, the values of the prior period are employed to predict the present period value (Yt) and expressed by Eq.(1).

\[ Y_t = C + \sum_{i=1}^{p} \theta_i Y_{t-i} + \epsilon_t - \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \]  

where C is a constant, \( \theta_i \) and \( \theta_i \) are the autoregressive coefficient and the moving average coefficient respectively, and \( \epsilon_t \) is white noise at time t.

The seasonal autoregressive integrated moving average (SARIMA) model [26, 27] is a revised form of the ARIMA model. The SARIMA model was designed for forecasting time series data with trend and seasonal tendency, and uses regular and seasonal differences to remove trends and seasonal presences. A SARIMA model (p, d, q)(P, D, Q)s produces a time series, \( \{Y_t, t = 1, 2. . . k\} \), with a mean M from an ARIMA [26] time series model satisfying Eq. (2):

\[ \Phi_p(L)(1-L)^s \{Y_t-M\} = \Theta_q(L)\epsilon_t \]  

where L is the backward shift operator, and \( \epsilon_t \) is white noise at time t, while p, d, q, P, D and Q are nonnegative integers, the orders p and P are a regular autoregressive operator and a seasonal autoregressive operator, the orders q and Q are a regular moving average operator and a seasonal moving average operator, d and D denote degrees of regular differencing and seasonal differencing, s indicates the seasonal interval, \( \Phi_p(L)=(1-\theta_1L-\theta_2L^2-\cdots-\theta_pL^p) \) is the regular autoregressive operator with order p, and \( \Theta_q(L)=(\theta_1L+\theta_2L^2+\cdots+\theta_qL^q) \) is the regular moving average operator with order q. In addition, \( \theta(L)=(1-\theta_1L-\theta_2L^2-\cdots-\theta_pL^p) \) is the seasonal autoregressive operator with order P, and \( \Theta_q(L) \) is the seasonal moving average operator with order Q. The degrees of differencing, d and D, are determined in terms of eliminating the tendency and seasonality of data. The autocorrelation function and the
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2929542, IEEE Access

partial autocorrelation function of the differenced series are employed to determine the values of \( \rho, P, q \) and \( Q \). In general, a four-step procedure involving identification, estimation, validation, and forecasting, is performed to find suitable parameters and to make a forecast in the SARIMA model [26-29].

**B. THE LEAST SQUARE SUPPORT VECTOR REGRESSION MODEL**

Revised from the support regression model [30-32] by decreasing computation load, the least square support vector regression model [33] is a popular and powerful technique for dealing with forecasting problems in both time series and multivariate regression formats. By introducing a least squares approach to the support vector regression model for a training data set \( \{X_i, Y_i\}, i = 1, \ldots, N \) in order to formulate a regression problem, the LSSVR model can be depicted as the following optimization problem [33, 34]:

\[
\text{Min } f = \frac{1}{2}w^T w + \frac{1}{2} \sum_{i=1}^{N} \varepsilon_i^2 \tag{3}
\]

subject to

\[
y_i = w^T \cdot \rho(X_i) + b + \varepsilon_i, i = 1, \ldots, N
\]

where \( w \) is the weight vector of the same dimension as the feature space, \( \rho \) is the regularization constant signifying the trade-off between the empirical error and the flatness of the function, \( N \) is the number of data points, \( e_i \) is the error training error of \( X_i, Y_i \) and \( Yi \) are the input and output data points for the \( i \)th pattern data, \( \rho(X) \) is a nonlinear mapping function, and \( b \) is the bias. By the Lagrange multipliers technique, the Lagrangian form of Eq. (3) is written as:

\[
\mathcal{L}(w, b, \varepsilon, \alpha) = \frac{1}{2}||W||^2 + \frac{1}{2} \sum_{i=1}^{N} \varepsilon_i^2 - \sum_{i=1}^{N} \alpha_i (y_i - (w^T \cdot \rho(X_i) + b + \varepsilon_i)) \tag{4}
\]

where \( \alpha_i \) are the Lagrange multipliers. Then, the partial derivative of Eq. (4) is performed with respect to \( w, b, \varepsilon \), and \( \alpha_i \), when all derivatives are equal to zero according to the Karush–Kuhn–Tucker conditions [35-37]. Using the least squares method to resolve the linear equations, the LSSVR model can be represented as Eq. (5):

\[
Y(X) = \sum_{i=1}^{N} \alpha_i \cdot \rho(X_i) \cdot X + b \tag{5}
\]

