Alexandria Engineering Journal (2021) 60, 1221–1229

HOSTED BY

ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

Multistep short-term wind speed prediction using nonlinear auto-regressive neural network with exogenous variable selection

Fuad Noman [a], Gamal Alkawsi [a], Ammar Ahmed Alkahtani [a,*], Ali Q. Al-Shetwi [a], Sieh Kiong Tiong [a], Nasser Alalwan [b], Janaka Ekanayake [c], Ahmed Ibrahim Alzahrani [b]

[a] *Institute of Sustainable Energy (ISE), Universiti Tenaga Nasional, Kajang, Malaysia*
[b] *Computer Science Department, Community College, King Saud University, Saudi Arabia*
[c] *Department of Electrical and Electronic Engineering, University of Peradeniya, Kandy, Sri Lanka*

Received 14 July 2020; revised 22 September 2020; accepted 14 October 2020
Available online 5 November 2020

KEYWORDS

Multistep;
Multivariate analysis;
NARX;
Neural networks;
Wind speed prediction

Abstract Precise wind speed prediction is a key factor in many energy applications, especially when wind energy is integrated with power grids. However, because of the intermittent and nonstationary nature of wind speed, modeling and predicting it is a challenge. In addition, using uncorrelated multivariate variables as exogenous input variables often adversely impacts the performance of prediction models. In this paper, we present a multistep short-term wind speed prediction using multivariate exogenous input variables. We implement different variable selection methods to select the best set of variables that significantly improve the performance of prediction models. We evaluate the performance of eight transfer learning methods, four shallow neural networks (NNs), and the persistence method on predicting the future values of wind speed using ultrashort-term, short-term, and multistep time horizons. We performed the evaluation over two-year high-sampled wind speed data averaged at 10-minute intervals. Results show that Nonlinear Auto-Regressive Exogenous (NARX) model outperformed all other methods, achieving an average mean absolute error (MAE) and root mean square error (RMSE) of 0.2205 and 0.3405 for multistep predictions, respectively. Despite the lower performance of the transfer learning methods (i.e., 0.43 and 0.58 for MAE and RMSE, respectively), it is believed that results could be further improved with a better enhancement of the feature selection and model parameters.

1. Introduction

The significant increase in gas emissions from conventional energy sources and their associated pollution has led to

* Corresponding author.
E-mail address: ammar@uniten.edu.my (A.A. Alkahtani).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2020.10.045
1110-0168

resorting to renewable energy sources as an urgent necessity [1]. Being clean and environmentally friendly, wind energy is one of the fastest and most important renewable sources that has gained significant importance [2]. It has been reported that wind energy has the dominant market share in renewable energy resources and expected to conserve that market share in the future [3]. However, the intermittency of the wind power generation due to the nature of wind presents critical challenges to integrate power energy to the grid [4]. To minimize the impact of the fluctuation, nonlinearity, and uncertainties of wind speed, an accurate short-term wind speed prediction is crucial for energy suppliers and grid operators [5]. The short-term prediction will not only benefit energy suppliers but also policymakers and scientists trying to balance the globally increasing demand for electricity and maintain a clean, pollution-free environment. Reliable wind speed prediction significantly contributes to power generation management and control systems in different ways [6]. Planning and preparing wind energy project necessitates an accurate wind prediction. In addition, the prediction can help maintain the stability and reduce the failure risk of the wind power system. Consequently, this can reduce the operation cost by controlling and optimizing the speed of turbine rotors, reducing the negative impact of wind speed intermittency on the power grid [7].

The literature shows that most of the related studies for wind speed prediction are focused on short-term prediction. Accurate short-term prediction of wind speed is required for effective control of wind plant operations. The methods for predicting wind speed are mainly categorized into five classes [8–11]: 1) Persistence method is based on the assumption that the future wind speed will equal to that at the prediction time. Due to this assumption and the intermittency nature of wind speed, this method is unsuitable for long-term wind speed prediction. However, this method is mostly used as a baseline to benchmark with other methods [6]. 2) Physical models use spatiotemporal relation of different variables (e.g., weather data) to predict the future wind speed. Examples of this method include numerical weather prediction (NWP), which utilizes complex mathematical models and is usually used for long-term predictions due to their high computational complexity [12], and computational fluid dynamics [13], which is used for wind power forecasting. Compared to statistical models, this method shows comparable but statistically less variation in prediction error. However, physical models perform poorly in short-term wind prediction [14]. 3) Statistical models employ the history of the observations of wind data to predict future wind speed values [15]. The well-known family of autoregressive models, which is superior in modeling heterogenous data in short-term intervals, is an example of these models. However, such models are strongly dependent on the stationarity assumption and initialization of their hyperparameters, and their performance usually degrades for long-term predictions. 4) Machine learning methods are composed of neural network (NN) and the recent deep learning approaches [16–19]. These methods use the supervised mechanism to automatically learn the underlying structure of wind data and their relation to the desired prediction wind speed values. The main advantage of these methods is that they are not restricted to the dynamic structure of the data (e.g., stationarity). However, machine learning methods lately show superior performance compared to other conventional methods, and this may fail according to the data availability and models' tuning procedure. 5) Hybrid models are based on an integration of more than one approach to benefit from their superior individual features to boost prediction performance. Most hybrid models deploy decomposition techniques with other prediction methods. Some of these hybrid approaches are thoroughly reviewed in [18,20].

