21th International Conference on Renewable Energies and Power Quality (ICREPQ'23)
Madrid (Spain), 24th to 26th May 2023
Renewable Energy and Power Quality Journal (RE&PQJ)
ISSN 2172-038 X, Volume No.21, July 2023

Forecasting of wind turbine synthetic signals based on convolutional neural networks

C. Blanco[1], J. E. Sierra-García[2] and M. Santos[3]

[1] ETSI Informática, UNED
28040-Madrid, Spain
cblanco177@alumno.uned

[2] Electromechanical Engineering Department
University of Burgos
09006-Burgos, Spain
jesierra@ubu.es

[3] Institute of Knowledge Technology
University Computense of Madrid
28040-Madrid, Spain
ORCID 0000-0003-1993-8368

Abstract. The importance and future prospects of offshore wind power generation invite great efforts and investments to make it an efficient technology. A crucial aspect is the development of efficient control strategies, which in many cases require models to identify the state of the turbine at a given time accurately. These models must be simple enough not to increase the computational complexity of the control algorithm while being able to capture the nonlinearity and coupling of wind systems. In this work we study the possibility of using neural networks to identify a wind turbine model to predict its power output. Two models, with different number of inputs, have been proposed. LSTM (Long-Short Term Memory) and RNN (Recurrent Neural Network) have been compared, with satisfactory results in terms of model accuracy on an offshore 5MW WT.

Key words. RNN, LSTM, Openfast, wind turbine, system identification.

1. Introduction

Offshore wind energy is experiencing an important growth with an increase of this kind of installation, up to 21% in 2021 [1]. Most of these offshore wind turbines are built in shallow water with fixed foundations, while floating turbines present significant challenges due to their greater complexity [2]. This complexity requires the use of models and control algorithms able to capture all dynamic effects to design and operate optimally the turbines. Besides, these controllers must be computationally cost efficient to operate in real time [3]. The main objectives of wind turbine control are to maintain safe operation, maximize power generation, mitigate fatigue loads on the structure, and avoid fault conditions [2].

Commercial wind turbines typically operate at variable speed, and use different control strategies depending on the operating regime [3]. The use of classical control technics like PID (Proportional-Integral-Derivative) regulators based on a single input single output (SISO) scheme require multiple control loops to stabilize the structure. For more complex turbines, multiple-input multiple-output (MIMO) strategies that are able to capture the most critical dynamics are a must. It is in this context where the identification of the most important features of the turbine system is an important step, in order to design effective control strategies such as MPC (Model Predictive Control) or State Space based control.

Some researchers have worked on this approach. In [4], a MPC based control strategy to stabilize the power generation and reduce the dynamic loads in the structure of the 5-MW floating turbine [5], in the constant speed operation regime, is proposed. An internal linear model, identified from the results of a highly complex non lineal model, is used to predict the system behavior and optimize the control signals. Another approach is to design a digital twin of a 1.5-MW turbine to monitor the state of the structure from the loads estimated with the twin [6]. The authors use a Kalman filter based linear model designed with operational data complemented by nonlinear simulations. This model allows estimating the state of the system at an instant of time with limited information about it. In a similar way, in [7] a monitoring system of a gear box based on data obtained from different wind farms is presented. Different machine learning (ML) and deep learning (DL) methods are used to design predictive models. The model based on deep neural networks performs the best. The use of neural networks to estimate turbine power generation is an interesting topic. In [8] an artificial neural network (ANN) is used to estimate the power generated by a five-turbines wind farm, and to propose optimized angles to reduce the wake effect on the turbines. Long short term memory (LSTM) neural networks have been shown as an effective tool to predict the power generation of the turbines in a wind farm [9]. The training of the networks is done using historical wind data and turbine power generation. In [10] and [11], an effective wind speed estimator based on neural networks is successfully used to improve the performance of a classical controller. These types of networks have also been used in the design of controllers as substitutes of complex and computationally costly strategies [12]. In [13] a blade pitch controller for a 7-MW turbine based on neural networks and reinforcement learning is design, showcasing the potential of the neural network to both substitute other control strategies, and complement and improve their performance.

