PERFORMANCE COMPARISON OF PROBABILISTIC AND ARTIFICIAL NEURAL NETWORK MODELS FOR LONG-SEQUENCE GENERATION OF WIND SPEED FORECASTS

Diana C Pina-Gongora[1], Behzad Kazemtabrizi[1], Christopher Crabtree[1]*

[1]*Department of Engineering, Durham University, Durham, United Kingdom*
**E-mail: diana.c.pina-gongora@durham.ac.uk*

Keywords: ARMA, LSTM, 1D-CNN-LSTM, TIME SERIES FORECASTING, WIND SPEED

Abstract

This paper presents a new method for generating long-sequence wind speed time-series forecasts for purposes of offshore wind farm asset and operations planning. Our goal is to develop a planning decision support tool with which wind farm planners and operators can make informed decisions for development and operation of future offshore wind assets considering revenue and power generation yield, as well as operation and maintenance expenditures.The proposed methodology should be computationally efficient and should be able to reliably generate accurate wind speed time-series forecasts for the required planning timescale. In this paper, we used an Autoregressive Moving average model as benchmark to evaluate and compare the performance of four different artificial neural network models namely, uni-variate Long-Short Term Memory (LSTM), uni-variate hybrid one dimensional convolutional neural network with LSTM (1D-CNN-LSTM), multivariate Long-short term memory and multivariate hybrid 1D-CNN-LSTM architectures. The performance evaluation is delivered through the statistical comparison of the metrics, RMSE, MAE and MAPE, and the final selection of the outperforming model is supported using Diebold-Mariano statistic test. Experiments consists of applying different types of pre-processing to the wind speed dataset and the modification of models' architecture to include either a batch normalization or a drop-out regularization layer are realized to aid in the selection of the most suitable model engineering. Results suggest the uni-variate hybrid 1D-CNN-LSTM is able to deliver short-term prediction for longer timescales while maintaining a suitable degree of accuracy.

1 Introduction

Wind speed prediction is an important task as future resource projection is used in the power generation estimation for wind farm planning, maintenance scheduling, operation management, energy dispatch scheduling, etc. Moreover, with the rapid development of higher capacity wind turbines, floating and far from shore wind farms, the power forecasting is subject to the changes in different scenarios. Thus, given that wind speed is proportional to power generation it is of utmost importance to find suitable models to accurately capture wind speed patterns and project them over time without compromising computational resources and preciseness.

Forecasting models can be classified into persistence, physical and statistical models [1]. The persistence methods are the most simplistic way to predict wind and have a good level of accuracy for short-time predictions. Physical models, also called deterministic[2], are mostly based on numerical weather prediction methods, well known for their high accuracy but they often have higher level of complexity and are computationally taxing. Statistical methods stand in middle ground and work on the basis of historical data. Time series Autoregressive and Artificial Neural Networks (ANNs) fall into this category. Models such as Autoregressive Moving Average (ARMA) or Autoregressive Integrated Moving Average (ARIMA) are among the most classical methods for short time forecasting and are widely accepted due to their low computational resource requirements. Authors such as [3] used them to generate short term predictions for either revenue or power forecasting for longer timescales. O0n the other hand, the use of neural networks in the wind speed and power forecasting has been gaining acceptance due to its capacity to model complex relations [4]. Among ANN models , Recurrent Neural Networks (RNNs) and, particularly Long-short Term Memory (LSTM) Networks have been addressed in the wind speed and power forecasting field due to their ability to capture and propagate time series patterns for longer periods of time [5].

Forecast model performance comparative studies for wind speed, power and energy in general have been widely addressed in literature. For example [6] concluded the designed ARMA model forecast's MAE was 3.6% smaller than the ANNs models for ultra short and short- term wind power forecasting applications, however . The performance of the models is measured using metrics such as Root Mean Square Error (RMST) and Mean Absolute Error (MAE). Likewise, in [7] a comparison between the ARMA, ANN and Support Vector Machines (SVM) is conducted for short-term forecasting of wind speed and direction. Comparative studies of hybrid ANN models are performed by [8, 9] where the forecasts horizons are for short and long term wind speed and direction, suggesting the use of hybrid models can increase the accuracy in the predictions. The performance evaluation delivered in these studies are mostly based on observing the RMSE and MAPE differences between models, and do not offer stronger evidence on whether the model outperforms because of the data set randomness or because is truly predicting accurately [10]. A good way to support the selection of a forecasting model over others is through statistical significance tests [11]. For example, [12] proposes a hybrid Factor Artificial Neural Network model and evaluates its performance against a Dynamic Factor model and an Autoregressive model, the comparison is based on the RMSE and Diebold-Mariano (DM) test. In [13] a NARNET models is proposed and compared against a traditional ARMA and persistence model . The study concludes the most suitable model for power forecasting based on short-time wind speed prediction is the ANN based on the RMSE and MAE values, however, the study supports this claim by obtaining the DM statistic for each model.

