Comparison of Recurrent Neural Networks for Wind Power Forecasting

Erick López[1(✉)], Carlos Valle[2], Héctor Allende-Cid[3], and Héctor Allende[1]

[1] Departamento de Informática, Universidad Técnica Federico Santa María, Valparaíso, Chile
{elopez,hallende}@inf.utfsm.cl
[2] Departamento de Computación e Informática, Universidad de Playa Ancha, Valparaíso, Chile
carlos.valle@upla.cl
[3] Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
hector.allende@pucv.cl

Abstract. Integrating wind power to the electrical grid is complicated due to the stochastic nature of the wind, which makes its prediction a challenging task. Then, it is important to devise forecasting tools to support this task. For example, a network that integrates an Echo State Network architecture and Long Short-Term Memory blocks as hidden units (ESN+LSTM) has been proposed, showing good performance against a physical model. This paper proposes to compare this network versus Echo State Network (ESN) and Long Short-Term Memory (LSTM), to forecast wind power from 1 to 24 h ahead. Results show than the ESN+LSTM model outperforms the performance reached for ESN and LSTM, in terms of MSE, MAE, and the metrics used in the Taylor diagram. In addition, we observe that the advantage of this network is statistically significant during the first moments of the forecast horizon, in terms of T-test and Wilcoxon-test.

Keywords: Wind power forecasting · Recurrent Neural Networks · Echo State Network · Long Short-Term Memory · Multivariate time series

1 Introduction

One of the current challenges in the world is the integration of Non-Conventional Renewable Energy (NCRE) sources into the global energy matrix. Among these sources, wind energy presents great challenges for its integration, where one of its critical factors is its stochastic nature. In this context, it is necessary to have different forecasting tools that allow us to make better schedule of the different sources that make up an electrical matrix, such that it allows us to support the operational and economic tasks of the system [9]. Among the models

K. M. Figueroa Mora et al. (Eds.): MCPR 2020, LNCS 12088, pp. 25–34, 2020.
https://doi.org/10.1007/978-3-030-49076-8_3

proposed in the literature, Recurrent Neural Networks (RNNs) have reported good performance in wind energy forecasting.

In particular, the ESN+LSTM recurrent neural network proposed in [12] has showed good performance in this task. Therefore, in this paper, we compare the ESN+LSTM model against its base models: LSTM and ESN.

To evaluate these models, we use a dataset from a wind farm in NorthEast Denmark. The time series to model is formed by wind speed, wind direction, temperature, month, day, hour, and wind power. Standardized metrics will be used, measuring cumulative performance, because we will address the multi-ahead step forecasting problem. In addition, the Taylor diagram is used to draw some conclusions about the performance of the models, as well as a parametric and non-parametric test to validate some results.

The rest of the paper is organized as follows. In Sect. 2 we describe the context of wind power problem. Section 3 we describe briefly the recurrent networks that will be compared. Next, we describe the experimental setting on which we tested the models and we review the results. Finally, the last section is devoted to conclusions and future work.

2 Wind Power Generation

The wind power generation is the result of transforming the kinetic energy of the wind into electrical energy, traditionally by rotating turbine blades. The power output from a wind turbine can be calculated by the following equation:

$$P = \left(\frac{1}{2}\rho\pi R^2 v_1^3\right) \cdot C_p, \tag{1}$$

where ρ is the density of the air, R is the length of blades plus the rotor radius, v_1 is wind speed that enters the turbine, and C_p is a coefficient of power provided by the manufacturer, with a theoretical upper limit of 16/27, known as Betz limit. Note that the power output is proportional to the wind speed, which it has a stochastic nature, depending on different meteorological factors, hindering its integration into the existing electricity supply system [9].

