applied sciences

MDPI

Article

Wind Turbine Prognosis Models Based on SCADA Data and Extreme Learning Machines

Pere Marti-Puig *, Alejandro Blanco-M., Moisès Serra-Serra and Jordi Solé-Casals *

Data and Signal Processing Group, University of Vic—Central University of Catalonia, 08500 Catalonia, Spain; alejandro.blanco@uvic.cat (A.B.-M.); moises.serra@uvic.cat (M.S.-S.)

* Correspondence: pere.marti@uvic.cat (P.M.-P.); jordi.sole@uvic.cat (J.S.-C.)

Abstract: In this paper, a method to build models to monitor and evaluate the health status of wind turbines using Single-hidden Layer Feedforward Neural networks (SLFN) is presented. The models are trained using the Extreme Learning Machines (ELM) strategy. The data used is obtained from the SCADA systems, easily available in modern wind turbines. The ELM technique requires very low computational costs for the training of the models, and thus allows for the integration of a grid-search approach with parallelized instances to find out the optimal model parameters. These models can be built both individually, considering the turbines separately, or as an aggregate for the whole wind plant. The followed strategy consists in predicting a target variable using the rest of the variables of the system/subsystem, computing the error deviation from the real target variable and finally comparing high error values with a selection of alarm events for that system, therefore validating the performance of the model. The experimental results indicate that this methodology leads to the detection of mismatches in the stages of the system's failure, thus making it possible to schedule the maintenance operation before a critical failure occurs. The simplicity of the ELM systems and the ease with which the parameters can be adjusted make it a realistic option to be implemented in wind turbine models to work in real time.

Keywords: wind turbine; fault diagnosis; renewable energy; Extreme Learning Machines (ELM); feature engineering; normal behaviour models

check for updates

Citation: Marti-Puig, P.; Blanco-M., A.; Serra-Serra, M.; Solé-Casals, J. Wind Turbine Prognosis Models Based on SCADA Data and Extreme Learning Machines. *Appl. Sci.* **2021**, *11*, 590. http://doi.org/10.3390/app11020590

Received: 29 November 2020
Accepted: 5 January 2021
Published: 9 January 2021

1. Introduction

One of the biggest economic impacts on the Levelized Cost of Energy (LCOE) of wind turbines (WT) is related to the operations and maintenance (O&M) tasks. For instance, in offshore plants it is estimated that this cost is around 25% to 30% [1]. Consequently, different strategies have been proposed to decrease it by reducing the percentage of O&M costs. An overview of these strategies is presented in References [2–4]. If abnormal behaviour can be detected before failures occur, WT inactivity can be reduced and this will have a direct impact on the LCOE. There are different approaches to this task, and in this work a data-driven Normal Behaviour Model (NBM) will be presented.

Most of the wind turbines have a SCADA system in place [5] which allows for to monitoring and collection of data, so therefore there is no need to implement any additional hardware/software for this purpose. However, the size of the available data is huge because the monitoring is done over many variables and parts or subparts of the turbine. Some examples include pressure data from the gearbox lubrication system, temperature data from many parts of the system, current and voltage values or tower vibration, among others. All these values usually come as 5-min or 10-min average values, together with the basic statistics (minimum value, maximum value and standard deviation) for each interval. All in all, around 300 variables are collected, making it unfeasible to analyse them manually.

One of the best options is to use machine learning algorithms to perform some type of exploration and/or analysis on the data. This can be done, for example, for predicting failures in the wind turbines using SCADA data [6]. Usually, the approaches for condition-monitoring based on SCADA data are grouped into Signal Trending techniques, Artificial Neural Networks (ANN) and Physical Models.

Traditionally, when SCADA data is used to forecast the health status of WTs, classification models are used to predict failure states. Two critical problems can make classification models not reliable: (1) On the one hand, the classes are extremely unbalanced with an over-representation of the normality state and a very small representation of the failure state (warning/alarm), which is specific to each subsystem. (2) On the other hand, the labelling of the data needed to train the models is not always reliable, as explained before, because it is done automatically and contain many errors.

This is why in this paper, the use of normality models is proposed for predicting one or more variables generated by a subsystem, using the other variables in the subsystem and detecting that the prediction is out of the norm when the subsystem deteriorates. A very efficient computational way to develop normality wind turbine subsystem models is presented. It is based on Single-hidden Layer Feedforward Networks (SLFNs), a type of single-layer artificial neural network, which are trained using the Extreme Learning Machine (ELM) technique in order to optimise the developing process. This strategy has been implemented using ELM and results have been compared using other well-known machine learning approaches such as PLS, SVM or DANN. Two main experiments are carried out to demonstrate the utility of the normality models. The first one (model of the generator subsystem) is used to exemplify how to build a SLFN normality model trained using the ELM strategy. It is a simple model used to highlight how the dimension of the network is fundamental in order to avoid over-training and is also used to explore the two different approaches for deriving the model: deriving one for each individual turbine of the plant, or grouping all the turbines together to derive a global model for the whole plant. This study makes it possible to check the differences between the models obtained from these two different approaches, for the same subsystem. In the second experiment, the model for the gearbox is detailed for the entire group of turbines of the plant (therefore, a global model approach), and then the model is used to evaluate the behaviour of all the turbines of the plant. The gearbox is the device used to multiply the low speed but high torque rotation of the rotor to an optimal speed in which the generator converts the maximum amount of mechanical energy produced by the wind into electric energy. That energy transformation stresses the gearbox's gears due to the difference of input torque to the opposite generator torque at the output. Therefore, parts of the gearbox suffer from fatigue and experience an increase of temperature that reduces the lubrication effectiveness. The malfunctioning of the gears, especially in the early stages, is often difficult to detect. It should be noted that the gearbox is the component with the highest failure downtime. Even if the gearbox manufacturing technology is mature and reliable, this is a subsystem prone to breakdowns and failures within a 5 year operational period due to the harsh working conditions it is subjected to, and it must be replaced [7]. In addition to the cost of replacement, there is a system shutdown involved with the failure of the gearbox that can last for a long period of time because this is one of the slowest systems to repair. A replacement of the gearbox can cost up to 14.5% of the maintenance cost of the wind turbine [3]. Unsurprisingly, predicting gearbox failures therefore becomes a priority.

Different studies, such as Reference [8], conclude that ANN outperforms traditional regression models. In Reference [9], ANN models are combined and derived from SCADA data, and in Reference [10] the authors implement a non-linear auto-regressive neural network to model the normal temperature of five different gearbox bearings. Other approximations can be found in References [11,12]. The interesting advantage of using ELM is that the training of SLFNs is done very quickly through the calculation of a Moore-Penrose inverse matrix and through simple matrix algebra manipulations. Because the models can be trained and tested very efficiently, modifying their size and obtaining an objective measure of their performance on a test data set becomes a feasible and simple task. In this way, a grid-search can be carried out for the parameters of the model, thereby optimising inputs, targets and the size of the model itself. As will be shown later, sizing the model correctly is often a key step to maximise its predictive ability while avoiding over-fitting.

The rest of the paper is organised as follows—in Section 2, the approach is briefly presented, describing the characteristics of the SCADA database, the normalisation method and the details of the ELM technique used for building the models. In Section 3, the experimental results are presented, comparing models of individual turbines and models of the ensemble of turbines of the wind plant. Finally, Section 4 is dedicated to discussions and conclusions.

2. Materials and Methods

2.1. Model of Normality

In previous works [13,14] we have applied ML techniques in the field of predictive maintenance of wind turbines and have been doing it from a classification point of view. The objective was to determine the health-related states of the turbines and their different systems and/or subsystems, paying special attention to pre-fault states, which we want to detect in the turbines we test. The main problem encountered with this strategy is the labelling of the data in terms of the health states. Obtaining correctly-labelled data is a very complicated step. Occasionally, a turbine can be found that has suffered a certain type of failure where the data is easy to label when referring to the health of some of its subsystems, but in a general way the labelling is a complicated process that needs a lot of human supervision and expert decision-making. Leaving aside the fact that different experts may differ in their labelling criteria, it is necessary to deal with the fact that there are few cases of failure in each subsystem, and furthermore they are difficult to extrapolate between wind turbines from different manufacturers, operating in different parks and in a regime of conditions that can vary greatly.

However, we have a lot of data from turbines and systems that work well, so it is feasible to establish a model of normality by means of regression strategies. For a given subsystem, the strategy consists on predicting a variable using the rest of the variables of the subsystem and detecting that it is out of normality in the event of a breakdown. Once again, here experience in choosing the variable is very helpful. The advantage of this strategy is that the models can be trained with all the turbines in the park and with data corresponding to all the operating modes. The training is carried out on the first half of the data from all the turbines which are working correctly, and although it is possible that some incorrectly labelled points are present in some cases, the over-representation of the cases where the turbines are working well makes the model very precise. The effectiveness of this strategy depends on: (i) having a system capable of predicting the model's variable with maximum effectiveness (minimum RMSE) when being in the normal (healthy) state; and (ii) with enough sensitivity to detect deviations in this variable when the predictions begin to fail due to the deterioration of the systems. This phenomenon is observed by exploring small time windows of data. In the regression lines of the normality state, this phenomenon is observed in a change of slope of the line and in the dispersion of the predicted points, which also move away from the 45-degree line. Because it can be observed before the failure occurs, the health status of turbines can be foreseen, performing what is therefore known as a prognosis.

Summarising, the key point of the method that will be presented in this work is to train a model using data collections recorded in different operating regimes when the turbines were working properly. For the model to be accurate enough, it is important to emphasise that the records must be sufficiently rich and varied to represent different wind and weather conditions. The underlying idea is that when the system is operating correctly, the estimation and measurement of the target signal follow the trained model and are practically the same, so that the plot of the measurements against the estimates is a 45-degree line. However, when systems deteriorate, this line is disturbed, thereby changing the slope as measurements and estimates begin to diverge.

