Wind Energy Predictions of Small-Scale Turbine Output Using Exponential Smoothing and Feed-Forward Neural Network

Zaccheus O. Olaofe[1, 2]

[1]ZakkWealth Energy

[2]Faculty of Engineering and Built Environment, University of Cape Town

Cape Town, South Africa

zakky201@gmail.com

***Abstract*-This article presents the comparisons of energy production predictions of a small-scale 40 kW wind turbine using an exponential smoothing technique and multilayer feed-forward neural network. For wind energy predictions, the developed mathematical model based on exponential smoothing was used to smoothen any seasonality arising in the time series data obtained at the site. This model was developed using three smoothing constant values of 0.20, 0.65, and 0.90, as well as a combination of a smoothing constant value of 0.90 with a seasonal adjustment factor for prediction of a small-scale wind turbine output for a period of 12 months. In addition, an energy model based on a multilayer feed-forward neural network was used to compute the energy generation of the turbine. The seasonally adjusted forecast model accurately predicted the wind energy output with the lowest forecast errors when compared to the chosen three smoothing constants. The energy forecasts obtained from the seasonal adjusted forecast model and multilayer feed-forward neural network were compared to the actual energy generation of the turbine at the considered tower height in terms of their forecast erroneous values.**

Keywords- Time Series Data (TSD); Smoothing and Seasonal Factor; Exponential Smoothing; Feed-Forward Neural Network (FNN), Small-Scale Wind Turbine

I. INTRODUCTION

Around the globe, small to large-scale wind energy conversion systems have increased in stand-alone and grid-tie energy applications due to the economic benefits of harnessing wind. However, wind intermittency at different locations, which influence the performance of wind energy systems, requires an integrated storage technology to make it a reliable energy compensator during times of limited wind energy generation. One advantage of the use of energy storage technologies with wind energy systems is that it provides solutions to grid-related problems during intermittency. For large-scale renewable energy integration, many industry experts believe the use of energy storage technologies are essential for voltage and frequency regulation, spinning reserve, load leveling, peak shaving, transmission and distribution supports, renewable energy smoothing and ramp rate control, and power quality. The various storage technologies available for small to large-scale energy applications include: pumped hydro storage, compressed air energy storage (CAES), fly wheel storage, molten salt storage, hydrogen storage, sodium-sulfur batteries, advanced lead acid batteries, nickel-cadmium storage, lithium-ion batteries, and redox-flow energy storage. However, few of these energy storage technologies have been implemented with renewable energy in developed countries due to the associated capital and maintenance costs of deploying them [1-2].

Several literature have presented the use of time series techniques and artificial neural networks (ANNs) for the prediction of wind speed and power output of a single or group of turbines (wind farm) at different time horizons. The time series and ANN techniques are based on training historical data taken over an extended period of time to learn the correlation between the weather input parameters (wind speed and direction, temperature, humidity, etc.) and wind energy output of the turbine (WECS). Both techniques are based on pattern recognition, which uses the error function (obtained from the difference between the predicted and the actual wind) to adjust the mathematical model or ANN. The advantage of the time series technique is that it is mathematically easy to use, less expensive to develop and accurate for short-time forecasts. Due to the forecast error involved with increasing time horizons, the time series technique for short to long-term forecasts has been confined. A more robust model was required for accurate short to long-term forests; hence, the ANN was proposed for use in wind forecasting. ANNs have been tested and found to be accurate in seasonal and non-seasonal wind speed and power forecasts. Furthermore, the capability of recognizing patterns between non-linear multivariate data has made ANNs the preferred and most widely used model in engineering and non-engineering applications compared to the time series model. As such, the use of ANNs for wind speed and power predictions has become a reliable forecast technique in wind energy trading applications. An overview of the time series and artificial neural network techniques is discussed in the Section II.

In this article, the wind energy production of a 40 kW turbine for a period of 12 months was predicted based on the exponential smoothing technique and the multilayer feed-forward neural network. The exponential smoothing forecast model is known to be an environmentally adaptive and self-adjusting forecast mathematical model that uses a weighted smoothing

DOI: 10.5963/IJEE0502002

constant and a seasonal factor to perform forecasting. This is similar to the artificial neural network that uses the computed error derivative to tune its forecast system. In addition, the energy models using appropriate model parameters are believed to be accurate for wind energy prediction at proposed wind sites.

The exponential smoothing energy forecast model was developed using three different smoothing constants, as well as a combination of a smoothing constant and seasonal factor to compute the wind power generation at a given time *t*. Also, these three smoothing constants were chosen to determine the influence of the smoothing factor on time series data obtained at the wind site. The response of a smoothing factor in accurate wind prediction depends on the chosen value, as the best smoothing value results in the lowest forecast errors. The single exponential smoothing technique with a smoothing constant is useful for wind prediction not influenced by trends and seasonality. Therefore, the single exponential model performs poorly when utilized for wind forecasts in conditions of trends or seasonality. As a result, a monthly seasonal factor was estimated from the time series data and used to damp any seasonality that may occur in the time series data. For the seasonal adjusted energy forecasts, the following steps were followed: the monthly seasonal factor was obtained from the time series data, the time series data was divided by the estimated seasonal adjustment factors, the forecast energy model was developed, and the seasonal adjustment was made on the forecasts. The exponential smoothing energy model updates the smoothing constant, as well as the seasonal factor, each time a new forecast is made. Two forecast results were obtained from this energy model: the energy prediction based on the use of (i) smoothing constant values 0.90, 0.65, 0.20 only, and (ii) a combination of a smoothing constant value of 0.90 and the estimated seasonal factor. Both forecast results were compared, and the energy forecast result with the lowest prediction errors was chosen as the best smoothing parameter value(s) for the proposed exponential smoothing energy model. The proposed second energy model based on the multilayer FNN was trained using the Levenberg Marquardt back-propagation learning rule to predict the energy output of a 40 kW wind turbine over a period of 12 months. For input wind records at time *t*, *t*+10, *t*+20,……, *t*+*k*, the energy models returned an equivalent of the energy output of the turbine at *t*, *t*+10, *t*+20…...., *t*+*k*. The energy outputs of the turbine were computed by both developed energy models and the results were compared with the actual energy generation at the site. The actual energy records of the turbine for the year 2011 were recorded at 40.871 MWh. The developed mathematical models were validated and the energy model with a smoothing constant value of 0.90 accurately predicted the wind energy potential at 40.865 MWh with an annual mean absolute percentage error of 3.642%, mean absolute error value of 11.120E-3, and root mean square error of 2.004E-2. The multilayer FNN energy model predicted the energy potential at 40.871 MWh with an annual mean absolute percentage error of 0.111%, mean absolute error value of 1.755E-3, and root mean square error of 2.917E-3. The accuracy test was performed on the FNN energy model using a 1-month untrained dataset and the test result was compared to the actual result.

II. OVERVIEW

A. Time Series Technique

The time series technique has received global recognition in financial econometrics and business applications that employ sales forecasts for managerial planning and inventory control. For the model parameter that can be used for tuning the forecast such that the forecast error between the predicted and the actual sales is marginal, the time series technique is widely used. The *time series-based* is known as the conventional statistical technique based on the auto-recursive algorithm. This time series technique is aimed at predicting the future values of a time-sequence series based on the historical records taken at a successive time interval *t*. The forecast skill and accuracy of this technique decreases with increasing time steps especially when trend or seasonal components in the historical time series is involved. As a result, the time series model performs better when used for uniformly spaced, stationary time series as compared to its use for stochastic time series data [3]. In addition, the time series technique can be considered in transformation of seasonal time series to achieve stationarity.

