Expert Systems With Applications 214 (2023) 119063

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Better wind forecasting using Evolutionary Neural Architecture search driven Green Deep Learning

Keerthi Nagasree Pujari[a,1], Srinivas Soumitri Miriyala[a,1], Prateek Mittal[b], Kishalay Mitra[a,c,*]

[a] Department of Chemical Engineering, Indian Institute of Technology Hyderabad, India
[b] Department of Computer Science, University College London, UK
[c] Adjunct Faculty, Department of Climate Change, Indian Institute of Technology Hyderabad, India

ARTICLE INFO

Keywords:
Renewable energy
Wind characteristics forecasting
Neural architecture search
Green deep learning
Effective wind farm design

ABSTRACT

Climate Change heavily impacts global cities, the downsides of which can be minimized by adopting renewables like wind energy. However, despite its advantages, the nonlinear nature of wind renders the forecasting approaches to design and control wind farms ineffective. To expand the research horizon, the current study a) analyses and performs statistical decomposition of real-world wind time-series data, b) presents the application of Long Short-Term Memory (LSTM) networks, Nonlinear Auto-Regressive (NAR) models, and Wavelet Neural Networks (WNN) as efficient models for accurate wind forecasting with a comprehensive comparison among them to justify their application and c) proposes an evolutionary multi-objective strategy for Neural Architecture Search (NAS) to minimize the computational cost associated with training and inferring the networks which form the central theme of Green Deep Learning. Balancing the trade-off between parsimony and prediction accuracy, the proposed NAS strategy could optimally design NAR, WNN, and LSTM models with a mean test accuracy of 99%. The robust methodologies discussed in this work not only accurately model the wind behavior but also provide a green & generic approach for designing Deep Neural Networks.

1. Introduction

The rapid growth of human civilization has led to a 67 % surge in energy demand across the world in the past three decades (World Energy Consumption Statistics | Enerdata, 2021). According to the Global Energy Yearbook 2021, fossil fuels account for ~81 % of the total energy consumption resulting in a 24 % increase in CO_2 emissions (World Energy Consumption Statistics | Enerdata, 2021). Numerous conferences on climate change, starting from the Earth Summit in 1992 (Grubb et al., 2019) to the 26th United Nations Climate change conference of the Parties (COP26) in Glasgow (Vogler, 2021), brought the world leaders together to address the issue of global warming and climate change by mitigation of greenhouse-gas-emissions. As a result of these efforts, the utilization of renewable energy sources has steadily increased in the last three decades, as shown in Fig. 1 (Renewables in Electricity Production | Statistics Map by Region | Enerdata, 2021).

Among various alternative sources of renewable energy generation, wind has attracted significant attention from researchers, practitioners, investors, and policymakers due to the aspects of easy availability, cleaner production, and scope for large-scale generation. As per the Global Wind Report 2021, the total cumulative installations of wind energy have reached 743 GW helping to avoid over 1.1 billion tonnes of CO_2 globally (Global Wind Report 2021 - Global Wind Energy Council, 2021). With 95 GW installations in 2020 alone (~53 % year-on-year increase), wind farms have emerged as the clean energy technology with the most decarbonization potential per MW. However, the report suggests that this rate needs to be tripled in the coming decade to stay on the path toward net carbon neutrality by 2050, calling for urgent action from policymakers to scale up wind power production at the necessary pace (Global Wind Report 2021 - Global Wind Energy Council, 2021).

Despite so much focus on wind energy, one of the biggest challenges it faces is its uncertain nature, which results in tremendous variability in energy production. Therefore, accurate prediction of its variability can be of great help to the wind-farm owners and the industries in planning and execution of better energy conversion and management systems. Conventionally, the wind is modeled by constructing a Probability Mass Function (PMF) using time-series data of wind speed and direction. This PMF is then used in applications such as wind-farm layout optimization (or micro-siting) and control (Ciri et al., 2019; Miao et al., 2018).

* Corresponding author.
E-mail addresses: ch20resch11006@iith.ac.in (K.N. Pujari), srinivas.soumitri@gmail.com (S.S. Miriyala), kishalay@che.iith.ac.in (K. Mitra).

[1] Equal contribution.

https://doi.org/10.1016/j.eswa.2022.119063
Received 30 May 2022; Received in revised form 9 October 2022; Accepted 14 October 2022
Available online 22 October 2022

Nomenclature

$\widetilde{C}_i^{m,p}$	Intermittent Cell value of i^{th} node in m^{th} hidden layer at time step p
$\widehat{X}^t$	Estimated data at time step t
b_i^m	bias of i^{th} node in m^{th} layer
B_{LB}^T, B_{UB}^T	Lower and upper bounds onB^T
B^T	Length of Unrolled Network
$C_i^{m,p}$	Cell state in i^{th} node in m^{th} hidden layer at time step p
$F_i^{m,p}$	Forget gate of i^{th} node in m^{th} hidden layer at time step p
$I_i^{m,p}$	Input gate of i^{th} node in m^{th} hidden layer at time step p
M_{LB}, M_{UB}	Lower and upper bound on M
N_{LB}, N_{UB}	Lower and upper bounds on N^m
N^m	number of nodes in m^{th} layer
N_P	Number of parameters in the model
$O_i^{m,p}$	Output gate of i^{th} node in m^{th} hidden layer at time step p
P_{curve}	power curve
$\bar{T}$	Number of Test data points
T^F	Forward propagation length in *t*-BPTT
T_{ij}	number of points in i^{th} direction sector and j^{th} speed bin
$u_{effective}$	effective velocity at a given turbine obtained after application of wake
u_r	values of speed in r^{th} interval
w_{ij}^m	weight on connection from j^{th} node in $(m-1)^{th}$ layer to i^{th} node in m^{th} layer
x_i^m	activated output of i^{th} node in m^{th} hidden layer
X^t	data at time step t
z_{ij}^m	Translated and dilated variable in m^{th} hidden layer from j^{th} node to i^{th}node in a WNN
y_i^m	weighted sum of inputs
A	Activation function in LSTMs
D	Number of Direction sectors
H	Hurst exponent
K	dimension of data
L	Loss function
M	Number of layers in the network (Hidden layers + output layer)
R^2	Correlation coefficient
T	Length of data
U	Number of Speed bins

Greek Symbol:

f	functional map
θ	Parameters of Neural network
$\mathscr{F}_{ij}$	frequency in i^{th} direction sector and j^{th} speed bin
ϕ_q	values of direction in qth interval
$\mathbb{N}$	Number of Turbines
φ	activation function in NAR and Wavelets
Ψ	output after application of a wavelet transform on z

Symbol:

GW	Gigawatt
MW	Megawatt
CO_2	Carbon dioxide
sq. Km	Square kilometre

Abbreviations

ADAM	Adaptive Momentum
ADF	Augmented-Dickey-Fuller
AEP	Annual Energy Production
AIC	Akaike Information Criterion
autoML	automated Machine Learning
BDS	Brock-Dechert-Scheinkman
INLP	Integer Nonlinear Programming
LSTM	Long Short Term Memory Networks
NAR	Nonlinear Autoregressive Models
NAS	Neural Architecture Strategy
NSGA-II	Non-dominated Sorting Genetic Algorithm II
NWP	Numerical Weather Prediction
PMF	Probability Mass Function
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
STL	Seasonal and Trend decomposition using Loess
t-BPTT	truncated-Backpropagation Through Time
WFM	Wind Frequency Map
WNN	Wavelet Neural Networks

Though this method in practice is the best possible practical way to handle the variability in the wind, since the PMF is built on a limited amount of time-series data, the ability to capture long-range variability in the wind is sacrificed, making the results unrealistic. This necessitates the requirement of novel methods capable of forecasting accurately by considering the long-term variability in the data. The importance of forecasting in the wind energy domain is presented in many recent articles. Some of them are presented in Table 1.

When it comes to modeling nonlinear trends in wind characteristics, traditionally, physics-based methods (e.g., NWP) have been employed. The complexity in modeling weather conditions using first principles, lack of professional staff for collection and maintenance of crucial data to validate these models, and high computational costs make these physics-based models difficult to handle. These difficulties of physics-based models turned researchers towards data-driven techniques. Under this category, researchers are found to be inclined toward the utilization of system identification tools such as linear and Nonlinear Auto-Regressive (NAR) models, Fuzzy inference systems, and Wavelet Neural Networks (WNNs) for modeling and forecasting wind characteristics. Some of the prominent works are reported in Table 2. Apart from conventional system identification techniques, the applicability of deep learning (such as LSTMs, and Gated Recurrent Units (GRUs)) has been increasing in recent times due to its ability to handle extreme transience and nonlinearities in data such as that in the wind time-series. Table 3 reports some of the prominent works. The ease in availability of open-source software for system identification and machine learning techniques helped in the tremendous rise of their applicability. However, in open-source software, the difficulty arises with the selection of hyperparameters, e.g., topology of the network, choice of activation functions, etc., which govern the predictability of these models.

To overcome the difficulty in heuristics associated with machine learning models, many recent works reported reinforcement learning, Bayesian optimization, and single-objective optimization-based frameworks. In (Cho et al., 2020; Wu et al., 2019), Bayesian Optimization has been used to determine the hyperparameters of Deep Neural networks, and Deep reinforcement learning is used to determine the hyperparameters (Dong et al., 2021; Wu et al., 2020). A genetic algorithm was applied by (Han et al., 2020) with the accuracy and the verification time as objectives to determine the hyperparameters in a simple model with a single convolution layer and a single fully connected layer. A parametric programming theory was proposed by (Tso et al., 2020) to determine the hyperparameters. To the best of our knowledge, no work has been reported that discusses the optimal design of state-of-the-art nonlinear system identification tools as well as deep learning methods, using a

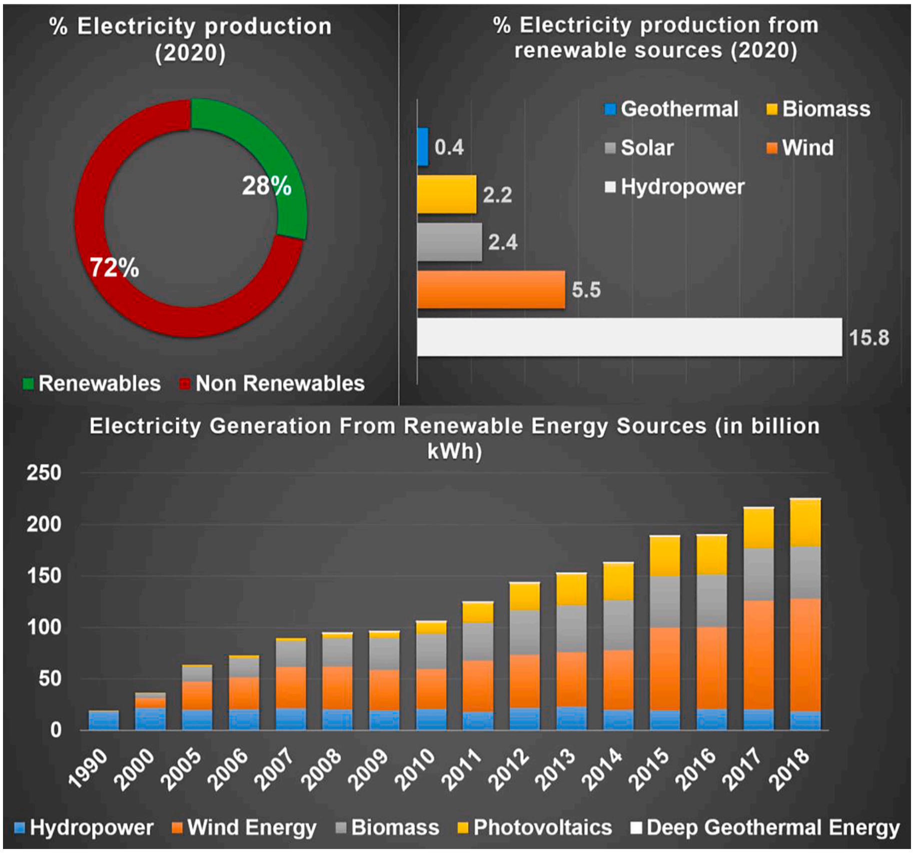


Fig. 1. Trends in utilization of renewable sources (1990–2020) (World Energy Consumption Statistics | Enerdata, 2021).

single, holistic, multi-objective evolutionary optimization-based framework balancing the aspects of over-fitting and accuracy and comparing them in terms of modeling long-term variability in wind data and predictability. The gaps and challenges in developing forecasting models for wind characteristics are as follows:

1. One of the biggest challenges of wind characteristics is its uncertain nature, which results in tremendous variability in energy production. The wind is modeled by constructing a PMF using wind speed and direction data and is used in wind farm layout optimization and control studies. Since the PMF is built on a limited amount of time-series data, the results may not be realistic.
2. Though many models were available in open source for forecasting the wind characteristics, these open-source software are associated with the heuristic selection of hyperparameters, e.g., topology of the network, choice of activation functions, etc., which govern the predictability of these models.

Based on these gaps, the aim of the current study is, therefore, to develop a method for optimally designing state-of-the-art models for forecasting the real, nonlinear, and transient wind characteristics data. To achieve this (see Fig. 2), first, the nature of data (nonlinearity, stationarity, and long-term dependency) is examined to justify the application of appropriate time-series modeling techniques. The hidden patterns and the effect of periodicities are then studied using STL decomposition. It is then proposed to use state-of-the-art modeling techniques from nonlinear system identification (NAR and WNNs) and deep learning (LSTMs). In all these techniques, the model hyperparameters (e.g., number of hidden layers, number of nodes in each hidden layer, choice of activation function, and number of unrolled time steps or the order of the model), are conventionally fixed using heuristics, thereby providing scope for inefficiencies, are estimated intelligently using optimal evolutionary search. The two conflicting attributes of data-based modeling: maximization of model accuracy and minimization of model complexity, drive the evolutionary neural architecture search strategy proposed in this work. Minimizing the model complexity reduces the computations required by the optimally designed models, thereby significantly decreasing the associated carbon footprint (Xu et al., 2021). The smaller optimally designed models have a high rate of deployment in real-world applications. The proposed methodology thus contributes to Green Deep Learning (Xu et al., 2021). After successfully training, the credibility of the forecasts from optimally designed models is validated by comparing with realistic wind characteristic data collected over four years from a wind farm in France. Additionally, a comparative study is performed among the optimal NAR, optimal WNNs, and optimal LSTMs by demonstrating the applicability of forecasts over a long range of time in the optimal design of a wind energy conversion system. The novel contributions of this work are summarized below:

1. Optimization of hyperparameters in NAR models, wavelet networks, and LSTM networks using a multi-objective optimization framework through evolutionary Neural Architecture Search (NAS) strategy.
2. Contributing to Green Deep Learning through the NAS strategy by reducing the number of parameters and thereby computations.
3. Validating the prediction scope of optimal NAR models, wavelet networks, and LSTMs with real wind characteristics data.
4. Comparison among various leading time-series modeling techniques for handling nonlinearities in wind data.
5. Method demonstrating the effective usage of past and forecasted data for accurate modeling of wind for sustainable production and management of wind energy.

In the rest of the paper, Section 2 presents time-series decomposition

Table 1
Prominent works on wind power forecasting reported in the literature.

Author/ Year	Formulation	Comments/Outcome
Allen et al., 2017	Boundary Layer Scaling method is employed, which is based on the scaling of reference climatological data from long-term average wind maps obtained from NWP models.	A mean percentage error of 1.5 % is reported with the proposed method which provided an improvement on commonly used Numerical Objective Analysis of Boundary layer wind map.
Liu et al., 2021	A Seasonal Auto-Regression Integrated Moving Average (SARIMA) model is proposed to predict hourly measured wind speed.	The SARIMA model provided the highest accuracy for short term forecasting when compared to LSTMs for the provided offshore wind speed dataset.
Liu et al., 2021	An ensemble forecasting system is proposed for short term wind speed forecasting. A novel multi-objective version of the Mayfly algorithm is used to estimate the optimal weight coefficients.	The ensemble forecasting method could achieve better performance when compared to individual forecasting methods with average MAPE values of 2 %.
Chen et al., 2022	A multi-objective error regression method is proposed which uses Convolutional Neural Networks and Long Short-term Memory Networks.	The proposed model is applied on three wind speed series data and the model outperforms with 40 % average improvement ratio compared to the state-of-the-art techniques.
Song et al., 2011	A first-order Markov chain transition matrix for wind speed time-series data using evolutionary algorithms was developed.	The paper formulates the mining process as an optimization model with constraints and develops multi-objective evolutionary strategy algorithms to solve the problem of wind farm design, wind farm control, wind speed simulations.
Wang et al., 2021	A hesitant fuzzy wind speed forecasting system with a novel defuzzification method and multi-objective optimization algorithm was proposed.	The proposed model has shown superiority in multi-step ahead forecasting with MAPE value of 10 %.
Wang et al., 2018	A deep Belief network is employed for wind power forecasting in which the Numerical weather prediction data is used as input to the proposed model.	The results are compared with Morlet Wavelet Neural network and Back propagation neural network and the proposed model outperformed by more than 40 %.
Xue et al., 2020	A novel method based on Gaussian processes is proposed to improve the probabilistic predictions of wind levels.	The results of comparison between static and dynamic Gaussian processes has shown that the dynamic Gaussian processes generates better prediction intervals.

and analysis techniques in brief, followed by a detailed description of the proposed novel algorithm for the optimal design of automated machine learning models (including NAR, WNNs, and LSTMs). Section 3 describes the results of the proposed work, followed by Section 4, which summarizes the conclusions of this work and presents the future scope.

2. Formulation

In this section, a brief description about the data and the methods of time-series analysis and decomposition are presented. The proposed holistic algorithm for the optimal design of neural networks is described. Fig. 2 illustrates the summary of the proposed methodology.

2.1. Data description and analysis

Wind characteristics data were collected from a French electricity utility company called ENGIE (La Haute Borne Data| ENGIE, 2020) over four years with a 6-hour resolution. This data was measured from four wind turbines placed in the corners of a rectangular wind farm of nine sq. Km area in La Haute Borne, France. Due to the placement of turbines only in the corners, resulting in a large inter-turbine distance, it is assumed that the measured data is not affected by wake effects. Fig. 3 gives a pictorial representation of collected wind time-series data on a limited timeframe.