In the literature, there are different proposals to address wind power forecasting, being possible group by in four categories [3,9]: (i) persistence method, that simply replicates the last recorded value to make the forecast; (ii) physical methods that takes a detailed description of the physical conditions of the wind farm (including turbines, terrain geography, and meteorological conditions) to model the wind power by means of differential equations and downscaling techniques [10,13]; (iii) statistical methods that try to exploit the possible underlying dependency structure of data, under certain assumptions, by time series modeling techniques [2,11]; (iv) machine learning methods that attempt to discover underlying relationships of dataset, without an a priori structural hypothesis [5,14].

The last category has been received special interest nowadays, achieving great performance in different tasks. Particularly in wind power forecasting, the recurrent neural network models stand out from other machine learning methods since it would allow modeling time series in a natural way.

3 Forecasting Models

In this section, we present three approaches of recurrent neural networks, which aim to solve the vanishing gradient problem, and have been reported good performances in time series modeling. Further, we use these models for experimental comparisons.

Long Short-Term Memory (LSTM)

In [6] is proposed a class of recurrent network replacing the basic unit (neuron) of a traditional network by a *block of memory*. This block contains one or more *memory cells*. Each memory cell is associated with "gates" (activation functions) for controlling the information flow moving through the cells. Each auto-connected memory cell is so-called "Constant Error Carousel" (CEC) linear unit, whose activation is the state of the cell as shown in Fig. 1. The CEC solves the problem of vanishing (or explosion) gradient [1]. Since the local error back flow remains constant within the CEC, without growing or decreasing, while not a new entry or external signal error appears. However, its training process can be computationally expensive, due to the complexity of its architecture. Besides, it might overfit depending on the values of its hyperparameters such as the number of blocks, the learning rate or the maximum number of epochs.

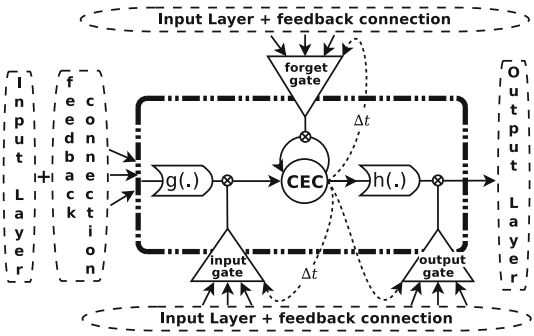

Fig. 1. LSTM architecture with 1 block and 1 cell.

Echo State Network (ESN)

Another RNN that has performed well in time series forecasting is the model proposed by Jaeger [8]. This model is very simple and easy to implement, consists of three layers (input, hidden and output), where the hidden layer is formed by a

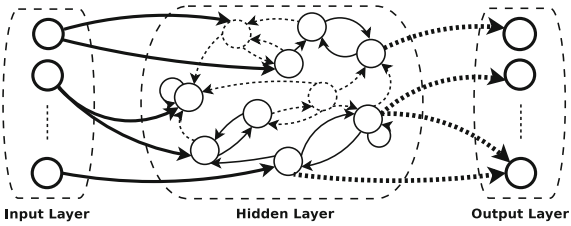

Fig. 2. ESN topology.

large number of perceptron kind neurons, with a low rate of connectivity among them (allowing self-connections), and they are randomly connected as depicted in Fig. 2. An interesting property is all weights are initialized randomly (usually using a normal o uniform centred on zero). Next, it rescales the recurrent weight matrix to get a spectral radius close to one. Finally, only the output layer weights are fixed using a ridge-regression. The output hidden neurons is computed by the following expression,

$$s(t) = (1 - a) \cdot s(t - 1) + a \cdot f(\text{net}(t)), \tag{2}$$

where $s(t)$ is the output of a hidden neuron in the instant t, $a \in [0, 1]$ is a leaking rate that regulates the speed update of the internal dynamics, i.e., it is adjusted to match the speed of the dynamics of $x(t)$ and $\hat{y}(t)$. Here, $x(t)$ is the input to the network and $\hat{y}(t)$ is the output of the network at time t. Moreover, $f(\cdot)$ is a hyperbolic tangent activation function and $\text{net}(t)$ is the input signal to the neuron. Furthermore, as the hidden states are initialized to zero $s(0) = 0$, it is necessary to define the number of steps θ that the recurrent states are updated without being considered in the process of adjusting the output layer. The above is because the states are initialized to zero, $s(0) = 0$.

