

## Article

# Exploring the Effect of Temporal Aggregation on SCADA Data for Wind Turbine Prognosis Using a Normality Model

Pere Marti-Puig <sup>1,\*</sup> , Alejandro Bennásar-Sevillá <sup>2</sup>, Alejandro Blanco-M. <sup>1</sup>  and Jordi Solé-Casals <sup>1,\*</sup> 

<sup>1</sup> Data and Signal Processing Group, University of Vic—Central University of Catalonia, 08500 Vic, Catalonia, Spain; alejandro.blanco@uvic.cat

<sup>2</sup> Faculty of Computer Science, Multimedia and Telecommunications, Open University of Catalonia, 08080 Barcelona, Catalonia, Spain; abennasars@uoc.edu

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

**Abstract:** Today, the use of SCADA data for predictive maintenance and forecasting of wind turbines in wind farms is gaining popularity due to the low cost of this solution compared to others that require the installation of additional equipment. SCADA data provides four statistical measures (mean, standard deviation, maximum value, and minimum value) of hundreds of wind turbine magnitudes, usually in a 5-min or 10-min interval. Several studies have analysed the loss of information associated with the reduction of information when using five minutes instead of four seconds as a sampling frequency, or when compressing a time series recorded at 5 min to 10 min, concluding that some, but not all, of these magnitudes are seriously affected. However, to our knowledge, there are no studies on increasing the time interval beyond 10 min to take these four statistical values, and how this aggregation affects prognosis models. Our work shows that, despite the irreversible loss of information that occurs in the first 5 min, increasing the time considered to take the four representative statistical values improves the performance of the predicted targets in normality models.

**Keywords:** wind turbine; fault diagnosis; renewable energy; extreme learning machines (ELM); feature engineering; normal behaviour models



**Citation:** Marti-Puig, P.; Bennásar-Sevillá, A.; Blanco-M., A.; Solé-Casals, J. Exploring the Effect of Temporal Aggregation on SCADA Data for Wind Turbine Prognosis Using a Normality Model. *Appl. Sci.* **2021**, *11*, 6405. <https://doi.org/10.3390/app11146405>

Academic Editor: Talal Yusaf

Received: 17 June 2021

Accepted: 8 July 2021

Published: 11 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Energy is one of the most important aspects of the evolution of our society. Easy and cheap access to energy has always been a challenge, and nowadays there is the added factor of pollution that can be caused by the type of energy source employed. Increasingly, carbon and gas-based energies are being restricted in favour of green energies, gathered from the sun, tides, or wind, to name but a few. These renewable energy sources are an excellent choice because they are clean and can exist, in different forms, over a wide geographical area. This is another important point in democratising access to energy, as for many countries it is not possible to access carbon-based energy.

One of the greatest financial effects on the Levelised Cost of Energy (LCOE) of wind turbines (WT) is linked to the operations and maintenance (O&M) tasks. Various systems have been proposed to decrease the O&M costs. In [1–3], the reader can find an overview of several strategies that could be used. One of the options is to detect, as soon as possible, abnormal behaviour in WTs, before failures occur. Then, because the malfunction can be solved faster and without important damages on the WT subsystems, the LCOE is reduced. There are various ways to deal with that, and in this work a data-driven Normal Behaviour Model (NBM) will be used, based on our previous work [4], exploring how the temporal scale affects the results of the prediction system.

Typically, WTs have a SCADA system that provides a set of data to monitor the status of the subsystems [5]. This data comes from many sensors placed in different parts of the WT, and comprises the minimum value, maximum value, mean value, and

standard deviation of various physical measurements, such as temperature, pressure, voltage, current, etc., in various parts of the subsystems.

Generally, the methodologies for condition-monitoring using SCADA data are based on signal trending strategies, artificial neural networks, and physical models. Machine learning algorithms are habitually used to perform some type of exploration and/or analysis on the data. This can be done, for example, for predicting failures in wind turbines using SCADA data [6]. The problem linked with using classification models to predict failure states are related to the extremely unbalanced scenario, in which almost all the examples are from one class (healthy state) and very few examples from the other class (alarm state) [7], in addition to the errors present in the labels of the data, which make it even more complicated to use it, unless a huge human expert-based pre-processing is carried out beforehand. This is why the use of normality models was proposed in [4] for predicting one variable generated by a subsystem, using the rest of the variables, and comparing it with the real value in order to detect if the prediction is out of the norm, in which case that will indicate that the subsystem deteriorates. In this first work, the model was implemented using extreme learning machines (ELM), comparing the experimental results with other well-known machine learning approaches such as partial least squares (PLS), support vector machines (SVM), or deep artificial neural networks (DANN). One of the reasons to use ELMs is the speed of training, which is done quickly by calculating an inverse Moore–Penrose matrix and simple matrix algebra manipulations. Having a fast training system enables parameters to be tuned very quickly, so that models can be tested and adapted in a very short time. Details on the ELM algorithm and its implementation can be found in [4].

The information gathered by the SCADA system is habitually provided at a frequency of 10-min intervals, but in some cases the system reports values every 5 min, obtaining 6 or 12 samples every hour, respectively. Recently, some works have investigated the effect of the sampling frequency in SCADA systems. For example, in [8] a technique for wind turbine performance monitoring based on high-frequency SCADA data is presented. The data is provided every 4 s and its use is demonstrated to be beneficial for performance monitoring purposes. As stated by the authors, the dynamic behaviour of the WT is understood, learned, and reproduced by multivariate non-parametric regression models, using data at a sampling frequency of 4 s. The use of 5-min data typically smooths out these dynamics and does not allow for the precision that high-frequency data provides. The recommendations provided by [9] also highlights that 5-min data is not suitable as a standalone solution, but is a good and important complement for designing condition monitoring systems (CMS). Another interesting work [10] compares SCADA data for low and high frequency data, providing evidence for monitoring at higher resolutions in practical scenarios. Therefore, high-frequency data seems to be a good choice to develop CMS and provide more intelligence for wind farm management.

To our best knowledge, however, there are no works on investigating the effect of the aggregation of data to higher temporal periods, such as 20-min or 60-min data obtained from the 5-min ones. The interest of such aggregation is basically to simplify models and systems in a way that less information needs to be stored, transmitted, and processed by them. In this paper we explore this effect in depth and demonstrate experimentally that aggregation to longer time periods from 5-min data does not degrade the results but instead tends to generate better normality models when predicting a target variable using the rest of the variables of the system/subsystem and computing the error deviation from the real target variable.

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. Section 3 contains the experimental results, while in Section 4 they are discussed and compared to highlight their interest. Finally, Section 5 is dedicated to conclusions and outlining future work.

## 2. Materials and Methods

In the following section a brief explanation of the effects of the sampling frequency and time-window considered for the data will be presented, together with details on the normality models based on ELM and of the database used to carry out the experiments.

### 2.1. The SCADA Data and the Loss of Information Due to Averaging

Replacing the instantaneous values of a variable by a value averaged over a period introduces an error. Therefore, in general, the error due to averaged values is expected to increase as the period of averaging of the data increases. This error is also likely to be highly dependent on the variability of the data. The main factor in this uncertainty is the variability of wind speed and direction, and the frequencies contained in the signals. Therefore the averaging can be crucial if the signal has frequencies higher than the sampling/averaging, or of little relevance if it hardly varies at all in an averaging period.

The autocorrelation of the signals involved will bring meaningful information. For example, if a signal has very low autocorrelation for certain time lags, this implies a very little statistical similarity between samples separated by these time lags. Thus, averaging over a similar period of time results in the signal significantly losing information. On the other hand, a signal with a high autocorrelation for a given time lag will suffer a minimal loss of information when averaged over periods close to such time lag. This dependence of the quality of SCADA data on its autocorrelation and the averaging period has already been studied previously in [11] and specially in [8], where a study with high-frequency SCADA data of a wind farm is presented, reporting that the autocorrelation of the variables shows a clear pattern consisting of a drop that fits an exponential that depends on the time of averaging. The exponential slope differs for each variable; for instance, the most pronounced is observed in the generated active power. In general, significant information losses due to 5-min aggregations, more commonly presented in SCADA systems, are reported especially in the following four variables: The wind speed, the orientation, the main rotor speed, and, as previously mentioned, the generated active power. In these cases, the normalized autocorrelation is practically zero in time lags of five minutes onwards.

