

# Short-Term Prediction of Wind Power Density Using Convolutional LSTM Network

**Deepak Gupta**

Department of Computer Science and Engineering, National Institute of Technology Arunachal Pradesh  
India

**Vikas Kumar**

Department of Computer Science and Engineering, National Institute of Technology Arunachal Pradesh  
India

**Ishan Ayus**

Department of Computer Science and Engineering, National Institute of Technology Arunachal Pradesh  
India

**M. Vasudevan**

Department of Civil Engineering, Bannari Amman Institute of Technology, Tamil Nadu  
India

**N. Natarajan**

Department of Civil Engineering, Dr. Mahalingam College of Engineering and Technology, Tamil Nadu  
India

*Efficient extraction of renewable energy from wind depends on the reliable estimation of wind characteristics and optimization of wind farm installation and operation conditions. There exists uncertainty in the prediction of wind energy tapping potential based on the variability in wind behavior. Thus the estimation of wind power density based on empirical models demand subsequent data processing to ensure accuracy and reliability in energy computations. Present study analyses the reliability of the ANN-based machine learning approach in predicting wind power density for five stations (Chennai, Coimbatore, Madurai, Salem, and Tirunelveli) in the state of Tamil Nadu, India using five different non-linear models. The selected models such as Convolutional Neural Network (CNN), Dense Neural Network (DNN), Recurrent Neural Network (RNN), Bidirectional Long Short Term Memory (LSTM) Network, and linear regression are employed for comparing the data for a period from Jan 1980 to May 2018. Based on the results, it was found that the performance of (1->Conv1D|2->LSTM|1-dense) is better than the other models in estimating wind power density with minimum error values (based on mean absolute error and root mean squared error).*

**Keywords:** Wind power, Bidirectional LSTM, CNN LSTM, Linear regression

## 1. INTRODUCTION

The current status of fossil fuel consumption and threatening impacts of conventional energy sources on the environment have motivated the researchers towards focusing on increased extraction of energy from renewable resources. Renewable energy sources like tidal, hydropower, wind, solar, geothermal, etc. are being widely employed as alternate sources of energy. Among them, wind energy promises to be suitable for large-scale applications due to its ease of availability, installation, environmental friendliness, and inexhaustibility [1,2]. Exploration of wind energy from offshore and onshore sites has attracted the attention of both researchers and capitalists as a potential area of investment. This is fuelled by the ever-increasing demand for energy in various production and service sectors [3]. As a rapidly growing source of energy, the current trend in wind energy extraction is expected to expand in the near future. Although this is one of the cheapest forms of energy, inappropriate placement of wind turbines may result in under-utilization of their capacity and can lead to huge loss of revenue. Thus, scientific evaluation of wind resources plays a vital role in providing secure wind energy utilisation and enhancement of the efficiency of wind energy markets [4,5]. Some of the researchers have adopted novel techniques for determining the optimal location of the wind farms [6-8].

As a basic pre-requisite, the knowledge of wind power density is critical since the variation in wind frequency distribution can result in varying wind power densities for the same wind speed. Although various predictive models are available to estimate the wind energy-associated parameters, there are a few inherent limitations from a computational perspective. As an obvious response to the recent trends in artificial intelligence-based modelling techniques, there are various attempts to employ them for simulation, estimation and optimisation of wind energy parameters. Taylor et al. [9] constructed density forecasts from weather ensemble predictions and statistical time series models. Weibull distribution-based Particle Swarm Optimisation (PSO) was reported as a useful tool in assessing the wind energy potential in Taiwan [10]. In another approach, Monte Carlo simulation was used by Jeon and Taylor [11] to compute conditional kernel density for the VARMA-GARCH model. It is to be noted that most of these models predict the wind power density based on the prediction of wind speed and direction functions. Bigdeli et al. [12] developed hybrid models to predict the wind power time series by combining Neural Network (NN) with Imperialistic Competitive Algorithm (ICA), Genetic Algorithm (GA), and PSO techniques. Using the PSO, Abductory Induction Mechanism (AIM), and the persistence (PER), Mohandes and Rehman [13] predicted the distribution of 12 hours wind speed in Saudi Arabia.

Towards deriving appropriate strategies for evolving hybrid solution systems, an Adaptive Neuro-Fuzzy Inference System (ANFIS) was employed to study the distribution of probability density functions (PDF) of

Received: February 2021, Accepted: April 2021

Correspondence to: N. Natarajan, Department of Civil Engineering, Dr. Mahalingam College of Engineering and Technology, Pollachi – 642003, Tamil Nadu, India.  
E-mail: natarajan@drmcet.ac.in

doi:10.5937/fme2103653G

© Faculty of Mechanical Engineering, Belgrade. All rights reserved

FME Transactions (2021) 49, 653-663 653

wind speed and directions by Shamshirband et al. [14]. The results were in good comparison with the conventional wind distribution approaches such as Weibull, Frechet, Gumbel, and joint probability functions. A similar adaptive model was designed by Baskar and Singh [15] namely Adaptive Wavelet Neural Network (AWNN) and Feed-Forward Neural Network (FFNN) to successfully predict the wind speed distributions. A combination of Support Vector Machine (SVM), Artificial Neural Network (ANN), and Genetic Programming (GP) was used by Mohammadi et al. [16] to predict the monthly values of wind power density which is in good comparison with the prediction accuracy of Extreme Learning Machine (ELM). In another hybrid approach, the Beveridge-Nelson decomposition method was used to predict the wind power potential at the Xinjian Uygur autonomous region in China [17]. Based on the relevance of the SVM in optimization, they also employed an ant-lion optimizer for predicting the wind power based on the hourly wind speed data. A combination of GA and ELM was used to predict the wind power by Wang et al. [18]. Another recent hybrid approach for the ANFIS models such as ANFIS-PSO, ANFIS-GA, ANFIS-DE (differential evolution) was attempted by Hossain et al. [19] for four different locations in Malaysia using the monthly and weekly wind power density values. An improved teaching-learning-based optimization (iTLBO) with ELM was proposed by Xue et al. [20] to predict the wind power by incorporating the recursive feature elimination (RFE) method for feature selection.

As far as deep learning methods are concerned, many researchers are coming up with deep learning techniques for forecasting wind speed [21-27]. However, only limited studies are available for forecasting wind power density using deep learning. Lu et al. [25] used encoder-decoder LSTM to predict short-term wind power. Xu et al. [28] employed an adaptive LSTM for short-term prediction of wind power. Yu et al. [29] adopted LSTM-EFG model for wind power prediction based on sequential correlation features. However, it is understood that studies pertaining to the comparison of statistical performance of the deep learning techniques are lacking in the literature. Therefore, in the present study, we have attempted to compare five different non-linear techniques, namely, convolutional neural network, dense neural network, recurrent neural network, bidirectional LSTM network, and linear regression to compare their accuracy in prediction so as to determine the most efficient deep learning method for forecasting wind power density.