2.2. Selection of the Modelling Technique

The particularities of the data, mainly summarised by the presence of inconsistent labelling and by the extreme imbalance in the representation of the classes, suggests the use of a normality model associated with a regression technique. Applying ELM to the problem is based on the following properties: (i) The model of the subsystem to be developed must be valid for all operating regimes of all wind turbines throughout the wind farm. Because a lot of data is available to build the model, records from several years can be used, ensuring that all operating regimes of all WTs are well represented; (ii) In addition, due tot he nature of the data, there is an over-representation of the normal state versus the state with warnings/alarms (state of malfunction). The ELM technique used in its original formulation is characterised by providing the solution that approximates all the points of the training by performing the minimum mean square error. This property is extremely interesting when we have an over-representation of the normal state, as the ELM model picks it up very well. This makes ELM very robust even with the inclusion of a small proportion of mislabelled data (i.e., if a few points of data corresponding to the fault state are included in the training of the normal state); (iii) The facility of interpretation and the speed of training makes the ELM paradigm the best technique for this solution.

Subsequent experiments using Support vector machines, Partial least square and Deep artificial neural networks, all of them in their regression forms, will confirm the suitability of this choice, which is presented in detail in the next subsection.

2.3. Regression Analysis Using ELM

A Single-hidden Layer Feedforward Neural network is trained as a regressor following the Extreme Learning Machines framework [15,16]. The parameters of the SLFN network that have to be determined are the so-called output weights, connecting the hidden layer neurons to the outputs, and the number of hidden neurons (H). These parameters can be assigned in a random way, following the ELM's main idea. Therefore, computing the SLFN output weights has been transformed to a (over-determined) linear problem, which can be solved in one step by means of the Moore-Penrose pseudoinverse [15,16].

Focusing on the regression problem, the L signals x_l of the WT subsystem are organised as the input vector $\mathbf{x}_i$, where the features are taken in the time-slot i: $\mathbf{x}_i = [x_0 \cdots x_{L-1}]^T$. The goal is to predict the target variable t_c in the same time-slot, with $C \geq 1$.

The matrix $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_i \cdots \mathbf{x}_N]$ of size $L \times N$ contains the set of features that will be employed to train the model given the N observations with the aim of predicting a generic number C of the target variable's values. The vector $\mathbf{t}_c = [t_1 \cdots t_i \cdots t_N]^T$ contains the N registers of the target variable c corresponding to the measures in $\mathbf{X}$. The vectors $\mathbf{t}_c$ are organised in the matrix $\mathbf{T} = [\mathbf{t}_1 \cdots \mathbf{t}_c \cdots \mathbf{t}_C]^T$. The input weight matrix $\mathbf{W}$ will be of size $L \times H$, where H is the number of hidden neurons, and L is the number of features (input variables). The element w_{lh} of $\mathbf{W}$ is the weight that connects the feature l with the hidden node h, and each one of the hidden nodes has a bias b_h.

Taking into account that the signals in vector $\mathbf{x}_i$ are the inputs of the SLFN for the time-slot i, the output values in the internal hidden nodes of network (prior to applying the activation function) can be written as $\mathbf{x}_i^T\mathbf{W} + \mathbf{b}^T$. Considering the sigmoid as the activation function, the output values in the hidden nodes wil be $sig(\mathbf{x}_i^T\mathbf{W} + \mathbf{b}^T)$, where $sig(\mathbf{A})$ applies the sigmoid function to each element a_{ij} of $\mathbf{A}$.

Then, when the resulting $1 \times H$ row vector is multiplied by the output weight matrix $\mathbf{B}$ of size $H \times C$, the resulting $1 \times C$ row vector will contain the prediction of the targets $\hat{\mathbf{y}} = [\hat{y}_1 \cdots \hat{y}_C]$. Therefore, when grouping the equations for all the N observations, we obtain:

$$sig(\mathbf{X}^T\mathbf{W} + \mathbf{u}\mathbf{b}^T)\mathbf{B} = \mathbf{T}, \tag{1}$$

$\mathbf{u}$ being the $N \times 1$ all-ones vector. In order to simplify the notation let us define the $N \times H$ matrix $\mathbf{H}$ as:

$$\mathbf{H} = sig(\mathbf{X}^T\mathbf{W} + \mathbf{u}\mathbf{b}^T). \tag{2}$$

Then, the equation can be written in a compact form as:

$$\mathbf{HB} = \mathbf{T}. \tag{3}$$

Once H is determined, **W** and **b** are randomly selected from a zero-mean, unit-variance Gaussian distribution function. The only unknown parameter is the matrix **B**. Note that Equation (3) is over-determined, because for ELM networks it is common to choose a number H of hidden nodes which is lower than the number N of classified cases used for training, that is, $N > H$. The output weight matrix **B**, provided that $\left(\mathbf{H}^T\mathbf{H}\right)^{-1}$ is non singular, can be computed as:

$$\mathbf{B} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{T} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{4}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose inverse.

Once **W**, **b** and **B** are defined, the SLFN can then be used to predict the target variables $\hat{\mathbf{y}}$ given a new vector of features **x** as follows:

$$\hat{\mathbf{y}} = sig\left(\mathbf{x}^T\mathbf{W} + \mathbf{b}^T\right)\mathbf{B}. \tag{5}$$

This equation corresponds to the estimation of C targets for a single WT. If we want to estimate a single target ($C = 1$), the equation can be simplified. In that case, the matrix **B** is a vector, β, and we obtain:

$$\hat{y} = sig\left(\mathbf{x}^T\mathbf{W} + \mathbf{b}^T\right)\beta. \tag{6}$$

In the general case in which we estimate the set of subsystems of the M turbines of an entire wind plant, the equation will be as follows:

$$\hat{\mathbf{Y}} = sig\left(\mathbf{X}^T\mathbf{W} + \mathbf{u}\mathbf{b}^T\right)\mathbf{B}. \tag{7}$$

where now **u** is a unit vector of size $M \times 1$ and the matrix $\hat{\mathbf{Y}}$ is of size $M \times C$.

In Figure 1 we can see the simple structure of a SLFN network for the particular case of a WT that uses the variables **x** to predict two target variables at $\hat{\mathbf{y}}$. Note that the normalisation process of the input variables is represented in the figure.

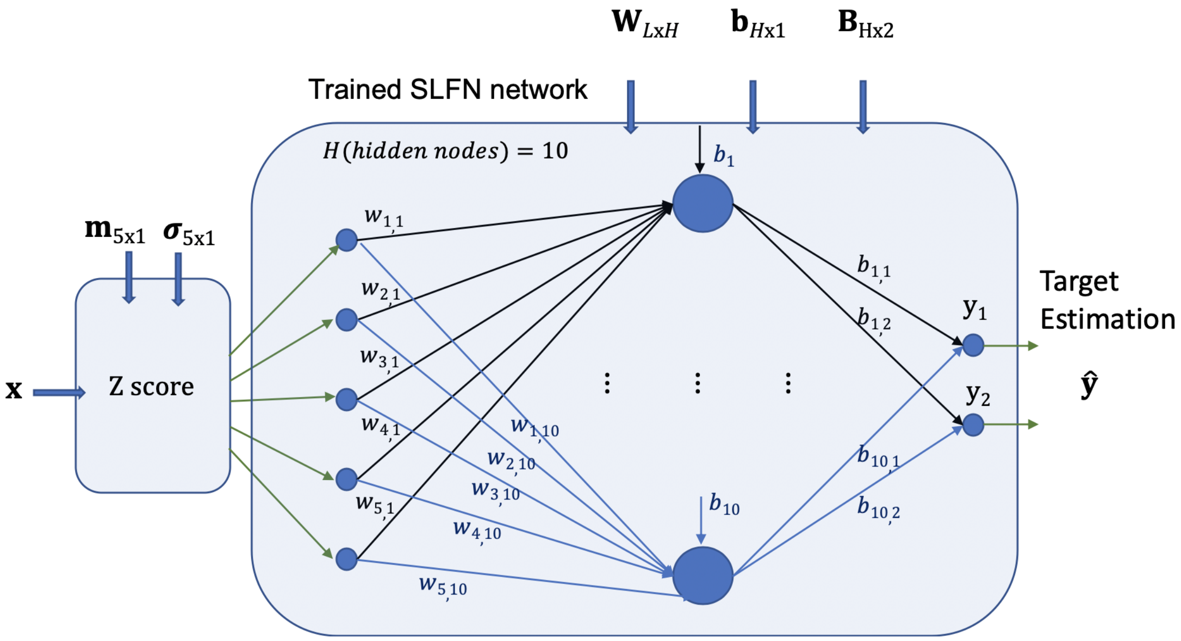


Figure 1. Detail of the structure of the SLFN in terms of the parameters that define it (**W**, **b** and **B**), the vector of input features (**x**), the z-score normalisation process and the output vector ($\hat{\mathbf{y}}$).

Selection of the Network Size

Once the input variables and targets are chosen, sizing the SLFN to fit the models only depends on the choice of the number of hidden nodes H. The optimal selection of H will depend on the number of input signals and the number of targets to be estimated. By properly normalising the signals as explained in Section 2.5.1, there are no further issues related to calculating the Moore-Penrose pseudoinverse matrix.

2.4. Other Regression Methods

To validate the results of the ELM model, three other different modelling techniques will be used and compared with the ELM. The methods selected for comparison with the ELM results are chosen for their popularity and variety. SVM is one of the best known classification methods, so the regression version maintains the same characteristics and potential. PLS is one of the most used regression systems based on the least square, so it is a good candidate for comparison with ELM. Finally, DANN has been chosen to have another model based on neural networks, to see if it could work better than the ELM. They are briefly presented and described in the following subsections.

2.4.1. Support Vector Machines

Support vector machines have been widely used in wind farm condition-monitoring and failure detection [17]. They can be used for linear or non-linear classification or regression purposes and are able to find the optimal hyperplanes by keeping the maximum possible margin between samples of different classes. They use a parameter (slack variable) to control the margin violations that may appear (for example, when the database contains outliers). The most interesting idea behind SVMs is that they increase the dimension of the feature space using non-linear kernels, which can transform the problem into a linearly separable one in the new higher-dimensional space. Among the drawbacks of SVMs, we have to mention the large amount of possible kernels to use and their parameters, making them difficult to adjust, and also the longer time needed to train the models. In our case, we will focus on the capability of SVMs to perform linear/non-linear regression. The main parameters of the model for the experiments are as follows—Gaussian kernel, optimisation of the hyperparameter based on a 10 k-fold cross-validation (CV) with a limit of 5000 iterations. The parameter to be optimised is the minimisation of the log of CV loss.