Most of the wind speed prediction methods reported in the recent literature use univariate history observations of wind speed to predict the future ones, limiting prediction methods to only model the temporal characteristics of lagged wind speed data and ignoring the possible spatiotemporal correlations of other multivariate exogenous variables. However, due to the possible strong correlations between some meteorological variables and the wind speed values, these variables can be used as input to improve the performance of prediction models. In this paper, to evaluate the performance of prediction models, real wind speed data sampled at 10-minute intervals for two years were used. Moreover, the large dimensionality and redundancy present in the input multivariate variables usually degrade the performance of prediction model and mostly lead to model over- or under-fitting. Thus, four feature selection methods were evaluated to choose the best variables that significantly impact the performance of wind speed prediction models. Using time horizons at different terms (i.e., ultrashort-term, short-term, and multistep), performance comparisons were conducted on eight different transfer learning methods, four shallow NNs, and a persistence method.

The rest of the paper is organized as follows: Section 2 presents wind datasets used in this study and outlines the related data preprocessing steps. Section 3 describes in detail the feature selection and prediction methods. Section 4 reports the analysis and comparative evaluations between the introduced methods under different criteria, as well as the verification and discussion of such results. Finally, Section 5 concludes the paper.

2. Database

The publicly available wind data were retrieved from the National Renewable Energy Laboratory meteorological tower located at the western edge of Flatirons Campus (M2 tower), south of Boulder, Colorado, formally known as National Wind Technology Centre (NWTC). The data were measured at six different hub-heights: 2, 5, 10, 20, 50, and 80 m [21]. Several variables are measured or derived during the data acquisition, including a total of 54 exogenous variables initially averaged over 1 min.

In this paper, as shown in Table 1, the wind data related to the 10-meter hub-height (18 variables) were selected for the prediction purpose. The wind speed data were preprocessed in which all the NaN and encoded values are imputed using a sliding window average. The 1-minute wind speed data were then downsampled to 10 min using a simple scalar average of the wind data observations, regardless of gust effects as recommended by the World Meteorological Organization (WMO) [22].

The NWTC-M2 tower contains a continuous recording of wind data from 1996 to 2020. As shown in Fig. 1, two-year data were randomly selected: the 2017 data is considered as the training and validation set, and the 2018 data is used as a testing set to evaluate the proposed models. The data

Table 1 Descriptive statistics of the training and testing sets of wind speed.

	Training		Testing	
Variable	Average	Standard deviation	Average	Standard deviation
Wind speed 10 m	3.4647	2.5921	3.2898	2.5455
Wind speed std 10 m	0.4513	0.4201	0.4513	0.4121
Global PSP accumulated	2.1860	2.3053	2.2174	2.3582
Specific humidity	5.0600	2.8026	4.7243	2.8857
Precipitation accumulated	0.6639	3.2236	0.5953	3.1521
Peak wind speed 10 m	4.3381	3.3773	4.1619	3.3164
Wind shear	0.0514	0.0430	0.0502	0.0554
Dew point temperature	−0.5550	7.9047	−1.6671	8.3262
Station pressure	817.0365	7.0678	813.8459	5.3541
Temperature 2 m	10.7146	9.6082	10.3064	10.3034
Turbulence intensity 10 m	0.1319	0.0597	0.1276	3.4904

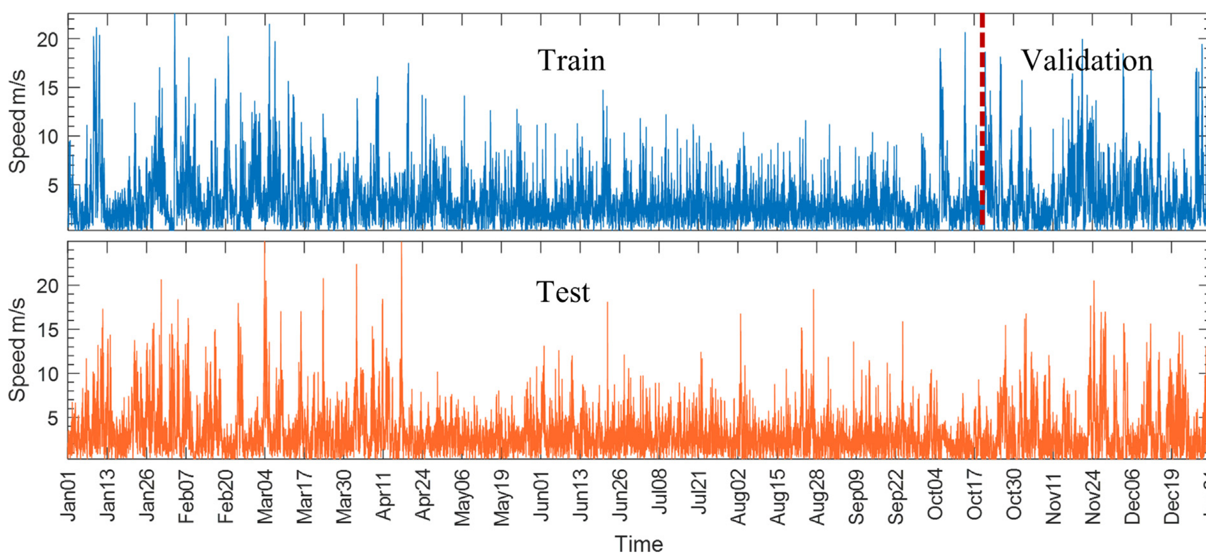

Fig. 1 Wind speed data at 10 m. All wind data variables of year 2017 data were partitioned into training and validation with a ratio of 80% and 20%, and the year 2018 data were used for testing.

variables were standardized by removing the mean and dividing by the standard deviation for the machine learning applications in this study.