In this paper, the use of recurring neural networks (RNN) to identify nonlinear systems is studied using simulated data of a 5-MW turbine [5]. Two models with different input signals are compared. In the first model, the power generated is estimated at *t* instant using several key signals, including the wind speed, in past instants. For the second model, the wind speed signal is ruled out. It is considered interesting to compare the performance of both models given that the simulated wind speed data coming from Openfast [14] is usually not available in the typical commercial wind turbine, or the sensor measurement may present uncertainties or faulty behaviour. Therefore, having a model that does not rely on this information is interesting.

The paper is structured with a first section in which the turbine, the software, and the models used in the study are described. The data used to train the neural networks, as well as its structure, are also presented. Next, the tests performed to study the performance of the models and results comparison are presented and discussed. Lastly, the conclusions and possible future lines of study are set out.

2. System and Models Description

A. Wind Turbine Description

The wind turbine used is described in [15]. It is an offshore 5-MW wind turbine with a fixed pillar as support. Table I shows a summary of the key characteristics of the turbine.

Table I – Wind Turbine Characteristics

Rating	5 MW
Rotor orientation, Configuration	Upwind, 3 Blades
Control	Variable Speed, Collective Pitch
Drivetrain	High Speed, Multiple-Stage Gearbox
Rotor, Hub Diameter	126 m, 3 m
Hub Height	90 m
Cut-In, Rated, Cut-Out Wind Speed	3 m/s, 11.4 m/s, 25 m/s
Cut-In, Rated Rotor Speed	6.9 rpm, 12.1 rpm
Rated Tip Speed	80 m/s
Overhang, Shaft Tilt, Precone	5 m, 5º, 2.5º
Rotor Mass	110000 kg
Nacelle Mass	240000 kg
Tower Mass	347500
Coordinate Location of Overall CM	(-0.2 m, 0.0m, 64.0 m)

This choice is particularly convenient since reliable simulation data for this turbine can be generated with Openfast software [14] in the validation tests, Test 19 in particular. Openfast is a software developed by NREL (National Renewable Energy Laboratory), formerly known as FAST (Fatigue, Aerodynamics, Structures, and Turbulence), used for wind turbine nonlinear simulation and multidisciplinary analysis.

This study uses data generated by Openfast to train the neural networks. Data of the wind speed is generated using the Turbsim [15], with an average speed of *12 m/s*.

B. Neural Network Models Signals Description

The objective of this work is to study the possibility of modeling the power generation in the wind turbine using neural networks. An analysis of the correlation between the Openfast output channels is carried out and the most representative signals for the case study are selected.

Table II – Key Signals for the Networks

SIGNAL	OPENFAST CHANNEL	INPUT/OUTPU (MODEL)
Wind Speed x component	Wind1VelX	Input (1)
Wind Speed y component	Wind1VelY	Input (1)
Wind Speed z component	Wind1VelZ	Input (1)
Blade Pitch	BldPitch1	Input (1 and 2)
Rotor Speed	RotSpeed	Input (1 and 2)
Rotor Torque	RotTorq	Input (1 and 2)
Generator Power	GenPwr	Output (1 and 2)

The output of the networks is the ouput power. Model 1 uses all the signals to train the networks, whereas in model 2 the wind speed signals are ruled out. The data are generated with an Openfast simulation for 3000 s, with time increments of 50 ms.

C. Neural Networks Description

The networks used for both models are recurrent networks, using LSTM and simple units RNN, respectively. Fig. 1 shows a typical LSTM unit, and Fig. 2 a simple RNN unit.

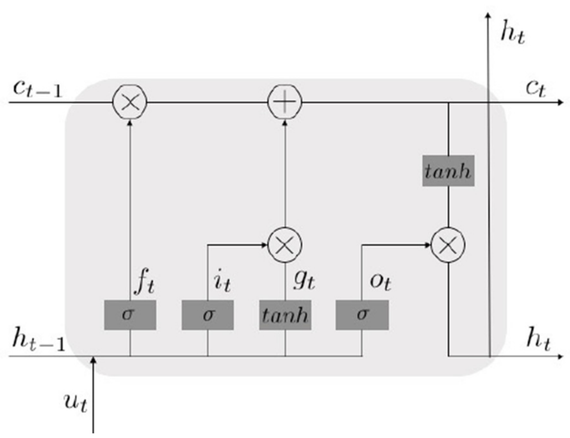

Fig. 1 – LSTM Unit Scheme [16]

Being i_t the input gate, c_t the memory gate, f_t the forget gate, and o_t the output gate. The gate g_t is a complement to the input gate.