### 1.1 Related works

Generally, there are three major types of forecasting models used for wind speed calculations, namely physical-based models, statistical models, and hybrid models. Physical models usually consider several parameters such as temperature, pressure, surface roughness, obstacles, etc. in the lower atmosphere for creating mathematical models of the atmosphere to predict the wind speed. Statistical models are based on the previously recorded data for forecasting the wind speed without the consideration of meteorological

conditions. Hybrid models consist of both physical and statistical approaches in forecasting wind speed. We have focused on the statistical approach in our study to forecast the wind power density. Non-linear statistical models are applied to the dataset to evaluate and compare the respective performances.

## 2. NONLINEAR MODELS

Recently, ANN models have been extensively applied in the real-time series problem to capture non-linearities prevailing in the dataset. ANN can handle such non-linear temporal correlation and can approximate a large class of functions with a high degree of accuracy. However, though acting as a good approximator, ANN has the problem of overfitting and may provide misleading results if not properly monitored. A single hidden layer feedforward network can be mathematically modeled as :

$$y_t = \alpha_0 + \sum_{k=1}^m \alpha_k \left( \beta_{0k} + \sum_{l=1}^n \beta_{lk} y_{t-l} \right) + \varepsilon_t \quad (1)$$

where  $\alpha_k (k=0,1,2,3,\dots,m)$  and  $\beta_{lk} (k=0,1,2,3,\dots,m; l=1,2,3,\dots,n)$  are connection weights,  $n$  is the number of input nodes,  $m$  is the number of hidden nodes,  $\varepsilon_t$  is the random error,  $y_t$  is the output and input are given by  $y_{t-1}, y_{t-2}, \dots, y_{t-n}$ .

### 2.1 Convolutional Neural Networks

The central idea in a convolutional neural network is a mathematical operation called convolution to detect specific features such as image pixels in pattern recognition problems. A kernel matrix is slid through the input image matrix to create feature maps for the next layer. If an image is denoted by  $f$ , kernel by  $h$ ,  $i$  and  $j$  are the relative position, indexes of resultant row and column are represented by  $q$  and  $r$ , then:

$$(f * h)[q, r] = \sum_i \sum_j h[i, j] \cdot f[q - i, r - j] \quad (2)$$

where  $f$  denotes the object,  $h$  denotes the kernel,  $i$  and  $j$  denote the relative positions and  $q$  and  $r$  represents the indexes of resultant row and column. After the convolution operation, certain activation functions are overlapped to introduce non-linear transformation followed by the max-pooling layers. Max-pooling layers down-sample the feature map output to make the representation approximately invariant to small translations. The nodes after the pooling layers are flattened into a fully connected layer to make predictions with subsequent layers.

### 2.2 Dense Neural Networks

Dense neural networks (DNN) have fully connected linear layers where the result of each node being passed through a non-linear activation function. In other words, each node of a layer receives an input from the previous layers in a densely connected way. The DNN thus adds non-linearity in the operation and can approximate complex mathematical functions [30].

### 2.3 Recurrent Neural Networks

The RNN is a widely used ANN in solving problems having temporal correlations and those exhibiting temporal dynamic behavior [31]. They connect the hidden layer with the former ones in a circular way. These recurrent units have the ability to save the historical information from the sequence, thusmaking them fit for problems whose output depends on the previous values. Unlike conventional ANN, the overfitting issues can be prevented by dropping out or randomly ignoring a certain proportion of neurons from the neural network where corresponding weights are not updated during the forward or backward pass of the training phase [32,33].

### 2.4 Long Short Term Memory Network

This particular approach was developed to overcome the drawbacks of simple RNN like vanishing gradient and exploding gradient problems [34, 35]. Here, theset of recurrently connected memory blocks are defined interms of memory cells and other resembling units such as multiplicative units like forget, input, and out gates with different functionality [34, 36]. Bidirectional LSTM emphasizes learning the most out of the input sequence by unrolling the network in both forward and backward directions. It is commonly used for sequence prediction and sequence generation. The CNN-LSTM uses a CNN layer where the input sequence helps in feature extraction and the LSTM layer accounts for interpretation and sequence prediction based on time [37].

### 3. STUDY AREA

The state of Tamil Nadu (TN) is situated in the southernmost region of the Indian peninsula. In this study, we have considered five stations, namely, Chennai, Coimbatore, Madurai, Salem, and Tirunelveli.

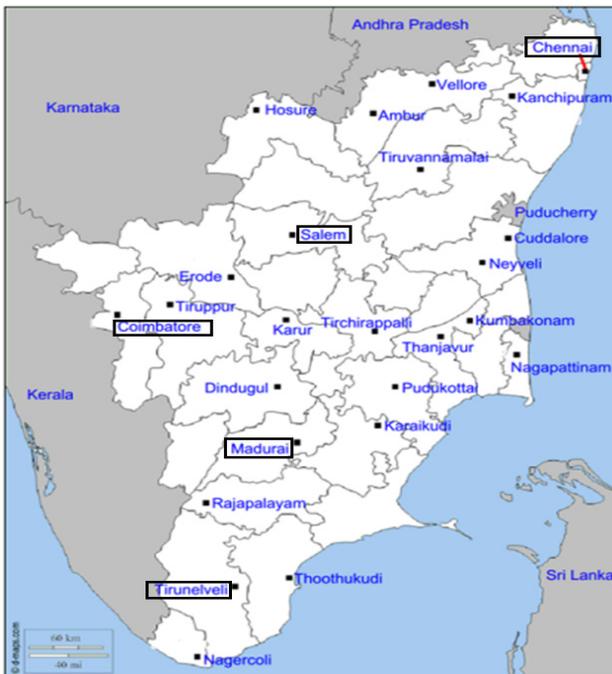


Figure 1. Map of Tamil Nadu showing the chosen five cities (marked with boxes)

The geographical location of these stations in the state is provided in Figure 1.

Table 1 provides the geographical details of the latitude, longitude, and altitude of the chosen stations. The table also provides the mean wind speed data collected from MERRA -2 reanalysis database (NASA) over a long-term duration at each site location. Hourly mean wind speed data recorded at a height of 50m above ground level from Jan 1980 to May 2018 was considered in this study. It is to be noted that the hourly wind speed data has been averaged to obtain the daily wind speed data for the purpose of calculating the wind power density values.

### 4. WIND POWER DENSITY ESTIMATION

While the term wind speed indicates the average velocity of movement of the wind, the wind power density refers to the capacity of the wind to generate power from a given mass of air for a given region [38]. Wind power density estimation is essential to assess the reliable wind resource potential. The wind power can be computed from the measured wind speed values using the probability distribution function. The following expression can be used to calculate the wind power density based on the Weibull probability density function [39]:

$$P = \frac{1}{2} \rho \int_0^{\infty} v^3 f_w(v) dv \quad (3)$$

where  $\rho$  is the density of air [ $\text{ML}^{-3}$ ] and  $v$  is the wind speed [ $\text{LT}^{-1}$ ]. The expression for  $f_w(v)$  is:

$$f_w(v) = \frac{k}{c} \left(\frac{v}{c}\right)^{k-1} e^{-\left(\frac{v}{c}\right)^k} \quad (4)$$

where  $v$  is the wind speed [ $\text{LT}^{-1}$ ],  $k$  is the shape factor [-] and  $c$  is the scale factor [ $\text{LT}^{-1}$ ]. The values of  $k$  and  $c$  are further determined using the standard deviation method and power density method.