The time series technique has found several applications in sales forecasts, economic planning (for determining the gross domestic product), financial risk management, inventory and stock control, budgeting, monitoring production line and capacity planning in industry, agriculture (for crop plantation and livestock production), meteorology, and wind energy industries (for daily weather and energy forecasts) [4]. The various time series techniques which have been adopted in forecasts include: the moving average (MA), auto regressive moving average (ARMA), auto regressive integrated moving average (ARIMA), exponential smoothing (ES), grey predictor (GP), Kalman filtering, etc. Out of these time series techniques, only the exponential smoothing technique was considered in this article for the development of the forecast energy model, as explained in Section III-A

B. Artificial Neural Network (ANN)

An ANN is an emulator of biological neurons in the brain. It is composed of a number of interconnected processing units called neurons, which have the synaptic weight for storing information acquired through the learning process, as well as making this information available for use when needed or recalled. Like biological neurons, the learning phase takes place during training of the neural network using a set of input and output variables. The arrangement of the neurons in the network layers and the pattern of connections between layers are called the neural network architecture. An ANN is made up of highly interconnected processing units called neurons, which perform the following functions: accept input signals through the input unit(s) layer of the connection link (synaptic weight), summon the weighted input signals to the neurons in the hidden layer

DOI: 10.5963/IJEE0502002

and generating an output from the neurons through an activation function [5-7]. The generated output signals are propagated to other units in the layer (if the network has more than one hidden layer) or sent directly to the output unit(s) of the network in single layer architecture. ANNs are the most widely used technique for the prediction of unknown events, storing and recalling information, detection of trends (patterns) between input and output variables, and for solving cluster-related problems. The advantage of an ANN over the time series technique is its processing units for making independent computations based on input signals, unlike the time series technique, which requires a developed mathematical model for its computations. In addition, an ANN can be used for training cyclic, non-stationary time series to obtain high forecast accuracy as compared to the accuracy of the time series technique. Other merits of ANNs include fault tolerance, self-learning capabilities, forecast accuracy with increasing time steps, etc. The ability of the neural network to handle complicated non-linear problems has made it a powerful tool in various applications such as science and engineering fields for modeling of network systems, recognition of patterns among input and output vectors, control systems and optimization, and signal processing in communication lines [8-11]; in medical research for the prediction of patient recovery based on known medical records over a period of time, learning and controlling arm joint movements, processing and detection of images in ultrasounds and x-rays [12-13]; in hydrology for prediction of quality of water on the earth's surface [14-15]; in financial institutions for asset allocation, predictions, and control of stock market prices, forecasting of revenue and effects of prices changes [16-17]; in sports for forecasting the probability of event outcomes such as horse racing, cricket, and football games, and greyhound racing based on previous sport statistics [18-20].

Secondly, the response of an ANN to input signals is dependent on the pattern of connections (neurons to other neurons) in the network layer(s), connection links (weights), and the activation function(s) in each network layer [21]. The process in which the connection links (weights) are tuned so that the neural network generates an optimal result is called accurate training (or learning). The activation function in the network layer determines the amplitude of the output signal(s) based on the weighted net inputs and the applied bias factor. In addition, the activation function acts as an output signal limiter that transforms these weighted net inputs of the network neuron into an activation state. The activation signal in the neuron(s) can be sent to either the subsequent neurons or directly to the output unit of the network layer. The most common type of activation function used for a single-layer neural network is the step activation function. However, for a multilayer neural network, a combination of the linear (purelin) activation function and non-linear activation function (log-sigmoid or tan-sigmoid) are often used for neural network architecture [22-23]. An overview of the various activation functions used in neural network architecture is discussed below.

C. Step Activation Function

The step activation function is known as the threshold or Heaviside function. This function is used in a simple neural model called a single-layered network. This network function is a result of feeding input signals through one layer of the processing unit(s); no computational speed is required for this network. In addition, this type of function is used where the output network (binary or bipolar) is simply the summation of all weighted input signals of the neuron (net weighted inputs) with a threshold θ_j. The threshold function is 0 if the net input is less than a certain threshold value, and 1 if the net input is greater than or equal to the threshold value.

D. Linear Activation Function

The linear function also known as the purelin function is often used in performing a linear transformation or approximation of the input vector(s). This type of function is widely used in the multilayer neural network where the output(s) of this function do produce any value outside the range of -1 to +1.

E. Nonlinear Activation Function

The nonlinear activation function can be grouped as logistic and Tan-sigmoid, as discussed below. The log-sigmoid and tan-sigmoid functions are examples of the non-linear activation function used in neural networks to learn linear and non-linear relationships between the input and output vectors. In addition, these sigmoid functions are used mainly for performing a nonlinear transformation of input vectors using back-propagation training rules. In a multilayer neural network, the sigmoid function is used to calculate weight updates in certain training rules.

F. Logistic Activation Function

The log-sigmoid function is often used in a multilayer network where the desired output values lie in the range of 0 to +1, as the input values range from negative to positive infinity. The logistic activation function is defined by Eq. (1):

$$f(x) = \frac{1}{1+\exp(-\sigma x)} \tag{1}$$

where σ is the slope (steepness) parameter, and f(x) is the log-sigmoid activation function.

DOI: 10.5963/IJEE0502002

As x tends from negative toward the positive infinity, the sigmoid shape becomes steep and the slope of the sigmoid becomes zero, as explained by Murphy [24]. However, in scenarios where the slope is not equal to zero, the outputs range between the values of 0 and +1.

G. Tan-sigmoid Activation Function

The tangential sigmoid is another multilayer activation function used in back-propagation training rules. The neural inputs of this activation function lie between negative to positive infinity, while the tan-sigmoid are used to constrain the output value within the range of -1 to +1. The tan-sigmoid activation function is defined by Eq. (2):

$$g(x) = \frac{1 - \exp^{(-\sigma x)}}{1 + \exp^{(-\sigma x)}} \tag{2}$$

where g(x) is the tan-sigmoid activation function.

From the above-summarized activation or transfer functions used in ANN architectures, the tan-sigmoid and the linear activation functions were used for the system architecture of the multilayer feed-forward neural network shown in Fig. 2.

Lastly, an ANN can be classified depending on its connection types: (i) the feed-forward neural network classified as a static network (feed-forward network from input unit to output unit); (ii) the feedback neural network: dynamic network characterised by the presence of a backward connection from the hidden or output layer of the network to the input units of the layer dynamic (feed-forward network with a feedback loop from the output or hidden unit into the input unit of the network). The dynamic neural network has been used in several applications because of its ability to store a learnt sequence or pattern during the training phase. This is useful in scenarios where the trained information needs to be recalled for prediction of unknown events based on a known input pattern. However, the dynamic network typically has a longer response time and is more difficult to train compared to the static feed-forward network. The static feed-forward network (FNN) is inherently static in behaviour because of its lack of a feedback element, which provides dynamics to the network. The advantage of the feedback over the static feed-forward network is its feedback loop with a time delay element (Z^{-1}), which provides dynamic response to the network [25]. Three neural examples of dynamic networks used in forecasts are the Jordan, Elman, and Hopfield. The Jordan and the Elman networks are used in wind speed and power predictions, while the Hopfield network is used in power system applications. The Jordan neural network, introduced in 1986, has an activation function of the output layer fed back into the input layer through a set of extra input unit(s) called state units. The Elman recurrent neural network (RNN) was later introduced in 1990, in which the activation signals are fed back from the hidden layer of the network into the input layer through a set of extra input unit(s) called context units [26]. The Elman RNN is a modification of the Jordan network with a sigmoid transfer function in the hidden layer and a purelin transfer function in the output layer of the network. The layered recurrent neural network (LRNN) is another dynamic neural network; it is a modification of the Elman RNN with an arbitrary number of hidden layers and transfer functions in each network layer. The LRNN has a feedback loop with a time-delay element in the hidden layer of the network. The feedback signal from the hidden layer is taken from the real output during the training of the network and fed back into the input layer via context units [27]. The echo state network "ESN" is another type of recurrent neural network with a sparsely-connected random hidden layer. The weights of the output neurons are only considered as part of the network that can be changed and trained [28]. The pipelined recurrent neural network (PRNN) is another recurrent neural network type [29-30] that is used for short-term wind speed predictions based on historical time series weather records. However, the major difficulties of RNNs are the associated scaling problems and computation time. For a large number of input units and neurons, the training of a dynamic network can be difficult. However, this network performs better when used with highly varying or noisy data compared to static feed-forward networks.

An ANN can be classified according to the learning rules or methods in which the training samples are presented to the network, including the supervised learning ANN, unsupervised learning ANN, and re-enforcement ANN. The supervised learning network is trained by provisions of the training dataset (the input and target variables). The input variables are used in network training and the target variables are used for comparisons to determine the accuracy of predictions obtained from output units of the layer. These target variables, often referred to as the expected variables, are crucial in the training of a supervised ANN. Depending on the training algorithm, the error function is calculated and the derivative is used to tune the synaptic weights (updates) the network for optimal training. The supervised learning network can be further grouped into the static and dynamic network. FNNs are widely used in applications such as function approximation, predictions of unknown sequenced events in which time constraints or factors are involved, as well as in non-linear dynamical systems. RNNs are often considered in applications including the classification of noisy and seasonal trend variables and modeling and controlling of dynamic systems, etc. Unlike the supervised learning network, the unsupervised learning network is a self-supervised training network that does not require target variables and the output of the network is not known before training. A self-supervised network is only provided with input variables, and the network itself is allowed to decide what features, initialization, or learning rules it will use to respond to a given input variable. The unsupervised learning network is another important feature of ANNs that is utilized for complicated non-linear problems that are time-consuming or difficult to handle using existing

DOI: 10.5963/IJEE0502002

mathematical models. In addition, the unsupervised learning network is often referred to as a self-organization or adaptive network. Examples of this network are the adaptive resonance theory (ART) and the Kohonen self-organizing feature map (SOFM) [31]. The re-enforcement learning network is another area of the machine learning procedure that learns by interacting with its network environment and getting feedback (response) from the environment in discrete time steps. In this learning procedure, the environment is formulated as a Markov decision process (MDP), and learning only takes place by continuous interaction with the environment. The re-enforcement neural network requires the use of training samples to optimize its performance and a function approximation to cope with its large environments. The re-enforcement learning network focuses on on-line performance as compared to the supervised training network, which focuses on off-line performance of the network.

