

## ORIGINAL RESEARCH

# Statistical evaluation of wind speed forecast models for microgrid distributed control

Marcos Eduardo Cruz-Victorio  | Behzad Kazemtabrizi  | Mahmoud Shahbazi 

Engineering Department, Durham University,  
Durham, UK

**Correspondence**

Behzad Kazemtabrizi, Engineering Department,  
Durham University, Durham, DH1 3LE, UK.  
Email: [behzad.kazemtabrizi@durham.ac.uk](mailto:behzad.kazemtabrizi@durham.ac.uk)

**Funding information**

Consejo Nacional de Ciencia y Tecnología, Grant/  
Award Number: 447877

**Abstract**

With the increasing needs to decarbonise existing energy systems, there is an effort to integrate small-scale distributed generation sources, such as wind generators, with the electric demand in circuits known as microgrids. The operation of distributed variable renewable resources is subject to an optimum operating regime, ahead of real-time, which relies on output forecast. However, many wind speed forecast models are designed for centralised controllers, which are vulnerable to control failures. A suitable wind forecast model for a distributed control system is, therefore, required for optimal and reliable use of renewable generation. This paper presents a comparison of wind speed forecast models suited for distributed control, evaluating them in terms of the statistical significant difference in accuracy and computational resource requirements. This is essential since computational resources are limited in distributed control schemes. The data used in this paper is the historical wind speed of the Auchencorth Moss Atmospheric Observatory from 2016 to the end of 2019. Two forecast model types based on Auto-regression and Artificial Neural Network (ANN) are compared using the Diebold-Mariano test. Results show that ANN models with parallel hidden layers have the highest accuracy with statistical significant difference, while remaining suitable for microgrid distributed control.

**KEYWORDS**

DM-test, microgrid, neural network, wind speed forecast

## 1 | INTRODUCTION

To increase the efforts to combat climate change, the electricity sector is encouraged to maximise integration of renewable generation, for example, wind turbines. Wind energy is, however, inherently variable and such variations are often hard to control in the short term. For this reason, any reliable power generation scheduling scheme for wind-integrated power systems depends on the accuracy of wind speed forecast to account for unforeseen wind variations and ensure balance is held between generation and demand, thereby allowing for an optimum (in economical terms) and stable operation of the system. In this regard, Artificial Neural Network (ANN) methods are among the latest time-series forecast models used [1–6].

Research efforts have been focussed in centralised control schemes and their respective forecast models. These forecast models are computationally complex and operate with several external inputs in central controllers [6]. Examples of ANNs in central controllers include the Non-linear Auto-regression (AR) model with exogenous variables (NARX) neural networks used for electricity price forecast [7, 8].

Notwithstanding this, maximising the efforts of energy decarbonisation requires the integration of distributed generation in the form of onshore wind integrated microgrids, where centralised control schemes could represent a vulnerability due to existence of a single point of failure.

To address this vulnerability, distributed control has emerged as an alternative for microgrid control with improved

**Abbreviations:** ANN, Artificial Neural Network; AR, Auto-regression; ARMA, Auto-regression Moving Average; NARNET, Non-linear Auto-regression Network; DM, Diebold-Mariano; NARX, Non-linear Auto-regression with Exogenous Variables; RMSE, Root Mean Square Error; DER, Distributed Energy Resource; ESS, Energy Storage System; RELU, Rectified Linear Unit; MSE, Mean-Squared Error; CDF, Cumulative Distribution Function; PDF, Probability Density Function; p-value, probability of null hypothesis.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2022 The Authors. *IET Smart Grid* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

reliability, as the operation of the microgrid does not depend of any single component. However, optimal implementation of distributed control schemes in a microgrid setting still requires accurate forecasting, in this case of the wind speed, ahead of real-time, whereas the rest of the microgrid control operate in real-time.

In distributed control, the calculation required for forecasting should be carried out locally at the distributed controllers, which may not necessarily have the same computational resources of a central controller; additionally, a fast time response is needed to allow real-time control. Moreover, using local forecast models in the distributed control architecture also improves the security of local information, as the information itself is not required to be shared directly with the rest of the system, communicating only the power schedule generated.

For these reasons, suitable forecast models implemented require provision of both accuracy and computational efficiency. Additionally, forecast methods are often applied to aggregated data, which may not be suitable for short-term forecasting in local applications [9, 10].

As a continuation of our previous work [11, 12], two main forecast model families suitable for microgrid distributed control are analysed: ANNs and AR models; the latter is optimised using the quasi-Newton Method. Both families are compared with the persistence model and the AR Moving Average (ARMA) model as a point of reference [13].

The Non-linear AR Network (NARNET) is a type of ANN used to forecast future time steps of a non-linear time series autonomously. This property is ideal for wind speed forecast in distributed control, given that the NARNET would not be subject to vulnerabilities caused by dependence of external signals during operation.

As ANNs, NARNETs are organised in groups of connected layers of neurons, with non-linear functions, known as activation functions, applied to the output of each layer. In other words, the activation function is the non-linear transformation applied to the data that allows the NARNET to model the non-linear properties of the data. Some of the most common activation functions used in forecast models include the Logistic Sigmoid function, Tangent Hyperbolic function and Rectifier Linear function [8, 14]. NARNETs may have a single layer or have layers connected in series or in parallel [1, 5].

On the other hand, AR models are statistical models, based on linear-regression for time-series forecasting. They do not implement any physical description of the model but rather are entirely based on the past values of the series to predict the future values [15, 16]. This property makes them easy to implement as forecasting models, which they can be used for comparison with other auto regressive forecast models. In a similar manner, the persistence model serves as the simplest auto regressive model that can be implemented, often used as a point of comparison for other forecast models [17–19].

The persistence model is based on the idea that the changes in two consecutive samples of a time series are small enough to obtain a good approximation of the next sample,

considering only the previous sample to forecast the next one [20]. An evaluation of the performance of the persistence model can also be used as an indication of the amount of variation of consecutive samples in a given dataset.

Many efforts have been channelled to develop forecast models with high accuracy, although very few studies report the statistical significant difference between the analysed models. In other words, many authors often do not explain if the reported performance could be attributed to randomness in the data used [2, 4, 6, 10, 14, 15, 19–30].

The determination of statistical significant difference could be as important as the performance of the forecast model. The statistical evaluation method in this paper is based on the Diebold-Mariano (DM) test, which is used to compare the performance of two forecast models between them and against true values, based on a given Probability Density Function (PDF) [31].

The use of statistical significant difference tests of forecast models has been reported previously, although the computational parameters of the models are not reported [27, 32], or only a single configuration is tested for each of the forecast models compared [30, 33, 34]. Similarly, the PDF used is omitted in previous studies such as Ref. [17, 34]. The use of the DM-test has been reported for large-scale wind farm applications [18, 35]; however, in these applications, the DM-test of the error residuals was applied to a normal distribution, which may not be suitable at the microgrid scale.

In other words, there is still a need to develop forecast models suitable for microgrid distributed control. To combat this shortcoming, this paper has the objective of proposing a development and validation method of a suitable forecast model for microgrid distributed control [36]. This requires that the forecast model developed is only dependable on historical data that could be stored locally, such that each distributed controller has access to the forecast within a few milliseconds. To achieve optimal operation in a distributed environment, AR forecast models are proposed as a suitable solution.

The proposed evaluation method allows the selection of the best short-term wind speed forecast model, suitable for distributed microgrid control. This method evaluates the performance of each forecast model in terms of accuracy using the Root Mean Squared Error (RMSE). The models are then compared in size to verify computational requirements. Finally, the models are compared using the DM-test to ensure the statistical significant difference between the models, taking into account randomness in the data.

## 1.1 | Key contribution

The contribution of this paper can be summarised with the following points:

- The proposition of a suitable short-term wind speed forecast model for distributed control schemes applied to microgrids (for performance optimisation), where the

execution of the forecast model is as autonomous as possible to maintain reliability.

- This work proposes the use of an evaluation method, based on the DM test that verifies the statistical significant difference between forecast models with different calculation requirements and the historical data. With this method, it can be verified that the accuracy results are not due to randomness and that a model can be selected for optimal use of computational resources.
- As an improvement from our previous work, it was found that the NARNET with the parallel architecture and Logsig activation function has the highest accuracy for the Auchencorth Moss wind speed data. Additionally, it was found that it has a statistical significant difference compared to more complex models and the historical data, which indicate a valid forecast model with smaller requirement of computational resources. This properties make this model ideal for microgrid distributed control.

The rest of the paper is organised as follows: Section 2 describes the wind speed forecast methods. Section 3 details how the DM-test is calculated for model comparison. Section 4 presents the test case and how it is applied to the forecast models. Section 5 presents the evaluation results, which are discussed and summarised in section 6.

## 2 | MICROGRID DISTRIBUTED CONTROL

As stated previously, accurate and computationally efficient forecasting of distributed variable renewable generation in a microgrid setting is a crucial step of any efficient and reliable distributed control scheme aimed at forward generation scheduling of such assets. Accurate forecasting would allow for planning the operation of the distributed generation resources in a safe and reliable manner.

To this end, in this section, a full description of a microgrid control scheme is given. The term ‘microgrid control’ in this

paper is used to represent any control scheme for such application as optimum generation scheduling coupled with real-time control of distributed generation assets within a microgrid to ensure demand is met at all times, subject to the microgrid's physical and the operational boundaries. The control schemes can be implemented either centrally or in a distributed fashion. The latter scheme is the focus of this paper.

The microgrid control presented in this paper is organised in a hierarchy, where each level has specific objectives and time response capabilities, with control references being supplied by the upper levels of the hierarchy. In the control scheme presented here, tertiary control is considered the highest control level and is used to coordinate different microgrids to allow the coordination of the power flow between them and the main grid. Secondary control is used to plan and coordinate an optimum generating schedule for the Distributed Energy Resources (DERs) within each microgrid by generating their corresponding power set points. The primary control directly regulates the voltage of the individual DERs, having one controller per resource, to maintain the power flow required considering stability and voltage regulation limits [11, 37, 38]. The hierarchical control framework is shown in Figure 1.