### 2.2. Normality Models Based on ELM

In wind turbine prognosis, cases representing a malfunction of a particular system or subsystem are extremely rare. Furthermore, the inconsistency of labelling when dealing with extreme imbalance in class representation suggests using a normality model associated with a regression technique rather than classification-based models. The critical point is to train a normality model using collections of data recorded in different operating regimes when the turbines are operating correctly. For the model to be sufficiently accurate, the records must be rich enough to represent different wind and weather conditions. Accordingly, we propose an SLFN (see Figure 1) trained under the ELM paradigm because it performs very well and the training process is fast and effective [4].

The procedure is simple. From a set of  $L$  variables of a given subsystem at the instant  $i$ , organised in the vector  $\mathbf{x}_i = [x_0 \cdots x_{L-1}]^T$ , we make the prediction of another target variable of this same subsystem and at the same instant of time,  $t_i$ . It is important to note that we will also have the real measure of  $t_i$ , but in our estimation of it, named  $\hat{y}_i$ , we will only use  $\mathbf{x}_i$ .

The underlying idea is that when the system is working properly, the estimate provided by the normality model and the measurement of the target signal will be nearly the same, and its scatter plot will fit well on a 45-degree line. However, when systems deteriorate, measurements and estimates begin to diverge. If the system malfunction persists, the slope of the regression line built by integrating this new data begins to change.

Given that the input variables can have a highly different dynamic range, we apply a Z-score normalisation (see Figure 1). The normalised variables are the inputs of the SLFN. If we follow the signal flow of Figure 1 we will see that, for a network with  $H$  hidden nodes, the variables are multiplied by the input weights of the network, organised as the matrix

$\mathbf{W}$  of dimensions  $L \times H$ , and the biases are added, as the vector  $\mathbf{b}$  of  $H$  elements, to finally obtain the signal at each node. Then, to calculate the estimation  $\hat{y}_i$  we multiply the signal of each node by the output weights, arranged in the vector  $\mathbf{o}$ , also of  $H$  elements. Finally, the activation function (sign function in this case) is applied to this result. Mathematically, for the time instant  $i$  we will have:

$$\hat{y}_i = \text{sig}\left(\mathbf{x}_i^T \mathbf{W} + \mathbf{b}^T\right) \mathbf{o}. \tag{1}$$

Half of the data is used to train the model. The training data, after normalisation, is organised into the matrix  $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_i \cdots \mathbf{x}_N]$  of size  $L \times N$ , together with the corresponding target for each sample, the vector  $\mathbf{t} = [t_1 \cdots t_i \cdots t_N]^T$ . The goal is to find the weights  $\mathbf{W}$ ,  $\mathbf{b}$  i  $\mathbf{o}$  that better fit the following equation:

$$\text{sig}(\mathbf{X}^T \mathbf{W} + \mathbf{u} \mathbf{b}^T) \mathbf{o} = \mathbf{t}, \tag{2}$$

$\mathbf{u}$  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:

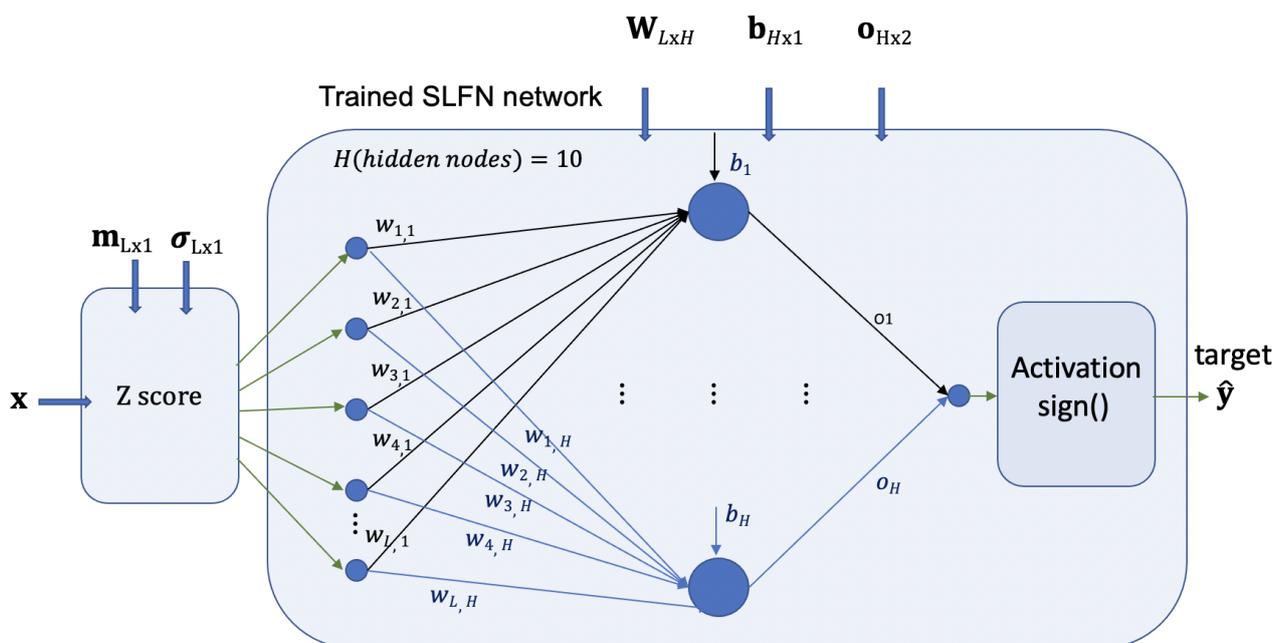
$$\mathbf{H} = \text{sig}(\mathbf{X}^T \mathbf{W} + \mathbf{u} \mathbf{b}^T). \tag{3}$$

According to the ELM paradigm, once  $H$  is determined,  $\mathbf{W}$  and  $\mathbf{b}$  are randomly selected, so  $\mathbf{H}$  will be completely defined. A zero-mean, unit-variance Gaussian distribution function is used to generate these random values. Then, the output weight matrix  $\mathbf{o}$ , provided that  $(\mathbf{H}^T \mathbf{H})^{-1}$  is non singular, can be computed as:

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

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

Once the network SLFN is defined, i.e., when  $\mathbf{H}$ ,  $\mathbf{W}$ ,  $\mathbf{b}$ , and  $\mathbf{o}$  are determined, the estimation of the target will just be derived by applying Equation (1) to the input vector, containing the normalised input variables at the desired instant of time.



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

ELM is used in this work following the same strategy of [4]. This technique, used in its original formulation is characterised by providing the solution that approximates all the training points with the minimum mean square error. This property is very interesting when we have an over-representation of the normal state, as the ELM model captures it very well. This makes the ELM very robust even with the inclusion of a small proportion of wrongly-labelled data. Moreover, training an ELM is a fast process, allowing one to try several architectures and/or parameters in a short time. Finally, the experimental results obtained in this work will be compared with the above-mentioned work, making it easier to see the effect of the temporal-aggregation strategy. We refer the reader to [4] and references therein for more details on normality models and the ELM technique.

### 2.3. Experimental Data

An extensive 3-year SCADA database of five Fuhrländer FL2500 2.5 MW 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. Every five minutes, 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. Variables are stored with a name symbolising the (sub)system and the variable type, separated by an underscore; the first term is the main physical system, e.g., generator = wgen, transmission = wtrm, nacelle = wnac, etc. The second term is the variable type of the 5-min interval; min = minimum, avg = average, etc. The third and fourth terms are the final name of the variable. 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>) (accessed on 1 June 2021) and has been used in other publications ([4,12–14]).

