Chapter 10

Wind Energy Prediction Using Artificial Neural Networks

Sumanta Pasari , Aditya Shah , and Utkarsh Sirpurkar

Abstract Renewable energy sources are one of the most vital alternatives to the conventional non-replenishable energy generating systems. Among several renewable power sources, the installed wind power capacity contributes to almost half of the total capacity. However, the variability and seasonality in wind speed, wind direction, atmospheric pressure, relative humidity and precipitation cause wind power generation to be highly volatile. In this regard, the present study aims to develop a wind speed prediction scheme using artificial neural network (ANN) techniques. Single step and multistep recurrent neural networks (RNNs) are implemented. The long short term memory (LSTM), rectified linear unit (ReLU) activation function and Adam optimization algorithm are considered to carry out daily to monthly prediction using the RNN process. Results, based on the data from Charanka solar energy park in Gujarat, indicate that root means square (RMS) errors for univariate single layer, multivariate single layer and univariate two-layer models are 0.601, 0.782 and 1.120, respectively. Therefore, a univariate single layer RNN architecture is recommended for wind speed prediction. We envisage that a multilayer RNN model may improve the prediction accuracy over longer period.

Keywords Wind energy prediction · Neural network · Charanka solar park

S. Pasari (✉) · A. Shah · U. Sirpurkar
Department of Mathematics, Birla Institute of Technology and Science, Pilani Campus, Pilani
333031, RJ, India

e-mail: sumanta.pasari@pilani.bits-pilani.ac.in

A. Shah

e-mail: f2014715p@alumni.bits-pilani.ac.in

U. Sirpurkar

e-mail: f2014785p@alumni.bits-pilani.ac.in

© The Author(s) 2020

K. S. Sangwan and C. Herrmann (eds.), *Enhancing Future Skills and Entrepreneurship*,
Sustainable Production, Life Cycle Engineering and Management,
https://doi.org/10.1007/978-3-030-44248-4_10

10.1 Introduction

The non-renewable fossil fuels, which include coal, oil and natural gas, play a dominant role in global energy consumption and industrial revolution. Currently fossil energy supplies to about 80% of the total world's energy. However, these carbon-rich reserved fossil fuels are not only exhaustible in about next ten decades, but also the primary source of air pollution, carbon emission and greenhouse gas production (Bhatia and Gupta 2018; Kumar et al. 2010; Wang et al. 2011). The planners and policy makers must therefore look for alternative energy sources to meet the increasing energy demand of developing economies in a sustainable way.

Renewable energy sources such as wind, solar radiation, hydroelectricity and biomass are regenerative and abundant. The cost of renewable power production, after a decade, will be much lower than the energy production cost from fossil fuels (Kumar et al. 2010). In India, as of 2018, about 20% of the total installed power comes from these clean energy resources against the national target of 40% by 2030. India sets for an ambitious goal of generating 100 GW of solar and 60 GW of wind power by 2022, and 250 GW of solar and 100 GW of wind power by 2030 (Bhatia and Gupta 2018). Achieving such an aspiring target requires dedicated research in various fields including technology development, site-identification, smart-grid planning, energy integration (to the main power grids), policy making, cost-benefit analysis and energy forecasting (Bhatia and Gupta 2018; Kumar et al. 2010; Wang et al. 2011; Rather 2018). In this study, we concentrate on wind energy prediction based on the artificial neural network (ANN) techniques.

Wind energy, like solar power, is plentiful, replenishable and a promising alternative to burning fossil fuels. It uses air flow through turbines to provide mechanical power convertible to electricity. Neither water is consumed during operation, nor is more land required. Wind farms comprise many individual onshore or offshore wind turbines connected to the electric power transmission network. However, wind power supply is highly volatile due to the variability and seasonality in heat energy from the sun, wind speed, wind direction, atmospheric pressure, relative humidity, precipitation and temperature gradients between land and sea (Kumar et al. 2010). Improved wind power predictions thus are crucial in effective market design, real-time grid management, power transmission capacity evaluation and ancillary information (Wang et al. 2011).