This work proposes four different ANN architectures for wind speed short term forecast for generating longer time series. These architectures are based on the use of the LSTM aiming to improve their accuracy by adding a CNN layer before the LSTM and aiding the forecast by the inclusion of other variables correlated to the forecast. Thus, the proposed ANN based models are a univariate LSTM, a univariate hybrid 1-dimension CNN-LSTM (1D CNN-LSTM), a multivariate LSTM and a multivariate 1D CNN-LSTM, which are compared against a classical ARMA(p,q) model and evaluated using the statistical metrics RMSE, Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Furthermore, the competing model forecasts are evaluated using th DM test for statistical significance. The predictions are one-hour ahead for sequences spanning one to two years. In summary, the key contributions of this paper are:

- Development of the methodology to determine a classic ARMA(p,q) model and ANN based model based on ablation studies to find the most suitable architecture and pre process for wind speed data.
- Implementation of hybrid ANN based model for wind speed prediction and the performance comparison with multivariate versions for long term sequences generation.
- a methodology for statistical performance evaluation of forecasting models by incorporating the classic statistical metrics RMSE, MAE and MAPE and the estimation of the DM statistic as a mean to support the performance of the model.

2 Forecast Models

2.1 Auto-regressive Moving Average

This model is composed by an autoregressive part (denoted by $AR(p)$) which takes the influence of p past values over the current value at time t and a moving average (denoted by $MA(q)$) which is a weighted sum of previous q errors. For time-series Y an ARMA(p,q) model is described as:

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \tag{1}$$

Where the y_t is the actual value of Y at time t, ϕ_i is the autorregresive coefficient, θ_j is the moving average coefficient and ϵ_t is the Gaussian Noise added to the model to represent the unpredicted component of a time series in nature. The modeling can be approached following the Box and Jenkins methodology [14] to identify stationarity, estimate the parameters p and q and evaluate the final result.

2.2 Long-Short Term Memory Neural Network

Recurrent Neural Networks (RNNs) are characterized by the use of recurrence in every timestamp and the use of backpropagation to fine-tune the weights in the neural network. The LSTM are a type of RNN that incorporates a memory cell that captures the long-term dependencies and transmit them for longer periods [15, 16]. The LSTM unit takes as inputs the values at time t, x_t , the previous hidden state h_{t-1} and the previous cell state (c_{t_1}), and passes them through three gates to filter, retain and transmit important information. First, the data is passed through a *forget gate* which computes a bit tensor (using a sigmoid function) to decide what elements of the input are relevant to be retained and passed to the *write gate*. The forget or keep gate can be denoted as 2:

$$f_i^t = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^t + \sum_j W_{i,j}^f h_j^{t-1}\right) \tag{2}$$

where b^f, U^f and W^f are the *forget gate*'s bias, gain and weights respectively. The *write gate* receives f_i^t and decides what information is going to be used to write the current cell memory state c_t. This is done using the $Tanh$ non-linearity and then approximating the result into a bit tensor using a sigmoidal function (σ), the process is expressed in equation 3.

$$g_i^t = \sigma\left(b_i^g + \sum_i U_{i,j}^g x_j^t + \sum_i W_{i,j}^g h_j^{t-1}\right) \tag{3}$$

The cell memory state c_t will serve as an input for the next timestamp, it is formed by the addition of the 2 and 3 states as follows:

$$c_i^t = f_i^t \cdot c_i^{t-1} + g_i^t \sigma(b_i \sum_j U_{i,j} x_j^t + \sum_j w_{i,j} h_i^{t-1}) \tag{4}$$

In addition, the current hidden state h_t will serve as input for the next unit, it is obtained by passing c_t through the *output gate* which consists in a $tanh$ activation function multiplied by a sigmoid function. The equation describing this process is the following:

$$h_i^t = tanh(c_i^t)\sigma(b_i^\circ + \sum_j U_{i,j}^\circ x_j^t + \sum_j W_{i,j}^\circ h_j^{t-1}) \quad (5)$$

Thus, by updating the memory cell and hidden states the LSTM can forecast the output variables $y = (y_1, y_2, ...y_t)$.