3. Methods

Fig. 2 shows the typical procedure and methodology steps of wind speed prediction. Two prediction approaches are used to allow a performance comparison of univariate and multivariate input data. The multivariate input data of L lag is transformed to image-like data of compatible dimension (32×32) with the pretrained deep learning models. In ANN methods, the multivariate data were flattened (vectorized) to be suitable with ANN input layer requirement.

3.1. Input variable selection

Wind speed modeling is challenging because it is a nonstationary time series and mostly contains seasonal effects. Different weather variables usually influence wind speed data measurements. Selecting the best combination of these variables, which have high correlations with wind speed data, can positively impact the performance of the prediction model. Besides, the large dimensionality and redundant data present in the input variables usually degrade the prediction model performance and mostly lead to model over- or under-fitting problem. In this study, four feature selection methods, which have been proposed in the literature, including sequential feature selection (SFS) [23], rank importance of predictors (ReliefF) [24], neighborhood component analysis (NCA) for regression [25], and Gaussian process regression (GPR) [26], were used to reduce the variables' dimension.

3.2. Prediction methods

In this subsection, we first introduce the transfer learning-based models and then present the shallow architectures of NNs.

3.2.1. Transfer learning

We consider several pretrained deep learning models offered by Keras applications [27] for the implementation of transfer

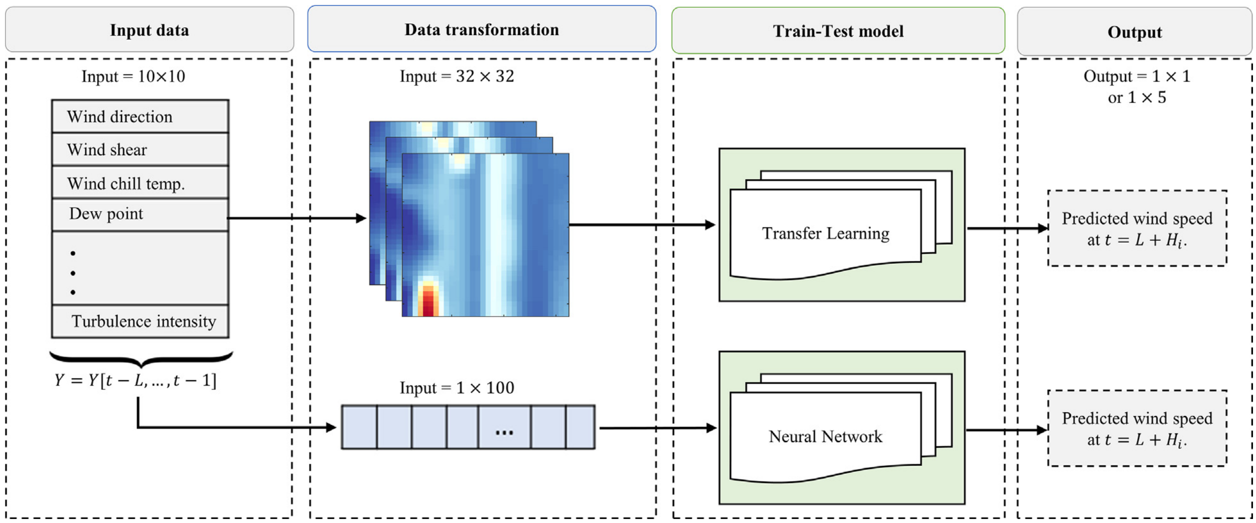

Fig. 2 Schematic diagram of the procedure used for wind speed prediction. The input data Y has the dimension of L × 10, where L = 10 is the time lag and 10 is the wind input variables. The output dimension is either for one-step-ahead prediction (1 × 1) or multistep prediction of five-horizon ahead (H_i : 1 × 5). For transfer learning, the L × 10 input is transformed to (32 × 32 × 3) image-like tensor. The added regression layer on top of the transfer learning model is customized to perform either one step or multistep ahead predictions. On the other hand, for neural network model, the aggregated L × 10 vector (by concatenation) is used as input for one or multistep predictions.

learning for wind speed prediction. These pretrained models were originally trained on the Imagenet dataset [28] using 1.28 million images and 1000 classes. For each pretrained model, we kept the feature extraction (convolutional) block and replaced the classification (fully connected) layers by a Global Average Pooling [29] and two fully connected layers with 1024 and 512 neurons, respectively, as shown in Fig. 3. We also attach each fully connected layer with a dropout layer [30] of 0.2 drop-ratio as regularization method to prevent the over-fitting. To make our wind data compatible with the original pretrained input layer, which uses image data of tensor shape 224 × 224 × 3, the wind data ($L \times N$) were transformed to gray image-like tensor shape 32 × 32 and then replicated three times to form a wind data tensor shape 32 × 32 × 3. The output layer consists of single neuron that outputs the predicted wind speed values with linear activation function.