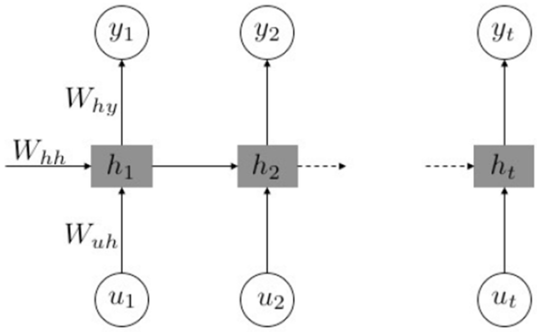

Fig. 2 – Simple RNN Scheme [16]

Where W_{hh}, W_{hy}, y W_{uh} are the weight matrix, u_t the input, y_t the output, y h_t the hidden state variable.

The neural networks are generated with Tensorflow [17]. Both neural networks have the same macro structure. It has not been thoroughly optimized, but parameters that yield acceptable results from a set of values tested have been selected. The networks have three layers of 100 recurrent units and an output layer with 1 standard unit, with a ReLu activation function. After each recurrent layer, a dropout layer is included to avoid overfitting issues. Fig. 3 shows a summary of the LSTM based network, being the RNN network the same.

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 lstm_3 (LSTM)                (None, 20, 100)           43200