There have been several attempts to wind energy prediction, varying from deterministic (physical) approach based on numerical weather prediction to statistical technique based on historical data analysis, time-series modeling or artificial intelligence (neural networks) (Bhatia and Gupta 2018; Kumar et al. 2010; Wang et al. 2011; Rather 2018). On the basis of different time-horizons, wind power prediction could be classified as immediate-short-term (up to 8 h), short-term (day ahead) and long-term (multiple days to months ahead) prediction (Wang et al. 2011). While real-time grid operations, energy trading and regulatory actions benefit from immediate-short-term prediction, economic load balance and operational security analysis depend on the short-term wind energy prediction. The long-term forecasting

is useful for cost-optimal energy storage strategy, operation management, maintenance planning and cost-benefit analysis (Bhatia and Gupta 2018; Kumar et al. 2010; Wang et al. 2011). However, due to the volatility of wind speed over the months, it is challenging to accurately predict the wind speed for a number of practical applications. Therefore to improve the short/long term forecasting accuracy, several methods have been proposed such as direct multistep recurrent neural network (RNN) using LSTM neural network combining fuzzy entropy (Qin et al. 2019), multi-variable (e.g., wind speed, temperature, humidity and pressure) stacked LSTMs model (MSLSTM) (Liang et al. 2018), univariate and multivariate autoregressive integrated moving average (ARIMA) with ANNs (Cao et al. 2012) and pipelined recurrent neural network (PRNN) based NARMAX ANN model (nonlinear autoregressive moving average artificial neural network with external inputs) (Liangyou et al. 2019). Nonetheless, there is no global best prediction model to be applicable irrespective of geographical region (Rather 2018; Qin et al. 2019; Liang et al. 2018; Cao et al. 2012; Liangyou et al. 2019). Thus, in the present work, we focus on daily to monthly wind speed prediction using a data driven single step and multistep recurrent neural network (RNN) process for the Charanka solar park of India.

10.2 Data Description

Our dataset comprises hourly wind speed and wind direction data of Charanka solar park (also known as Gujarat solar park 1; 23.95° N, 71.15° E) for a period of 15 years, 2000–2014. This dataset is publicly available from the National Solar Radiation Database maintained by National Renewable Energy Laboratory (NREL) (NREL homepage 2019). In addition to wind speed and wind direction, the dataset also provides information of temperature, solar radiation, such as DHI (Diffuse Horizontal Irradiance), DNI (Direct Normal Irradiance), GHI (Global Horizontal Irradiance), solar zenith angle, clear-sky DHI, clear-sky DNI and clear-sky GHI along with the information of some meteorological parameters such as dew point, atmospheric pressure, relative humidity, precipitable water and snow depth.

10.3 Methodology

The methodology for the present study comprises three major steps: data preparation, RNN model implementation and wind speed prediction. The RNN model implementation further consists of three sequential steps, namely training, validation and testing. Before we move on, some preliminary discussion about RNN process would be helpful.

RNNs are a type of artificial neural network with loops in them to persist information in sequences of inputs. The chain-like architecture in RNNs allow them not only to learn from training (similar to feed-forward neural networks), but also to

remember things learnt from all prior inputs while generating outputs. The beauty and strength of RNN therefore comes from its *hidden state* which manages to span many time steps as it marches forward through sequences of input, output or both vectors (Graves 2012; An Introduction to Recurrent Neural Networks 2019). In order to describe the workflow for the RNN process at time step t , let x_t be the input vector, h_t be the hidden state and y_t be the output vector. Needless to mention that the output vector in RNN is influenced not only by the immediate current input, but also the entire history of inputs (Graves 2012; An Introduction to Recurrent Neural Networks 2019). The hidden state, in general, can be mathematically expressed as

$$h_t = f(h_{t-1}, x_t) \quad (10.1)$$

The hidden state h_t is updated as

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t) \quad (10.2)$$

Thus h_t is a function of the input x_t modified by a weight matrix W_{xh} (as used in feed-forward networks) added to the previous hidden state h_{t-1} multiplied by the transition matrix (hidden state to hidden state matrix) W_{hh} . These two weight matrices serve as filters by providing appropriate weights (importance) to the present input and past hidden state. The sum of these two intermediates, weighted input and hidden state, is then squashed by the function f , either a logistic sigmoid, tan-hyperbolic, ReLU (rectified linear unit) or others (Graves 2012; An Introduction to Recurrent Neural Networks 2019). Using the hidden state h_t , modified by the weight matrix W_{hy} , the output vector y_t is computed as

$$y_t = W_{hy} h_t \quad (10.3)$$

Comparing the model output to the actual (target) output, the error is generated. This error is then back-propagated to update the weights recursively until an allowable error limit is achieved. In this way, the RNN gets trained via the long short term memory (LSTM) technique (Graves 2012). As RNN has a memory and can remember every information through time, it is a powerful tool for time series prediction (Graves 2012; An Introduction to Recurrent Neural Networks 2019).