#### 4.1 Standard deviation method

The values of  $k$  and  $c$  are estimated using the following equations.

$$k = \left(\frac{\sigma}{\bar{v}}\right)^{-1.086} \quad (5)$$

$$c = \frac{\bar{v}}{\Gamma\left(1 + \frac{1}{k}\right)} \quad (6)$$

where  $\bar{v}$  and  $\sigma$  are the mean wind speed and standard deviation of the wind speed. The  $\Gamma(x)$  is the gamma function which is commonly used in various calculus models including in probability distribution integrals as defined by the following equation.

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \quad (7)$$

**Table 1. Topographical details of the chosen stations**

S.No	Location	Latitude (°N)	Longitude (°E)	Altitude (m)	Average wind speed (m/s)	Average wind power density (W/m <sup>2</sup> ) by standard deviation method	Average wind power density (W/m <sup>2</sup> ) by power density method
1	Chennai	13.08	80.27	13	5.40	383.85	390.42
2	Coimbatore	11.01	76.95	420	4.70	231.16	230.15
3	Madurai	9.92	78.11	137	4.53	241.31	240.09
4	Salem	11.66	78.14	286	4.73	246.12	244.70
5	Tirunelveli	8.71	77.75	41	5.60	377.37	375.54

**4.2 Power density method**

To estimate the shape and scale factors via the power density method, the energy pattern factor ( $E_{pf}$ ) has to be calculated. The  $E_{pf}$  is a parameter defined by Akdag and Dinler [40]:

$$E_{pf} = \frac{\overline{v^3}}{\bar{v}^3} = \frac{\Gamma(1 + \frac{3}{k})}{\Gamma(1 + \frac{1}{k})} \quad (8)$$

where  $\overline{v^3}$  is the mean of the cube of wind speed and  $\bar{v}^3$  is the cube of mean speed. The scale factor is computed using Eqn. (4) and the shape factor is computed as per the equation given below [39]:

$$k = 1 + \frac{3.69}{E_{pf}^2} \quad (9)$$

The wind power density values thus calculated based on the above methods for the chosen five cities has been presented in Table 1.

**5. EXPERIMENTAL SETUP**

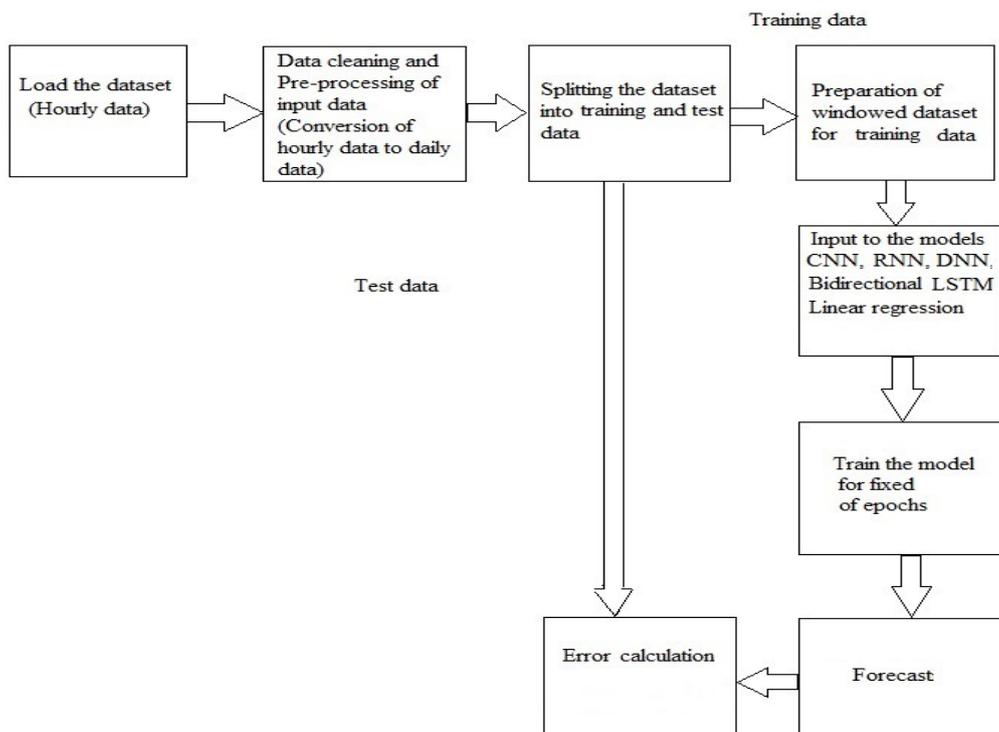
The experiments were performed on Google Colaboratory which included the following specifications:

- Intel Xeon @2.20GHz with two cores.
- Nvidia Tesla P100 GPU
- 13 GB RAM

For training and inferencing of the proposed approach, the popular software library, Tensorflow is used along with the Keras framework. For preprocessing and normalization of the dataset, functions from the Scikit learn are used. All the codes are written in Python3.

**6. PROPOSED MODELLING APPROACH**

In this study, five different models are evaluated involving four kinds of ANNs: Convolutional Neural Network, Dense Neural Network, Recurrent Neural Network, Bidirectional LSTM network, and the other model is linear regression (ARIMA). We have compared the performance of the traditional statistical model against the neural networks and linear regression for the prediction of wind power density for the selected stations. The schematic diagram of our proposed methodology is present in Figure 2 with systematic steps required for the analysis.



**Figure 2. Schematic representation of the proposed methodology for the prediction of wind power density**

The modeling framework consists of loading the dataset, pre-processing, splitting the dataset into training and test, windowed dataset, building model, train model, forecast, and error calculation. The structure of our proposed methodology may be described as:

- *Loading the dataset:* We have considered five stations, namely, Chennai, Coimbatore, Madurai, Salem, and Tirunelveli from the time period of January 1980 to May 2018 (more than 39 years) with hourly data.
- *Data cleaning and Pre-processing:* Data cleaning is performed to remove noise from the input dataset after which it is converted from hourly data to daily data.
- *Splitting the dataset:* The given datasets were split into training data (from January 1980 to December 2009, i.e. about 78.097%) and test data (from January 2010 to May 2018, i.e. 21.903%).
- *Preparation of windowed dataset:* By windowing, the sequence of time series data was restructured into a supervised learning problem. In this context, we have used 30 previous time step (window size) entries as an input vector (X) and the next entry (i.e. 31<sup>st</sup> time step) as the output (Y). Thus, the windowed dataset is defined using the parameters window size as 30 and batch size as 32.

### 6.1 Feeding inputs to the models

A sequential modeling approach is applied to the given data in which the hyper-parameters required to train the model were identified as stochastic gradient descent as an optimizer with a momentum value of 0.9. The CNN-LSTM Conv1D has used 32 filters with kernel size as 3 and number of strides as 1. The details of the remaining parameters are provided in Table 2.