Echo State Network with Long Short-Term Memory (ESN+LSTM) Given the advantages and some limitations identified on LSTM and ESN models, ESN+LSTM [12] is proposed to integrate the architecture of an ESN with LSTM units as hidden neurons (see Fig. 3). This proposal permits to train all network weights through the following strategy: i) The input and hidden layer is trained by an online gradient descent (OGD) algorithm with one epoch, using as target the input signal; ii) Next, the output layer is adjusted with a regularized quantile regression, using as target the desired output; iii) Finally, the whole network is trained with an OGD algorithm with one epoch and the desired target.

The first step aims to extract characteristics automatically as the autoencoder approach. The second step aims to use a quantile regression in order to obtain a robust estimate of the expected target. It should be noted that the hidden layer is sparsely connected, and its weights matrix keep a spectral radius close to one. In this model, the main hyperparameters to be tune are the hidden units number, the spectral radius, and the regularization parameter.

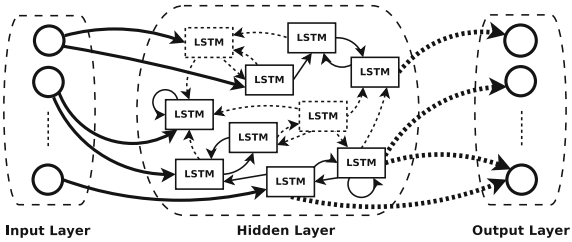

Fig. 3. ESN+LSTM topology.

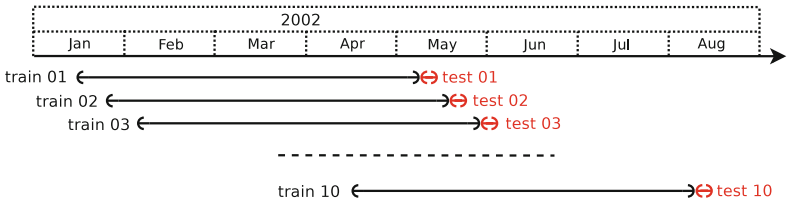

Fig. 4. Time series split scheme for cross-validation approach

4 Experiments and Results

For assessing the presented models, we use a dataset from the Klim Fjordholme wind farm (57.06°N, 9.15°E) [7], that consists of generated power measurements and predictions of some meteorological variables (wind speed, wind direction and ambient temperature).

We work with hourly time series, no missing values. The dataset is composed of 5376 observations, starting at 00:00 on 14 January 2002 to 23:00 on 25 August 2002. The attributes considered to model are: wind speed, wind direction, temperature, month, day, hour and wind power. All features are normalized to the $[-1, 1]$ range using the min-max function and predictions are denormalized before computing the performance metrics.

A cross-validation approach is used to train and select the best hyperparameters configuration. The time series is divided into $R = 10$ subseries, and the performance of each model is evaluated in each of them (see Fig. 4).

To evaluate the performance of the models, we use some standardized metrics: Mean Squared Error (MSE) and the Mean Absolute Error (MAE). Additionally, we show a Taylor diagram to compare graphically the Pearson correlation coefficient (ρ), the root-mean-square error (RMSE), and the standard deviation (SD). In this work, the performance will be checked over multi-step ahead forecasting, which is generated using the multi-stage approach [4].

For a single subseries r, the different metrics are based on the error $e_r(\cdot)$ at h hours ahead, defined for equation (3),

$$e_r(T + h|T) = y_r(T + h) - \hat{y}_r(T + h|T), \tag{3}$$

Table 1. Parameters for tuning.

