Energy Engineering

Tech Science Press

DOI: 10.32604/ee.2023.026185

ARTICLE

Improving Performance of Recurrent Neural Networks Using Simulated Annealing for Vertical Wind Speed Estimation

Shafiqur Rehman[1,*], **Hilal H. Nuha**[2], **Ali Al Shaikhi**[3], **Satria Akbar**[2] **and Mohamed Mohandes**[1,3]

[1]Interdisciplinary Research Center for Renewable Energy and Power Systems (IRC-REPS), KFUPM, Dhahran, 31261, Saudi Arabia

[2]HUMIC Engineering, School of Computing, Telkom University, Bandung, 40257, Indonesia

[3]Electrical Engineering Department, King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, 31261, Saudi Arabia

[*]Corresponding Author: Shafiqur Rehman. Email: srehman@kfupm.edu.sa

Received: 22 August 2022 Accepted: 26 December 2022

ABSTRACT

An accurate vertical wind speed (WS) data estimation is required to determine the potential for wind farm installation. In general, the vertical extrapolation of WS at different heights must consider different parameters from different locations, such as wind shear coefficient, roughness length, and atmospheric conditions. The novelty presented in this article is the introduction of two steps optimization for the Recurrent Neural Networks (RNN) model to estimate WS at different heights using measurements from lower heights. The first optimization of the RNN is performed to minimize a differentiable cost function, namely, mean squared error (MSE), using the Broyden-Fletcher-Goldfarb-Shanno algorithm. Secondly, the RNN is optimized to reduce a non-differentiable cost function using simulated annealing (RNN-SA), namely mean absolute error (MAE). Estimation of WS vertically at 50 m height is done by training RNN-SA with the actual WS data a 10–40 m heights. The estimated WS at height of 50 m and the measured WS at 10–40 heights are further used to train RNN-SA to obtain WS at 60 m height. This procedure is repeated continuously until the WS is estimated at a height of 180 m. The RNN-SA performance is compared with the standard RNN, Multilayer Perceptron (MLP), Support Vector Machine (SVM), and state of the art methods like convolutional neural networks (CNN) and long short-term memory (LSTM) networks to extrapolate the WS vertically. The estimated values are also compared with real WS dataset acquired using LiDAR and tested using four error metrics namely, mean squared error (MSE), mean absolute percentage error (MAPE), mean bias error (MBE), and coefficient of determination (R^2). The numerical experimental results show that the MSE values between the estimated and actual WS at 180 m height for the RNN-SA, RNN, MLP, and SVM methods are found to be 2.09, 2.12, 2.37, and 2.63, respectively.

KEYWORDS

Vertical wind speed estimation; recurrent neural networks; simulated annealing; multilayer perceptron; support vector machine

Nomenclature

AI	Artificial intelligence
ANN	Artificial neural networks
BFGS	Broyden-Fletcher-Goldfarb-Shanno

CNN	Convolution neural networks
GHG	Greenhouse gases
GW	Gigawatt
GWh	Gigawatt hour
kW	Kilowatt
kWh	Kilowatt hour
LSTM	Long short-term memory
m	Meter
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MBE	Mean bias error
MLP	Multilayer perceptron
MSE	Mean square error
MW	Megawatt
MWh	Megawatt hour
RNN	Recurrent neural networks
RNN-SA	Recurrent neural networks with simulation annealing
SA	Simulation annealing
SMO	Sequential minimal optimization
SVM	Support vector machine
R^2	Coefficient of determination
WS	Wind speed (m/s)

1 Introduction

Wind power plants have been used globally as a renewable energy source for household and industrial energy demand. Total deployed wind power capacity, worldwide in 2020 cumulatively, reached 743 GW [1], and more wind power plants are being planned to be installed to meet the demand and to combat the greenhouse gases (GHG) emissions, globally. However, wind power plant installation requires accurate and precise data on the availability and changes in wind speed (WS) of any site since WS may vary rapidly and increase with height as shown in Fig. 1 where WS at 180 m is higher than that at 140 to 20 m. Therefore, more energy can be harvested if the hub height of the turbine is high. Today's modern wind turbines have a rotor diameter of 120 m or more with a hub height of 100 m or more where the investment and installation costs are in proportion with the dimension. This indicates that an accurate estimation of the wind profile is needed at the turbine hub height for accurate estimation of the energy that can be generated from the turbine to be installed and prevents failure to achieve return of investment due to inaccurate wind profile investigation.

The increase in WS with height is due to less human activities and lower surface roughness at higher heights. In addition, atmospheric conditions at a higher height have less friction and are more stable which may account for higher WS at higher heights [2]. However, in most countries, WS is measured at lower heights from 10 to 40 m. Therefore, for accurate estimation of wind power potential in an area for turbine positions above 40 m, WS must be estimated for higher turbine hub heights using physical, statistical, or machine learning intelligence techniques.