The SCADA database consists of 502 variables, one of which is a timestamp, another is the turbine identifier, 188 are system alarms, and the remaining 312 are derived from physical magnitude readings made by the sensors installed in each turbine. Specifically, there are 78 physical magnitudes collected by their respective sensors, split into four variables: The mean during the data collection interval, the maximum, minimum, and standard deviation during that interval. The SCADA data of these WTs are provided by default using a 5-min timeslot. Then, aggregations of the data are performed from the 5-min values to longer time intervals of 10, 15, 30, and 60 min to perform the experiments. Only the average values were used in the experiments, although it is trivial to extend them to maximum or minimum values, for example. In the same way, the alarms are reallocated to the corresponding timeslot according to the time interval extension.

#### 2.4. Gearbox

To carry out the experiments, the Gearbox system was chosen, following the line of work developed in [4] and very similarly in [15], by making models of normality from simple neural networks used as regressors. The gearbox is the part of the turbine that has the greatest impact on maintenance, repair, and manufacturing costs. As can be seen in Table 4 of the report from NREL [16], the drivetrain, to which the gearbox belongs to, represents the second most expensive element to maintain, no matter the way the amortised cost of generation is calculated (CapEx & LCOE). Notice in the same table that the costs of failure or maintenance of the tower module (i.e., foundations, maintenance of the concrete/metal structure of the tower) are higher, but we have not focused on the detection of foundation failures, as this is a system that tends to have fewer failures and is not as important for the operators.

In the case we are dealing with, we will estimate a target variable from different predictors that, according to an expert, are relevant to characterise the gearbox system. They are listed in Table 1, while Table 2 summarises the codes in the database associated with the alarms in the Gearbox system, together with a brief description. The data recording period, for each of the WTs and for each time scale, is represented in Table 3. The temporal representation of the distribution of alarms generated by the Gearbox system for each WT can be seen in Figure 2.

**Table 1.** The variables and the target selected by an expert as being relevant for the gearbox operation with a brief description.

Variable Name	Description
wgen_avg_RtrSpd_IGR	Rotor speed
wtrm_avg_TrmTmp_GnCIWtBkw	Generator cooling system, cold side (inlet from radiator)
wtrm_avg_TrmTmp_GnCIWtFrw	Generator cooling system, hot side (outlet to radiator)
wgdc_avg_TriGri_PwrAt	Wind Turbine active power
wnac_avg_ExlTmp	Ambient temperature, outside the nacelle
wnac_avg_WSpd1	Wind speed in m/s (provided by anemometer, outside nacelle)
wtrm_avg_TrmTmp_Brg2	Temperature transmission bearing 2 (main bearing system)
Target Name	Description
wtrm_avg_TrmTmp_Brg1	Temperature transmission bearing 1 (main bearing system)

**Table 2.** Description of the error codes provided by the SCADA database that are associated to any malfunction of the transmission chain system.

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.

**Table 3.** Temporal distribution of data, depending on temporal granularity and turbine identifier.

WT id	60 min	30 min	15 min	10 min	5 min	
start	80	1 January 2012 01:00	1 January 2012 01:05			
	81	1 January 2012 01:00	1 January 2012 01:00	1 January 2012 01:00	1 January 2012 01:10	1 January 2012 01:10
	82	1 January 2012 01:00	1 January 2012 01:05			
	83	1 January 2012 01:00	1 January 2012 01:05			
end	80	8 December 2014 06:00	8 December 2014 06:00	8 December 2014 06:15	8 December 2014 06:10	8 December 2014 06:15
	81	8 December 2014 06:00	8 December 2014 06:00	8 December 2014 06:15	8 December 2014 06:10	8 December 2014 06:15
	82	8 December 2014 06:00	8 December 2014 06:00	8 December 2014 06:15	8 December 2014 06:10	8 December 2014 06:15
	83	8 December 2014 06:00	8 December 2014 06:00	8 December 2014 06:15	8 December 2014 06:20	8 December 2014 06:20

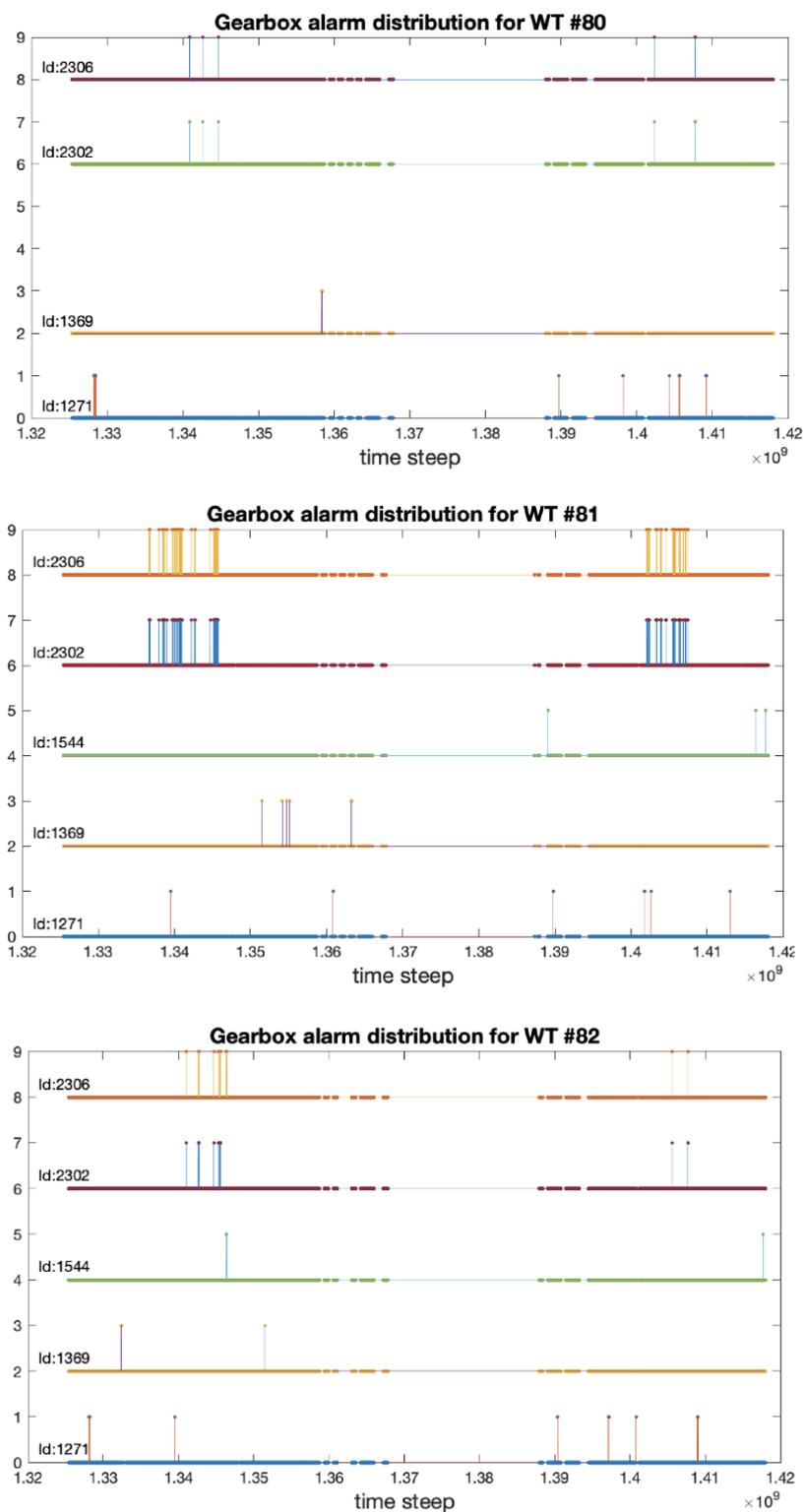
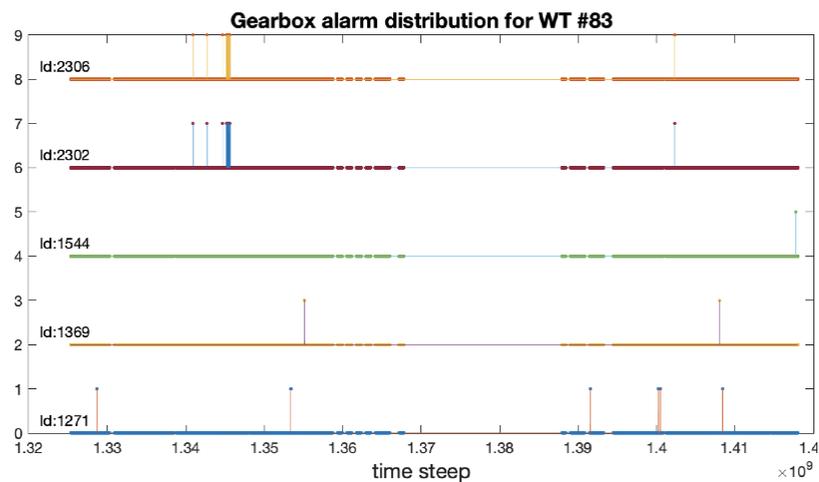


Figure 2. Cont.