In this work, we implement three different RNN architectures, namely univariate unit-step single layer, multivariate two-step single layer and univariate multistep two-layer model for one-day to one-month wind speed prediction. While the first model considers single step inputs, that is, inputs from only the immediate previous state to predict future values, the second model considers previous two states for prediction. The multistep model uses inputs from several earlier steps for providing outputs. For each model, Adam optimization algorithm is used to update network weights iteratively based on training data, whereas the loss function is evaluated on the basis of mean absolute error (MAE). These supervised algorithms utilize about 70% of total data points for training and remaining 30%, divided equally, for model

Table 10.1 Parameters used in different RNN models

	Univariate unit-step single layer	Multivariate two-step single layer	Univariate multistep two-layer
Hidden layers	1	1	2
Input neurons	1	15	1
Steps	1	2	30
Neurons in hidden layer	100	100	100
Epochs	10000	200	100
Learning rate	0.001	0.001	0.001
Activation function	tan-hyperbolic	ReLU	ReLU
Optimization	Adam	Adam	Adam
Loss function	MAE	MAE	MAE

validation and testing. A comparison of the parameters used in these three different models is provided in Table 10.1.

For the univariate unit-step single layer RNN model, entire 15 years of daily wind speed data is used for training towards next day's wind speed prediction. Using this one-day predicted information, model then predicts the wind speed of the day next to the previously forecasted day. The process continues to output 365 days (one day at a time) wind speed prediction. In the multivariate two-step single layer model, daily wind speed data along with other 14 variables, such as wind direction, temperature, solar zenith angle, DHI, DNI, GHI, clear sky DHI, clear sky DNI, clear sky GHI, dew point, pressure, relative humidity, precipitable water and snow depth is considered. Recall that, physically, the wind speed depends on many variables like solar radiation and other meteorological parameters. This multivariate RNN also considers previous two steps while forecasting future wind speed data. The univariate multistep two-layer model, in contrary to the previous ones, uses two LSTM layers to predict one month's wind speed data with an input of current month's wind speed data.

The entire methodology is implemented in python using open-source *keras* library with *TensorFlow* backbend. Experimental results are provided in the next section.

10.4 Results and Conclusions

The root means square (RMS) errors corresponding to univariate unit-step single layer, multivariate two-step single layer and univariate multistep two-layer model for the year 2014 turn out to be 0.601, 0.782 and 1.120, respectively. The predicted wind speed is plotted against the actual wind speed in Fig. 10.1. It clearly demonstrates that complex architectures in multivariate RNN or univariate multistep RNN does not produce desirable fit. The reason for such poor performance could be due to the

Fig. 10.1 Predicted versus actual wind speed corresponding to **a** univariate unit-step single layer, **b** multivariate two-step single layer and **c** univariate multistep two-layer RNN architecture (forecasts of six alternative months are shown)

inclusion of several variables in multivariate RNN or the incorrect choice of training method and/or activation function.

Thus, the present study leads to the conclusion that a simple univariate unit-step single layer RNN model is the most suitable architecture for short-term wind speed prediction. The proposed scheme is equally applicable to forecast other renewable energy sources, such as solar radiation.

References

- An Introduction to Recurrent Neural Networks (2019). <https://medium.com/explore-artificial-intelligence/an-introduction-to-recurrent-neural-networks-72c97bf0912>, Accessed 24 June 2019
- Bhatia SC, Gupta RK (2018) Textbook of renewable energy. Woodhead Publishing India, pp 1–334
- Cao Q, Ewing BT, Thompson MA (2012) Forecasting wind speed with recurrent neural networks. *Eur. J Oper Res* 221(1):148–154
- Graves A (2012) Supervised sequence labelling with recurrent neural networks. Springer, Berlin, pp 1–146
- Kumar A, Kumar K, Kaushik N, Sharma S, Mishra S (2010) Renewable energy in India: current status and future potentials. *Renew Sustain Energy Rev* 14:2434–2442
- Liangyou H, Dongxiang J, Qian H, Yongshan D (2019) Wind speed forecasting using fully recurrent neural network in wind power plants. <https://pdfs.semanticscholar.org/fe5/4b2b2ae713ebf8b08ad963d7dbbea15ad2fc.pdf>, Accessed 24 Aug 2019
- Liang S, Nguyen L, Jin F (2018) A multivariable stacked LSTM network for wind speed forecasting. In: 2018 IEEE conference on big data. <https://arxiv.org/pdf/1811.09735.pdf>, Accessed 24 Aug 2019
- NREL homepage. <https://www.nrel.gov/>, Accessed 24 June 2019
- Qin Q, Lai X, Zou J (2019) Direct multistep wind speed forecasting using LSTM neural network combining EEMD and fuzzy entropy. *Appl Sci* 9(126):1–19
- Rather NUR (2018) Introduction to renewable energy technologies in India. Education Publishing, pp 1–140
- Wang X, Guo P, Huang X (2011) A review of wind power forecasting models. *Energy Procedia* 12:770–778

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.