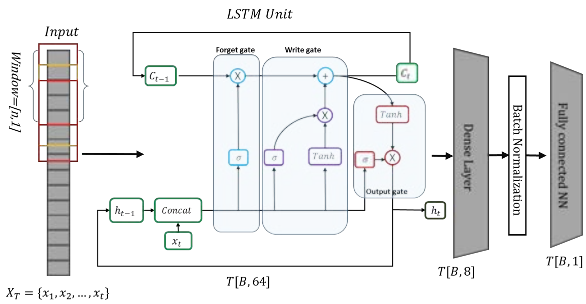


Fig. 1: Architecture of the LSTM unit and network

The proposed LSTM architecture consist of an input layer that receives a data sequence of dimension [n,1] , which passes through a LSTM layer of 64 neurons, 1 dense layer with ReLU activation function, 1 drop out layer of 0.2, 1 dense layer with linear activation. The model is trained using ADAM with a learning rate of 0.001 and RMSE as metric for the loss function. The figure 1 illustrates the LSTM architecture and process.

2.3 1D-CNN-LSTM

CNNs are widely used in Machine Learning for a variety of tasks such as image processing or language recognition, its use in combination with LSTMs has been widely addressed [17–23] . The term *convolution* comes from the usage of kernels or filters to extract intrinsic characteristics from the input data, also known as features. These kernels convolve along the data set in different directions, thus in the 1D-CNN the kernel moves only in one direction, which is the time axis for the time series case. When combined with an LSTM, the 1D-CNN layer would pass the filtered features to the LSTM units to generate the prediction [24, 25]. The architecture proposed consists in one convolutional layer with 32 filters, a kernel size of 3 and activation function ReLU, followed by a batch normalization and then an LSTM unit of similar architecture as the one presented in the previous section. The diagram in figure 2 shows this layer arrangement, where the input shape is two dimensional since only requires the samples and width of the data set.

2.4 Multivariate LSTM and 1D-CNN-LSTM

The training data set for this application is expanded to included the 3 more variables, namely ambient temperature which contains hourly temperature readings in Celsius degree; air density which contains hourly readings of air density at ground level in kg/m^3; and cloud cover fraction in a [0,1] scale. This last variable is valid as is has been shown there is

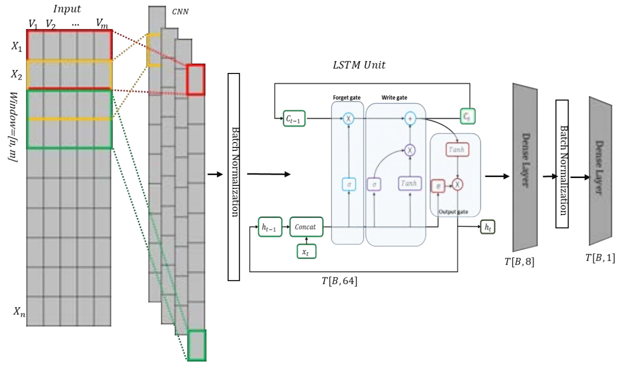


Fig. 2: Architecture of a multivariate 1D-CNN-LSTM Neural Network

a correlation between the cloud cover degree and wind speed [26]. Both architecture will process the input data to mine the information from the variables and their correlation with the output to adjust the weights of the numerical model. Thus, both the multivariate LSTM and the multivariate 1D-CNN-LSTM can produce forecasts for each of the variables if decided. The architectures are the same as the univariate models but the input layer is modified to process the new data set dimension.

3 Methodology

3.1 Dataset

Weather and wind speed data were obtained from global reanalysis models provided by the NASA [27] and extracted using an open source wind power simulator [28]. Ten years of hourly wind speed, ambient temperature at 2m above ground, air density at ground level and cloud cover fraction where collected from the geographical coordinate (54.9889, 2.22778), which is a location near the Dogger Bank A offshore wind farm off the east coast of the UK. Therefore the data set contains $87,600$ wind speed hourly samples which, together with the rest of the variables, sums up to a total of $438x10^3$ hourly samples. For the purposes of this investigation, the data set is divided in three subsets, 7 years are used for training the models, 2 years are for validation and 1 year for testing the models. The ANN based models are build using keras with TensorFlow 2.9, and the experiments were performed using a GPU NVIDIA Tesla V100, and the ARMA (p,q) model and the proper time series analysis is generated using Matlab 2022b.