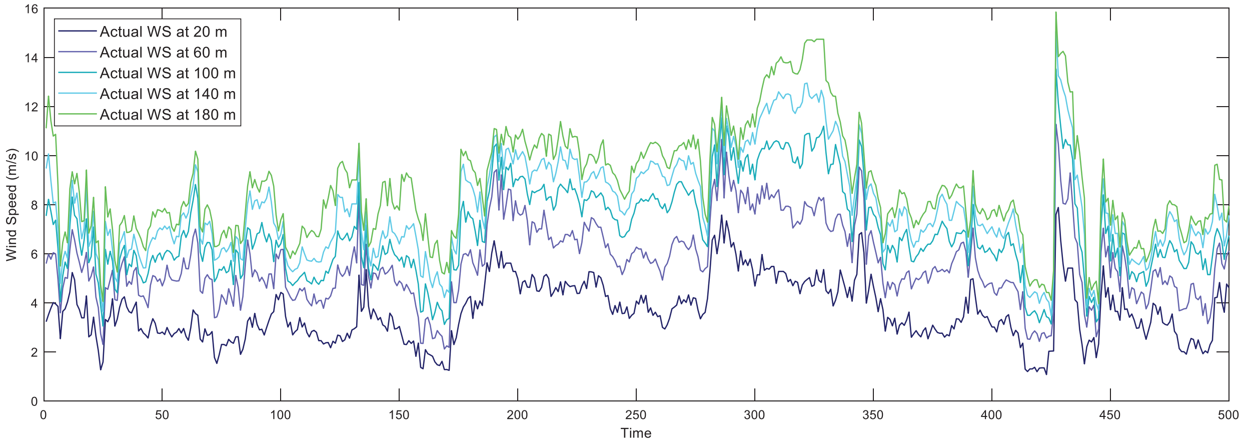


Figure 1: Measured wind speeds at 20, 60, 100, 140, and 180 m

Several approaches have been used to estimate WS at higher positions based on measurements at lower heights as indicated by increasing WS average and its percentage increment in Fig. 2. Under stable atmospheric conditions, logarithmic approaches and power-law techniques for WS estimation have shown good accuracy [3]. Parametric approaches require fields survey to obtain meteorological coefficients as reported by Banuelos-Ruedas et al. [4] that used Hellman's law, 1/7th power law, and a logarithmic approach to estimate WS at a given altitude. In addition to parametric methods, artificial intelligence (AI) approaches have been widely used for the temporal prediction of WS [5,6] due to its practicality. For example, Islam et al. [7] proposed a hybrid artificial neural network for WS estimation up to 100 m using measurements at the height of 10–40 m. Türkan et al. [8] compared seven different AI techniques for vertical WS estimation at a height up to 30 m using a WS measured at a height of 10 m. Emeksiz [9] developed multigene genetic programming for the temporal estimation of WS at different altitudes. Mohandes et al. [10] employed the Restricted Boltzmann Machine (RBM) technique to pre-train the neural network to estimate WS up to 120 m height using WS measurements at lower altitudes. Ti et al. [11] proposed artificial neural networks (ANN) based wake model to predict power produced by a wind farm. Wake modelling was also combined with the bilateral convolutional neural networks for dynamic wind farm power prediction [12]. Yang et al [13] developed double layer machine learning framework using ANN and Bayesian machine learning for wind farm power prediction. It can be noticed that the modern and advanced AI approaches have been widely investigated for temporal WS estimation, but limited number of works are used for vertical WS estimation tasks using WS measured at lower heights. Therefore, the literature above has shown the necessity for the investigation of a hybrid AI method for WS estimation above 120 m.

The contributions and novelties of this study are as follows:

1. The paper introduces two steps performance enhancement of recurrent neural networks using gradient based and simulated annealing methods to estimate WS at a higher turbine position using the measured value at lower heights.
2. This paper presents the comparison of the proposed method with the standard RNN, Multilayer Perceptron (MLP), Support Vector Machine (SVM), and state of the art methods like convolutional neural networks (CNN) and long short-term memory methods.
3. A real dataset from Dhahran, Kingdom of Saudi Arabia is used to evaluate the accuracy of the proposed method.

4. All methods are evaluated using error metrics such as mean squared error (MSE), mean absolute percentage error (MAPE), mean bias error (MBE), and coefficient of determination (R^2).

The remainder of this paper is structured as methodology, experimental results, and conclusion.

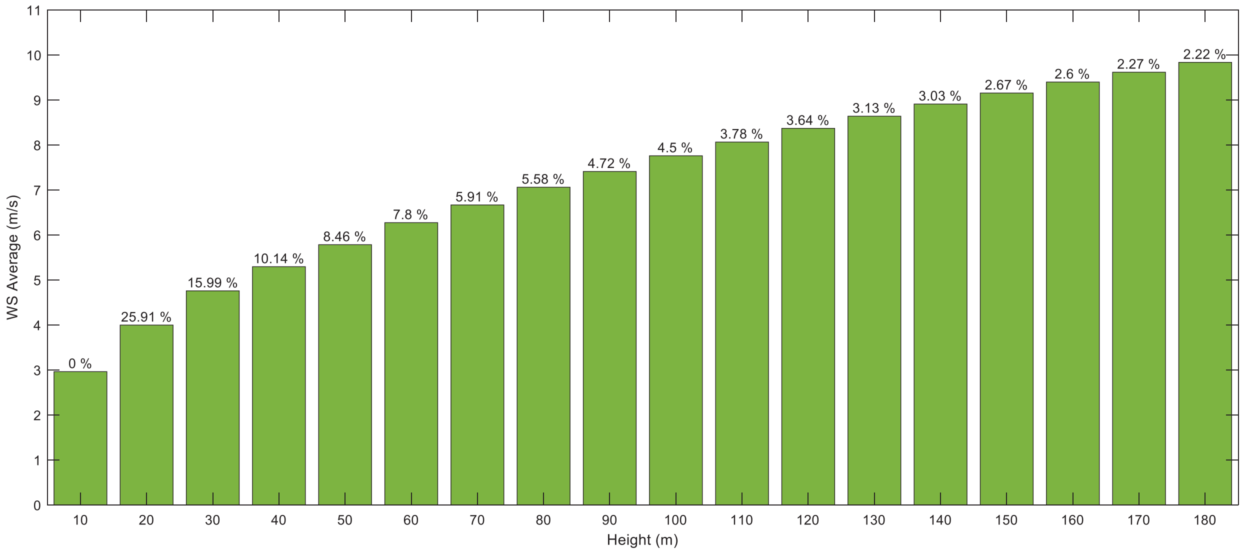


Figure 2: Wind speed average and its corresponding percentage increment

2 Methodology

2.1 Proposed Method and Its Mathematical Models

In this paper, we propose a hybrid method, namely recurrent neural networks with simulated annealing (RNN-SA) for vertical WS extrapolation. The RNN uses the Elman model where the hidden unit output is connected back to itself using an adjustable recurrent weight (w)in addition to feeding the output layer with weights (V), as shown in Fig. 3. Although the architecture is simple, this model is capable of performing a variety of prediction tasks [14,15]. The data is pre-processed in two stages. First, the data from the input (U) and recurrent connections are provided to the hidden units as inputs. Next, the output from the hidden units is sent to both the output layer and through the recurrent connections back to the hidden units. Mathematically, the output of the kth hidden unit $h_k(n)$ given the nth input vector is given by:

$$h_k(n) = \sigma\left(U_k x(n) + W_k h_k(n-1) + b_k\right) \tag{1}$$

where b is the bias vector. The input x represents the actual and estimated WS from the lower heights. The output layer provides the WS value at one step higher. The output $y(n)$ of the nth sample is given by:

$$y(n) = \sigma\left(Vh(n) + b_y\right) \tag{2}$$

This study uses the tangent-hyperbolic activation function defined by:

$$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{3}$$

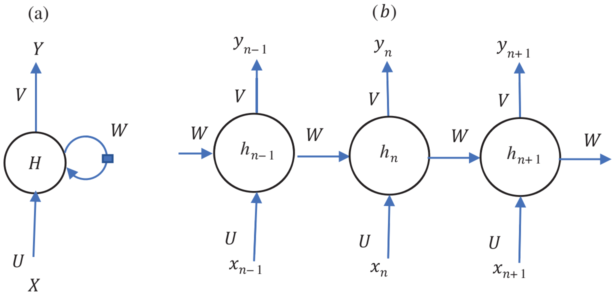


Figure 3: RNN with Elman model for WS extrapolation

In this study, the numerical experiment is carried out by dividing the data into three parts, i.e., 70%, 10%, and 20% for training, validation, and testing, respectively. The RNN-SA is trained using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [16]. Since the model uses a medium number of units and layers, the BFGS algorithm is the most suitable for unconstrained optimization [17]. The BFGS algorithm is utilized to obtain the RNN model with minimum mean squared error (MSE):

$$E = MSE = \frac{\sum_{n=1}^{N} \left(y_n - \hat{y}_n\right)^2}{N} \tag{4}$$

It can be noticed that the MSE is a continuously differentiable function that represents the difference between the predicted outputs ($\hat{y}$) and the actual values (y). The BFGS algorithm uses an iterative update to obtain the weights that minimize the cost function $e\left(w\right) = E$ from Eq. (4). The weight update is specified by:

$$w_{k+1} = w_k + \mu_k d_k \tag{5}$$

where μ_k and d_k denote the step size and direction at iteration k. The direction update is given by:

$$\begin{cases} d_k = -B_k^{-1}\nabla \mathrm{e}(w_k) \\ B_{k+1} = B_k - \dfrac{B_k s_k s_k^T B_k}{S_k^T B_k s_k} + \dfrac{y_k y_k^T}{y_k^T s_k}, B_0 = I \\ s_k = w_{k+1} - w_k, y_k = \nabla \mathrm{e}\left(w_{k+1}\right) - \nabla \mathrm{e}(w_k) \end{cases} \tag{6}$$

where B_k is an approximation of the Hessian matrix that represent the second derivative of the cost function. The results of the validation data showed that the RNN method with $L = 20$ hidden units and number of iterations, $K = 20$, exhibited the best performance. The model is further optimized using simulated annealing (SA), which is an optimization method that accepts not only better solution candidates but also includes not so good ones with a certain likelihood to enable the optimization to explore a new area for new candidates [18]. In the sense of the RNN, the SA is utilized further to optimize the trained RNN with respect to a non-differentiable cost function, namely, the mean absolute error (MAE):

$$MAE = \frac{1}{N}\sum_{n=1}^{N}\left|y_n - \hat{y}_n\right| \times 100\% \tag{7}$$

Given the cost function above, the solution candidate acceptance is given by:

$$p = \frac{1}{1 + e^{\frac{\Delta E}{\max(T)}}} \tag{8}$$

where ΔE is the difference between new and old cost functions and T is the temperature governed by the following relation:

$$T = T_0 \times 0.95^k \tag{9}$$

where the initial temperature T_0 is set to 100 and k is the iteration number. The proposed RNN-SA utilizes two-stage optimizations. First, the BFGS method trains the RNN to minimize the MSE. The first cost function uses a smooth quadratic function that allows gradient-based optimization to obtain the best solution. Second, the SA optimizes the trained RNN weights to achieve the minimum MAE. Since the MAE uses a non-smooth absolute function, the SA offers an optimal solution for a complex recurrent model. The proposed two steps optimization approach to improve the performance metrics with different methods namely, the MSE with continuous derivative and the non-smooth MAE. The traditional RNN is typically trained using gradient based method with differentiable cost function. Whereas, the proposed RNN, in addition to the gradient based method, is further optimized using SA to enhance the performance. Both RNNs use the same architecture with recurrent units as shown in Fig. 4. In the implementation, the maximum number of iterations of the SA method is $K = 10$.

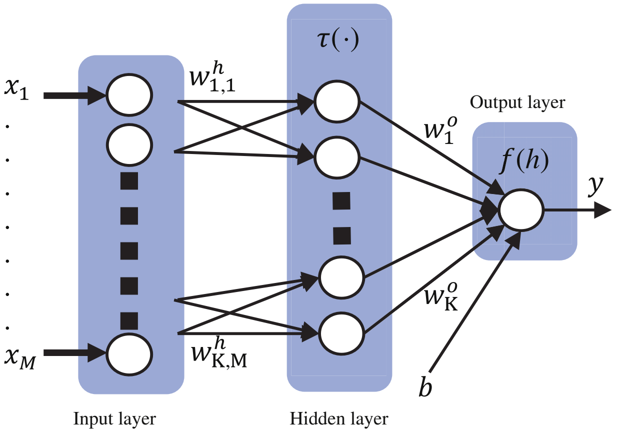


Figure 4: Multi-layer perceptron model

2.2 Multilayer Perceptron

In this study, the multilayer perceptron (MLP) is used as a benchmark for the comparison of the performance of RNN-SA. MLP has a much simpler structure than RNN [19], as shown in Fig. 4. In contrast to RNN, MLP uses a feed-forward structure that sequentially processes the data from the input layer to the output layer. Mathematically, the output of the kth hidden unit $h_k(n)$ given the nth input vector is obtained by:

$$h_k(n) = \tau\left(\mathrm{W}^{\mathrm{h}} x(n) + b_k\right) \tag{10}$$

where V_k is the input weight matrix. Given the hidden unit values, the output of the MLP is defined by:

$$y(n) = \mathrm{W}^{\mathrm{o}} h(n) + b \tag{11}$$

This study utilizes gradient descent with momentum and adaptive learning rate (μ) backpropagation to train the MLP. This method is able to train any model with differentiable activation and cost function, which is suitable in this case where the activation and cost function uses differentiable tangent hyperbolic and MSE, respectively. Each weight is adjusted according to gradient descent defined by:

$$w_k = w_{\mathrm{k-1}} + \partial \mathrm{w}_k \tag{12}$$

where the weight update ($\partial \mathrm{w}_{\mathrm{n}}$) is calculated by:

$$\partial \mathrm{w}_k = m\partial w_{\mathrm{k-1}} + m\mu \frac{(\partial e_k)}{\partial w_k} \tag{13}$$

where $m = 0.9$ and $\mu = 0.01$ denote the momentum coefficient and learning rate, respectively [20]. The cost function is evaluated at each epoch. If the cost function decreases toward the cost target, then the learning rate is increased by the factor $\Delta\mu_{(+)} = 1.05$. If the cost function improves by more than the factor $\Delta MSE_{max} = 1.04$, the weight update that increased the cost function is canceled, and the learning rate is updated by the factor $\Delta\mu_{(-)} = 0.7$. The experiments use $H = 20$ hidden units. Training halts when the maximum number of iterations ($K = 1200$) is exceeded.