Fig. 1 shows the structural representation of a simple neural model and the interconnection between its various network elements. The single neural model has a number of input units where each of the input variables ($x_1, x_2, x_3 \ldots\ldots, x_n$) are fed into the neural model through the synaptic weights ($w_1, w_2, w_3 \ldots\ldots w_n$). The weighted net inputs u_j to the neuron is the summation of all the weighted input signals, as expressed in Eq. (3). The summation of weighted inputs makes the input of the activation function to produce an output signal y_o. The output of the neural model is expressed in Eq. (4):

$$u_j = w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots\ldots w_n x_n \tag{3}$$

$$y_o = \varphi\left(u_j + b_j\right) \tag{4}$$

where n is the number of elements in the input vector x, w_n is the associated synaptic weight, b_j is the fixed bias factor that is applied externally to change the net weighted inputs of the activation function, θ_j is the threshold, and φ is the activation function [32-33].

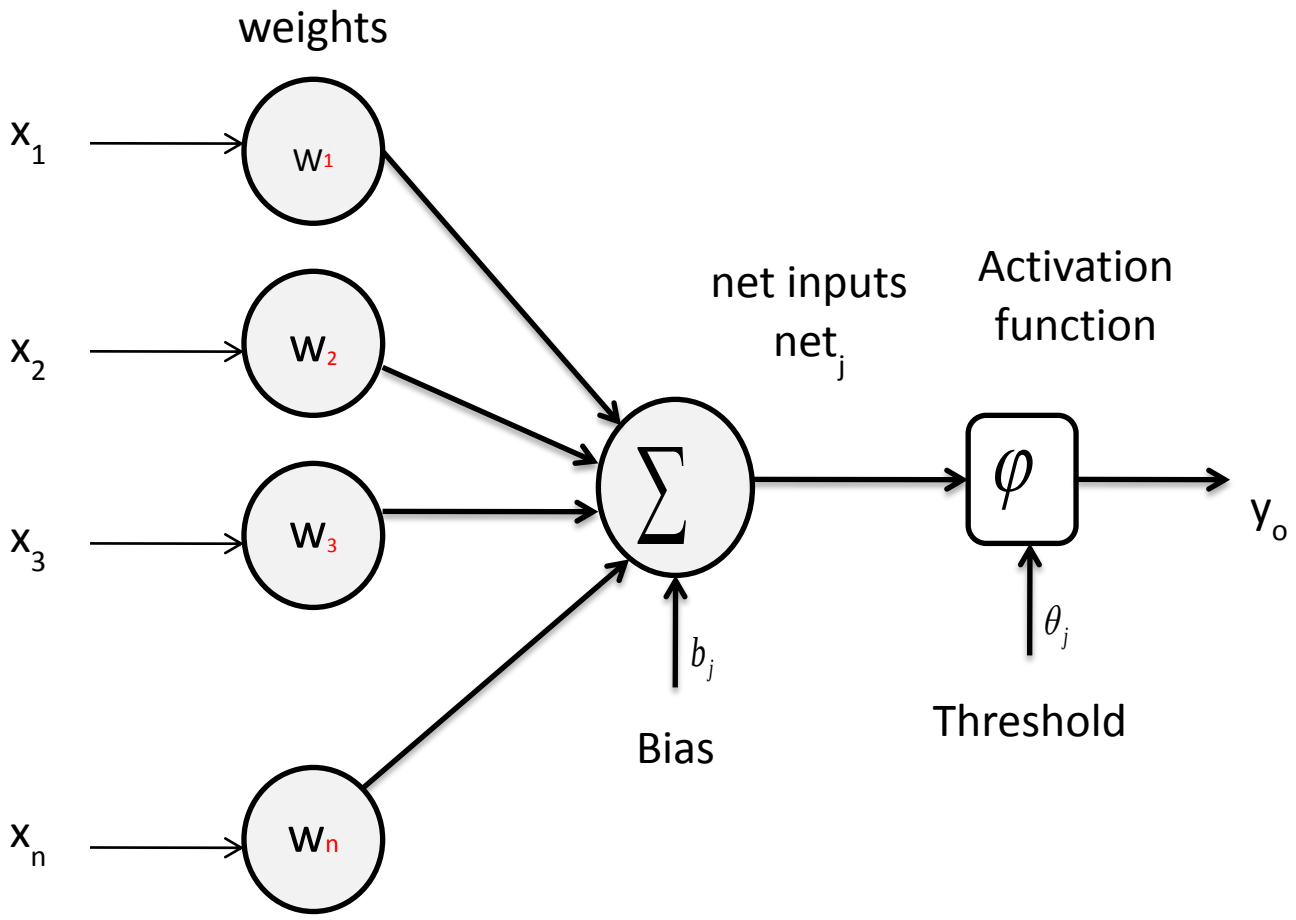


Fig. 1 Structural representation of a simple neural model [5]

In the design of a neural network architecture, the following network parameters are often considered: the learning rate and the momentum gain, the number of layers (hidden and output) in the network, the number of neurons per layer, the number of training epochs or iterations, the activation function type, and training (network) function. The selection of an appropriate

DOI: 10.5963/IJEE0502002

number of processing unit(s) is usually one of the most important parameters considered in a network design. However, for a given task it is essential to define these training parameters as the network performance depends on the chosen network parameters. For example, in a simple network design, one hidden layer can be considered for a simple and linear system, while for a complex network design two or more hidden layers can be considered. In a complex network design, the neural network will be prone to over-fitting and generalization problems. Using a small network design can help to avoid this over-fitting problem, but may result in a poor approximation of the network [34]. One way to solve this problem, as suggested by Macleod, *et al.* [35**]**, is to optimize the network size by varying the number of the input variables, the connection links, and adjusting the neurons in the hidden layers.

III. MODEL DEVELOPMENT FOR WIND ENERGY PREDICTION

One of the most important tasks in developing an accurate forecast model is the selection of important input variables as well as the network parameters that determine the response or accuracy of the forecast model. To estimate the wind power output of the 40 kW turbine for the period of 12 months, 10-minute time series data were obtained on a 20 m mast at the Vredenburg wind site. The longitude and latitude of the wind site are 18.1099 °E and 32.84633 °S. Though the weather records obtained at this site included the wind speed, wind direction, atmospheric pressure, air temperature, humidity, and rainfall records, etc., only important weather parameters that affect the performance of the wind turbine were selected. The criteria utilized for the selection of the input variables for training of the neural model was based on its influence on the energy output of the turbine. The selected input data from the site records for the training of the energy models consisted of the following weather data: the wind speed x_1, the wind direction x_2, the air temperature x_3, and the atmospheric pressure x_4. The geographic location of the Vredenburg site lies within the Western Cape Province of South Africa. The wind site is located at the Western Coastal region with a Mediterranean climate, and is subjected to two ocean currents (the warm Agulhas and the cold Benguela currents that sweep the coastline). Along the West Coast, the chilly Benuela Current from Antarctica sweeps past Cape Town, while the East Coast and Durban benefits from the warm Agulhas Current flowing down from the tropics.

A. Exponential Smoothing (ES) Technique

The exponential smoothing technique was considered for this study to detect the underlying trends and seasonality in the time series data, and to learn how the seasonality of these wind data changed over time. This technique is often used in smoothing trends, irregularities, or seasonal components in a given data because seasonal variations in the time series cannot be easily predicted using moving average (MA) and auto regressive (AR) techniques. As a result, they are usually filtered using the exponential smoothing technique. The exponential smoothing is known to be the most accurate forecast technique among various time series forecast techniques used for damping of time series data with trends and seasonality (seasonal components).

The exponential smoothing technique is often defined as a weighted moving average forecast technique that is based on an unequal allocation of weights to the time series data with a smoothing constant and a seasonal factor. The use of an exponential smoothing technique is more complicated than the simple moving average (MA) technique because greater weights are given to the most recent data while lesser weights are given to past time series data, which is different than the equal weight allocation to past data in the moving average technique. In addition, the weights allocated to the past data decline in an exponential manner with increasing forecast times, i.e., greater weights are given to the recent forecasts and takes less consideration of the long past historical events or forecasts [36]. With an unequal assignment of weights to recent values, it is easier to adjust the noise or forecast errors in past values to tune the ES model for future forecasts.