2.4.2. Partial Least Square

Partial Least Square (PLS) is a multivariate statistical method that analyses the relationship between variables to find a subspace of latent variables that synthesises the predicted or independent variables (X), with the aim of understanding the dispersion of the dependent or observed variables (Y) in a linear way [18]. The most important parameter of the PLS model is the number of components. The rank of X is the maximum number of components, but usually a lower number is used to avoid the description of the noise present in the data and to prevent the problem of collinearity (having a zero determinant for the matrix X). To determine the number of components in our experiment, we will use the method based on calculating the knee of the MSE curve when using cross-validation. The knee of the curve is defined as the point where the curve is best approximated by a pair of lines [19]. This method provides a consistent and mathematically justifiable answer when there is no obvious location of the inflexion point along the curve. In our experiment, the number of components usually turns out to be 2.

2.4.3. Deep Artificial Neural Networks

Deep Artificial Neural Networks (DANN) are very popular in many fields, and have been used in wind farms for failure detection before [20,21]. Among the inconveniences when using DANNs, we have to note their structure itself (number of hidden layers, number of units in each hidden layer), the activation function and the optimisation algorithm (several options available) and the computational time required for the training step.

Another important problem is related to the overfitting effect, which may appear when the structure has a lot of parameters to adjust and a small number of training samples. To avoid overfitting, dropout will be implemented [22] in the experiments of our DANN with 3 hidden layers, used as a regression model. The main parameters of the DANN are: Adam optimizer, max epochs 50, minibatch size 128. The structure of the DANN is as follows: input layer (6 neurons); hidden layer 1 (20 neurons) with ReLu as the activation function and a dropout percentage of 20%; hidden layer 2 (10 neurons) with ReLu as the activation function and a dropout percentage of 10%; hidden layer 3 (5 neurons) with ReLu as the activation function; output layer (1 neuron).

2.5. Experimental Data

An extensive 3-year SCADA database of five Fuhrländer FL2500 2.5MW wind turbine is presented. Data is generated by the wind turbine's SCADA, collected via an Open Platform Communications (OPC), following the IEC 61400-25 format. Therefore, the data is structured as follows: (i) wind turbines are represented with logical devices; (ii) physical systems or subsystems are represented with nodes. Every 5 min, events and statistics indicators are recorded. The reported values for each sensor are: minimum, maximum, mean and standard deviation. The database contains 312 analogous variables from 78 different sensors. All events in the database are originally labelled with one of the following three numbers: '0' indicates normal operation, '1' indicates a warning state (in this case the turbine is working but should be checked as soon as possible) and '2' indicates an alarm state (in this case the turbine is stopped). Alarms are very scarce in all wind turbine databases, because most of the time the turbine is working properly. A warning that is not properly checked and addressed could result in an alarm. Therefore, in our database we merge the two labels ('1' and '2') in the same group, reducing the problem to a 2-classes scenario (operation and failure). The goal will be to detect in advance when it will be that a wind turbine will start working with potential problems, which would lead to a warning or an alarm state. In the experiments, only some of the variables will be used, those related to the system of the analysed subsystem. The database was provided by Smartive (http://smartive.eu) and has been used in other publications [14,23,24].

2.5.1. Data Normalisation

The set of SCADA variables present a very different dynamic range between them. Therefore, a standardisation step is required. For this purpose, a *z-score* normalisation is applied to the data, according to which the mean m_i from each variable s_i is subtracted from that variable and the result is divided by that variable's standard deviation σ_i so that: $x_i = (s_i - m_i)/\sigma_i$. The set of means and standard deviations are stored in the vectors $\mathbf{m}$ and $\mathbf{s}$ respectively since they will be necessary to perform normalisation and denormalisations in the test step.

3. Results

Two main experiments are conducted. In the first one, the generator is modelled. The model is very simple but is useful to illustrate how to use the proposed method. The second experiment involves modelling the gearbox due to the importance of this subsystem in a WT.

In all the experiments, the database was split into two halves. The first one contains the data from the first temporal period of all the turbines, while the second one contains the rest of the time. For building the model of a subsystem, the first set of data was used, selecting the SCADA signals provided by the subsystem to be modelled as well as the alarms associated with it. Then, the data was pre-processed by deleting all the points corresponding to failure/malfunction of the subsystem in any of the turbines of the park. This step was performed based on the available information of the warnings and alarms. Finally, the model was built, predicting one variable using the others.

The test was run using the data corresponding to the second period. Although many different tests can be carried out, for the monitoring of the systems in real time it makes sense to try to predict the target signals from the turbine's data to represent them with the real data on the regression line. This operation can be done point by point or by using short time intervals to improve decision-making in order to be able to generate a line and to evaluate through slopes or dispersion measures if it moves away from the normality model.

3.1. Experiments for the Generator

One of the subsystems that needs to be repaired the most often is the generator (WGEN). According to Reference [25], about 30% of the failures of the generator are of major importance. On the other hand, the WGEN is the subsystem with the smallest number of associated sensors and thus we can use all of them without the need to apply a feature selection step. The main objective of this section is to show the design procedure with a simple model, so therefore the WGEN is a good candidate.

The modelling procedure is implemented by taking the *avg* value of all the sensors of the WGEN and estimating one of the values (target) using the rest of them (input features) at the same time t. The names of the input variables, according to the manufacturer's nomenclature, are wgen_avg_GnTmp_phsB, wgen_avg_GnTmp_phsC, wgen_avg_RtrSpd_IGR, wgen_avg_Spd, and wgen_avg_RtrSpd_WP2035, while the target variable is wgen_avg_GnTmp_phsA. A short description of all these variables is shown in Table 1.

Table 1. Summary of the variables and target used for the generator model.

Generator System Model	
Variable Name	**Description**
wgen_avg_GnTmp_phsB	Average temperature of the generator's winding, phase B
wgen_avg_GnTmp_phsC	Average temperature of the generator's winding, phase C
wgen_avg_RtrSpd_IGR	Average speed of the rotor at the inductive sensor
wgen_avg_Spd	Average speed of the generator
wgen_avg_RtrSpd_WP2035	Average speed of the rotor at the mita-teknik WP2035 monitor
Target Name	**Description**
wgen_avg_GnTmp_phsA	Average temperature of the generator's winding, phase A

3.1.1. Experiments at Individual Turbine Level

In this first experiment, individual WT data is considered. The 5-min data is aggregated into one-hour periods, deleting records with missing data. Then, half of the data is used to train the model and the other half is used to test it. The training step shows that increasing the size of the network leads to a consistent improvement in the performance of the model when evaluated with the same data used for training. In Figure 2, the training performance of networks of size $H = 10$, 20 and 30 is presented in terms of the regression plot. The plot represents the pair drawn by the model output against its associated target. In the ideal case, a 45-degree line would be drawn. The autocorrelation between the estimate and the target, for $H = 10$, 20 and 30, takes the values of $R = 0.99795$, 0.99974 and 0.99989, respectively.

Note that in the training part, the performance increases with the size of the network. As is usual in ELM training (or in general in any type of system with parameters to be tuned), there is a tendency to improve results in the training phase when the size of the network increases. If it increases indefinitely, the network could learn all the training data and work correctly in the training set, but generating an overfitting. As the essential point is that the network works well when used with new data, that is, with data that has not been used for training, the H parameter must be adjusted to the optimal value that collects the essential characteristics of the data. As the essential point is that the network works well when used with new data, that is, with data that has not been used for training, the H parameter must be adjusted to the optimal value that collects the essential characteristics

of the data. The optimum value of H is found using the network in the test data, where performance improves as H increases to a point where it becomes saturated and begins to worsen due to the overfitting effect. A significant advantage of ELMs is that training the models is a very fast process. Therefore the identification of the optimal H parameter is done in a short time.

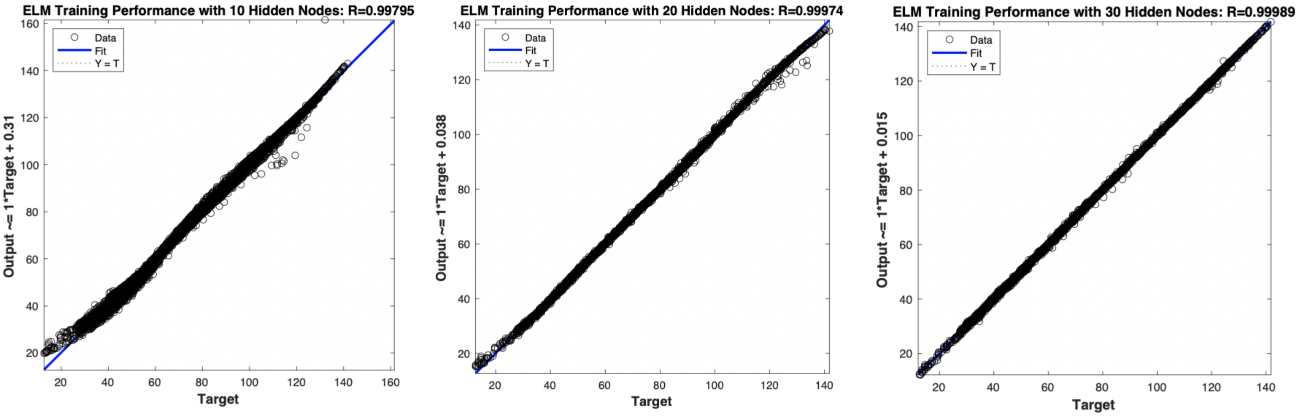


Figure 2. Regression plots of the individual model for the WT#80 in the training step as the size of the network increases ($H = 10$, 20 and 30).

Figure 3 shows the performance of networks of size $H = 15, 20, 25, 30, 35$ and 40 nodes in terms of the regression plot, but now evaluated with the testing data. Although the results of the autocorrelation are still very good, a slight decrease is observed when the size for this model is not close to the optimal one, which experimentally has been found to be $H = 35$.

