Energy and Power Engineering, 2014, 6, 340-348
Published Online October 2014 in SciRes. http://www.scirp.org/journal/epe
http://dx.doi.org/10.4236/epe.2014.611029

Bootstrapped Multi-Model Neural-Network Super-Ensembles for Wind Speed and Power Forecasting

Zhongxian Men[1,2], Eugene Yee[2,3], Fue-Sang Lien[1,2], Hua Ji[1], Yongqian Liu[4]

[1]Waterloo CFD Engineering Consulting Inc., Waterloo, Ontario, Canada
[2]Department of Mechanical & Mechatronics Engineering, University of Waterloo, Waterloo, Onatrio, Canada
[3]Defence Research and Development Canada, Suffield Research Centre, Medicine Hat, Alberta, Canada
[4]School of Renewable Energy, North China Electric Power University, Beijing, China
Email: eyee0309@gmail.com

Received 15 July 2014; revised 18 August 2014; accepted 4 September 2014

Abstract

The bootstrap resampling method is applied to an ensemble artificial neural network (ANN) approach (which combines machine learning with physical data obtained from a numerical weather prediction model) to provide a multi-ANN model super-ensemble for application to multi-step-ahead forecasting of wind speed and of the associated power generated from a wind turbine. A statistical combination of the individual forecasts from the various ANNs of the super-ensemble is used to construct the best deterministic forecast, as well as the prediction uncertainty interval associated with this forecast. The bootstrapped neural-network methodology is validated using measured wind speed and power data acquired from a wind turbine in an operational wind farm located in northern China.

Keywords

Artificial Neural Network, Bootstrap Resampling, Numerical Weather Prediction, Super-Ensemble, Wind Speed, Power Forecasting

1. Introduction

There has been an increasing emphasis towards a greater use of renewable energy (e.g., solar, wind, geothermal) as a strategy to reduce greenhouse gas emissions and to mitigate climate change. In this context, one of the fastest growing sources of renewable energy for the generation of "green electricity" is the power obtained from wind turbines. The ever increasing use of wind power poses new challenges. One important challenge is how to

How to cite this paper: Men, Z.X., Yee, E., Lien, F.-S., Ji, H. and Liu, Y.Q. (2014) Bootstrapped Multi-Model Neural-Network Super-Ensembles for Wind Speed and Power Forecasting. *Energy and Power Engineering*, **6**, 340-348.
http://dx.doi.org/10.4236/epe.2014.611029

accommodate the unpredictable fluctuations in wind speed and direction which lead to variability and uncertainty in the wind power generation. The latter has significant implications for unit commitment and determination of scheduling and dispatch decisions (economic dispatch) needed for the optimal utilization of wind energy within a mixed power system. In this regard, wind power forecasting has become a critical component in the efficient management of a green electrical power system (required by generation companies and utilities) and in electrical market operations (required by energy market analysts and traders).

The development of wind power forecasting models for improving the efficiency and reliability of mixed electrical power systems and for supporting electrical market operations has been reviewed by Costa *et al.* [1], Ma *et al.* [2] and Foley *et al.* [3]. Methodologies for wind power forecasting can be categorized into three broad classes as follows: statistical, physical, and machine learning. Because time series of wind speed and power are frequently measured in the vicinity of wind farms, statistical approaches based on time series analysis and forecasting have been applied for the prediction of wind speed and power. Towards this purpose, time series forecasting based on the popular Box-Jenkins methodology [4], as applied to autoregressive (AR) models, moving average (MA) models, and autoregressive moving average (ARMA) models, has been utilized for wind speed and power forecasting using historical time series of wind speed and power. As an example, Erdem and Shi [5] demonstrated the application of an ARMA model for wind speed and direction forecasting. More sophisticated nonlinear time series models that accommodate the temporal evolution of the variance (heteroskedasticity) such as the autoregressive conditional heteroskedasticity (ARCH) model by Engle [6] and the generalized autoregressive conditional heteroskedasticity (GARCH) models by Bollerslev [7] have been applied to describe the intrinsic variability in the wind speed and the associated generated power. For instance, the ARMA-GARCH and the GARCH-in-mean (GARCH-M) models proposed by Liu *et al.* [8] have been applied to model the mean and volatility of the wind speed.

The second general class of models is physically-based models for wind speed prediction based on numerical weather prediction (NWP) or computational fluid dynamics (CFD). Utilizing equations of physics such as the conservation principles of mass, momentum, and energy in conjunction with various parameterizations for subgrid scale physical processes that cannot be resolved explicitly by the necessarily finite number of grid points that are used to represent the atmospheric flow, NWP and/or CFD models provide hydrodynamic and thermodynamic models of the atmosphere that can be used to furnish a prediction of the flow field in a prescribed region. The prediction of the wind velocity field can be used in conjunction with the power curve for a wind turbine to provide a generated power forecast. Numerical weather prediction models have a number of limitations, including limited spatial resolution resulting in a coarse representation of the local terrain [9]. To overcome the latter problem, Liu *et al.* [10] considered the possibility of coupling a synoptic scale flow model to a large-eddy simulation model for wind energy applications and Li *et al.* [11] recently introduced a short-term wind forecasting methodology based on the use of CFD pre-calculated flow fields.