While many microgrids have a centralised secondary control to maximise the optimal coordination of all DERs, this kind of control structure tends to be computationally expensive and leaves all primary controllers dependent on a single secondary controller, which means that the entire operation and privacy of the microgrid components are vulnerable to a single point of failure.

To combat this vulnerability, distributed secondary control emerges as an alternative to centralised control, with separate control units that coordinate with each other to realise the power management objectives in a communication network. This also adds flexibility to the structure of the microgrid, as any DER can be added or removed from the microgrid without the need to redesign the secondary control, as the system adjusts itself [37, 39, 40]. The centralised and distributed control systems are illustrated in Figure 1.

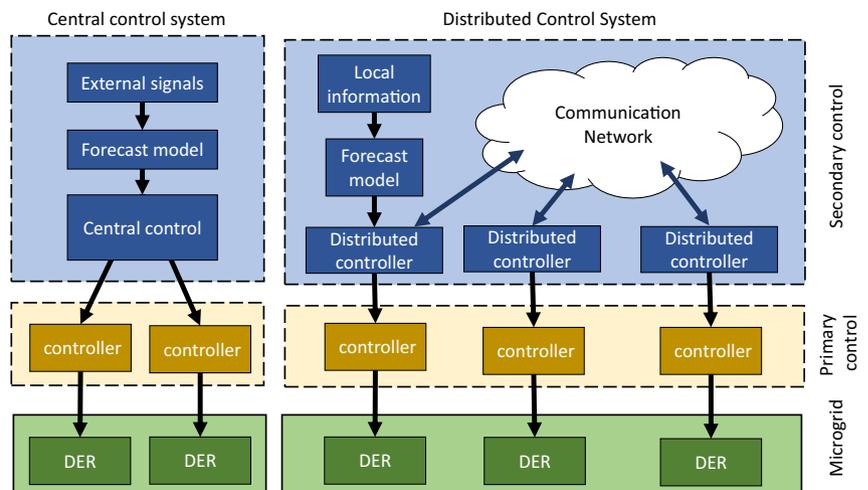


FIGURE 1 Microgrid control hierarchy for centralised and distributed control. Only secondary and primary levels are shown

As mentioned in the introduction, to achieve optimal operation at the secondary and tertiary control levels, it is necessary for the microgrids to have information about available generation in the DERs in advance. This is due to start-up cost for controllable DERs, the states of charge for the Energy Storage Systems and stochastic generation for the renewable generation sources, which make the operation of the microgrid dependent on its previous states. Therefore, it is essential that decisions taken at the secondary and tertiary levels are supported by accurate forecasts of relevant renewable resources.

As distributed control systems are required to maintain a communication network additionally to realise power management optimisation, it is necessary to minimise the computational cost of the forecasting methods used. For this reason, suitable forecast models for microgrid distributed control, in this case wind speed forecast models, must have the highest accuracy possible with the least computational resource requirements.

In this paper, the wind speed forecast models are required to estimate electric power generated by wind resources, with the purpose of minimising renewable energy curtailment and maximising the lifespan of the batteries used to regulate power from wind turbines. For this reason, it is important to overview the process to calculate expected wind power output given a forecast model in the next subsection.

## 2.1 | Wind power generation

For the case of wind turbines, the electrical power output  $P_{wt}$  is defined in general as:

$$P_{wt}(V_{ws}) = \begin{cases} 0, & V_{ws} < V_{\min} \vee V_{ws} > V_{cut} \\ C_p(V_{ws})^3, & V_{rated} < V_{ws} < V_{cut} \\ P_{\max}, & \text{else} \end{cases} \quad (1)$$

where  $V_{ws}$  is the wind speed,  $V_{\min}$  is the minimum wind speed for generation or cut-in speed,  $V_{rated}$  is the rated wind speed of the wind turbine,  $V_{cut}$  is the maximum wind speed for generation or cut-out speed,  $P_{\max}$  is the maximum wind turbine power output, and  $C_p$  is the coefficient of performance of the wind turbine.

In this case, the expected power generation  $\mathbb{E}(P_{out})$  over some period  $\tau = [1 \dots \tau_{\max}]$  with duration  $\tau_{\max}$ , given that the wind speed is random, is defined by [41]:

$$\mathbb{E}(P_{out}) = \sum_i P_{wt}(V_{ws} = v_{w_i}) p_{ws,i}, \forall i \in \tau \quad (2)$$

where  $p_{ws,i}$  is the probability that  $V_{ws}$  equals the  $i$ th realisation  $v_{w_i}$ . The expected power generation is then an estimation of electrical power output taking into account the randomness in the wind speed over a finite time period. Given that accurate estimations depend on the wind speed forecast methods applied, the next section describes suitable wind speed forecast models for distributed control, as they rely only on historical data that can be stored and shared locally in each controller.

## 3 | WIND SPEED FORECAST METHODS

As mentioned in the previous section, the methods in this section are suitable for wind speed forecast in a distributed control architecture for microgrids. All of the methods presented in this section are ultimately data-driven and differ in how the data is handled in each case, which would generate different forecast outputs. These models will be analysed in terms of accuracy and statistical significant difference.

The forecast methods in the next subsections are used to model the true wind speed as follows:

$$Y_{ws}(t) = F(t) + \epsilon(t) \quad (3)$$

where  $Y_{ws}$  is the historical wind speed,  $t$  is time,  $F$  represents one of the forecast model in the following subsections and  $\epsilon$  is the error at time  $t$ , in other words, the difference between the real wind speed and the wind speed forecast for the same  $t$ . Each of the forecast models are described next.

### 3.1 | Persistence model

The persistence model is the simplest forecast model and is defined as [20]:

$$F_1(t) = Y_{ws}(t-1) \quad (4)$$

where  $F_1$  is the wind speed forecast. The basic assumption of the model is that the difference between one sample and the next is small when the sampling frequency is sufficiently small, and therefore the wind speed one step ahead will remain mostly unchanged. The effectiveness of the persistence model depends on the location and sampling frequency. This model would not be suitable for generating forecasts with time horizons larger than 1 hour. This model will be used as a baseline to compare the Neural Network models, with the use of the DM-test.

### 3.2 | ARMA model

This method was been recently used in wind speed forecast in scenarios of limited information, making it suitable for distributed control systems and therefore is useful as point of comparison for other forecast models [13].

This model combines the AR model with a correction of the average depending on the error obtained in previous evaluations of the model. The model is defined as follows:

$$F_2(t) = \mathcal{W}_0 + \sum_{i \in \mathcal{P}} \mathcal{W}_a(i) Y_{ws}(t-i) + \sum_{j \in \mathcal{Q}} \mathcal{W}_b(j) \epsilon(t-j) \quad (5)$$

where  $F_2$  is the ARMA model,  $\mathcal{W}$  are the model parameters,  $\mathcal{W}_0$  is a constant,  $\mathcal{W}_a$  is a vector containing the weights for the AR model and  $\mathcal{W}_b$  contains the elements for the moving average model,  $\mathcal{P} = \{1, \dots, \mathcal{P}_{ARMA}\}$ ,  $\mathcal{P}_{ARMA}$  refers to the order of the AR component,  $\mathcal{Q} = \{1, \dots, \mathcal{Q}_{ARMA}\}$  and  $\mathcal{Q}_{ARMA}$  is

the order of the moving average model. The parameters of this model and the AR model are obtained with the quasi-Newton method, which will be discussed in the next subsection. This model will also be used as a point of comparison against the other models in terms of accuracy and with the use of the DM-test.

### 3.3 | Auto-regression model

This model also produces the forecast of the wind speed only with past values with the difference that it does not require the information of the previous errors. The advantages of this method are the convergence of the solution and the quicker generation of the solution compared to the NARNET.

This forecast model is defined as follows:

$$F_3(t) = \sum_i \mathcal{W}_c(i) Y_{ws}(t-i), \forall i \in D_{AR} \quad (6)$$

where  $F_3$  is the AR wind speed forecast, realised by the weighted sum of wind speed samples  $Y_{ws}$  with a lag order, or delay size,  $d_{AR}$ . In 6 the set  $D_{AR} = [1 \dots d_{AR}]$  is multiplied by the vector containing the  $W_i$  optimised weights, from  $Y_{ws}(t-d_{AR})$  to  $Y_{ws}(t-1)$ .

The weights  $\mathcal{W}$  are optimised with the quasi-Newton method by solving the following least squares problem:

$$\min_{\mathcal{W}} \sum_t (Y_{ws}(t) - F_k(t))^2, \forall t \in T \quad (7)$$

where the set  $T = [1, \dots, N_T]$  represents the times at which the historical data is taken for  $N_T$  total available samples, for the  $k$ th forecast model.

In Equation (7), all the weights are tuned to minimise the total summation of the squared errors applied to the entire set of historical sample data available.

In this case, the fitting algorithm of the model starting with all the weights is set to 1 and the weights are updated in the direction of highest decrease of Equation (7), also known as the cost function, until the stop criterion is met. The stop criterion is met in this case in any of the following cases: **1)** the cost function is equal to 0. **2)** The change in the cost function is equal or below the  $1 \times 10^{-5}$ . **3)** The change in the weights is equal or below  $1 \times 10^{-6}$ . **4)** Iterations have reached a maximum of 1200.

### 3.4 | NARNET model

This method is based on the Non-linear Auto-regressive ANN. In this type of ANN, the output is feedbacked to the input, with an internal matrix called Delay that stores information and serves as a memory in time-series forecasting. This network then produces the forecast based only on past values of the time series, in this case, past values of wind speed.

In this study, the connections between the layers are defined by three different architectures tested that are shown in Figure 2.

