Sirte University Scientific Journal Vol. 14, No. 2 (2024) 85-90

Sirte University Scientific Journal (SUSJ)

Journal home page: http://journal.su.edu.ly/index.php/SUSJ/index

DOI: 10.37375/susj.v14i2.3096

Utilizing Artificial Neural Networks for Wind Speed Estimation

A Case Study of Dernah City, Libya

Magdi Buaisha [1], Essa Shawail [2]

[1] *The General Directorate of Inspection and Measurement - National Oil Corporation (NOC), Tripoli, Libya*

[2] *Department of chemical engineering, university of Sirte, Sirte, Libya*

© SUSJ2022.

DOI: 10.37375/susj.v14i2.3096

ARTICLE INFO:

Received 11 October 2024.
Accepted 21 November 2024.
Available online 24 December 2024.

Keywords: Renewable energy, ANN, Levenberg-Marquardt Method, electrical energy, Wind Speed.

A B S T R A C T

Electric power is universally acknowledged as a crucial factor in enhancing living standards. As a result, safe electrical energy consumption is crucial to efficient national energy management. To do this, meticulous assessments of the electricity demand are required. Finding viable sites for turbine placement through feasibility studies and measuring local wind speeds are essential steps before establishing the plant wind power. Estimation of wind speed and simulations can be used to conduct these evaluations.. This study uses an artificial neural network (ANN) with the Levenberg-Marquardt (LM) learning algorithm to estimate wind speed for the Libyan city of Dernah. One-year data from the Libya Meteorology Center has been utilized to train, test, and validate the ANN to to predict hourly wind speed . The structure of the ANN was evaluated with neuron counts of 10, 20, 30, 40, and 50, allowing us to determine the optimal number of neurons for accurate predictions. The estimation analysis was performed using results obtained from the Levenberg-Marquardt method (LMA), along with the mean square error (MSE) and the coefficient of determination (R^2). The results show that the Levenberg-Marquardt method with 10 neurons performs the best, with values of 0.99661 for R2 and 0.000250 for MSE. These findings confirm that wind speeds can be calculated within reasonable bounds since they show that the estimates of wind speeds based on the scant meteorological data available nearly match the measured values.

1 Introduction

Wind energy is a sustainable and environmentally friendly resource, as it is both renewable and non-polluting. Its eco-friendly characteristics ensure that it does not release any gases that could impact the surrounding environment including carbon dioxide (CO_2), unlike fossil fuels. Moreover, wind is a renewable power source that is both widely recognized and feasible, and it can be used to produce energy in large quantities at a low cost. Subsequently, the substantial wind energy potential must be evaluated now. To effectively plan the production, maintenance, and management of wind power plants, it is essential to assess wind speed. However, it is challenging to accurately estimate wind speed due to its irregular and discrete nature. Wind power's diverse advantages, particularly in socioeconomic and ecological aspects, have contributed to its swift emergence as a valuable renewable energy source.

Wind usage is more likely to occur globally which could lead to warmer climates compared to other sources such as oil, which can quickly deplete. The increase in wind power was driven by its advantages for power generation on a large scale. Predicting wind speed in advance is very important and hence it should be prepared to prevent economic losses, facilitate wind system regulation, and improve the operational efficiency of industries by making safer decisions. Wind speed forecasting is one of the most relevant and challenging research problems in the world today [1]. Efforts have been made to exploit wind energy at a high level taking into account the technical aspects. So far, wind turbines' aerodynamic optimization gathered attention from numerous research groups [2]. A variety of models were utilized to predict wind energy output, including a combination of meteorology and background or past generation data [3]. Many predictive applications have utilized artificial neural network models [4]. ANN is a clever computing method that functions similarly to a neural network in humans. The nature of generalization, adaptability, non-linearity, and large-scale data handling are among the best features of neural networks. These inherent characteristics enhance the efficiency of the neural network, allowing it to accurately estimate wind speed based on the provided input parameters. Neural networks have been effectively utilized across various applications for tasks such as image processing, control systems, association, prediction, and recognition.

Meticulous wind speed estimation is crucial for efficient production planning, maintenance, and management of wind energy facilities. By improving wind speed predictions, we can enhance the amount of energy generated [5].

For this study, Libya was selected because of its distinctive geographical position. It is situated in North Africa on the Mediterranean Sea with a coast reaching about 2000km. In addition, to the presence of sufficient winds in this region, winds can become an alternative source of power generation. Further, there are a lot of winds passing through this region for the whole year and this region also is exposed to Ghibli winds [6]. Wind energy is abundant in Libya's coastal areas, such as Derna City, and can be used for vital functions like desalinating seawater and producing power.

A previous study indicated that Derna City represents the most attractive location for such research, as it has an average wind speed of about 7.5 m/s [7].