With respect to Mercer’s principle [38], the following kernel function is defined:

\[
K(X, X_i) = \rho(X)^T \cdot \rho(X_i) \tag{6}
\]

In this study, the radial basis function with the kernel width \( \sigma \) indicated by Eq.(7) serves as a kernel function.

\[
K(X, X_i) = \exp(-\frac{||X-X_i||^2}{2\sigma^2}) \tag{7}
\]

In this investigation, suitable LSSVR model parameters are provided by genetic algorithms [25, 39].

**C. DEEP BELIEF NETWORKS**

A deep belief network [40] consists of several stacked restricted Boltzmann machines (RBMs), which are the main components of a deep belief network. The RBM was revised from the original Boltzmann machine BM [41] with node connections both within layers and between layers. The RBM only allows links between nodes in adjacent layers. An RBM consists of two layers: a visible layer and a hidden layer. Each node in the visible layer is linked to all hidden nodes without directions. Deep belief network learning strategies include unsupervised learning in the pre-training stage, and supervised learning in the fine-tuning stage. The main task of unsupervised learning is to select appropriate initial parameters, including biases and weights, for the supervised learning using only independent variables. Thus, the pre-training phase rebuilds training samples by adjusting parameters in order to maximize the likelihood estimation. Based on the initial parameters provided by the pre-training phase, the supervised learning phase can further tune biases and weights efficiently. Figure 1 illustrates an RBM structure consisting of a visible layer with \( i \) units, and a hidden layer with \( j \) units. The outputs of units in a visible layer are treated as inputs of units in the hidden layer. An energy function of a joint structure \((v,h)\) of visible and hidden units can be depicted as Eq. (8):

\[
\text{Energy}(v,h) = \sum_{m=1}^{N} \alpha_m v_m - \sum_{n=1}^{N} \beta_n h_n - \sum_{m=1}^{N} \sum_{n=1}^{N} \gamma_{mn} v_m w_{mn} h_n \tag{8}
\]

where \( \alpha_m \) is the weight of connection unit \( m \) in the visible layer with the bias \( \alpha_m \) and unit \( n \) in the hidden layer with the bias \( \beta_n \). Values of \( V_m \) and \( h_n \) satisfy a binary distribution stochastically, and can be represented as Eq. (9):

\[
p(v,h) = \frac{\exp\text{-Energy}(v,h)}{\sum_{v,h} \exp\text{-Energy}(v,h)} \tag{9}
\]

Then, the derivative of \( \ln p(v,h) \) can be expressed as Eq. (10):

\[
\frac{\partial \ln p(v,h)}{\partial \alpha_m} = \sum_{h} \frac{\partial \text{-Energy}(v,h)}{\partial \alpha_m} \tag{10}
\]

The sigmoid function can be expressed as Eq.(12):

\[
sigm(x) = \frac{1}{1 + e^{-x}} \tag{12}
\]

The aim of the RBM is to search three appropriate parameters, namely \( \alpha, \beta, \) and \( W \), represented collectively as a parameter set \( \Omega \). The pre-trained parameter set serves as an initial parameter set for the supervised learning stage of the deep belief network. Thus, features of the training data set can be depicted well by the parameter set, and the likelihood function can be maximized. A constructive divergence algorithm [42] was presented to optimize the likelihood function when distributions of the visible layer and the hidden layer are in steady states [43]. The likelihood function is expressed as Eq. (13):

\[
\ln L(\Omega) = \ln \prod_{T=1}^{T_{\text{max}}} P(V_T) = \sum_{T=1}^{T_{\text{max}}} \ln P(V_T) \tag{13}
\]

where \( T \) is the number of training data. The gradient of Eq.(13) with respect to the parameter set can be expressed as Eq. (14):

\[
\frac{\partial \ln L(\Omega)}{\partial \alpha_m} = \sum_{T=1}^{T_{\text{max}}} \frac{\partial \ln P(V_T)}{\partial \alpha_m} \tag{14}
\]