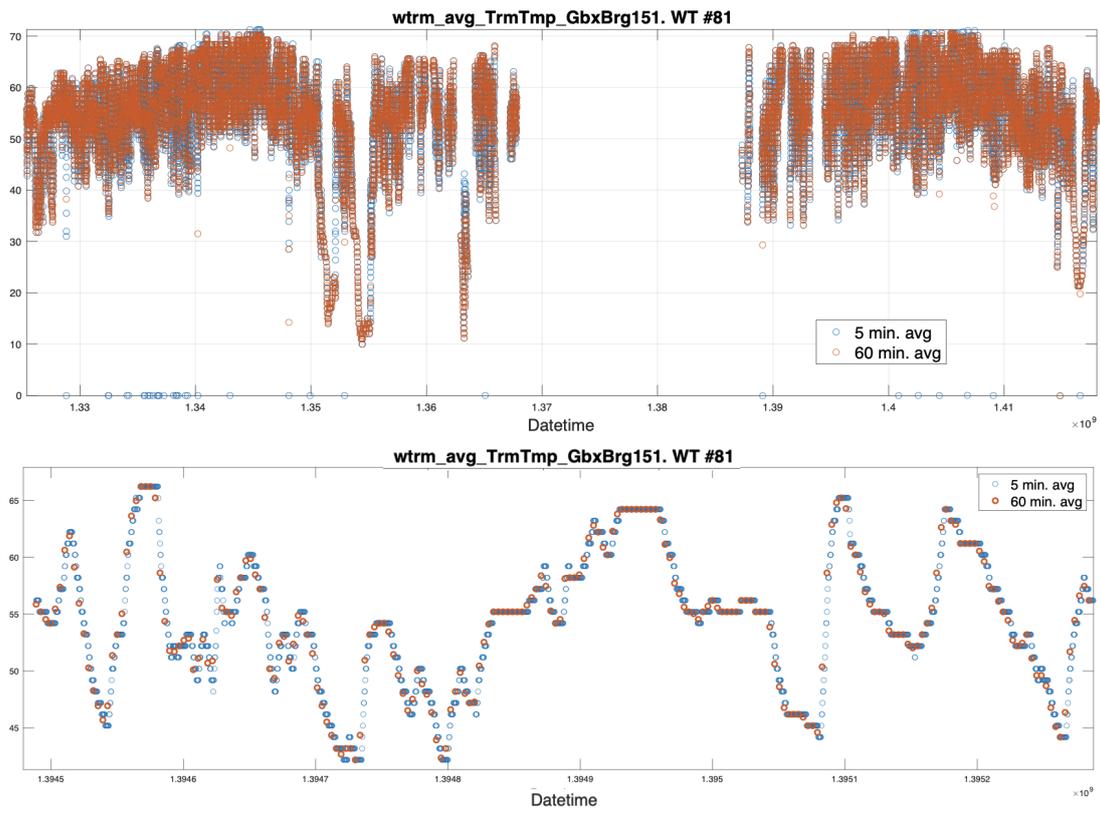


**Figure 2.** Time distribution of alarms generated by the Gearbox system for each WT (#80, #81, #82, and #83) and separated by its type. The horizontal axis, common to all the plots, is the time reference. The vertical lines indicate the temporal location of the occurrence of each type of alarm. The thick horizontal lines indicate the presence of data and the thin horizontal lines indicate the absence of data. The alarm type identifier is detailed in Table 2.

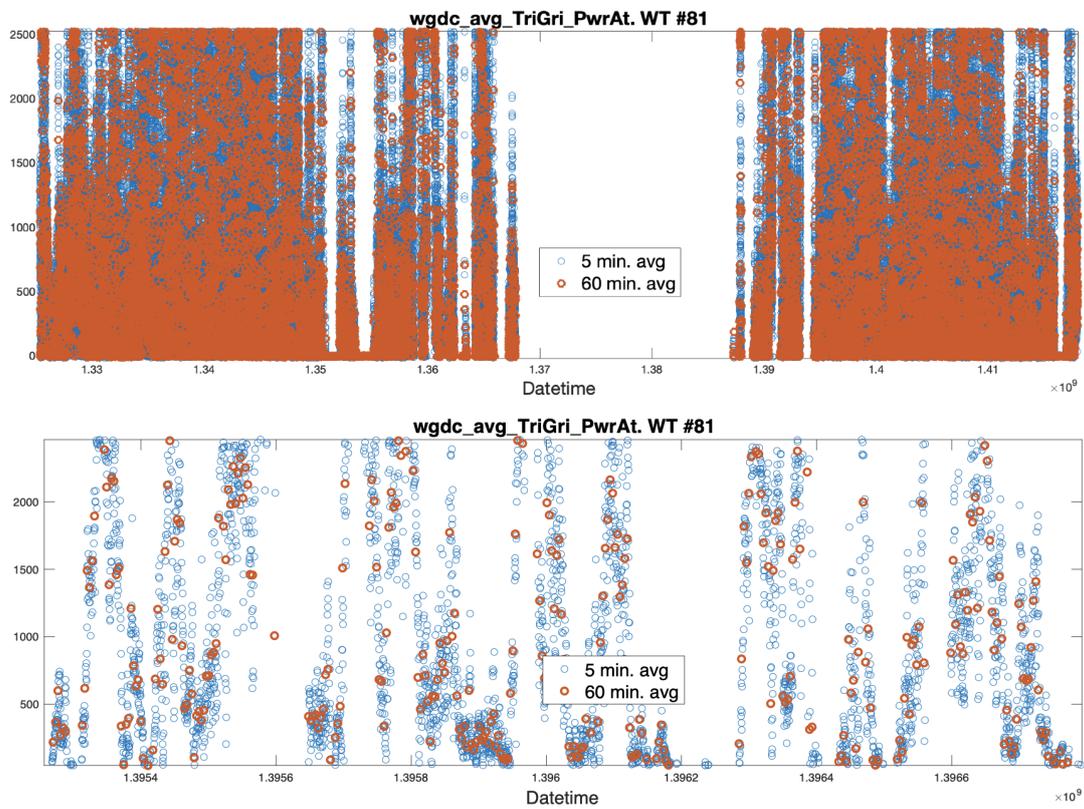
### 2.5. Type of Chosen Signals

In this section, the simultaneous representation of the data at different time scales is presented. It is not an exhaustive description of each signal but a representation for a slow variation signal (a temperature at a point of the turbine) and a fast variation signal (the power generated) in order to visually contrast the statistical data obtained from the 5-min average with those of the 60-min average. The variation of the temperature in a point of the turbine, which corresponds to the variable *wtrm\_avg\_TrmTmp\_GbxBrg151* in our model, is presented in Figure 3. The whole set of available data for WT#81, with values obtained at a 5-min time scale (blue) and 60-min time scale (brown) is depicted at the top of the Figure. The bottom part of this Figure shows a detail of this signal in which the level of resolution provided by the temperature sensors and the signal quantification process is observed. Visually, it can be seen that both representations preserve the signal detail in a similar way. In Figure 4, the active power generated by the turbine *wgdc\_avg\_TriGri\_PwrAt* is shown, which is a faster variation signal. The upper part shows the whole signal recorded in the database for the WT#81, again superimposing the 5-min resolution with the 60-min resolution. The bottom part shows a detail of this signal.