The next stage consists of training the model for a fixed number of epochs until the loss saturates to a minimum value. Here we have used 100 epochs for Bidirectional LSTM and simple regression, whereas in simple RNN, Dense Neural network, and CNN-LSTM, we have used 200 epochs to train the model.

During the final stage of data processing, the daily expected results of wind power density were forecasted for the period January 2010 to May 2018 using the

**Table 2. Details of the parameter used by the different NN models**

NN Models	Neuron	Loss Function	Learning rate
2->Bidirectional LSTM   1->Dense	64+64+1	Mean Square Error	0.000001
Simple linear regression	--	Mean Square Error	0.000001
2->SimpleRNN   1->dense	40+40+1	Huber loss	0.00001
2->Dense	10+10+1	Mean Square Error	0.000008
1->Conv1D 2->LSTM 1->dense	32+32+32+1	Huber loss	0.00001

**Table 3. Performance evaluation measures for different models on Chennai station**

Measures /Models	Simple Linear Regression	2-Dense network	1-Conv-2-LSTM	2-Simple-RNN	2-Bidirectional – LSTM
IA	0.7962	0.768	0.8141	0.7958	0.8139
MAE	1.1201	1.0808	1.0661	1.1491	1.0301
MAPE	36.0432	36.7967	31.2511	30.9505	36.0493
MSE	2.0809	1.8847	1.883	2.16	1.7858
RMSE	1.4425	1.3728	1.3722	1.4697	1.3364
SMAPE	0.2722	0.2633	0.2641	0.2881	0.2507

trained model. Further, the accuracy of model predictions were compared using the available data with the predicted data using statistical error estimates.

## 7. RESULTS AND DISCUSSION

The different network architecture and structures are used in this research to compare the forecasting accuracy based on the calculated error. Statistically, there are various expressions for representing the error values depending on the condition of the data type and required data analysis. The Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Index of Agreement (IA), Root Mean Square Error (RMSE), and Symmetric Mean Absolute Percentage Error (SMAPE) were used in this study to evaluate the prediction accuracy of wind power density. Mathematically, they are defined as:

$$MAE = \frac{1}{K} \sum_{m=1}^K |X_m^{actual} - X_m^{predicted}| \quad (10)$$

$$MSE = \frac{1}{K} \sum_{m=1}^K (X_m^{actual} - X_m^{predicted})^2 \quad (11)$$

$$RMSE = \sqrt{\frac{1}{K} \sum_{m=1}^K (X_m^{actual} - X_m^{predicted})^2} \quad (12)$$

$$MAPE = \frac{1}{K} \sum_{m=1}^K \frac{|X_m^{actual} - X_m^{predicted}|}{X_m^{actual}} \quad (13)$$

$$IA = 1 - \frac{\sum_{m=1}^K (X_m^{actual} - X_m^{predicted})^2}{\sum_{m=1}^K (|X_m^{actual} - X_m^{actual}| + |X_m^{predicted} - X_m^{actual}|)^2} \quad (14)$$

$$SMAPE = \frac{100\%}{K} \sum_{m=1}^K \frac{|X_m^{predicted} - X_m^{actual}|}{(|X_m^{actual}| + |X_m^{predicted}|) / 2} \quad (15)$$

where  $K$  represents the number of samples,  $X_m^{actual}$  denotes actual wind power density,  $X_m^{predicted}$  represents forecasted wind power density,  $\overline{X_m^{predicted}}$  refers to the average of forecast values and  $\overline{X_m^{actual}}$  refers to the average of original values and  $0 \leq IA \leq 1$ .

**Table 4. Performance evaluation measures for different models on Coimbatore station**

Measures /Models	Simple Linear Regression	2-Dense network	1-Conv-2-LSTM	2-Simple-RNN	2-Bidirectional - LSTM
IA	0.8271	0.8819	0.8834	0.7713	0.8838
MAE	0.9515	0.7661	0.7558	1.1583	0.7665
MAPE	34.0429	28.7142	27.759	49.4754	28.5666
MSE	1.5218	1.0002	0.9836	1.9736	0.9921
RMSE	1.2336	1.0001	0.9917	1.4048	0.9961
SMAPE	0.2756	0.2228	0.2199	0.3104	0.222

**Table 5. Performance evaluation measures for different models on Madurai station**

Measures /Models	Simple Linear Regression	2-Dense network	1-Conv-2-LSTM	2-Simple-RNN	2-Bidirectional - LSTM
IA	0.7163	0.6264	0.7836	0.7157	0.7863
MAE	1.0455	1.5436	0.8945	1.0343	0.8840
MAPE	45.653	68.577	36.8507	47.606	41.6159
MSE	1.8207	3.6575	1.3417	1.8886	1.3201
RMSE	1.3493	1.9125	1.1583	1.3743	1.1489
SMAPE	0.3154	0.3955	0.2773	0.3238	0.2705

**Table 6. Performance evaluation measures for different models on Salem station**

Measures /Models	Simple Linear Regression	2-Dense network	1-Conv-2-LSTM	2-Simple-RNN	2-Bidirectional - LSTM
IA	0.7547	0.8106	0.8186	0.6042	0.8150
MAE	0.8999	0.7478	0.7475	1.0530	0.7483
MAPE	37.4874	33.0435	31.5056	48.2932	32.4613
MSE	1.3479	0.9642	0.9634	1.8318	0.9643
RMSE	1.1610	0.9820	0.9816	1.3534	0.9820
SMAPE	0.2873	0.2398	0.2422	0.3225	0.2402

**Table 7. Performance evaluation measures for different models on Tirunelveli station**

Measures /Models	Simple Linear Regression	2-Dense network	1-Conv-2-LSTM	2-Simple-RNN	2-Bidirectional - LSTM
IA	0.8244	0.8668	0.8718	0.8405	0.8591
MAE	1.0621	0.8758	0.8665	1.0365	0.8968
MAPE	33.254	29.6367	29.1285	36.2961	28.3281
MSE	1.9082	1.3371	1.3019	1.8678	1.3697
RMSE	1.3814	1.1563	1.1410	1.3667	1.1704
SMAPE	0.2509	0.2071	0.2054	0.2562	0.2116

**Table 8. Average rank based on performance evaluation measures for all models on all five stations**

Measures /Models	Simple Linear Regression	2-Dense network	1-Conv-2-LSTM	2-Simple-RNN	2-Bidirectional - LSTM
IA	3.8	3.6	1.4	4.4	1.8
MAE	4.2	2.8	1.4	4.4	2.2
MAPE	3.6	3.8	1.4	4	2.2
MSE	4	3	1.4	4.6	2
RMSE	4	3.1	1.4	4.6	1.9
SMAPE	3.8	2.6	2	4.8	1.8

The performance evaluation measures for different models for the selected stations are shown in tables (Tables 3-7). It is evolved to identify two significant metrics namely, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for measuring the deviation of the predicted value from the actual value. This highlights the overall accuracy in the prediction of wind power density based on the acquired training experience imparted by the selected algorithms. The highest error in prediction was, however, observed with MAPE for all the stations due to the fact that MAPE values can reach extremes if the unsigned actual values are small. Further, an overall comparison of the performance evaluation measures for all models applied for all stations is comprehended in Table 8. From the results, we can clearly observe the robustness of the 1->Conv|2->LSTM network model in all the time-series data considered. It

is inferred that restructuring the time-series problem to a supervised learning problem using a sliding window concept has facilitated the application of many supervised neural network algorithms and comparisons between them. The 1->Conv|2->LSTM outperformed all other models on a 6/7 ratio for the metrics considered. The specific advantages of this model are attributed to the robustness in addressing multiple issues simultaneously due to the hybrid configuration.