The third general class of models for wind power forecasting is based on machine learning approaches such as artificial neural networks, fuzzy systems, and support vector machines [3] [12] [13]. Unlike the parametric models used for time series forecasting, machine learning uses either a “gray” or “black” box (essentially nonparametric) representation for the underlying physical processes (defining a nonlinear mapping from an input to an output), and then utilizes various learning algorithms and historical time series of wind speed or power to “train” the gray (black) box. The black box trained in this manner can be applied subsequently to make predictions of the future wind speed or generated power (from a wind turbine).

In this paper, we propose to use the bootstrap resampling method in conjunction with an ensemble artificial neural network (ANN) approach for the multi-step-ahead forecasting of wind speed and generated power. The artificial neural network combines machine learning with physical modeling by using NWP wind speed data from a physical model as the exogenous input to the network. The purpose of the bootstrap resampling method is to reduce the bias in prediction of the wind speed and power and to obtain more accurate estimates for the standard deviation (uncertainty) of these predictions. More importantly, the confidence bands in these predictions can be determined, which can be used to provide a more rigorous uncertainty assessment in wind speed and power forecasting.

2. Bootstrapping Ensembles of Artificial Neural Networks

As discussed in the previous section, a major concern of wind energy management is the uncertainty quantification of multi-step-ahead predictions of the wind speed (at the turbine hub height) and the corresponding power generated by the turbine. Instead of choosing a single best ANN for forecasting, we propose instead to use the bootstrap resampling method in the context of an ensemble of ANNs for predictive uncertainty analysis.

Let $\boldsymbol{y} \equiv \left(y_1, y_2, \cdots, y_T\right)'$ be a sample of realizations of a (scalar) random variable y, where the positive integer T represents the sample size and the $'$ denotes the transpose of a vector (or matrix). In this paper, y represents either the measured wind speed (at turbine hub height) or the power generated by a wind turbine. Similarly, let $\boldsymbol{x} \equiv \left(x_1, x_2, \cdots, x_T\right)'$ to be a sample of the corresponding predicted wind speed (at turbine hub height) obtained from a numerical weather prediction model. Finally, let $\boldsymbol{\zeta} \equiv \left(\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_T\right)'$ with $\boldsymbol{z}_t = \left(y_t, x_t\right)$ ($t = 1, 2, \cdots, T$ where t is the time step). Note that $\boldsymbol{\zeta}$ consists of the sample of collections of the measured wind speed (or power) y and the corresponding modeled wind speed x.

We want to first represent (model) the functional relationship between y and x based on the training set $\boldsymbol{\zeta}$, by using an ensemble of artificial neural networks, and then employ all of the trained ANNs in the ensemble to forecast y when new values of x become available. In this sense, the forecasting of wind speed and generated power is obtained by conditioning not only on a single best ANN (model), but on an entire ensemble of plausible ANNs (models).

The nonlinear parameterized mapping $\boldsymbol{f}$ from an input $\boldsymbol{x}$ to an output $\boldsymbol{y}$ can be described generally by an ANN given by

$$\boldsymbol{y} = \boldsymbol{f}\left(\boldsymbol{x}; \boldsymbol{\theta}\right). \tag{1}$$

The output of the ANN is a continuous function of the input and $\boldsymbol{\theta}$ is a collection of weights and biases (parameters) that determine the architecture of the neural network. By virtue of the universal approximation theorem demonstrated by Hornik *et al.* [14], it is known that an ANN with three or more layers can approximate any continuous function provided the activation function is a locally bounded, piecewise continuous function. In view of this, in our application, we use a three-layer neural network structure consisting of an input layer, a hidden layer, and an output layer. More specifically, the mapping for the ANN for our application has the following explicit form:

$$v_j^{(1)} = \sum_l w_{jl}^{(1)} x_l + b_j^{(1)}; \quad h_j = f^{(1)}\left(v_j^{(1)}\right) \tag{2}$$

for the hidden layer and

$$v_i^{(2)} = \sum_j w_{ij}^{(2)} h_j + b_i^{(2)}; \quad y_i = f^{(2)}\left(v_i^{(2)}\right) \tag{3}$$

for the output layer. In Equations (2) and (3), the index l varies over the input, the index j varies over the hidden units, and the index i varies over the output. Furthermore, $f^{(1)}(\cdot)$ and $f^{(2)}(\cdot)$ are activation functions for the hidden and output layers of the network, respectively. For our application, the activation functions used to define the neural network architecture are $f^{(1)}(v) = 1\big/\left(1 + \exp(-v)\right)$ (logistic sigmoidal function) and $f^{(2)}(v) = v$ (simple linear function).

Note that each neuron in the network is a unit that combines and processes all the data coming into the layer and then passes the transformed data (output of the activation function) to all the neurons of the successive layer. Specifically, the input of a neuron is a weighted sum of the outputs of all the neurons in the previous layer plus a bias. The weights w and the biases b collectively make up the parameter vector $\boldsymbol{\theta}$. The ANN is trained using the data set $\boldsymbol{\zeta}$ by selecting the parameter vector $\boldsymbol{\theta}$ so as to minimize some error function which measures how close the modeled output $\boldsymbol{f}\left(\boldsymbol{x}; \boldsymbol{\theta}\right)$ is to the measured output $\boldsymbol{y}$ of the training set. Two specific forms of the error function that will be used for this purpose are the root-mean-square error (RMSE)

$$E_{\text{RMSE}}\left(\boldsymbol{\theta}\right) = \left(T^{-1} \sum_{t=1}^{T} \left(y_t - f_t\left(\boldsymbol{x}; \boldsymbol{\theta}\right)\right)^2\right)^{1/2} \tag{4}$$

and the mean absolute error (MAE)

$$E_{\text{MAE}}\left(\boldsymbol{\theta}\right) = T^{-1} \sum_{t=1}^{T} \left| y_t - f_t\left(\boldsymbol{x}; \boldsymbol{\theta}\right) \right|. \tag{5}$$

The minimization of these error functions is achieved using the particle swarm optimization algorithm [15]. Once an ANN has been trained, it can be used for the out-of-sample forecasting of the dependent (output) variable y.