We report several transfer learning methods in this study, including VGG16 [31], VGG19 [31], ResNet50 [32], ResNet101 [32], ResNet152 [32], DenseNet121 [33], DenseNet169 [33], and DenseNet201 models [33]. However, when all available pretrained models in Keras application were tested, only the models reported here show acceptable prediction performance. When training these models, we only trained the added fully connected layers with randomly initialized weights and froze the pretrained convolutional blocks. Here we used Adam optimizer [34] with a learning rate of 10^{-3}, momentum exponential decay rates of $\beta_1 = 0.9$ and $\beta_2 = 0.999$, regularization rate of 10^{-8}, and learning rate decay of 10^{-6}. The models were trained with batch size of 128 for 100 epochs on Google Colab. Pro. cloud [35]. We also implement early stopping criteria to monitor the prediction performance loss (error) on the validation wind dataset, and the best models were saved for testing purpose.

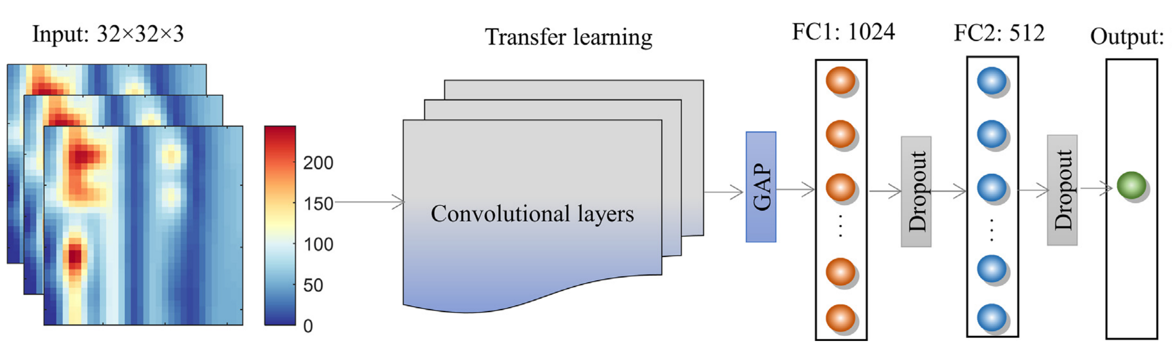

Fig. 3 The architecture of transfer learning with regression layers for wind speed prediction. FC is a fully connected layer, GAP is global average pooling layer. The wind speed history data are transformed to compatible dimension of that required by transfer learning models. The convolution layers of the transfer learning model extract more meaningful features and outputs a three-dimensional tensor which is then aggregated to one-dimensional vector by using the GAP layer. The last FC layers act as prediction layers where the output layer performs the regression and outputs either one or multistep predictions.

3.2.2. Neural networks

The ANN models have been proven to be useful in predicting wind power and wind speed due to their robustness to solve nonlinear and multivariate modeling [17], as well as their efficiency with physical processes that are complicated or not understood [36]. ANNs are more efficient in short-term wind speed prediction than other conventional methods (e.g., statistical or physical methods). In this study, three ANN methods were deployed to predict wind speed as discussed below.

A. Feed-Forward Neural Network

The Feed-Forward Neural Networks (FFNNs) are the most applied recurrent NNs containing multilayer architecture and representing the static nonlinear models [37]. The FFNN architecture consists of input and output neural layers and has two or more hidden layers. Every layer has a number of neurons, and each neuron with assigned set of weights is fully interconnected to all neurons in the adjacent layer. The activation function in the hidden layer passes the summed-up signals to the output layer. The FFNN architecture with a single layer or multiple layers can be used as a universal approximator [38]. To predict wind speed at a very complex topography, the FFNN was proposed by [39]. They indicated that the proposed method in their work could accurately predict the wind speed despite the unsteady characteristics of the wind. Differently, to forecast the wind power, [5] proposed a two-stage model, which comprised of FFNN and a wavelet ANN, and demonstrated the effectiveness of the proposed methods.

B. Time-Delay Neural Network

Fundamentally, the Time-Delay Neural Network (TDNN) is a feed-forward network, but the connections between layers are amended. TDNN is one of the dynamic network used for prediction [40]. The tapped delay lines are used at the input of the network or each neuron's input. The TDNN can recognize the local features (e.g., bursts of heavy rain) within a huge pattern, separated from the local features [41]. To assess the architectures of different NNs in predicting rainfall, [42] conducted a comparison of Recurrent Neural Network, Multilayer Feed-Forward NN, and TDNN based on data size, delay, training memory, and learning speed. The TDNN was found to be the best model in predicting rainfall as the TDNN architecture has the tapped delay lines at the hidden or input layers. In [43], a variety of NNs and other methods were evaluated for next-hour prediction of wind speed. The TDNN was found to achieve the best performance among the other NNs with 7.8% lower root mean square error (RMSE) [43].

C. Nonlinear Auto-Regressive Exogenous

The Nonlinear Auto-Regressive Exogenous (NARX) network is composed of a multilayer perceptron (MLP), which takes as an input the earlier independent inputs and earlier outputs, and computes the current output. Unlike the conventional recurrent NN, the NARX network has restricted feedback expected only from the output layer, rather than from the hidden layer [44]. The output of the NARX network is fed back as input to the FFNN. The learning process of the NARX network is more effective than other NNs due to the improvement of its generalization and convergence compared to the other NNs [45]. The NARX network is capable of using the same structure to make up different models, decreasing the computation cost. For example, in [46], the multivariate NARX model and the univariate ARIMA model were used to predict wind speed, and results show a better achievement from the NARX model. In another study [46], the nonlinear auto-regressive model was compared with the NARX model against the persistence method based on mean absolute error (MAE), MSE, and MAPE. The NARX model found to be as the most accurate among the three models, concerning the significance of involving the additional features in prediction models.