2.1.1. Time-series analysis

Let the data at each time step t be denoted by $X^t = \left[X_1^t X_2^t \cdots X_K^t\right] \forall t = 1 : T$, where K is the dimensions and T is the length of the data. In this work, the available data is modeled as two univariate time-series, corresponding to wind speed and direction, respectively. Thus, in the current work, $K = 1$. A nonlinear relationship between input and output variables, along with the irregular temporal behavior, makes the nature of time-series data nonlinear, which can be detected using a hypothesis test, called the Brock-Dechert-Scheinkman (BDS) test (Akintunde et al., 2015). If the statistical properties such as mean and variance of the time-series do not depend on time, then the time-series is said to be

Table 2
Prominent works on wind power forecasting based on data driven techniques reported in the literature.

Author/Year	Formulation	Comments/Outcome
Abhinav et al., 2017	Presents a wavelet-based neural network forecast model which is robust enough to predict wind power generation in the short term with significant accuracy.	The wavelet neural network model shows an improvement of 51 % for 1 day ahead prediction over the Neural Network model.
An et al., 2011	A prediction model is constructed with a combination of wavelet transform, a weighted one rank local region method for wind farm power forecasting.	The proposed model is applied on a wind power time series data from a wind farm in China and the result has proven that wavelet-based method is more accurate.
Brahimi, 2019	The artificial neural networks (ANNs) method is proposed as a means of predicting daily wind speed in a few locations in the Kingdom of Saudi Arabia based on multiple local meteorological measurement data.	ANN has proved its efficiency in terms of accuracy and computational costs when compared to atmospheric models such as weather research forecasting method.
Daniel et al., 2020	A two-day ahead wind speed forecasting utilizing statistical and machine learning models along with their combination was presented.	The additive Quantile Regression average method was proven to be better among the other models for wind speed forecasting and the uncertainty measures of prediction intervals.
Jahangir et al., 2020	A multi-modal method is designed based on denoising and prediction modules utilizing stacked denoising auto-encoders and ANNs for short term wind speed forecasting.	In the prediction module, different ANNs have been employed in various scenarios and their results have been fully compared. The use of stacked denoising auto-encoders in the prediction section have improved the accuracy of forecasting results.
Prasetyowati et al., 2017	A hybrid model was proposed with wavelet decomposition and Nonlinear Auto-regressive Neural network model (NARX-NN) for forecasting wind power.	The proposed wavelet-NARX NN has shown better results when compared to Back propagation method with a mean error of 12 %.
Salcedo-Sanz et al., 2011	A support vector regression model using evolutionary algorithms was proposed in for short-term wind speed predictions.	The evolutionary algorithm-based support vector regression model has shown better result when compared to multi-layer perceptron models.
Zhang et al., 2022	A combination of fuzzy cluster and wind simulation model using Betz's theory was proposed for better estimation wind power.	A range of wind power generation at each time point was forecasted accurately using the proposed method.

Table 3
Prominent works on wind power forecasting based on deep learning techniques reported in the literature.

Author/Year	Formulation	Comments/Outcome
Ding et al., 2019	GRUs are employed for short-term wind power forecasting.	The proposed forecasting model outperformed the benchmark models such as ANNs and SVMs.
Li et al., 2022	A forecasting model based on variational mode decomposition and temporal convolutional networks has been proposed.	The proposed model has considerably better prediction performance than the other models such as ARIMA, LSTMs.
Kumar Dubey et al., 2021	ARIMA and SARIMA models are compared with LSTM models.	The results have shown that the LSTMs are more prominent with mean absolute error of 0.23 while forecasting the time series data.
Ningsih et al., 2019	Recurrent neural networks and LSTMs for accurate forecasting of wind speed.	The predictions of wind speed with RNN and Adam optimizer could provide 93 % accuracy when compared to LSTMs and SGD optimizer.
Olaofe, 2014	Layer recurrent neural network was employed for 5-day forecasting of wind speed and power.	The proposed model is validated with different datasets and the overall mean absolute scaled error is reported as 0.67 %.
Trebing & Mehrkanoon, 2020	A study on utilizing multidimensional convolutional neural networks for wind speed forecasting was proposed.	The results had shown that the proposed model performed best when compared to 2D and 3D convolutional neural networks.
Wang et al., 2022	An optimized decomposed and ensemble multi-feature forecasting method for wind speed is presented using Stacked Autoencoder, Variational mode decomposition, and Cuckoo search algorithm.	The proposed model was validated with a real wind data set obtained from Chinese wind farm and the proposed model was better than other models with better accuracy and stronger generalization ability.
Wu et al., 2019	A data-driven wind speed forecasting using auto-encoders and LSTM is presented.	The proposed model outperforms other benchmark models such as Random Forest, Support vector regression with at least 17 % accuracy.
Neshat et al., 2021	A novel hybrid model which uses Bidirectional LSTMs and a hierarchical decomposition technique for forecasting wind speed is adapted.	The effectiveness of the proposed approach was evaluated using real data and comparing with six other applied machine learning models.
Neshat et al., 2022	A novel Quaternion convolution neural network combined with Bidirectional LSTM is proposed for wind speed forecasting.	The proposed model achieved considerable accuracy improvements when compared with five existing machine learning and two hybrid models.

stationary. A hypothesis test, called Augmented-Dickey-Fuller (ADF) test, is used to determine stationarity in the data (Dickey & Fuller, 1979). The long-term dependency test determines the extent of dependency of data at time instance t on the previous values. The presence of long-term dependency can be determined using the Hurst exponent analysis (Kalo et al., 2019). The results of time-series analysis on considered wind data are presented in Section 3.

2.1.2. Time-series decomposition

The time-series data consists of hidden patterns that have a sequential influence on the data points. A common way of determination proposed by several authors in the literature is to decompose the time-series data into the trend, cycle, and seasonal patterns (Guignard et al., 2019). There are several ways to decompose time-series, such as classical-additive, multiplicative, X11, and STL (Hyndman & Athanasopoulos, 2018). Apart from STL, there exist several prominent methods for feature selection in time-series data, such as Empirical Model Decomposition (Jiang et al., 2020), Wavelet Decomposition (H. Liu et al., 2015), and Variable Model Decomposition (Neshat et al., 2022). However, in the current work, the emphasis is laid on developing an evolutionary Neural Architecture Search algorithm that focuses on utilizing the features obtained from any given state-of-the-art feature selection method to optimally design the models without involving the heuristics. The proposed method would work in a similar fashion irrespective of the method used for feature selection. To avoid any bias toward recently developed feature selection methods, we have utilized the standard STL decomposition (Hyndman & Athanasopoulos, 2018) to extract the features and perform NAS using them. Therefore, we consider the STL method for decomposing wind time-series data. The decomposition in the STL method is done through two loops (Cleveland et al., 1990). The outer loop assigns the robustness weights to each data point for Loess smoothing, while the inner loop performs the decomposition. The results are presented in Section 3.

2.2. Methods for modeling nonlinear time-series data.

In univariate time-series modeling, an estimate of data at time step t, $\widehat{X}^t$, is calculated as a function of B^T previous data points: $X^p|_{p=t-B^T tot-1}$ and a set of tunable parameters θ, which are optimized to minimize the error/loss (L) between the original variable, X^t and estimate, $\widehat{X}^t$, measured $\forall$t up to the sequence length T. This exercise is called training the time-series model, which is illustrated in Eq. (1) to Eq. (3).

$$\widehat{X}^t = f\left(X^p|_{p=t-B^T \text{ to } t-1} \text{ and } \theta\right) \tag{1}$$

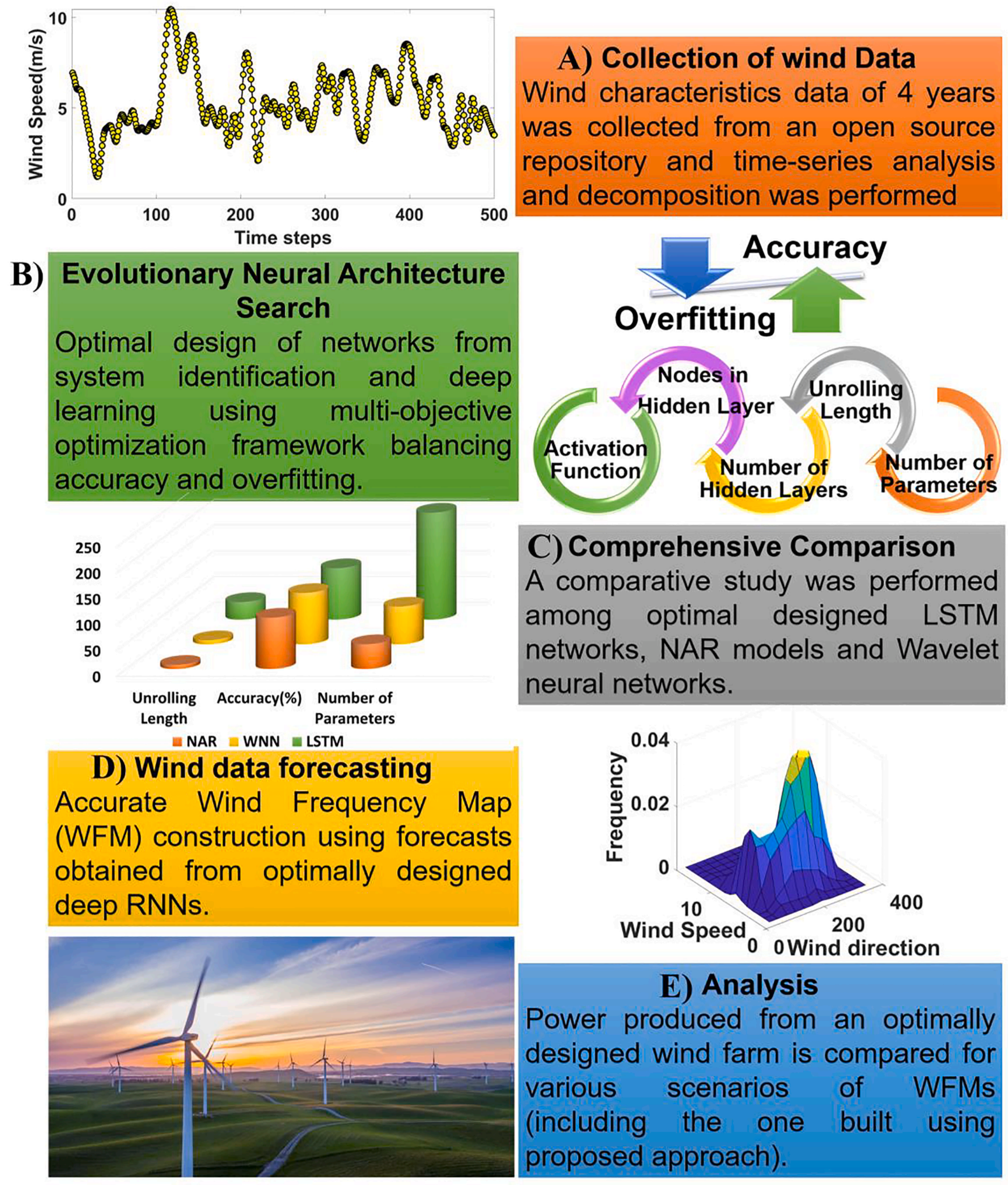
$$L = \frac{1}{T-B^T}\sum_{p=t}^{T}\left(X^p - \widehat{X}^p\right)^2 \tag{2}$$

$$\theta^* = \text{argmin}(L) \tag{3}$$

In Eq. (1), f is a functional map. In this manuscript, three nonlinear functional maps are utilized for modeling the wind time-series data. In the first case, f is represented by a neural network regressing on previous data points, thus called a nonlinear autoregressive (NAR) model (Boussaada et al., 2018; Diaconescu, 2008). The θ is the set of weights and biases in the neural network. The description of the NAR model is described briefly in Appendix A. In the second case, f is a wavelet neural network (WNN), and θ is the set of translational and dilational parameters (Alexandridis & Zapranis, 2013). The description about WNN model is described briefly in Appendix B. In the third case, f is a deep recurrent neural network called LSTM network, and θ is the set of weights and biases in the LSTM network. The description about LSTM is described briefly in Appendix C. In this work, we compare and contrast each of these methods for their abilities and disabilities to model nonlinear time-series data such as wind speed and direction. We also discuss the problems associated with each of these models and present a novel algorithm to alleviate them optimally. The idea behind the proposed algorithm is described below.

2.2.1. Hyperparameters and idea behind the novel algorithm

In this study, multi-layered (stacked) networks are used in the case of NAR, WNN, and LSTM models for modeling the wind characteristics data. Therefore, before these models are trained, certain hyperparameters which govern these models need to be fixed. These hyperparameters include the number of hidden layers in the network, the number of nodes in each hidden layer, activation function, B^T, and learning rate. Conventionally, these hyperparameters are fixed heuristically, allowing severe inaccuracies and limiting the potential of these models. In this work, while the ADAM algorithm (Kingma & Ba, 2014) ensures proper tuning of the learning rate, all other hyperparameters are estimated optimally using a novel evolutionary multi-objective algorithm. The proposed algorithm is based on a bias versus variance trade-off in machine learning: a simpler model with less number of parameters will have more bias for usage due to its simplicity and high variance in error due to its incapability.



Wind characteristics data of 4 years was collected from an open source repository and time-series analysis and decomposition was performed

Optimal design of networks from system identification and deep learning using multi-objective optimization framework balancing accuracy and overfitting.

A comparative study was performed among optimal designed LSTM networks, NAR models and Wavelet neural networks.

Accurate Wind Frequency Map (WFM) construction using forecasts obtained from optimally designed deep RNNs.

Power produced from an optimally designed wind farm is compared for various scenarios of WFMs (including the one built using proposed approach).

Fig. 2. Overall framework of wind characteristics forecasting, and its application proposed in this manuscript.

2.3. *Algorithm for the optimal design of networks*

Utilizing the trade-off between the complexity of the model (in terms of the number of parameters and order of the model B^T) and prediction accuracy (in terms of R^2 on the test set), we present a multi-objective optimization formulation with the objectives of minimizing the complexity of the network and maximizing the accuracy of the model simultaneously. All the hyperparameters: number of hidden layers, nodes, activation choice and B^Tserve as decision variables. Since the objectives are nonlinear and decision variables are integral, the proposed framework becomes a multi-objective Integer Nonlinear Programming (INLP) problem, which we solved using binary-coded Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb, 2001). The INLP formulation is presented in Eq. (4), and the algorithm is presented in Table 4.

$$\underset{\{N^m:m=1:M_{UB}\},B^T\text{and }A}{\text{minimize}} \; -R^2, N_P \text{ and } B^T \tag{4}$$

where,

$$R^2 = \left(\frac{\text{covariance}(\text{original and predicted data})}{\sqrt{\text{var}(\text{original data})\text{var}(\text{predicted data})}}\right)^2$$

$$\text{covariance} = \bar{T}\sum_{t=1}^{\bar{T}}(X^t\hat{X}^t) - \sum_{t=1}^{\bar{T}}(X^t)\sum_{t=1}^{\bar{T}}(\hat{X}^t)$$

$$\text{variance} = \bar{T}\sum_{t=1}^{\bar{T}}(X^t)^2 - \left(\sum_{t=1}^{\bar{T}}(X^t)\right)^2$$

N_P is the number of parameters in the network.
such that,

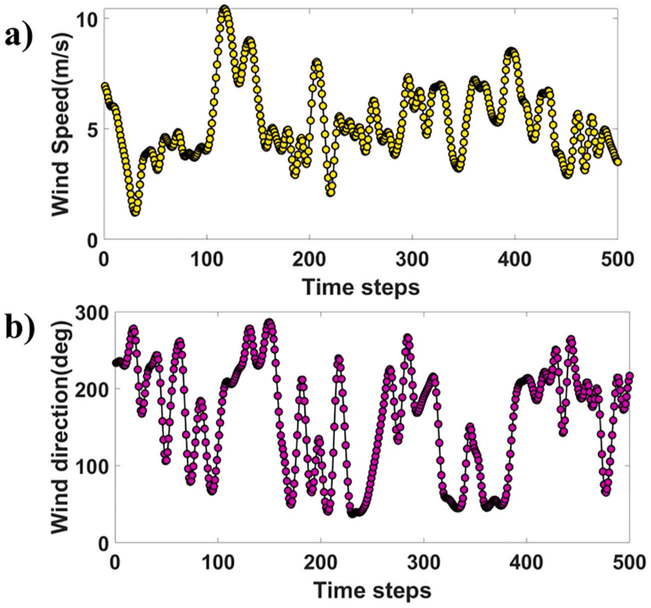


Fig. 3. Pictorial representation of wind characteristics data.

Table 4
Algorithm for optimal design of NAR, WNN and LSTM models.

Step 1	Initialize the number of binary variables as$\mathbf{M_{UB}} + 2$ and real variables as 0.
Step 2	Set the parameters of NSGA-II and start the algorithm.
Step 3	For a given population, initialize $M = 0$ and use the first M_{UB} variables to build the architecture and the last 2 decision variables to determine A and B^T: for $m \rightarrow 1$ to M_{UB} if (decision variable m is not 0) then set N^m = decision variable m $M = M + 1$. else exit the loop end if end for loop.
Step 4	Assign architecture as [1, {N^m: m = 1 to M}, 1] and obtain B^T and A from remaining two decision variables.
Step 5	Train and validate the model using backpropagation or *t*-BPTT and ADAM.
Step 6	Test the model using a test size of $\bar{T}$ and evaluate R^2.
Step 7	Evaluate the total parameters N_P in the given network.
Step 8	Increment the population counter, go to step 2 and repeat till a generation is evaluated.
Step 9	Perform the operations of NSGA-II: Crossover, Mutation, Selection and sorting and create the new generation.
Step 10	Repeat Step 3 to 9 till convergence of NSGA-II.

$$B^T_{LB} \leq B^T \leq B^T_{UB} \text{ and } N_{LB} \leq N^m \leq N_{UB} \text{ where } N_{LB} = \begin{cases} 1, & if\ m = 1 \\ 0, & if\ m > 1 \end{cases}$$

$$A \in \{1,2\} \mid \text{ if } A = \begin{cases} 1, & tansigmoid\ for\ NAR\ and\ LSTMs\ or\ Mexican\ Hat\ for\ WNNs \\ 2, & logsigmoid\ for\ NAR\ and\ LSTMs\ or\ Morlet\ for\ WNNs \end{cases}$$

$\{B^T_{LB}, B^T_{UB}, N_{UB}, M_{UB}\} \in \mathbb{Z}_+$ and

M: Number of hidden layers (determined in the proposed algorithm).