To apply the bootstrap resampling procedure [16] to the ANN, we have to impose a statistical distribution on the sample data ζ. To this purpose, we follow the standard bootstrap method and assume that the empirical distribution function $\hat{F}$ of $\boldsymbol{z}_t$ $\left(t=1,2,\cdots,T\right)$ is a uniform distribution (viz., one assigns equal probability to each sample value $\boldsymbol{z}_t$). More specifically, we assign to each $\boldsymbol{z}_t$ a probability of T^{-1} and realize the distribution $\hat{F}$ such that $\hat{p}\left(\boldsymbol{z}=\boldsymbol{z}_t\right)=T^{-1}$ for $t=1,2,\cdots,T$. This distribution implies that any statistic of the observed data (viz., any functional of the data) is invariant under all permutations of the components in the sample ζ. The nonparametric simulation of bootstrap data sets based on the empirical distribution function $\hat{F}$ is simple: we sample with replacement from the components of ζ (viz., because $\hat{F}$ places equal probability on each of the original data values in ζ, each sample is obtained by independently sampling at random from these data values). By so doing, we can obtain a bootstrap sample with a sample size of T. We can repeat this process, say N times. Then, we will have N bootstrap samples drawn from the data values in ζ, which we will denote by $\zeta^{(1)},\zeta^{(2)},\cdots,\zeta^{(N)}$.

For each of these bootstrap samples, we can train an ensemble of ANNs with the same network architecture, but with each member of the ensemble having different numbers of neurons in the hidden layer (recall that the number of neurons in the input and output layers are determined *a priori* by the dimension of the input and output vectors, respectively). To be more specific, assume that the number of neurons in the hidden layer of the network architecture varies from j to q $\left(j\le q\right)$ inclusive. Furthermore, in order to explicitly treat the initialization uncertainty (viz., the uncertainty arising from the initialization of the weights used in training an ANN), we will train each ANN model structure (with a fixed number of hidden nodes) starting from m different sets of initial weights $\boldsymbol{\theta}$. In consequence, we have a super-ensemble of ANNs consisting of $Nm\left(q-j+1\right)$ members [viz., $\left(q-j+1\right)$ different ANN model structures, each of which is trained starting from m different random initializations of the weights on a particular bootstrap sample $\zeta^{(n)}$ with $n=1,2,\cdots,N$]. Each member of this super-ensemble of ANNs can be used to conduct a multi-step-ahead prediction of y.

The procedure for bootstrapping an ensemble of neural networks is summarized as follows:

1. Assign the nonparametric distribution $\hat{F}$ to the observed data $\zeta\equiv\left(\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_T\right)'$,

$$\hat{p}\left(\boldsymbol{z}=\boldsymbol{z}_t\right)=\frac{1}{T},\quad t=1,2,\cdots,T. \tag{6}$$

2. Draw a (nonparametric) bootstrap sample (with replacement) from the empirical distribution function $\hat{F}$,

$$\boldsymbol{z}_1^{(*)},\boldsymbol{z}_2^{(*)},\cdots,\boldsymbol{z}_T^{(*)}\overset{\text{iid}}{\sim}\hat{F}, \tag{7}$$

and train m ANNs starting from m different (random) initializations for the weight vector $\boldsymbol{\theta}$ for a fixed ANN model structure (viz., an ANN with a fixed number of nodes in the hidden layer). Repeat the process for different ANN model structures with the number of nodes in the hidden layer varying from j to q $\left(j\le q\right)$ inclusive. The multi-step-ahead forecasting of the wind speed and generated power $\tilde{\boldsymbol{y}}$ is carried out for each of the trained ANNs in the ensemble.

3. Repeat Step 2 N times to obtain bootstrap replications for the forecasted wind speed and power; namely, $\tilde{\boldsymbol{y}}^{(1)},\tilde{\boldsymbol{y}}^{(2)},\cdots,\tilde{\boldsymbol{y}}^{(K)}$ with $K=Nm\left(q-j+1\right)$. Calculate the mean and standard deviations of the bootstrap predictions of $\tilde{\boldsymbol{y}}$ using the procedure described below. These quantities can be used in conjunction with a bootstrap-*t* method to obtain confidence intervals for the forecasts of the wind speed and generated power.

We use a two-stage weighted averaging method to provide the predictive uncertainty assessment for the wind speed and power. For *each* bootstrap sample, we calculate the predictions (forecasts) of the multi-step-ahead wind speed and power using the $L\equiv m\left(q-j+1\right)$ members of the ensemble of ANNs trained using the bootstrap sample. A statistical combination of these forecasts is used to obtain the best forecast based on the current ensemble whose members have been trained using the *given* bootstrap sample. This procedure is repeated for each bootstrap sample, and the optimum forecast is calculated as a weighted average (statistical combination) of these N best forecasts obtained from the N bootstrap samples in the super-ensemble. The information can be used also to determine the confidence intervals in the forecasted quantities.