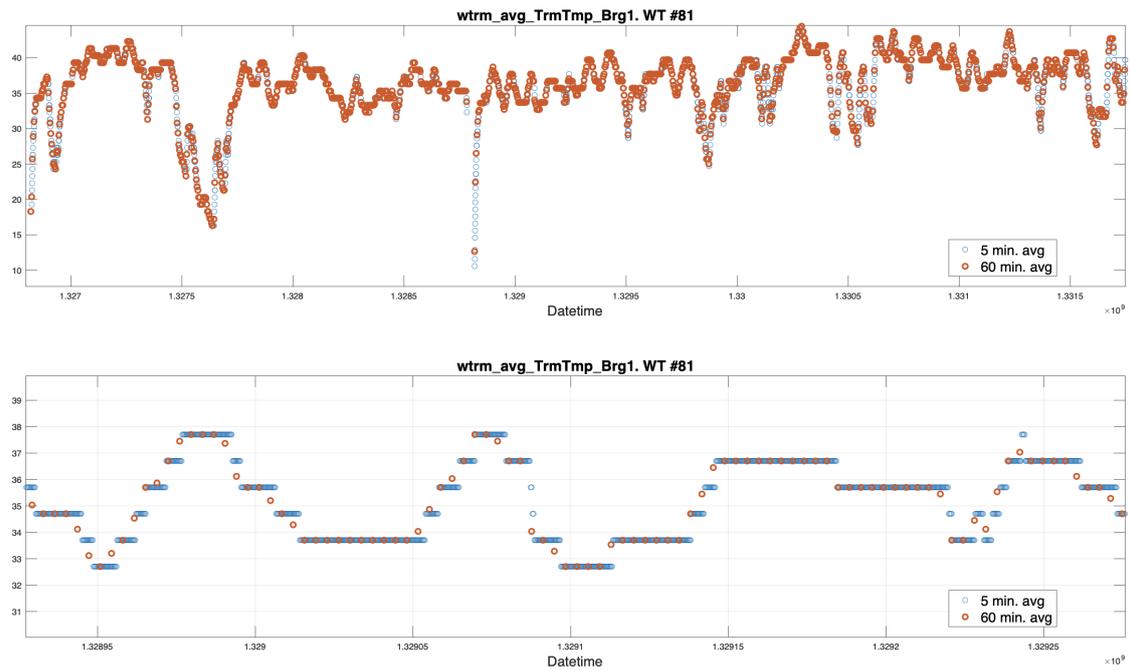
Figure 5 shows the details of the signal *wtrm\_avg\_TrmTmp\_Brg1* used as a target to build the model. It can be seen that it is again a slow-varying signal where the fact of changing the time scale of the data seems to have little effect. It is observed that the temperature in many moments remains constant. Finally, from the inspection of the signals, values equal to zero appear occasionally both in the target and in some of the variables. This phenomenon can be seen in Figure 6, where on the 5-min scale there are points that randomly turn to zero. These points correspond to an error of the sensors, of the transmission of signals, or of the SCADA system itself, since the physical process of heat transfer that is measured does not have the capacity to vary so quickly. These points, which would clearly introduce an error, will be discarded when making the models.



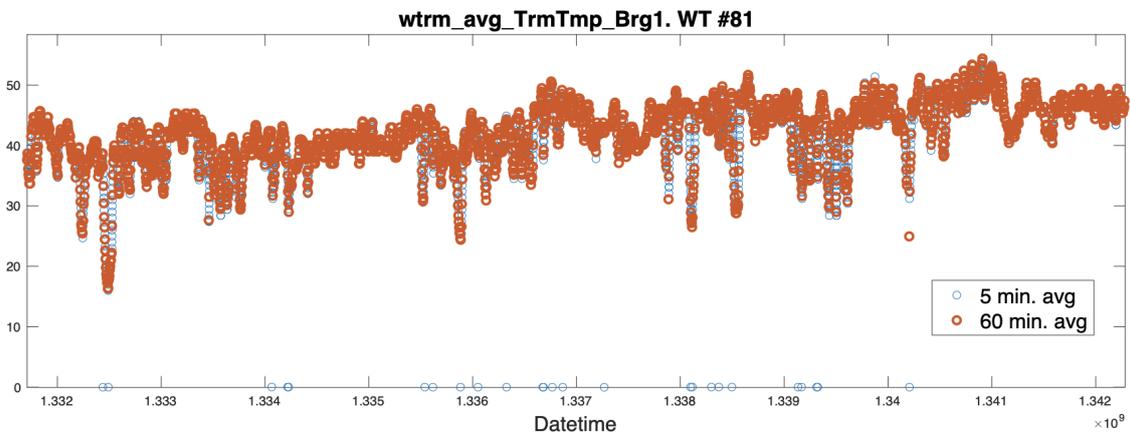
**Figure 3.** *wtrm\_avg\_TrnTmp\_GbxBrg151* signal. The top is a representation of the whole set of data available from the WT#81 for the time scales of 5 min (blue) and 60 min (brown). Below is a zoom of this signal.



**Figure 4.** *wgdg\_avg\_TriGri\_PwrAt* signal. The top is a representation of the whole set of data available from the WT#81 for the time scales of 5 min (blue) and 60 min (brown). Below is a zoom of this signal.



**Figure 5.** *wtrm\_avg\_TrmTmp\_Brg1* signal. The top is representation of the whole set of data available from the WT#81 for the time scales of 5 min (blue) and 60 min (brown). Below is a zoom of this signal.

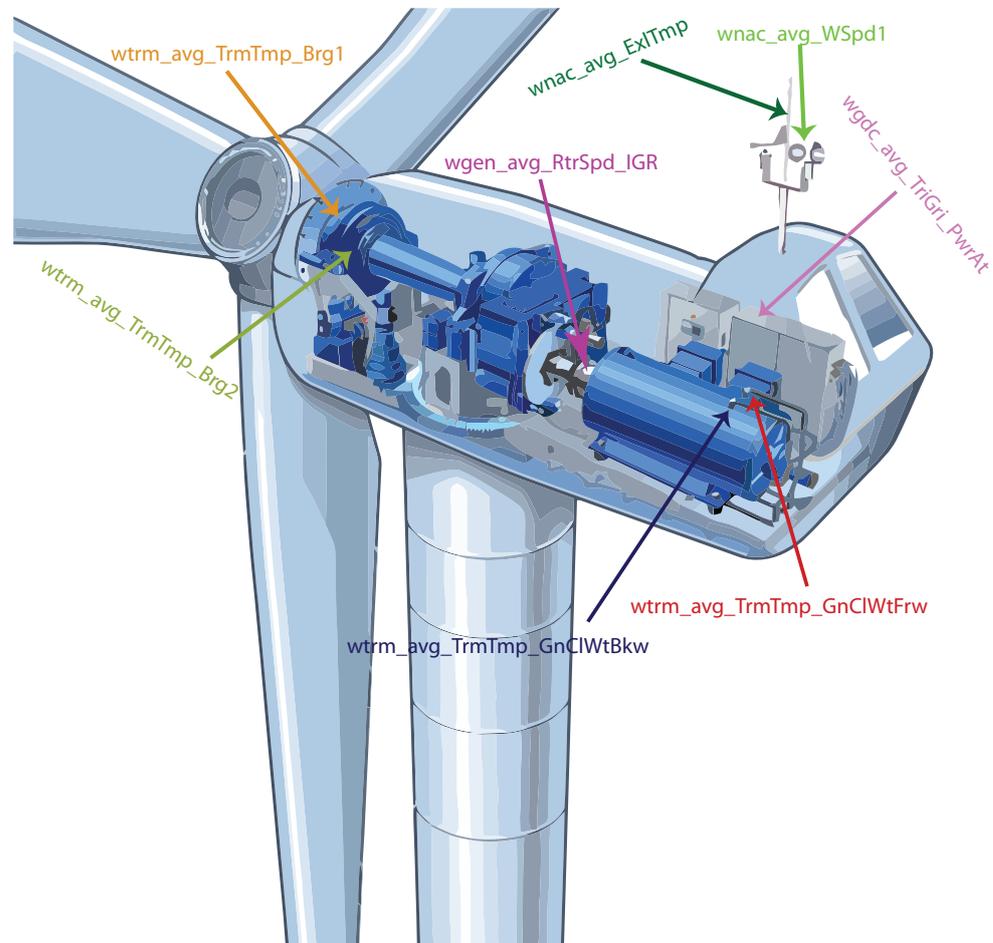


**Figure 6.** Detail of the *wtrm\_avg\_TrmTmp\_Brg1* used as a target. We note that on the 5-min scale (blue), some zero values appear occasionally. This phenomenon occurs randomly on the 5-min scale. At larger scales it disappears due to the averaging effect.