M_{LB},M_{UB}: Lower and upper bound on M (values to be defined a priori – see Table 5).

Table 5
List of parameters used in proposed algorithm for evolutionary NAS.

S. No	Parameter	Value
1	Number of binary and real variables in NSGA-II	5 and 0
2	Number of population and generations in NSGA-II	200 and 100
3	Mutation and Crossover Probability in NSGA-II	0.01 and 0.9
4	M_{LB}, M_{UB}: Lower and upper bound on number of hidden layers	1 and 3
5	N_{LB}, N_{UB}: Lower and upper bounds on nodes in each hidden layer	{1,0,0} and {16, 15, 15}
6	B^T_{LB}, B^T_{UB}: Lower and upper bounds onB^T	2 and 65
7	$\bar{T}$: Number of test data points	1200

N^m: Number of nodes in hidden layer m (determined in the proposed algorithm).

N_{LB}, N_{UB}: Lower and upper bounds on N^m (values to be defined a priori – see Table 5).

B^T: Length of the unrolled network in case of LSTMs (see Supplementary file) or the order of NAR and WNN models. (determined in the proposed algorithm and objective).

B^T_{LB}, B^T_{UB}: Lower and upper bounds on B^T (values to be defined a priori – see Table 5).

A: Choice of the activation function (determined in the proposed algorithm).

$\bar{T}$: Number of test data points (values to be defined a priori – see Table 5).

All the models and the optimizer NSGA-II have been coded in Fortran 90 language without the use of any open-source libraries. The simulations are run on Intel® Xeon CPU E5-26900 @ 2.90 GHz dual processor 128 GB RAM workstation.

3. Results and discussions

As the wind time-series data was collected from anemometers, it was first processed using a 5-point moving average approach to remove the measurement noise present in the data. The autocorrelation plot of the residual is shown in Fig. 4. The presence of data only between the 95 % confidence lines indicates that the residual is white noise (Hyndman & Athanasopoulos, 2018). To confirm that the data used for plotting the subfigures 4a and 4b is white noise, we also plotted the histograms for them, as shown in subfigures 4c and 4d. These Gaussian histograms confirm that the data is indeed white noise. We now present the results of time-series analysis followed by the decomposition of wind data and optimal design of NAR, WNN, and LSTM models.

3.1. Hypothesis tests for time-series analysis

The nature of time-series data expressed as nonlinearity, stationarity, and long-term dependencies, was examined to determine the appropriate technique for modeling the data. The characteristic of nonlinearity in the data was examined by the BDS test. The null hypothesis is H_0: The time-series data is linear, while the alternate hypothesis is H_1: The time-series data is nonlinear. The level of significance was taken as 5 %. The p-values for the wind speed and direction data using the BDS test were observed as 0.001 and 0.003, respectively, which were less than the level of significance. Therefore, the null hypothesis was rejected in

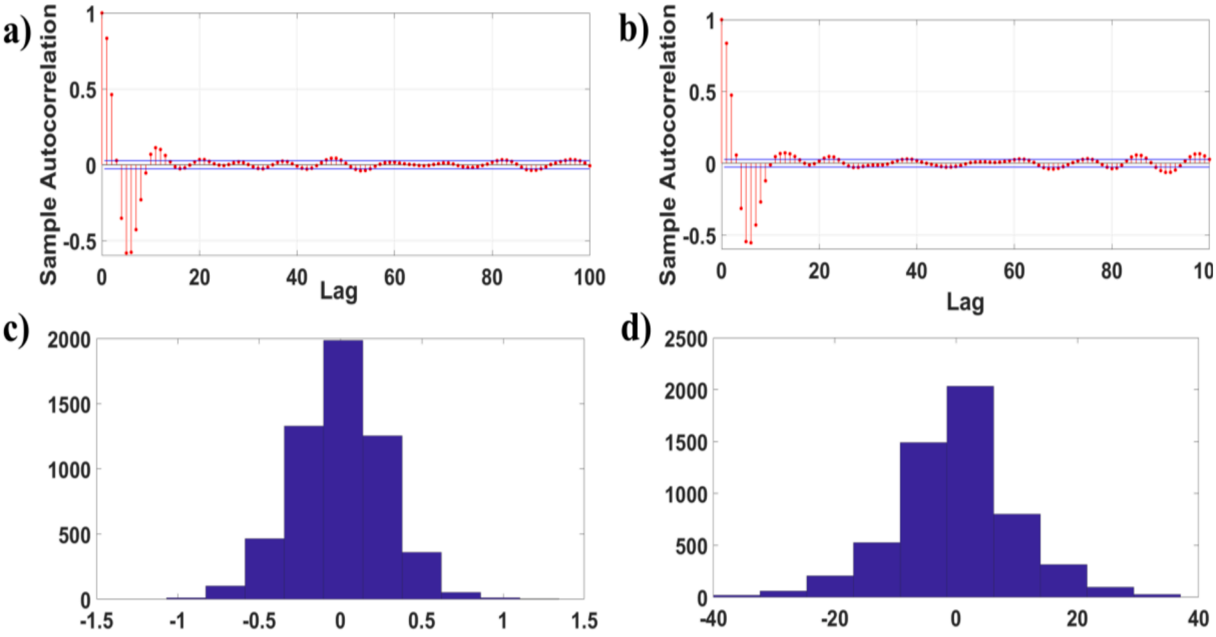


Fig. 4. Subfigures (a) and (b) represent the Autocorrelation plots and Subfigures (c) and (d) represent the histograms for residuals in wind speed and direction, respectively.

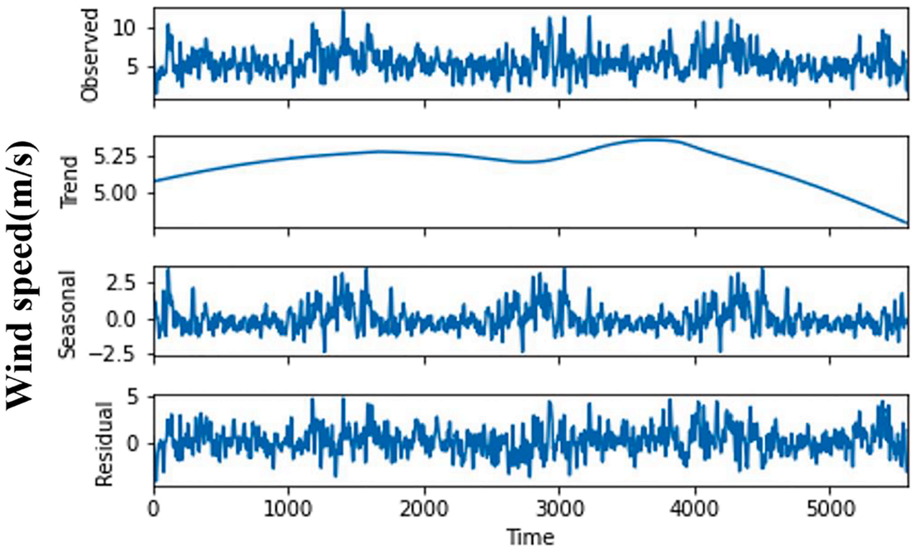


Fig. 5. STL Decomposition for Wind Speed.

both cases, and the considered wind time-series data is statistically inferred to be nonlinear.

Similarly, the characteristic of stationarity was examined using the ADF test, where the null hypothesis is H_0: The wind time-series data has a Unit root, while the alternate hypothesis is H_1: The time-series data is stationary. The p-values for wind speed and wind direction were reported as 0.008 and 0.01, respectively. Hence, the null hypothesis is rejected, and the considered data is inferred statistically to be stationary.

The long-term dependencies in the data were examined using the Hurst Exponent analysis. The Hurst exponent H was determined using the rescale range analysis on periods of observed data. The H values for wind speed and wind direction were observed as 0.83 and 0.79, respectively. As the H values are in the range of 0.5 and 1, the considered data of wind speed and direction is inferred to contain long-term dependencies.

3.2. Time-series decomposition using STL method

The time-series data has hidden patterns, which influence the sequence in the data. Therefore, it was decomposed into three components: trend, seasonal, and remainder/residual, to determine different behavioral patterns using STL decomposition. The results of STL decomposition for wind speed and direction are shown in Fig. 5 and Fig. 6, respectively. After decomposition, trend and remainder components are modeled together using NAR, WNN, and LSTM networks. Later, seasonality is combined with the forecasts as per STL decomposition for further analysis.

3.3. Optimal design of NAR, WNN, and LSTM models

Once the time-series data of four years is decomposed to extract the seasonality, it is divided into 3-years and 1-year data. The modeling

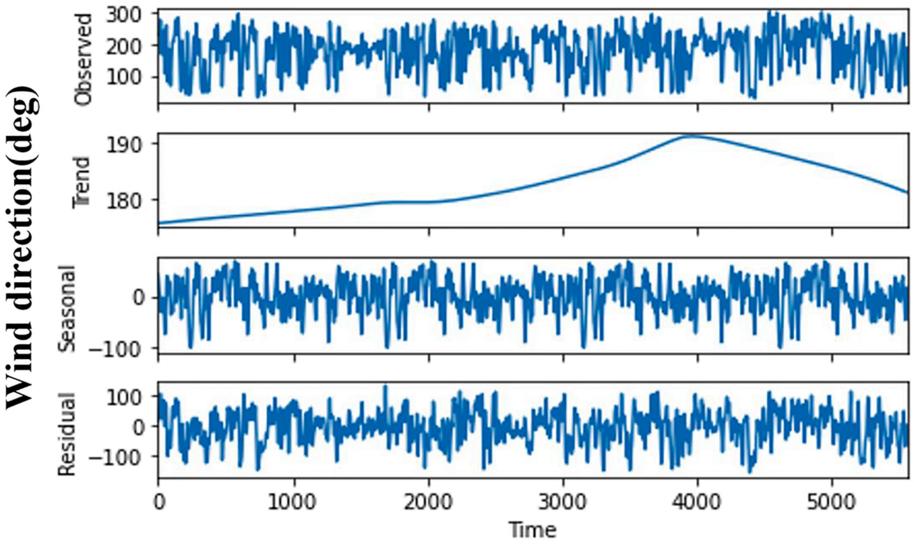


Fig. 6. STL Decomposition for Wind Direction.

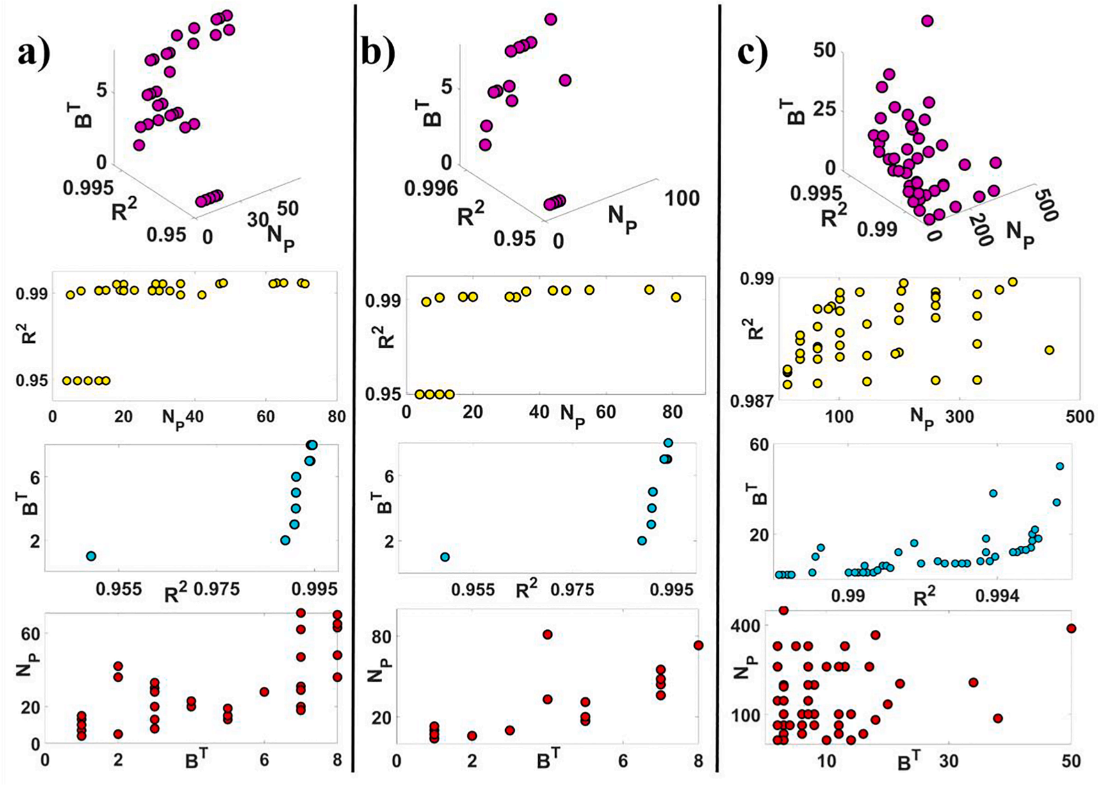


Fig. 7. Converged Pareto fronts for (a) NAR, (b) WNN and (c) LSTM for Wind Speed. Row 1 shows the 3D Pareto front while Rows 2 to 4 show the 2D projections.

(training and validation) is performed with 3-year data. The trained models are then used to forecast data for the next 1-year, which is compared with the left-out original 1–year data to prove the credibility of forecasts. From the 3-year data, 70 % is used for training the models, and the remaining 30 % is used for validating the trained models. For each of the three models (NAR, WNN, and LSTMs), two parallel simulations of the proposed evolutionary NAS algorithm are run for modeling wind characteristics, one for speed and the other for direction. All the values of bounds on the decision variables (see Eq. (4)) and settings of NSGA-II are listed in Table 5.

Three-dimensional Pareto Fronts as solutions were obtained within the first 6–8 generations of the NSGA-II. To confirm the convergence, NSGA-II was also run with different initializations and far more generations than that listed in Table 5. The obtained solutions are presented in Figs. 7 and 8 for wind speed and direction, respectively. It can be seen that there lies a trade-off between model accuracy and overfitting, as moving from one point to the other in the Pareto front improves one objective at the cost of the other. Decision variables corresponding to these points are called Pareto solutions. Each of these solutions is an embodiment of a distinct architecture of NAR, WNN, and LSTMs. The Pareto solutions are shown in Tables D.1 to D.6 in Appendix D. A single solution from the Pareto list is selected using the Akaike Information Criterion (AIC) (Akaike, 1987), a robust model selection method, as shown in Eq. (5), which penalizes the models for an increase in the

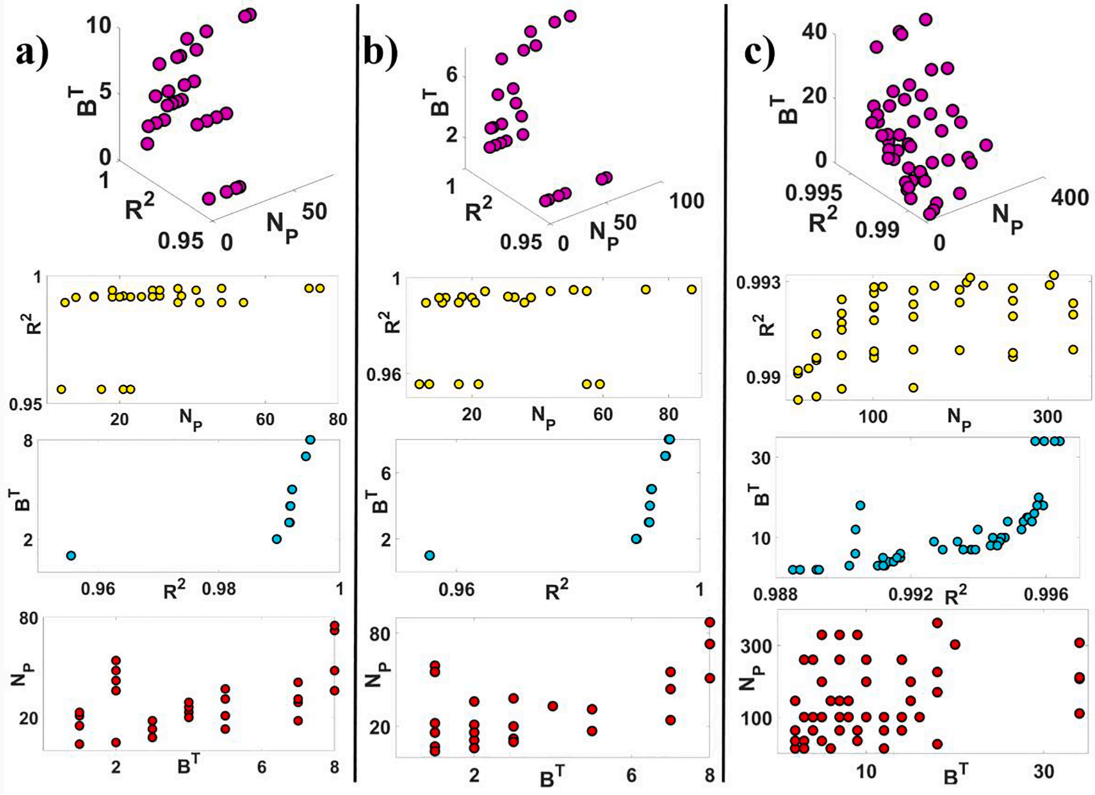


Fig. 8. Converged Pareto fronts for (a) NAR, (b) WNN and (c) LSTM for Wind Direction. Row 1 shows the 3D Pareto front while Rows 2 to 4 show the 2D projections.