4. Results

4.1. Evaluation metrics

The performance of the prediction models was evaluated using two statistical metrics: MAE as in (1) and RMSE as in (2):

$$MAE = \frac{1}{M}\sum_{n=1}^{M}|e(n)| \tag{1}$$

$$RMSE = \sqrt{\frac{1}{M}\sum_{n=1}^{M}e(n)^2} \tag{2}$$

where $e(n) = \hat{y}(n) - y(n)$ and $M$is the number of wind speed observation samples in the testing set. $\hat{y}(n)$ and $y(n)$ are the predicted wind speed output and the actual wind speed output for the n^{th} sample, respectively.

4.2. Input variable selection

During the sensitivity analysis of input variable selection, to evaluate the variable selection methods, we used data that is equivalent to one month (3000 data samples) for the training and for the test. The original wind data at 10 m consists of a total of 18 variables, including the response variable (wind speed at 10 m). The selection methods were designed to select the best 10 variables from the remaining 17 predictors according to the attribute weights or scores of these variables for each particular method. We used the support vector regression (SVR) [47] and long short-term memory (LSTM) [48] to evaluate the prediction performance of the selected variables for one-step-ahead prediction, which reveals the best-performing variable selection method.

Fig. 4 shows the performance of the suggested variable selection methods. Fig. 4(a) summarizes the SVR and LSTM prediction performance. Since the SVM and LSTM ignore the spatial correlation of the wind data, the input wind data of the best selected 10 variables were flattened to form a vectorized input data of shape 1 × 100, which allows the SVM and LSTM to leran the temporal relation of these variables. All methods show comparable MAE performance when using the SVR method. However, the GPR method outperforms all other selection methods, maintaining the lowest MAE prediction errors for both SVR and SVM models. Fig. 4(b) shows the fitting performance of the predicted wind speed vs the actual values when using the entire set of variables. In this case, the

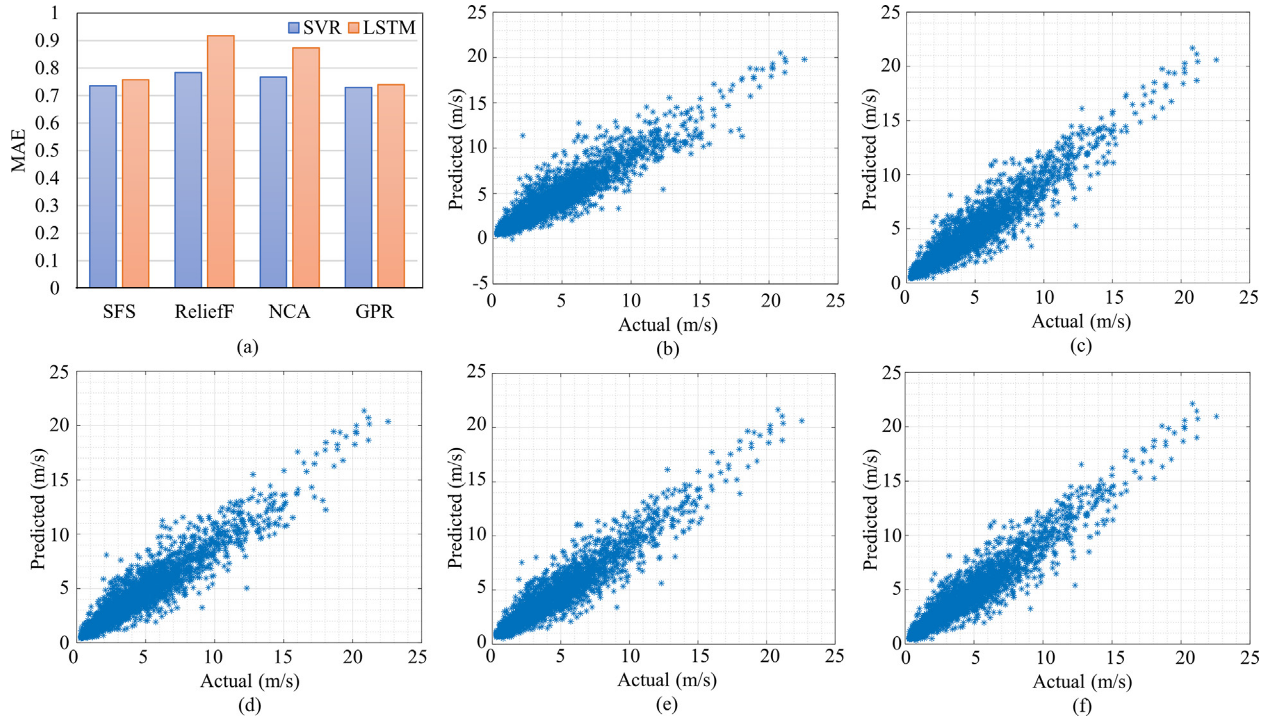

Fig. 4 Performance of wind variable selection methods. (a) Comparison of the SVR and LSTM prediction MAE when using the best variable set of each selection method. (b) The SVR prediction performance without variable selection (MAE = 0.8385) represented by a scatterplot of the predicted vs actual wind speed values. (c, d, e, and f) The predicted vs actual wind speed using SFS, ReliefF, NCA, and GPR variable selection methods, respectively.

prediction performance is slightly low with MAE of 0.8385, confirming that there are some variables in the input data that negatively impact the prediction performance. Fig. 4(c–f) show the scatterplots of the predicted vs actual wind speed data for all the utilized variable selection methods. Fig. 4(f) of the GPR selection method shows slight improvement compared to other variable selection methods.