LSTM	
Hidden layer size	: $J \in \{10, 20, \ldots, 100, 110, 120, \ldots, 500\}$
Number of epochs	: $\texttt{epoch} \in \{1, 10, 50, 100, 150, 200\}$
ESN	
Hidden layer size	: $J \in \{10, 20, \ldots, 100, 110, 120, \ldots, 500\}$
Leaking rate	: $a \in \{0.1\,,\ 0.2\,,\ 0.3\,,\ \ldots\,,\ 0.8\,,\ 0.9\,,\ 1\}$
Spectral radius	: $\alpha \in \{0.1\,,\ 0.2\,,\ 0.3\,,\ \ldots\,,\ 0.8\,,\ 0.9\,,\ 1\}$
Regularization coeff	: $\lambda \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$
ESN+LSTM	
Hidden layer size	: $J \in \{10, 20, \ldots, 100, 110, 120, \ldots, 500\}$
Spectral radius	: $\alpha \in \{0.1\,,\ 0.2\,,\ 0.3\,,\ \ldots\,,\ 0.8\,,\ 0.9\,,\ 1\}$
Regularization coeff	: $\lambda \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$

where $y_r(T+h)$ is the desired output at instant $T+h$ of subseries r, T is the index of the last point of the series used during training, h is the number of steps ahead, and $\hat{y}_r(T+h|T)$ is the estimated output at time $T+h$ generated by the model for subserie r. Then, the metrics are calculated by the following equations,

$$\text{MSE}_{(r)} = \frac{1}{H}\sum_{h=1}^{H}(e_r(T+h|T))^2, \qquad \text{MSE} = \frac{1}{R}\sum_{r=1}^{R}\text{MSE}_{(r)}, \tag{4}$$

$$\text{MAE}_{(r)} = \frac{1}{H}\sum_{h=1}^{H}|e_r(T+h|T)|, \qquad \text{MAE} = \frac{1}{R}\sum_{r=1}^{R}\text{MAE}_{(r)}, \tag{5}$$

where R is the total number subseries, and H is the ahead step limit used.

The parameters that will be tuning for the different networks are shown in Table 1. For the ESN and ESN+LSTM, first we tune the number of hidden units, keeping fixed $\alpha = 0.5$ and $\lambda = 0.001$. Next, α and λ are tuned. In three networks, the output layer uses the identity function as the activation function. In addition, the ESN and ESN+LSTM use direct connections from the input layer to the output layer. The weights of each matrix are generated from a uniform distribution $(-0.1;\ 0.1\ )$, independently of each other. Sparse arrays were also used when connecting the input layer with the hidden layer. Thus, configurations with the lowest error over the test set obtained for each model are: LSTM ($J = 30$, $\texttt{epoch} = 100$), ESN ($J = 470$, $a = 0.4$, $\alpha = 0.6$, $\lambda = 10^{-5}$), and ESN+LSTM ($J = 190$, $\alpha = 0.5$, $\lambda = 10^{-3}$).

Figure 5 shows the MSE for different values of H used. It can be seen that ESN+LSTM achieves a lower error, extending this advantage from $H = 1$ to $H = 12$, later its performance is similar to the one of obtained from the LSTM. We also view that ESN is the network that presents the greatest error to different

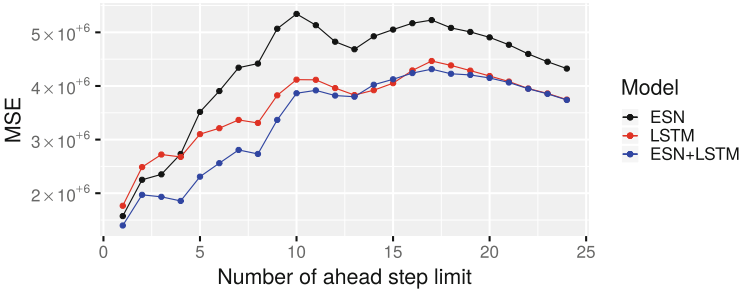

Fig. 5. MSE by each ahead step limit, H, used.