Next to 1->Conv|2->LSTM, Bidirectional LSTM performed satisfactorily in the prediction of the time-series data for the selected locations. We hypothesize about the robustness of the 1->Conv|2->LSTM that convolutional layers cause a denoising effect in the time series and the LSTM layer, being one of the superior variants of RNN, provides better sequence-dependent handling and thus achieves promising results. This also

supports the notion that the performance of hybrid models are superior to the basic NN models, thus signifying their flexibility and adaptability in capturing different forms of relationships in time series data.

From the prediction plots (Figures 3-7), it is evident that the applied NN variants are sophisticatedly capturing the non-linearities present in the given time series data. There is an overall similarity in capturing the generic trend of variation in average wind power density values during the selected testing period in all five stations. It is also observed that windowing of the dataset has reduced the complexity of the model and caused the model to better capture the non-linearities. On a rank-wise comparison, the two poor-performing models are Simple RNN and Simple LR as shown in Table 8. This may be because Simple RNNs and their variants have loops in their recurrent layers to maintain memory over time. Hence it is to be understood that Simple RNNs are less powerful in solving problems that require learning of long-term temporal dependencies as opposed to the LSTMs which have special units for memory in addition to standard units thereby preserving long-term information. On a similar comparison, the suboptimal performance of DNN in the time series dataset come from the fact that they cannot detect repetition in time series and may produce different results on the same input.

Another important aspect causing the deviation in the accuracy of predictions in various modelling approaches is the difference in selected window size. It is concomitantly established from the results that the window size plays an important role in model performance. We have observed that using a window size of 30 gives a superior performance instead of using 10 or 20. This can be explained by the monthly periodicity of dataset distribution. It is also important to note that the commonly used linear models such as ARIMA are incapable of capturing non-linearity in long-term time series. Similarly, the results obtained from ARIMA and AR models were also not significant and promising towards efficient prediction.

The empirical results obtained for the selected stations namely, Chennai, Coimbatore, Madurai, Salem, and Tirunelveli in the state of Tamil Nadu emphasizes the poor performance of ARIMA and the five models that have been presented in this study. A comparison of the performance of those five models undoubtedly establishes the need to have a comprehensive approach in attempting any long-time data for predictive simulation studies. The performances of the Simple RNN and LR models are the least significant among the attempted models. The next in the ranking sequence are dense network and CNN-LSTM. Hence it can be inferred from the present study that the LSTM performance is better for the prediction of wind power density among the five different ANN-based predictive models.

## 8. CONCLUSION

In this study, a prediction model framework is proposed using five different neural network models to present and compare different forms of relationships in estimating wind power density based on 39 years of data from five different stations in Tamil Nadu, India. Among the five non-linear models used for the forecasting, it was found that the 1->Conv1D|2->|LSTM->1 dense model performed better than other models. The performance of simple RNN and linear regression models is poor compared to other models considered in this study. The accuracy of the prediction was confirmed with the ranking of various statistical error measures employed to express the estimated error values as an average of the individual stations. The present model is capable of forecasting wind speed and power density for other offshore locations as well based on the availability of climatological and meteorological datasets. Similar studies may be performed to forecast the wind power density of other stations. In future studies, other models like wavelet threshold denoising (WTD), adaptive neuro-fuzzy inference system (ANFIS), and WTD-RNN-ANFIS model can be used for probabilistic wind power forecasting. Such studies will ensure the efficiency and reliability of different models for wind power forecasting.

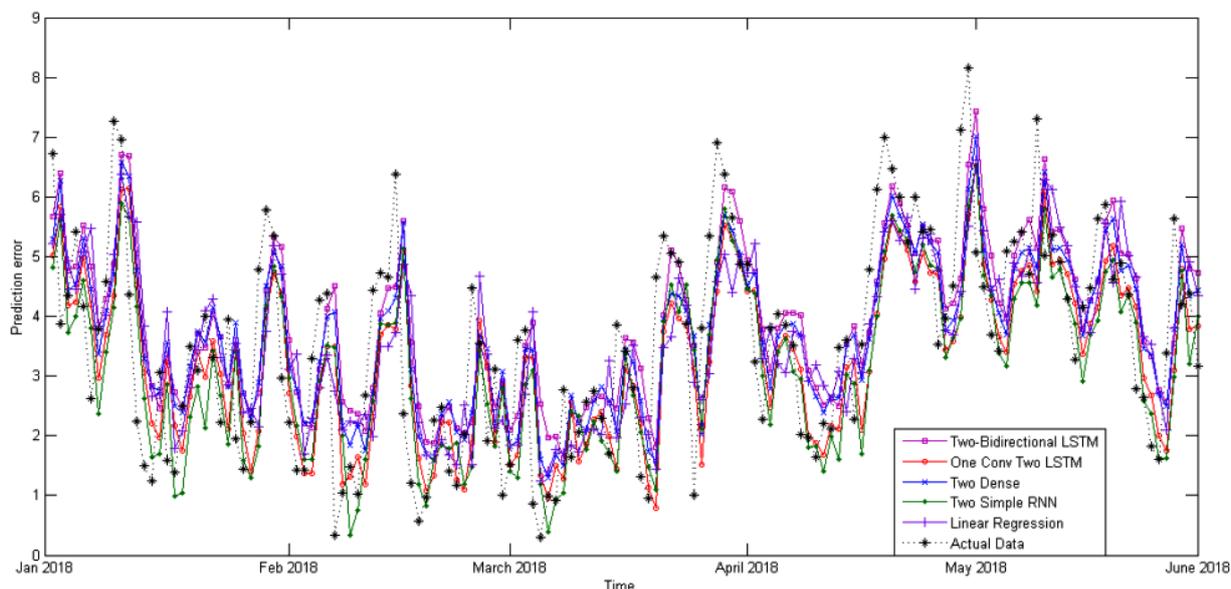


Figure 3. Prediction plot of wind speed data of considered model in Chennai Dataset from Jan 2018 to June 2018.