It is worth mentioning that the temporal lag of the input data in this paper is randomly selected to be equal to the number of input variables (10). Additional lag-selection results not reported here showed that there is no significant relation of increasing the amount of data lag on the prediction performance of the proposed methods in this paper. This is due to the exogenous nature of the input variables, unlike the heterogenous multivariate data, which can be easily modeled with auto-regressive models and processed in any model-based lag-selection methods. On the other hand, the sequential increase of lag data reveals that the deep learning-based methods are strongly dependent on fine-tuning the model hyperparameters rather than slightly increasing or decreasing the amount of history data lag.

4.3. Prediction results

As shown in Table 2 and Table 3, data-driven transfer learning methods performed comparably. Among the deep learning-based models, DenseNet201 shows slightly better performance for ultrashort term but yields higher MAE and RMSE errors when increasing the time steps or using multistep predictions. This is because of the following reasons: 1.) the deep learning methods were trained for image classification; 2.) during the transfer learning procedure in this experiment, the pretrained model weights were kept frozen (fixed); and 3.) the pretrained models are limited by input data dimension, in which we upsample the input data from 10 × 10 to 32 × 32 to be suitable for the models' input layer. DenseNet201 improves RMSE by 10.32% for 10-minute-ahead prediction compared to VGG19. Such improvement reaches 11.05% for multistep prediction.

Among the shallow NN architectures, the NARX performed best. It also notably outperforms the deep learning models duo to the self-feedback structure of output wind speed as input. Compared to DenseNet201, NARX has improved the average MAE and RMSE by 54.04% and 52.13%, respectively, over the five time horizons and also showed a 58.61% and 54.97% improvement in MAE and RMSE, respectively, for multistep horizons. This shows the ability of NARX, which utilizes the 1 × 100 input variables plus the self-feedback prediction.

Table 4 summarizes the comparison of the best-performing methods (DenseNet201 and NARX) with persistence method (PR) and with the roughly stacked denoising auto-encoder proposed by Khodayar et al., [8]. As shown in Table 4, the PR method achieved a satisfactory prediction performance compared to the transfer learning approach. However, the PR performance degrades as the number of time steps increases. Meanwhile, NARX showed superior performance and outperformed all other methods, showing consistent errors throughout all time steps. Nevertheless, NARX is leveraging the self-feedback structure, allowing it to use wind speed observations as input leading to more robust estimates of the future

Table 2 MAE of prediction methods for different time horizons.

	One step					Multistep
Method	10 min	30 min	1 hr	2 hr	3 hr	10 min–3 hr
VGG16	0.4897	0.5603	0.5851	0.6391	0.6355	0.5700
VGG19	0.4933	0.5403	0.5948	0.6108	0.6548	0.5990
ResNet50	0.4531	0.5343	0.5411	0.5959	0.6242	0.5588
ResNet101	0.4451	0.5295	0.5669	0.5980	0.6266	0.5585
ResNet152	0.4606	0.5156	0.5450	0.5996	0.6096	0.5487
DenseNet121	0.4528	0.5111	0.5823	0.5986	0.6158	0.5606
DenseNet169	0.4588	0.5199	0.5714	0.6121	0.6209	0.5643
DenseNet201	**0.4317**	**0.5035**	**0.5413**	**0.5927**	**0.6138**	**0.5328**
FFNN	0.2432	0.3756	0.4908	0.5610	0.6151	0.4700
MLP	0.2596	0.3886	0.4761	0.5547	0.6057	0.4607
TDNN	0.3205	0.4348	0.4981	0.5687	0.6089	0.4837
NARX	**0.2408**	**0.2426**	**0.2421**	**0.2441**	**0.2450**	**0.2205**

Table 3 RMSE of prediction methods for different time horizons.

	One step					Multistep
Method	10 min	30 min	1 hr	2 hr	3 hr	10 min–3 hr
VGG16	0.6539	0.7479	0.8108	0.8870	0.8919	0.7928
VGG19	0.6570	0.7314	0.8230	0.8545	0.9043	0.8206
ResNet50	0.6192	0.7381	0.7688	0.8503	0.8833	0.7916
ResNet101	0.6175	0.7505	0.8037	0.8492	0.8760	0.7962
ResNet152	0.6269	0.7149	0.7699	0.8404	0.8703	0.7727
DenseNet121	0.6217	0.7185	0.8263	0.8459	0.8877	0.7833
DenseNet169	0.6269	0.7156	0.7854	0.8613	0.8937	0.7841
DenseNet201	**0.5892**	**0.7001**	**0.7687**	**0.8457**	**0.8800**	**0.7561**
FFNN	0.3779	0.5550	0.7198	0.8074	0.8713	0.7391
MLP	0.4637	0.5738	0.6863	0.6281	0.8598	0.6786
TDNN	0.4728	0.6113	0.7254	0.7998	0.8563	0.7122
NARX	**0.3605**	**0.3600**	**0.3580**	**0.3330**	**0.3590**	**0.3405**