Each hidden layer in the NARNET has the following function:

$$A_{l+1} = f_a(\omega_l A_l + B_l) \quad (8)$$

where  $A_{l+1}$  is the output of the layer  $l$ ,  $A_l$  is the input of the layer,  $\omega_l$  is the weight matrix of the layer,  $B_l$  is the bias of the layer and  $A_1$  is the input of the NARNET. Each hidden layer has an activation function  $f_a$  at the end, which encodes the non-linear properties of the data; in other words, it is the function that allows the NARNET to forecast non-linear time series. The output layer function is:

$$A_{l+1} = \omega_l A_l + B_l \quad (9)$$

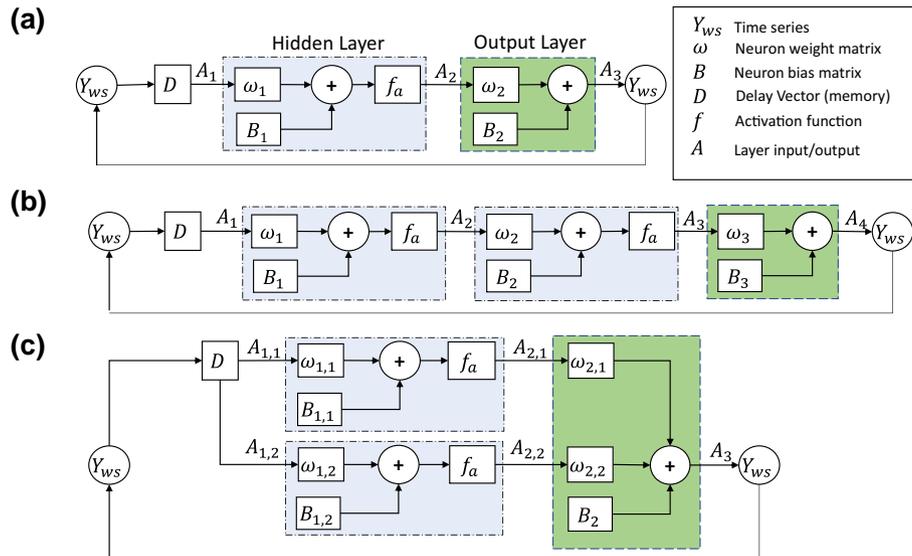


FIGURE 2 NARNET Architectures used for this study. (a) Single hidden layer, (b) series hidden layers and (c) parallel hidden layers

In this study, three activation functions are used for the hidden layers of the NARNET forecast model:

$$f_1 = \frac{1}{1 + e^{-x}} \quad (10)$$

$$f_2 = \frac{2}{1 + e^{-x}} - 1 \quad (11)$$

$$f_3 = \max(0, x) \quad (12)$$

where  $f_1$  is the Logistic Sigmoid function,  $f_2$  is the hyperbolic tangent Sigmoid function and  $f_3$  is the Rectified Linear Unit (ReLU) function [8, 14, 42].

The input  $A_1$  at a specific time  $t$  is stored in the NARNET as part of an internal vector  $D$  that also contains past values of the input. In other words,  $D$  represents the ‘memory’ of the NARNET. In this case,  $D$  contains past values of  $Y_{ws}$  up to  $t$  that are used to generate the wind speed at time  $t + 1$ .

The general transfer functions of the entire NARNET for each of the architectures used in this paper, with  $j$  delay size,  $q \times 1$  input size and  $k$  neurons in each layer can be derived by the equations of each layer [42]. For the case of the single hidden layer it is described as follows [12, 42]:

$$F_4(t) = \omega_{q \times k} (f_a(\omega_{k \times jq} D_{jq \times 1}(t-1) + B_{k \times 1})) + B_{q \times 1} \quad (13)$$

For the series hidden layers it can be described as:

$$F_5(t) = \omega_{q \times k} (f_a(\omega_{k \times k} (f_a(\omega_{k \times jq} D_{jq \times 1}(t-1) + B_{k \times 1}) + B_{q \times 1})) + B_{q \times 1}) \quad (14)$$

Finally, for the parallel hidden layers we will have:

$$F_6(t) = \omega_{q \times k} (f_a(\omega_{k \times jq} D_{jq \times 1}(t-1) + B_{k \times 1})) + \omega_{q \times k} (f_a(\omega_{k \times jq} D_{jq \times 1}(t-1) + B_{k \times 1})) + B_{q \times 1} \quad (15)$$

In equations (13), (14) and (15), the subscripts show the size of each matrix to illustrate the complexity of the NARNET, in particular from the matrix product  $\omega_{k \times jq} D_{jq \times 1}$ .

Given the complexity and randomness introduced in the training, the NARNET model may produce different results in terms of accuracy for the same size of the internal matrices for each training realisation, which will be discussed in the next subsection.

### 3.4.1 | The Non-linear Auto-regression Network training method

The NARNET is a machine learning method driven by historical data. The model produces different results each time it

is run even when the inputs are the same because of the randomness induced in the training process.

The training process is randomised for two reasons: to mitigate the bias from the selection of the training, testing and validation sets and to better explore the accuracy the model can achieve depending on the sets mentioned before. In other words, this mechanism aids in the generalisation of the model for new data.

The training algorithm for the NARNET is the Levenberg–Marquardt algorithm, which minimises the MSE of the NARNET between the training values, which are the actual wind speeds, and the output values by adjusting the weights and biases at each layer. The minimisation problem solved by the training algorithm is defined as follows:

$$\min_{\omega, B} \frac{\sum_t (Y_{ws}(t) - F_k(t))^2}{N_T}, \forall t \in T \quad (16)$$

The method uses the back-propagation algorithm to obtain the Jacobian of the errors with respect to the weights and updates the weights using a method similar to the quasi-Newton methods. The training method can be summarised as follows. The network is initialised with all the weights and biases randomised with values between  $-1$  and  $1$ , using a uniform distributed PDF. From this starting point, the weights and biases are updated using the Levenberg–Marquardt back-propagation algorithm, following the direction of steepest descend until the stop criterion is met. The stop criterion is met when any of these conditions is met: **1)** the epochs have reached a maximum of 1000; an epoch is completed when the algorithm uses all the training data once. **2)** The performance goal of  $MSE = 0$  is achieved. **3)** The gradient of the performance is  $1 \times 10^{-5}$  or lower. **4)** The maximum validation check of six epochs is reached.

In this training method, the entire dataset is divided in three sets, namely, the training set, the validation set and the testing set. The training set is the set of samples used directly to obtain the weight and bias updates of the NARNET; therefore, the weights and biases are heavily biased by the training set. To prevent over-fitting of the NARNET, the validation set is used as a separate calculation of the performance of the NARNET. When the performance from the validation check is continuously deteriorating up to the maximum of six epochs, the training stops. This allows the NARNET model to be validated for the general case. However, the validation set indirectly influences the weights of the NARNET during training. For this reason, the test set is not used during training but as its name implies, it is used to test the performance of the NARNET once it is training. The training, testing and validation set are randomly selected from the entire dataset in ratios of 70%, 15% and 15% accordingly. The entire training method for the NARNET is summarised in Figure 3.

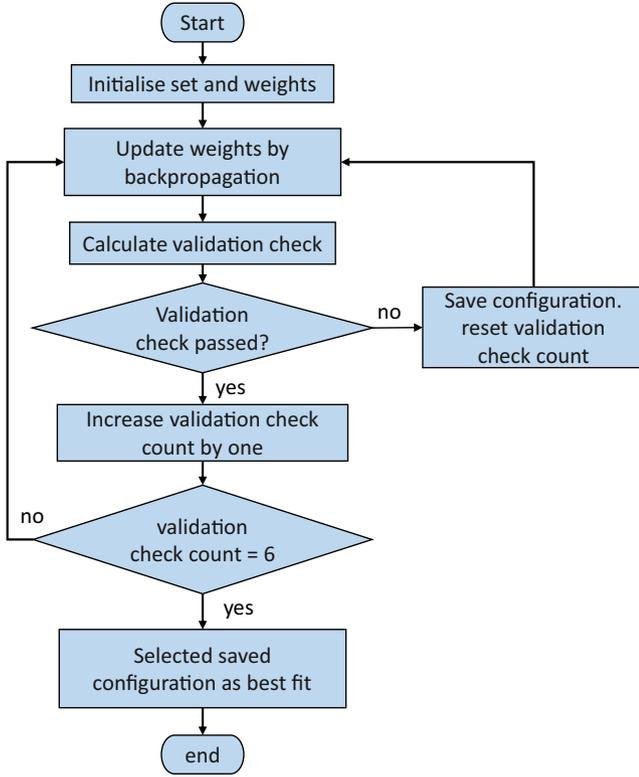


FIGURE 3 Summary of the training process of the non-linear auto-regression network (NARNET) model

#### 4 | DM-TEST FOR FORECAST MODEL EVALUATION

To compare the forecast models using the DM-test, each model is first evaluated in terms of RMSE, following the results from Ref. [12]. The RMSE is defined as follows for the  $k$ th forecast model:

$$RMSE = \sqrt{\frac{\sum_t (Y_{ws}(t) - F_k(t))^2}{N_T}}, \forall t \in T \quad (17)$$

Then, the wind speed forecast models are analysed in terms of statistical significant difference, repeating the experiment and performing the DM-test of every realisation of each of the models presented in the previous section.

Each method is compared against the others and the true wind speed values to determine if the differences in accuracy are due to randomness in the true dataset or because a forecast model is actually more accurate than others.

In this work, the null hypothesis is that the forecast models have the same accuracy, and the difference in the performance is due to randomness in the dataset, or because the compared models are nested. A model  $F_x$  is nested in  $F_y$  when  $F_y = F_x + F_z$ , for any  $F_z$  function. Therefore, a rejection of the Null hypothesis concludes that there is a statistically significant difference between the models and a better performance is due to a better model.

The DM-test is essentially based on the ratio of the mean and covariance of the errors, encapsulated in the difference of errors between models. The DM-test is based on the principle that the errors from a valid forecast model are covariance stationary, which means that the residuals, or errors, of a valid model should behave as a random series. This test also assumes that the time series requiring forecast is infinite, and therefore any set used for test calculation is a sample of the entire series.