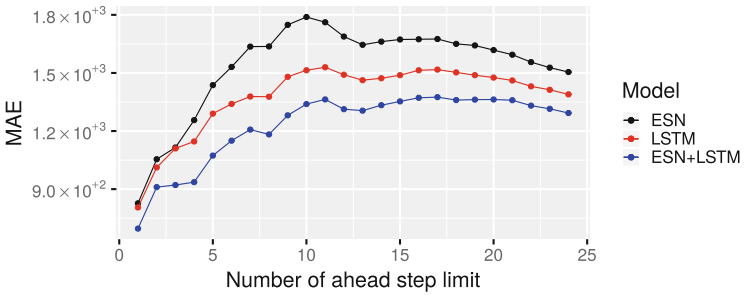

Fig. 6. MAE by each ahead step limit, H, used.

steps ahead. A similar scenario can be seen by observing the behavior of MAE (see Fig. 6). In this case, ESN+LSTM presents better performance for all H values. While ESN again presents the highest error, reaching a performance similar to LSTM during the first moments $H \in \{1, 2, 3\}$.

Although ESN+LSTM achieves the lowest error in terms of MSE and MAE for different H, this advantage may be due to the randomness of the system. Then, a hypothesis test for paired samples will be evaluated, since each model was tested with the same sub-series, to later build the global indicators.

$$\begin{array}{lllll} \text{Test 1} & H_0 : \mu_{esn} \leq \mu_{esn+lstm} & \text{vs} & H_1 : \mu_{esn} > \mu_{esn+lstm} \\ \text{Test 2} & H_0 : \mu_{lstm} \leq \mu_{esn+lstm} & \text{vs} & H_1 : \mu_{lstm} > \mu_{esn+lstm} \end{array}$$

For both cases, a parametric and nonparametric test will be used: t-test and wilcoxon-test. Table 2 and 3 show the p-values to validate the statistical significance if MSE of ESN+LSTM is lower than other models.

Using the t-test, it is observed that in the case of Test 1, ESN+LSTM is significantly lower than ESN from $H = 4$ to $H = 9$, for a significance level of 10%. Next, the advantage observed in Fig. 5 would not be significant. On another hand, in the Test 2, the advantage presented by ESN+LSTM over LSTM is significative from $H = 3$ to $H = 5$, using the same significance level. If we use the wilcoxon-test, the advantage shown by ESN+LSTM over ESN is significant from

Table 2. T-test's p-values by each ahead step limit used, for MSE. Best results at highlighted in gray background.

H	1	2	3	4	5	6	7	8	9	10	11	12
Test 1	0.2202	0.2455	0.1973	0.0563	0.0560	0.0642	0.0660	0.0659	0.0922	0.1249	0.1464	0.1634
Test 2	0.1645	0.1417	0.0722	0.0366	0.0636	0.1342	0.2103	0.2188	0.2906	0.3812	0.3966	0.4150
H	13	14	15	16	17	18	19	20	21	22	23	24
Test 1	0.1770	0.1748	0.1722	0.1701	0.1695	0.1824	0.1936	0.2031	0.2104	0.2192	0.2308	0.2256
Test 2	0.4794	0.5675	0.5483	0.4688	0.4111	0.4093	0.4504	0.4766	0.4882	0.4969	0.4945	0.4944

Table 3. Wilcoxon-test's p-values by each ahead step limit used, for MSE. Best results at highlighted in gray background.

H	1	2	3	4	5	6	7	8	9	10	11	12
Test 1	0.2768	0.3178	0.2768	0.0486	0.0963	0.0801	0.0527	0.0527	0.0801	0.0967	0.1611	0.1611
Test 2	0.0654	0.2158	0.0322	0.0137	0.0527	0.0801	0.1377	0.2158	0.2461	0.3125	0.2783	0.2461
H	13	14	15	16	17	18	19	20	21	22	23	24
Test 1	0.1875	0.1611	0.2461	0.2783	0.2158	0.2158	0.1875	0.1875	0.2783	0.3125	0.3125	0.2783
Test 2	0.3477	0.3477	0.2461	0.2461	0.2158	0.2461	0.2783	0.3477	0.3125	0.3477	0.3477	0.3477

Table 4. T-test's p-values by each ahead step limit used, for MAE. Best results at highlighted in gray background.