3.2 time series pre-processing for ARMA

Given the nature of wind speed time series it is unlikely the dataset is stationary, hence in this paper the *Augmented Dickey Fuller* (ADF) and *Kwiatkowski-Phillips-Schmidt-Shin* (KPSS) unit root tests [29] are used to test stationarity of the dataset time-series. The ADF tests for the presence of a unit root, thus the time series is stationary (or can be if a differentiation is applied)if $p - value < 0.05$. On the other hand, the KPSS tests for the absence of a unit root, meaning the rejection of the null hypothesis suggest the time series is not stationary.

The outcome of both tests applied to the dataset is summarized in the table 1. Based on these observations, the KPSS statistic fall outside the critical values, thus the null hypothesis is rejected indicating the time series is not trend-stationary. Since the p-value for the ADF is smaller than 0.05, it is possible to reject the null hypothesis and said there is strong evidence the time series is difference stationary. Following a Box-Jenkins methodology, an order 1 differentiation was applied to the time series. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) can work as stationarity indicators as well, the figure 3 show the stem plot of the wind speed with a 1st degree differentiation, it can be observed both function gradually decrease until fall inside the significance threshold, this indicates the time series is stationary.

Based on the ACF/PACF plots given in Fig. 3 applied to the Dogger Bank dataset an ARMA(4,4) was chosen. The model was trained, validated and then used to generate sequences of one-hour ahead forecasts for 1 year.

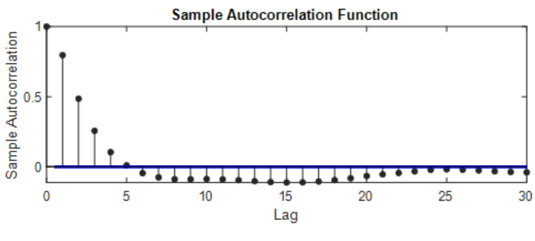
3.3 Data Pre-Processing for ANN based models

The performance of the model is greatly affected by the pre-process of the data since the raw data usually contains outlier, missing values or different scales in the case of multiple variables. There are two types of data transformations commonly used in ANNs, the MinMax normalization and the standardization. For a time series X the Min-Max scale the data between 0 to 1 as shown in equation 6.

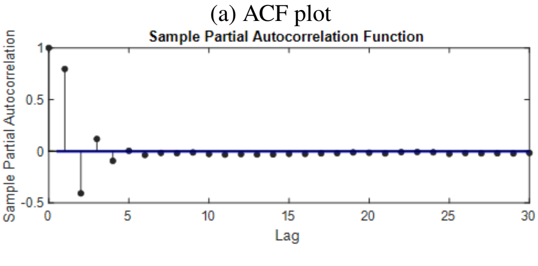
$$x_{norm} = \frac{x - x_{min}}{x - x_{max}} \tag{6}$$

Table 1 Statistical tests on the wind speed time series.

Test	p-value	τ-stat	cValue
ADF	0.001	-7.067	-1.9416
KPSS	0.01	9.502	0.146



(a) ACF plot



(b) PACF plot

Fig. 3: ACF and PACF plots for the differenced time series

where x_{min} and x_{max} are the minimum and maximum values in the time series respectively. The standardization fits the time-series data into a distribution of mean zero and standard deviation of one, $x \in N(0, 1)$ as per equation 7.

$$x_{std} = \frac{x - E[x]}{\sigma(x)} \tag{7}$$

This study discusses the accuracy and forecasting error obtained from applying a determined pre-processing technique to each ANN model architecture and the proposed ARMA(p,q) model. The use of pre-process in the ARMA(p,q) would help to reduce computational processing time of the model more than the improvement in accuracy since the modeling approach mentioned in the previous section already contemplates cleaning and transformation of the data.

3.4 Ablation Studies

The selection of the ANN architecture with the highest accuracy is approached as an optimisation problem where the objective is to minimize the forecast error, thus, let $\hat{f}_{error}$ be the difference between the real data and the predicted data from an architecture characterised by h which is the set of hyper-parameters in H denoting a hyper-parameter space.

$$\hat{f}_{error}(h) = \min_{h \in H} f_{error}(h) \tag{8}$$

The set of hyper-parameters used are a combination of a type of data pre-process with a technique to avoid overfitting, the pre-processing method applied are Min-Max normalization and standardization; and the techniques to avoid overfitting are the batch normalization and dropout regularization.