Letting $P_1, P_2, P_3, \ldots P_n$ be the past wind power values at a successive time t, the single exponential smoothing equation for the period t is defined by Eq. (5):

$$\begin{aligned} M_t &= \mu P_t + \mu(1-\mu)P_{t-1} + \mu(1-\mu)^2 P_{t-2} + \ldots\ldots\mu(1-\mu)^n P_{t-n} \\ &= \mu P_t + (1-\mu)\left[\mu P_{t-1} + \mu(1-\mu)P_{t-2} + \ldots\ldots\mu(1-\mu)^{n-1} P_{t-n}\right] \\ &= \mu P_t + (1-\mu)M_{t-1} \end{aligned} \tag{5}$$

where M_{t-1} is the exponential smoothing of the previous weighted moving average; μ is the weighting or damping factor known as the smoothing constant, which is used to smoothen the time series data; P_t is the noisy wind power value at time t; and M_t is the smoothed value at any time period t.

From Eq. (5), the new forecast value for periods beyond time t is defined by Eq. (6):

$$\hat{Y}_{t+1} = \mu P_t + (1-\mu)\hat{P}_t \tag{6}$$

For time series data with trends, the smoothed values tend to lag behind. As a result, double exponential smoothing is used to capture or smoothen the trends in data, as defined by Eq. (7):

DOI: 10.5963/IJEE0502002

$$M_t = \mu P_t + (1-\mu)(M_{t-1} + \psi_{t-1})$$
$$\psi_t = \gamma(M_t - M_{t-1}) + (1-\gamma)\psi_{t-1} \quad (7)$$

where γ is the second parameter that controls the smoothening of the trend.

Note: For time series data with seasonality, triple exponential smoothing its applicable.

The smoothing constant value ranges between 0.0 - 1.0 and denotes how the weights are allocated to the time series data. For example, a value of $\mu = 0.3$ denotes that a slow damping and less weight given to the past value, while a higher weight is given to the most recent value and indicates a quick damping. The rate at which the past time series data are smoothed is dependent on the smoothing constant value with or without the seasonal factor. There is no defined criterion for the selection of the value of the smoothing constant; however, this depends on the time series data being considered. Several methods have been used to estimate the exponential smoothing parameters from a given time series data. The trial and error method, the least square method, or the mean square error method can be used in the selection of the value of μ. These techniques are not discussed in this study; however, the smoothing constant values were chosen based on the trial and error method. A smoothing constant value close to 1 has a less smoothing effect on past series data (that is, it reduces the damping effect on the past weighted moving average), and a greater effect on the most recent value. For value of μ close to 0, it has a greater smoothing effect on the past weighted moving average and less response to the recent forecast.

For a given time series data with seasonality, exponential smoothing forecasts can be made using a combination of smoothing constant and estimated seasonal factor values. To develop an exponential smoothing model, the following steps were taken: (i) A single exponential smoothing equation with three smoothing constant values of 0.90, 0.65, and 0.2 were considered as summarized in Table 1A-C; (ii) A combination of the smoothening constant value of 0.90 and estimated seasonal factor were considered as the exponential smoothing parameters. This seasonal adjusted exponential smoothing approach was used for wind power forecasts at time step *t*, as summarized in Table 1D. The seasonal factor was estimated from the time series data and the estimated mean seasonal factors were used to develop the forecast model, as well as for adjusting the seasonality in the forecast. The developed forecast energy model returned the forecast values of the dependent weather variables based on the chosen damping/smoothing constant values, as summarized in Table 1A-D. The forecast result with the lowest value of the MAPE, MAE, and RMSE, as summarized in Fig. 1a-c, was chosen as the smoothing constant that best fit the single exponential smoothing model for the energy forecasts of a 40 kW wind turbine. The monthly wind energy predictions of a small-scale 40 kW turbine based on a single exponential smoothing with smoothing values of 0.90, 0.65, and 0.2 were compared to the seasonally adjusted forecast approach (single exponential smoothing with a smoothing value of 0.90 coupled with a seasonal factor). The validation of the energy model showed that there were underlying seasonal components in the obtained 12-month time series data, as only the seasonal adjusted forecast model was useful for smoothing the seasonality in this data.

Using a smoothing constant $\mu = 0.90$

TABLE 1A COMPARISONS OF MONTHLY WIND ENERGY OUTPUTS OF THE 40 KW TURBINE USING EXPONENTIAL SMOOTHING CONSTANT ONLY; AND THE ESTIMATED MEAN ABSOLUTE PERCENTAGE ERROR (MAPE), ROOT MEAN SQUARE ERROR (RMSE), AND MEAN ABSOLUTE ERROR (MAE) FOR THE YEAR 2011

Month	***Actual Energy (MWh)***	***Estimated Energy (MWh)***	***MAPE (%)***	***MAE***	***RMSE***
Jan	5.330	5.330	3.666	13.595E-3	2.374E-2
Feb	3.692	3.692	3.936	11.395E-3	1.983E-2
Mar	3.257	3.257	4.044	11.565E-3	1.974E-2
Apr	2.682	2.682	4.119	10.698E-3	1.814E-2
May	2.041	2.040	3.746	9.037E-3	1.874E-2
Jun	3.159	3.159	1.337	6.408E-3	1.373E-2
Jul	1.754	1.753	3.837	8.297E-3	1.689E-2
Aug	2.291	2.291	4.040	10.106E-3	1.798E-2
Sep	2.805	2.804	3.840	11.630E-3	2.023E-2
Oct	2.826	2.826	3.893	10.910E-3	1.885E-2
Nov	5.410	5.409	3.889	16.142E-3	2.802E-2
Dec	5.624	5.624	3.356	13.655E-3	2.458E-2
Sum/Aver.	**40.870**	**40.865**	**3.642**	**11.120E-3**	**2.004E-2**

DOI: 10.5963/IJEE0502002

Using a smoothing constant $\mu = 0.65$

TABLE 1B COMPARISONS OF MONTHLY WIND ENERGY OUTPUTS OF THE 40 KW TURBINE USING EXPONENTIAL SMOOTHING CONSTANT ONLY; AND THE ESTIMATED MEAN ABSOLUTE PERCENTAGE ERROR (MAPE), ROOT MEAN SQUARE ERROR (RMSE), AND MEAN ABSOLUTE ERROR (MAE) FOR THE YEAR 2011

Month	***Actual Energy (MWh)***	***Estimated Energy (MWh)***	***MAPE (%)***	***MAE***	***RMSE***
Jan	5.330	5.328	14.225	49.799E-3	8.548E-2
Feb	3.692	3.691	15.659	42.789E-3	7.389E-2
Mar	3.257	3.255	16.054	42.435E-3	7.067E-2
Apr	2.682	2.680	16.219	39.087E-3	6.502E-2
May	2.041	2.036	14.275	32.819E-3	6.636E-2
Jun	3.159	3.158	3.322	11.335E-3	2.536E-2
Jul	1.754	1.749	14.509	29.715E-3	5.871E-2
Aug	2.291	2.288	15.767	36.033E-3	6.373E-2
Sep	2.805	2.803	14.501	41.225E-3	7.081E-2
Oct	2.826	2.824	14.866	39.206E-3	6.701E-2
Nov	5.410	5.408	15.322	58.720E-3	9.955E-2
Dec	5.624	5.624	13.039	50.201E-3	8.895E-2
Sum/Aver.	**40.870**	**40.844**	**13.980**	**39.447E-3**	**6.963E-2**

Using a smoothing constant $\mu = 0.20$

TABLE 1C COMPARISONS OF MONTHLY WIND ENERGY OUTPUTS OF THE 40 KW TURBINE USING EXPONENTIAL SMOOTHING CONSTANT ONLY; AND THE ESTIMATED MEAN ABSOLUTE PERCENTAGE ERROR (MAPE), ROOT MEAN SQUARE ERROR (RMSE), AND MEAN ABSOLUTE ERROR (MAE) FOR THE YEAR 2011

Month	*Actual Energy (MWh)*	*Estimated Energy (MWh)*	*MAPE (%)*	*MAE*	*RMSE*
Jan	5.330	5.313	70.006	18.197E-2	29.205E-2
Feb	3.692	3.690	27.616	4.279E-2	6.684E-2
Mar	3.257	3.240	78.723	15.029E-2	24.005E-2
Apr	2.682	2.655	74.219	13.442E-2	21.172E-2
May	2.041	2.006	54.753	10.825E-2	20.159E-2
Jun	3.159	3.150	14.599	2.724E-2	5.407E-2
Jul	1.754	1.719	52.374	8.966E-2	16.756E-2
Aug	2.291	2.260	65.513	11.553E-2	20.014E-2
Sep	2.805	2.786	59.178	13.463E-2	22.285E-2
Oct	2.826	2.808	64.175	13.425E-2	22.085E-2
Nov	5.410	5.388	77.855	21.180E-2	33.365E-2
Dec	5.624	5.621	68.108	18.809E-2	31.150E-2
Sum/Aver.	40.870	40.638	58.927	12.658E-2	21.024E-2