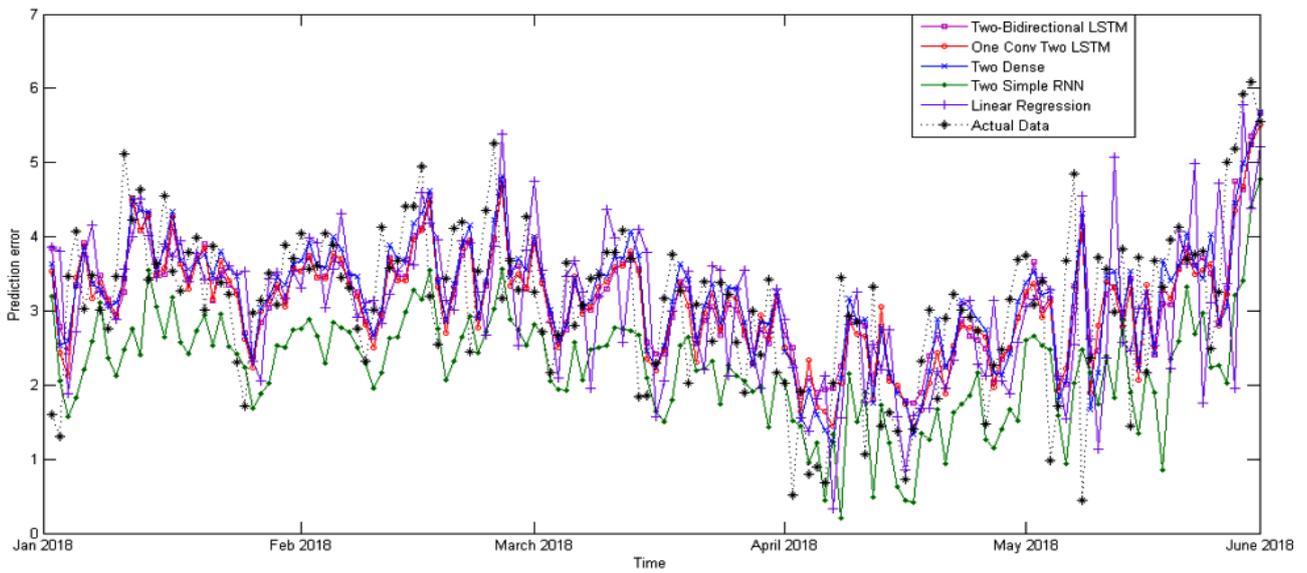


Figure 4. Prediction plot of wind speed data of considered model in Coimbatore dataset from Jan 2018 to June 2018.

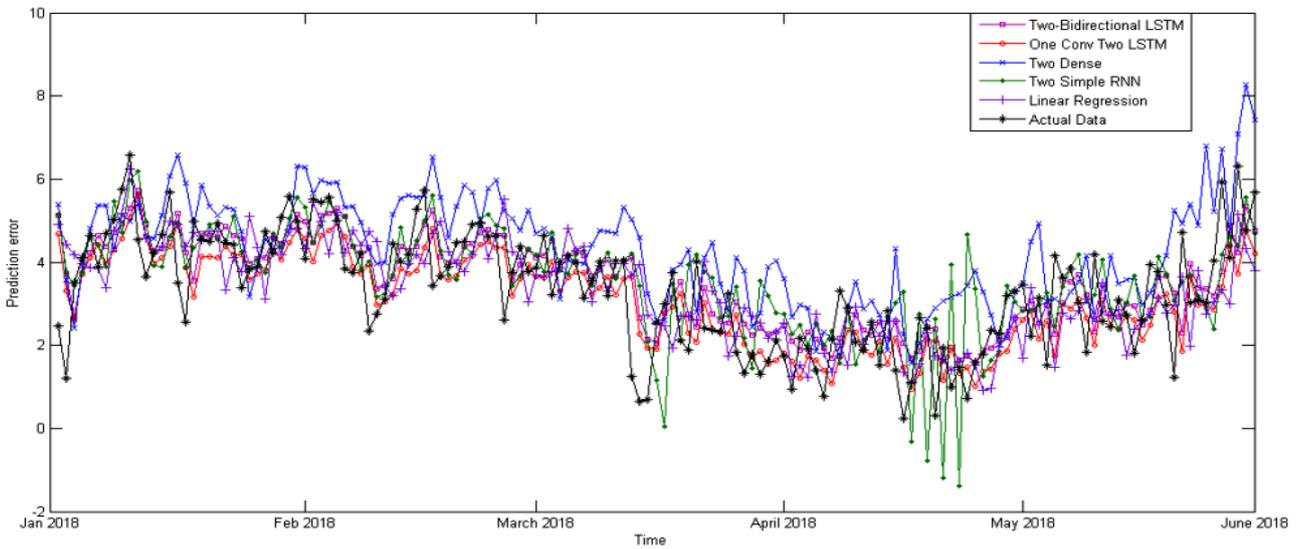


Figure 5. Prediction plot of wind speed data of various model in Madurai Dataset from Jan 2018 to June 2018.

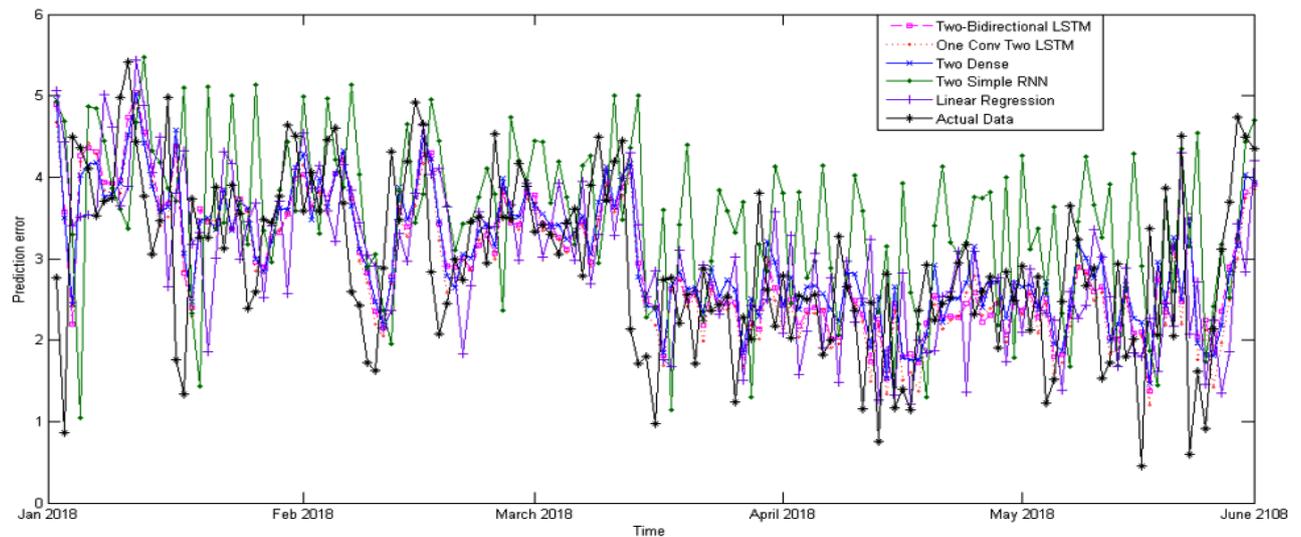


Figure 6. Prediction plot of wind speed data of considered model in Salem Dataset from Jan 2018 to June 2018.