The test requires 3 sets as inputs and provides a probability ( $p$ -value) that the null hypothesis is true. The inputs are the true wind speed  $Y_{ws}$ , and the output of the two models that are being compared. When the  $p$ -value is sufficiently small, the null hypothesis is rejected. When this occurs, a statistically significant difference in accuracy between the forecast models is concluded.

From each pair of forecast model outputs  $F_a$  and  $F_b$ , the squared errors  $\mathbf{e}$  are calculated as:

$$\mathbf{e}_a(t) = (Y_{ws}(t) - F_a(t))^2, \forall t \in T \quad (18a)$$

$$\mathbf{e}_b(t) = (Y_{ws}(t) - F_b(t))^2, \forall t \in T \quad (18b)$$

The loss differential function  $d$  is defined as:

$$d(t) = \mathbf{e}_a(t) - \mathbf{e}_b(t) \quad (19)$$

From this new set, the sample mean of the differential  $\bar{d}$  is obtained. To determine if a deviation from zero of this value is statistically significant, this value is divided by an estimation of the standard deviation of the differential series.

Next, the autocovariance  $\gamma$  of each lag up to  $h$  is calculated:

$$\gamma(b) = \text{Cov}(d(t), d(t+b)) \quad (20)$$

where Cov is the covariance function. Then, the variance is estimated as [31]:

$$v_d = \frac{\gamma(0) + 2\sum_i \gamma(i)}{N_T}, \forall i \in H \quad (21)$$

where  $H = [1 \dots h]$ , which represents the set of lags of the autocovariance. Finally, the DM-test output is calculated as:

$$DM_0 = \frac{\bar{d}}{\sqrt{v_d}} \quad (22)$$

This test is adjusted using the Harvey adjustment, which corrects the test for heavy-tailed distributions of the loss differential. This correction is used because the dataset contains outliers with respect to the normal distribution function, given the nature of wind speed; therefore, the normal distribution function does not accurately describe the loss differential distribution [31]:

$$DM_1 = \sqrt{\frac{(N_T + 1 - 2b + N_T^{-1}b(b-1))}{N_T}} DM_0 \quad (23)$$

The heavier tails are illustrated in the Forecast Error Histogram appendix.

The  $p$ -value, also represented simply as  $p$ , is obtained by calculating the student's-t Cumulative Distribution Function of  $DM_1$ :

$$p = \int_{-\infty}^{DM_1} \frac{\Gamma(\frac{K+1}{2})}{\sqrt{K\pi}\Gamma(\frac{K}{2})} \left(1 + \frac{t^2}{K}\right)^{-\frac{K+1}{2}} dt \quad (24)$$

where  $\Gamma(\cdot)$  is the Gamma function,  $t$  is the integration variable and  $K$  is the number of degrees of freedom [41]. For this test,  $K = N_T - 1$ .  $p$  represents the probability that the null hypothesis is true. In this work, if  $0.05 \leq p < 0.10$  the significance is regarded as weak and if  $p < 0.05$  the forecast models are regarded as statistically significantly different; in either of these cases, the null hypothesis is rejected. The null hypothesis is not rejected in any other case.

Given that a specific time horizon requirement in a microgrid application depends on factors such as available storage and the cost functions of other resources, the forecast methods will be evaluated at the prediction of one step ahead, such that it is possible to realise the model evaluation between any combination of two models. While it is possible to increase the time horizon as done in our previous work [12], all of these methods continuously update the forecast to always be one step ahead as time progresses.

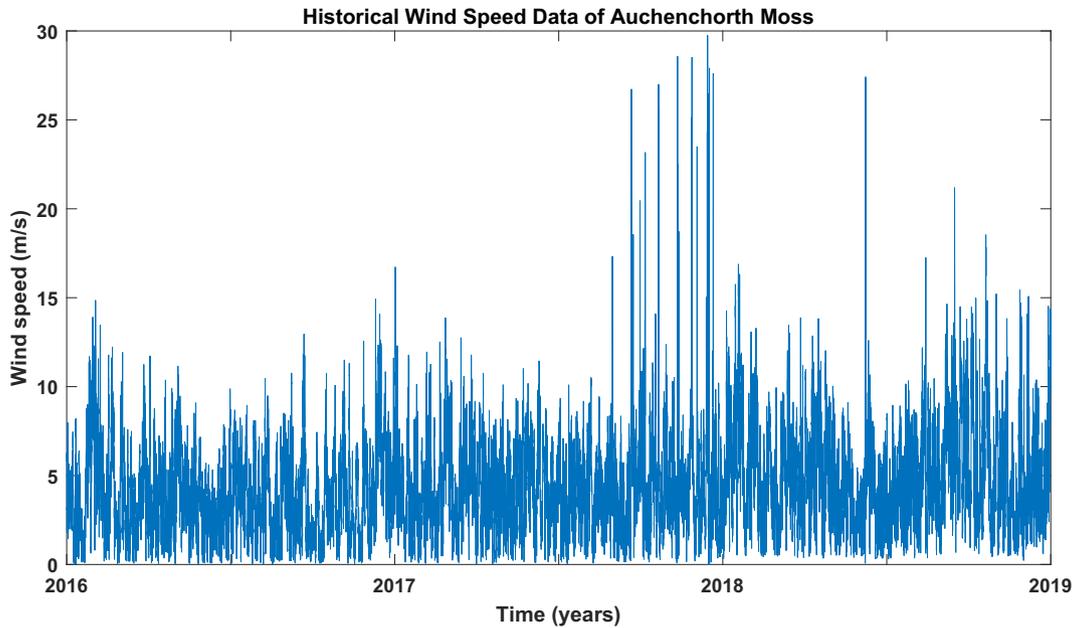
## 5 | TEST CASE

### 5.1 | Wind speed data

The wind speed data used in this work is the historical wind speed data of three years, 2016, 2017, 2018 and 2019, taken every 30 min, at the Auchencorth Moss Atmospheric Observatory in Scotland, UK. The original dataset was obtained from Ref. [43]. The samples in this dataset are measured in  $m/s$ , with a total of 58,655 samples.

For this study, the samples with values equal or higher than  $30m/s$  were removed, with the remaining dataset having 58,448 samples, of which 16,499 represent the data for 2016, 17,014 for 2017, 17,514 for 2018 and 7628 for 2019, which is 99.6% of the original dataset. This set is composed of 97.45% of the possible samples, given that there are samples missing in this set. No form of extrapolation was used to fill the gaps given that the natural wind speed variations is enough to treat the samples as consecutive. For reference, 87 samples of this set are above  $15m/s$  (0.1485%) and 58,568 (99.8515%) are between  $0m/s$  and  $15m/s$ . The variation of the wind speed over this period is illustrated in Figure 4.

This set was selected, among other reasons, because it is freely available and has a good resolution compared to others, which average for days or months. Additionally, the set is sufficiently long to allow the statistically significantly difference test, which relies on the variance of the error in a time series and becomes more reliable with more data. Finally, this location provides data applicable for microgrid applications with onshore wind resources, as opposed to large offshore wind farms, which have different environmental conditions.



**FIGURE 4** Auchencorth Moss wind speed, South East Scotland ( $55^{\circ}47'36''$  N,  $3^{\circ}14'41''$  W). This dataset represents 99.6% of the total data available for this period

## 5.2 | Forecast model training

Each of the models was fitted, optimised or trained according to the wind speed data and the models described in previous sections. The hardware used for this study includes the Durham University super computer known as Hamilton. One node of Hamilton was used, comprising 24 CPU cores with 2x Intel Xeon E5-2650v4 model processors at 2.6 GHz and 64 GB RAM. Other hardware used for training includes a desktop with an i7-6700 CPU at 3.40 GHz with 16 GB of RAM and a desktop with an i5-7500 CPU at 3.40 GHz with 8 GB of RAM. The software for training and fitting is MATLAB R2019a (9.6.0.1099321) 64-bit and the DM-test was done in the Scientific Python Development Environment, Spyder 4.1.5, with Python 3.8.5 64-bit.

The NARNET model architectures were trained with different number of delay sizes ranging from 20 (10 h) up to 672 (2 weeks), number of neurons from 5 to 265 and the three activation functions discussed previously, with the most relevant ones presented in the results section. The selection of these parameters was based on our previous work [12] and time required to realise the training and performance of the model.

The ARMA model, used as point of comparison, was optimised for the order  $\mathcal{P} = 5$  and  $\mathcal{Q} = 30$ , compared with Ref. [13] that used orders  $\mathcal{P} = 2$  and  $\mathcal{Q} = 1$ , and following the autocorrelation of the dataset, this is considered enough to maximise the performance of the ARMA model. The AR model was optimised for different sizes of delay, varying from 1 to 240. The ARMA and AR models are fitted using the data from 2016 to 2018 and the data from 2019 is left for error testing. For the case of the NARNET, the subsets are selected at random, as explained earlier. A total of 95 realisations were done for the single layer architecture, 50 realisations for the parallel architecture and 18, for the series architecture. The difference in the amount of realisations done is due to the training time and performance in terms of RMSE. However, each combination of architecture and activation function was realised at least 6 times to account for randomness in the NARNET training. A total of 405 forecast models were computed, and every pair combination was evaluated with the DM-test; the number of realisations are summarised in Table 1.