Using a smoothing constant 0.90 with a seasonal factor

TABLE 1D COMPARISONS OF MONTHLY WIND ENERGY OUTPUTS OF THE 40 KW TURBINE USING EXPONENTIAL SMOOTHING SEASONAL ADJUSTED FORECAST APPROACH; AND THE ESTIMATED MEAN ABSOLUTE PERCENTAGE ERROR (MAPE), ROOT MEAN SQUARE ERROR (RMSE), AND MEAN ABSOLUTE ERROR (MAE) FOR THE YEAR 2011

Month	***Actual Energy (MWh)***	***Estimated Energy (MWh)***	***MAPE (%)***	***MAE***	***RMSE***
Jan	5.330	5.330	0.688	5.327E-3	0.906E-2
Feb	3.692	3.692	1.468	8.629E-3	1.584E-2
Mar	3.257	3.257	0.481	2.673E-3	0.445E-2
Apr	2.682	2.682	0.534	2.419E-3	0.423E-2
May	2.041	2.040	1.363	5.719E-3	1.204E-2
Jun	3.159	3.159	0.668	4.384E-3	1.080E-2
Jul	1.754	1.753	2.099	7.309E-3	1.624E-2
Aug	2.291	2.291	0.698	3.030E-3	0.750E-2
Sep	2.805	2.804	0.647	3.328E-3	0.776E-2
Oct	2.826	2.826	0.753	3.717E-3	0.739E-2
Nov	5.410	5.409	0.543	4.733E-3	1.036E-2
Dec	5.624	5.624	0.286	2.486E-3	0.469E-2
Sum/Aver.	**40.870**	**40.865**	**0.852**	**4.480E-3**	**0.920E-2**

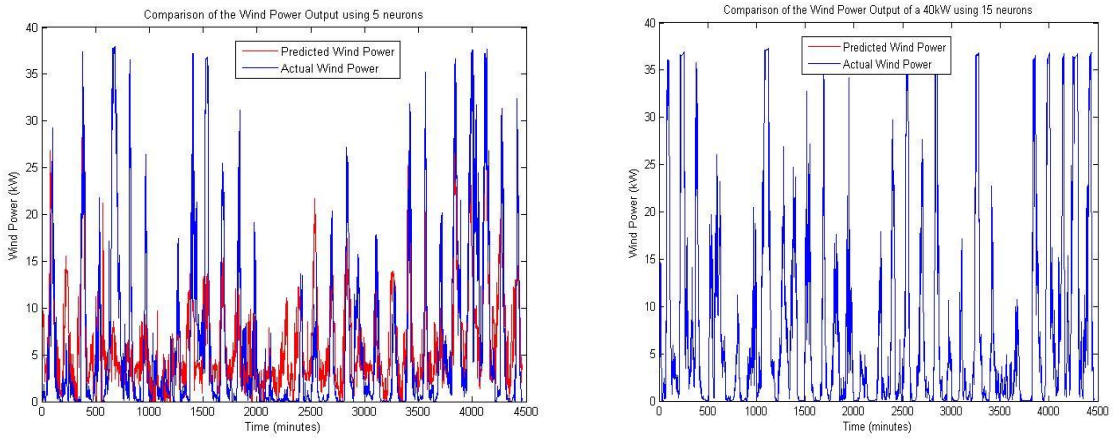
B. Multilayer Feed-Forward Neural Network

A multilayer feed-forward neural network with the following network parameters was selected for the development of the energy model: two-layer network consisting of 1 hidden and 1 output layer; 4 input units; 5-15 neurons (default value of five neurons and due to the poor generalization of the network during the training of the model as shown in Fig. 2A, the number of neurons was increased to a maximum of fifteen, as shown in Fig. 2B); 30 training epochs for convergence and generalization

DOI: 10.5963/IJEE0502002

of the network; tan-sigmoid (non-linear) activation function for the hidden layer and the purelin (linear) activation function for the output layer; training momentum gain value of 0.03 to prevent the network from converging to a local minimum, as well as controlling the learning speed of the neural network; the learning rate value of 0.06 to control rate at which the weight sizes were adjusted during the training of the network. The optimization of network size by the selection of appropriate network parameters tends to solve the problem associated with over-fitting and poor generalization.

The energy model based on the multilayer feed-forward neural network (FNN) was trained using the supervised Levenberg Marquardt back-propagation algorithm. The system architecture of the FNN used to develop the energy model is shown in Fig. 3. A total of 210,480 weather data was processed and the data divided into 2 datasets: 75% of the total dataset were used for training of the network, while the remaining 25% were used for stopping the training process at the point of generalization (for validating the network performance).



(a) (b)

Fig. 2 Comparison of the predicted vs. the actual wind power outputs using (a) 5; and (b) 15 neurons; and the estimated RMSE of 4.56E-02 and 1.79E-05, respectively

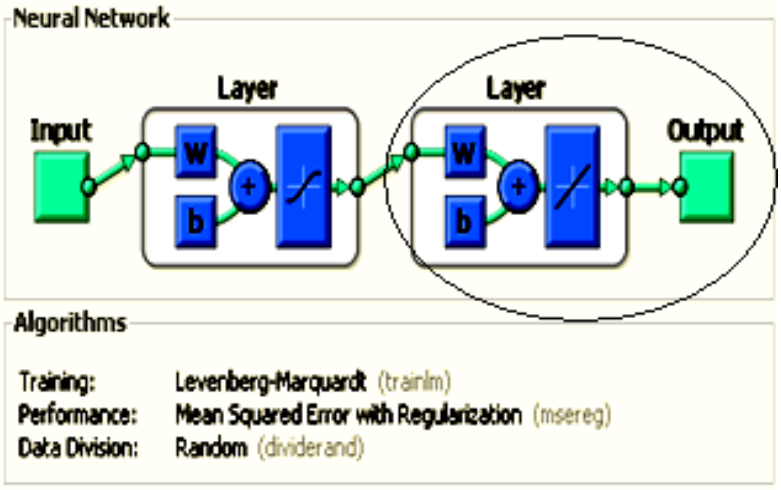


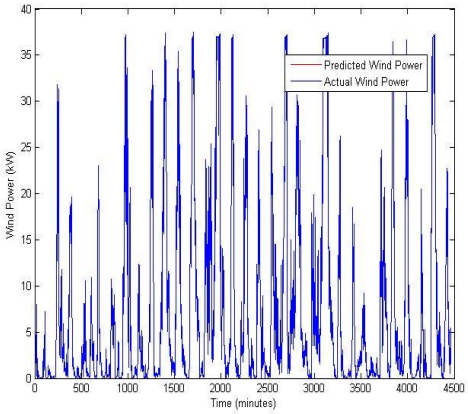
Fig. 3 The system architecture of a multilayer feed-forward neural network

For the training phase, the training dataset were fed through the input units of the network. Then each processing unit made its own computation based on its net weighted inputs and passed the result into the subsequent layer. For each training sample, the Levenberg Marquardt back-propagation algorithm computed the error derivative of the weights. The Levenberg Marquardt training function was used to optimize the network performance by updating the connection weights in order to reduce the error between the predicted and actual power output. These error values were only back-propagated if they exceeded the threshold (tolerance) error value as specified in the network design. The training process continued for a maximum of 30 iterations (epochs), but stopped at a point of generalization. The criteria used to stop the training process of the network were important depending on whether the network was optimally trained or if over-fitting occurred during the training. The gradient and the number of validation checks were used by the network to terminate the training process. The number of validation checks denoted the number of successive epochs that the validation performance failed to decrease. If this validation check reached a specified value in the network, the training automatically stopped. Fig. 4 shows the comparisons of the monthly wind energy prediction of a 40 kW turbine at this site for the period of 12 months.