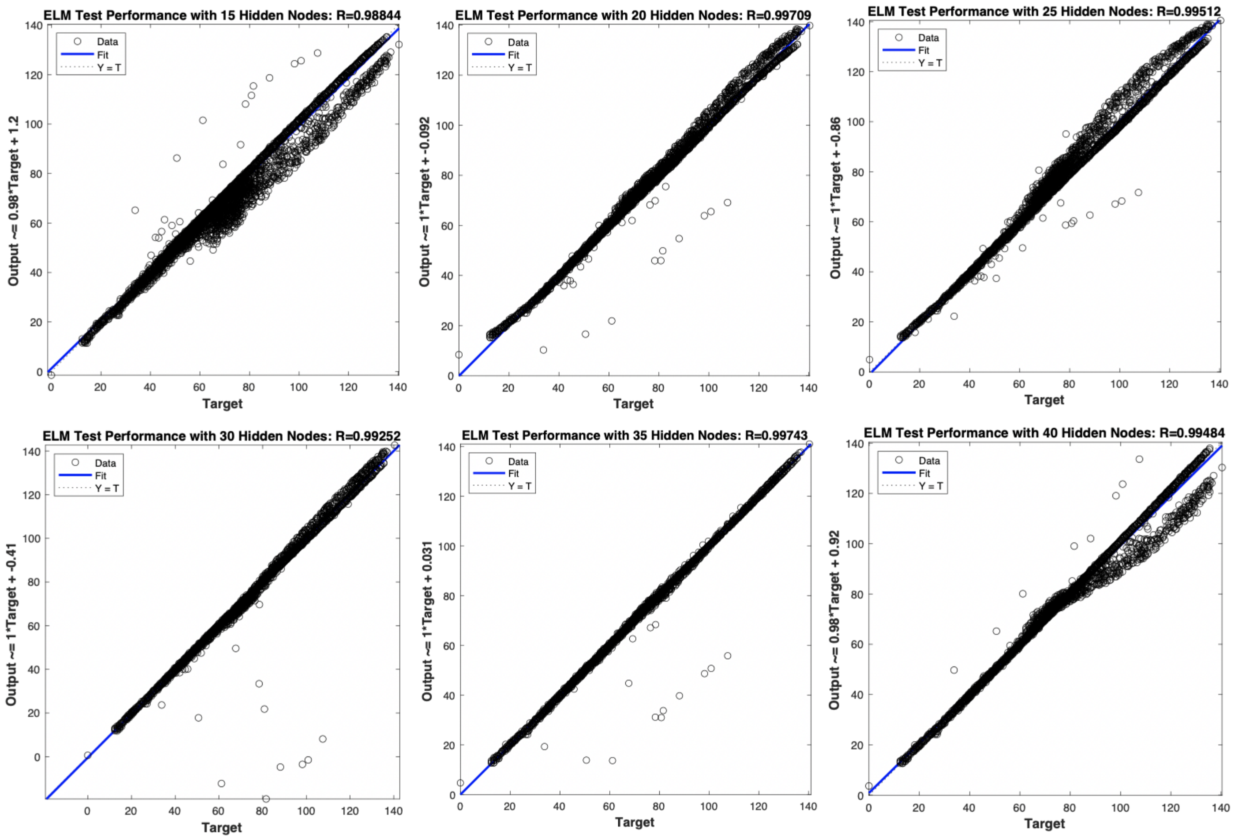


Figure 3. Regression plots of the individual model for the WT#80 in the test step. Values around $H = 35$ obtain optimal results.

It should also be noted that the network has been evaluated with half of the available data. However, the test can be performed using shorter signal periods, which can be in the order of months, weeks or days (see Figure 4 as an example).

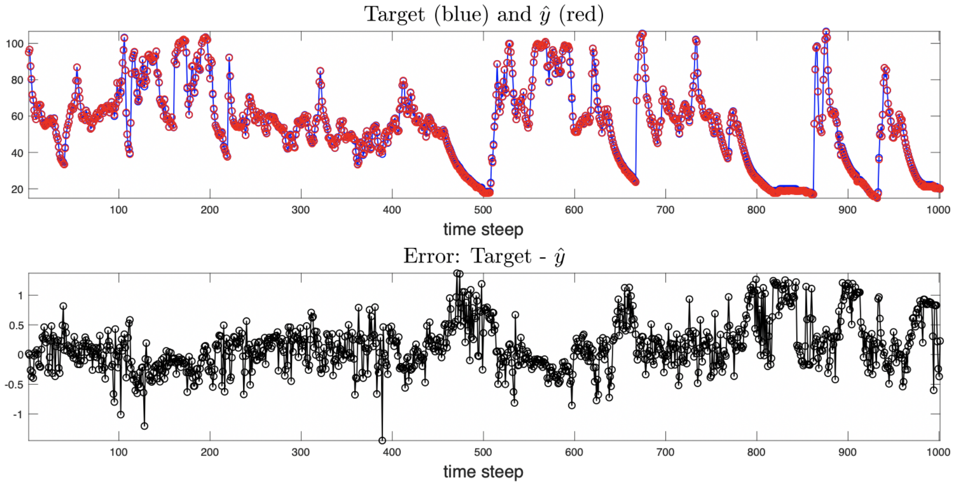


Figure 4. (**Top**) temporal sequence of the target (blue line) and the prediction made by the model (red line) are presented. (**Bottom**) the prediction error is shown.

3.1.2. Experiments for the Ensemble of the Wind Plant Turbines

In this section, instead of modelling the WGEN system turbine by turbine, a model is built for all the turbines in the wind plant. Thus, the same model will be used to test the WGEN system of each turbine. Following the example above, the SCADA data are added at intervals of 1 h and records are removed in case of missing data. The first half of the available data from each turbine is used to build the model and the other half is used to test it. To determine the size of the SLFN, an exploration of the different models obtained for a wide range of H (up to $H = 90$) is performed. Experimentally, the optimal value is H = 43 (see the minimum achieved in the test performance of Figure 5), which is slightly greater than the value obtained in Section 3.1.1 for the single turbine model WT#84 (H = 35). The regression plots of the plant model for all the turbines are shown in Figure 6).

As will be shown, the results obtained with the full park models are significant because they give us an idea of the predictive and generalising capacity that the normality models could have. The advantage of a single model for the whole wind farm is evident because they usually have many turbines (from tens to hundreds or more). In addition, the turbines have many subsystems, so the testing of all the turbines could be done very efficiently using a model of the whole park for each system/subsystem.

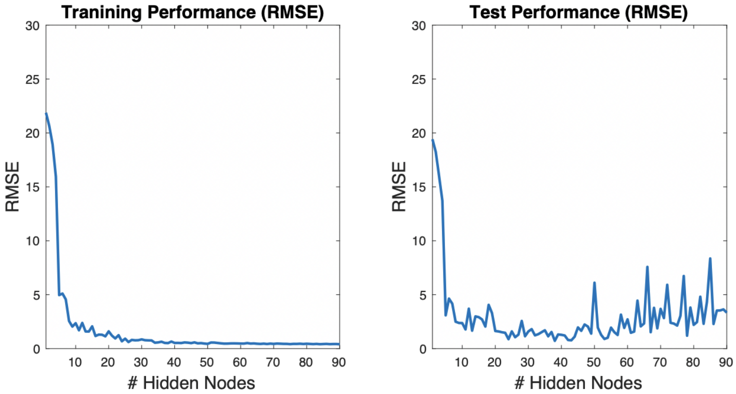


Figure 5. Performance of the model, in terms of the RMSE, for all the turbines of the plant versus the number of hidden nodes, H. On the left, the training results; On the right, the test results.

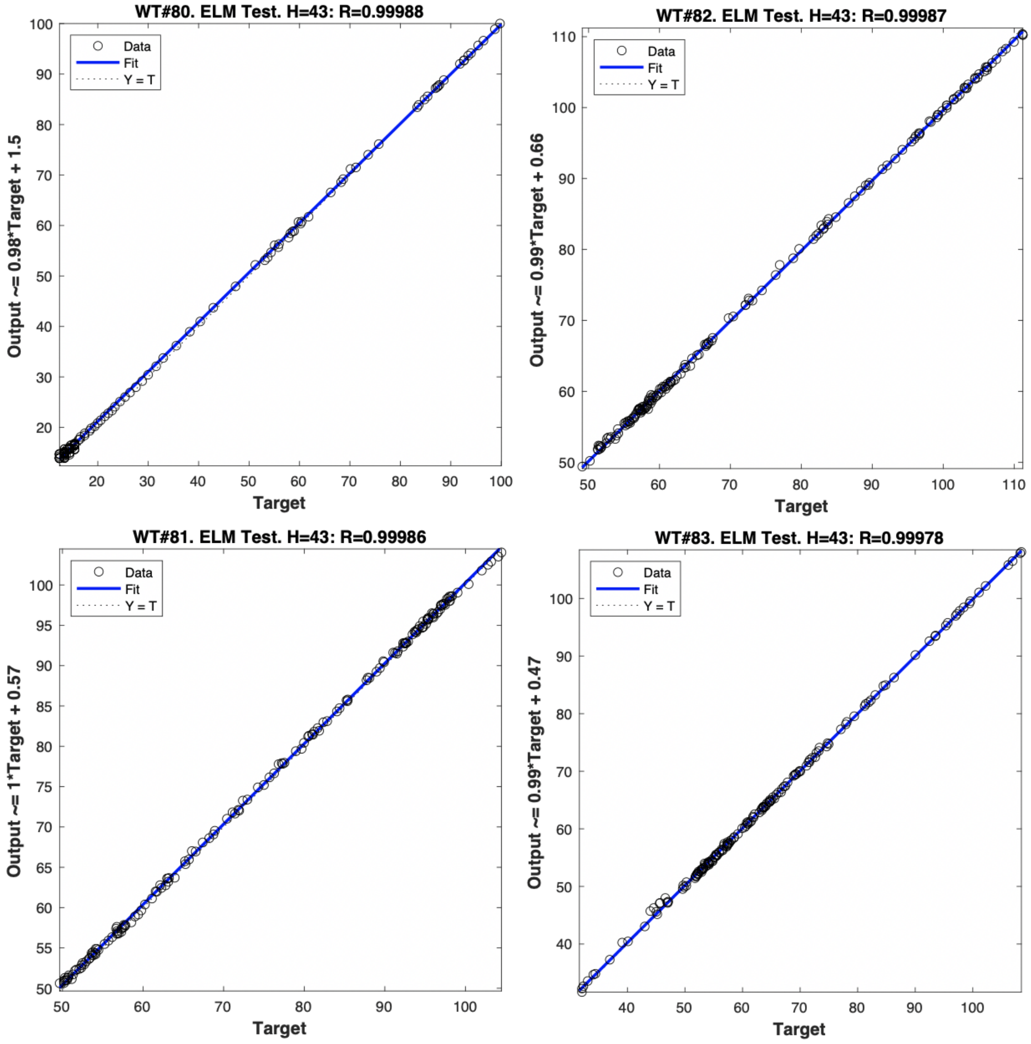


Figure 6. Regression plots of the plant model for turbines WT#80 to WT#83, using the SCADA data generated during one week for the case of $H = 43$.