**TABLE 1** Summary of forecasts model realisations

Model	Architecture	Number of realisations
NARNET	Single	95
	Series	18
	Parallel	50
AR	-	240
ARMA	-	1
Persistence	-	1

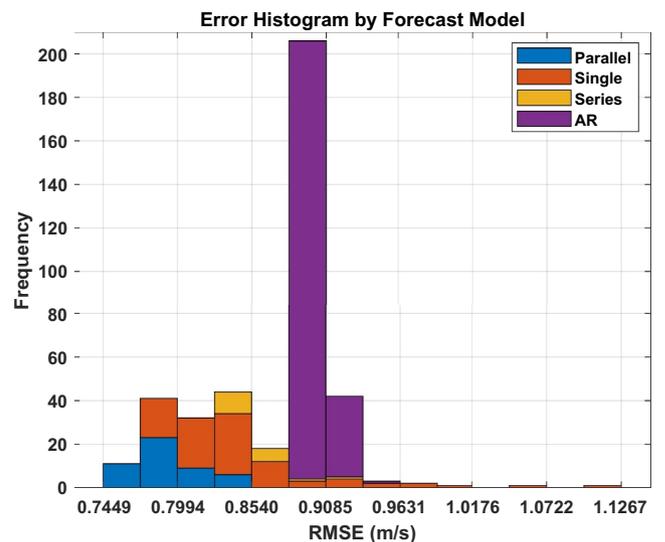
## 6 | DISCUSSION

The RMSE of all the realisations varies between 0.74,485  $m/s$  and 1.1267  $m/s$ . Most of the models performed better than the persistence model at 0.9437  $m/s$  RMSE as expected, except for 7 realisations, all of which correspond to the combination of single layer with the Tansig activation function which is the best architecture found in our previous work in Ref. [12], although, with a different dataset. However, other runs of the same NARNET with variations in the size of delay and number of neurons performed much better with regards to accuracy.

The results in terms of RMSE are shown as a histogram to illustrate the variations between the same forecast model, which are shown in Figures 5 and 6. The results of the DM-test are shown in Figures 8 and 9. The RMSE results of the best individual realisation and corresponding  $p$ -values are shown in Tables 2 and 3.

The best realisations of each NARNET architecture, AR model, ARMA model and the persistence model are shown in Table 2 with their respective  $p$ -values from the DM-test. While the null hypothesis is not rejected for the comparison between the parallel and series NARNET model, the size of the  $\omega$  matrices of the parallel NARNET is in total eight times smaller than the series model  $\omega$  matrix as described in Figure 9. Table 2 also shows that the parallel NARNET model has an RMSE reduction between 3% and 21%, compared to the rest of the analysed forecast models.

In Figure 5, the realisations are grouped by NARNET architecture and compared with the AR model. It is notable that all the AR model realisations are grouped within two bins of the histogram, even when the delays range up to 240 samples, equivalent to 5 days of data. For the case of the NARNET models, it is notable that the distribution that



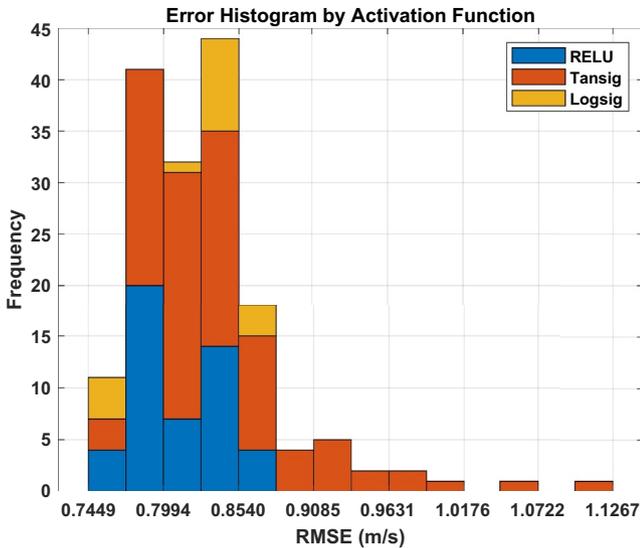
**FIGURE 5** Histogram of the total root mean squared error (RMSE) by forecast model. The persistence and auto regression moving average (ARMA) models are omitted as they only have one iteration each

accounts for the lowest RMSE is the parallel architecture and that the series architecture has the largest variation.

In Figure 6, the same results of Figure 5 for the NARNET realisations are shown, grouped in a histogram by the activation function. It can be seen that the RELU function seems to have the best performance in terms of distribution. However, looking at the most accurate forecast model realisation in Figure 9, it can be noted that the best activation function in terms of RMSE is the Logsig, while the Tansig has the highest variation in performance, being present in all bins.

Table 3 is built in the same way as Table 2 for the best realisations of each NARNET activation function. In this case, it can be seen that the  $p$ -values show that the models are statistically significantly different for every test except between the RELU and the Logsig comparison, with a weak statistical difference.

To demonstrate all the  $p$ -values for all the possible DM-tests, Figures 8 and 9 are presented. However, Figure 7 is presented first to illustrate the information that is included in those figures. The following observations apply for the DM-



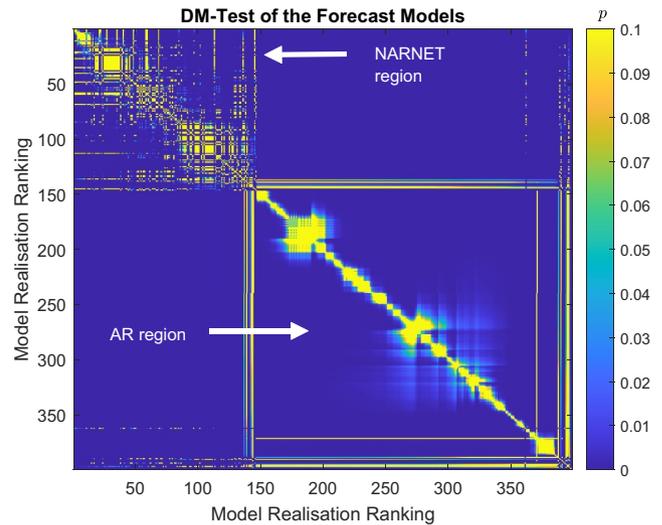
**FIGURE 6** The persistence model, AR model and auto regression moving average (ARMA) model are omitted, as they do not have activation functions. Tansig refers to the Hyperbolic Tangent Sigmoid function. Logsig refers to the Logistic Sigmoid function

P-value from DM-test for forecast models						
	Region	NARNET models		AR models		
Region	Performance Ranking	1	2	3	4	
NARNET models	1	1	0.1	0.04	0.001	0.74
	2	0.1	1	0.08	0.06	0.8
AR models	3	0.04	0.08	1	0.01	0.9
	4	0.001	0.06	0.01	1	0.91
		0.74	0.8	0.9	0.91	RMSE

**FIGURE 7** Illustrative table for the presentation of DM-test results

test figures: **a)** the performance ranking goes from the lowest to the highest RMSE, following natural numbers. **b)** Each square in the figure encodes the  $p$ -value from the DM-test between the row and column pair of models. **c)** The diagonal is always 1 because a forecast model is compared with itself (100% probability of no difference). **d)** The colour of each square goes from 0 to 0.1, and above 0.1 the colour remains yellow, which means no statistical difference. **e)** Any square that is not yellow indicates a statistical difference between the pair of models. **f)** Any square in blue of any shade has  $p$ -value below 0.05 and therefore a statistical significant difference between the models. **g)** In general, the NARNET models outperform the AR models, separating them in two regions in Figure 8.

In Figure 8, the  $p$ -value results of every combination of two forecast realisation for the DM-test are shown. The model realisations are ranked from the lowest RMSE to the highest in both axes while the colour indicates the  $p$ -values, where yellow represents a  $p$ -value equal or higher than 0.1, or in other words, no statistically significant difference. The blue colour indicates that the difference in performance in RMSE is statistically significant; therefore, the difference in performance is due to difference in accuracy of the models rather than randomness in the data. It can be verified that the NARNET outperforms the AR in terms of RMSE, forming regions indicated by the arrows. Additionally, this figure illustrates that the NARNET models in general are either statistically significantly different or not at all from the high contrast in colour among them compared with the smoother transition in  $p$ -values in the AR region. Figure 8 also illustrates that in most cases the NARNET models are statistically significantly different from the AR



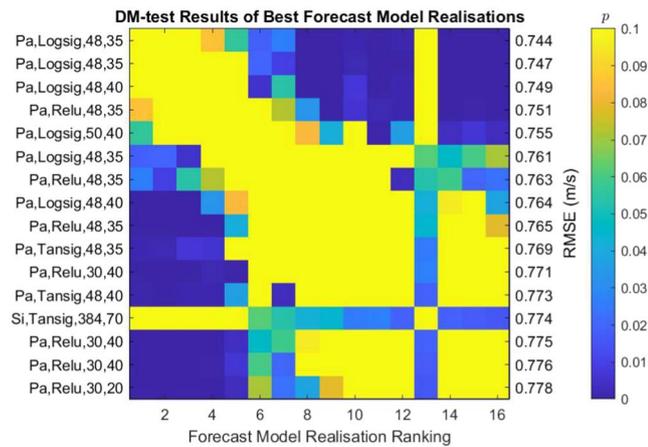
**FIGURE 8** Entire DM-test  $p$ -value results for every combination of the forecast model realisation by colour. The colours show the probability  $p$  that the null hypothesis is true. A  $p < 0.05$  indicates statistical significant difference,  $0.05 < p < 0.1$  indicates weak statistical difference and  $p = 0.1$  indicates no statistical difference, shown in yellow. The realisations are ranked from lowest RMSE to highest, as shown in the horizontal and vertical axis. The blue regions show that the NARNET region is in general at a lower RMSE than the AR models with an statistical significance difference

**TABLE 2** DM-test  $p$ -values between the best realisation of each forecast model. Si refers to Single, Se to Serial, and Pa to parallel NARNET, AR to auto-regression (AR), auto regression moving average (ARMA) to AR with moving average, and base to persistence model. The models are sorted by root mean squared error (RMSE), shown in the second row

Model	Pa	Si	Se	AR	ARMA	Base
RMSE	0.7448	0.7730	0.8289	0.8721	0.888	0.9437
Pa	1	0.277	8.9E-10	1.7E-17	2.57E-17	0
Si	0.277	1	1.06E-7	8.16E-19	8.73E-20	7.32E-43
Se	8.9E-10	1.06E-7	1	2.53E-7	4.56E-8	3.41E-134
AR	1.7E-17	8.16E-19	2.53E-7	1	1.69E-5	2.17E-118
ARMA	2.57E-17	8.73E-20	4.56E-8	1.69E-5	1	7.91E-113
Base	0	7.32E-43	3.41E-134	2.17E-118	7.91E-113	1

**TABLE 3** DM-test  $p$ -values between the best realisation of each activation function of the non-linear auto-regression Network (NARNET), sorted by RMSE

RMSE	DM	Logsig	RELU	Tansig
0.7448	Logsig	1	0.0875	0.0012
0.7519	RELU	0.0875	1	0.006
0.7694	Tansig	0.0012	0.006	1



**FIGURE 9** Entire DM-test results by colour for the best 16 forecast models realisations ranked from lowest RMSE to highest, as shown in the horizontal and right axis. Each colour indicates the probability  $p$  that the null hypothesis is true. A  $p < 0.05$  indicates statistical significant difference,  $0.05 < p < 0.1$  indicates weak statistical difference and  $p > 0.1$  indicates no statistical difference, shown in yellow. The left axis details the model's architecture, activation function, size of  $D$  and number of neurons per hidden layer

models, except for those with close performance in RMSE. A yellow diagonal is shown across the figure as it shows the comparison of a realisation with itself, and therefore there is no difference calculated.