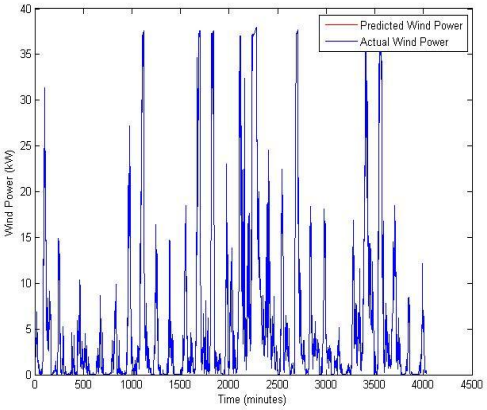
DOI: 10.5963/IJEE0502002

Table 2 shows the summary of usable monthly energy generation of the turbine for 2011 using the multilayer FNN energy model. The monthly energy prediction showed that the month of July had the lowest energy generation estimated at 1.754 MWh, followed by the month of May. The energy generation in the month of July showed that only a fraction of the energy was available for utilization, and was generated within a period of between 3,204-3,504 minutes (534-584 hrs) and 3,804-4,026 minutes (634-671 hrs). The month of December had the highest wind energy potential, followed by the month of November. The annual energy generation of the 40 kW turbine for the year 2011 was estimated at 40.871 MWh, as summarized in Table 2.

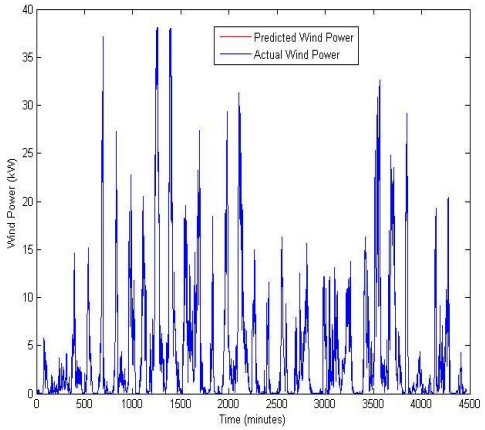
JAN 2011



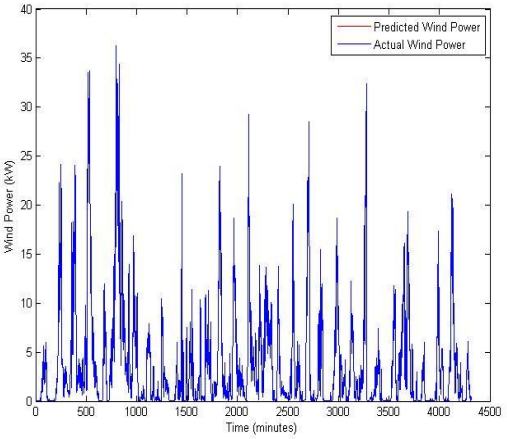
FEB 2011



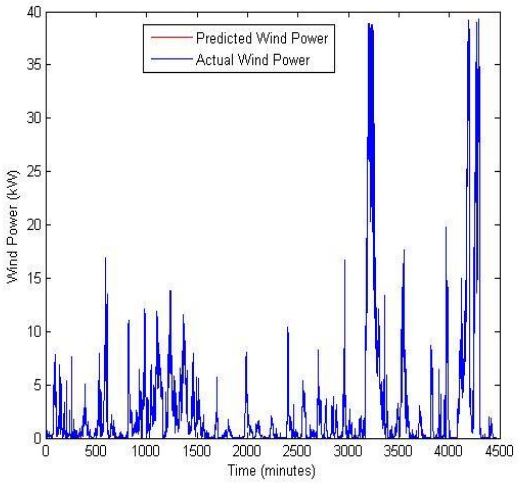
MAR 2011



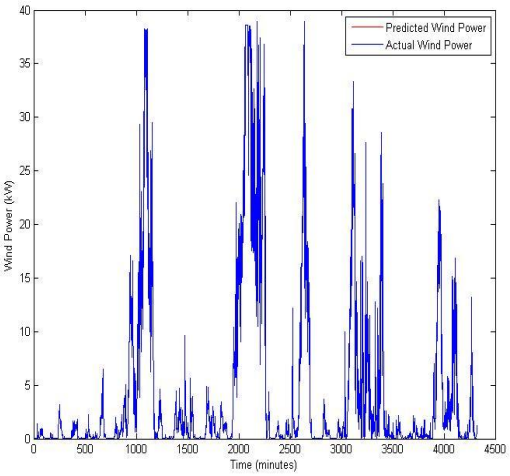
APR 2011



MAY 2011



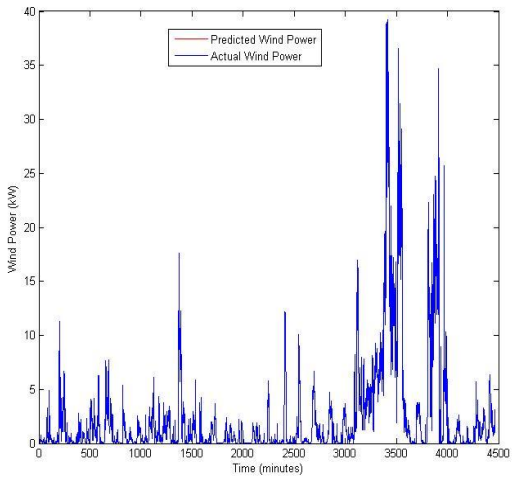
JUN 2011



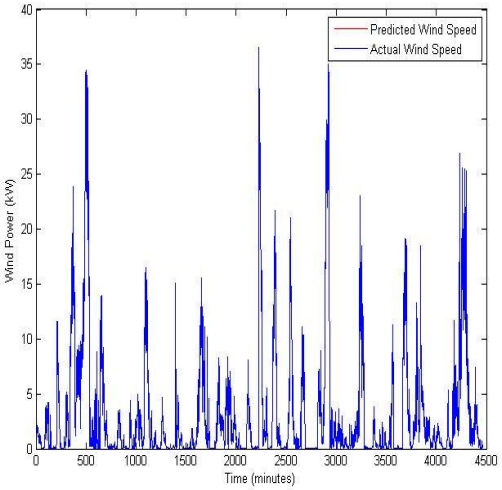
JULY 2011

AUG 2011

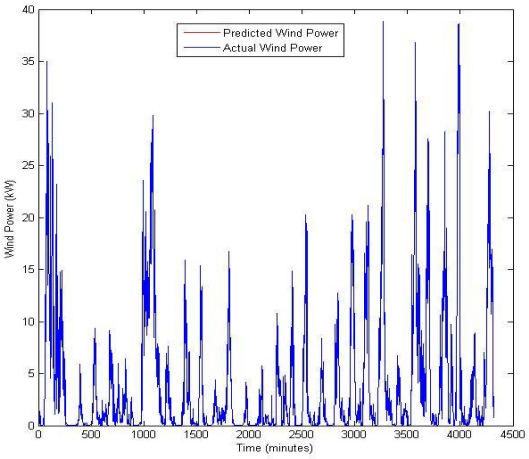
DOI: 10.5963/IJEE0502002



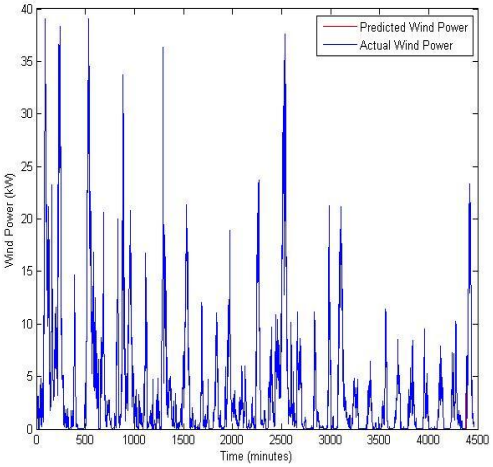
SEPT 2011



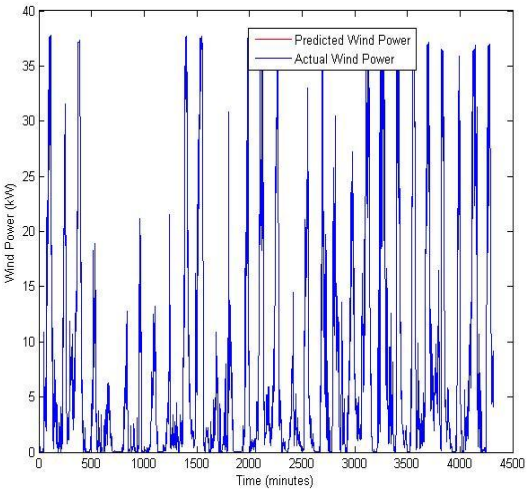
OCT 2011



NOV 2011



DEC 2011



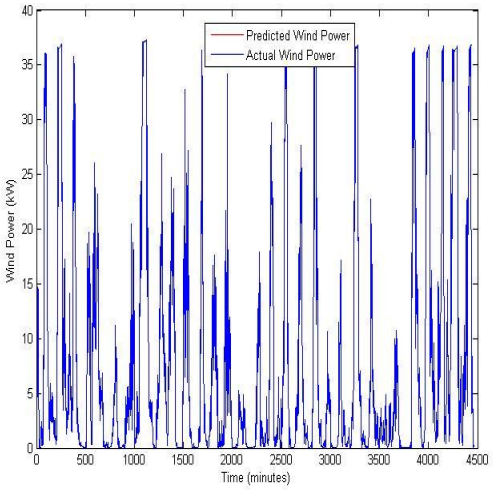


Fig. 4 Comparison of the predicted and target power output

DOI: 10.5963/IJEE0502002

TABLE 2 COMPARISONS OF THE ESTIMATED MONTHLY OUTPUTS OF THE 40 KW TURBINE USING THE FNN FOR THE YEAR 2011; AND THE ESTIMATED MEAN ABSOLUTE PERCENTAGE ERROR (MAPE), ROOT MEAN SQUARE ERROR (RMSE), AND MEAN ABSOLUTE ERROR (MAE), AS WELL AS ITS COMPUTATIONAL TIME