Then, the derivative of \( \ln P(V_T) \) with respect to \( \Omega \) can be
expressed as Eq. (15). Thus, the gradient of the likelihood function can be represented as Eq. (16):

$$\frac{\partial \ln P(V_T)}{\partial \Omega} = \frac{\partial}{\partial \Omega} \left( \ln \sum_h e^{-E(V_T, h)} - \ln \sum_{v,h} e^{-E(v,h)} \right)$$

$$= - \sum_h P(h|V_T) \frac{\partial E(V_T, h)}{\partial \Delta} + \sum_v P(v, h) \frac{\partial E(v, h)}{\partial \Delta}$$

(15)

Finally, the learning algorithm with the momentum and the learning rate of the RBM is generated and shown as Eq. (17):

$$\Omega(t + 1) = MO_1 \Omega(t) + \eta_1 \frac{\partial \ln L(\Omega)}{\partial \Delta}$$

where \( MO_1 \) and \( \eta_1 \) are the momentum and the learning rate, respectively, in the unsupervised learning stage. Because the computations of Eq. (17) are hard, a \( k \)-step constructive divergence (CD-\( k \)) algorithm [42] is employed to address this issue. In this study, the CD-1 policy is employed. Thus, the learning rules of the three parameters can be represented as Eqs. (18-20):

$$W(t + 1) = MO_1 W(t) + \eta_1 \frac{1}{L} \sum_{l=1}^{L} \left( P(h_l|V^{(0)}) v^{(0)}_{tm} - P(h_l^{(1)}|v^{(1)}_{tm}) v^{(1)}_{tm} \right)$$

(18)

$$a(t + 1) = MO_1 a(t) + \eta_1 \frac{1}{L} \sum_{l=1}^{L} \left( v^{(0)}_{tm} - v^{(1)}_{tm} \right)$$

(19)

$$\beta(t + 1) = MO_1 \beta(t) + \eta_1 \frac{1}{L} \sum_{l=1}^{L} \left( P(h_l^{(1)}|v^{(1)}_{tm}) - P(h_l^{(0)}|v^{(0)}_{tm}) \right)$$

(20)

In addition, the \( \tau \) is depicted as Eq. (25):

$$\tau = \begin{cases} (A_n - Y_n) \left( \frac{\partial a(NET_n)}{\partial NET_n} \right) & \text{if connections are hidden -- to -- output} \\ \sum_{n=1}^{N} \left( A_n - Y_n \right) \left( \frac{\partial a(NET_n)}{\partial NET_n} \right) & \text{others} \end{cases}$$

(25)

where \( NET_n \) is the input a node receives from a previous layer, \( a(\cdot) \) is an activation function, and \( a(NET_n) \) is the output of a node, serving as an input for the next layer.

III. WIND SPEED FORECASTING BY THE DBNGA MODEL AND NUMERICAL RESULTS

A. THE PROPOSED FRAMEWORK

Taiwan has fairly plentiful wind energy resources [47], and the Penghu archipelago area has the strongest wind speed and produces the most wind power among several regions in Taiwan [48]. In this study, a deep belief network with genetic algorithm model was designed to forecast wind speed at Penghu. To examine the feasibility and applicability of the developed model, datasets from two other weather stations in Taiwan, namely Kinmen and Matsu, were used to forecast wind speed. The flowchart of this study is presented in Fig. 3, and stated as follows. First, the hourly wind speed and weather-related data at the Penghu station in Taiwan were collected from the Central Weather Bureau in Taiwan (http://e-service.cwb.gov.tw/HistoryDataQuery/index.jsp).