3.2. Experiments for the Gearbox

Now, a model of the turbine gearbox is analyzed. The main reason for choosing the gearbox is because it is an expensive subsystem with failures that take a long time to repair. According to Reference [25], the gearbox is the subsystem with the highest cost per failure, on average. In addition, it is an expensive process because it requires special equipment such as cranes. Because the replacement failure rate of the gearbox is high and the time needed to repair it is also notably noticeable, the costs derived from the damages of its components represent a high portion of the maintenance costs of wind plants. Therefore, having information about the deterioration of the system's operation is one of the objectives that can most directly affect the reduction of maintenance and operating costs of the wind plant.

The variables summarized in Table 2 are used to build the model. The set of these variables can be determined, as in our case, by an experienced expert who knows the physics of the system. These variables could also be obtained by feature selection algorithms. A list of feature selection algorithms reported in the literature, used in the context of this experiment, is presented in Table 3. However, according to previous works, the variables selected by an expert give excellent results [23], and this is in fact the method used in this study. The goal is to build a model to predict the temperature in one of the axes that turns faster by using different variables measured in other points of the same system and external parameters such as ambient temperature or wind speed.

Table 2. The variables selected by an expert as being relevant for the gearbox operation. Note that to avoid a linear combination of the variables provided by the SCADA system, three variables are artificially generated.

GearBox System Model	
Variable Name	**Description**
wgdc_avg_TriGri_PwrAt/wgdc_avg_TriGri_PF	Gross power output of the turbine, in KW
wtrm_avg_TrmTmp_GbxOil	Temperature of the gearbox oil, in degrees Celsius.
wgen_avg_RtrSpd_WP2035	Speed of the rotor main shaft before gearbox, in revolutions per minute (RPM)
wnac_avg_WSpd1	Wind speed in m/s measured by the anemometer at the wind turbine's nacelle
wtrm_avg_TrmTmp_GbxOil - wnac_avg_ExlTmp	Difference between the temperature of the gearbox oil and the external temperature
wtrm_avg_TrmTmp_GbxBrg151/wtrm_avg_TrmTmp_GbxOil	Ratio between the speed of the rotor main shaft before gearbox in the point 151 divided by the temperature of the gearbox oil.
Target Name	**Description**
wtrm_avg_TrmTmp_GbxBrg152	Average temperature of the gearbox bearing 152, at the high speed shaft (output)

Table 3. Feature selection algorithms used in Reference [23] to identify the relevant variables, as an alternative to expert-based variable-selection.

GearBox System Model	
Algorithm	**Reference**
Mutual Information Feature Selection (MIFS)	Battiti [26]
Conditional Mutual Information (CMI)	Cheng et al. [27]
Joint Mutual Information (JMI)	Yang and Moody [28]
Min-Redundancy Max-Relevance (mRMR)	Peng et al. [29]
Double Input Symmetrical Relevance (DISR)	Meyer and Bontempi [30]
Conditional Mutual Info Maximisation (CMIM)	Fleuret [31]
Interaction Capping (ICAP)	Jakulin [32]

A gearbox wind plant model is built using the SCADA data aggregated in one-hour periods. Again, the records with missing data are deleted, following the same criteria of the previous experiments. Then, the first half of all available data from each of the turbines is selected for training SLFNs of size from $H = 1$ to $H = 150$. Each one of these networks is tested with the other half of the data. The results, in terms of the RMSE, are presented in Figure 7, on the left side. For each one of the networks with different H, the parameters $\mathbf{W}$, $\mathbf{B}$ and $\mathbf{b}$ are stored. With this first analysis, the behaviour of the model and its degree of improvement depending on the size of the network, H, can be seen. To determine the optimal size, the models are evaluated using the test data. The results are shown in Figure 7, on the right side. As shown, the RMSE values obtained are slightly higher than the ones from the training step. Unlike in the previous model, large networks now show a better behaviour. The optimum case is obtained when $H = 92$, and values around or slightly above this would give the models a similar performance.

Once the model is selected, it is tested on each of the turbines. In Figures 8–12, the regression plots of the model are presented. On the right-hand side of the figures, the temporal behaviour of the model is depicted, illustrating the degree of accuracy of the model as well as the error generated. The time interval chosen for this representation was selected randomly, but is then the same for each turbine. In these figures, the target, the output provided by the model ($\hat{y}$) and the error made in the estimation are represented. It can be seen that the turbines follow the model accurately, with the occasional presence of malfunction points. Note that the WT#84 (Figure 12) has more malfunction points than the rest.

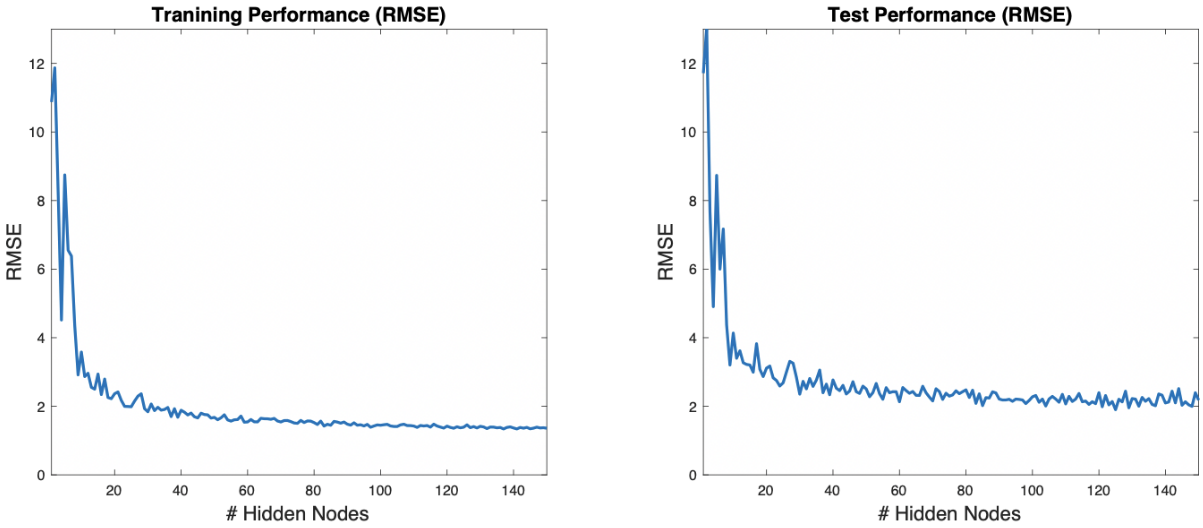


Figure 7. (**Left**) the Root-Mean-Square Error (RMSE) generated by a model of size H during the training step. (**Right**) the RMSE in the testing step for each H.

The regression plots allow us to monitor the performances of the turbines over long periods of time, providing an overview of the model's behaviour in relation to the different turbines in the plant. In the case shown, the period is of two and a half years, which corresponds to the time of the second half of the dataset. The first half has been used to build the model in the training step. During the time interval used for training, the turbines work properly, although repairs or changes of components have been made at some point. Even when including data corresponding to small periods of malfunction, the model ends up adjusting for the cases of good behaviour. This is because comparatively, the number of good cases is much greater than the number of cases of malfunctioning. Consequently, most of the outputs produced by the turbine fit considerably well with the model outputs. Therefore, most points fall on the 45-degree line. The turbines, however, have continued to record alarms and failures in the gearbox subsystem during the test period. Therefore, when a deterioration in performance occurs, some points start to be further away from the 45-degree line occasionally, and the slope of the line changes. After a repair/replacement of the component, the subsystem once again registers a behaviour in line with the originally trained model.

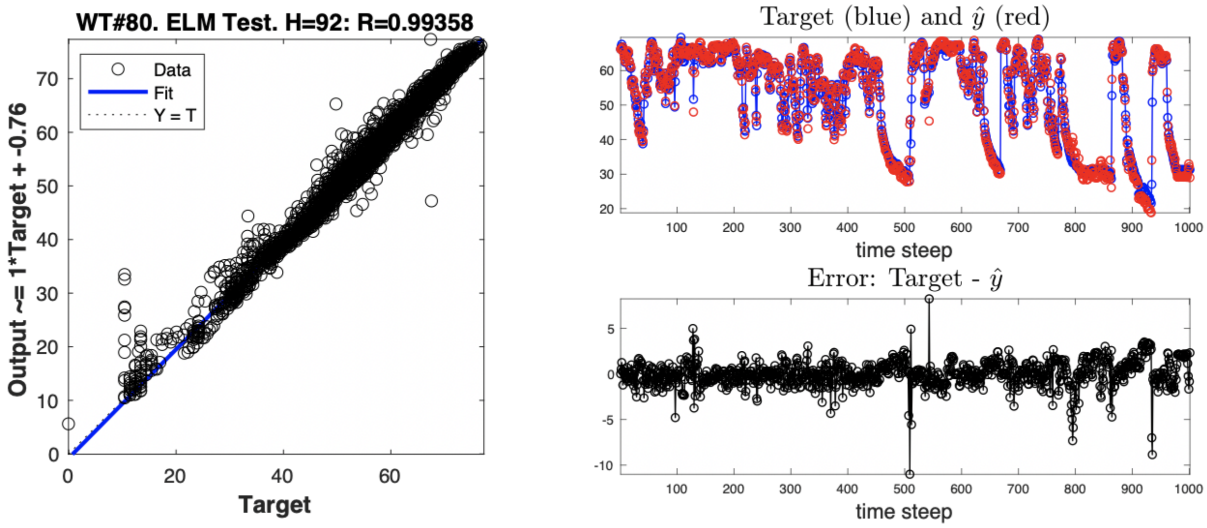


Figure 8. Test of the selected model ($H = 92$) for the WT#80. (**Left**) the regression plot using all the test data. (**Right**) from a temporal window where the target $\hat{y}$ (**top**) and the error (**bottom**) can be observed in more detail.