Table 6
Optimal NAR, WNN and LSTM models obtained from the list of Pareto solutions by implementing the AIC criteria (See Tables D.1 to D.6 in Appendix D for AIC values of each solution).

		Speed			
Model	Architecture	Activation Function	Unrolling Length	RMSE on Validation set	R^2 on Validation set
NAR	[1-2-3-4-1]	2	8	0.0102	0.9946
WNN	[1-2-3-4-1]	1	8	0.0105	0.9943
LSTM	[1-5-2-1]	1	34	0.0677	0.9956
		Direction			
Model	Architecture	Activation Function	Unrolling Length	RMSE on Validation set	R^2 on Validation set
NAR	[1-2-3-4-1]	2	8	0.0118	0.9951
WNN	[1-2-3-4-1]	1	8	0.0119	0.9950
LSTM	[1-1-6-1]	1	34	0.0623	0.9963

number of parameters thus filtering the overfitted models.

Among all models, the one with the least AIC value is selected (Akaike, 1987). The utilization of AIC that ensures the selection of models with less complexity for deciding the final candidate from the list of Pareto solutions once again reinforces the applicability of Green Deep Learning (Xu et al., 2021) in the proposed algorithm. The optimal NAR, WNN, and LSTM models obtained for emulating wind speed and direction are shown in Table 6. Figs. 9 and 10 present the performance of these models for emulating wind speed and direction, respectively.

$$\text{AIC} = \text{Sample size for training} \times \log\left(\text{RMSE}^2\right) + 2 \times \text{Number of parameters} \tag{5}$$

3.4. Comparisons and discussions

- In the present paper, the authors have collected the real wind time-series data from an open-source wind farm. The proposed evolutionary NAS algorithm is used to model the wind characteristics data. As can be seen in Table 6, two and three hidden layered architectures have emerged as the best solutions. The proposed algorithm was able to determine the optimal architecture by evaluating only 521 architectures (maximum obtained in case of LSTM among all three varieties for wind speed) from the discrete search space of 524,288 alternatives (16^3x64x2). With the search space of 524,288 alternatives, finding the best solution through heuristics would have been extremely time-consuming and laborious, which signifies the potential and scope of the proposed NAS method. To prove the validity of the proposed model and its advantages, the authors have taken two other time-series datasets and used the proposed methodology to construct the optimal models. The details and results of the other two datasets are presented in Appendix E.
- It can be inferred from results in Table 6 that LSTMs have more parameters than NAR and WNN models; however, the accuracy in predicting the training and validation data remains similar. Further, the number of previous time steps required for modeling wind characteristics data is also higher for LSTMs when compared with the other two models. This speaks about the superiority of NAR and WNN models while emulating the training and validation data (see Figs. 9 and 10).
- LSTM networks are known for capturing long-term dependencies in the data. And for this functionality, a single LSTM block hosts 4 nodes, each of which is similar to 1 RNN node (see Appendix C). This essentially leads to an increase in parameters of the LSTMs, as seen in the results. At the same time, through Hurst exponent analysis, it was established that the data contains long-term dependencies. Thus, even though NAR and WNN prove to be simpler and more accurate, the capability of LSTMs cannot be undermined. This fact is proven

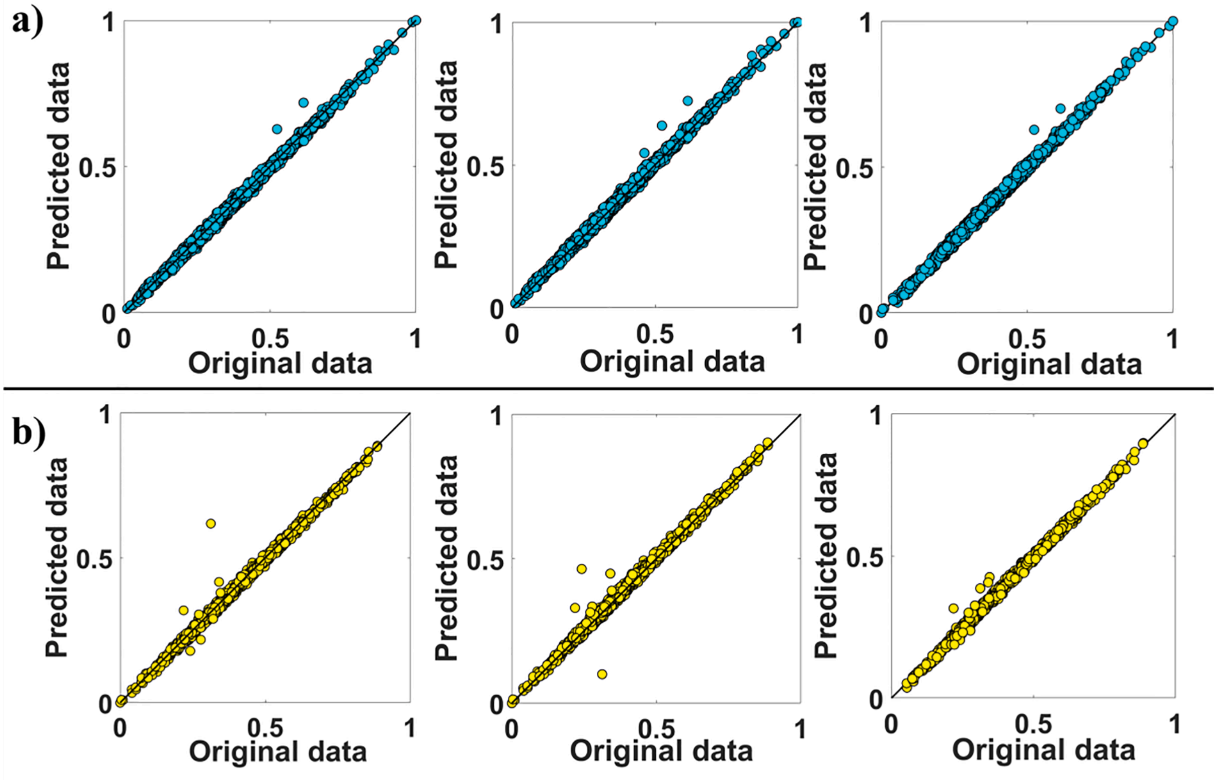


Fig. 9. Parity plots of wind speed (a) training data and b) validation data for NAR (column 1), WNN (column 2) and LSTMs (column 3).

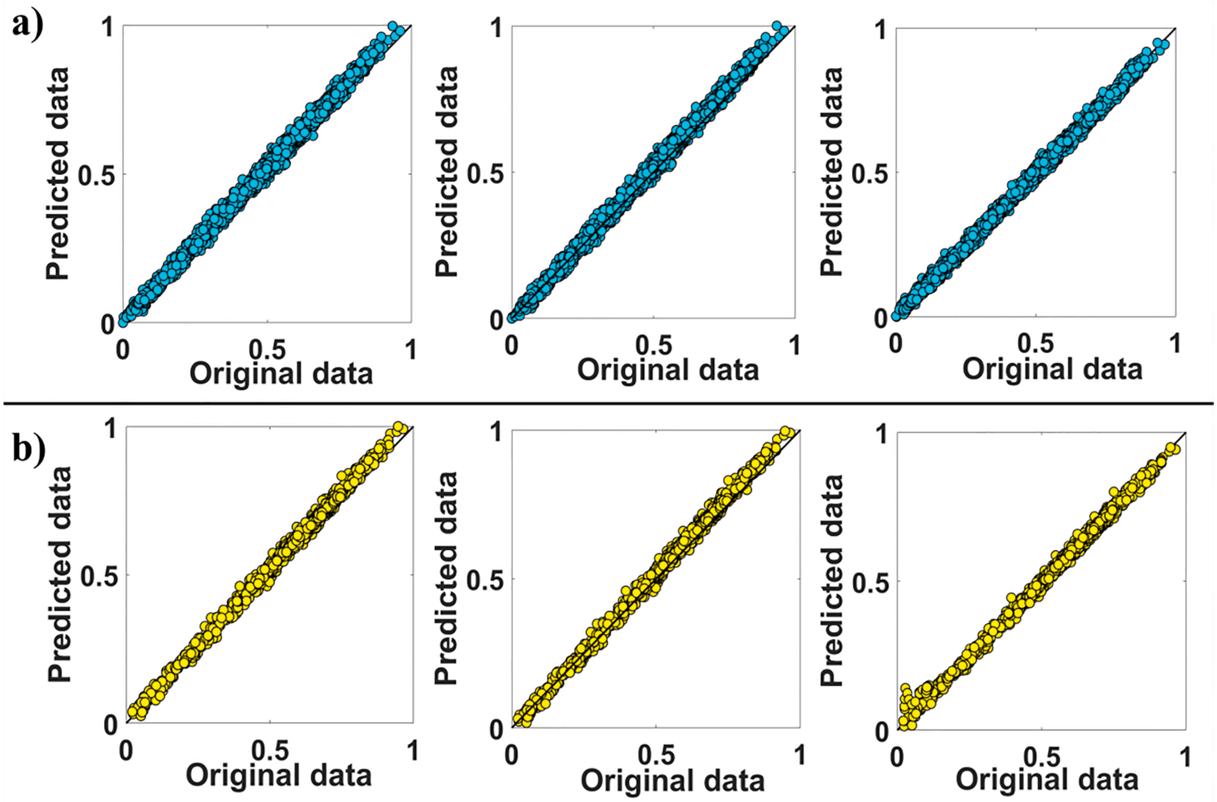


Fig. 10. Parity plots of wind direction (a) training data and b) validation data for NAR (column 1), WNN (column 2) and LSTMs (column 3).

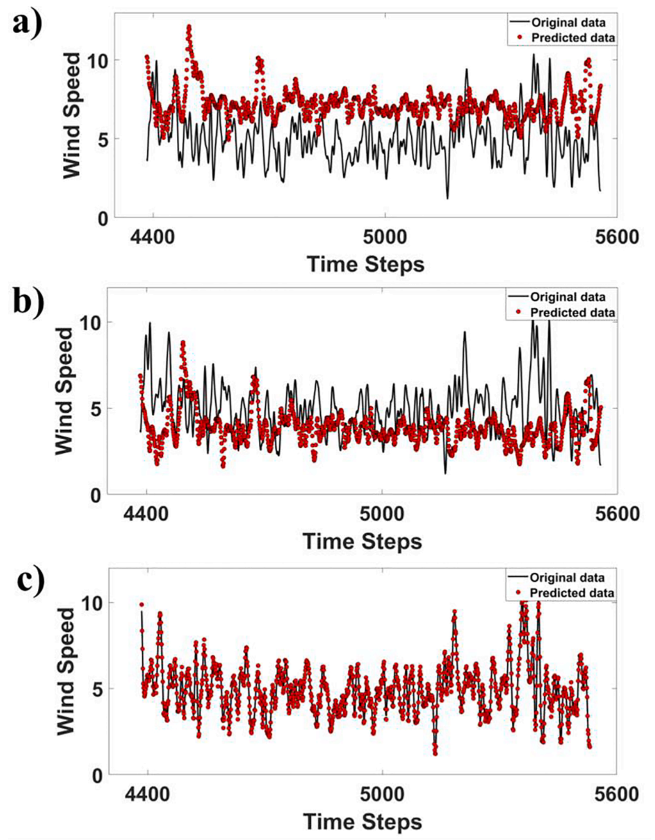


Fig. 11. Forecasting of Wind Speed using (a) NAR, (b) WNN and (c) LSTMs.

when the three models are compared in terms of forecasts (see Figs. 11 and 12).

- The failure of NAR and WNN models to forecast accurately for long-range indicates their inability to capture all dynamic features of the time-series data. On the other hand, accurate forecasts of LSTM justify the necessity of the large number of parameters in them and higher values of B^T. These inferences are also justified by the parity plots in Fig. 13. In all three models, B^T is optimally determined and fixed as constant while training. However, only in LSTMs, the extent of dependency on previous data varies to accommodate the dynamics in the data. This is made possible by the forget and input gates which regulate the extent of dependency. Unlike the LSTMs, NAR and WNN models consider the dataset as samples and do not share the parameter information across the timestamps. Hence, these models fail to learn the long-term dynamical behavior in the time-series data. These reasons might have led to the failure of NAR and WNN to forecast the time-series data over a longer range compared to LSTMs.
- However, when it comes to forecasts over the short range, NAR, WNN, and LSTM models perform similarly. Thus, for applications that demand limited forecasts, optimal NAR and WNN models should be considered rather than LSTM models. This is due to the computational load associated with LSTMs when compared with NAR and WNN models. The importance of short-range forecasts is well established in the domain of wind energy conversion systems (Boussaada et al., 2018). Therefore, we now present a unique application in the design of wind farms, which necessitates long-range forecasts and justifies the applicability of optimal LSTMs.

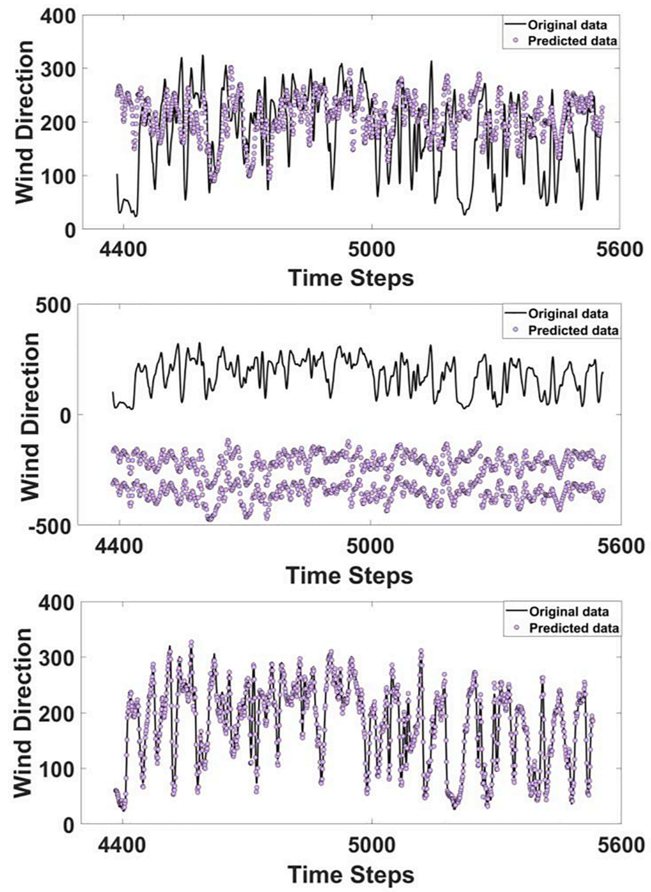


Fig. 12. Forecasting of Wind Direction using (a) NAR, (b) WNN and (c) LSTMs.

3.5. *Significance of LSTM forecasts and analysis*

Conventionally, energy is extracted from wind using the establishment of a wind farm. An optimal wind farm is where the turbines are arranged in a systematic manner such that the capital expenditure of establishment is minimized, and energy obtained from the farm is maximized while considering the wake effects. In this process, called micro-siting, to estimate the energy from a plausible layout,

a) first, a long-range wind time-series data is collected and utilized to construct a Probability Mass Function (PMF) called Wind Frequency Map (WFM),
b) then a suitable method for modeling the wake arising due to the arrangement of turbines is considered to obtain effective velocities at each turbine,
c) the power from each turbine is then evaluated as a function of the effective velocities using a relationship provided by the turbine manufacturer, called the Power curve, and
d) finally, the annual energy from the layout is obtained as a function of the expected value of power from the layout evaluated over the considered WFM, as shown in Eq. (6).

$$\text{Energy} = 8760\sum_{p=1}^{\mathbb{N}}\sum_{q=1}^{D}\sum_{r=1}^{U}\left[P_{\text{curve}}\left(u_{\text{effective}}\left(\phi_q,\ u_r,\ p\right)\right)\ x\ \text{WFM}\left(\phi_q, u_r\right)\right] \tag{6}$$

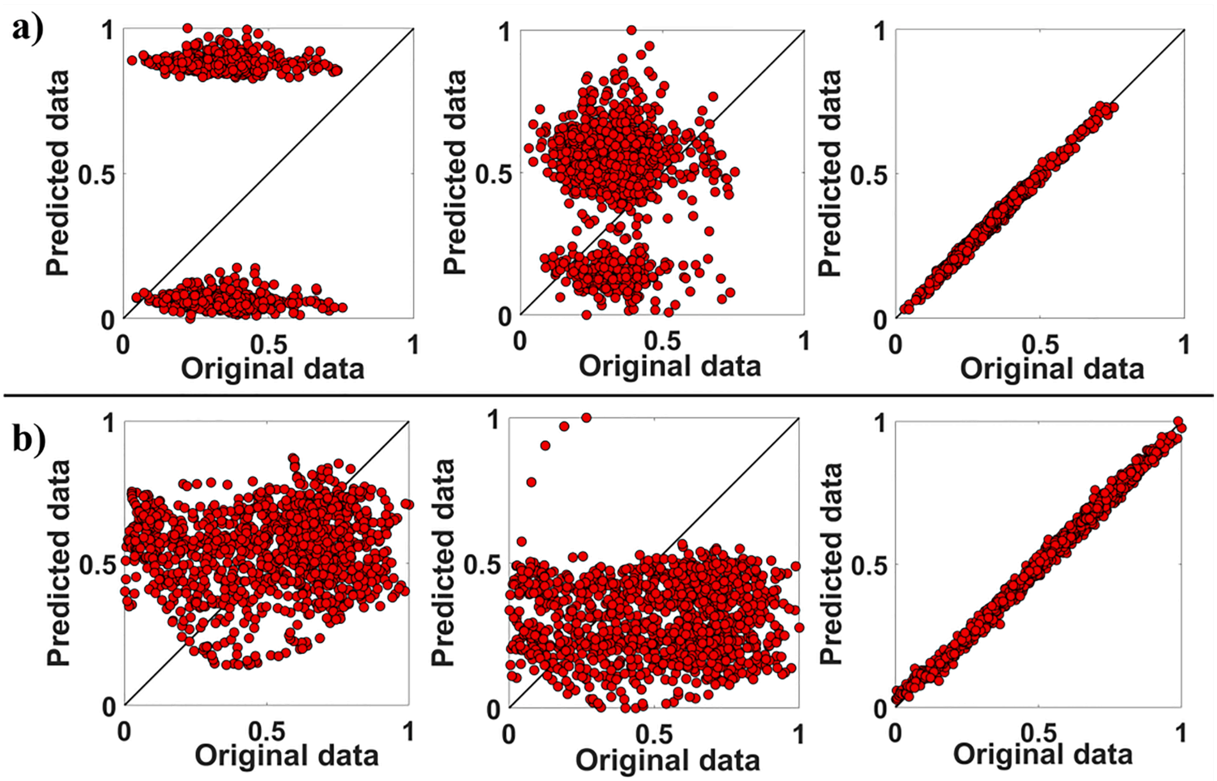


Fig. 13. Parity plots of test/forecast data (a) wind speed and (b) wind direction for NAR (column 1), WNN (column 2) and LSTMs (column 3).