H	1	2	3	4	5	6	7	8	9	10	11	12
Test 1	0.1037	0.1748	0.1129	0.0399	0.0397	0.0469	0.0350	0.0343	0.0393	0.0454	0.0605	0.0649
Test 2	0.0684	0.0650	0.0077	0.0057	0.0235	0.0675	0.1223	0.1037	0.1085	0.1275	0.1260	0.0865
H	13	14	15	16	17	18	19	20	21	22	23	24
Test 1	0.0790	0.0822	0.0861	0.0960	0.0932	0.0939	0.0955	0.1107	0.1218	0.1223	0.1282	0.1216
Test 2	0.1153	0.1442	0.1540	0.1468	0.1536	0.1570	0.1825	0.2055	0.2268	0.2241	0.2197	0.2144

Table 5. Wilcoxon-test's p-values by each ahead step limit used, for MAE. Best results at highlighted in gray background.

H	1	2	3	4	5	6	7	8	9	10	11	12
Test 1	0.1181	0.2386	0.1432	0.0486	0.0776	0.0654	0.0322	0.0322	0.0322	0.0527	0.0801	0.0801
Test 2	0.0527	0.0801	0.0029	0.0098	0.0244	0.0801	0.0967	0.1377	0.1611	0.1611	0.1611	0.0801
H	13	14	15	16	17	18	19	20	21	22	23	24
Test 1	0.1377	0.0967	0.1377	0.1377	0.1377	0.1377	0.1377	0.1611	0.2158	0.1875	0.1611	0.1611
Test 2	0.1162	0.1162	0.1162	0.1377	0.1611	0.1875	0.1611	0.1875	0.1875	0.1875	0.2158	0.1875

$H = 4$ to $H = 10$. While comparing ESN+LSTM and LSTM, it is appreciated that better performance is achieved when $H = 1$ and from $H = 3$ to $H = 6$.

When using the MAE metric, the Table 4 shows the results of applying the t-test. We observe that ESN+LSTM is significatively lower than ESN from $H = 4$ to $H = 14$ for a significance level of 10%. While in Test 2, ESN+LSTM model presents a better performance from $H = 1$ to $H = 6$ to a significance level of

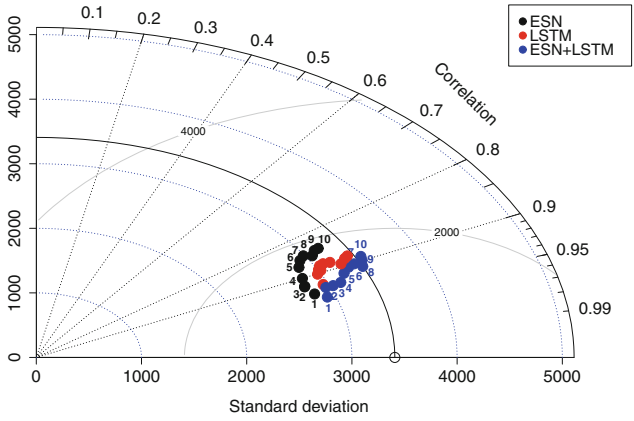

Fig. 7. Taylor Diagram. Each number close to point represents the ahead step limit used H.

10%. Finally, Table 5 shows the results using the wilcoxon-test for MAE. The ESN error improvement is presented when $H \in \{4, \ldots, 12\}$ and $H = 14$. And when $H \in \{1, \ldots, 7\}$, ESN+LSTM improves to LSTM with a 10% of significance.

Additionally, Fig. 7 shows the Taylor Diagram using only from $H = 1$ to $H = 10$. It is observed that ESN+LSTM achieves better performance based on correlation, RMSE, and SD, compared ESN and LSTM model for the same ahead step limit H. It is also appreciated that, as the forecast horizon increases, the correlation decreases, as does the RMSE increases, but the SD generated by the model is close to the real one.