2.3 Support Vector Machine

The last benchmarking method used in this paper is the support vector machine (SVM) for regression [21]. Theoretically, SVM is represented by a mapping function $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, with input vectors $\boldsymbol{x}_i \in \mathbb{R}^M$ and output targets $y_i \in \mathbb{R}$. The SVM attempts to create the relationship between $\boldsymbol{x}_i$ and expected results $f(\boldsymbol{x}_i)$ as follows:

$$y = f(\boldsymbol{x}_i) = (\boldsymbol{\omega} \cdot \boldsymbol{x}_i) + b \tag{14}$$

where $\boldsymbol{\omega} \in \mathbb{R}^M$ and b are the weight vectors and threshold, respectively. From this model, the target of SVM is to obtain the weights that minimize the following error function:

$$E(\boldsymbol{\omega}) = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} |y_i - f(\boldsymbol{x}_i)|_\epsilon + \lambda\, |\omega|^2 \tag{15}$$

where λ is a constant and the robust error function $|z|_\epsilon$ is defined as follows:

$$|z|_\epsilon = \begin{cases} 0, & \mathit{if}\ |z| < \epsilon \\ |z|, & \mathit{otherwise}. \end{cases} \tag{16}$$

The SVM model is trained using sequential minimal optimization (SMO), where the number of inputs determines the dimension of ω. Since the model has four input elements (10, 20, 30, and 40 m), the model for estimating WS at 50 m uses $M = 4$.

2.4 State of the Art Methods

Recently, state of the art methods like deep learning have been used for many tasks including wind speed prediction. For instance, convolutional neural networks (CNN) is used for WS prediction in a wind farm in Hebei Province, China with high accuracy [22]. In addition to that, long short-term memory (LSTM) network is used for WS prediction on a wind farm in Inner Mongolia [23]. These two methods are also used as benchmarking methods in this paper.

2.5 Error Measures

This study employs four error measures based on the difference between the measured values (y), the estimated outputs ($\hat{y}$), and the average of the measured values ($\overline{y}$), including MSE, mean bias error (MBE), mean absolute percent error (MAPE), and the coefficient of determination (R^2). These error measures are calculated using the following equations:

$$MBE = \frac{\sum_{n=1}^{N} \left(\hat{y}_n - y_n\right)}{N} \tag{17}$$

$$MAPE = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{y_n - \hat{y}_n}{y_n}\right| \times 100\% \tag{18}$$

$$R^2 = \left(1 - \frac{\sum_{n=1}^{N}(y_n - \overline{y})^2}{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}\right) \times 100\% \tag{19}$$

3 Numerical Experimental Results

This section provides the description of the experimental setup and the results. The estimation starts by training each model using WS values at heights 10, 20, 30, and 40 m, as inputs and actual WS at the height of 50 m as the target. Next, the actual WS at 10–40 m and the estimated WS at a height 50 m are used to train new model with five inputs to estimate the WS at 60 m height. This process, which uses the actual and estimated WS values at lower heights to predict the WS values at one level higher is further repeated until the estimation of the WS at a height of 180 m using the actual WS at 10–40 m, and the estimated WS at 50–170 m is obtained.

3.1 Wind Speed Dataset

Due to lack of resources in some countries, WS is typically measured up to a height of 40 m and its extrapolation to higher levels require sophisticated methods. Therefore, the WS at the higher heights is obtained using models, given the WS measured at lower heights. As shown in Fig. 1, the WS increases with height but in a random fashion [24,25]. This paper uses WS data acquired in Dhahran, Saudi Arabia during 20 June 2015 to 29 February 2016 using a LiDAR wind measurement device at 10, 20, 30, ..., 180 m heights. The WS is scanned every three second and averages at each 10 min are saved and used. The data is divided into training, validation, and testing at 70%, 10%, and 20%, respectively. The LiDAR based measured WS is used as the reference for the actual WS.

Table 1: The performance of the RNN-SA under validation data for WS estimation at 50–180 m heights based on measured WS at 10–40 m heights

Heights (m)	50	60	70	80	90	100	110	120	130	140	150	160	170	180
MSE	0.024	0.098	0.162	0.263	0.371	0.516	0.628	0.780	0.931	1.051	1.193	1.359	1.457	1.649
MBE	0.012	0.037	0.048	0.068	0.081	0.093	0.061	0.024	0.000	0.056	0.054	0.053	0.049	0.144
MAPE (%)	2.06	3.79	4.57	5.57	6.31	7.22	7.63	8.10	8.49	8.85	9.15	9.37	9.54	10.16
R^2 (%)	99.03	96.56	94.72	92.24	89.78	86.96	84.75	82.32	80.35	78.74	77.07	75.10	73.94	72.53

3.2 Results

Table 1 shows the performance of the RNN-SA under validation data for WS estimation at 50–180 m heights based on measured WS at 10–40 m heights. The model with the best performance is further used for testing. Table 2 summarizes the testing data experimental results of the RNN-SA, standard RNN, MLP, SVM, CNN, and LSTM techniques in terms of MSE. It is obvious that MSE values for RNN-SA increases progressively from 0.027 to 2.089 while that for MLP from 0.049 to 2.367, corresponding to 50 to 180 m heights. Similarly, for the SVM method, the MSE values varied from 0.032 at 50 to 2.632 at 180 m. In all the numerical experiments, that RNN-SA method resulted in lowest values of MSE at all heights. The developed method is able to achieve better MSE than the standard RNN for all heights. It also can be noticed that the RNN-SA outperformed other methods based on the obtained MSE values. Table 3 reports the MBE of all methods and heights. MBE may not quantify the performance or accuracy of the estimation methods; however, it indicates whether the method is over or under predictive [26]. Positive MBE represents over prediction where the estimated values are larger than the actual one. Whereas, negative MBE indicates under-prediction where the estimated values are less than the actual. The experimental results show that all methods have negative values for all the heights. This indicate that the predicted values are less than the measured WS or the model underestimated the WS at all heights.