In Eq. (6), ℕ is the total number of turbines, D is the number of direction sectors, U is the number of speed bins, $\phi_q$andu$_r$ are the values of direction and speed in qthandrth intervals, respectively, $u_{effective}$ is the effective velocity at a given turbine obtained after application of the wake model, P_{curve} is the power curve, the relationship provided by the turbine manufacturer to determine the rated power, and 8760 is the total number of hours in a year. Therefore, it is crucial to construct the WFM accurately (using the frequentist's approach) from the wind time-series data. However, owing to various factors such as unavailability of past wind data due to lack of measurement devices and its archival, unavailability of future wind data due to lack of efficient forecasting techniques, wind farm micro-siting is generally performed based on WFMs constructed using wind characteristics data of shorter duration. Such data depicts minimal or fixed wind characteristics making the wind farm design prone to generate unrealistic estimates of the power. Through this analysis, an effort is made to show the benefit of using accurate and more volume of wind data while determining the energy production from the wind farms. In what follows next, we first present the procedure for construction of WFM from time-series data, then consider an optimal layout of a wind farm obtained using a micro-siting study (Mittal & Mitra, 2018) and evaluate the annual energy produced from this layout using different WFMs obtained by varying the length of time-series used to build them, for comparison. The wind speed and direction are divided into disjoint intervals (direction sectors and speed bins), which are considered as random variables. The PMF on these random variables is then constructed by the process of counting or the frequentist's approach, as shown in Eq. (7).

WFM is a set of discrete probabilities

$$\mathscr{F}_{ij} = T_{ij}/T \tag{7}$$

where T is the total number of points in the wind time-series data and.

T_{ij} is the number of points in ith direction sector and jth speed bin.

In this work, we consider four different WFMs obtained in the following manner by assuming that we currently have access to the first three-year wind time-series data:

a) $WFM_{aggressive}$ – the map was constructed by using the most recent year's data.
b) $WFM_{conservative}$ – the map was constructed by using all three previous years' data.
c) $WFM_{realistic}$ – the map was constructed by using all three previous years' data and 1-year data forecasted using optimal LSTM obtained in this work,
d) $WFM_{benchmark}$ – the map was constructed by using all four years' original data. Since we have assumed that we have access to only three years' data, this map is an ideal case and is created only to show the validity of the results obtained in this study. This map serves as the benchmark for comparing the other three WFMs.

The frequency maps obtained as described above are shown in Fig. 14. To avoid any bias towards the considered WFMs, we use an optimal layout consisting of 33 turbines spread over an area of 3000 sq. Km (see Fig. 15), obtained using a micro-siting simulation as described above (Mittal & Mitra, 2018). We then evaluate the expected power from the layout as shown in Eq. (6). The values of annual energy obtained for the four different WFMs mentioned previously are listed in

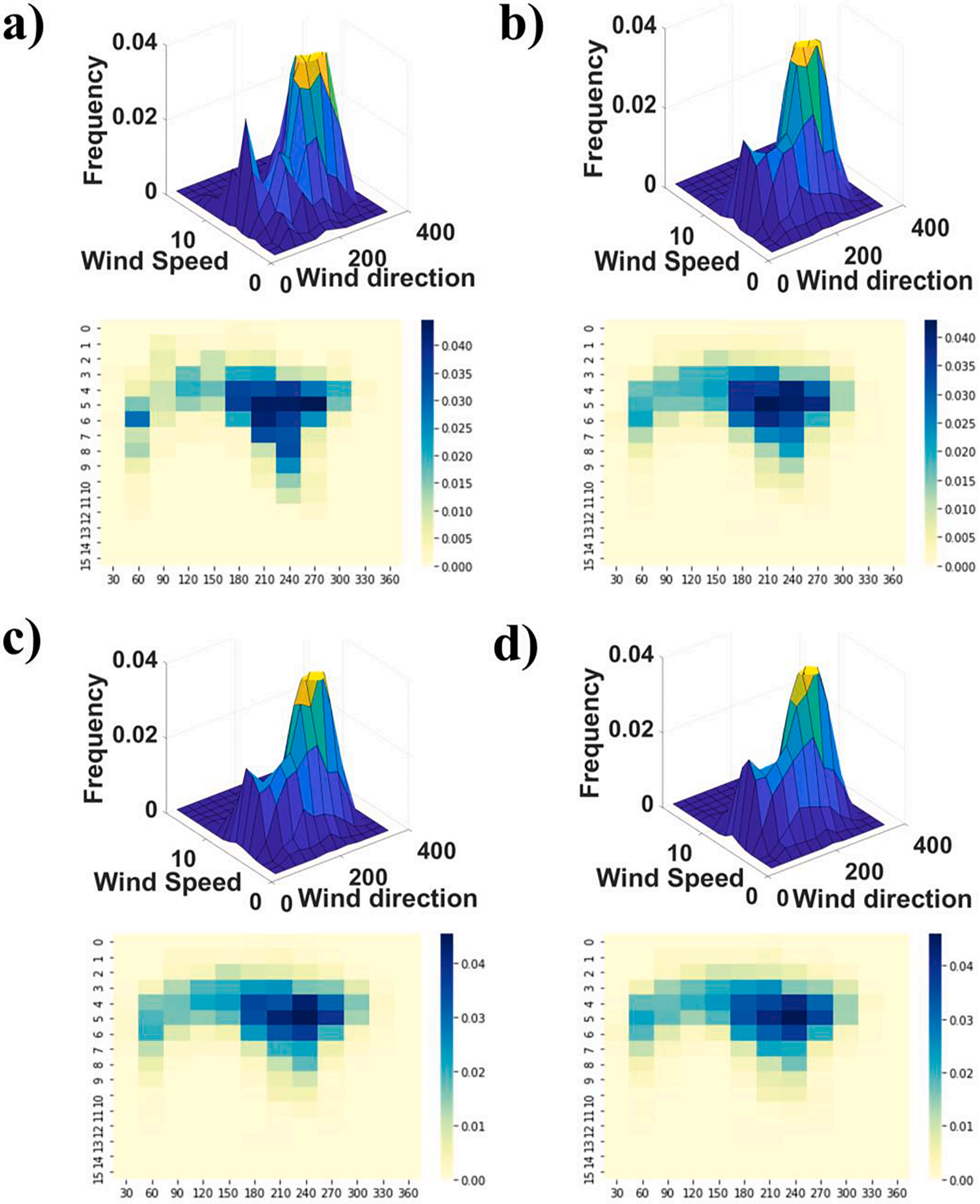


Fig. 14. Wind frequency distribution and corresponding heat maps. (a) represents $\mathrm{WFM}_{\mathrm{aggressive}}$. (b) represents $\mathrm{WFM}_{\mathrm{consevative}}$. (c) represents $\mathrm{WFM}_{\mathrm{realistic}}$. (d) represents $\mathrm{WFM}_{\mathrm{benchmark}}$.

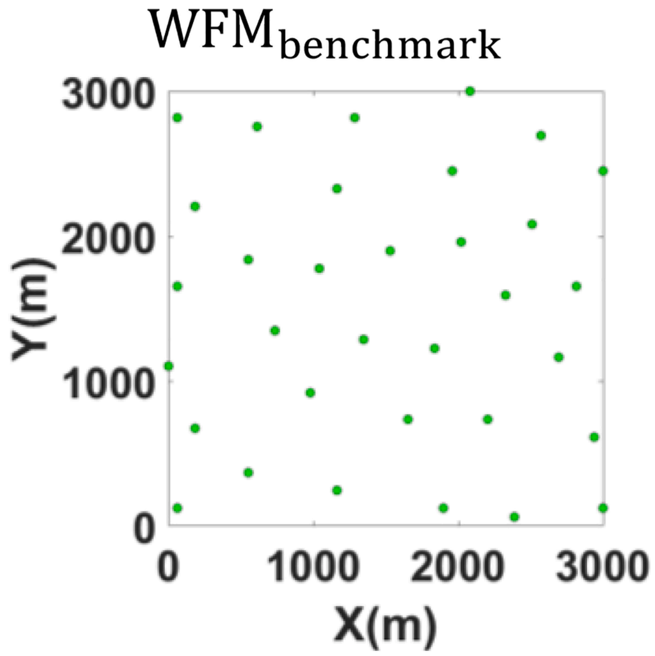


Fig. 15. Optimal Wind-farm layout used in this work for analysis using forecasts over long-range of time. The shaded circles indicate turbine locations.

Table 7
Values of Power calculated using the frequency maps obtained from given data and forecasted data.

S. No	Frequency	AEP (kW)
1	$\mathrm{WFM}_{\mathrm{aggressive}}$	14739.92
2	$\mathrm{WFM}_{\mathrm{consevative}}$	12273.48
3	$\mathrm{WFM}_{\mathrm{benchmark}}$	11548.02
4	$\mathrm{WFM}_{\mathrm{realistic}}$	11398.84

Table 7.

The analysis has revealed several interesting insights. Firstly, the energy values for the same layout are widely different with different WFMs. Therefore, if the entire micro-siting study is performed using these layouts, it would reveal entirely different optimal layouts. This would lead to the question of which of them is the correct estimate of the original power. Again the analysis conducted here provides the solution. It shows that the WFM constructed using the combination of original and forecasted data of 4 years has resulted in energy, which is in close approximation to the energy resulted from the benchmark WFM constructed using original data. This justifies the necessity of accurate long-range forecasts of wind characteristics for efficient modeling and simulation of wind energy conversion systems.

4. Conclusion

In this work, we compared three state-of-the-art models from the domains of nonlinear system identification and deep learning in terms of their abilities to model and forecast the wind characteristics time-series data. In this process, first, the wind characteristics time-series data is analyzed for nonlinearities, non-stationarity, and long-term dependencies and then decomposed to remove the seasonal component from the data. Then, the justification for selecting the NAR, WNN, and LSTM models is presented, and the problems associated with their heuristic-based design are articulated. To resolve these issues, a novel evolutionary neural architecture search strategy along the lines of automated machine learning is proposed in this study to optimally design NAR, WNN, and LSTM models. The proposed algorithm not only estimates the hyperparameters of the models but also ensures the optimal design is driven by the objective to minimize the carbon footprint involved in training and inferring large and deep neural networks. Finally, the significance of accurate forecasts over a long range of time is presented using a study of annual energy production from an optimally designed wind farm. The work is summarized as follows:

- The proposed algorithm provides the best architectures in terms of Pareto solutions, which give information about the hyperparameters of the model. The number of Pareto points from the proposed algorithm was reported as 38, 22, and 45 for NAR, WNN, and LSTM, respectively, for speed. Similarly, the Pareto points for wind direction were reported as 36, 33, and 48 for NAR, WNN, and LSTM, respectively. From the obtained solutions, one Pareto point is selected using the AIC criterion, where the point with minimum AIC is selected for further analysis to prevent overfitted models.
- The training accuracy for modeling wind speed is reported by calculating a statistical metric, R^2. The R^2 values for NAR, WNN and LSTM were reported as 0.9946, 0.9943, and 0.9956, respectively, for wind speed. The R^2 values for NAR, WNN, and LSTM were reported as 0.9951, 0.9950, and 0.9963, respectively, for wind direction. The results have shown that all three models have performed well while training.
- However, when compared to the test data over a long-range, NAR and WNN models have failed with high RMSE values and very low R^2 values. But LSTM did well on test data for both wind speed and direction with ~99 % accuracy.
- Compared to the deep learning models, the system identification techniques do not share the parameters across the timestamps leading to their failure to learn the long-term dependencies in the data. However, for applications requiring predictions for short-range, system identification techniques are more suitable due to their less complexity in terms of model parameters.
- While exploring different designs of the aforementioned models, the proposed algorithm creates a balance between overfitting and parsimony. Though it is shown in this work that the proposed algorithm is capable of designing optimal feed-forward and recurrent networks, the idea can also be used to design optimal convolutional networks to model image-based datasets. Thus, the idea is generic and contributes to the novel paradigm of research in machine learning called autoML, aimed at developing automated models without the intervention/implementation of heuristics.
- The significance of accurate forecasting is analyzed for improving the annual energy production from an optimally designed wind farm, leading to sustainable clean energy production and a world with near zero carbon footprints.

CRediT authorship contribution statement

Keerthi Nagasree Pujari: Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing - original draft. **Srinivas Soumitri Miriyala:** Conceptualization, Methodology, Supervision, Formal analysis, Investigation, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Prateek Mittal:** Conceptualization, Formal analysis, Supervision, Visualization. **Kishalay Mitra:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Project administration, Resources, Software, Visualization, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Datasets related to this article can be found at an open-source online repository, ENGIE (La Haute Borne Data| ENGIE, 2020).

Acknowledgements

The authors would like to acknowledge the support provided by the grant # SPARC/2018-2019/P1084/SL funded by the Ministry of Human Resources Development (MHRD), and grant # DST/NSM/R&D_HPC_Applications/2021/23 funded by the Department of Science and Technology, Government of India for this work.

Appendix A. Nonlinear autoregressive models (NAR)

In case of the NAR model, the sequential data is rearranged into input–output data pairs based on the value of B^T. Fig. A.1 presents a typical example of this rearrangement with $B^T = 3$ and corresponding NAR model with a neural network as the nonlinear map f. The inputs are processed across the network to generate the estimate of the target (see Eq. (A.1) to Eq. (A.3)), which results in the generation of a loss function that can be used to optimize the weights and biases in the network (see Eq. (2) and (3) in the manuscript). To model nonlinearity in the data, the network hosts a set of nonlinear transformation functions called activation functions. Log-sigmoid and Tan-sigmoid functions shown in Eq. (A.4) are two commonly used activation functions in ANNs.

- Evaluation of activated output of i^th^ node in the first hidden layer (x_i^1):

$$y_i^1 = \sum_{p=t-B^T}^{t-1}\left(w_{ij}^1 X^p\right) + b_i^1 \ \text{ Where, } \ j = p - t + B^T + 1 \ \text{ and } \ x_i^1 = \varphi\left(y_i^1\right) \tag{A1}$$

Here, y is the weighted sum of inputs $X^p|_{p=t-B^T \text{ to } t-1}$ W and b are weights and biases, respectively, and φ is the activation function. The superscript (on y, x, w, and b) indicate the layer number, the first subscript indicates the current node in the given layer, and the second subscript indicates the node in the previous layer, which is connected to the current node. For example, w_{ij}^m indicates the weight on a connection from j^th^ node in $(m-1)^{th}$ layer to i^th^ node in m^th^ layer.

- Evaluation of activated output of i^th^ node in m^th^ hidden layer (x_i^m):

$$y_i^m = \sum_{j=1}^{N^{m-1}}\left(w_{ij}^m x_j^{m-1}\right) + b_i^m \ \text{ and } \ x_i^m = \varphi\left(y_i^m\right)\forall m = 2 \text{ to } M-1 \tag{A2}$$

Here, N^{m-1} is the number of nodes in (m-1)^th^ layer, and M is total number of layers (hidden layers + output layer) in the network

- Evaluation of network output ($\widehat{X}^t$):

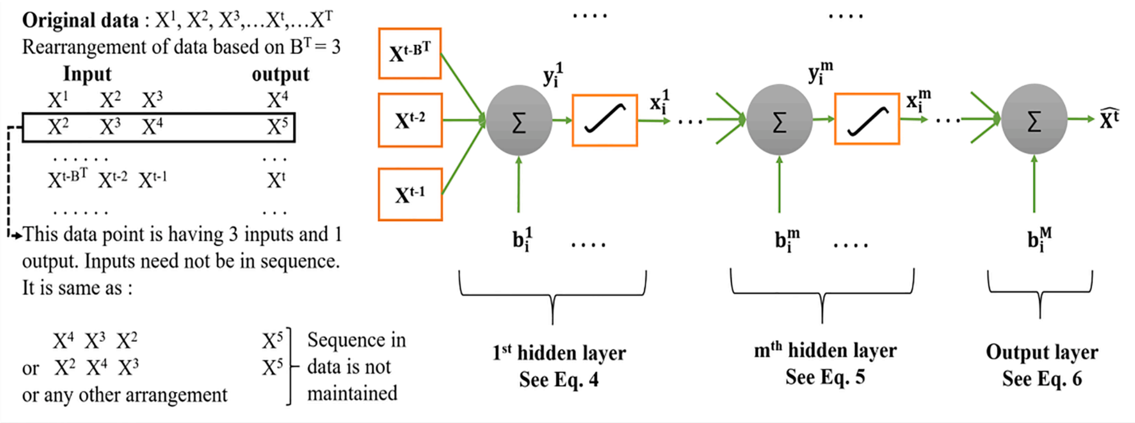

Original data : $X^1, X^2, X^3, \ldots X^t, \ldots X^T$
Rearrangement of data based on $B^T = 3$
Input
output
X^1 X^2 X^3 X^4
X^2 X^3 X^4 X^5
. . . .
X^{t-B^T} X^{t-2} X^{t-1} X^t
. . . .
This data point is having 3 inputs and 1 output. Inputs need not be in sequence. It is same as :
X^4 X^3 X^2 X^5
or X^2 X^4 X^3 X^5
or any other arrangement
Sequence in data is not maintained
X^{t-B^T}
X^{t-2}
X^{t-1}
y_i^1
x_i^1
b_i^1
y_i^m
x_i^m
b_i^m
b_i^M
Σ
$\widehat{X}^t$
1st hidden layer See Eq. 4
mth hidden layer See Eq. 5
Output layer See Eq. 6

Fig. A1. Pictorial representation of NAR model.

$$\widehat{X}^{t} = \sum_{j=1}^{N^{M-1}} \left(w^{M}_{1j} x^{M-1}_{j} \right) + b^{M}_{1} \tag{A3}$$

- Commonly used activation functions (φ):

$$\text{Log}-\text{sigmoid}: \varphi(y) = \frac{1}{1+\exp(-y)} \text{ and } \text{Tan}-\text{sigmoid}: \varphi(y) = \frac{2}{1+\exp(-2y)} - 1 \tag{A4}$$

Appendix B. Wavelet neural networks

Wavelet neural networks are similar to feed-forward neural networks, where the sigmoid activation functions are replaced with wavelet functions. In contrast to sigmoid neural networks, the wavelet networks, often considered as a generalization of Radial Basis Function (RBF) networks, are efficient in initializing the parameters such that they converge to the global minimum of the error function (Alexandridis & Zapranis, 2013). The given time-series data is rearranged in a similar way as it is done in case of NAR models. In the input layer of the wavelet network, the explanatory variables ($X^{p}|_{p=t-B^{T} \text{ to } t-1}$) are introduced. Nodes in the hidden layers, called wavelons, transform the input variables to translated and dilated versions of the mother wavelet. The translation controls the position of the mother wavelet and dilation controls the scaling parameter. The output layer approximates the estimated target value. The structure of a simple wavelet network with two hidden layers is shown in Fig. B.1, and the equations for evaluation of network output are shown in Eq. (B.1) to Eq. (B.5).