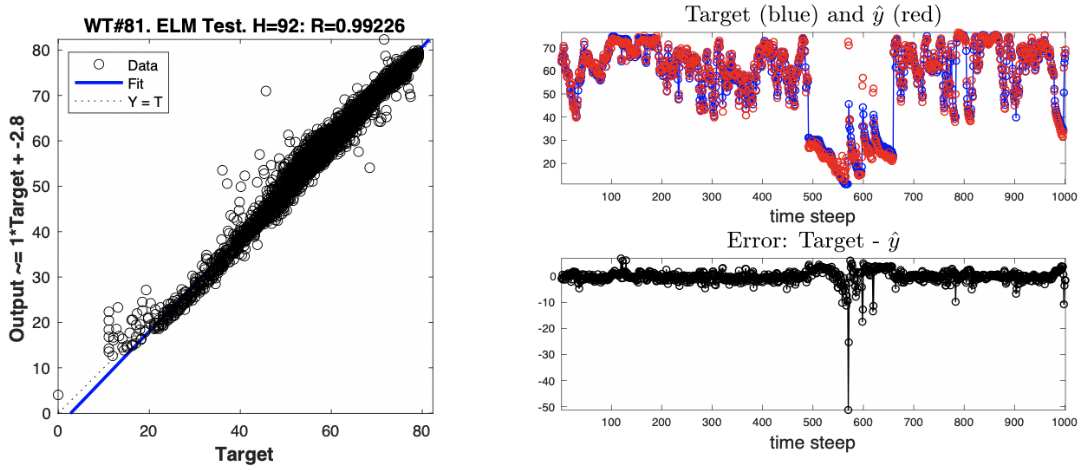


Figure 9. Test of the selected model ($H = 92$) for the WT#81. (**Left**) the regression plot using all the test data. (**Right**) from a temporal window where the target $\hat{y}$ (**top**) and the error (**bottom**) can be observed in more detail.

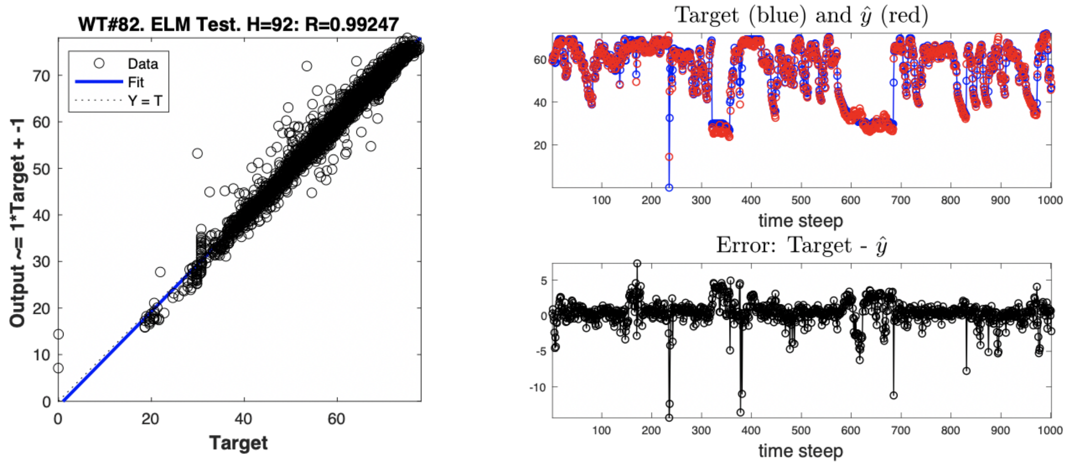


Figure 10. Test of the selected model ($H = 92$) for the WT#82. (**Left**) the regression plot using all the test data. (**Right**) from a temporal window where the target $\hat{y}$ (**top**) and the error (**bottom**) can be observed in more detail.

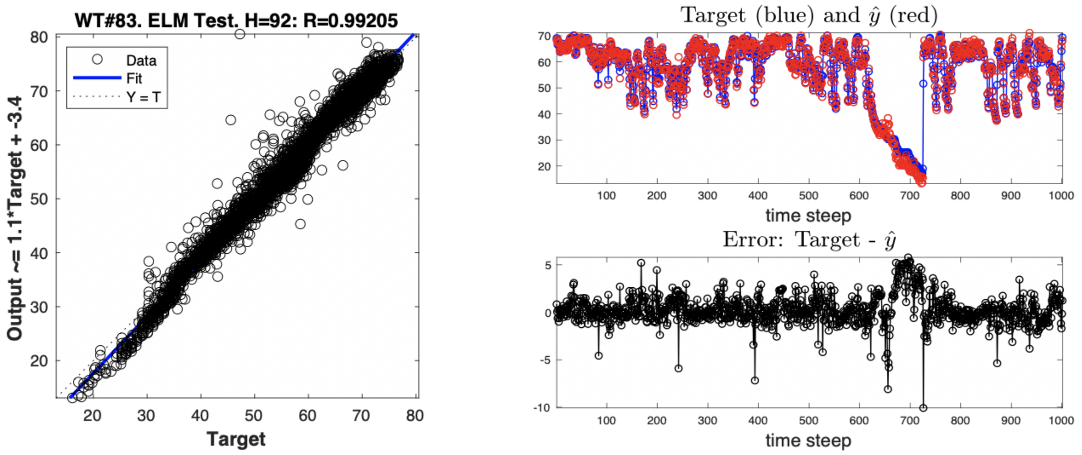


Figure 11. Test of the selected model ($H = 92$) for the WT#83. On the left, the regression plot using all the test data. On the right, from a temporal window where the target $\hat{y}$ (**top**) and the error (**bottom**) can be observed in more detail.

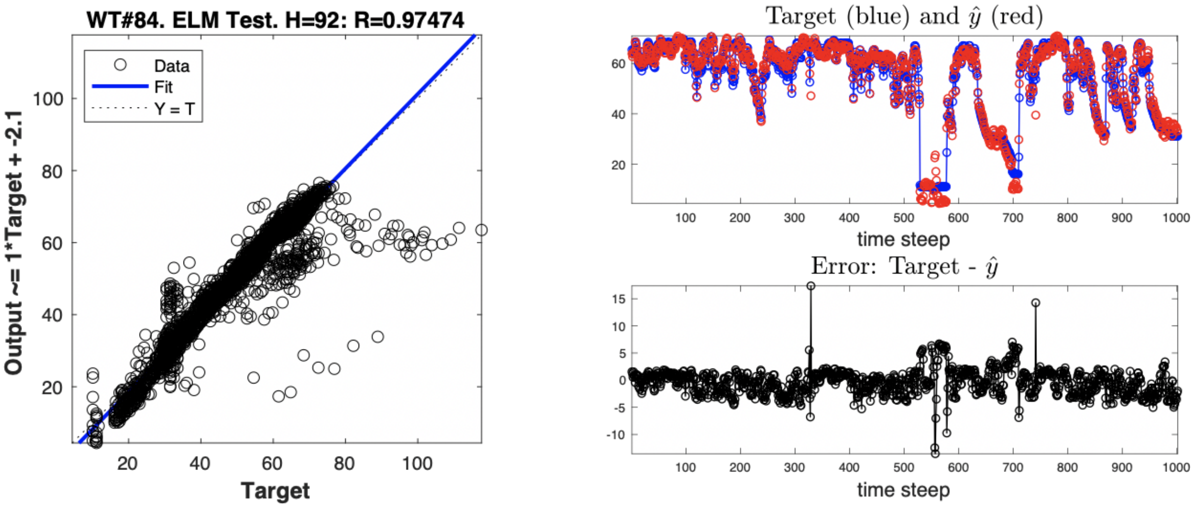


Figure 12. Test of the selected model ($H = 92$) for the WT#84. (**Left**) the regression plot using all the test data. (**Right**) from a temporal window where the target $\hat{y}$ (**top**) and the error (**bottom**) can be observed in more detail.

3.3. Prognosis Experiment

Wind Turbine Failure Prediction is a very complicated task due to the previously mentioned problems of wrong labelling and unbalancing data. The results of the classification models that have been applied before in this wind farm using SCADA data are very good in the training step. However, when the models ares tested on new data, they generate a huge proportion of false positives. To give an idea, it is quite hard to reach a Kappa value higher than 0.1, and it is common to obtain values far bellow. Moreover, classification models neither explain the meaning of the classification. All this makes classification very tricky. This is why now we present a strategy to perform wind turbine failure prediction based on normality models and regression.

Since we have access to the alarm records we can study the behaviour of the model in relation to the different turbines in the preceding moments to each of the alarms. For this experiment, the alarms recorded from the gearbox subsystem were selected. Out of the possible alarms associated with this subsystem, those that are present in the history of each of the turbines were checked. In Figure 13a, the time distribution of the five types of alarms is depicted, corresponding to the WT#84 turbine. These five types of alarms correspond to the indicators id_1271, id_1369, id_1544, id_2302 and id_2306, described in Table 4. The experiment consists of grouping all these alarms, similarly to what is represented in Figure 13b. Once the alarms have been grouped, without distinguishing between them, the pair $(target, \hat{y})$ is represented by the points corresponding to the time interval 12h before each of these alarms occurred. As expected, when the turbine works according to the model, the points $(target, \hat{y})$ fall very close to the 45-degree line. The goal is to detect, with only this small set of data, if the model has the ability to anticipate the failures and capture points outside of the line.

Table 4. Description of the error codes.