Originally, there are a total of 14 weather attributes, including wind speed, on this web-site. However, due to some missing or uncertain values, a total of 11 weather attributes were used in this study. The data, including wind speed, station pressure, sea pressure, temperature, dew point, temperature, relative humidity, wind direction, max gust, direction of max gust, precipitation amount, precipitation...
hours, and sunshine hours were used in this study to predict wind speed by time series models and multivariate models. These 11 weather attributes were taken from the work of Wang et al. [5]. The hourly weather data from January 1, 2017 to December 31, 2017 were used to model wind speed prediction models, and the hourly wind speed data from January 1, 2018 to January 14, 2018 were used as testing data. For multivariate forecasting models, a rolling and a one hour forecast format were used to forecast wind speed. Fourteen time windows from 12 to 168 hours, with 12 hour time intervals were examined to test the performance of forecasting models.

In this study, both time series forecasting methods and multivariate forecasting methods were used for predicting wind speed. The time series forecasting models used were the seasonal autoregressive integrated moving average method (SARIMA), and the least squares support vector regression for time series with genetic algorithm method (LSSVETSGA). These forecasting methods used past wind speed data as input data to forecast future wind speed.

Using the principles of evolution and natural selection, Holland [25] developed genetic algorithms as a structured random search method for discovering optimal or near optimal solutions with relatively mild computational burden. In this study, genetic algorithms are employed to select the parameters for the DBN, LSSVR and LSSVRTS models. In the deep belief network model, two parameters, namely the momentum and learning rate, are included in both unsupervised learning and supervised learning stages. In the unsupervised learning stage and the supervised learning stage, the negative values of the energy function of restricted Boltzmann machines and negative RMSE values of wind speed forecast serve as the fitness of genetic algorithms, respectively. Furthermore, one hidden layer is employed for the deep belief network in this study. Similarly, two parameters of the LSSVR and LSSVRTS models are provided by genetic algorithms using negative RMSE values of wind speed forecast as fitness function in the training stage. In this investigation, parameters are expressed by a chromosome composed of 40 genes in the form of binary numbers. The population size is set to 10. The crossover and mutation rates are both set to 0.7. The single-point crossover method is employed for genetic algorithms in this study. The varying range of the momentum and the learning rate is set to 0.1 to 0.9, correspondingly. The searching range of the two parameters for LSSVR and LSSVRTS is set to 1 to 500. 30 generations is used as a stop criterion for the genetic algorithms. The goal of selecting proper controlling parameters for metaheuristics is to optimize problems. The selection of controlling parameters of genetic algorithms used in this study can provide forecasting models with appropriate parameters, and result in accurate forecasting. Figure 4 shows the parameter selection mechanism for both the unsupervised pre-training stage, and the supervised fine-tuning stage in a deep belief network.

Figure 3. The proposed flowchart of wind speed forecasting.

Figure 4. The parameters selection mechanism a deep belief network.
B. NUMERICAL RESULTS
This study employed the least squares support vector regression with genetic algorithms and deep belief networks with genetic algorithms to predict wind speed with various weather attributes. Three forecasting performance measurements, the root mean square error (RMSE) and the mean absolute percentage error (MAPE) were used to measure the forecasting accuracy of the tested models, expressed as Eqs. (26-27):

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^{N} (A_t - F_t)^2}
\]

\[
\text{MAPE}(\%) = \frac{100}{N} \sum_{t=1}^{N} \left| \frac{A_t - F_t}{A_t} \right|
\]

where \(N\) is the number of forecasting periods, \(A_t\) is the actual value at period \(t\), \(F_t\) is the forecasting value at period \(t\), \(A_n\) is highest actual value, and \(A_1\) is the lowest actual value.