For each bootstrap sample, $\boldsymbol{\zeta}^{(n)}$ for $n=1,2,\cdots,N$, we have $L \equiv m(q-p+1)$ in-sample predictions of $\boldsymbol{y}$, which we denote as $\hat{\boldsymbol{y}}^l \equiv \left(\hat{y}_1^l, \hat{y}_2^l, \cdots, \hat{y}_T^l\right)'$ $(l=1,2,\cdots,L)$. Similarly, we have L out-of-sample predictions $y^l \equiv \left(\tilde{y}_{T+1}^l, \tilde{y}_{T+2}^l, \cdots, \tilde{y}_{T+d}^l\right)'$ for $l=1,2,\cdots,L$, where $d \geq 1$ is the prediction horizon. Define the weight vector $\boldsymbol{w}^{(n)} \equiv \left(w_1^{(n)}, w_2^{(n)}, \cdots, w_L^{(n)}\right)'$ with $w_l^{(n)} \geq 0$ and $\sum_{l=1}^{L} w_l^{(n)} = 1$. For each bootstrap sample (fixed *n*), the weights $\boldsymbol{w}^{(n)}$ can be chosen to define that statistical combination of the forecasts from the various (L) members of the ensemble that provide the best forecast. To this purpose, the statistical weights $\boldsymbol{w}^{(n)}$ for each model are determined from the training data set by a constrained[1] minimization of the following quadratic (objective) function:

$$\mathcal{E}_L\left(\boldsymbol{w}^{(n)}\right) = \frac{1}{L}\left\|\hat{\boldsymbol{E}}\boldsymbol{w}^{(n)}\right\|^2 = \boldsymbol{w}^{(n)\prime}\boldsymbol{H}_L\boldsymbol{w}^{(n)}, \tag{8}$$

where $\hat{\boldsymbol{E}} \equiv \left(\hat{\boldsymbol{e}}^1, \hat{\boldsymbol{e}}^2, \cdots, \hat{\boldsymbol{e}}^L\right)$ is the error matrix, with $\hat{\boldsymbol{e}}^l \equiv \boldsymbol{y} - \hat{\boldsymbol{y}}^l$ $(l=1,2,\cdots,L)$. Here, $\|\cdot\|$ denotes the Euclidean norm and $\boldsymbol{H}_L \equiv \frac{1}{L}\hat{\boldsymbol{E}}'\hat{\boldsymbol{E}}$ is an $(L \times L)$ matrix.

Once the weights $w_l^{(n)}$ for $l=1,2,\cdots,L$ have been obtained for each bootstrap sample (fixed n), we can use them to perform the best in-sample predictions $\hat{\boldsymbol{y}}$ and best multi-step-ahead predictions $\tilde{\boldsymbol{y}}$ of the output variable $\boldsymbol{y}$. More specifically, for a given ensemble of ANNs that has been trained for a fixed bootstrap sample, the in-sample predictions of $\boldsymbol{y}$ for the *l*-th member of the ensemble are denoted explicitly by $\hat{\boldsymbol{y}}^l \equiv \left(\hat{y}_1^l, \hat{y}_2^l, \cdots, \hat{y}_T^l\right)'$ and the multi-step-ahead predictions of $\boldsymbol{y}$ for the *l*-th member of the ensemble are denoted by $\tilde{\boldsymbol{y}}^l \equiv \left(\tilde{y}_{T+1}^l, \tilde{y}_{T+2}^l, \cdots, \tilde{y}_{T+d}^l\right)'$ [recall $d > 0$ is the prediction horizon]. These best predictions (in-sample and multi-step-ahead) are given by

$$\hat{y}_t^{(n)}\left(w^{(n)}\right) = \sum_{l=1}^{L} \hat{w}_l^{(n)} \hat{y}_t^l, \quad t = 1,2,\cdots,T, \tag{9}$$

$$\tilde{y}_t^{(n)}\left(w^{(n)}\right) = \sum_{l=1}^{L} \hat{w}_l^{(n)} \tilde{y}_t^l, \quad t = T+1,\cdots,T+d, \tag{10}$$

where $w^{(n)} = \left(\hat{w}_1^{(n)}, \hat{w}_2^{(n)}, \cdots, \hat{w}_L^{(n)}\right)$ denotes the solution to the minimization of the quadratic error function $\mathcal{E}_L\left(\boldsymbol{w}^{(n)}\right)$ given in Equation (8).