3.5 Performance comparison and statistical significance tests

In this paper we have used a modified Diebold-Mariano test first proposed by Harvey, Leybourne and Newbold (HLN) in 1993 to compare the different forecast models [30]. The DM test works on the grounds of the model residuals and states that such values are stationary and unbiased. Let $f(t)$ and $g(t)$ be two different forecast models with errors $e_f(t) = (y(t) - \hat{y}(t)_f)^2$ and $e_g(t) = (y(t) - \hat{y}(t)_g)^2$ for all t in the set T which is the set of all timesteps. We define the *loss differential function* $d(t) = e_f(t) - e_g(t)$ for all t and $\hat{d} = \mu = E[d]$ which is related to the mean absolute error. Both values are used to estimate the autocoraviance at lag k. We define that for any time series of length $n > k > 1$,

$$\gamma_k = \frac{1}{n} \sum_{i=k+1}^{n} (d_i - \hat{d})(d_{i-k} - \hat{d}) \tag{9}$$

Hence, for a forecast horizon $h \geq 1$, the DM statistic is defined as follows,

$$DM = \frac{\hat{d}}{\sqrt{[\gamma_0 + 2\sum_{k=1}^{h-1} \gamma_k](\frac{1}{n})}} \tag{10}$$

Therefore, it does make sense to state the null hyphotesis as $H_0 : E(d_t) = 0 \quad \forall t \in T$ meaning that the two forecasts have the same accuracy, whereas the alternative hypothesis $H_1 : E(d_t) \neq 0$ would indicate the two models have different levels of accuracy. Under the assumption of null stationarity, the DM tends to a Normal distributions so it is safe to find the critical values using a t-statistic distribution where the boundaries are $\pm z_\alpha/2$ and the p-value is estimated as well.

3.6 Evaluation metrics

The indicators used in this paper are the RMSE, MSE, MAE and MAPE, and are defined by the following equations ($\forall t \in T$):

$$MSE = \frac{\sum_{t=1}^{n}(y(t) - \hat{y}(t))^2}{n} \tag{11}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{t=1}^{n}(y(t) - \hat{y}(t))^2}{n}} \tag{12}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y(t) - \hat{y}(t)| \tag{13}$$

The MAPE is a measure of quality of the prediction and it is useful since it scales the relative error units into percentage. However even though this metric provides a way to measure the accuracy of a model it is sensible to values close to zero, therefore, it is used to support the evaluation of the model but the selection of an architecture or model is not entirely based on this metric.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|y(t) - \hat{y}(t)|}{|y(t)|} \tag{14}$$

4 Results and discussions

4.1 Architecture selection

The ablation experiments where done for predictions 12 hours ahead for 1 year sequence, the table 2 contains the metrics obtained from each prediction where the values in bold belongs to the selected architectures. Initially, tests in the ARMA(4,4) consisted in measuring the error when the data was pre processed using either a standardization or a Min-Max normalization. The error from each forecasts differ by 0.5% , being the Min-Max Normalization data treatment with better MAPE. Models based on ANN employ techniques known as batch normalization and Dropout regularization to reduce the risk to overfitting. In the table can be observed that both univariate and multivariate LSTM architectures including a batch normalization and a Min-Max normalization data pre process are the ones with the smallest error. Whereas, the batch normalization with a standardization pre process showed to be the most suitable arrangement for the 1D-CNN-LSTM univariate and multivariate models.

4.2 Prediction accuracy comparison

Once selected the architecture and pre-process most suitable for each model,different sequences are generated to measure the accuracy decay in the predictions. A similar comparative experiment as the previous section is performed for the 12, 24 and 48 hours forecast horizons and 1 and 2 year sequences, the results are reported in the table 3. Both multivariate models forecast errors are bigger than their univariate counterparts, even for longer forecast horizons, this behaviour can be attributed to the ANNs capacity to handle bigger data sets. The hybrid 1D-CNN-LSTM produced the least errors for all the time horizons and both sequence lengths,this shows the addition of the CNN layer can indeed improve the predictions. Finally, the ARMA(4,4) model output error for horizons of 12 hours for both 1 and 2 year sequences still fall into the acceptable, but this is not the case for larger horizons.

Table 2 Statistical metrics from different architectures and pre-process applied.