 dropout_3 (Dropout)          (None, 20, 100)           0

 lstm_4 (LSTM)                (None, 20, 100)           80400

 dropout_4 (Dropout)          (None, 20, 100)           0

 lstm_5 (LSTM)                (None, 100)               80400

 dropout_5 (Dropout)          (None, 100)               0

 dense_1 (Dense)              (None, 1)                 101

=================================================================
Total params: 204,101
Trainable params: 204,101
Non-trainable params: 0
_________________________________________________________________
```
Fig. 3 – Networks Macro Structure Summary

The input data for the networks are the state vectors with the selected signals over the last second, i.e. 20 temporal increments of 50 ms each. Thus the input has 20 state vectors. The network output is the estimated generator power.

D. System Identification

To perform the system identification, the neural networks have to be trained. It is done using the Openfast generated data over 3000 s, with time increments of 50 ms. That is, there are 60,000 data points for each selected signal. A 60%, 20%, 20% split is performed for training, validation, and test sets. Since it is a time series, this division is done in chronological order.

1) Model 1

The objective of the first model is to estimate the generator power at time t using an input state vector with the complete set of signals shown in Table II. The networks performance over the test data set is shown in Table III, Fig. 4 (LSTM) and Fig. 5 (RNN). The MAE (mean absolute error) and RMSE (root mean square error) are used as metrics.

Table III – Model 1 Training Performance Summary

NETWORK	ERROR	
	MAE [kW]	RMSE [kW]
LSTM	26.23	40.67
RNN	30.76	50.14

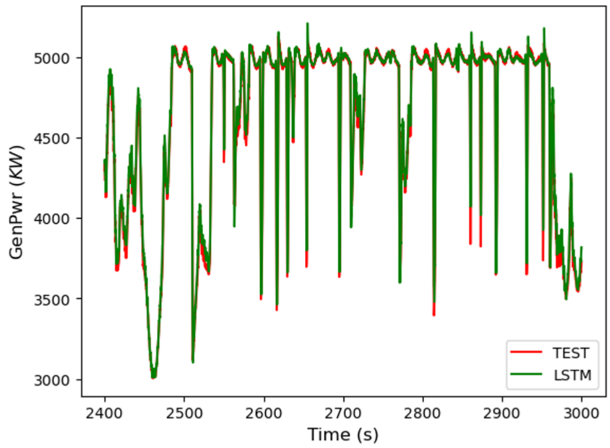

Fig. 4 – Training Results Model 1, LSTM Network

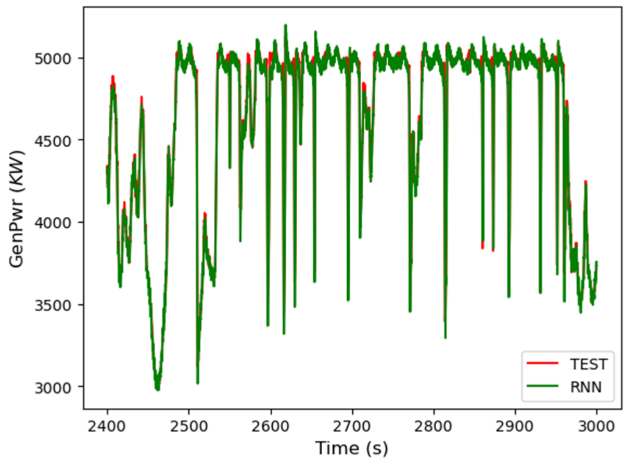

Fig. 5 – Training Results Model 1, RNN Network

As it is possible to see, both networks have a good performance on the training data. The LSTM networks presents smaller errors.

2) Model 2

The objective of the second model is to estimate the output power at time t using an input state vector with the set of signals shown in Table II, except the wind speed that has been ruled out. This may be a common scenario due to the lack of real measurements, uncertainty, noise, or failures in the sensors.

The networks performance over the test data set is shown in Table IV, Fig. 6 (LSTM) and Fig. 7 (RNN).

Table IV – Model 2 Training Performance Summary

NETWORK	ERROR	
	MAE [KW]	RMSE [KW]
LSTM	33.78	58.31
RNN	57.94	77.73

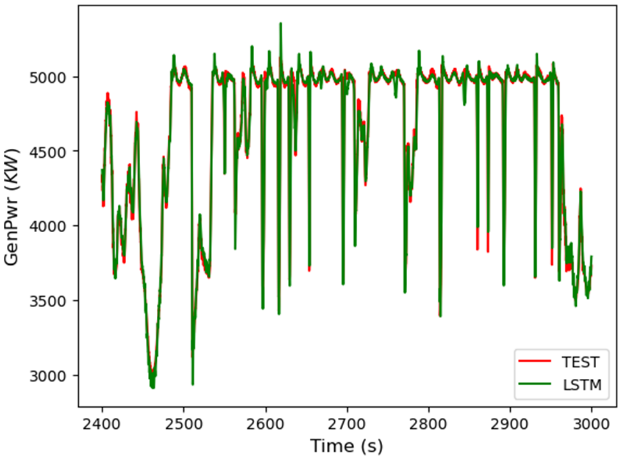

Fig. 6 – Training Results Model 2 LSTM Network

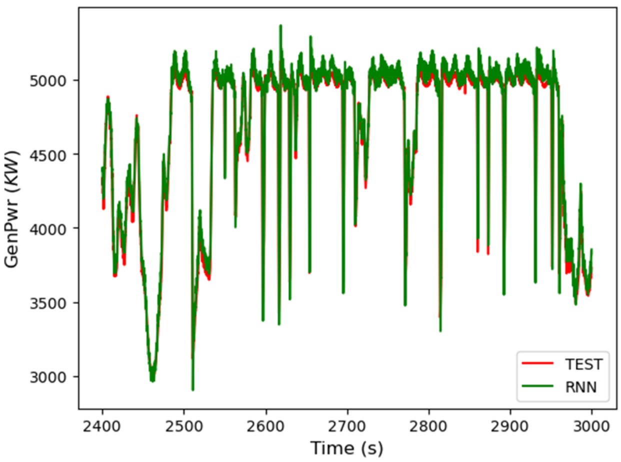

Fig. 7 – Training Results Model 2 RNN Network

Again, both networks have a good performance but in this case the errors are bigger. The LSTM networks gives better results, as in the previous case.

3. Discussion of the Results