For a *fixed* bootstrap sample, the standard deviation vector $\tilde{\boldsymbol{\sigma}}^{(n)} \equiv \left(\tilde{\sigma}_{T+1}^{(n)}, \tilde{\sigma}_{T+2}^{(n)}, \cdots, \tilde{\sigma}_{T+d}^{(n)}\right)'$ for the multi-step-ahead prediction of $\boldsymbol{y}$ is estimated as follows (using the L forecasts $\tilde{y}_t^l$ $(l=1,2,\cdots,L)$ for y_t obtained from the ANN ensemble at the fixed time index t):

$$\tilde{\sigma}_t^{(n)} = \left(\sum_{l=1}^{L} \hat{w}_l^{(n)} \left[\tilde{y}_t^l - \bar{y}_t\right]^2\right)^{1/2}, \qquad t = T+1,\cdots,T+d, \tag{11}$$

where $\bar{y}_t \equiv \sum_{l=1}^{N} \tilde{y}_t^l / L$ is the (ensemble) mean of the samples in the set $\left\{\tilde{y}_t^1, \cdots, \tilde{y}_t^L\right\}$. Because this is a biased estimator of the standard deviation, we use instead the following formula

$$\tilde{\sigma}_t^{(n)} = \left(\left[1 - \sum_{l=1}^{L}\left(\hat{w}_l^{(n)}\right)^2\right]^{-1} \cdot \sum_{l=1}^{L} \hat{w}_l^{(n)} \left[\tilde{y}_t^l - \bar{y}_t\right]^2\right)^{1/2}, \qquad t = T+1,\cdots,T+d, \tag{12}$$

[1] Recall the weights $w_l^{(n)}$ are non-negative and satisfy $\sum_{l=1}^{L} w_l^{(n)} = 1$.

to obtain an unbiased estimator of the standard deviation.

At this point, we have N different forecasts for the output variable y (along with the standard deviations in these forecasts) obtained by applying the weighted-averaging schema described above to the ensemble of ANNs trained on each of the N bootstrap samples. The second stage of the process is to apply the weighted-averaging schema again to calculate the bootstrap predictions of $\tilde{y}$ based on these N different forecasts. To this purpose, we define another weight vector $\boldsymbol{\eta} = \left(\eta_1,\cdots,\eta_N\right)'$, whose estimate $\hat{\boldsymbol{\eta}} = \left(\hat{\eta}_1,\cdots,\hat{\eta}_N\right)'$ can be evaluated similarly by minimizing an objective function similar to Equation (8), except now the error matrix is constructed from the residuals between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}^{(n)}$ $\left(n=1,2,\cdots,N\right)$, where $\hat{\boldsymbol{y}}^{(n)} = \left(\hat{y}_1^{(n)},\hat{y}_2^{(n)},\cdots,\hat{y}_T^{(n)}\right)'$ is calculated by using Equation (9). Once the weight vector $\boldsymbol{\eta}$ has been estimated, the weighted bootstrap predictions of y are obtained from

$$\tilde{y}_t^B\left(\hat{\boldsymbol{\eta}}\right) = \sum_{n=1}^{N} \hat{\eta}_n \tilde{y}_t^{(n)}, \qquad t = T+1,\cdots,T+d, \tag{13}$$

and the corresponding bootstrapped standard deviations from

$$\tilde{\sigma}_t^B = \left(\sum_{n=1}^{N} \hat{\eta}_n \left[\tilde{\sigma}_t^{(n)}\right]^2\right)^{1/2}, \qquad t = T+1,\cdots,T+d, \tag{14}$$

where $\tilde{\sigma}_t^{(n)}$ (standard deviation obtained from the ANN ensemble trained on the *n*-th bootstrap sample) is computed in accordance to Equation (12).

Confidence intervals for the forecast of the output variable y will be obtained using the bootstrap-*t* method. Towards this purpose, the confidence intervals of the multi-step-ahead prediction $\tilde{\boldsymbol{y}}^B$ of $\boldsymbol{y}$ can be determined in accordance to the following formula:

$$\left(\tilde{y}_t^B\right)_{\text{est}} = \tilde{y}_t^B \pm \tilde{\sigma}_t^B t_{\alpha,v}, \qquad t = T+1,\cdots,T+d, \tag{15}$$

with $t_{\alpha,v}$ being the α-level critical value of a Student's-*t* distribution with $v = K-1$ degrees of freedom [recall $K = Nm\left(q-p+1\right)$]. Alternatively, $t_{\alpha,v}$ can be replaced by the bootstrap percentiles of the sample $\tilde{y}_t^{(k)}$ $\left(k=1,2,\cdots,K\right)$. So, for example, extracting the 2.5% and 97.5% bootstrap percentiles of this sample would allow one to construct a bootstrap-t based confidence interval for the prediction at a coverage level of 95%.

3. Application of the Methodology

3.1. Data Preparation

The two data sets that we analyse were collected from a specific wind turbine, referred to as WT24 hereafter, located in a wind farm in northern China. One of these data sets corresponds to the hourly-averaged wind speeds measured at the turbine hub height and the other corresponds to the associated hourly-averaged power generated by the turbine. The wind speed and generated power time series, consisting of 432 observations each, were measured over a period of 18 days. The measurements collected over the first 15 days (corresponding to 83% of the entire length) of the time series were used as the training set and the remaining 3 days were reserved for the forecast assessment and validation. In addition to these measured time series, wind speed data at turbine hub height obtained from a numerical weather prediction model was available and this information was used as the exogenous input for artificial neural network training. In particular, the modeled wind speed and direction data over the region occupied by the wind farm were obtained from a NWP model executed with a temporal resolution of 1 h on a computational grid with a 3-km spatial resolution centered on the location of the wind farm. We applied a simple bilinear interpolation (BI) on this coarse-resolution NWP wind speed data to obtain the wind speed at the location of the WT24 wind turbine, which was then subsequently used as an exogenous input for ANN training.