Alarm Code	Description
id_1271	MGB FuseTripCoolWatPp, Trip of the water pump's fuse in charge of cooling the gearbox oil.
id_1369	MGB Temp OilSump < SL, The temperature of the oil sump is below the expected result for a operation regime.
id_1544	PT100 defective, temperature sensor PT100 defective.
id_2302	MGB TempBear150 > SH, The temperatue of output high speed bearing super high.
id_2306	MGB Temp ErrTimeLimit, The timeout at reading the gearbox temperature sensors.

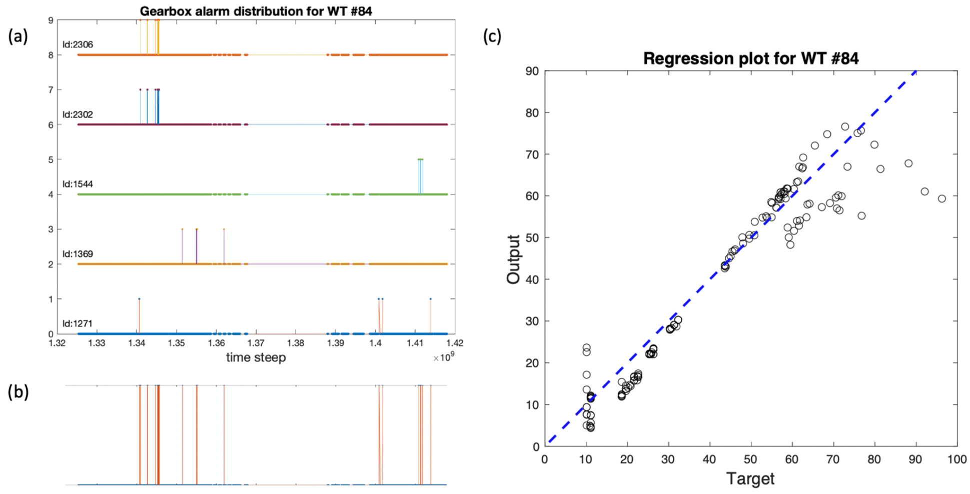


Figure 13. (**a**) Temporal distribution of the appearance of alarms in the gearbox subsystem to the WT#84. There are 5 different alarms (indicated by the identifiers of the Führlander). (**b**) Aggregation of the set of alarms represented in (**a**) in a single time frame. (**c**) Representation of the $(target, \hat{y})$ pair generated by the model using the data corresponding to the intervals that go from 36 h to 12 h before each alarm is generated.

The WT#84 has many points out of the 45-degree line. Different behaviours are captured with respect to the model, as can be seen for the selected time frame in Figure 13c. The mismatch is observed for both regimes of operation, corresponding to the high part of the line (target above 60) and for the low part of the line (target below 30).

With respect to the other turbines, for WT#80 (Figure 14a) and WT#81 (Figure 14b) the test also captures distant (discordant) points, especially in low operating regimes (target below 30). On the other hand, for WT#82 (Figure 14c) and WT#83, the malfunction, if any, is more difficult to detect for this portion of data considered.

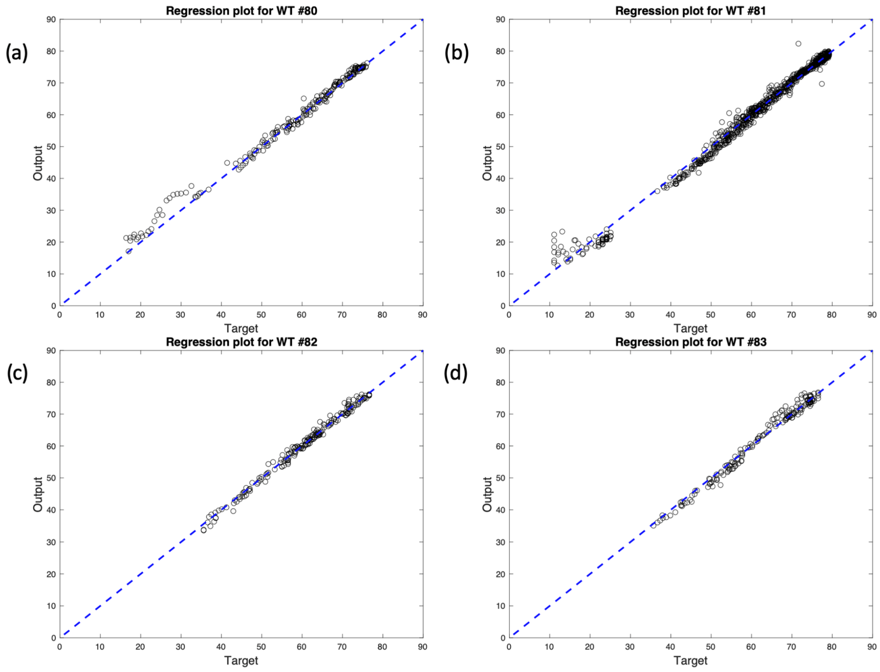


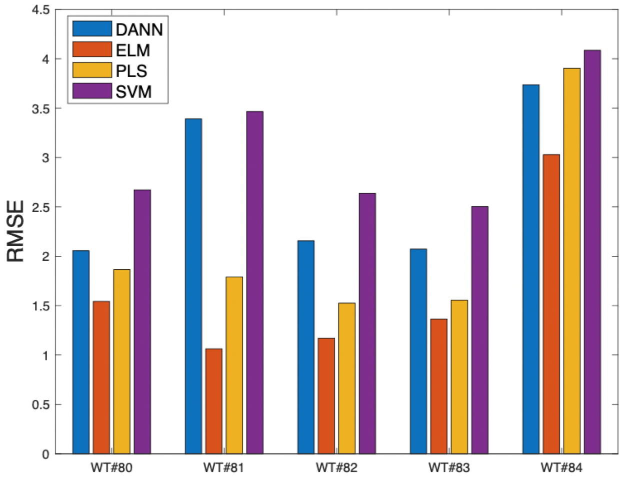
Figure 14. Representation of the pair $(target, \hat{y})$ generated by the model using the data corresponding to the intervals that go from 36 h to 12 h before each alarm is produced for the WT#80 (**a**), WT#81 (**b**), WT#82 (**c**) and WT#83 (**d**).

3.4. Comparative Results

The gearbox model previously developed based on ELM is now compared with other normality models that predict exactly the same target from the same input variables but using PLS, SVM and DANN instead. All of these alternative models have already been previously explored in the field of wind turbines [17,18,20,21]. The comparison between models is made on the basis of their ability to correctly predict the target variable for each of the different turbines, as can be seen in Figure 15. In all these cases a model of the park is obtained and the comparison is performed by calculating the average quadratic error between the prediction and the real value. The models have been trained under the same conditions, using the data of the whole park. The first half of the data is used for the training and the second half is used for the test. The resulting park model is tested on each of the turbines individually.

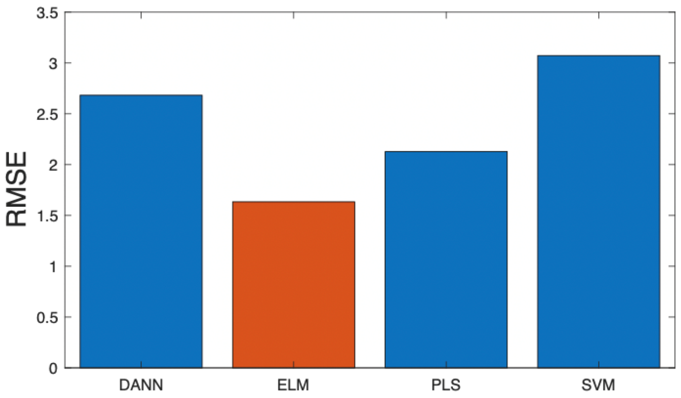
The optimisation of the ELM scheme follows the methodology previously described, exploring the size of the network for optimising the performance. In this case, 10 training sessions are carried out for each measurement, keeping the one with the best results. In the PLS-based scheme, the number of components N_{comp} is determined by the optimal CV MSE knee curvature method [18]. The parameter to be optimised is the minimum of the logarithm of cross-validation loss, which is $log(1 + cv_{loss})$. Finally, for the optimisation of the DANN, after several empirical tests to determine its architecture, we ended up choosing an architecture that presents an input layer of 6 neurons, a hidden layer of 20 neurons with ReLu the as activation function and a dropout percentage of 20%, a second hidden layer of 10 neurons with ReLu as the activation function and a dropout percentage of 10%, a third hidden layer of 5 neurons with ReLu as the activation function and an output layer of 1 neuron. For the training of this architecture the Adam optimizer agorithm was used with this configuration: *max epochs* = 50 and *minibatch size* = 128.

The RMSE obtained for each of the models on each individual turbine of the wind park is shown in Figure 15, together with the mean of all the turbines for each type of model. In all cases (individual turbines RMSE or mean RMSE) the ELM-based model obtains the best accurate prediction, suggesting that the ELM technique seems optimal when used in regression models. PLS works quite well, close to ELM in performance but always with a major error, while SVM and DANN are the worst out of all the models for all turbines. Note that each turbine behaves differently, with WT#84 being the most difficult to predict (higher RMSE). In this case, even the PLS behaves badly, with results similar to those of the SVM and DANN, while the ELM is able to maintain a low RMSE.



(**a**)

Figure 15. *Cont.*



(**b**)

Figure 15. 3-dimensional (**a**) and 2-dimensional (**b**) space of the pressure measurements.

4. Conclusions

Predictive maintenance of wind plants exclusively through SCADA data is a complicated task because wind turbines are made up of complex systems/subsystems that work in a variety of operating regimes. That is why it is a challenging task which is still in progress, but that can represent a very big competitive advantage over other O&M options that involve the installation of extra equipment, since data collection is simple and cost-effective by taking advantage of the sensors already present in the WT. Moreover, the processing of these data can be done in the cloud, activating economies of scale in such a way that the monitoring of many plants can end up in specialized centres that offer their services at a competitive cost and using state-of-the-art technology.