Table 2: MSE of the WS estimation at 50–180 m heights based on measured WS at 10–40 m heights

Heights (m)	50	60	70	80	90	100	110	120	130	140	150	160	170	180
RNN-SA	0.027	0.105	0.182	0.313	0.463	0.674	0.834	1.054	1.259	1.416	1.601	1.817	1.929	2.089
RNN	0.028	0.113	0.185	0.314	0.475	0.681	0.855	1.064	1.262	1.426	1.614	1.831	1.950	2.123
MLP	0.049	0.120	0.209	0.351	0.510	0.739	0.931	1.165	1.321	1.552	1.735	2.005	2.168	2.367
SVM	0.032	0.145	0.239	0.401	0.581	0.834	1.042	1.364	1.516	1.852	2.017	2.323	2.365	2.632
CNN	0.125	0.265	0.539	0.392	1.059	1.402	1.092	1.411	2.241	1.833	2.437	1.940	3.265	2.378
LSTM	0.054	0.140	0.221	0.458	0.570	0.726	1.469	1.160	1.440	2.396	1.750	1.982	2.676	3.497

Table 3: MBE (m/s) of the WS estimation at 50–180 m heights based on measured WS at 10–40 m heights

Heights	50	60	70	80	90	100	110	120	130	140	150	160	170	180
RNN-SA	−0.02	−0.03	−0.05	−0.07	−0.09	−0.11	−0.15	−0.20	−0.22	−0.18	−0.20	−0.20	−0.20	−0.13
RNN	−0.02	−0.03	−0.04	−0.06	−0.08	−0.14	−0.15	−0.13	−0.18	−0.19	−0.18	−0.18	−0.20	−0.14
MLP	−0.01	−0.04	−0.06	−0.07	−0.11	−0.15	−0.17	−0.18	−0.20	−0.21	−0.20	−0.20	−0.21	−0.20
SVM	−0.02	−0.07	−0.09	−0.12	−0.15	−0.18	−0.20	−0.24	−0.24	−0.26	−0.25	−0.28	−0.26	−0.26
CNN	−0.16	−0.31	−0.39	0.03	−0.56	−0.84	−0.45	−0.58	−0.98	−0.58	−0.82	−0.31	−1.14	−0.18
LSTM	0.983	0.959	0.940	0.908	0.870	0.838	0.810	0.776	0.758	0.724	0.710	0.685	0.675	0.656

Fig. 5 compares the MAPE values for all the methods and show almost the same pattern as in case of MSE (Table 1). The MAPE value of RNN-SA, standard RNN, MLP, and SVM increased from 2.2% to 11.2%, 2.3% to 11.3%, 2.9% to 11.7%, and 2.4% to 12.2% for 50 to 180 m of heights, respectively. The proposed method achieves the least MAPE over all methods for all the heights. RNN methods proved to be the next best estimator of WS at all heights while MLP the third best option. Fig. 6 visualizes the R^2 values for all heights and methods. The figure shows that the estimated values deviate more and more from the actual values as height increases. It may be accounted to the fact that actual WS values at 10–40 m only are being used for the estimation at higher heights. So, the estimated WS near the 40 m height has higher accuracy, and as the height increases, the correlation between the estimated and the actual WS decreases. The RNN-SA outperforms the other methods and achieved the highest R^2 values for all heights.

The scatter plots between the estimated RNN-SA with actual WS values at 60, 100, 140, and 180 m are shown in Figs. 7a–7d, respectively. It is observed that the scatter plots start to diverge as height increases. This is confirmed by the temporal variation of actual and extrapolated WS values at 60, 100, 140, and 180 m heights, shown in Figs. 8a–8d, respectively. It is noticed that the estimated WS values follow the temporal trend of actual values at all heights, as depicted in Fig. 8. The figure also shows the coefficient of determination values for all methods. However, the trends of estimated WS at 180 (Fig. 8d) m are away from the actual values compared to the trends at 60 m (Fig. 8a), 100 m

(Fig. 8b), and 140 m (Fig. 8c). The maximum number of estimated WS values for training contribute to these deviations.

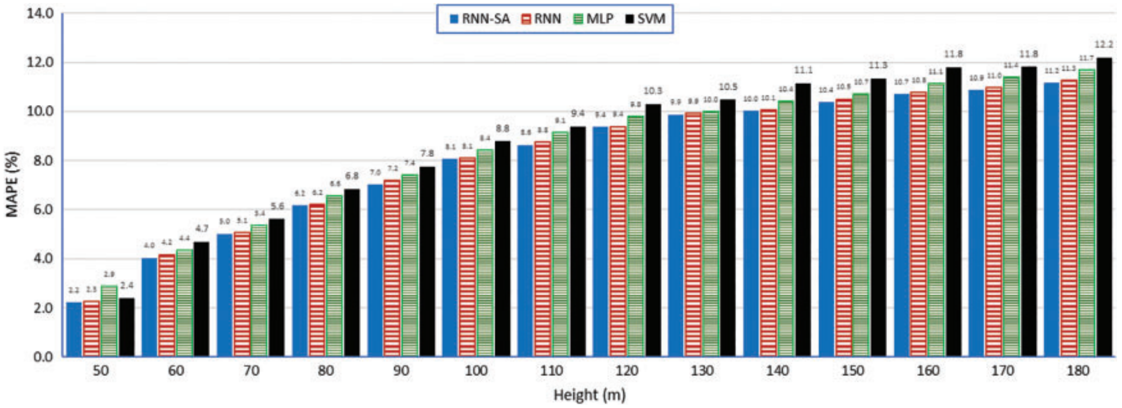


Figure 5: MAPE of different methods and all heights

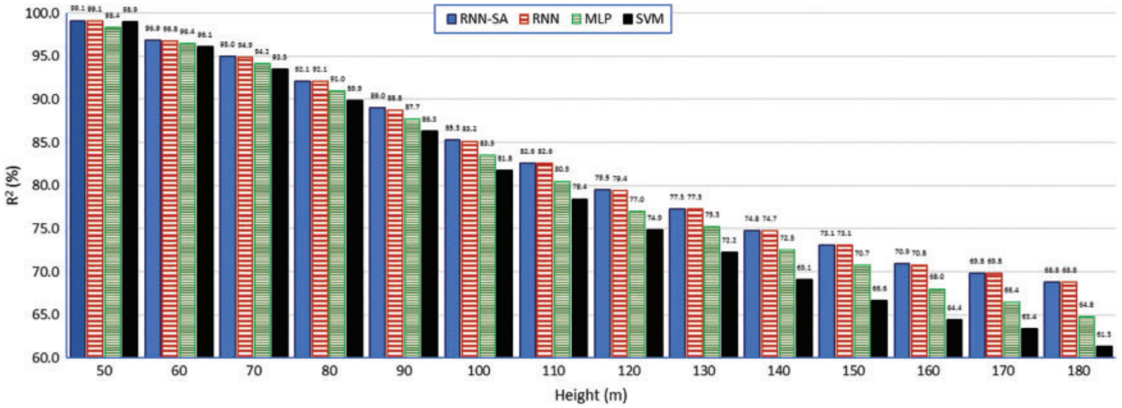
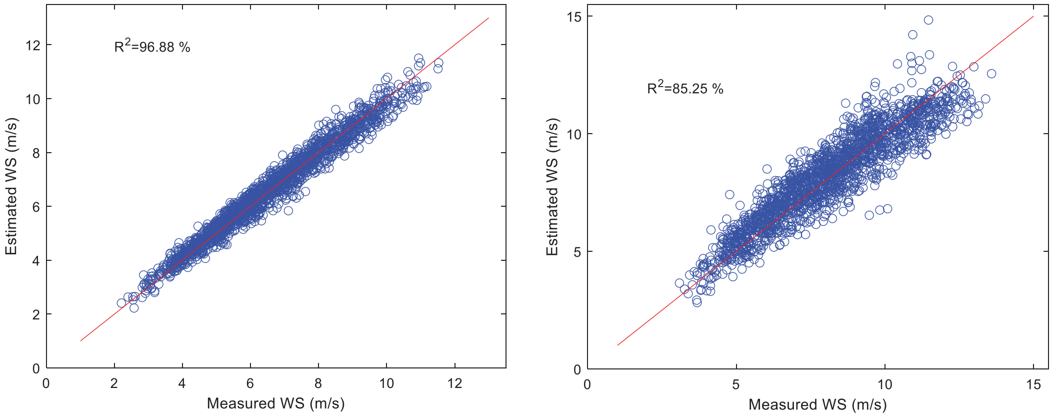