Tables 1-3 list the parameters of the four forecasting models at three weather stations. The wind speed forecasting values of different forecasting models and actual values are shown in Figs. 5-7 at three weather stations. Figures 5-7 show that the DBNGA model can capture wind speed patterns well, especially at turning points. However, the other three models are unable to effectively follow wind speed data patterns in mid-term time windows. Tables 4-6 list various measurements of forecasting accuracy by the four models with various time windows at three weather stations. Table 7 lists the averages of forecasting measurements for three weather stations. In terms of forecasting accuracy, it is clear that the DBNGA model outperforms the other three forecasting models in mid-term time windows, as well as in terms of the average for all time windows.

**TABLE 1. Parameters of forecasting models at the Penghu station.**

<table>
<thead>
<tr>
<th>Stations</th>
<th>Models</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Penghu</td>
<td>DBNGA</td>
<td>((LSSVRG, \eta_1) = (0.83, 0.45); (LSSVRG, \eta_2) = (0.99, 0.83))</td>
</tr>
<tr>
<td></td>
<td>LSSVRG</td>
<td>((0, \sigma) = (347.11, 1.22))</td>
</tr>
<tr>
<td></td>
<td>LSSVRSGA</td>
<td>((0, \sigma) = (359.46, 6.47))</td>
</tr>
<tr>
<td></td>
<td>SARIMA</td>
<td>((p, d, q) = (1, 1, 1)(0, 0, 2))</td>
</tr>
</tbody>
</table>

**TABLE 2. Parameters of forecasting models at the Kinmen station.**

<table>
<thead>
<tr>
<th>Stations</th>
<th>Models</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinmen</td>
<td>DBNGA</td>
<td>((LSSVRG, \eta_1) = (0.65, 0.04); (LSSVRG, \eta_2) = (0.32, 0.93))</td>
</tr>
<tr>
<td></td>
<td>LSSVRG</td>
<td>((0, \sigma) = (313.01, 0.68))</td>
</tr>
<tr>
<td></td>
<td>LSSVRSGA</td>
<td>((0, \sigma) = (351.20, 4.05))</td>
</tr>
<tr>
<td></td>
<td>SARIMA</td>
<td>((p, d, q) = (1, 1, 1)(0, 0, 2))</td>
</tr>
</tbody>
</table>

**TABLE 3. Parameters of forecasting models at the Matsu station.**

<table>
<thead>
<tr>
<th>Stations</th>
<th>Models</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matsu</td>
<td>DBNGA</td>
<td>((LSSVRG, \eta_1) = (0.88, 0.49); (LSSVRG, \eta_2) = (0.98, 0.94))</td>
</tr>
<tr>
<td></td>
<td>LSSVRG</td>
<td>((0, \sigma) = (335.44, 8.92))</td>
</tr>
<tr>
<td></td>
<td>LSSVRSGA</td>
<td>((0, \sigma) = (463.41, 4.86))</td>
</tr>
<tr>
<td></td>
<td>SARIMA</td>
<td>((p, d, q) = (1, 1, 1)(0, 0, 1))</td>
</tr>
</tbody>
</table>

FIGURE 5. Actual and forecasted values of hourly wind speed at the Penghu station.

**FIGURE 6. Actual and forecasted values of hourly wind speed at the Kinmen station.**

**FIGURE 7. Actual and forecasted values of hourly wind speed at the Matsu station.**
### TABLE 4. Measurements of forecasting accuracy by four models with various time windows at the Penghu station.

<table>
<thead>
<tr>
<th>Time Window (hours)</th>
<th>Measurements of forecasting accuracy</th>
<th>Forecasting models</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>RMSE</td>
</tr>
<tr>
<td>12</td>
<td>6.034</td>
<td>0.492</td>
</tr>
<tr>
<td>24</td>
<td>6.499</td>
<td>0.494</td>
</tr>
<tr>
<td>36</td>
<td>7.048</td>
<td>0.531</td>
</tr>
<tr>
<td>48</td>
<td>8.820</td>
<td>0.576</td>
</tr>
<tr>
<td>60</td>
<td>13.439</td>
<td>0.628</td>
</tr>
<tr>
<td>72</td>
<td>12.831</td>
<td>0.598</td>
</tr>
<tr>
<td>84</td>
<td>12.516</td>
<td>0.604</td>
</tr>
<tr>
<td>96</td>
<td>14.156</td>
<td>0.659</td>
</tr>
<tr>
<td>108</td>
<td>13.439</td>
<td>0.648</td>
</tr>
<tr>
<td>120</td>
<td>12.560</td>
<td>0.667</td>
</tr>
<tr>
<td>132</td>
<td>13.146</td>
<td>0.660</td>
</tr>
<tr>
<td>144</td>
<td>13.192</td>
<td>0.684</td>
</tr>
<tr>
<td>156</td>
<td>13.606</td>
<td>0.704</td>
</tr>
<tr>
<td>168</td>
<td>18.930</td>
<td>0.743</td>
</tr>
</tbody>
</table>