In the wind parks the SCADA system collects, every 5 min, a vector of points corresponding to the values of the sensors used to predict the value of the target variable. When the system is working properly, the scatter plot representation (target, predicted value) will fall very closely on the 45-degree line. Because a new point is available every 5 min, 12 new points are accumulated in 1 h. With the set of accumulated points (obtained in a few hours), the regression line can be drawn. If it works properly, this regression line will fit very well to the 45-degree line. In the event of deterioration, the regression line will change its slope slightly. The decision, therefore, is never based on a single point. The more points used to draw the regression line, the more reliable the result. The park manager should establish the number of hours and the degree of change considered in order to decide how and when the turbine needs to be checked.

The ELM model is chosen because beside its simplicity, it solves the overdetermined system through the inverse of Moore-Penrose, providing the solution that approximates the points with the Euclidean minimum norm. An additional advantage is that with very few changes we can build models to estimate multiple variables or a single one and we can obtain estimates of multiple variables directly for the whole wind farm. In this work we have shown that ELM provides the solution that approximates the points used in the test by getting the minimum quadratic error. This fact is an advantage when training the models because it avoids the overfitting observed in more sophisticated methods, such as SVM or DANN. It is also robust against possible wrongly-labelled data that can be introduced in the training phase due to the impossibility of correctly filtering the whole database. In the experiments, we have shown that normality models built using ELM are adequate to detect possible malfunctions of the wind turbines up to 12 days in advance. With the set of points generated by a turbine during a day, regression lines can be drawn that are adjusted to 45 degrees very accurately when the system under testing is working correctly. Therefore, under these conditions, deviations in the slope of the line can generate a warning or an alarm. The threshold for deciding when it is a warning or an alarm will depend on the park and the strategy of the managers, who may be more or less conservative in their decision to send a technician to supervise the turbines. In addition, other factors must be taken into account, such as the variance of the error and, very importantly, the range of value of the variables and the operational state of the turbine, as it could be inactive. All this is important for a correct interpretation of the results. Despite its simplicity, the ELM achieves the best performance amongst the compared models.

Having an extremely efficient, low-cost computational model training approach such as ELM allows for the performance of many tests, and also lets us apply a parallel grid-search strategy on a wide set of parameters. Therefore, the model can be trained with more or less variables and targets, and it can be optimised with the test data in a very efficient way. With ELM, the models obtained are sensitive enough to detect the deterioration of the turbine behaviour even before the activation of alarms. These characteristics make ELM systems a candidate to be considered when deriving real time wind turbine models.

Author Contributions: Conceptualization: P.M.-P., A.B.-M. and J.S.-C.; methodology: P.M.-P. and J.S.-C.; software: P.M.-P. and A.B.-M.; validation: P.M.-P. and J.S.-C.; formal analysis: P.M.-P. and J.S.-C.; investigation: P.M.-P., A.B.-M. and J.S.-C.; resources: J.S.-C. and M.S.-S.; data curation: A.-B.M.; writing—original draft preparation: J.S.-C. and P.M.-P.; writing—review and editing: J.S.-C. and M.S.-S.; visualization: M.S.-S.; supervision: P.M.-P. and J.S.-C.; project administration: P.M.-P. and J.S.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in the experiments is owned by Smartive (http://smartive.eu/).

Acknowledgments: The authors would like to thank the Smartive company (http://smartive.eu/) for providing the data used in the experimental part. The authors would also like to thank the anonymous reviewers for their suggestions and comments, which were very helpful in improving the final version of this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	artificial neural networks
CV	cross-validation
DANN	deep artificial neural networks
ELM	extreme learning machines
LCOE	levelized cost of energy
MSE	mean squared error
NBM	normal behaviour model
O&M	operation and maintenance
RMSE	root-mean-square error
PLS	partial least squares
ReLU	rectified linear unit
RMSE	root mean square error
SCADA	supervisory control and data acquisition data
SLFN	single-hidden layer feedforward neural networks
SVM	support vector machines
WGEN	wind generator
WT	wind turbine

References

1. Lei, X.; Sandborn, P.; Bakhshi, R.; Kashani-Pour, A.; Goudarzi, N. PHM based predictive maintenance optimization for offshore wind farms. In Proceedings of the 2015 IEEE Conference on Prognostics and Health Management (PHM), Austin, TX, USA, 22–25 June 2015; pp. 1–8.
2. Wang, H. A survey of maintenance policies of deteriorating systems. *Eur. J. Oper. Res.* **2002**, *139*, 469–489. [CrossRef]
3. Pérez, J.M.P.; Márquez, F.P.G.; Tobias, A.; Papaelias, M. Wind turbine reliability analysis. *Renew. Sustain. Energy Rev.* **2013**, *23*, 463–472. [CrossRef]

4. Zhang, P.; Lu, D. A survey of condition monitoring and fault diagnosis toward integrated O&M for wind turbines. *Energies* **2019**, *12*, 2801.
5. Badrzadeh, B.; Bradt, M.; Castillo, N.; Janakiraman, R.; Kennedy, R.; Klein, S.; Smith, T.; Vargas, L. Wind power plant SCADA and controls. In Proceedings of the 2011 IEEE Power and Energy Society General Meeting, Detroit, MI, USA, 24–28 July 2011; pp. 1–7.
6. Wilkinson, M.; Darnell, B.; van Delft, T.; Harman, K. Comparison of methods for wind turbine condition monitoring with SCADA data. *IET Renew. Power Gener.* **2014**, *8*, 390–397. [CrossRef]
7. Ragheb, A.; Ragheb, M. Wind turbine gearbox technologies. In Proceedings of the IEEE 2010 1st International Nuclear & Renewable Energy Conference (INREC), Amman, Jordan, 21–24 March 2010; pp. 1–8.
8. Schlechtingen, M.; Santos, I.F. Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. *Mech. Syst. Signal Process.* **2011**, *25*, 1849–1875. [CrossRef]
9. Zaher, A.; McArthur, S.; Infield, D.; Patel, Y. Online wind turbine fault detection through automated SCADA data analysis. *Wind. Energy Int. J. Prog. Appl. Wind. Power Convers. Technol.* **2009**, *12*, 574–593. [CrossRef]
10. Bangalore, P.; Tjernberg, L.B. An artificial neural network approach for early fault detection of gearbox bearings. *IEEE Trans. Smart Grid* **2015**, *6*, 980–987. [CrossRef]
11. Yuan, T.; Sun, Z.; Ma, S. Gearbox fault prediction of wind turbines based on a stacking model and change-point detection. *Energies* **2019**, *12*, 4224. [CrossRef]
12. López de Calle, K.; Ferreiro, S.; Roldán-Paraponiaris, C.; Ulazia, A. A Context-Aware Oil Debris-Based Health Indicator for Wind Turbine Gearbox Condition Monitoring. *Energies* **2019**, *12*, 3373. [CrossRef]
13. Marti-Puig, P.; Blanco-M, A.; Cárdenas, J.J.; Cusidó, J.; Solé-Casals, J. Effects of the pre-processing algorithms in fault diagnosis of wind turbines. *Environ. Model. Softw.* **2018**, *110*, 119–128. [CrossRef]
14. Marti-Puig, P.; Blanco-M, A.; Cárdenas, J.J.; Cusidó, J.; Solé-Casals, J. Feature selection algorithms for wind turbine failure prediction. *Energies* **2019**, *12*, 453. [CrossRef]
15. Huang, G.B.; Chen, L.; Siew, C.K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* **2006**, *17*, 879–892. [CrossRef] [PubMed]
16. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
17. Kusiak, A.; Li, W. The prediction and diagnosis of wind turbine faults. *Renew. Energy* **2011**, *36*, 16–23. [CrossRef]
18. Geladi, P.; Kowalski, B.R. Partial least-squares regression: A tutorial. *Anal. Chim. Acta* **1986**, *185*, 1–17. [CrossRef]
19. Li, K.; Nie, H.; Gao, H.; Yao, X. Knee Point Identification Based on Trade-Off Utility. *arXiv* **2020**, arXiv:2005.11600.
20. Wang, L.; Zhang, Z.; Long, H.; Xu, J.; Liu, R. Wind turbine gearbox failure identification with deep neural networks. *IEEE Trans. Ind. Inform.* **2016**, *13*, 1360–1368. [CrossRef]
21. Teng, W.; Cheng, H.; Ding, X.; Liu, Y.; Ma, Z.; Mu, H. DNN-based approach for fault detection in a direct drive wind turbine. *IET Renew. Power Gener.* **2018**, *12*, 1164–1171. [CrossRef]
22. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
23. Blanco-M, A.; Sole-Casals, J.; Marti-Puig, P.; Justicia, J.J.C.I.; Cusido, J. Impact of target variable distribution type over the regression analysis in wind turbine data. In Proceedings of the 2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI), Funchal, Portugal, 10–13 July 2017; pp. 1–7.
24. Marti-Puig, P.; Gibert, K.; Cusidó, J.; Solé-Casals, J. A text-mining approach to assess the failure condition of wind turbines using maintenance service history. *Energies* **2019**, *12*, 1982.
25. Carroll, J.; McDonald, A.; McMillan, D. Failure rate, repair time and unscheduled O&M cost analysis of offshore wind turbines. *Wind Energy* **2016**, *19*, 1107–1119.
26. Battiti, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions Neural Netw.* **1994**, *5*, 537–550. [CrossRef] [PubMed]
27. Cheng, H.; Qin, Z.; Feng, C.; Wang, Y.; Li, F. Conditional mutual information-based feature selection analyzing for synergy and redundancy. *ETRI J.* **2011**, *33*, 210–218. [CrossRef]
28. Yang, H.H.; Moody, J.E. *Data Visualization and Feature Selection: New Algorithms for Nongaussian Data*; Oregon Graduate Institute of Science and Technology: Beaverton, OR, USA, 1999; Volume 99, pp. 687–693.
29. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [CrossRef] [PubMed]
30. Meyer, P.E.; Bontempi, G. On the use of variable complementarity for feature selection in cancer classification. In *Applications of Evolutionary Computing*; Springer: Berlin, Germany, 2006; pp. 91–102.
31. Fleuret, F. Fast binary feature selection with conditional mutual information. *J. Mach. Learn. Res.* **2004**, *5*, 1531–1555.
32. Jakulin, A. Machine Learning Based on Attribute Interactions. Ph.D. Thesis, Univerza v Ljubljani, Ljubljana, Slovenia, 2005.