- Evaluation of activated output of ith node in the first hidden layer (x^{1}_{i}):

$$z^{1}_{ij} = \left(X^{p} - w1^{1}_{ij} \right) \Big/ \left(w2^{1}_{ij} \right) \text{ and } \Psi^{1}_{ij} = \varphi\left(z^{1}_{ij} \right)$$
$$\forall\ p = t - B^{T} \text{ to } t-1 \ \&\ j = p - t + B^{T} + 1 \tag{B1}$$

$$x^{1}_{i} = \prod_{j=t-B^{T}}^{t-1} \left(\Psi^{1}_{ij} \right) \tag{B2}$$

Here, z is the input variable translated and dilated using the weights w1 and w2, respectively, Ψ is the output after application of a wavelet transform on z, and φ is the wavelet function (see Eq. (B.6)). The superscript (on z, x, w1, w2, and Ψ) indicates the layer number, the first subscript indicates the current node in the given layer and second subscript indicates the node in the previous layer which is connected to the current node.

- Evaluation of activated output of ith node in mth hidden layer (x^{m}_{i}):

$$z^{m}_{ij} = \left(x^{m-1}_{j} - w1^{m}_{ij} \right) \Big/ \left(w2^{m}_{ij} \right) \text{ and } \Psi^{m}_{ij} = \varphi\left(z^{m}_{ij} \right)$$
$$\forall\ j = 1 \text{ to } N^{m-1} \ \&\ m = 2 \text{ to } M-1 \tag{B3}$$

$$x^{m}_{i} = \prod_{j=1}^{N^{m-1}} \left(\Psi^{m}_{ij} \right) \ \forall\ m = 2 \text{ to } M-1 \tag{B4}$$

Here, N^{m-1} is the number of nodes in (m-1)th layer, and M is total number of layers (hidden layers + output layer) in the network.

- Evaluation of network output ($\widehat{X}^{t}$):

$$\widehat{X}^{t} = \sum_{j=1}^{N^{M-1}} \left(w^{M}_{1j} x^{M-1}_{j} \right) + b^{M}_{1} \tag{B5}$$

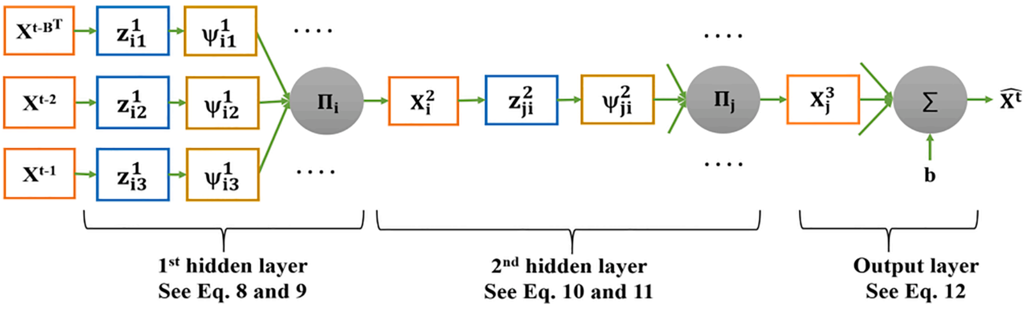


Fig. B1. Pictorial representation of WNN model.

- Commonly used wavelet functions (φ):

  Mexican hat: $\varphi(\mathrm{y}) = \frac{2}{\sqrt{3}}\pi^{\frac{-1}{4}}(1 - \mathrm{y}^2)\exp\left(\frac{-\mathrm{y}^2}{2}\right)$ and

  Morlet:

$$\varphi(\mathrm{y}) = \cos(1.75\mathrm{y})\exp\left(\frac{-\mathrm{y}^2}{2}\right) \tag{B6}$$

Appendix C. Long short-term memory networks

A primary difference between the auto-regressive models, such as NAR and WNNs, and the RNNs is that, while the former regress on previous data points, the recurrent networks regress on previous hidden states, as shown in Eq. (C.1) to (C.3). This is made possible using a feedback loop on every node in the hidden layers (see Fig. C.1). Several such hidden layers connected in a sequence between input and output layers together constitute the recurrent network. As opposed to a feed-forward network, the recurrent network has two dimensions – one goes forward in layers (input to output layer), and the other goes forward in time, as shown in Fig. C.1. This additional dimension in time is such that the same network is simulated repeatedly with feedback from previous time point and information at current time point. Therefore, to differentiate the network variables from one time step to another, an additional superscript is added, which indicates time (the other superscript indicates layers). However, since the network remains the same across all time instances, the parameters, i.e., the weights and biases, do not change with time steps. This kind of architecture maintains the sequence in the data while training the model, unlike NAR and WNN models (Alexandridis & Zapranis, 2013). Eq. (C.1) and (C.2) are valid $\forall\ \mathrm{p} = \mathrm{t} - \mathrm{B}^{\mathrm{T}}$ to $\mathrm{t} - 1$, but Eq. (C.3), which is used to generate the network output, is applicable only when $\mathrm{p} = \mathrm{t} - 1$.

- Evaluation of activated output of i^th node in the first hidden layer at time step p $\left(x_i^{1,\mathrm{p}}\right)$:

$$y_{\mathrm{i}}^{1,\mathrm{p}} = \mathrm{w}_{\mathrm{i1}}^{1}\mathrm{X}^{\mathrm{p}} + \sum_{\mathrm{k}=1}^{\mathrm{N}^{\mathrm{m}}}\left(\bar{\mathrm{w}}_{\mathrm{ik}}^{1}x_{\mathrm{k}}^{1,\mathrm{p}-1}\right) + \mathrm{b}_{\mathrm{i}}^{1} \text{ and } x_{\mathrm{i}}^{1,\mathrm{p}} = \varphi\left(y_{\mathrm{i}}^{1,\mathrm{p}}\right) \tag{C1}$$

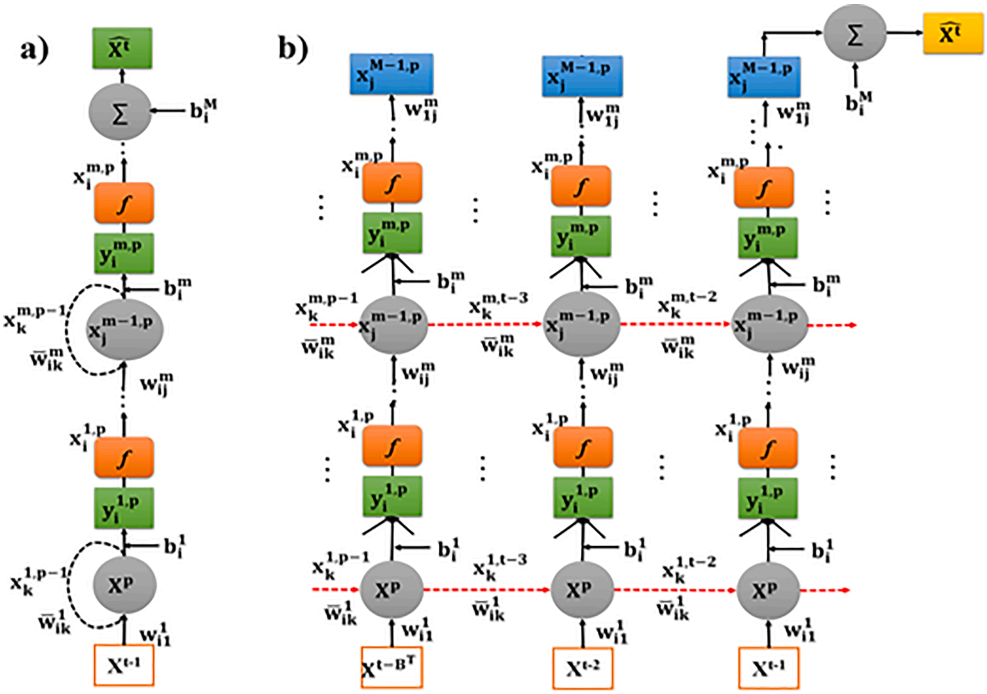


Fig. C1. Pictorial representation of (a) RNN model and (b) unrolled network.

Here, $y_i^{1,p}$ is the weighted sum of inputs at time step p, $x_i^{1,p-1}$ is the activated output of ith node in the first hidden layer at time step p-1 (x is also known as the hidden state), w, $\bar{w}$ and b are feed-forward weight, feedback weight, and bias, respectively, φ is the activation function and N^m is the number of nodes in first hidden layer.

- Evaluation of activated output of ith node in mth hidden layer at time step p ($x_i^{m,p}$):

$$y_i^{m,p} = \sum_{j=1}^{N^{m-1}}\left(w_{ij}^{m}x_j^{m-1,p}\right) + \sum_{k=1}^{N^m}\left(\bar{w}_{ik}^{m}x_k^{m,p-1}\right) + b_i^m \text{ and.}$$

$$x_i^{m,p} = \varphi(y_i^{m,p}) \ \forall \ m = 2 \quad \text{to} \quad M-1 \tag{C2}$$

- Evaluation of network output ($\widehat{X}^t$) only when p = t-1:

$$\widehat{X}^t = \sum_{j=1}^{N^{M-1}}\left(w_{1j}^{M}x_j^{M-1,p}\right) + b_1^M \tag{C3}$$

Conventionally, in recurrent networks, the output layer does not have a feedback loop. Also, similar to feed-forward networks, the activation function in the output layer is linear. Thus, Eq. (C.3) is neither having any feedback connections, nor it is having any activation function similar to Eq. (B.5) for WNNs and Eq. (A.3) for NAR models. Further, as opposed to the one-to-one style of RNN architecture, where output is evaluated at every time step, the kind of architecture style used in this manuscript (see Fig. C.1a) is called many-to-one, where output is evaluated once for every B^T inputs (see Fig. C.1b). Since, in this work, the RNNs are compared with WNN and NAR models, which consume B^T inputs to generate one output (see Figs. A.1 and B.1), many-to-one style of RNN, which also requires B^T inputs to generate one output are implemented.

In this manuscript, we use LSTM networks, which belong to the category of deep recurrent neural networks, to model the wind time-series data. This is because, when compared with simple RNNs, the LSTM networks are known to work better when long-term dependencies exist in the dataset. The fact that long-term dependencies are known to be present in natural time-series data, such as wind speed and direction, is iterated repeatedly in literature and also checked in the current work. For this reason, LSTMs are implemented in the current work instead of simple RNNs.

A node in LSTM network is compared with a node in a simple recurrent network in Fig. C.2. A primary difference is an additional output from the LSTM node, called the cell state (C), which prevents the problem of vanishing gradients. Based on the context, the LSTM network is trained to regulate the amount of previous information needed to predict the current output. To facilitate this, LSTM node has four fundamental units described below.

1. Forget gate of ith node in mth hidden layer at time step p $\left(F_i^{m,p}\right)$,

$$F_i^{m,p} = \text{logsig}\left[\sum_{j=1}^{N^{m-1}}\left(wF_{ij}^{m}x_j^{m-1,p}\right) + \sum_{k=1}^{N^m}\left(\bar{w}F_{ik}^{m}x_k^{m,p-1}\right) + bF_i^m\right] \tag{C4}$$

2. Input gate of ith node in mth hidden layer at time step p $\left(I_i^{m,p}\right)$,

$$I_i^{m,p} = \text{logsig}\left[\sum_{j=1}^{N^{m-1}}\left(wI_{ij}^{m}x_j^{m-1,p}\right) + \sum_{k=1}^{N^m}\left(\bar{w}I_{ik}^{m}x_k^{m,p-1}\right) + bI_i^m\right] \tag{C5}$$

3. Cell of ith node in mth hidden layer at time step p $\left(\widetilde{C}_i^{m,p}\right)$,

$$\widetilde{C}_i^{m,p} = \varphi\left[\sum_{j=1}^{N^{m-1}}\left(wC_{ij}^{m}x_j^{m-1,p}\right) + \sum_{k=1}^{N^m}\left(\bar{w}C_{ik}^{m}x_k^{m,p-1}\right) + bC_i^m\right] \tag{C6}$$

4. Output gate of ith node in mth hidden layer at time step p $\left(O_i^{m,p}\right)$,

$$O_i^{m,p} = \text{logsig}\left[\sum_{j=1}^{N^{m-1}}\left(wO_{ij}^{m}x_j^{m-1,p}\right) + \sum_{k=1}^{N^m}\left(\bar{w}O_{ik}^{m}x_k^{m,p-1}\right) + bO_i^m\right] \tag{C7}$$

In these equations, logsig indicates the log-sigmoid function (see Eq. (A.4)), which outputs a real value between 0 and 1; thus, the entities in Eq. (C.4), (C.5), and (C.7) are called gates with reference to the logical gates in computer science theory. In contrast with a simple RNN node, as mentioned previously, every LSTM node has two outputs – the cell state and the hidden state (which is equivalent to the output of RNN node in Eq. (C.2)). Both these outputs are evaluated using the aforementioned four fundamental units of LSTM node as shown in Eq. (C.8) and (C.9).

- Evaluation of Cell state in ith node in mth hidden layer at time step p$\left(C_i^{m,p}\right)$

$$C_i^{m,p} = F_i^{m,p}C_i^{m,p-1} + I_i^{m,p}\widetilde{C}_i^{m,p-1} \tag{C8}$$

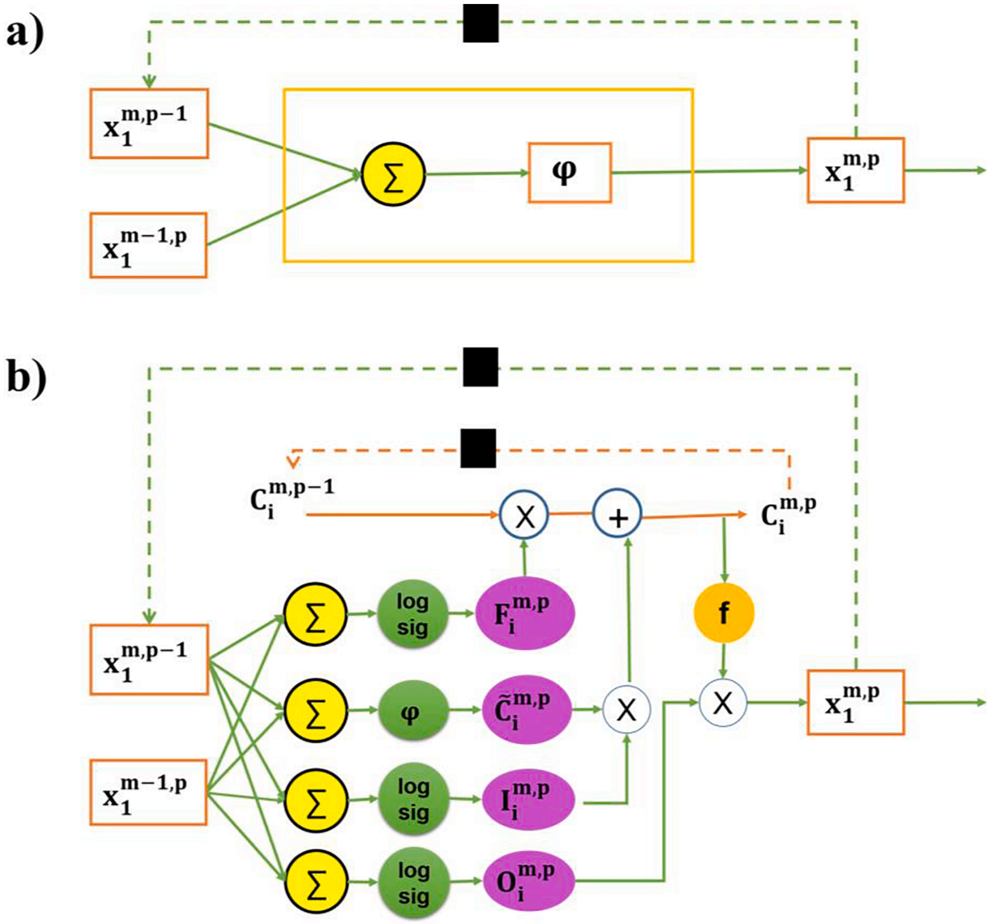


Fig. C2. Comparison between an (a) RNN node and (b) LSTM node.

- Evaluation of activated output of ith node in mth hidden layer at time step p ($x_i^{m,p}$):

$$x_i^{m,p} = O_i^{m,p}\varphi(C_i^{m,p}) \tag{C9}$$

The auto-regression equation, which has been fundamental to model the time-series, is implemented on the cell state in case of LSTMs (see Eq. (C.8)), a hidden state in case of RNNs (see Eq. (C.2)), and previous data points in case of WNNs and NAR models (see Eq. (B.3), (B.4) and (A.2)). Following are few important points which are of relevance when LSTMs are considered.