### TABLE 5. Measurements of forecasting accuracy by four models with various time windows at the Kinmen station.

<table>
<thead>
<tr>
<th>Time Window (hours)</th>
<th>Measurements of forecasting accuracy</th>
<th>Forecasting models</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>RMSE</td>
</tr>
<tr>
<td>12</td>
<td>13.263</td>
<td>0.742</td>
</tr>
<tr>
<td>24</td>
<td>11.831</td>
<td>0.708</td>
</tr>
<tr>
<td>36</td>
<td>13.116</td>
<td>0.708</td>
</tr>
<tr>
<td>48</td>
<td>14.949</td>
<td>0.744</td>
</tr>
<tr>
<td>60</td>
<td>15.401</td>
<td>0.690</td>
</tr>
<tr>
<td>72</td>
<td>14.789</td>
<td>0.686</td>
</tr>
<tr>
<td>84</td>
<td>16.756</td>
<td>0.720</td>
</tr>
<tr>
<td>96</td>
<td>18.704</td>
<td>0.709</td>
</tr>
<tr>
<td>108</td>
<td>23.780</td>
<td>0.726</td>
</tr>
<tr>
<td>120</td>
<td>22.466</td>
<td>0.716</td>
</tr>
<tr>
<td>132</td>
<td>21.572</td>
<td>0.719</td>
</tr>
<tr>
<td>144</td>
<td>20.951</td>
<td>0.711</td>
</tr>
<tr>
<td>156</td>
<td>20.529</td>
<td>0.696</td>
</tr>
<tr>
<td>168</td>
<td>21.789</td>
<td>0.694</td>
</tr>
</tbody>
</table>

### TABLE 6. Measurements of forecasting accuracy by four models with various time windows at the Matsu station.

<table>
<thead>
<tr>
<th>Time Window (hours)</th>
<th>Measurements of forecasting accuracy</th>
<th>Forecasting models</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>RMSE</td>
</tr>
<tr>
<td>12</td>
<td>16.284</td>
<td>0.880</td>
</tr>
<tr>
<td>24</td>
<td>16.284</td>
<td>0.880</td>
</tr>
<tr>
<td>36</td>
<td>14.119</td>
<td>0.734</td>
</tr>
<tr>
<td>48</td>
<td>12.528</td>
<td>0.659</td>
</tr>
<tr>
<td>60</td>
<td>12.560</td>
<td>0.621</td>
</tr>
</tbody>
</table>
with genetic algorithms model for predicting wind speed in Taiwan. Both wind speed data and various weather factors are employed to forecast wind speed at three weather stations in Taiwan. Numerical results indicate that DGNBA models can achieve more accurate forecasting results over mid-term periods, and in terms of the average forecasting accuracy for all time windows than the other models. Thus, the proposed DBNGA model is a feasible and promising method for forecasting wind speed. However, this study only used limited weather factors for multivariate forecasting methods of predicting wind speed. Future work will include some other factors, such as landforms, to increase the forecasting accuracy. Another possible direction for future study is to include the number of hidden layers and the number of hidden nodes in the fitness function of genetic algorithms used to select deep belief network models. Finally, other metaheuristics could be employed to select deep belief network models in order to improve forecasting performance.

### REFERENCES