Metric	Standardization Batch normalization	Standardization drop-out 0.2	Min-Max Normalization Batch normalization	Min-Max Normalization drop-out 0.2
ARMA(4,4)				
RMSE	2.925		**2.910**	
MAE	2.271		**2.257**	
MAPE	0.285		**0.281**	
LSTM				
RMSE	0.275	0.487	**0.274**	0.505
MAE	0.183	0.381	**0.177**	0.405
MAPE	0.022	0.050	**0.021**	0.055
1D -CNN-LSTM				
RMSE	**0.272**	0.393	0.324	0.395
MAE	0**0.176**	0.294	0.232	0.293
MAPE	**0.02**	0.040	0.025	0.036
Multivariate LSTM				
RMSE	2.923	2.963	**0.439**	2.037
MAE	2.134	2.256	**0.330**	1.593
MAPE	0.345	0.356	**0.044**	0.217
Multivariate 1D-CNN-LSTM				
RMSE	**0.280**	0.784	0.454	1.729
MAE	**0.185**	0.618	0.327	1.363
MAPE	**0.021**	0.084	0.038	0.198

4.3 Significance tests, Diebold-Mariano

For practical purposes, the DM statistic was calculated for the univariate 1D-CNN-LSTM forecast against the ARMA(4,4), univariate LSTM and multivariate models, results are reported in table 4. In general, DM tests for the null hypothesis H_0 of the two methods have the same accuracy, and the alternative hypothesis H_1 is that the two forecasts have different levels of accuracy. Therefore, in the table 4 can be seen that for almost all cases the H_0 hypothesis can be rejected in favor of the alternative, meaning there is strong evidence the hybrid univariate 1D-CNN-LSTM model outperforms in the majority of cases.

5 Conclusion

The performance evaluation of four ANNs based models for wind speed forecasting was presented. The univariate hybrid 1D-CNN-LSTM model yielded the lowest error and higher accuracy for all prediction horizons closely followed by the univariate LSTM, which based on the DM test observations,

Table 3 Statistical metrics from different time horizons and pre-process applied.

	1 year			2 years		
Metrics	12h	24h	48h	12h	24h	48h
ARMA(4,4)						
RMSE	2.911	3.483	3.741	2.909	3.415	3.603
MAE	2.258	2.748	2.979	2.228	2.677	2.845
MAPE	0.281	0.351	0.383	0.275	0.341	0.365
LSTM						
RMSE	0.274	0.516	0.274	0.279	0.988	0.282
MAE	0.177	0.457	0.170	0.179	0.739	0.176
MAPE	0.021	0.054	0.019	0.021	0.086	0.019
1D -CNN-LSTM						
RMSE	**0.272**	**0.281**	**0.269**	**0.275**	**0.946**	**0.277**
MAE	**0.176**	**0.181**	**0.171**	**0.174**	**0.686**	**0.177**
MAPE	**0.020**	**0.020**	**0.019**	**0.020**	**0.79**	**0.019**
Multivariate LSTM						
RMSE	0.439	0.822	2.926	0.911	1.124	1.241
MAE	0.330	0.620	2.147	0.685	0.842	1.150
MAPE	0.044	0.086	0.347	0.091	0.103	0.121
Multivariate 1D-CNN-LSTM						
RMSE	0.280	0.281	0.283	0.318	0.947	0.294
MAE	0.185	0.185	0.080	0.225	0.681	0.196
MAPE	0.021	0.021	0.022	0.025	0.079	0.022

Table 4 P-values for each DM statistic for model predictions

Model	1 year		2 years	
	DM $statistic$	p $-value$	DM $statistic$	p $-value$
12 h-steps-ahead				
1DCNNLSTM vs ARMA	-19.014	< 0.01	-25.0455	< 0.01
1DCNNLSTM vs LSTM	-0.733	0.463	-1.527	0.126
1DCNNLSTM vs M-LSTM	-16.712	< 0.01	-17.313	< 0.01
1DCNNLSTM vs M-1DCNNLSTM	-3.717	< 0.01	-14.921	< 0.01
24 h-steps-ahead				
1DCNNLSTM vs ARMA	-16.264	< 0.01	-21.136	< 0.01
1DCNNLSTM vs LSTM	-65.986	< 0.01	-10.172	< 0.01
1DCNNLSTM vs M-LSTM	-9.872	< 0.01	-9.702	< 0.01
1DCNNLSTM vs M-1DCNNLSTM	-0.207	0.834	-0.004	0.996
48 h-steps-ahead				
1DCNNLSTM vs ARMA	-13.243	< 0.01	-18.067	< 0.01
1DCNNLSTM vs LSTM	-2.249	0.0244	-2.404	0.0162
1DCNNLSTM vs M-LSTM	-12.470	< 0.01	-46.749	< 0.01
1DCNNLSTM vs M-1DCNNLSTM	-4.769	< 0.01	-7.111	< 0.01