### 3. Results

In this section we present some experiments, based on the signal representation described above for time scales of 5, 10, 15, 30, and 60 min. Specifically, three type of experiments are conducted. In the first one, a target variable is estimated using other variables, while the data is used at different time-windows, from 5 min to 60 min. In the second experiment, we repeat the previous experiment but explore the effect of changing the number of the nodes in the SLFN network. Finally, the third experiment explores the effect of estimating a target when a variable correlated with the target is not considered as one of the input variables. Again, the performance of the models when aggregating the data from 5-min to 60-min records are compared.

The same SLFN-type network structure is used, of a specific size, which has been found to work correctly for this type of purpose [4,15], except for the second experiments in which we validate the effect of this size, in order to obtain a comparable statistical feedback between experiments. The experiments are carried out for the set of variables and targets described in Figure 7.



**Figure 7.** Internal view of a wind turbine. The main parts of the transmission chain are visible: Main bearing, gearbox, brake, generator, power control, and ambient sensors. The actual physical position where the sensors are located, for the variables listed in Table 1, are indicated by arrows.

### 3.1. Experiment 1

In this experiment, a regression model is built to estimate the target  $wtrm\_avg\_TrmTmp\_Brg1$  from the variables indicated in Table 1. The experiment consists of taking half of the available data to train the model, then testing it with the remaining half of the data. The particularity is that the models are trained and tested using different temporal aggregation. It starts with the default 5-min data provided by the SCADA system and increases to 10, 15, 30, and 60 min. In all cases, the performance of the network is evaluated through the root mean square error (RMSE) and the performance of the model is provided both in the training and test phases. In all cases, a SLFN network of 40 hidden nodes is used. For each data aggregation and for each individual WT, 25 models have been trained and the average result is provided.

Numerical values for this experiment are presented in Table 4, in which the temporal aggregation providing the lower RMSE for each WT is highlighted in bold.

**Table 4.** Results obtained averaging the RMSE of 25 models both in the training and test steps when trained with data of different temporal scales. In all cases, the trained network is a SLFN with 40 hidden nodes. The best values for all the WTs (highlighted in bold), are obtained when using 60-min aggregation.

		RMSE				
	WT_id	5 min	10 min	15 min	30 min	60 min
<b>Training</b>	80	0.9419	0.8835	0.9013	0.8622	<b>0.8548</b>
	81	0.9641	0.9218	0.9694	0.8622	<b>0.9096</b>
	82	1.9675	1.9386	2.0395	1.9727	<b>1.8542</b>
	83	0.9792	0.9999	0.9848	0.9328	<b>0.9098</b>
<b>Test</b>	80	1.2313	1.1700	1.1733	1.2094	<b>1.1414</b>
	81	3.3673	3.2796	3.2759	3.0293	<b>2.7340</b>
	82	2.4562	2.3924	2.3594	2.3460	<b>1.9705</b>
	83	2.3632	2.3699	2.2608	2.1164	<b>1.8810</b>

The results presented in Table 4 have been obtained using the same network model in all the cases (SLFN of 40 hidden nodes), to fairly compare the results between WTs and the different temporal aggregations. We have already studied SLFNs and shown that they work well for this purpose regardless of how they are trained, but there is always an optimal value for the network size (number of nodes) that achieves the best performance [4].

### 3.2. Experiment 2

To check how the network size can affect the results when changing the temporal aggregation, the same experiment of Section 3.1 is repeated, but now the number of nodes of the SLFN are changed to 30, (Table 5), 50 (Table 6), and 60 (Table 7). Likewise, in Figure 8 the RMSE obtained by a network of  $H$  hidden nodes for the range  $3 \leq H \leq 90$  and for each one of the WTs is depicted, when the models are trained with 5-min data and 60-min data.

**Table 5.** Results obtained by averaging the RMSE of 25 models both in the training and test steps when trained with data of different temporal aggregation. In all cases, the trained network is a SLFN with 30 hidden nodes. The minimum values, highlighted in bold, are obtained when 60-min temporal aggregation is used in all WTs.

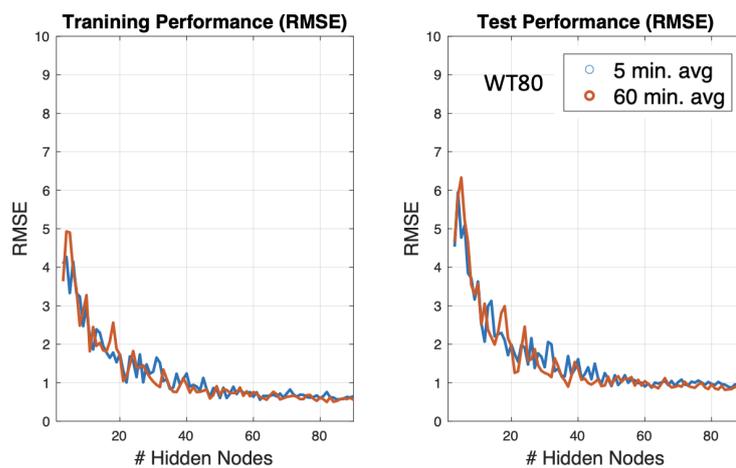
		RMSE				
	WT_id	5 min	10 min	15 min	30 min	60 min
<b>Training</b>	80	1.1329	1.1540	1.1616	1.0955	<b>1.0876</b>
	81	1.1369	1.1301	1.1522	1.1183	<b>1.1157</b>
	82	2.1591	2.2547	2.2765	2.2210	<b>2.1099</b>
	83	1.1560	1.1387	1.1599	<b>1.0820</b>	1.1101
<b>Test</b>	80	1.4261	1.4087	1.4689	1.3606	<b>1.3408</b>
	81	3.4268	3.3469	3.3642	3.1364	<b>2.7909</b>
	82	2.5473	2.4813	2.4294	2.3299	<b>2.0261</b>
	83	2.4484	2.4414	2.3644	2.1850	<b>2.0420</b>

**Table 6.** Results obtained by averaging the RMSE of 25 models both in the training and test steps when trained with data of different temporal aggregation. In all cases, the trained network is a SLFN with 50 hidden nodes. The minimum values, highlighted in bold, are obtained when 60-min temporal aggregation is used in all WTs.

		RMSE					
		WT_id	5 min	10 min	15 min	30 min	60 min
<b>Training</b>		80	0.8279	0.8294	0.7929	<b>0.7590</b>	0.7788
		81	0.7896	0.8501	0.8009	0.7809	<b>0.7495</b>
		82	1.7718	1.7437	1.8209	1.7915	<b>1.7439</b>
		83	0.8792	0.8899	0.8233	0.8402	<b>0.8165</b>
<b>Test</b>		80	1.1227	1.1235	1.0766	<b>1.0123</b>	1.0572
		81	3.3120	3.2281	3.1987	2.9994	<b>2.6630</b>
		82	2.4299	2.3658	2.3636	2.2869	<b>1.9865</b>
		83	2.3006	2.3108	2.1712	2.0670	<b>1.8304</b>

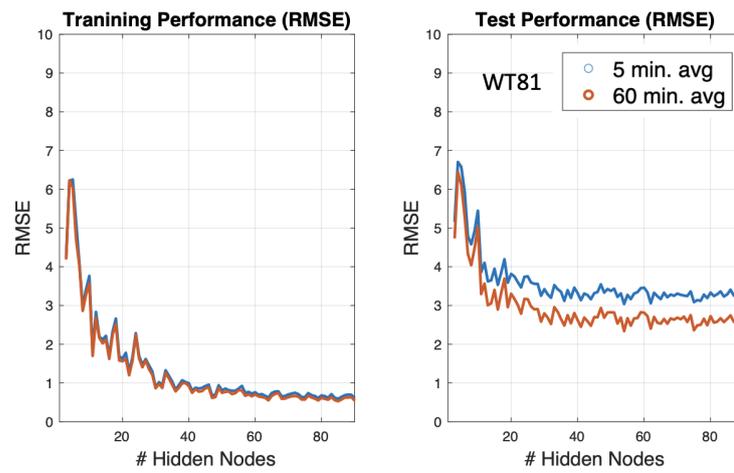
**Table 7.** Results obtained by averaging the RMSE of 25 models both in the training and test steps when trained with data of different temporal aggregation. In all cases, the trained network is a SLFN with 60 hidden nodes. The minimum values, highlighted in bold, are obtained when 60-min temporal aggregation is used in all WTs.