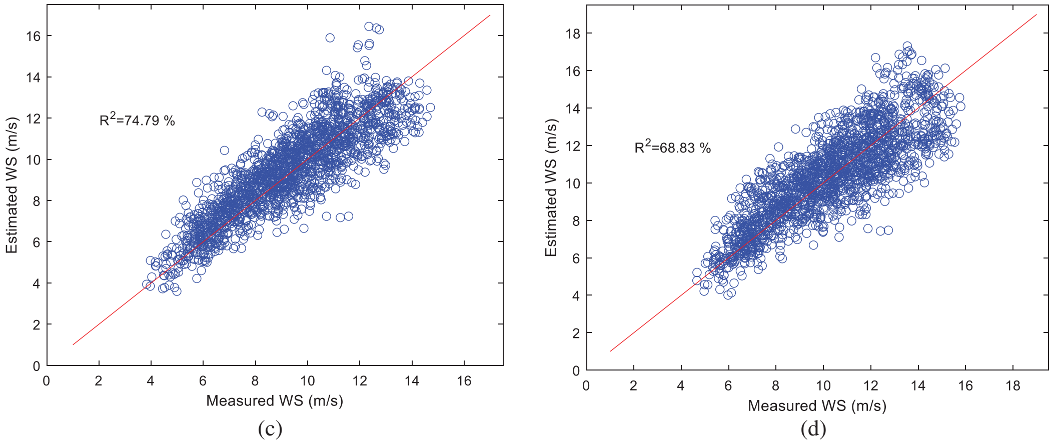
Figure 6: R^2 of different methods and all heights



(a)

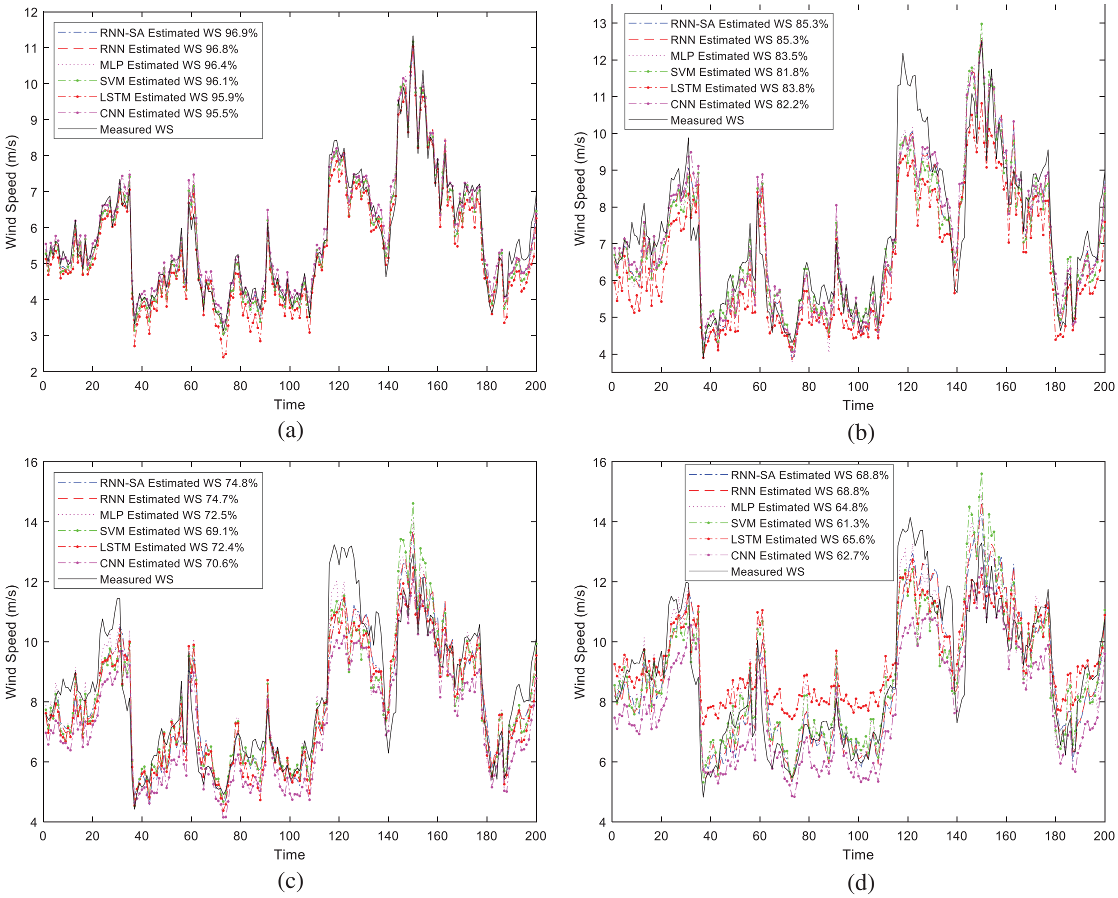
(b)

Figure 7: (Continued)



(c) (d)