This last section presents the results of the tests that have been carried out with the two models defined above. To compare the models performance with reliable data, the regression test data no. 19 of the Openfast software is used. The regression test has data for the signals used in this study for a period of 60 s, with time intervals of 50 ms. Moreover, the data used in this section are different from the training and testing data used during the validation of the networks.

A. Model 1 Test

The results of the model 1 test are summarized in Fig. 8 and Table V. The results are consistent with the training results. Both networks types show good performance, being the LSTM slightly better.

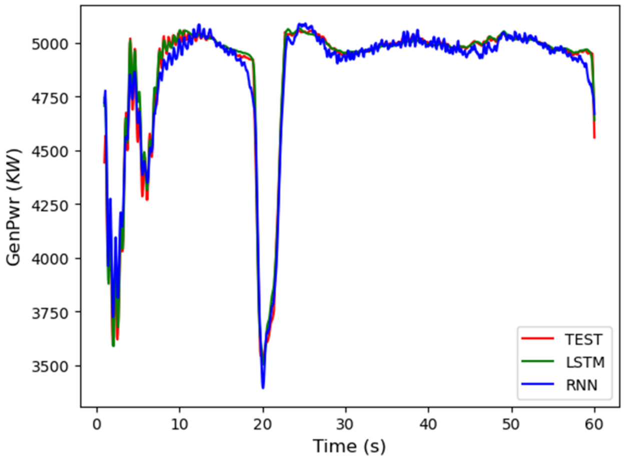

Fig. 8 – Model 1 Test Results

Table V – Model 1 Results Summary

NETWORK	ERROR	
	MAE [KW]	RMSE [KW]
LSTM	17.67	33.56
RNN	34.02	53.36

It is worth it to remark that the errors are even smaller than in training and validation, at least for the LSTM network.

B. Model 2 Test

The results of the model 2 test are summarized in Fig. 9 and Table VI. The results again are consistent with the training results.

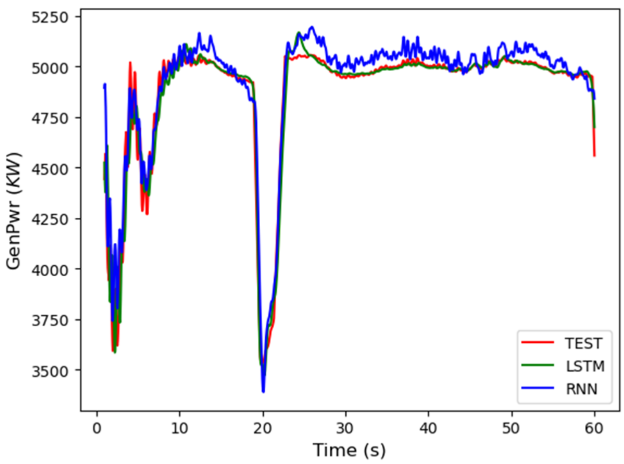

Fig. 9 – Model 2 Test Results

Table VI – Model 2 Results Summary

NETWORK	ERROR	
	MAE	RMSE
	[KW]	[KW]
LSTM	38.21	62.21
RNN	62.22	85.13

Again results are good and errors are similar to the ones obtained during the training, but slightly bigger.

The comparison between both models leads to interesting conclusions about the effect of the wind speed on the behaviour of the models. As it can be seen, both types of networks worsen the performance in the second case (model 2), when the wind speed is not considered. In particular, the x component of the wind speed is the best correlating with the generator power.

4. Conclusions and future works

The suitability of recurrent neural networks to estimate the generator power of a wind turbine is studied. Two models with different input signals are trained and used to compare their performance. The second model, with fewer inputs given that the wind speed is not used, is worse than the first model. Still, it gives good results regarding the prediction of the output power. However, it is an interesting exercise given that the wind speed is a variable that might not be available or may be not reliable. Both models show good potential to be used, and in both cases the LSTM networks surpasses the RNN.

Even though the results when an abrupt variation of the signal is not desirable, this performance can be improved optimizing the hyper-parameters of the networks or even implementing different mechanisms as a function of these changes [18]. The number of state vectors considered as input to the networks is possibly one of such parameters that might be sensitive to the mentioned behaviour. Moreover, it could be interesting to modify or increase the Openfast signals used as inputs to improve the predictions. Lastly, a future line of study could also be to design a similar model specifically tailored to use as part of a model predictive controller.

Acknowledgement

This work was partially supported by the Spanish Ministry of Science, Innovation and Universities under MCI/AEI/FEDER Project no. PID2021-123543OB-C21.

References

[1] Global Wind Energy Council, «Global Wind Report 2022,» GWEC, Brussels, 2022.