both models hold equal accuracy in certain scenarios. Nevertheless, taking into account the combination of the statistical metrics and significance test outcome, it can be said the univariate 1D-CNN-LSTM model outperformed among the proposed wind speed forecasting models. Furthermore, the election of this particular architecture and modeling technique is based on the performance evaluation of different modelling set up combinations, which consisted in using a determined pre-processing (classic standardization or a Min-Max Normalization) and architectures including either a batch normalization or a drop-out regularization. Contrary to the expectations about any multivariate model performing better than their univariate counterparts, it was observed both univariate models produced better forecasts. This can be attributed to the fact both multivariate models are shallow or lack of capability to properly handle the size of the data set. Nonetheless, all set up combinations for the multivariate proposed architectures performed better than the ARMA(4,4) in all prediction scenarios. Future work will focus on the incorporation of variables correlation index in the data set construction and the integration of more hidden layers to increase the multivariate models depth with the expectation of sharpening their forecast accuracy. Additionally, the statistical performance comparison can be applied in wind power and energy forecasting, including a sensitivity analysis as a measure of the uncertainty in the elected model.

6 Acknowledgements

This work has been supported by the Mexican Council of Science and Technology (CONACyT).

References

[1] S. Hanifi, X. Liu, Z. Lin, and S. Lotfian, "A Critical Review of Wind Power Forecasting Methods-Past, Present and Future," 2020.

[2] S. H. Wu and Y. K. Wu, "Probabilistic wind power forecasts considering different NWP models," *Proceedings - 2020 International Symposium on Computer, Consumer and Control, IS3C 2020*, pp. 428–431, nov 2020.

[3] C. D. Dao, B. Kazemtabrizi, C. J. Crabtree, and P. J. Tavner, "Integrated condition-based maintenance modelling and optimisation for offshore wind turbines," *Wind Energy*, no. September 2020, pp. 1–19, 2021.

[4] S. A. Kalogirou, "Artificial neural networks in renewable energy systems applications: a review," *Renewable and Sustainable Energy Reviews*, vol. 5, pp. 373–401, dec 2001.

[5] "Accurate photovoltaic power forecasting models using deep LSTM-RNN," *Neural Computing and Applications*, vol. 31, no. 7, pp. 2727–2740, 2019.

[6] Q. Chen and K. A. Folly, "Wind Power Forecasting," *IFAC-PapersOnLine*, vol. 51, pp. 414–419, jan 2018.

[7] E. Yatiyana, S. Rajakaruna, and A. Ghosh, "Wind speed and direction forecasting for wind power generation using ARIMA model," *2017 Australasian Universities Power Engineering Conference, AUPEC 2017*, vol. 2017-November, pp. 1–6, feb 2018.

[8] M. Joudaki, "Long-Term Wind Speed and Power Forecasting Based on LSTM: A Comprehensive Study," in *2022 9th Iranian Conference on Renewable Energy & Distributed Generation (ICREDG)*, (Iran), pp. 1–4, IEEE, feb 2022.

[9] M. Bou-Rabee, K. A. Lodi, M. Ali, M. F. Ansari, M. Tariq, and S. A. Sulaiman, "One-month-ahead wind speed forecasting using hybrid AI model for coastal locations," *IEEE Access*, vol. 8, pp. 198482–198493, 2020.

[10] M. E. Cruz-Victorio, B. Kazemtabrizi, and M. Shahbazi, "Statistical evaluation of wind speed forecast models for microgrid distributed control," *IET Smart Grid*, jun 2022.

[11] D. R. Cox, "STATISTICAL SIGNIFICANCE TESTS," *J. clin. Pharmac*, vol. 14, pp. 325–331, 1982.

[12] A. Babikir and H. Mwambi, “A factor-Artificial neural network model for time series forecasting: The case of South Africa,” in *2014 International Joint Conference on Neural Networks (IJCNN)*, IEEE, july 2014.