As described in the previous section, we bootstrapped (resampled with replacement) the training data set to generate N = 200 "phantom" (bootstrap) data sets. We used each of these bootstrap data sets to train three-layer

ANNs with a variable number of nodes in the hidden layer ranging from 5 to 30 nodes inclusive (so, $j = 5$ and $q = 30$).[2] In order to reduce the uncertainty arising from initialization of the network parameter (weight) vector, each network was trained 5 times each time starting from a different randomly chosen initialization of the weight vector (viz., $m = 5$). For each bootstrap sample (training data set), we have an ensemble consisting of $L \equiv m(q-p+1) = 130$ member ANNs trained on sample. In consequence, the super-ensemble of ANNs trained on the entire set of bootstrapped samples consists of $K \equiv Nm(q-p+1) = 26,000$ member ANNs. The information embodied in this ensemble can be used for the multi-step-ahead forecasting of the wind speed and generated power, along with a quantitative assessment of the prediction confidence.

3.2. Results

Figure 1 compares the measured wind speed at the hub height for WT24 with the out-of-sample forecasted wind speed obtained from bootstrapping an ensemble of ANNs. The best deterministic forecast for the wind speed based on a statistical combination of the individual forecasts in the super-ensemble [determined in accordance to Equations (10) and (13)] is shown in this figure (dot-dashed line labeled “Best forecast”). A comparison of the best deterministic forecast with the measured wind speed shows that the forecast captures adequately the longer temporal trends in the measured wind speed. In addition, the 95% prediction confidence intervals obtained using the two-stage weighted-averaging method is exhibited in **Figure 1** as the dotted lines demarcated using an open circle. Note that the 95% prediction uncertainty range appears to cover most of the observations, providing an observation coverage that is consistent (approximately or better) with the quoted level of confidence in the uncertainty interval. A quantitative assessment of the forecast performance in this case is summarized in **Table 1**, which summarizes the RMSE and MAE in the wind speed prediction using the bootstrapped neural-network methodology. The performance of this forecast methodology can be compared to that obtained from a simple persistence model forecast which uses the current wind speed to predict the value of the future wind speed.

Next, we consider the forecast performance for the wind power using the bootstrapped neural-network methodology. The forecast for the generated power is more complicated than that for the wind speed owing to

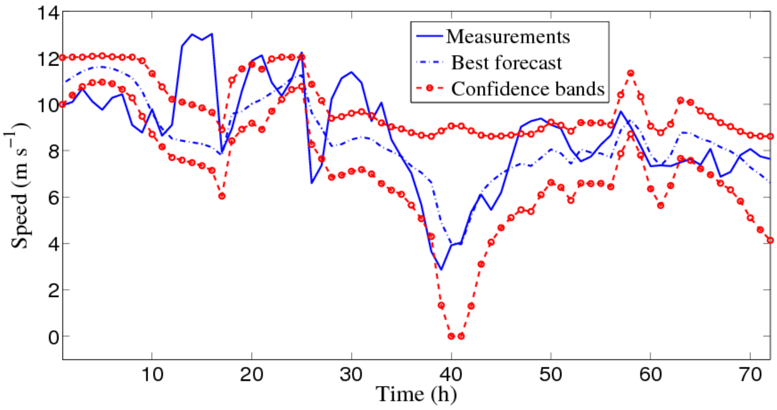


Figure 1. Out-of-sample wind speed forecasting obtained using the bootstrapped neural-network methodology. The 95% confidence interval for the forecast was calculated by using the two-stage weighted-averaging method applied to the ANN members of the super-ensemble.

Table 1. Wind speed forecast assessment of the bootstrapped neural-network methodology.

Criterion	Persistence	Best
RMSE	2.3554	1.6394
MAE	1.8693	1.2145

[2] Recall that the number of nodes in the input and output layers of the neural network are determined (fixed) by the dimensionality of the input and output vectors, respectively.

the fact that the wind power is censored from above. More specifically, wind turbines are designed so that when the wind speed exceeds a certain value (referred to as the rated output wind speed), a limit to the power generation is imposed implying that the power generated is censored from above. To account for this maximum limit in the wind power generation by a wind turbine, the bilinear interpolation of the NWP wind speeds to the location of WT24 were censored (to the rated output wind speed of the turbine) before they were used as the exogenous input for the neural network training. Furthermore, the measured wind power used in the training of the ANNs were already censored from above by the maximum limit for power generation by the turbine. Indeed, for the current example, if the modeled wind speeds exceeded 11.5 m·s^{-1} (rated output wind speed for WT24), then the generated wind power associated with this range of wind speeds was limited above to 1550 W (rated output power for WT24).

Figure 2 compares the measured power generated by the turbine WT24 with the forecasted wind power obtained using the bootstrapped neural-network methodology. The best deterministic forecast ("Best forecast") is shown based on a particular statistical combination of the individual forecasts obtained from the various members of the super-ensemble using the two-stage weighted-averaging process. In addition, the 95% confidence intervals for the predicted wind power is superimposed on the plot. **Table 2** summarizes the values of the RMSE and MAE for power forecasts obtained using the persistence methodology and the bootstrapped neural-network methodology. A comparison of the best deterministic forecast of the power (dash-dotted line) with the measured power shows that the broad features in the variation of the power is captured fairly well. From **Table 2**, it is seen that this best deterministic forecast gives misfits (either RSME or MAE) that are roughly a factor of three less than those obtained using the simple naïve persistence forecast. The observation coverage for the 95% prediction uncertainty intervals captures the power measurements fairly well, although qualitatively it is judged that this coverage is not as good as for the case of wind speed prediction. This would indicate that there may be sources of uncertainty in the wind power forecasting that have not been accounted for in the bootstrapping process.