		RMSE					
		WT_id	5 min	10 min	15 min	30 min	60 min
<b>Training</b>		80	0.7244	0.7091	0.7004	<b>0.6702</b>	0.6981
		81	0.7413	0.7524	0.7127	0.6868	<b>0.6723</b>
		82	1.5828	1.6552	1.7083	1.5828	<b>1.5679</b>
		83	0.7930	0.7589	0.7639	0.7425	<b>0.7245</b>
<b>Test</b>		80	1.0282	1.0155	0.9925	<b>0.9641</b>	0.9903
		81	3.3127	3.2158	3.1985	2.9432	<b>2.6237</b>
		82	2.3972	2.3464	2.3563	2.2557	<b>1.9317</b>
		83	2.2666	2.2465	2.1481	2.0148	<b>1.7600</b>

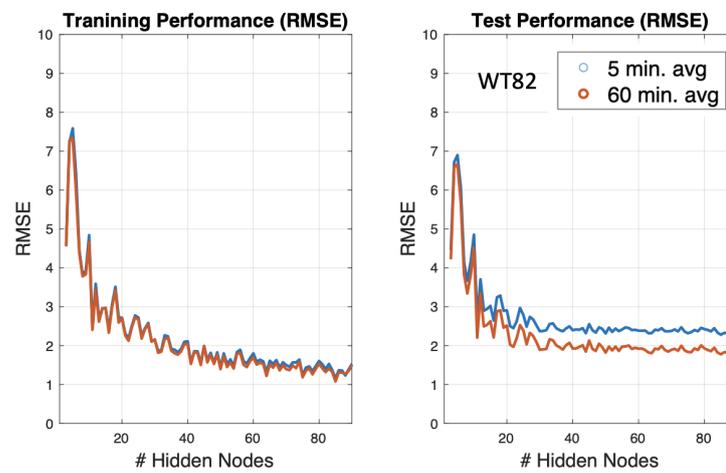


(a)

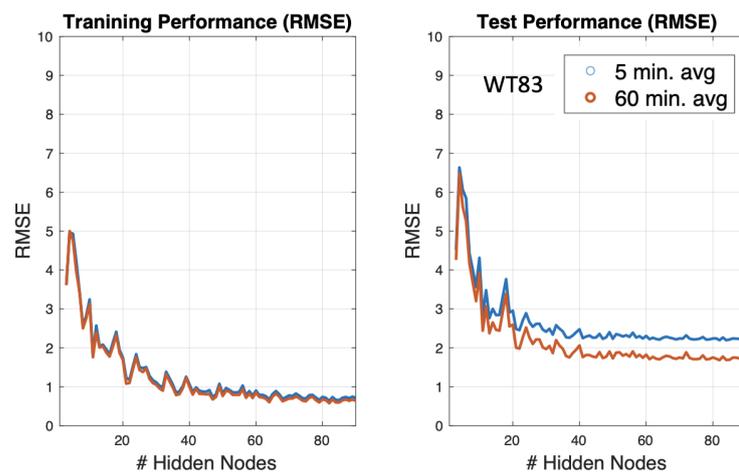
Figure 8. Cont.



(b)



(c)



(d)

**Figure 8.** RMSE achieved by a hidden  $H$ -node network both in the training and test steps when trained with 5-min and 60-min data: (a) WT#80, (b) WT#81, (c) WT#82, and (d) WT#83.

### 3.3. Experiment 3

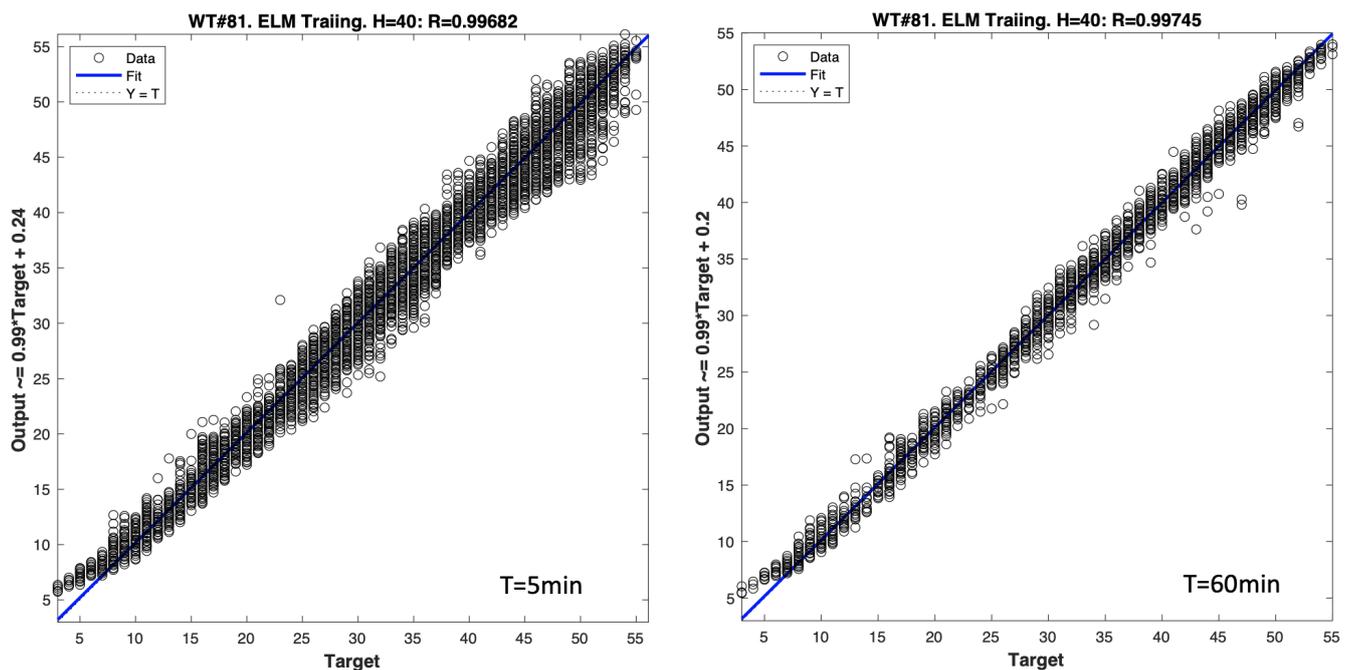
In this third experiment, the same target  $wtrm\_avg\_TrmTmp\_Brg1$  is re-estimated, in the same conditions as in Section 3.1, except that now the temperature  $wtrm\_avg\_TrmTmp\_Brg2$  is not included as a predictor. Therefore, a very important reference is drawn because there is an important degree of correlation between these two temperatures (target and predictor). Under these conditions, the predictions are expected to be less accurate. The performance of the models made for each WT and for data with different temporal aggregation are summarised in Table 8. The experiment uses the same type of network, with the same size (40 hidden models), where each value in the table is the result of averaging the performance of 25 models.

**Table 8.** Results obtained by averaging the RMSE of 25 models both in the training and test steps when trained with data of different temporal aggregation. In all cases, the trained network is a SLFN with 40 hidden nodes. The minimum values, highlighted in bold, are obtained when 60-min time aggregation is used in all WTs. Notice that now the results are less accurate than those in Table 3.