5 Conclusions and Future Work

This work compares the performance of three recurrent neuronal networks for wind power forecasting task. The experimental results show that the ESN+LSTM model is able to capture the underlying dynamics and to predict several steps ahead better than ESN and LSTM models. Since the metrics used were defined cumulatively, it is expected that all three models will exhibit an increase in error as the forecast horizon increases. As our results show, the performance of the chosen networks begins to deteriorate as H grows. However, at least up to $H = 10$, ESN+LSTM performes better based on the Taylor diagram. Besides, according to t-test over MAE results, ESN+LSTM outperformed ESN during the first 19 steps ahead (except for $H = 1, 2, 3$) and outperformed LSTM during the first 6 steps ahead. Similar behavior is observed in the other cases. The above suggests that ESN+LSTM model is a good alternative to consider for short-term forecasting, especially considering that this model uses just 2 epochs, while the LSTM needs more epochs to learn the time series. As a future work, we would like to test these models using more data from other locations around

the world. Also, given that ESN+LSTM uses as hidden units LSTM blocks, it would be interesting to evaluate the performance of this model by changing those units to GRU blocks. Finally, one can explore the capabilities of these networks to address the problem of prediction intervals.

Acknowledgments. This work was supported in part by Fondecyt 1170123 and in part by Basal Project AFB 1800082.

References

1. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
2. Cadenas, E., Rivera, W., Campos-Amezcua, R., Heard, C.: Wind speed prediction using a univariate ARIMA model and a multivariate NARX model. Energies **9**(2), 109 (2016)
3. Chang, W.Y.: A literature review of wind forecasting methods. J. Power Energy Eng. **2**, 161–168 (2014)
4. Cheng, H., Tan, P.-N., Gao, J., Scripps, J.: Multistep-ahead time series prediction. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 765–774. Springer, Heidelberg (2006). https://doi.org/10.1007/11731139_89
5. De Aquino, R.R.B., Souza, R.B., Neto, O.N., Lira, M.M.S., Carvalho, M.A., Ferreira, A.A.: Echo state networks, artificial neural networks and fuzzy systems models for improve short-term wind speed forecasting. In: IJCNN, pp. 1–8. IEEE (2015)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Iversen, E.B., Morales, J.M., Møller, J.K., Trombe, P.J., Madsen, H.: Leveraging stochastic differential equations for probabilistic forecasting of wind power using a dynamic power curve. Wind Energy **20**(1), 33–44 (2017)
8. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks. GMD Report 148, GMD - German National Research Institute for Computer Science (2001)
9. Jung, J., Broadwater, R.P.: Current status and future advances for wind speed and power forecasting. Renew. Sustain. Energy Rev. **31**, 762–777 (2014)
10. Li, L., Liu, Y.Q., Yang, Y.P., Han, S., Wang, Y.M.: A physical approach of the short-term wind power prediction based on CFD pre-calculated flow fields. J. Hydrodyn. Ser. B **25**(1), 56–61 (2013)
11. Liu, Y., Roberts, M.C., Sioshansi, R.: A vector autoregression weather model for electricity supply and demand modeling. J. Mod. Power Syst. Clean Energy **6**(4), 763–776 (2018). https://doi.org/10.1007/s40565-017-0365-1
12. López, E., Valle, C., Allende, H., Gil, E., Madsen, H.: Wind power forecasting based on echo state networks and long short-term memory. Energies **11**(3), 526 (2018)
13. Madsen, H., Nielsen, H.A., Nielsen, T.S.: A tool for predicting the wind power production of off-shore wind plants. In: Proceedings of the Copenhagen Offshore Wind Conference & Exhibition, Copenhagen (2005)
14. Perera, K.S., Aung, Z., Woon, W.L.: Machine learning techniques for supporting renewable energy generation and integration: a survey. In: Woon, W.L., Aung, Z., Madnick, S. (eds.) DARE 2014. LNCS (LNAI), vol. 8817, pp. 81–96. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13290-7_7