[2] R. Hu, L. Conghuan, H. Ding y P. Zhang, «Implementation and evaluation of control strategies based on an open controller for a 10 MW floating wind turbine,» Renewable Energy, vol. 179, pp. 1751-1766, 2021.

[3] A. D. Wright y L. J. Fingersh, «Advanced Control Design for Wind Turbines Part I: Control Design, Implementation, and Initial Tests,» NREL, Golden, 2008.

[4] T. Wakui, A. Nagamura y R. Yokoyama, «Stabilization of power output and platform motion of a floating offshore wind turbine-generator system using model predictive control based on previewed disturbances,» Renewable Energy, vol. 173, pp. 105-127, 2021.

[5] J. Jonkman, S. Butterfield, W. Musial y G. Scott, «Definition of a 5-MW Reference Wind Turbine for Offshore System Development,» NREL, 2009.

[6] E. Branlard, J. Jonkman, S. Dana y P. Doubrawa, «A digital twin based on OpenFAST linearizations for real-time load and fatigue estimation of land-based turbines,» Journal of Physics: Conference Series, nº 1618, 2020.

[7] L. Wang, Z. Zhang, H. Long, J. Xu y R. Liu, «Wind Turbine Gearbox Failure Identification With Deep Neural Networks,» IEEE Transactions on Industrial Informatics, vol. 13, nº 3, pp. 1360-1368, 2017.

[8] H. Sun, C. Qiu, L. Lu, X. Gao, J. Chen y H. Yang, «Wind turbine power modelling and optimization using artificial neural network with wind field experimental data,» Applied Energy, vol. 280, 2020.

[9] J. Zhang, J. Yan, D. Infield, Y. Liu y F.-s. Lien, «Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and Gaussian mixture model,» Applied Energy, vol. 241, pp. 229-244, 2019.

[10] J. E. Sierra-García y M. Santos, «Improving wind turbine pitch control by effective wind neuro-estimators,» IEEE Access, vol. 9, pp. 10413-10425, 2021.

[11] J. E. Sierra-García y M. Santos, «Deep learning and fuzzy logic to implement a hybrid wind turbine pitch control,» Neural Computing and Applications, vol. 34, nº 13, pp. 10503-10517, 2022.

[12] C. Blanco Fernández, J. E. Sierra-García y M. Santos, «Control de un laboratorio de control de temperatura mediante redes neuronales recurrentes,» XLIII Jornadas de Automática, pp. 193-200, 2022.

[13] J. E. Sierra-García y M. Santos, «Redes neuronales y aprendizaje por refuerzo en el control de turbinas eólicas,» Revista Iberoamericana de Automática e Informática industrial, vol. 18, nº 4, pp. 327-335, 2021.

[14] J. Jonkman y W. Musial, «Offshore Code Comparison Collaboration (OC3) for IEA Task 23 Offshore Wind Technology and Deployment,» NREL, Golden, 2010.

[15] «https://github.com/OpenFAST/openfast,» National Renewable Energy Laboratory, [En línea]. Available: https://github.com/OpenFAST/openfast. [Último acceso: 27 enero 2023].

[16] S. Spielberg, P. Kumar, A. Tulsyan, B. Gopaluni y P. Loewen, «A Deep Learning Architecture for Predictive Control,» IFAC Papers On Line, vol. 51, nº 18, pp. 512-517, 2018.

[17] M. Abadi, P. Barham, J. Chen, Z. Chen, Z. Davis, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard and et al., «Tensorflow: A system for large scale machine learning,» OSDI, vol. 16, pp. 265-283, 2016.

[18] J.E. Sierra-García, M. Santos, «Switched learning adaptive neuro-control strategy». Neurocomputing, 452, 450-464, 2021.