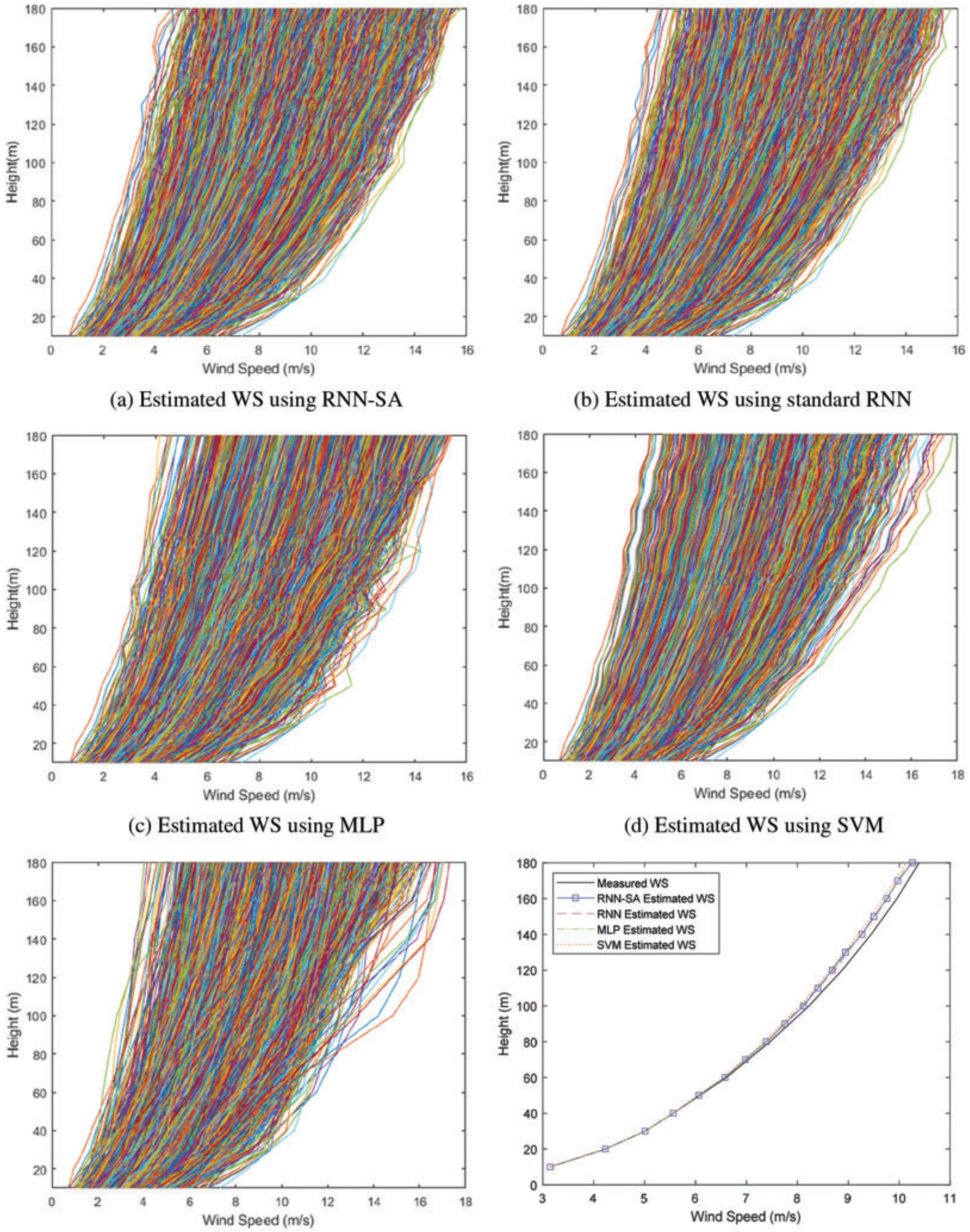
Figure 7: Scatter plots actual and estimated WS using the proposed method at 60, 100, 140, and 180 m



(a) (b) (c) (d)

Figure 8: Wind profile of measured and estimated WS at (a) 60, (b) 100, (c) 140, and (d) 180 m

The estimated WS profiles along with the actual ones up to 180 m, are compared in Fig. 9. Each line in Figs. 9a–9e represents an individual WS value at a particular time sample. It is evident from the figure that the estimated WSs at lower heights are very close to the actual values. In contrast, the estimated WS higher heights tend to deviate from the actual. Fig. 9f represents the average of WS for all time samples. The average estimated WS for all methods confirm the observation in Table 2, where the average of estimated WS is less than the actual one.



(a) Estimated WS using RNN-SA

(b) Estimated WS using standard RNN

(c) Estimated WS using MLP

(d) Estimated WS using SVM

(e) Actual WS

(f) WS profile of actual and all methods

Figure 9: Wind profile of actual and estimated for all heights

Finally, the training duration statistics is shown in Fig. 10. It can be noticed that the proposed training method takes a bit more extra training time as indicated by higher training duration. Maximum training duration of the proposed and traditional RNN approaches are 47.77 and 36.94 s, respectively. Despite the relative high difference of the maximum duration (10.8 s), the average training duration of the proposed and traditional RNN methods are 32.39 and 31.62 s, respectively. In other words, the additional step consumes a relatively small-time duration since the average training time difference is only 0.77 s. The extra training duration is compensated by the performance improvement, as shown by the data in Tables 2 and 3.

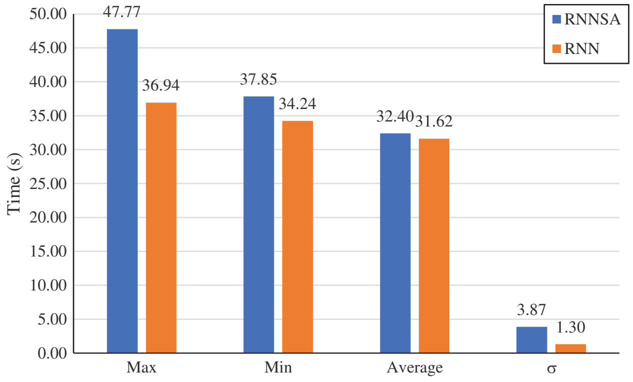


Figure 10: Training duration

4 Conclusions

This paper introduced a novel optimization method for the RNN using simulated annealing for accurate vertical WS extrapolation tasks. Each model is trained using the actual WS at lower heights (10–40 m) to estimate the WS up to 180 m. The novel method outperformed the standard RNN as well as other methods and state of the art models for all error measures on a real WS dataset. The novel method achieved the highest coefficient of determination (R^2) values of 96.88%, 82.25%, 74.79%, and 68.83% between the estimated and the actual WS values at 60, 100, 140, and 180 m, respectively. Similarly, the other error matric measures MAPE and MSE showed the lowest values of error between the measured and estimated wind speeds at all the heights. The proposed novel method is recommended for wind speed extrapolation to higher heights using the available measurements at lower heights.