To conduct our research, we used an artificial neural network (ANN), which is made up of intelligent neurons that work similarly to a human brain and have distinct input-output links. A neural network is suggested as a promising method in this field and employed to assess the link between input and output; therefore, the basis of this research is to study wind speed using the popular algorithm (ANN). Since neural networks are inspired by nature and modeled after the biological functions of the human brain, the artificial neuron serves as their fundamental building element [8]. ANN is utilized as a powerful and adaptable method to forecast the nonlinear behavior of the chosen wind prediction system. The artificial neural network (ANN) autonomously reduces errors using available input and output data, eliminating the need for mathematical models or equations.

careful attention is needed in identifying the wind power potential in a particular area due to climate changes. In this sense, forecasting both wind power generation and wind power potential is essential. ANN model is proposed to predict the potential wind power generation with the help of projected climatic scenarios in future years

The main objective of this study is to predict wind speed in Dernah City, Libya, utilizing an Artificial Neural Network (ANN) model. The ANN was developed in MATLAB to effectively analyze wind speed in the city. Specifically, this research focuses on creating the ANN model, determining the optimal number of hidden neurons in the hidden layer, and accurately predicting wind speed with the resulting model. The algorithms are implemented in MATLAB through Nftool, which utilizes a backpropagation algorithm (BPA) for neural network fitting.

2 Materials and Methods

The Artificial Neural Network approach is used to evaluate wind speed through the BP method along with a feedforward neural network, utilizing the MATLAB toolbox [9]. The Libyan State Meteorological Service provided a dataset containing wind speed measurements for Dernah, Libya. This dataset comprises 600 data points, reflecting a complete year of hourly average measurements taken at an elevation of 10 meters above ground level, January 1, 2023, through December 30, 2023. The selected input parameters include average atmospheric pressure per hour, average relative humidity per hour (RH in %), average atmospheric temperature per hour (T in °C), and pressure (P in mbar). The output variable is the average wind speed (WS) recorded hourly at the specified place. The Neural Network Toolbox™ offers various algorithms, applications, and functions for the development, training, monitoring, and modeling of Neural Networks (NNs). These tools aid in organizing information, recognizing patterns, grouping data, conducting regression analysis, analyzing time series, exploring dynamic systems, and employing deep learning techniques [10, 11,12].

The method of estimating the wind speed with high precision is a useful approach to reduce to a minimum the problems derived from the variability and uncontrollability of the wind speed [13].

2.1 Training and Testing Processes for Artificial Neural Networks

To develop, apply, and test the neural network (NN), MATLAB's nftool was utilized, focusing on a forward-propagating network that incorporates a chain of neurons within a hidden layer. The training was conducted using the Levenberg-Marquardt algorithm (LM), recognized as the quickest training method for networks of moderate size. In the framework of a wind energy system in Dernah city, an artificial neural network (ANN) designed as an established feed-forward back-propagation network was utilized. This network utilizes a tansig function in the hidden layer, along with a purelin function at the output node. In this research, MATLAB code was utilized to train a neural network (see Appendix A).

3 Results

The dataset utilized in this study is divided into three parts: 70% for training, 15% for validation, and 15% for network testing. The proposed (ANN) employs a backpropagation feedforward model, incorporating a tansig function in the hidden layer, which consists of a varying number of neurons, and a purelin function at the output layer. The implementation was carried out using embedded MATLAB code. To train a machine-driven neural network with one concealed layer, MATLAB code was generated five times using the neural fitting (nftool) function along with the Levenberg-Marquardt algorithm. Each configuration retained 15% of the original dataset for validation and testing. The performance of the Levenberg-Marquardt algorithm was evaluated using different neuron counts (10, 20, 30, 40, and 50) according to the coefficient of determination (R^2) and the mean squared error (MSE). The findings indicated that the network with 10 neurons exhibited the finest performance, achieving an R^2 value of 0.99661 and an acceptable MSE of 0.000250, as depicted in Figures 1 and 2. The comparison between the recorded and the predicted Average wind speed for one year are shown in Figure 3.

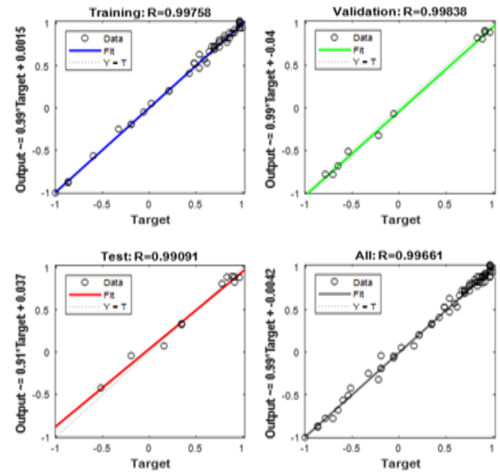

Figure 1. Regression plot of the neural network employing with 10 neurons

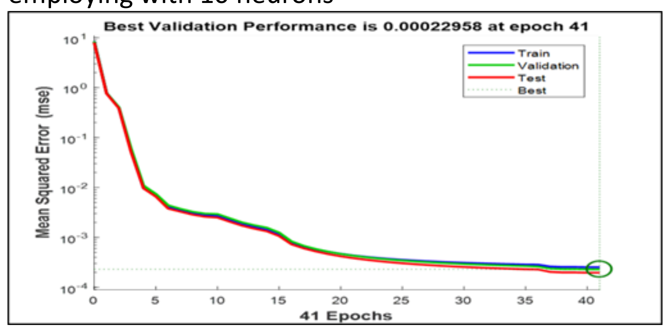

Figure 2. Plots depicting the training performance results with 10 neurons.