As opposed to fixed weights in auto-regression equations in RNNs, WNNS, and NAR models, the weights in auto-regression equations in LSTM are the forget and input gates. Since the values of these gates vary with every data point, they regulate the previous and current information needed to evaluate the output of LSTM node at every time point. This allows the LSTM node to have long and short-term memories based on the context of the data.

Since the LSTM node provides an auto-regressive output in terms of cell state, which is devoid of any nonlinear activation function (see Eq. (C.8)), it allows the evaluation of gradients (necessary for training) across a large length of time-series without vanishing. This allows the LSTM networks to prevent the problem of vanishing gradients.

As represented in Fig. C.2, except for the difference between the node, rest of the network remains same in case of both simple RNNs and LSTMs. In terms of equations, it means that instead of evaluating the output of node using Eq. (C.2) in RNNs, the same output is evaluated using Eq. (C.4) to Eq. (C.9) in case of LSTMs. Further, if F = 0 in Eq. (C.4), I = 1 in Eq. (C.5) and O = 0 in Eq. (C.7), cell state will be same as the hidden state in simple RNN. Thus, LSTMs enable all the functionalities of simple RNNs. The additional capabilities of LSTM networks, however, come at the cost of additional parameters (about 4 times that of RNN) in them.

Appendix D. List of Pareto solutions obtained for wind speed and wind direction

See Tables D1–D6.

Table D1
List of Pareto solutions for wind speed with NAR model.

S. No.	Number of nodes			Activation Function Choice	Unrolling Length	R^2	Number of parameters	AIC
	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**					
1	1	0	0	1	1	0.949	4	−17013
2	1	0	0	1	1	0.949	4	−17013
3	1	0	0	1	1	0.949	4	−17013
4	1	0	0	1	2	0.989	5	−21622
5	1	0	0	1	2	0.989	5	−21622
6	1	0	0	1	2	0.989	5	−21622
7	1	1	0	1	3	0.991	8	−22227
8	1	1	0	2	3	0.991	8	−26476
9	1	1	2	1	3	0.990	13	−22227
10	1	1	4	2	4	0.991	20	−26558
11	1	1	5	1	4	0.991	23	−22305
12	1	2	0	1	5	0.991	13	−22336
13	1	2	1	2	5	0.991	15	−26572
14	1	2	2	1	5	0.991	19	−22326
15	1	2	4	1	6	0.991	28	−22321
16	1	3	0	1	7	0.993	18	−23433
17	1	3	1	2	7	0.994	20	−27669
18	1	3	4	1	8	0.994	36	−23553
19	1	4	0	1	1	0.949	15	−16993
20	1	4	4	2	2	0.989	36	−25813
21	1	4	5	1	2	0.989	42	−21549
22	1	5	0	1	3	0.990	20	−22213
23	2	0	0	1	1	0.949	7	−17008
24	2	3	0	1	7	0.994	29	−23501
25	2	3	1	1	7	0.994	31	−23499
26	**2**	**3**	**4**	**2**	**8**	**0.994**	**48**	**−27979**
27	2	3	7	2	8	0.994	63	−27949
28	3	0	0	1	1	0.949	10	−17002
29	3	0	0	1	1	0.949	10	−17002
30	3	3	2	1	7	0.994	47	−23471
31	3	3	5	2	8	0.994	65	−27952
32	3	3	6	1	8	0.994	70	−23702
33	3	7	1	1	7	0.994	62	−23441
34	3	7	2	1	7	0.994	71	−23429
35	4	0	0	1	1	0.949	13	−16996
36	5	1	0	1	3	0.990	28	−22205
37	5	1	1	1	3	0.990	30	−22203
38	5	1	2	2	3	0.991	33	−26447

Table D2
List of Pareto solutions for wind speed with WNN model.

S. No.	Number of nodes			Activation Function Choice	Unrolling Length	R^2	Number of parameters	AIC
	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**					
1	1	0	0	2	1	0.949	4	−21263
2	1	0	0	2	1	0.949	4	−21263
3	1	0	0	1	2	0.989	6	−25868
4	1	0	0	1	2	0.989	6	−25868
5	1	1	0	1	3	0.990	10	−26466
6	1	2	0	1	5	0.991	17	−26567
7	1	2	1	1	5	0.991	20	−26565
8	1	7	0	1	7	0.993	36	−27419
9	1	7	1	1	7	0.993	44	−27604
10	2	0	0	1	1	0.949	7	−21259
11	2	0	0	1	1	0.949	7	−21259
12	2	0	0	1	1	0.949	7	−21259
13	2	1	4	1	4	0.991	33	−26476
14	2	2	0	1	5	0.991	31	−26555
15	2	3	1	1	7	0.993	48	−27603
16	2	3	2	1	7	0.994	55	−27705
17	**2**	**3**	**4**	**1**	**8**	**0.994**	**73**	**−27765**
18	2	5	4	1	4	0.991	81	−26383
19	3	0	0	1	1	0.949	10	−21253
20	3	0	0	2	1	0.949	10	−21253
21	4	0	0	2	1	0.949	13	−21247
22	4	0	0	2	1	0.949	13	−21247

Table D3
List of Pareto solutions for wind speed with LSTM model.

S. No.	Number of nodes			Activation Function Choice	Unrolling Length	R^2	Number of parameters	AIC
	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**					
1	1	0	0	1	2	0.988	14	−12788
2	1	0	0	2	3	0.989	14	−13537
3	1	0	0	1	10	0.989	14	−13379
4	1	0	0	1	14	0.989	14	−13558
5	1	2	2	1	38	0.993	87	−14112
6	2	0	0	2	3	0.990	35	−13804
7	2	0	0	1	6	0.990	35	−13969
8	2	0	0	1	12	0.991	35	−14142
9	2	0	0	1	16	0.991	35	−14134
10	3	0	0	1	2	0.988	64	−13234
11	3	0	0	2	3	0.990	64	−13658
12	3	0	0	1	4	0.990	64	−13891
13	3	0	0	1	6	0.991	64	−14086
14	3	0	0	1	8	0.992	64	−14594
15	3	0	0	1	12	0.993	64	−15097
16	3	1	0	1	18	0.993	82	−13824
17	3	4	0	2	3	0.990	193	−13368
18	4	0	0	2	3	0.990	101	−13566
19	4	0	0	1	6	0.991	101	−14102
20	4	0	0	2	7	0.992	101	−13382
21	4	0	0	1	8	0.993	101	−14515
22	4	0	0	1	12	0.994	101	−15309
23	4	0	0	1	14	0.995	101	−15557
24	4	1	1	1	20	0.995	134	−14233
25	5	0	0	1	2	0.988	146	−13170
26	5	0	0	2	3	0.990	146	−13575
27	5	0	0	2	7	0.992	146	−14016
28	5	1	2	1	22	0.995	203	−14873
29	**5**	**2**	**0**	**1**	**34**	**0.995**	**207**	**−16103**
30	5	5	0	1	18	0.995	366	−14976
31	6	0	0	2	3	0.990	199	−13325
32	6	0	0	2	7	0.993	199	−14104
33	6	0	0	1	8	0.993	199	−14636
34	7	0	0	1	2	0.988	260	−12941
35	7	0	0	2	7	0.993	260	−14075
36	7	0	0	1	10	0.994	260	−14501
37	7	0	0	1	12	0.994	260	−14792
38	7	0	0	2	13	0.994	260	−13994
39	7	0	0	2	17	0.994	260	−15349
40	7	3	0	1	50	0.995	388	−15703
41	7	4	0	2	3	0.990	449	−12786
42	8	0	0	1	2	0.988	329	−12808
43	8	0	0	2	5	0.991	329	−13461
44	8	0	0	2	7	0.993	329	−14162
45	8	0	0	2	13	0.994	329	−14361

Table D4
List of Pareto solutions for wind Direction with NAR model.

S. No.	Number of nodes			Activation Function Choice	Unrolling Length	R^2	Number of parameters	AIC
	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**					
1	1	0	0	2	1	0.955	4	−20452
2	1	0	0	1	1	0.955	4	−16201
3	1	0	0	1	1	0.955	4	−16201
4	1	0	0	2	1	0.955	4	−20452
5	1	0	0	1	2	0.989	5	−20603
6	1	0	0	1	2	0.989	5	−20603
7	1	0	0	2	2	0.989	5	−24854
8	1	1	0	1	3	0.991	8	−21277
9	1	1	0	2	3	0.991	8	−25526
10	1	1	4	1	4	0.991	20	−21311
11	1	1	4	2	4	0.991	20	−25559
12	1	1	5	1	4	0.991	23	−21305
13	1	2	0	2	5	0.992	13	−25685
14	1	3	0	2	7	0.994	18	−26688
15	1	3	4	2	8	0.995	36	−27062
16	1	4	0	1	1	0.955	15	−16181
17	1	4	2	1	1	0.955	23	−16165
18	1	4	4	1	2	0.989	36	−20546
19	1	4	5	1	2	0.989	42	−20535
20	1	4	6	1	2	0.989	48	−20523
21	1	4	7	1	2	0.989	54	−20511
22	2	1	0	1	3	0.991	13	−21319
23	2	1	0	2	3	0.991	13	−25568
24	2	1	2	2	3	0.991	18	−25560
25	2	1	4	2	4	0.991	26	−25581
26	2	1	5	2	4	0.991	29	−25575
27	2	2	0	1	5	0.992	21	−21443
28	2	2	3	2	5	0.992	31	−25669
29	2	3	0	2	7	0.994	29	−26677
30	2	3	1	2	7	0.994	31	−26684
31	2	3	3	2	7	0.994	41	−26665
32	**2**	**3**	**4**	**2**	**8**	**0.995**	**48**	**−27078**
33	2	4	0	2	1	0.955	21	−20419
34	2	6	0	1	5	0.992	37	−21413
35	3	3	7	2	8	0.995	75	−27037
36	4	3	4	2	8	0.995	72	−27041

Table D5
List of Pareto solutions for wind Direction with WNN model.

S. No.	Number of nodes			Activation Function Choice	Unrolling Length	R^2	Number of parameters	AIC
	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**					
1	1	0	0	1	1	0.955	4	−20452
2	1	0	0	1	1	0.955	4	−20452
3	1	0	0	1	1	0.955	4	−20452
4	1	0	0	1	2	0.989	6	−24831
5	1	0	0	1	2	0.989	6	−24831
6	1	0	0	1	2	0.989	6	−24831
7	1	1	0	1	3	0.991	10	−25523
8	1	1	1	1	3	0.991	12	−25520
9	1	2	0	1	5	0.991	17	−25641
10	1	3	0	1	7	0.994	24	−26641
11	1	3	4	1	8	0.994	51	−26920
12	2	0	0	2	1	0.955	7	−20447
13	2	0	0	2	1	0.955	7	−20447
14	2	0	0	2	2	0.989	11	−24838
15	2	0	0	2	2	0.989	11	−24838
16	2	1	1	1	3	0.991	20	−25511
17	2	1	4	1	4	0.991	33	−25533
18	2	2	0	2	5	0.992	31	−25650
19	2	3	0	1	7	0.994	44	−26656
20	2	3	2	1	7	0.994	55	−26624
21	**2**	**3**	**4**	**1**	**8**	**0.995**	**73**	**−26970**
22	2	3	6	1	8	0.995	87	−26957
23	2	5	0	1	3	0.991	38	−25502
24	3	0	0	2	2	0.989	16	−24846
25	4	0	0	2	2	0.989	21	−24834
26	4	0	0	2	2	0.989	21	−24834
27	4	4	2	1	1	0.955	59	−20344
28	5	0	0	2	1	0.955	16	−20429
29	5	4	0	1	1	0.955	55	−20351
30	7	0	0	1	1	0.955	22	−20417
31	7	0	0	1	1	0.955	22	−20417
32	7	0	0	1	2	0.989	36	−24806
33	7	0	0	1	2	0.989	36	−24806

Table D6
List of Pareto solutions for wind Direction with LSTM model.

S. No.	Number of nodes			Activation Function Choice	Unrolling Length	R^2	Number of parameters	AIC
	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**					
1	1	0	0	1	2	0.988	14	−13350
2	1	0	0	2	3	0.990	14	−14013
3	1	0	0	1	6	0.990	14	−13954
4	1	0	0	1	12	0.990	14	−13953
5	1	1	0	1	18	0.990	26	−13947
6	**1**	**6**	**0**	**1**	**34**	**0.996**	**211**	**−16606**
7	2	0	0	1	2	0.988	35	−13532
8	2	0	0	2	3	0.991	35	−14156
9	2	0	0	2	5	0.991	35	−14215
10	2	0	0	2	9	0.992	35	−14502
11	3	0	0	1	2	0.989	64	−13575
12	3	0	0	1	4	0.991	64	−14328
13	3	0	0	2	7	0.992	64	−14248
14	3	0	0	2	9	0.993	64	−14859
15	3	0	0	1	12	0.993	64	−14943
16	3	0	0	1	14	0.994	64	−14910
17	3	2	0	1	34	0.995	111	−14897
18	3	6	0	1	34	0.996	307	−16524
19	4	0	0	2	3	0.991	101	−14066
20	4	0	0	1	4	0.991	101	−14289
21	4	0	0	2	5	0.991	101	−14294
22	4	0	0	2	7	0.993	101	−14662
23	4	0	0	1	8	0.994	101	−15429
24	4	0	0	1	10	0.994	101	−15334
25	4	0	0	1	12	0.995	101	−15888
26	4	0	0	1	14	0.995	101	−16035
27	4	0	0	1	16	0.995	101	−16251
28	5	0	0	1	2	0.989	146	−13300
29	5	0	0	1	6	0.991	146	−14230
30	5	0	0	2	7	0.993	146	−14138
31	5	0	0	1	8	0.994	146	−15434
32	5	0	0	2	15	0.995	146	−15028
33	5	1	0	1	18	0.995	170	−15887
34	5	2	0	1	34	0.995	207	−16365
35	6	0	0	2	5	0.991	199	−14108
36	6	0	0	1	10	0.994	199	−15442
37	6	0	0	2	15	0.995	199	−15524
38	6	1	0	1	18	0.995	226	−16081
39	7	0	0	2	3	0.991	260	−13829
40	7	0	0	1	4	0.991	260	−14015
41	7	0	0	2	7	0.993	260	−14360
42	7	0	0	1	10	0.994	260	−15338
43	7	0	0	1	14	0.995	260	−15885
44	7	1	1	1	20	0.995	302	−15428
45	8	0	0	2	5	0.991	329	−13821
46	8	0	0	2	7	0.993	329	−14426
47	8	0	0	2	9	0.994	329	−14782
48	8	1	0	1	18	0.995	362	−15779

Appendix E. Results

In this section, the additional results are summarized. First, the wind rose figures for the considered wind data are presented in Fig. E1. The considered data is modeled using the proposed NAS algorithm, and the results with initial values of hyperparameters are presented in Table E1. The MAPE values of the forecasting models corresponding to Table 6 have been provided in Table E2.

To prove the validity of the proposed model and its advantages, the authors have taken two other datasets as case studies and used the proposed NAS methodology to model the data. The first dataset is obtained by simulating an industrial integrated grinding circuit (Mitra & Gopinath, 2004). The dataset consists of three inputs and six outputs, where the three inputs are ore feed rate and water flow rates to primary and secondary pumps, which are manipulated to control the grinding operation. Six properties that measure the quality of product from the grinding circuit are considered outputs: throughput (an indicator of the productivity of the plant), recirculation load (an indicator of energy consumed by the plant), percentage of solids, and the fraction of three size classes (coarse, mid and fine sizes) (an indicator of the quality of the product). A Multiple-Input-Multiple-Output (MIMO) dynamic system is considered for modeling with the proposed algorithm using LSTM. The second dataset is about the effect of atmospheric pollutants and weather conditions on Particular Matter dynamics. The dataset and a detailed description can be obtained from (Lee et al., 2020). A single-input single-output system is considered for modeling with the proposed algorithm using LSTM. Architectures of up to 3 hidden layers, each containing a maximum of 16 nodes, were explored in this work to model the nonlinear data from the integrated grinding circuit and Particulate Matter dynamics. A three-dimensional Pareto Front was obtained as a solution, and each of the solutions is a representation of the distinct architecture of LSTMs; and these Pareto solutions are presented below for both the datasets. The results are presented in Table E3–E4.

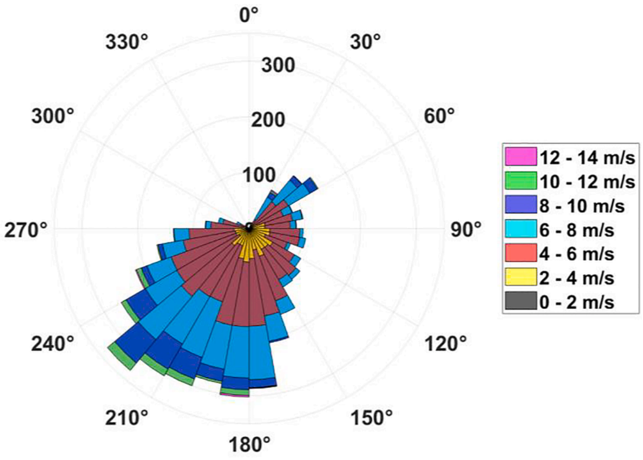


Fig. E1. Wind rose figure of wind characteristics data.

Table E1
Results of NAR, WNN and LSTM models with initial values of hyperparameters.