Figure 9 shows in more detail the top 16 forecast model realisation  $p$ -values from the DM-test by colour, all of which are NARNET models, describing the specific configuration of the model in the left axis and the RMSE performance in the right axis. The same ranking from top to bottom of the left axis is from left to right in the horizontal axis. Given that the top 3 models are parallel NARNET models with the Logsig activation function with identical RMSE performance for the first

two decimal places it is expected that those have no statistically significant difference among them.

The best performing model has an RMSE 0.74485 m/s, delay size of 48, or one day of memory, with parallel hidden layer with 35 neurons, which implies 3360 weights in total for both of the hidden layers. It is also notable that the Single layer with Tansig activation function model, that was found to be best model in our previous work, in this case, has a delay of 384, equivalent to 8 days of memory, and 70 neurons in the hidden layer, which implies an internal matrix of 26,880 weights has 0.773 RMSE. The performance of this forecast model in terms of the expected power calculation is included as an appendix.

As a forecast model developed for distributed control, which has the trade-off from not having to external variables in exchange to increased reliability and privacy of the controller, there could be weather uncertainties independent of wind speed. However, the correlation of the errors of this realisation with rainfall from the same location is calculated at 0.06, calculated with 5000 samples from 2018.

Moreover, the method is tested with several years of data, covering all of the normal climatic scenarios. Therefore, it can be concluded that uncertain climatic scenarios do not affect this realisation of the NARNET model although the accuracy of the NARNET model is affected by sudden changes in the wind speed trend, such as in the case of severe and extreme weather conditions.

## 7 | CONCLUSION AND FUTURE WORK

In this work, wind speed forecast models suited for real-time distributed control applied to microgrids with distributed wind generation were studied and analysed using historical wind speed data from Auchencorth Moss. It was found that the NARNET model with the parallel architecture, delay size of 48 and 35 neurons per layer, has the highest accuracy in this regard, with an overall performance of 0.74485 m/s RMSE for short-term wind speed forecasting. It was also found that within this architecture, the best performance is achieved by the Logsig activation function. This function has a statistical significant difference with the ReLU and Tansig activation functions of 0.0875 and 0.006, respectively, with the same architecture.

Additionally, it was found that the model with the highest accuracy is equivalent in statistical significant difference

with the NARNET with a single layer and a delay size of 384 and 70 neurons in the hidden layer. This finding indicates a reduction of 87.50% in the required number of weights and therefore of computational resources required for training and execution.

This study illustrates the importance of not only determining the best forecast model in terms of minimal errors but also the importance of verifying that the difference in performance between two forecast models is actually due to an actual superior model either by accuracy or by efficient use of computational resources. This method, therefore, can equally be applied to other systems with distributed control and local forecasting requirements to ensure selection of the best model in terms of least errors and computational performance.

The NARNET is considered a suitable forecast model for distributed control in microgrids given that it relies on historical data that can be stored locally, which means that the rest of the control system is not subjected to dependence of other external signals, which could represent single points of failure in terms of control reliability.

Evaluation of the forecast models using statistical significant test supports the selection of such models. Adding the comparison in terms of computational cost could improve the selection of forecast models in applications where optimal use of computational resources is required, for example, in case of distributed control architectures. In the context of microgrid distributed control, this could be applied to electricity price, weather and demand. An optimal selection of the forecast models contributes to the accurate, optimal and fast response of the control systems applied to microgrids.

Future work includes further exploring artificial intelligence-based methods for forecast requirements in microgrid distributed control and for other types of renewable generation and local electricity demand as well as increasing the robustness of the evaluation method to validate the selection of the forecast models developed.

## NOMENCLATURE

$P_{wt}$	Wind turbine power Output
$V_{wt}$	Wind speed
$V_{min}$	Minimum generation wind speed
$V_{rated}$	Rated wind speed
$V_{cut}$	Cut-out wind speed
$C_p$	Coefficient of performance
$P_{max}$	Maximum wind turbine power output
$\mathbb{E}$	Expected power generation
$i$	Time index variable
$\tau$	Period of generation
$\tau_{max}$	Duration of generation period
$v_{w_i}$	Wind speed realisation
$Y_{ws}$	Historical wind speed
$t$	Time
$F$	Wind speed forecast model
$e$	Wind speed forecast error
$\mathcal{W}$	ARMA model parameters
$\mathcal{P}$	AR order of the ARMA model
$\mathcal{Q}$	MA order of the ARMA model

$W$	Weights of AR model
$d_{AR}$	Delay size of AR model
$D_{AR}$	Set of delays for AR model
$T$	Set of time samples in historical data
$N_T$	Total available samples in historical data
$\omega$	Neuron weight matrix
$B$	Neuron bias matrix
$D$	Neuron delay vector
$f$	Activation function
$A_l$	ANN $l$ layer input
$l$	ANN layer index variable
$j$	ANN delay size
$q$	ANN input size
$k$	Number of neurons in layer
$e$	Squared forecast error
$d$	Loss differential function
$\gamma$	Autocovariance
$h$	Maximum lag for DM-test
$H$	Set of lags for DM-test
$v_d$	Variance estimation for DM-test
$DM_0$	DM-test output
$DM_1$	DM-test output with Harvey adjustment
$t$	Integration slack variable
$K$	Degrees of freedom
$p$	Probability of null hypothesis

## ACKNOWLEDGEMENT

This work is supported by the CONACYT-SENER energy sustainability programme with scholarship number 447877.

## CONFLICTS OF INTEREST

The authors declare no conflict of interest.

## DATA AVAILABILITY STATEMENT

The dataset used in this study is available from archive. ceda.ac.uk. The datasets used where the Auchencorth Moss Atmospheric Observatory annual-half-hourly meteorology for 2016, 2017 and 2018 and can be found at [https://data.ceda.ac.uk/badc/deposited/2017/auchencorth\\_moss\\_met/data](https://data.ceda.ac.uk/badc/deposited/2017/auchencorth_moss_met/data). The result datasets are available on request from the corresponding author. The DM-test code used in this work is based on the code found at <https://github.com/johntwk/Diebold-Mariano-Test> and the equations from Ref. [31].

## ORCID

Marcos Eduardo Cruz-Victorio  <https://orcid.org/0000-0003-2604-176X>

Behzad Kazemtabrizi  <https://orcid.org/0000-0003-0916-4063>

Mahmoud Shabbazi  <https://orcid.org/0000-0002-6057-3228>

## REFERENCES

1. Wan, C., et al.: A hybrid approach for probabilistic forecasting of electricity price. *IEEE Trans. Smart Grid.* 5(1), 463–470 (2014). <https://doi.org/10.1109/TSG.2013.2274465>