Month	*Actual Energy (MWh)*	*Estimated Energy (MWh)*	*MAPE (%)*	*MAE*	*RMSE*	*Computational time*
Jan	5.330	5.330	0.170	2.474E-3	4.050E-3	05:31mins
Feb	3.692	3.692	0.100	0.825E-3	1.127E-3	05:38 mins
Mar	3.257	3.257	0.006	0.053E-3	0.081E-3	05:30 mins
Apr	2.682	2.682	0.156	3.524E-3	5.212E-3	05:31 mins
May	2.041	2.042	0.153	1.993E-3	3.133E-3	05:30 mins
Jun	3.159	3.158	0.056	0.676E-3	0.969E-3	05:32 mins
Jul	1.754	1.754	0.106	1.470E-3	2.928E-3	05:29 mins
Aug	2.291	2.290	0.006	0.072E-3	0.133E-3	05:36 mins
Sep	2.805	2.805	0.013	0.155E-3	0.185E-3	05:35 mins
Oct	2.826	2.826	0.401	8.439E-3	1.510E-2	05:30 mins
Nov	5.410	5.410	0.032	0.212E-3	0.317E-3	05:31 mins
Dec	5.624	5.625	0.131	1.168E-3	1.779E-3	05:30 mins
Sum/Aver.	**40.870**	**40.871**	**0.111**	**1.755E-3**	**2.917E-3**	**05:32mins**

1) Validation Phase:

Upon the completion of the training process, the performance of the FNN energy model was validated. The validation dataset (remaining 25%) were used to stop the training early if the maximum number of epochs (repetitions) was reached, or the network performance on the validation dataset failed to improve or remained the same for maximum fail epochs. The testing dataset was used for evaluation of the model to ensure that the network generalized well, but this did not have any effect on the network training. There were several error measurement techniques that can be used as the metrics for evaluating the accuracy of the developed energy models. They include: the mean square error (MSE), correlation coefficient (R), mean absolute error (MAE), mean square relative error (MSRE), root mean square error (RMSE), standard deviation of the absolute error (Std.), and coefficient of determination (COD) [34]. In this study, only the MAPE, MAE, and the RMSE were used as metrics for validating the accuracy of the energy model based on the multilayer FNN. These metrics provide a means of validating and choosing the best training results. Columns 4-6 in Table 2 show the summary of the monthly errors obtained from the trained network, and the results were used as metrics to determine how well the neural network was able to work during training of the network using the dataset. The closer the estimated values were to the actual values, the lower the training error, and hence the more accurate the trained network (prediction). The accuracy of the exponential smoothing and multilayer neural network were validated using error metrics such as the MAPE, MAE, and the RMSE; and the results are summarized in Columns 4-6 of Tables 1 and 2. The MAE was used to measure the closeness of the estimated energy outputs to the actual outputs of the 40 kW turbine; and the MSE was the amount of error by which the predicted values differed from the actual values. The month of March had the lowest training error, while the month of October had the highest training error followed by the month of January. This was a result of the network's inability to recognize the existing relationship between the input and power output during the training process in both months. Hence, the training errors increased during these two months of energy generation prediction.

2) Testing Phase:

To test the accuracy of the trained wind energy model, a new dataset consisting of the air density, wind speed distribution, and humidity were obtained and fed as the new inputs into the model. This new dataset differed from the training dataset and was obtained as 10-minute weather records. Ten percent of the total dataset was used for training of the network, while ninety percent of the dataset were used for testing the performance of the multilayer FNN energy model. Fig. 5 shows the performance result of the energy model using an independent dataset for the month of December 2011. The energy model gave a good response when presented with an untrained (unseen) dataset. Comparing Fig. 4L (Dec 2011) to Fig. 5, the energy model predicted the wind energy outputs of the turbine based on the inputed air density, wind speed distribution, and humidity data. Furthermore, the testing result of the FNN energy model showed a similarity to the actual energy generation at this site for the month of December 2011.

DOI: 10.5963/IJEE0502002

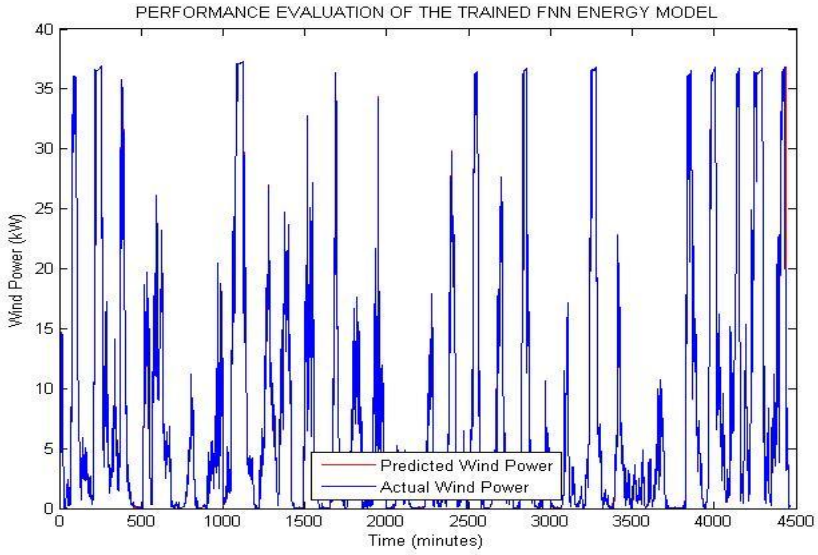


Fig. 5 Performance evaluation of the multilayer FNN energy model

IV. CONCLUSION

Energy models based on exponential smoothing (smoothing constant only and the seasonal adjusted factor) and multilayer FNN were presented for determining the energy potentials of a 40 kW turbine at the Vredenburg wind site. The obtained weather data with the selected model parameters played a significant role in accuracy energy predictions of the turbine. For stationary time series data not influenced by trend and seasonality, the single exponential energy model performed well; however, for a stochastic time series with seasonality, the model performed poorly. As a result, an exponential smoothing seasonal adjusted approach was considered for optimization of the network for seasonality in the time series. Similar to neural network architecture, a large dataset with the adjustment of the selected network parameters were required for optimal training. In the development of energy models using a large historical dataset, it may be computationally time intensive to obtain accurate forecast results within a very short period of time, and could lead to network over-training or over-fitting with the wrong selection of model parameters. However, a sufficient dataset should be presented to the energy models in order to capture most of the underlying trends and seasonal components in the time series to improve the performance of the energy models.

The energy outputs of a 40 kW turbine based on three exponential smoothening constants, as well as the use of monthly seasonal factors, were compared as summarised in Tables 1A to 1D. The forecast results comparison showed that the use of a single smoothing constant was only accurate for smoothing the time series input data without trends or seasonality, while the use of the seasonal adjustment forecast approach was accurate for smoothening the time series weather data with seasonality. The comparisons of the energy outputs of the turbine summarized in Tables 1A-C show that the smoothing constant of 0.90 had the greatest smoothing effects on the time series data as compared with the allocation of 0.20 weighting constant to the same single exponential smoothing model. For noisy time series data in some months, the prediction error increased using a smoothing constant only for the developed forecast model. As a result, the use of seasonal adjusted forecast model based on the estimated seasonal factors was found to be more accurate for smoothening the seasonal components in each month of the year. The energy model with only a single exponential smoothing constant value of 0.90 accurately predicted the energy potential at 40.865 MWh with an annual mean absolute percentage error of 3.642%, mean absolute error value of 11.120E-3, and root mean square error of 2.004E-2. However, using a seasonally adjusted forecast model based on the estimated seasonal factor, the exponential smoothing model predicted the energy potential at 40.865 MWh with an annual mean absolute percentage error of 0.852%, mean absolute error value of 4.480E-3, and root mean square error of 0.920E-2.