			Speed		
Model	Architecture	Activation Function	Unrolling Length	RMSE on Validation set	R^2 on Validation set
NAR	[1-3-5-6-1]	2	4	0.0131	0.9910
	[1-3-3-5-1]	2	8	0.0101	0.9946
	[1-1-6-7-1]	1	6	0.0259	0.9912
WNN	[1-3-5-6-1]	2	4	0.0714	0.7423
	[1-3-3-5-1]	2	8	0.053	0.8562
	[1-1-6-7-1]	1	6	0.0132	0.9907
LSTM	[1-3-5-6-1]	2	31	0.0956	0.9939
	[1-4-5-7-1]	2	33	0.0803	0.9945
	[1-8-2-7-1]	2	49	0.0973	0.9908
			Direction		
Model	**Architecture**	**Activation Function**	**Unrolling Length**	**RMSE on Validation set**	**R^2 on Validation set**
NAR	[1-3-5-6-1]	2	4	0.0156	0.9914
	[1-3-3-5-1]	2	8	0.0119	0.9949
	[1-1-6-7-1]	1	6	0.0301	0.9920
WNN	[1-3-5-6-1]	2	4	0.1758	0.0046
	[1-3-3-5-1]	2	8	0.1468	0.2692
	[1-1-6-7-1]	1	6	0.0158	0.9912
LSTM	[1-3-5-6-1]	2	31	0.0985	0.9948
	[1-4-5-7-1]	2	33	0.0741	0.9954
	[1-8-2-7-1]	2	49	0.0905	0.9936

Table E2
MAPE (mean absolute percentage error) values of forecasting models.

			Speed		
Model	Architecture	Activation Function	Unrolling Length	MAPE on Validation set	R^2 on Validation set
NAR	[1-2-3-4-1]	2	8	0.2782	0.9946
WNN	[1-2-3-4-1]	1	8	0.1695	0.9943
LSTM	[1-5-2-1]	1	34	0.0202	0.9956
			Direction		
Model	**Architecture**	**Activation Function**	**Unrolling Length**	**MAPE on Validation set**	**R^2 on Validation set**
NAR	[1-2-3-4-1]	2	8	0.2310	0.9951
WNN	[1-2-3-4-1]	1	8	0.7970	0.9950
LSTM	[1-1-6-1]	1	34	0.0242	0.9963

For both the cases, (Mitra & Gopinath, 2004; Lee et al., 2020), an architecture was selected from the list of solutions. The architecture was selected based on highest accuracy for first dataset (Mitra & Gopinath, 2004) and a higher order information called Akaike Information Criterion for second dataset (Lee et al., 2020). Among all models, the one with the least AIC value is selected which penalizes the models for an increase in the number of parameters thus filtering the overfitted models. For the selected architecture, the MAPE values are reported in Table E5.

Table E3
List of Pareto solutions for Grinding dataset using LSTM.

S. No.	Number of nodes			Activation Function Choice	Unrolling Length	R^2	Number of parameters
	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**				
1	11	0	0	1	38	0.989	732
2	6	0	0	1	40	0.77	282
3	14	0	0	1	37	0.993	1098
4	10	0	0	1	37	0.984	626
5	8	0	0	1	37	0.964	438
6	1	0	0	1	40	0.478	32
7	3	0	0	1	38	0.667	108

Table E4
List of Pareto solutions for Pollutants dataset using LSTM.

S. No.	Number of nodes			Activation Function Choice	Unrolling Length	R^2	Number of parameters
	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**				
1	2	0	0	2	3	0.992	35
2	1	0	0	1	3	0.991	14
3	1	8	0	1	33	0.995	341
4	1	2	1	2	9	0.993	62
5	3	4	0	1	17	0.994	193
6	11	13	0	1	53	0.995	1886
7	2	7	1	1	29	0.995	350
8	2	3	3	1	13	0.994	192
9	1	14	0	1	57	0.995	923
10	2	6	0	1	25	0.994	255

Table E5
MAPE (mean absolute percentage error) values of grinding and pollutants dataset.

			Grinding		
Output	Architecture	Activation Function	Unrolling Length	MAPE on Validation set	R^2 on Validation set
Output 1	[3-14-6]	1	37	0.0506	0.993
Output 2	[3-14-6]	1	37	0.1530	0.993
Output 3	[3-14-6]	1	37	0.1010	0.993
Output 4	[3-14-6]	1	37	0.1109	0.993
Output 5	[3-14-6]	1	37	0.0634	0.993
Output 6	[3-14-6]	1	37	0.1019	0.993
			Pollutants		
Output	**Architecture**	**Activation Function**	**Unrolling Length**	**MAPE on Validation set**	**R^2 on Validation set**
	[1-1-14-1]	1	57	0.2421	0.995

References

Abhinav, R., Pindoriya, N. M., Wu, J., & Long, C. (2017). Short-term wind power forecasting using wavelet-based neural network. *Energy Procedia, 142*, 455–460. https://doi.org/10.1016/J.EGYPRO.2017.12.071

Akaike, H. (1987). Factor Analysis and AIC. In: Selected papers of hirotugu akaike. In *Springer*. Springer, New York, NY. https://doi.org/10.1007/978-1-4612-1694-0_29.

Akintunde, M. O., Oyekunle, J. O., & A., O. G. (2015). Detection of non-linearity in the time series using BDS Test. *Science Journal of Applied Mathematics and Statistics*, *3*(4), 184. https://doi.org/10.11648/J.SJAMS.20150304.13.

Alexandridis, A. K., & Zapranis, A. D. (2013). Wavelet neural networks: A practical guide. *Neural Networks, 42*, 1–27. https://doi.org/10.1016/J.NEUNET.2013.01.008

Allen, D. J., Tomlin, A. S., Bale, C. S. E., Skea, A., Vosper, S., & Gallani, M. L. (2017). A boundary layer scaling technique for estimating near-surface wind energy using numerical weather prediction and wind map data. *Applied Energy, 208*, 1246–1257. https://doi.org/10.1016/J.APENERGY.2017.09.029

An, X., Jiang, D., Liu, C., & Zhao, M. (2011). Wind farm power prediction based on wavelet decomposition and chaotic time series. *Expert Systems with Applications, 38* (9), 11280–11285. https://doi.org/10.1016/J.ESWA.2011.02.176

Boussaada, Z., Curea, O., Remaci, A., Camblong, H., & Bellaaj, N. M. (2018). A nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation. *Energies 2018, Vol. 11, Page 620*, *11*(3), 620. https://doi.org/10.3390/EN11030620.

Brahimi, T. (2019). Using artificial intelligence to predict wind speed for energy application in Saudi Arabia. *Energies 2019, 12, Page 4669*, *12*(24), 4669. https://doi.org/10.3390/EN12244669.

Chen, J., Liu, H., & Chen, C. (2022). Wind speed forecasting using a novel multi-scale feature adaptive extraction ensemble model with multi-objective error regression correction. *Expert Systems with Applications*. https://doi.org/10.1016/J.ESWA.2022.117358, 117358.

Cho, H., Kim, Y., Lee, E., Choi, D., Lee, Y., & Rhee, W. (2020). Basic enhancement strategies when using Bayesian optimization for hyperparameter tuning of deep neural networks. *IEEE Access*, *8*, 52588–52608. https://doi.org/10.1109/ACCESS.2020.2981072

Ciri, U., Santoni, C., Bernardoni, F., Salvetti, M. V., & Leonardi, S. (2019). Development of a surrogate model for wind farm control. *Proceedings of the American Control Conference*, *2019-July*, 2849–2854. https://doi.org/10.23919/ACC.2019.8814766.

Cleveland, R., Cleveland, W., McRae, J., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on Loess (with discussion). *Journal of Official Statistics, 6*, 3–73.

Daniel, L. O., Sigauke, C., Chibaya, C., & Mbuvha, R. (2020). Short-term wind speed forecasting using statistical and machine learning methods. *Algorithms 2020, 13, Page 132*, *13*(6), 132. https://doi.org/10.3390/A13060132.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

Diaconescu, E. (2008). The use of NARX neural networks to predict chaotic time series. *WSEAS Transactions on Computer Research, 3*(3), 182–191.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association, 74*(366a), 427–431. https://doi.org/10.1080/01621459.1979.10482531

Ding, M., Zhou, H., Xie, H., Wu, M., Nakanishi, Y., & Yokoyama, R. (2019). A gated recurrent unit neural networks based wind speed error correction model for short-term wind power forecasting. *Neurocomputing, 365*, 54–61. https://doi.org/10.1016/J.NEUCOM.2019.07.058

Dong, X., Shen, J., Wang, W., Shao, L., Ling, H., & Porikli, F. (2021). Dynamical hyperparameter optimization via deep reinforcement learning in tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(5), 1515–1529. https://doi.org/10.1109/TPAMI.2019.2956703

Global Wind Report 2021 – Global Wind Energy Council. (2021). https://gwec.net/global-wind-report-2021/.

Grubb, M., Koch, M., Thomson, K., Sullivan, F., & Munson, A. (2019). The "Earth Summit" agreements: A guide and assessment: An analysis of the Rio '92 UN. In *Conference on Environment and Development* (p. 9). Routledge.

Guignard, F., Lovallo, M., Laib, M., Golay, J., Kanevski, M., Helbig, N., & Telesca, L. (2019). Investigating the time dynamics of wind speed in complex terrains by using the Fisher-Shannon method. *Physica A: Statistical Mechanics and Its Applications, 523*, 611–621. https://doi.org/10.1016/J.PHYSA.2019.02.048

Han, J. H., Choi, D. J., Park, S. U., & Hong, S. K. (2020). Hyperparameter optimization using a genetic algorithm considering verification time in a convolutional neural network. *Journal of Electrical Engineering & Technology 2020 15:2*, *15*(2), 721–726. https://doi.org/10.1007/S42835-020-00343-7.

Hyndman, J. R., & Athanasopoulos, G. (2018). Forecasting: principles and practice - Rob J Hyndman, George Athanasopoulos - Google Books. In *otexts*.

Jahangir, H., Golkar, M. A., Alhameli, F., Mazouz, A., Ahmadian, A., & Elkamel, A. (2020). Short-term wind speed forecasting framework based on stacked denoising auto-encoders with rough ANN. *Sustainable Energy Technologies and Assessments, 38*, Article 100601. https://doi.org/10.1016/J.SETA.2019.100601

Jiang, Y., Liu, S., Zhao, N., Xin, J., & Wu, B. (2020). Short-term wind speed prediction using time varying filter-based empirical mode decomposition and group method of data handling-based hybrid model. *Energy Conversion and Management, 220*, Article 113076. https://doi.org/10.1016/J.ENCONMAN.2020.113076

Kalo, L., Kamalanathan, P., Pant, H. J., Cassanello, M. C., & Upadhyay, R. K. (2019). Mixing and regime transition analysis of liquid-solid conical fluidized bed through RPT technique. *Chemical Engineering Science, 207*, 702–712. https://doi.org/10.1016/J.CES.2019.07.005

Kingma, D. P., & Ba, J. L. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Kumar Dubey, A., Kumar, A., García-Díaz, V., Kumar Sharma, A., & Kanhaiya, K. (2021). Study and analysis of SARIMA and LSTM in forecasting time series data. *Sustainable Energy Technologies and Assessments, 47*, Article 101474. https://doi.org/10.1016/J.SETA.2021.101474

La Haute Borne Data| ENGIE. (2020). https://opendata-renewables.engie.com/explore/.

Lee, M., Lin, L., Chen, C. Y., Tsao, Y., Yao, T. H., Fei, M. H., & Fang, S. H. (2020). Forecasting air quality in taiwan by using machine learning. *Scientific Reports, 10*(1). https://doi.org/10.1038/S41598-020-61151-7

Li, H., Jiang, Z., Shi, Z., Han, Y., Yu, C., & Mi, X. (2022). Wind-speed prediction model based on variational mode decomposition, temporal convolutional network, and sequential triplet loss. *Sustainable Energy Technologies and Assessments, 52*, Article 101980. https://doi.org/10.1016/J.SETA.2022.101980

Liu, H., Tian, H. Q., & Li, Y. F. (2015). Four wind speed multi-step forecasting models using extreme learning machines and signal decomposing algorithms. *Energy Conversion and Management, 100*, 16–22. https://doi.org/10.1016/J.ENCONMAN.2015.04.057

Liu, X., Lin, Z., & Feng, Z. (2021). Short-term offshore wind speed forecast by seasonal ARIMA - A comparison against GRU and LSTM. *Energy, 227*, Article 120492. https://doi.org/10.1016/J.ENERGY.2021.120492

Liu, Z., Jiang, P., Wang, J., & Zhang, L. (2021). Ensemble forecasting system for short-term wind speed forecasting based on optimal sub-model selection and multi-objective version of mayfly optimization algorithm. *Expert Systems with Applications, 177*, Article 114974. https://doi.org/10.1016/J.ESWA.2021.114974

Miao, S., Yang, H., & Gu, Y. (2018). A wind vector simulation model and its application to adequacy assessment. *Energy, 148*, 324–340. https://doi.org/10.1016/J.ENERGY.2018.01.109

Mitra, K., & Gopinath, R. (2004). Multiobjective optimization of an industrial grinding operation using elitist nondominated sorting genetic algorithm. *Chemical Engineering Science, 59*(2), 385–396. https://doi.org/10.1016/J.CES.2003.09.036

Mittal, P., & Mitra, K. (2018). Determining layout of a wind farm with optimal number of turbines: A decomposition based approach. *Journal of Cleaner Production, 202*, 342–359. https://doi.org/10.1016/J.JCLEPRO.2018.08.093

Neshat, M., Majidi Nezhad, M., Mirjalili, S., Piras, G., & Garcia, D. A. (2022). Quaternion convolutional long short-term memory neural model with an adaptive decomposition method for wind speed forecasting: North Aegean islands case studies. *Energy Conversion and Management, 259*, Article 115590. https://doi.org/10.1016/J.ENCONMAN.2022.115590

Neshat, M., Nezhad, M. M., Abbasnejad, E., Mirjalili, S., Tjernberg, L. B., Astiaso Garcia, D., … Wagner, M. (2021). A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the Lillgrund offshore wind farm. *Energy Conversion and Management, 236*, Article 114002. https://doi.org/10.1016/J.ENCONMAN.2021.114002

Ningsih, F. R., Djamal, E. C., & Najmurrakhman, A. (2019). Wind speed forecasting using recurrent neural networks and long short term memory. *Proceedings of the 2019 6th International Conference on Instrumentation, Control, and Automation, ICA 2019*, 137–141. https://doi.org/10.1109/ICA.2019.8916717.

Olaofe, Z. O. (2014). A 5-day wind speed & power forecasts using a layer recurrent neural network (LRNN). *Sustainable Energy Technologies and Assessments, 6*, 1–24. https://doi.org/10.1016/J.SETA.2013.12.001

Prasetyowati, A., Sudibyo, H., & Sudiana, D. (2017). Wind power prediction by using wavelet decomposition mode based NARX-neural network. *ACM International Conference Proceeding Series, 275–278*. https://doi.org/10.1145/3168390.3168434

Renewables in Electricity Production | Statistics Map by Region | Enerdata. (2021). https://yearbook.enerdata.net/renewables/renewable-in-electricity-production-share.html.

Salcedo-Sanz, S., Ortiz-García, E. G., Pérez-Bellido, Á. M., Portilla-Figueras, A., & Prieto, L. (2011). Short term wind speed prediction based on evolutionary support vector regression algorithms. *Expert Systems with Applications, 38*(4), 4052–4057. https://doi.org/10.1016/J.ESWA.2010.09.067

Song, Z., Geng, X., Kusiak, A., & Xu, C. (2011). Mining Markov chain transition matrix from wind speed time series data. *Expert Systems with Applications, 38*(8), 10229–10239. https://doi.org/10.1016/J.ESWA.2011.02.063

Trebing, K., & Mehrkanoon, S. (2020). Wind speed prediction using multidimensional convolutional neural networks. *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, 713–720. https://doi.org/10.1109/SSCI47803.2020.9308323.

Tso, W. W., Burnak, B., & Pistikopoulos, E. N. (2020). HY-POP: Hyperparameter optimization of machine learning models through parametric programming. *Computers & Chemical Engineering, 139*, Article 106902. https://doi.org/10.1016/J.COMPCHEMENG.2020.106902

Vogler, J. (2021). The international politics of COP26. *Scottish Geographical Journal, 136* (1–4), 31–35. https://doi.org/10.1080/14702541.2020.1863610

Wang, J., Gao, D., Zhuang, Z., & Wu, J. (2022). An optimized complementary prediction method based on data feature extraction for wind speed forecasting. *Sustainable Energy Technologies and Assessments, 52*, Article 102068. https://doi.org/10.1016/J.SETA.2022.102068

Wang, J., Li, H., Wang, Y., & Lu, H. (2021). A hesitant fuzzy wind speed forecasting system with novel defuzzification method and multi-objective optimization algorithm. *Expert Systems with Applications, 168*, Article 114364. https://doi.org/10.1016/J.ESWA.2020.114364

Wang, K., Qi, X., Liu, H., & Song, J. (2018). Deep belief network based k-means cluster approach for short-term wind power forecasting. *Energy, 165*, 840–852. https://doi.org/10.1016/J.ENERGY.2018.09.118

World Energy Consumption Statistics | Enerdata. (2021). https://yearbook.enerdata.net/total-energy/world-consumption-statistics.html.

Wu, J., Chen, S. P., & Liu, X. Y. (2020). Efficient hyperparameter optimization through model-based reinforcement learning. *Neurocomputing, 409*, 381–393. https://doi.org/10.1016/J.NEUCOM.2020.06.064

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology, 17*(1), 26–40. https://doi.org/10.11989/JEST.1674-862X.80904120

Wu, Y. X., Wu, Q. B., & Zhu, J. Q. (2019). Data-driven wind speed forecasting using deep feature extraction and LSTM. *IET Renewable Power Generation, 13*(12), 2062–2069. https://doi.org/10.1049/IET-RPG.2018.5917

Xu, J., Zhou, W., Fu, Z., Zhou, H., & Li, L. (2021). *A survey on green deep learning. 3*.

Xue, H., Jia, Y., Wen, P., & Farkoush, S. G. (2020). Using of improved models of Gaussian Processes in order to Regional wind power forecasting. *Journal of Cleaner Production, 262*, Article 121391. https://doi.org/10.1016/J.JCLEPRO.2020.121391

Zhang, S., Wang, C., Liao, P., Xiao, L., & Fu, T. (2022). Wind speed forecasting based on model selection, fuzzy cluster, and multi-objective algorithm and wind energy simulation by Betz's theory. *Expert Systems with Applications, 193*, Article 116509. https://doi.org/10.1016/J.ESWA.2022.116509