Figure 3. a comparison between the recorded and the predicted Average wind speed for one year

4 Conclusions

The Data from the Libya Meteorological Center was utilized to develop a specialized model capable of accurately predicting wind velocity based on various factors. The suggested model is a feedforward backpropagation (BP) framework that employs a tangent sigmoid transfer (tansig) function connecting the hidden layer and the input layer, which contains a variety of neurons. The output node uses a linear transfer (purelin) function, implemented using the neural network fitting tool in MATLAB (nftool). To evaluate the model, training was conducted with different numbers of neurons—10, 20, 30, 40, and 50—in the hidden layer. The Levenberg-Marquardt method was assessed across these configurations, and the forecast results were compared with data from the Libya Meteorological Center. The findings revealed that the Levenberg-Marquardt method performed the best among backpropagation algorithms, achieving the lowest mean squared error (MSE) with 10 neurons for Dernah city. The model achieved the lowest mean squared error (MSE) of 0.000250 and a remarkable R^2 value of 0.99661, indicating that the predictions made by the artificial neural network (ANN) for wind velocity closely matched the actual measurements. In conclusion, the artificial neural network model detailed within this research proves effective to forecast wind speed, which is essential for power generation. To optimize outcomes, it is vital to utilize theoretical architectures in conjunction with the ANN to refine the modeling process. Accordingly, the results of the research open a new era for the wind power generations in Derna city to plan its energy demand and supply in the future

References

1. Li, G., & Shi, J. (2010). On comparing three artificial neural networks for wind speed forecasting. Applied Energy, 87(7), 2313-2320.

2. Abohedma, M. B., & Alshebani, M. M. (2010). Wind load characteristics in Libya. International Journal of Civil and Environmental Engineering, 4(3), 88-91.

3. De Freitas, N. C., Silva, M. P. D. S., & Sakamoto, M. S. (2018). Wind speed forecasting: a review. Int. J. Eng. Res. Appl, 8, 4-9.

4. Kariniotakis, G., Pinson, P., Siebert, N., Giebel, G., & Barthelmie, R. (2004, October). The state of the art in short term prediction of wind power-from an offshore perspective. In SeaTech week-ocean energy conference ADEME-IFREMER., 12(2): 9-8.

5. Fu, L. M. (2003). “Neural networks in computer intelligence”, Tata McGraw-Hill Education, New York, USA, 34-41.

6. Mohamed, A. A., & Elmabrouk, A. M. (2009). Assessment of the wind energy potential on the coast of Tripoli. Nasser University, Faculty of Engineering, Mechanical Department, 1-10.

7. Mohamed, O. A., & Masood, S. H. (2018, June). A brief overview of solar and wind energy in Libya: Current trends and the future development. In IOP Conference Series: Materials Science and Engineering (Vol. 377, No. 1, p. 012136). IOP Publishing.

8. More, A., & Deo, M. C. (2003). Forecasting wind with neural networks. Marine structures, 16(1), 35-49.)

9. Imrie, C. E., Durucan, S., & Korre, A. (2000). River flow prediction using artificial neural networks: generalisation beyond the calibration range. Journal of hydrology, 233(1-4), 138-153.

10. Demuth, H., Beale, M., & Hagan, M. (1992). Neural network toolbox. For Use with MATLAB. The MathWorks Inc, 2000.

11. Arora, J. K., & Srivastava, S. (2010, June). Neural network modeling and simulation of sorption of Cd (II) ions from waste water using agricultural waste. In Proceedings of the world congress on engineering (Vol. 3, pp. 1-4).

12. Buaisha, M., BALKU, Ş., & YAMAN, Ş. Ö. (2019). ANN-assisted forecasting of adsorption efficiency to remove heavy metals. Turkish Journal of Chemistry, 43(5), 1407-1424.

13. Bulut, M., Tora, H., & Buaisha, M. (2021). Comparison of three different learning methods of multilayer perceptron neural network for wind speed forecasting. Gazi University Journal of Science, 34(2), 439-454.

Appendix A : MATLAB code

```
% Solve an Input-Output Fitting problem with a Neural Ne
% Script generated by Neural Fitting app
% Created 10- September -2024 14:30:10
% This script assumes these variables are defined:
%   Input - input data.
%   Output - target data.
R=load('lmb.dat');
x = Input' = [R(1:600,1:3)];
t= Output'=[R(1:600,4)];
% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging pr
% 'trainscg' uses less memory. Suitable in low memory situ
trainFcn = 'trainlm';  % Levenberg-Marquardt backpropaga
% Create a Fitting Network
   hiddenLayerSize = 10;
% hiddenLayerSize = 20;
% hiddenLayerSize = 30;
% hiddenLayerSize = 40;
% hiddenLayerSize = 50;
net = fitnet(hiddenLayerSize,trainFcn);
% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminma
net.output.processFcns = {'removeconstantrows','mapminm
% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivid
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
% Train the Network
[net,tr] = train(net,x,t);
% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
% View the Network
view(net)
% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```