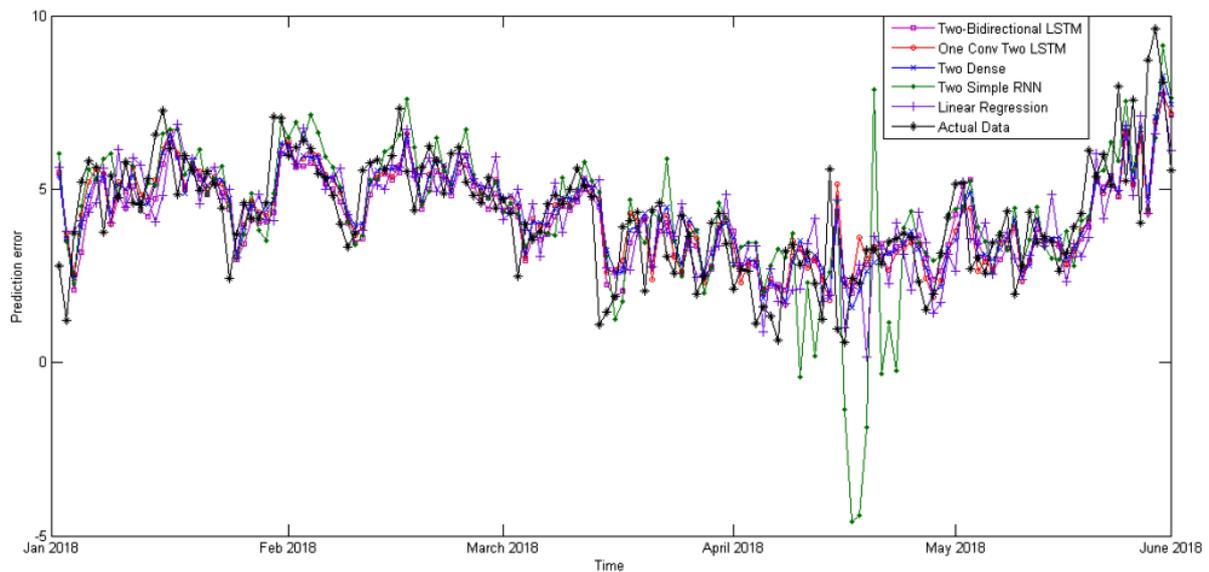


Figure 7. Prediction plot of wind speed data of various model in Tirunelveli Dataset from Jan 2018 to June 2018.

### NOMENCLATURE

$\alpha_k$	Connection weights ( $k=0,1,2,3,\dots,m$ )
$\beta_{lk}$	Connection weights ( $k=0,1,2,3,\dots,m, l=0,1,2,3,\dots,n$ )
$n$	Number of input nodes
$m$	Number of hidden nodes
$y_{t-1}$	Input for single hidden layer feedforward network
$y_t$	Output for single hidden layer feedforward network
$f$	activation function
$h$	kernel
$q$	Indices of resultant row
$r$	Indices of resultant column $i,j$ – relative positions
$P$	Wind power density based on Weibull probability density function $\rho$ – density of air in $\text{kg/m}^3$
$v$	Wind speed in $\text{ms}^{-1}$
$f_w(v)$	Expression as per Arslank – shape factor $c$ – scale factor ( $\text{ms}^{-1}$ )
$\bar{v}$	Mean wind speed
$\sigma$	Standard deviation of the wind speed
$\Gamma(x)$	Gamma function
$E_{PF}$	Parameter defined by Akdag and Dinler
$\bar{v}^3$	Mean cube of wind speed
$\bar{v}^3$	Cube of mean speed
$K$	Number of samples
$X_m^{actual}$	Actual wind power density
$X_m^{predicted}$	Forecast wind power density
$\overline{X_m^{predicted}}$	Average of forecast values
$\overline{X_m^{actual}}$	Average of original values

### ACRONYMS

PSO	Particle Swarm Optimisation
NN	Neural network
ICA	Imperialistic Competitive Algorithm

GA	Generic algorithm
AIM	Abductory Induction Mechanism
PER	Persistence
ANFIS	Adaptive Neuro-Fuzzy Inference System
AWNN	Adaptive Wavelet Neural Network
FFNN	Feed-Forward Neural Network
SVM	Support Vector Machine
ANN	Artificial Neural Network
GP	Genetic Programming
ELM	Extreme Learning Machine
iTLBO	Improved Teaching-Learning-Based Optimisation
RFE	Recursive Feature Elimination
LSTM	Long Short Term Memory Network
TN	Tamil Nadu
NASA	National Aeronautics and Space Administration
EFG	Enhanced Forget Gate
MERRA	Modern-era Retrospective Analysis for Research and Application
ARIMA	Autoregressive Integrated Moving Average
DNN	Dense Neural network
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LR	Linear Regression
DE	Differential Evolution
MSE	Mean Square Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
IA	Index of agreement
RMSE	Root Mean Square Error
SMAPE	Symmetric Mean Absolute Percentage Error
WTD	Wavelet Threshold Denoising

### REFERENCES

- [1] Rehman, S.: Wind energy resources assessment for Yanbo, Saudi Arabia. Energy Conversion Management, Vol. 45, pp. 2019-2032, 2004.
- [2] Rehman, S., Natarajan, N., Vasudevan, M.: Assessment of wind energy potential across varying topo-