4. Summary and Conclusions

In this paper, we proposed a novel bootstrapped artificial neural-network approach for wind speed and generated power forecasting. The approach provides a multi-ANN model super-ensemble that can be used to provide a best deterministic forecasting for these quantities, as well as to provide a quantitative assessment of the related

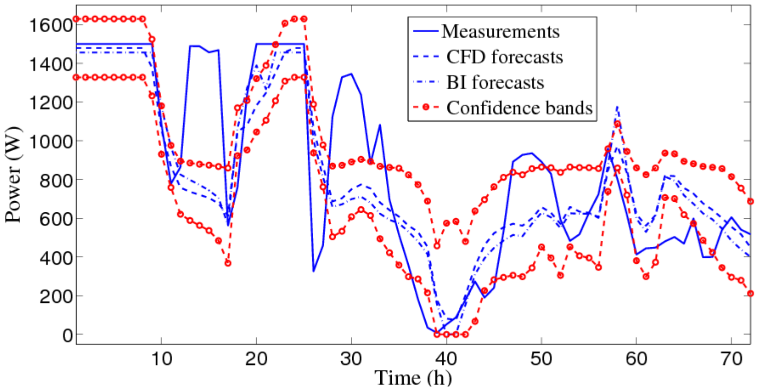


Figure 2. Out-of-sample wind power forecasting using the bootstrapped neural-network methodology. The 95% confidence intervals for the forecast were calculated by using the two-stage weighted-averaging method applied to the ANN members of the super-ensemble.

Table 2. Wind power forecast assessment of the bootstrapped neural-network methodology.

Criterion	Persistence	Best
RMSE	811.652	292.7335
MAE	653.252	210.8805

prediction uncertainty. In this approach, the individual ANNs that comprise the super-ensemble are first trained using a data set of available wind speed and power measurements and wind speed predictions (obtained from a numerical weather prediction model). The training consists of fitting various artificial neural network architectures (model structures) against the observation and exogenous input to determine the optimal statistical weights for each model.

The advantage of this methodology is that the biases in the forecast can be reduced and good predictions (forecasts) can be obtained through a statistical combination of the individual forecasts from the super-ensemble to give the best deterministic forecast. Applications to a wind turbine in northern China show that our proposed method works quite well. Because the method also provides prediction uncertainty bounds in the forecasts, it is anticipated that this approach would be very useful for green electrical system power management. Indeed, with the rapid pace of increases in computational power, it will become easier in the near future for power system managers and energy system traders/analysts to take advantage of the super-ensemble approach, for providing optimal forecasts and quantitative assessments of the uncertainty associated with these forecasts (allowing this information to be used in a more accurate and reliable manner for various applications).

References

[1] Costa, A., Crspo, A., Navarro, J., Lizano, G., Madsen, H. and Feitosa, E. (2008) A Review on the Young History of the Wind Power Short-Term Prediction. *Renewable and Sustainable Energy Reviews*, **12**, 1725-1744. http://dx.doi.org/10.1016/j.rser.2007.01.015

[2] Ma, L., Luan, S., Jiang, C., Liu, H. and Zhang, Y. (2009) A Review on the Forecasting of Wind Speed and Generated Power. *Renewable and Sustainable Energy Reviews*, **13**, 915-920. http://dx.doi.org/10.1016/j.rser.2008.02.002

[3] Foley, A.F., Leahy, P.G., Marvugliaand, A. and McKeogh, E.J. (2012) Current Methods and Advances in Forecasting of Wind Power Generation. *Renewable Energy*, **37**, 1-8. http://dx.doi.org/10.1016/j.renene.2011.05.033

[4] Box, G.E.P. and Jenkins, G. (1970) Time Series Analysis, Forecasting and Control. Holden-Day, San Francisco.

[5] Erdem, E. and Shi, J. (2011) ARMA Based Approaches for Forecasting the Tuple of Wind Speed and Direction. *Applied Energy*, **88**, 1405-1414. http://dx.doi.org/10.1016/j.apenergy.2010.10.031

[6] Engle, R.F. (1982) Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, **50**, 987-1007. http://dx.doi.org/10.2307/1912773

[7] Bollerslev, T. (1986) Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, **31**, 307-327. http://dx.doi.org/10.1016/0304-4076(86)90063-1

[8] Liu, H., Erdem, E. and Shi, J. (2011) Comprehensive Evaluation of ARMA-GARCH(-M) Approaches for Modeling the Mean and Volatility of Wind Speed. *Applied Energy*, **88**, 724-732. http://dx.doi.org/10.1016/j.apenergy.2010.09.028

[9] Martí Perez, I., Nielsen, T.S., Madsen, H., Navarro, J., Roldán, A., Cabezón, D. and Barquero, C.G. (2001) Prediction Models in Complex Terrain. *Proceedings of the European Wind Energy Conference*, Copenhagen, 2001, 875-878.