Funding Statement: This research was funded by KFUPM under Grant No. DF191024.

Availability of Data and Materials: The program is written using Matlab and is available at (https://github.com/hilalnuha/RNNverticalWS).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. GWEC (2020). Global Wind Report 2021—Annual Market Update (Gwec-2021). https://gwec.net/global-wind-report-2021/.

2. Högström, U., Smedman, A. S., Bergström, H. (2006). Calculation of wind speed variation with height over the sea. *Wind Engineering, 30(4),* 269–286. DOI 10.1260/030952406779295480.
3. Newman, J. F., Klein, P. M. (2014). The impacts of atmospheric stability on the accuracy of wind speed extrapolation methods. *Resources, 3(1),* 81–105. DOI 10.3390/resources3010081.
4. Banuelos-Ruedas, F., Angeles-Camacho, C., Rios-Marcuello, S. (2011). Methodologies used in the extrapolation of wind speed data at different heights and its impact on the wind energy resource assessment in a region. In: *Wind farm—technical regulations, potential estimation and siting assessment*, pp. 97–114. http://www.intechopen.com/.
5. Gupta, D., Kumar, V., Ayus, I., Vasudevan, M., Natarajan, N. (2021). Short-term prediction of wind power density using convolutional RNN-SA network. *FME Transactions, 49(3),* 653– 663. DOI 10.5937/fme2103653G.
6. Mohandes, M., Rehman, S., Nuha, H., Islam, M. S., Schulze, F. H. (2021). Accuracy of wind speed predictability with heights using long short-term memory. *FME Transactions, 49(4),* 908–918. DOI 10.5937/fme2104908M 2021.
7. Islam, M. S., Mohandes, M., Rehman, S. (2017). Vertical extrapolation of wind speed using artificial neural network hybrid system. *Neural Computing Applications, 28(8),* 2351–2361. DOI 10.1007/s00521-016-2373-x.
8. Türkan, Y. S., Aydoğmuş, H. Y., Erdal, H. (2016). The prediction of the wind speed at different heights by machine learning methods. *An International Journal of Optimization and Control: Theories & Applications, 6(2),* 179–187. DOI 10.11121/Ijocta.01.2016.00315.
9. Emeksiz, C. (2022). Multi-gen genetic programming based improved innovative model for extrapolation of wind data at high altitudes, case study: Turkey. *Computers and Electrical Engineering, 100(1),* 107966. DOI 10.1016/j.compeleceng.2022.107966.
10. Mohandes, M. A., Rehman, S. (2018). Wind speed extrapolation using machine learning methods and LiDAR measurements. *IEEE Access, 6,* 77634–77642. DOI 10.1109/ACCESS.2018.2883677.
11. Zilong, Ti, Deng, X. W., Zhang, M. (2021). Artificial neural networks based wake model for power prediction of wind farm. *Renewable Energy, 172,* 618–631. DOI 10.1016/j.renene.2021.03.030.
12. Li, R., Zhang, J., Zhao, X. (2022). Dynamic wind farm wake modeling based on a bilateral convolutional neural network and high-fidelity les data. *Energy, 258(8),* 124845. DOI 10.1016/j.energy.2022.124845.
13. Yang, S., Deng, X., Ti, Z., Yan, B., Yang, Q. (2022). Cooperative yaw control of wind farm using a double-layer machine learning framework. *Renewable Energy, 193(5),* 519–537. DOI 10.1016/j.renene.2022.04.104.
14. Mohandes, M., Rehman, S., Nuha, H., Islam, M. S., Schulze, F. H. (2021). Accuracy of wind speed predictability with heights using recurrent neural networks. *FME Transactions, 49(4),* 908–918. DOI 10.5937/fme2104908M.
15. Cao, Q., Ewing, B. T., Thompson, M. A. (2012). Forecasting wind speed with recurrent neural networks. *European Journal of Operational Research, 221(1),* 148–154. DOI 10.1016/j.ejor.2012.02.042.
16. Fletcher, R. (2000). *Practical methods of optimization*. Second Edition. The Atrium, Southern Gate, Chichester, West Sussex PO 19 8SQ, England: John Wiley & Sons, Ltd. DOI 10.1002/9781118723203.
17. Lv, J., Deng, S., Wan, Z. (2020). An efficient single-parameter scaling memoryless Broyden-Fletcher-Goldfarb-Shanno algorithm for solving large scale unconstrained optimization problems. *IEEE Access, 8,* 85664–85674. DOI 10.1109/ACCESS.2020.2992340.
18. Da, Y., Xiurun, G. (2005). An improved PSO-based ANN with simulated annealing technique. *Neurocomputing, 63,* 527–533. DOI 10.1016/j.neucom.2004.07.002.
19. MATHWORKS (2022). Gradient descent with momentum and adaptive learning rate backpropagation. https://www.mathworks.com/help/deeplearning/ref/traingdx.html.

20. Yu, C. C., Liu, B. D. (2002). A backpropagation algorithm with adaptive learning rate and momentum coefficient. *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, vol. 2, pp. 1218–1223. IEEE.
21. Wang, X., Wen, J., Zhang, Y., Wang, Y. (2014). Real estate price forecasting based on SVM optimized by PSO. *Optik, 125(3),* 1439–1443. DOI 10.1016/j.ijleo.2013.09.017.
22. Zhu, X., Liu, R., Chen, Y., Gao, X., Wang, Y. et al. (2021). Wind speed behaviors feather analysis and its utilization on wind speed prediction using 3D-CNN. *Energy, 236(5),* 121523. DOI 10.1016/j.energy.2021.121523.
23. Hu, Y. L., Chen, L. (2018). A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and differential evolution algorithm. *Energy Conversion and Management, 173(2),* 123–142. DOI 10.1016/j.enconman.2018.07.070.
24. Rehman, S., Al-Hadhrami, L. M., Alam, M. M., Meyer, J. P. (2013). Empirical correlation between hub height and local logarithmic law for different sizes of wind turbines. *Sustainable Energy Technologies and Assessments, 4(3),* 45–51. DOI 10.1016/j.seta.2013.09.003.
25. Haby, J. (2022). Wind speed increasing with heights. https://www.theweatherprediction.com/habyhints3/749/.
26. Shrestha, R., Di, L. P., Eugene, G. Y., Kang, L., Shao, Y. Z. et al. (2017). Regression model to estimate flood impact on corn yield using MODIS NDVI and USDA cropland data layer. *Journal of Integrative Agriculture, 16(2),* 398–407. DOI 10.1016/S2095-3119(16)61502-2.