- graphical features of Tamil Nadu, India. *Energy Explor. Exploit.*, Vol. 38, pp. 175-200, 2020.
- [3] Haque, A.U., Mandal, P., Kaye, M.E., Meng, J., Chang, L., Senjyu, T.: A new strategy for predicting short-term wind speed using soft computing models. *Renewable Sustainable Energy Reviews*, Vol. 16, pp. 4563-4573, 2012.
- [4] Tascikaraoglu, A., Uzunoglu, M.: A review of combined approaches for prediction of short-term wind speed and power. *Renewable Sustainable Energy Reviews*, Vol. 34, pp.243-254, 2014.
- [5] Natarajan, N., Rehman, S., Nadhini, S.S., Vasudevan, M.: Evaluation of wind energy potential of the state of Tamil Nadu, India based on trend analysis. *FME Transaction* 49: 244-251, 2020
- [6] Rasuo, B.P., Bengin, A.C.: Optimisation of wind farm layout. *FME Transaction*, Vol.38, pp.107-114.
- [7] Rasuo, B., Dinulovic, M., Veg, A., Grbovic, A., Bengin, A.: Harmonization of new wind turbine rotor blades development process: A review. *Renew. Sust. Energy Rev.*, Vol. 39, pp. 874-882.
- [8] Rasuo, B., Bengin, A., Veg, A.: On aerodynamic optimisation of wind farm layout. *Proc. Appl. Math. Mech.*, Vol. 10(1), pp. 539-540.
- [9] Taylor, J.W., Mcsharry, P., Buizza, R.: Wind power density forecasting using ensemble predictions and time series models. *IEEE Transactions on energy conversion*, Vol. 24(3), pp. 775-782, 2009.
- [10] Chang, T.P.: Wind energy assessment incorporating particle swarm optimization method. *Energy Conversion Management*, Vol. 52, pp.1630-1637, 2011.
- [11] Jeon, J., Taylor, J.W.: Using conditional kernel density estimation for wind power density forecasting. *Journal of the American Statistical Association*, Vol. 107(497), pp. 66-79, 2012.
- [12] Bigdeli, N., Afshar, K., Gazafroudi, A.S., Ramandi, M.S.: A comparative study of optimal hybrid methods for wind power prediction in wind farm of Alberta, Canada. *Renewable and Sustainable Energy Reviews*, Vol. 27, pp.20-29, 2013.
- [13] Mohandes, MA, Rehman, S.: Short term wind speed estimation in Saudi Arabia. *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 128, pp.37-53, 2014.
- [14] Shamshirband, S., Iqbal, J., Petkovic, D., Mirhashemi, M.A.: Survey of four models of probability density functions of wind speed and directions by adaptive neurofuzzy methodology. *Adv Eng Softw.*, Vol. 76, pp.148-53, 2014.
- [15] Bhaskar, K., Singh, S.N.: WNN-assisted wind power forecasting using feedforward neural network. *IEEE Trans Sustain Energy*, Vol. 3, pp.306-315, 2012.
- [16] Mohammadi, K., Shamshirband, S., Yee, P.L., Petkovic, D., Zamani, M., Sudheer, Ch.: Predicting the wind power density based upon extreme learning machine. *Energy*, Vol. 86, pp. 232-239, 2015.
- [17] Guo, S., Zhao, H., Zhao, H.: A new hybrid wind power forecasting using the Beveridge-Nelson decomposition method, a relevance vector machine optimized by the ant lion optimizer. *Energies*, Vol.101-20, 2017.
- [18] Wang, X., Wang, C., Li, Q.: Short-term wind power prediction using GA-ELM. *The open electric and electronic engineering journal*, Vol. 11, pp. 48-56, 2017.
- [19] Hossain, M., Mekhilef, S., Afifi, F., Halabi, L.M., Olatomiwa, L., Seyedmahmoudian, M., Horan, B., Stojcevski, A.: Application of the hybrid ANFIS models for long term power density prediction with extrapolation capability. *PLUS One*, 2018.
- [20] Xue, W., Wang, C., Tian, J., Li, K.: Hybrid wind power forecasting based on extreme learning machine and improved TLBO algorithm. *Journal of renewable and sustainable energy* 12: 053309, 2020.
- [21] Araya, I.A., Valle, C., Allende, H.: LSTM-based multi-scale model for windspeed forecasting. *Iberoamerican Congress on Pattern Recognition, CIARP*, 38-45, 2018.
- [22] Chen, J., Zeng, G.Q., Zhou, W., Du, W., Lu, K.D.: Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. *Energy Conversion and Management*, Vol. 165, pp. 681-695, 2018.
- [23] Chen, M.R., Zeng, G.Q., Lu, L.D., Weng, J.: A two-layer nonlinear combination method for short-term wind speed prediction based on ELM, ENN and LSTM. *IEEE Internet of Things Journal*, Vol. 6(4), pp. 6997-7010, 2019.
- [24] Hu, Y.L., Chen, L.: A nonlinear hybrid wind speed forecasting model using lstm network, hysteretic elm and differential evolution algorithm. *Energy Conversion and Management*, Vol. 173, pp.123-142, 2018.
- [25] Lu, K., Sun, W.X., Wang, X., Meng, X.R., Zhai, Y., Li, H.H., Zhang, R.G.: Short-term wind power prediction model based on encoder-decoder LSTM. *IOP Conf. Series: Earth and Environmental Science*, Vol. 186, 012020.
- [26] Yang, T., Wang, H., Aziz, S., Jiang, H., Peng, J.: A novel method of wind speed prediction by peephole LSTM. *International Conference on Power System Technology*, 364-369, 2018.
- [27] Zhang, L., Wang, B., Fang, B., Ma, H., Yang, Z. et al.: Two-stage short-term wind speed prediction based on LSTM-LSSVM-CFA. *2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, 1-6, 2018.
- [28] Xu, G., Xia, L.: Short-term prediction of wind power based on adaptive LSTM. *2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1-5, 2018.
- [29] Yu, R., Gao, J., Yu, M., Lu, W., Xu, T., Zhao, M., Zhang, J., Zhang, R., Zhang, Z.: LSTM-EFG for wind power forecasting based on sequential

- correlation features. *Future Generation Computer Systems*, Vol. 93, pp. 33-42, 2019.
- [30] Scarselli, Franco, and Ah Chung Tsoi. "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results." *Neural networks* 11.1 (1998): 15-37.
- [31] Miljanovic, M.: Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction. *Indian Journal of Computer and Engineering*, Vol. 3 (1), 2012.
- [32] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, Vol. 15, pp. 1929–1958, 2014.
- [33] Cheng, Lilin, et al.: Ensemble Recurrent Neural Network Based Probabilistic Wind Speed Forecasting Approach. *Energies*, Vol. 8, doi:10.3390/en11081958, 2018.
- [34] Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, Vol. 5(2), pp. 157–166, 1994.
- [35] Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, Vol. 6, pp. 107–116, 1998.
- [36] Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural. Comput.*, Vol. 9, pp. 1735–1780, 1997.
- [37] Jason, B.: *Long Short-Term Memory Networks With Python*. Machine Learning Mastery, 2-94, 2017.
- [38] Pishgar-Komleh, Keyhani, A., Sefeedpari, P.: Wind speed and power density analysis based on Weibull and Rayleigh distributions(a case study: Firouzkoo county of Iran). *Renew. Sust. Energy Rev.*, Vol. 42, pp. 313-322, 2015.
- [39] Mohammadi, K., Mostafaeipour, A.: Economic feasibility of developing wind turbined in Aligoodarz, Iran. *Energy Convers. Manage.*, Vol. 76, pp. 645-653, 2013.
- [40] Akdag, S.A., Dinler, A.: A new method to estimate Weibull parameters for wind energy applications. *Energy Convers. Manage.*, Vol. 50, pp. 1761-1766, 2009.

---

### **КРАТКОРОЧНО ПРЕДВИЂАЊЕ ГУСТИНЕ СНАГЕ ВЕТРА КОРИШЋЕЊЕМ КОНВОЛУЦИОНЕ LSTM МРЕЖЕ**

**Д. Гупта, В. Кумар, И. Ајус, М. Васудеван, Н.  
Натараџан**

Ефикасно добијање обновљиве енергије из ветра зависи од поузданости прорачуна карактеристика ветра, оптимизације инсталирања ветропарка и радних услова. Предвиђање искоришћавања потенцијала енергије ветра није извесно на основу варијабилности понашања ветра. Прорачун густине снаге ветра на основу емпиријских модела захтева напредну обраду података како би се обезбедила тачност и поузданост ирачунавања потенцијала ветра. Анализира се поузданост машинског учења базираног на вештачким неуронским мрежама у предвиђању густине снаге ветра за пет станица у држави Тамил Наду у Индији коришћењем пет различитих нелинеарних модела. Модели CNN, DNN, RNN, LSTM мрежа и линеарна регресија су примењени за поређење података за период јан. 1980 – мај 2018. Резултати показују да је перформанса (1->Conv1D\2->LSTM) боља од других модела за израчунавање густине снаге ветра са минималном грешком (средња апсолутна грешка и средња квадратна грешка).