2. Alhussein, M., Haider, S.I., Aurangzeb, K.: Microgrid-level energy management approach based on short-term forecasting of wind speed and solar irradiance. *Energies*. 12(8), 1487. <https://doi.org/10.3390/en12081487>. <https://www.mdpi.com/1996-1073/12/8/1487>
3. Wu, L., Shahidepour, M.: A hybrid model for day-ahead price forecasting. *IEEE Trans. Power Syst.* 25(3), 1519–1530 (2010). <https://doi.org/10.1109/TPWRS.2009.2039948>
4. Wang, Z., et al.: Short-term wind speed forecasting based on information of neighboring wind farms. *IEEE Access*. 8, 16760–16770 (2020). <https://doi.org/10.1109/ACCESS.2020.2966268>
5. Mosbah, H., El-hawary, M.: Hourly electricity price forecasting for the next month using multilayer neural network. *Can. J. Electr. Comput. Eng.* 39(4), 283–291 (2016). <https://doi.org/10.1109/cjee.2016.2586939>
6. Faraji, J., et al.: Optimal day-ahead self-scheduling and operation of prosumer microgrids using hybrid machine learning-based weather and load forecasting. *IEEE Access*. 8, 157284–157305 (2020). <https://doi.org/10.1109/ACCESS.2020.3019562>
7. Hubicka, K., Marcjasz, G., Weron, R.: A note on averaging day-ahead electricity price forecasts across calibration windows. *IEEE Trans. Sustain. Energy*. 10(1), 321–323 (2019). <https://doi.org/10.1109/TSTE.2018.2869557>
8. Uniejewski, B., Weron, R., Ziel, F.: Variance stabilizing transformations for electricity spot price forecasting. *IEEE Trans. Power Syst.* 33(2), 2219–2229 (2018). <https://doi.org/10.1109/TPWRS.2017.2734563>
9. Jeong, G., Park, S., Hwang, G.: Time series forecasting based day-ahead energy trading in microgrids: mathematical analysis and simulation. *IEEE Access*. 8, 63885–63900 (2020). <https://doi.org/10.1109/ACCESS.2020.2985258>
10. Eseye, A.T., et al.: Machine learning based integrated feature selection approach for improved electricity demand forecasting in decentralized energy systems. *IEEE Access*. 7, 91463–91475 (2019). <https://doi.org/10.1109/ACCESS.2019.2924685>
11. Cruz Victorio, M.E., Kazemtabrizi, B., Shahbazi, M.: Distributed real-time power management in microgrids using multi-agent control with provisions for fault-tolerance. In: 2020 IEEE 29th International Symposium on Industrial Electronics, pp. 108–113. ISIE) (2020)
12. Cruz Victorio, M.E., Kazemtabrizi, B., Shahbazi, M.: Price forecast methodologies comparison for microgrid Control with multi-agent systems. In: 2021 IEEE Madrid PowerTech (2021)
13. Wang, Y., Wang, D., Tang, Y.: Clustered hybrid wind power prediction model based on arma, pso-svm, and clustering methods. *IEEE Access*. 8, 17071–17079 (2020). <https://doi.org/10.1109/ACCESS.2020.2968390>
14. Toubeau, J.-F., et al.: Capturing spatio-temporal dependencies in the probabilistic forecasting of distribution locational marginal prices. *IEEE Trans. Smart Grid*. 12(3), 2663–2674 (2021). <https://doi.org/10.1109/TSG.2020.3047863>
15. Vu, D.H., et al.: Short-term forecasting of electricity spot prices containing random spikes using a time-varying autoregressive model combined with kernel regression. *IEEE Trans. Ind. Inf.* 15(9), 5378–5388 (2019). <https://doi.org/10.1109/TII.2019.2911700>
16. Golmohamadi, H., Keypour, R.: Stochastic optimization for retailers with distributed wind generation considering demand response. *J. Mod. Power Syst. Clean Energy*. 6(4), 733–748 (2018). <https://doi.org/10.1007/s40565-017-0368-y>
17. Ahmed, A., Khalid, M.: An intelligent framework for short-term multi-step wind speed forecasting based on functional networks. *Appl. Energy*. 225, 902–911 (2018). <https://doi.org/10.1016/j.apenergy.2018.04.101>. <https://www.sciencedirect.com/science/article/pii/S0306261918306664>
18. Wang, J., Wang, S., Yang, W.: A novel non-linear combination system for short-term wind speed forecast. *Renew. Energy*. 143, 1172–1192 (2019). <https://doi.org/10.1016/j.renene.2019.04.154>. <https://www.sciencedirect.com/science/article/pii/S096014811930638X>
19. Fan, F., Bell, K., Infield, D.: Probabilistic real-time thermal rating forecasting for overhead lines by conditionally heteroscedastic autoregressive models. *IEEE Trans. Power Deliv.* 32(4), 1881–1890 (2017). <https://doi.org/10.1109/TPWRD.2016.2577140>
20. 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>
21. Lydia, M., et al.: Linear and non-linear autoregressive models for short-term wind speed forecasting. *Energy Convers. Manag.* 112, 115–124 (2016). <https://doi.org/10.1016/j.enconman.2016.01.007>
22. Pearre, N.S., Swan, L.G.: Statistical approach for improved wind speed forecasting for wind power production. *Sustain. Energy Technol. Assessments*. 27, 180–191 (2018). <https://doi.org/10.1016/j.seta.2018.04.010>. <https://www.sciencedirect.com/science/article/pii/S221313881730512X>
23. Kim, G.G., et al.: Prediction model for pv performance with correlation analysis of environmental variables. *IEEE J. Photovoltaics*. 9(3), 832–841 (2019). <https://doi.org/10.1109/JPHOTOV.2019.2898521>
24. Aranizadeh, A., et al.: Wind turbine and ultra-capacitor harvested energy increasing in microgrid using wind speed forecasting. *Engineering Science and Technology. Int. J.* 22(5), 1161–1167 (2019). <https://doi.org/10.1016/j.jestech.2019.08.006>. <https://www.sciencedirect.com/science/article/pii/S2215098618317221>
25. Zhang, L., Dong, Y., Wang, J.: Wind speed forecasting using a two-stage forecasting system with an error correcting and nonlinear ensemble strategy. *IEEE Access*. 7, 176000–176023 (2019). <https://doi.org/10.1109/ACCESS.2019.2957174>
26. Akhtar, I., et al.: Average monthly wind power forecasting using fuzzy approach. *IEEE Access*. 9, 30426–30440 (2021). <https://doi.org/10.1109/ACCESS.2021.3056562>
27. Zhang, Y., et al.: Wind speed prediction research considering wind speed ramp and residual distribution. *IEEE Access*. 7, 131873–131887 (2019). <https://doi.org/10.1109/ACCESS.2019.2940897>
28. Li, W., et al.: A new distributed energy management strategy for smart grid with stochastic wind power. *IEEE Trans. Ind. Electron.* 68(2), 1311–1321 (2021). <https://doi.org/10.1109/TIE.2020.2970627>
29. Zhang, C., et al.: An improved elm model based on ceemd-lzc and manifold learning for short-term wind power prediction. *IEEE Access*. 7, 121472–121481 (2019). <https://doi.org/10.1109/ACCESS.2019.2936828>
30. Shahzad, M.N., Kanwal, S., Hussanan, A.: A new hybrid arar and neural network model for multi-step ahead wind speed forecasting in three regions of Pakistan. *IEEE Access*. 8, 199382–199392 (2020). <https://doi.org/10.1109/ACCESS.2020.3035121>
31. Harvey, D., Leybourne, S., Newbold, P.: Testing the equality of prediction mean squared errors. *Int. J. Forecast.* 13(2), 281–291 (1997). [https://doi.org/10.1016/S0169-2070\(96\)00719-4](https://doi.org/10.1016/S0169-2070(96)00719-4)
32. Ranganai, E., Sigauke, C.: Capturing long-range dependence and harmonic phenomena in 24-hour solar irradiance forecasting: a quantile regression robustification via forecasts combination approach. *IEEE Access*. 8, 172204–172218 (2020). <https://doi.org/10.1109/ACCESS.2020.3024661>
33. Xiang, L., Deng, Z., Hu, A.: Forecasting short-term wind speed based on icwt-lssvm model optimized by bird swarm algorithm. *IEEE Access*. 7, 59333–59345 (2019). <https://doi.org/10.1109/ACCESS.2019.2914251>
34. Mutavhatsindi, T., Sigauke, C., Mbuva, R.: Forecasting hourly global horizontal solar irradiance in South Africa using machine learning models. *IEEE Access*. 8, 198872–198885 (2020). <https://doi.org/10.1109/ACCESS.2020.3034690>
35. Wang, J., Li, Q., Zeng, B.: Multi-layer cooperative combined forecasting system for short-term wind speed forecasting. *Sustain. Energy Technol. Assessments*. 43, 100946 (2021). <https://doi.org/10.1016/j.seta.2020.100946>. <https://www.sciencedirect.com/science/article/pii/S2213138820313746>
36. Sahoo, S.K., Sinha, A.K., Kishore, N.K.: Control techniques in ac, dc and hybrid ac–dc microgrid: a review. *IEEE Journal of Emerging and Selected Topics in Power Electronics*. 6(2), 738–759 (2018). <https://doi.org/10.1109/jestpe.2017.2786588>
37. Han, Y., et al.: Mas-based distributed coordinated control and optimization in microgrid and microgrid clusters: a comprehensive overview.

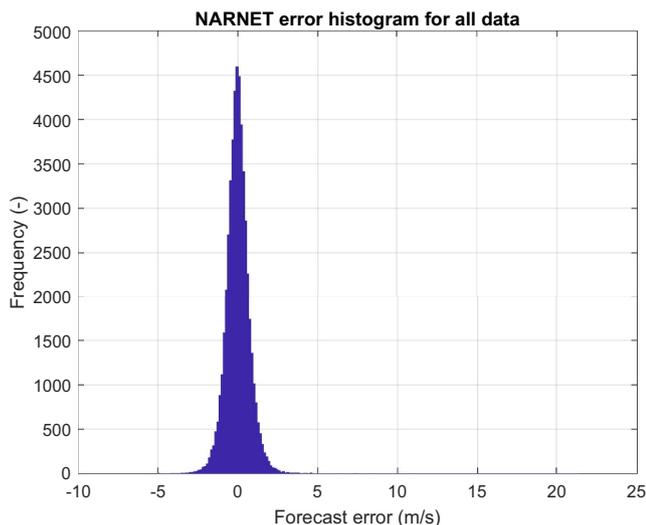
- IEEE Trans. Power Electron. 33(8), 6488–6508 (2018). <https://doi.org/10.1109/TPEL.2017.2761438>
38. Cruz Victorio, M.E., Kazemtabrizi, B., Shahbazi, M.: Real-time cost optimisation for power management in microgrids using multi-agent control. In: 2019 9th International Conference on Power and Energy Systems, pp. 1–6. ICPEES (2019)
  39. Dou, C., et al.: Mas-based hierarchical distributed coordinate control strategy of virtual power source voltage in low-voltage microgrid. IEEE Access. 5, 11381–11390 (2017). <https://doi.org/10.1109/ACCESS.2017.2717493>
  40. Dou, C., et al.: Mas-based distributed cooperative control for dc microgrid through switching topology communication network with time-varying delays. IEEE Syst. J. 13(1), 615–624 (2019). <https://doi.org/10.1109/JSYST.2017.2726081>
  41. Zörnig, P.: Probability Theory and Statistical Applications: A Profound Treatise for Self-Study. De Gruyter Textbook, Princeton University Press (2016)
  42. Bhattacharyya, S., et al.: Convolutional Neural Networks: A Bottom-Up Approach, pp. 21–50. De Gruyter, Berlin (2020). <https://doi.org/10.1515/9783110670905-002>. <https://www.degruyter.com/document/doi/10.1515/9783110670905/html>
  43. Auchencorth moss atmospheric observatory (au): Annual half-hourly meteorology since 1995 near edinburgh : [https://data.ceda.ac.uk/badc/deposited2017/auchencorth\\_moss\\_met/data](https://data.ceda.ac.uk/badc/deposited2017/auchencorth_moss_met/data), accessed: 28-10-2021