[13] M. E. C. Victorio, B. Kazemtabrizi, and M. Shahbazi, “Price Forecast Methodologies Comparison for Microgrid Control with Multi-Agent Systems,” in *2021 IEEE Madrid PowerTech, PowerTech 2021 - Conference Proceedings*, Institute of Electrical and Electronics Engineers Inc., jun 2021.

[14] G. E. Box, G. M. Jenkins, and G. Reinsel, “Time series analysis: forecasting and control holden-day san francisco,” *BoxTime Series Analysis: Forecasting and Control Holden Day1970*, 1970.

[15] F. Shahid, A. Zameer, and M. Muneeb, “A novel genetic LSTM model for wind power forecast,” *Energy*, vol. 223, p. 120069, 2021.

[16] W. Zhang, Z. Lin, and X. Liu, “Short-term offshore wind power forecasting - A hybrid model based on Discrete Wavelet Transform (DWT), Seasonal Autoregressive Integrated Moving Average (SARIMA), and deep-learning-based Long Short-Term Memory (LSTM),” *Renewable Energy*, vol. 185, pp. 611–628, feb 2022.

[17] Z. Olaofe, “Assessment of LSTM, Conv2D and ConvLSTM2D Prediction Models for Long-Term Wind Speed and Direction Regression Analysis,” 2021.

[18] I. Koprinska, D. Wu, and Z. Wang, “Convolutional Neural Networks for Energy Time Series Forecasting,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 2018-July, oct 2018.

[19] K. Nishikawa, R. Hirakawa, H. Kawano, K. Nakashi, and Y. Nakatoh, “Detecting System Alzheimer’s Dementia by 1d CNN-LSTM in Japanese Speech,” in *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*, vol. 2021-January, pp. 1–3, IEEE, jan 2021.

[20] L. Xiao, L. Zhang, F. Niu, X. Su, and W. Song, “Remaining useful life prediction of wind turbine generator based on 1D-CNN and Bi-LSTM,” *International Journal of Fatigue*, vol. 163, p. 107051, oct 2022.

[21] A. S. Hati, “Convolutional neural network-long short term memory optimization for accurate prediction of airflow in a ventilation system,” *Expert Systems With Applications*, vol. 195, no. February, p. 116618, 2022.

[22] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2015-August, pp. 4580–4584, IEEE, apr 2015.

[23] A. K. Ozcanli and M. Baysal, “Islanding detection in microgrid using deep learning based on 1D CNN and CNN-LSTM networks,” *Sustainable Energy, Grids and Networks*, vol. 32, p. 100839, dec 2022.

[24] A. P. Sari, H. Suzuki, T. Kitajima, T. Yasuno, D. A. Prasetya, and N. Nachrowie, “Prediction model of wind speed and direction using convolutional neural network - Long short term memory,” *PECon 2020 - 2020 IEEE International Conference on Power and Energy*, pp. 356–361, dec 2020.

[25] H. Zhang, L. Zhao, and Z. Du, “Wind power prediction based on CNN-LSTM,” in *2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2)*, pp. 3097–3102, IEEE, oct 2021.

[26] Z. M. Abbood, O. T. Al-Taai, and W. G. Nassif, “Impact of wind speed and direction on low cloud cover over Baghdad city,” *Current Applied Science and Technology*, vol. 21, no. 3, pp. 590–600, 2021.

[27] M. M. Rienecker, M. J. Suarez, R. Gelaro, R. Todling, J. Bacmeister, E. Liu, M. G. Bosilovich, S. D. Schubert, L. Takacs, G.-K. Kim, S. Bloom, J. Chen, D. Collins, A. Conaty, A. da Silva, W. Gu, J. Joiner, R. D. Koster, R. Lucchesi, A. Molod, T. Owens, S. Pawson, P. Pegion, C. R. Redder, R. Reichle, F. R. Robertson, A. G. Ruddick, M. Sienkiewicz, and J. Woollen, “MERRA: NASA’s Modern-Era Retrospective Analysis for Research and Applications,” *Journal of Climate*, vol. 24, pp. 3624–3648, jul 2011.

[28] I. Staffell and S. Pfenninger, “Using bias-corrected reanalysis to simulate current and future wind power output,” *Energy*, vol. 114, pp. 1224–1239, nov 2016.

[29] S. E. SAID and D. A. DICKEY, “Testing for unit roots in autoregressive-moving average models of unknown order,” *Biometrika*, vol. 71, no. 3, pp. 599–607, 1984.

[30] F. X. Diebold and R. S. Mariano, “Comparing Predictive Accuracy,”