The usable wind energy outputs of the turbine using a multilayer FNN energy model are summarized in Column 3 of Table 2. The monthly wind energy predictions showed a correlation with the actual wind energy generation at the site except for the months of January and October, where the prediction error increased due to the poor training of the FNN energy model. Though the monthly energy generation at the wind site was very close to the actual energy generation, the aggregated energy generation errors for both months (January and October) were a result of the accumulated over-prediction of the model at different time of the months. The multilayer FNN energy model predicted the energy potential of the 40 kW turbine at 40.871MWh with an annual mean absolute percentage error of 0.111%, mean absolute error value of 1.755E-3, and root mean square error of 2.917E-3. Although the use of multilayer FNNs are not the best when utilized for wind forecasts when compared to a multilayer RNN, their implementation outperforms all other time series techniques used for stochastic time

DOI: 10.5963/IJEE0502002

series forecast. Other advantages of multilayer FNNs over exponential smoothing models are the high computational speed, no required mathematical model development, and its ability to handle stochastic time series data.

In conclusion, the inability to manage wind variability through the development of an accurate energy forecast model will lead to additional costs to maintain the reliability of the power grid; energy imbalance (between demand and supply) and potential risk to the power system network; limitations in the share of wind energy technologies; and heavy reliance on conventional power plants, etc. However, an accurate energy prediction will enhance the optimal integration of wind energy technologies into the grid, enabling the increased reliance on wind energy technologies.

REFERENCES

[1] N. C. Nair and N. Garimella, "Battery energy storage systems: Assessment for small-scale renewable energy integration," *Energy and Buildings*, vol. 42, pp. 2124-2130, 2010

[2] P. J. Hall and E. J. Bain, "Energy-storage technologies and electricity generation," *Energy Policy*, vol. 36, no. 12, pp. 4352-4355, 2008.

[3] D. R. Brillinger, "Time Series Data Analysis and Theory," ISBN-10: 0898715016, p. 540, Jul. 19, 2010

[4] C. Chatfield, "Time Series Forecasting," ISBN 1-58488-063-5, p. 265, CRC Press LLC, New York Washington, D.C., 2000.

[5] M. K. Deshmukh and CB. Moorthy, "Application of Genetic Algorithm to Neural Network Model for Estimation of Wind Power Potential," *Journal of Engineering, Science and Management Education*, vol. 2, pp. 42-48, 2010

[6] M. A. Nayak and M. C. Deo, "Wind Speed Prediction by different Computing Techniques," Conference of BALWOIS 2010, Ohrid, Republic of Macedonia, May 25-29, 2010.

[7] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial Neural Networks: A Tutorial," Michigan State University, March 1996.

[8] L. Y. Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and LD. Jackel, "Handwritten Digit Recognition with a back-propagation Network," DS. Touretsky, Ed., Advances in Neural Information Processing Systems 2,San Mateo, CA, pp. 396-404, 1990.

[9] D. Nguyen and B. Widrow, "The Truck Backer-Upper: An example of Self-Learning in Neural Networks," *International Joint Conference on Neural Networks*, Washington DC II, pp. 357-363, 1989.

[10] B. Widrow and S. D. Stearns, "Adaptive Signal Processing," Englewood, Cliffs, NJ: Prentice Hall, 1985.

[11] P. Jain and M. C. Deo, "Neural Networks in ocean engineering," SAOS 2006, vol. 1, no. 1, pp. 25-35, Woodhead publishing Ltd., 2006.

[12] N. Lan, "Neural network generation of muscle stimulation patterns for control of arm movements," *IEEE Transaction on Rehabilitation Engineering*, vol. 2, no. 4, pp. 213-224, December 1994.

[13] J. A. Anderson, R. M. Golden, and G. L. Murphy, "Concepts in Distributed Systems," HH Szu, Ed., *Optical and Hybrid Computing, Society of Photo Optical Instrumentation Engineers*, vol. 634, pp. 260-272, 1986.

[14] R. S. Govindaraju and A. R. Rao, "Artificial Neural Networks in Hydrology," *Water Science and Technology Library*, vol. 36, pp. 348, 2000, publisher: Springer, ISBN 978-0-7923-6226-5.

[15] A.R. Rao and EC. Hsu, "Hilbert-Huang Transform Analysis of Hydrological and Environmental Time Series," *Water Science and Technology Library*, vol. 60, no. 12, 2008, ISBN 978-1-4020-6453-1.

[16] S.A. Hamid and Z. Iqbal, "Using neural networks for forecasting volatility of S&P 500 Index futures prices," *Journal of Business Research*, vol. 57, no. 10, pp. 1116-1125, October 2004.

[17] J. H. Stock and M. W. Watson, "New Indexes of Coincident and Leading Economic Indicators," *Handbook of the National Bureau of Economic Research*, MIT Press, vol. 4, pp. 351-409, 1989, ISBN: 0-262-02296-6.

[18] A. D. Blaikie, G. J. Abud, J. A. David, and R. D. Pasteur, "NFL & NCAA Football Prediction using Artificial Neural Networks," *Proceedings of the 2011 Mid-states Conference on Undergraduate Research in Computer Science and Mathematics*, 2011.

[19] S. R. Iyer and R. Sharda, "Prediction of athletes' performance using neural networks: An application in cricket team selection," *Expert Systems with Applications*, vol. 36, iss. 3, pp. 5510-5522, April 2009.

[20] H. Chen, P. B. Rinde, L. She, S. Sutjahjo, C. Sommer, and D. Neely, "Expert prediction, symbolic learning, and neural networks. An experiment on greyhound racing", *IEEE Explore*, vol. 9, no. 6, pp. 21-27, 1994.

[21] L. Fausett, "Fundamentals of Neural Networks: Architectures, Algorithms and Applications," Prentice Hall International, Inc., 1994, USA, ISBN 0-13-042250-9.

[22] http://en.labs.wikimedia.org/wiki/Artificial_Neural_Networks/Activation_Functions.

[23] J. W. Hines, "Fuzzy and Neural Approaches in Engineering," A Wiley-Interscience Publication, John Wiley & Sons, Inc. 1997.

[24] S. Murphy "Development of neural networks for system identification," Final Project Report School of Electronic Engineering, Dublin City University. [Online]. Available: www.eeng.dcu.ie/~brutonj/Reports/SiobhanMurphy02.pdf.

[25] K. P. Mohandas and A. Deepthy, "Partial Recurrent Neural Networks for Identification and Control of Non-Linear Systems," *Proceedings of the IASTED international Conference Control and Applications*, August 12-14, 1998, USA.

[26] B. Krose and P. Van Der Smagt, "An Introduction to Neural Network," University of Amsterdam, Kruislaan, Netherlands, November 1996.

[27] Z. O. Olaofe, "A 5-Day Wind Speed &Power Forecasts using a Layer Recurrent Neural Network," *Sustainable Energy Technologies and Assessments*, vol. 6, pp. 1-24, 2014.

[28] H. Jaeger, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, pp. 78-80, 2004.

DOI: 10.5963/IJEE0502002

[29] P. D. Mandic and J. A. Chambers, "Toward an Optimal PRNN-Based Nonlinear Predictor," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, 1999.

[30] S. Haykin, "Neural Networks: A comprehensive foundation," 2nd ed., Prentice Hall International Inc., 1999, ISBN: 13-978-81315-0488-8.

[31] G. Cheng, T. Liu, J. Han, and Z. Wang, "Towards Growing Self-Organizing Neural Networks with Fixed Dimensionality," World *Academy of Science, Engineering and Technology*, vol. 22, 2006.

[32] K. Sreelakshmi and P. Ramakanthkumar, "Neural Networks for Short Term Wind Speed Prediction," *World Academy of Science, Engineering and Technology*, no. 42, 2008.

[33] M. A. Nayak and MC Deo, "Wind Speed Prediction by different Computing Techniques," Conference of BALWOIS 2010, Ohrid, Republic of Macedonia, May 2010.

[34] S. Lawrence, C. L. Giles, and A. C. Tsoi, "Lessons in Neural Network Training: Over-fitting may be harder than Expected," *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, "AAAI-97", AAAI Press, Menlo Park, California, pp. 540-545, 1997.

[35] C. Macleod and G.M. Maxwell, "Incremental Evolution in ANNs: Neural Nets which Grow," *Artificial Intelligence Rev.*, vol. 163, pp. 201-224, 2001.

[36] R. J. Hyndman, A. B. Koehler, J. Kord., and R. D. Snyder, "Forecasting with Exponential Smoothing: The State Space Approach," p. 360, 2008, ISBN: 978-3-540-71916-8.

DOI: 10.5963/IJEE0502002