**How to cite this article:** Cruz-Victorio, M.E., Kazemtabrizi, B., Shahbazi, M.: Statistical evaluation of wind speed forecast models for microgrid distributed control. IET Smart Grid. 1–16 (2022). <https://doi.org/10.1049/stg2.12073>

## APPENDIX

### Forecast error histogram

This section presents the error distribution of the forecast methods against all wind speed data. This illustrates that the distributions have heavier tails than the normal distribution as

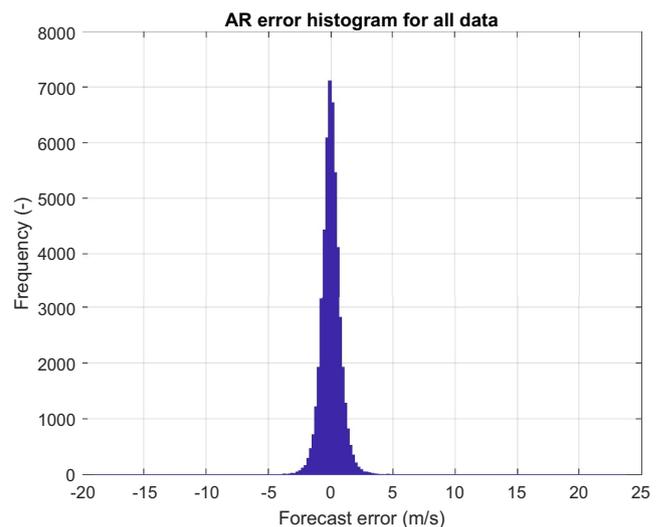


**FIGURE A1** Error histogram to validate the most accurate non-linear auto-regression network (NARNET) forecast method for all the data

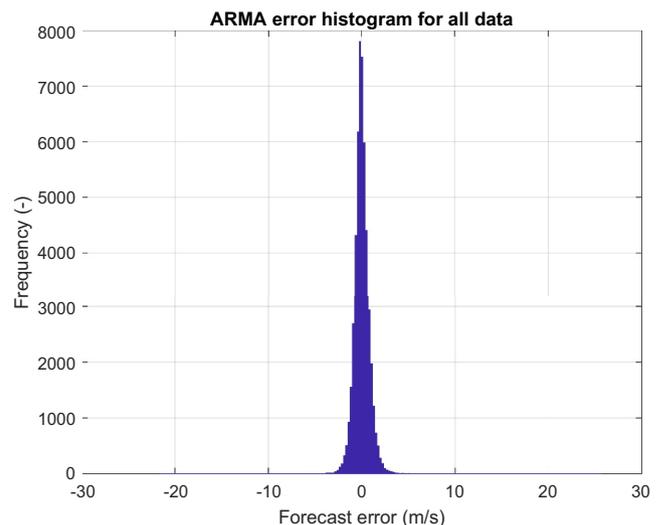
shown in previous studies on wind speed forecast; however, in all cases it can be seen from the shape of the distributions that the forecast models studied in this paper are validated.

Figures A1, A2 and A3 illustrate the error histograms for the best realisation of the NARNET, which has parallel hidden layers with a delay size of 48, equivalent to a day of memory and 35 neurons in each hidden layer, the AR model with 240 weights and the ARMA model with order  $\mathcal{P} = 5$  and order  $\mathcal{Q} = 30$ . The histograms are done for all of the wind speed data to illustrate the heavier tails in the error distribution and validate the use of the Harvey adjustment.

Figure A4 show the error distributions of the forecast models, using the 2019 wind speed data to illustrate the suitability of the models over the test data subset.

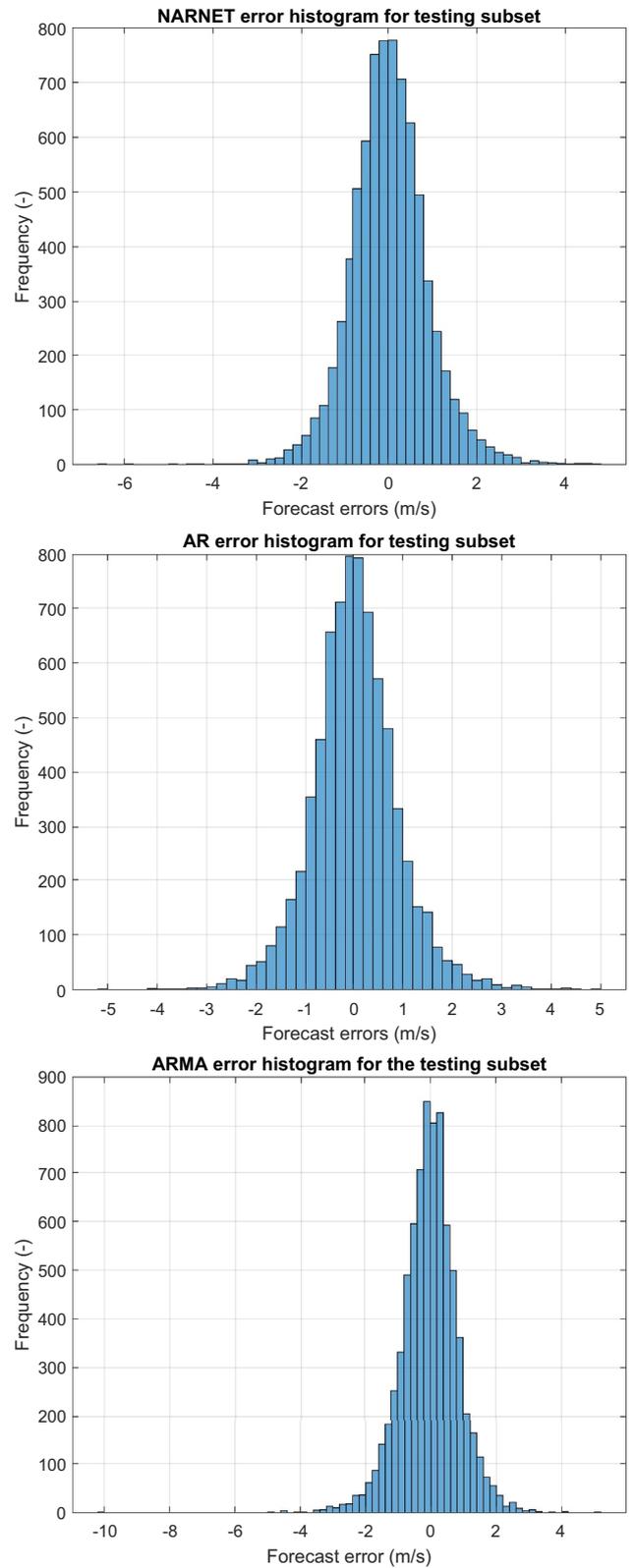


**FIGURE A2** Error histogram to validate the most accurate auto-regression (AR) forecast method for all the data



**FIGURE A3** Error histogram to validate the auto regression moving average (ARMA) forecast method for all the data

**FIGURE A4** Error histogram to validate the forecast methods with the testing data



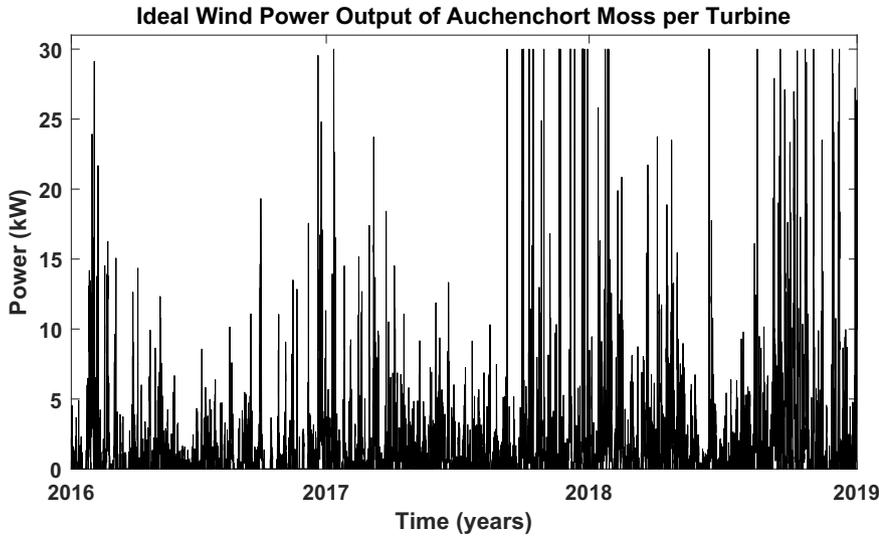


FIGURE A5 Ideal power generation per turbine from the historical data

### Wind power calculation

To help putting in the context of microgrids, this appendix will briefly cover how the forecast results shown previously compare in terms of the expected wind power generation that could be extracted from the site, given ideal conditions. First, in Figure A5 the total wind power that could be generated using the wind speed data from 2016, 2017 and 2018 is shown.

For the case of the expected power output  $\mathbb{E}$ , following Equations (1) and (2) and considering  $V_{\min} = 3.5$  m/s,

$V_{rated} = 15$  m/s,  $V_{cut} = 35$  m/s,  $P_{\max} = 30$  kW and  $C_p = 0.0089$  kW (m/s)<sup>-3</sup>,  $\mathbb{E}_1 = 21.9053$  kW per turbine using the historical data of the 23/10/2018 as input,  $\mathbb{E}_2 = 20.8627$  kW per turbine using the model with the lowest RMSE, and  $\mathbb{E}_3 = 21.0047$  kW for the case of the model ranked 13 in Figure 9, which does not have statistical significant difference with the rank 1 model but at a higher computational cost, given that it requires more weights to achieve this. The calculation of  $\mathbb{E}$  can be simplified to the mean power output calculated with the wind speed from each case.