Table 4 Comparison of prediction performance with persistence method and roughly stacked denoising auto-encoder [8]

		One step					Multistep
Metric	Method	10 min	30 min	1 hr	2 hr	3 hr	10 min–3 hr
MAE	PR	0.2422	0.3918	0.5049	0.6271	0.6943	0.4921
	DenseNet201	0.4317	0.5035	0.5413	0.5927	0.6138	0.5328
	NARX	**0.2408**	**0.2426**	**0.2421**	**0.2441**	**0.2450**	**0.2205**
	[8]	0.2130	0.5830	0.7210	0.8910	1.3610	NA
RMSE	PR	**0.3546**	0.5814	0.7425	0.9114	1.0049	0.7557
	DenseNet201	0.5892	0.7001	0.7687	0.8457	0.8800	0.7561
	NARX	0.3605	**0.3600**	**0.3580**	**0.3330**	**0.3590**	**0.3405**
	[8]	0.5210	0.6980	1.2400	1.6680	1.8520	NA

wind speed values. This property can be used as a limitation of NARX since other methods are designed to predict the future values of wind speed using exogenous variables without any involvement of history wind speed observations.

5. Conclusion

Wind speed prediction is challenging because of the nonstationary and seasonal behavior of wind data. In this paper, transfer learning models and shallow NNs were presented for ultrashort-term, short-term, and multistep ahead wind speed prediction. Variable selection methods were implemented to reduce the dimension on input wind data and select the most relevant variables to improve the overall prediction performance. The prediction models were trained on one-year wind data and evaluated on another hidden set of one year data. Among the proposed methods, shallow architecture NN of NARX showed superior prediction performance with limited prior knowledge of wind speed data history. The transfer learning models achieved acceptable MAE and RMSE prediction errors; however, the wind data transformation to image-like tensors could be further enhanced for better performance compared to baseline methods (i.e., persistence method). The codes for the development, evaluation, and performance measures of prediction methods are open sourced with no restrictions and are available in https://github.com/MNFuad/Multivariate-Multistep-Wind-Speed-Prediction.

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.

Acknowledgment

The authors would like to acknowledge the support provided by Universiti Tenaga Nasional (UNITEN), Malaysia, under BOLD2025 and RJO10517844/114. The work was also funded by the Researcher Supporting Project (RSP-2020/157), King Saud University, Riyadh, Saudi Arabia.