[10] Liu, Y., Warner, T., Liu, Y., Vincent, C., Wu, W., Mahoney, B., Swerdlin, S., Parks, K. and Boehnert, J. (2011) Simultaneous Nested Modeling From the Synoptic Scale to the LES Scale for Wind Energy Applications. *Journal of Wind Engineering and Industrial Aerodynamics*, **99**, 308-319. http://dx.doi.org/10.1016/j.jweia.2011.01.013

[11] Li, L., Liu, Y., Yang, Y. and Han, S. (2013) Short-Term Wind Speed Forecasting Based on CFD Pre-Calculated Flow Fields. *Proceedings of the Chinese Society of Electrical Engineering*, **33**, 27-32.

[12] Kariniotakis, G., Stavrakakis, G.S. and Nogaret, E.F. (1996) Wind Power Forecasting Using Advanced Neural Network Models. *IEEE Transactions on Energy Conversion*, **11**, 762-767. http://dx.doi.org/10.1109/60.556376

[13] Kariniotakis, G. (2003) Forecasting of Wind Parks Production by Dynamic Fuzzy Models with Optimal Generalisation Capacity. *Proceedings of the* 12*th Intelligent System Application to Power Systems* 03, ISAP 03/032, Lemnos, September 2003.

[14] Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, **2**, 359-366. http://dx.doi.org/10.1016/0893-6080(89)90020-8

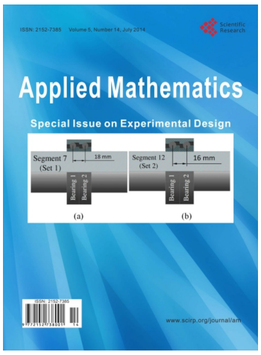
[15] Kennedy, J. and Eberhart, R.C. (2001) Swarm Intelligence. Morgan Kaufmann Publishers, San-Francisco.

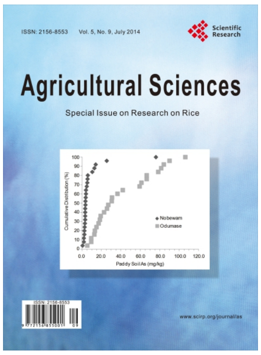
[16] Efron, B. and Tibshirani, B. (1993) An Introduction to the Bootstrap. Chapman & Hall/CRC, Boca Raton.

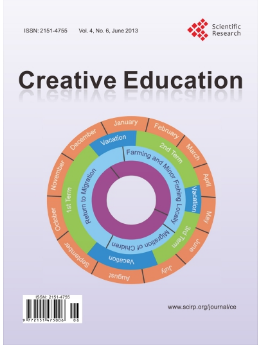
Advances in Bioscience and Biotechnology

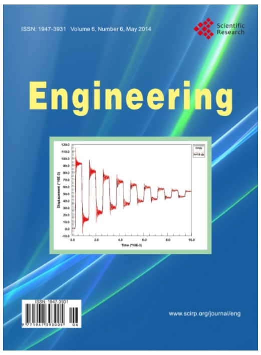
American Journal of Analytical Chemistry

American Journal of Plant Sciences
Special Issue on Photosynthesis

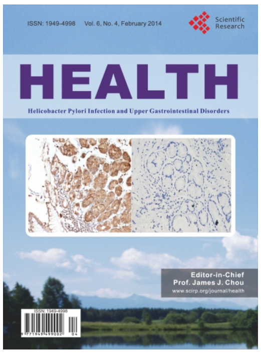

Applied Mathematics
Special Issue on Experimental Design


Agricultural Sciences
Special Issue on Research on Rice


Creative Education

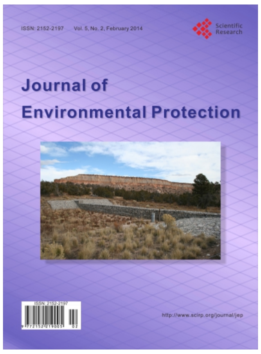

Engineering

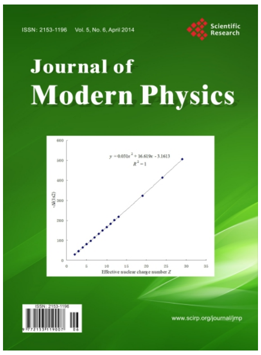
Food and Nutrition Sciences


HEALTH
Helicobacter Pylori Infection and Upper Gastrointestinal Disorders
Editor-in-Chief
Prof. James J. Chou

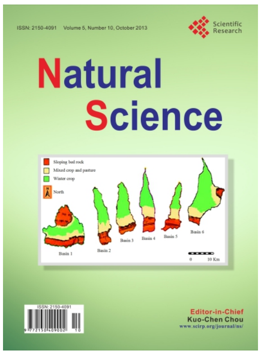
Journal of Computer and Communications

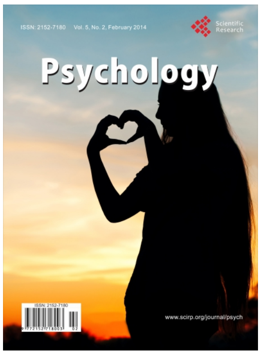
Journal of Cancer Therapy
Special Issue on Bone Tumor Research


Journal of Environmental Protection


Journal of Modern Physics

Modern Economy


Natural Science
Editor-in-Chief
Kuo-Chen Chou


Psychology