References

[1] J.F. Manwell, J.G. McGowan, A.L. Rogers, Wind energy explained: Theory, design and application, John Wiley & Sons, 2010.
[2] W.Y.Y. Cheng, Y. Liu, A.J. Bourgeois, Y. Wu, S.E. Haupt, Short-term wind forecast of a data assimilation/weather forecasting system with wind turbine anemometer measurement assimilation, Renew. Energy 107 (2017) 340–351.
[3] C. Huang, F. Li, Z. Jin, Maximum power point tracking strategy for large-scale wind generation systems considering wind turbine dynamics, IEEE Trans. Ind. Electron. 62 (4) (2015) 2530–2539.
[4] S. Fang, H.-D. Chiang, A high-accuracy wind power forecasting model, IEEE Trans. Power Syst. 32 (2) (2016) 1589–1590.
[5] K. Bhaskar, S.N. Singh, AWNN-assisted wind power forecasting using feed-Forward Neural Network, IEEE Trans. Sustain. Energy 3 (2) (2012) 306–315.
[6] Z. Sun, H. Sun, J. Zhang, Multistep wind speed and wind power prediction based on a predictive deep belief network and an optimized random forest, Math. Probl. Eng. 2018 (2018) 1–15.
[7] C. Wang, J. Wu, J. Wang, Z. Hu, Short-term wind speed forecasting using the data processing approach and the support vector machine model optimized by the improved cuckoo search parameter estimation algorithm, Math. Probl. Eng. 2016 (2016) 1–17.
[8] M. Khodayar, O. Kaynak, M.E. Khodayar, Rough deep neural architecture for short-term wind speed forecasting, IEEE Trans. Ind. Inf. 13 (6) (2017) 2770–2779.
[9] M.S. El Moursi, A. Al Hinai, E.B. Ssekulima, M.B. Anwar, Wind speed and solar irradiance forecasting techniques for enhanced renewable energy integration with the grid: A review, IET Renew. Power Gener. 10 (7) (2016) 885–989.
[10] F.M. Noman, G.A. Alkawsi, D. Abbas, A.A. Alkahtani, S.K. Tiong, J. Ekanayake, Comprehensive review of wind energy in Malaysia: Past, present, and future research trends, IEEE Access 8 (2020) 124526–124543.
[11] H. Liu, C. Chen, Data processing strategies in wind energy forecasting models and applications: A comprehensive review, Appl. Energy 249 (2019) 392–408.
[12] S. Buhan, Y. Özkazanç, I. Çadırcı, Wind pattern recognition and reference wind mast data correlations with NWP for improved wind-electric power forecasts, IEEE Trans. Ind. Inf. 12 (3) (2016) 991–1004.
[13] Y. Wang, Y. Liu, P. Song, H. Liu, and Z. Hu, Impact of the accuracy of NWP wind speed forecasts on wind power forecasting, 2019.
[14] N. Huang et al, Short-term wind speed forecast with low loss of information based on feature generation of OSVD, IEEE Access 7 (2019) 81027–81046.
[15] M. Lydia, S. Suresh Kumar, A. Immanuel Selvakumar, G. Edwin Prem Kumar, Linear and non-linear autoregressive models for short-term wind speed forecasting, Energy Convers. Manag. 112 (2016) 115–124.
[16] H. Liu, C. Chen, X. Lv, X. Wu, M. Liu, Deterministic wind energy forecasting: A review of intelligent predictors and auxiliary methods, Energy Convers. Manag. 195 (2019) 328–345.
[17] A.P. Marugán, F.P.G. Márquez, J.M.P. Perez, D. Ruiz-Hernández, A survey of artificial neural network in wind energy systems, Appl. Energy 228 (2018) 1822–1836.
[18] J. Heinermann, O. Kramer, On heterogeneous machine learning ensembles for wind power prediction, in: Workshops at the twenty-ninth AAAI conference on artificial intelligence, 2015.
[19] J. Manero, J. Béjar, U. Cortés, Wind energy forecasting with neural networks: A literature review, CyS 22 (4) (2018).
[20] Z. Qian, Y. Pei, H. Zareipour, N. Chen, A review and discussion of decomposition-based hybrid models for wind energy forecasting applications, Appl. Energy 235 (2019) 939–953.
[21] A. Clifton, 135-m meteorological masts at the National Wind Technology Center: Instrumentation, data acquisition, and processing, Report, July 2014.
[22] B.A. Harper, J.D. Kepert, J.D. Ginger, Guidelines for converting between various wind averaging periods in tropical cyclone conditions, WMO Geneva, Switzerland, 2010.
[23] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1–2) (1997) 273–324.
[24] I. Kononenko, Estimating attributes: Analysis and extensions of RELIEF, in: Lecture Notes in Computer Science European conference on machine learning, pp. 171–182, 1994.
[25] J. Goldberger, G.E. Hinton, S.T. Roweis, R.R. Salakhutdinov, Neighbourhood components analysis, Adv. Neural Inf. Process. Syst. (2005) 513–520.
[26] C. E. Rasmussen and C. Williams, Gaussian processes for machine learning, MIT Press: Cambridge, MA, 2006.
[27] F. Chollet, Keras, 2015 [Online]. Available: https://keras.io.
[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, IEEE conference on computer vision and pattern recognition 2009 (2009) 248–255.
[29] M. Lin, Q. Chen, S. Yan, 'Network in network', *arXiv Prepr*, ArXiv1312.4400 (2013).
[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[31] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv Prepr, ArXiv1409.1556, 2014.
[32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[33] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[34] D.P. Kingma, J. Ba, 'Adam: A method for stochastic optimization', *arXiv Prepr*, ArXiv1412.6980 (2014).
[35] Google, Google Colab pro, 2020 [Online]. Available: https://colab.research.google.com/.

[36] Z.O. Olaofe, A 5-day wind speed & power forecasts using a layer recurrent neural network (LRNN), Sustain. Energy Technol. Assess. 6 (2014) 1–24.
[37] G. Capizzi, F. Bonanno, C. Napoli, A wavelet based prediction of wind and solar energy for long-term simulation of integrated generation systems, SPEEDAM 2010 (2010) 586–592.
[38] A. Laudani, G.M. Lozito, F. Riganti Fulginei, A. Salvini, On training efficiency and computational costs of a Feed Forward Neural Network: A review, Comput. Intell. Neurosci. 2015 (2015) 818243.
[39] K. Philippopoulos, D. Deligiorgi, Application of artificial neural networks for the spatial estimation of wind speed in a coastal region with complex topography, Renew. Energy 38 (1) (2012) 75–82.
[40] T.G. Barbounis, J.B. Theocharis, Locally recurrent neural networks for wind speed prediction using spatial correlation, Inf. Sci. (Ny) 177 (24) (2007) 5775–5797.
[41] K.C. Luk, J.E. Ball, A. Sharma, An application of artificial neural networks for rainfall forecasting, Math. Comput. Modell. 33 (6–7) (2001) 683–693.
[42] M.P. Darji, V.K. Dabhi, H.B. Prajapati, Rainfall forecasting using neural network: A survey, in: Conference Proceeding International Conference on Advances in Computer Engineering and Applications, 2015, 2015,, p. 2015.
[43] I.J. Ramırez-Rosado, L.A. Fernández-Jiménez, An advanced model for short-term forecasting of mean wind speed and wind electric power, Control Intell. Syst. 32 (1) (2004) 21–26.
[44] A. Di Piazza, M.C. Di Piazza, G. Vitale, Solar and wind forecasting by NARX neural networks, Renew. Energy Environ. Sustain. 1 (2016).
[45] Y. Gao, M.J. Er, NARMAX time series model prediction: Feedforward and recurrent fuzzy neural network approaches, Fuzzy Sets Syst. 150 (2) (2005) 331–350.
[46] E. Cadenas, W. Rivera, R. Campos-Amezcua, C. Heard, Wind speed prediction using a univariate Arima model and a multivariate NARX model, Energies 9 (2) (2016).
[47] H. Drucker, C.J.C. Burges, L. Kaufman, A.J. Smola, V. Vapnik, Support vector regression machines, Adv. Neural Inf. Process. Syst. (1997) 155–161.
[48] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.