		RMSE				
	WT_id	5 min	10 min	15 min	30 min	60 min
<b>Training</b>	80	3.2367	3.1966	3.1894	3.0865	<b>2.9774</b>
	81	2.6836	2.6648	2.6396	2.5883	<b>2.5166</b>
	82	4.5850	4.6142	4.5633	4.4968	<b>4.3784</b>
	83	3.1191	3.1055	3.0547	3.0083	<b>2.9019</b>
<b>Test</b>	80	4.0724	3.9918	3.9882	3.8870	<b>3.7598</b>
	81	5.0538	5.0294	5.0006	4.8525	<b>4.6026</b>
	82	4.8108	4.7921	4.7446	4.6730	<b>4.5200</b>
	83	4.6635	4.6540	4.5470	4.4431	<b>4.2168</b>

## 4. Discussion

For the first experiment (Section 3.1), the results shown in Table 4 indicate that the models using 60-min aggregation are the ones that best fit the data. Note that in this experiment, the target to be estimated is a slowly varying temperature with a dynamic range of variation from 0 °C to 55 °C. In this case, the model fits very well, possibly due to the fact that one of the input variables is also a temperature value in the same transmission chain, as shown in Figure 7, although the dynamic range of the target is from 0 °C to 55 °C, for the input variable the range is from 10 °C to 70 °C (see Figure 3). Another way to visually observe the results presented in Table 4 is by means of the regression line obtained when training and testing a model. Figure 9 compares, for the WT#81, the regression line obtained using 5-min and 60-min data. As can be seen, the best adjustment is obtained when using 60-min data, with a regression value of  $R = 0.99745$ , while  $R = 0.99682$  using 5-min data. When these models are tested with the second half of the available data, the difference is even more visible in favour of the model using 60-min data. In Figure 10, the curve on the left shows the line obtained in the test of the 5-min model, obtaining a regression value of  $R = 0.9224$ , while on the right the 60-min model obtains a better result with  $R = 0.9487$ .



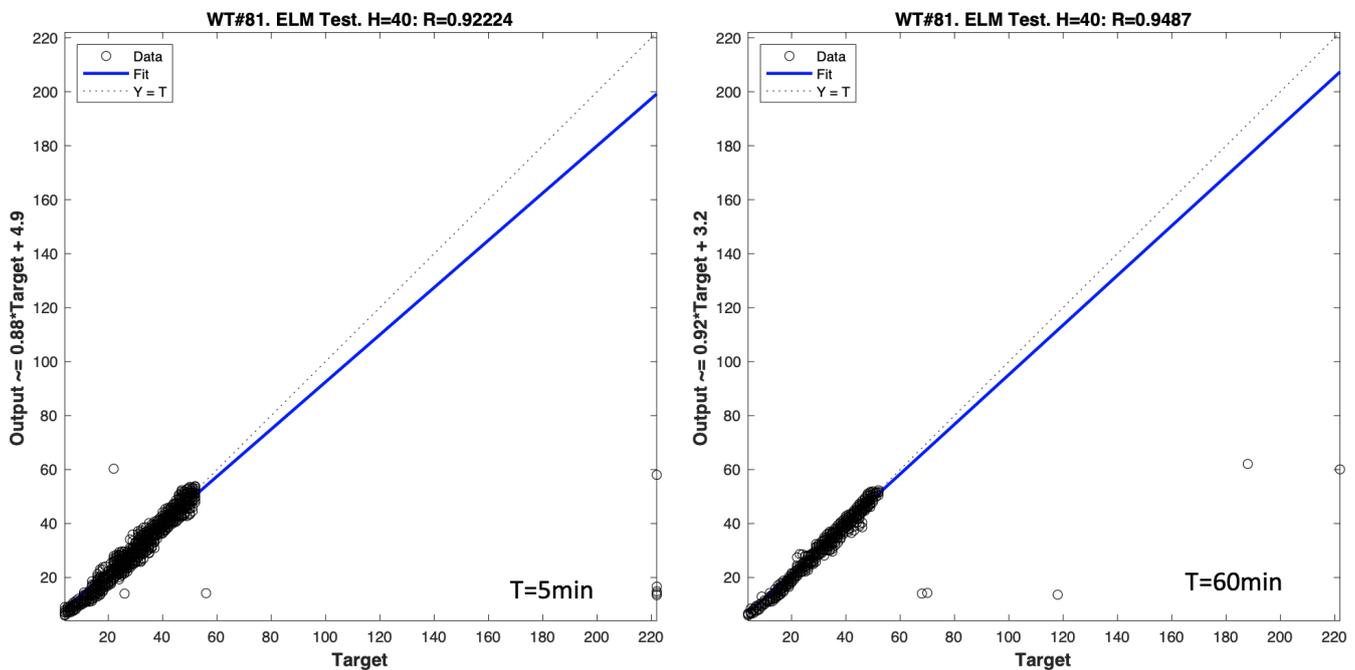
**Figure 9.** Regression lines obtained in the training step for the WT#81 model using 5-min data and (left) 60-min data (right). In both cases, a SLFN network of  $H = 40$  hidden nodes is trained to build the model.

This same result can be visualised in terms of the deviation of the regression lines of the output data with respect to the 45-degree line that would represent the perfect estimation. Note that in these two representations the data and target axes have been extended beyond the dynamic range of the data in order, first, to make the deviation of the regression lines more evident and, second and more importantly, to show how the two models detect the presence of inconsistent values between the prediction made of the target and the measurement obtained from it. This happens when the variables are changing because of a failure (warning or alarm), which generates a bad estimation detected by our normality model, or are due to a miss-function of some sensor. Indeed, in some cases the problem is due to a failure of the sensors that is particularly difficult to detect in other ways. It occurs when the signals are ‘stuck’ and the sensor sends the same value every 5 min. This can happen both in the variables and in the target. This type of failure is not detected by the outlier tests, and hence it is transparent in the pre-processing stages, but the normality model can allow one to detect them.

Results obtained for experiment 2 (Section 3.2) show that 60-min aggregation obtains the best results in all the cases, except for a couple of cases in which 30-min aggregation is proven to be better, only for the WT#80. As shown in Figure 8, in the training step the effect of the value of  $H$  is not noticeable, but in the test step we can observe how 60-min aggregation always outperforms the original 5-min time window, specially for WT#81, WT#82, and WT#83. For this specific experiment,  $H = 60$  seems to provide even better results than  $H = 40$ , but checking the numbers in detail, available in Table 4–7 (for  $H = 40$ ,  $H = 30$ ,  $H = 50$ , and  $H = 60$ , respectively), we can confirm that  $H = 40$  is the only one in which the 60-min aggregation is always the best one in the test phase. The results for  $H = 50$  and  $H = 60$  have one case in which 30-min aggregation is better than 60-min aggregation. Finally, for  $H = 30$  the results are all better for 60-min aggregation but they are worse when compared with  $H = 40$ .

In [4], the ELM paradigm was validated by addressing the problem of the optimal size of the SLFN network. Given the simplicity of SLFN, its architecture only depends on the number of hidden nodes  $H$  (and to a much lesser extent on the activation function considered). As detailed in [4], the accuracy in the training phase is increased when  $H$  is

increased, but in the test phase there is a value of  $H$  at which the accuracy starts to stabilise and the improvements are no longer significant.



**Figure 10.** Regression lines obtained in the test step for the WT#81 model using 5-min data and (left) 60-min data (right). In both cases, a SLFN network of  $H = 40$  hidden nodes is used to test the models.

In fact, from this stabilisation point, there is a wide range of  $H$  values that provide good performance in the test phase. It is worth noting that the optimal measure of  $H$ , the value in which the best accuracy is obtained in the test phase, varies slightly for each WT. Beyond these values, if the number of nodes continue to increase, the test results will worsen due to an over-fitting problem in the training. As shown in the experiment, the network starts to work well from 40 hidden nodes onward. The improvements in the test phase obtained by using 50 and 60 nodes are very small, although still appreciable.

Finally, and with regard to the third experiment (Section 3.3), it confirms that not using a signal as an input variable highly correlated with the target when estimating it (the temperature  $wtrm\_avg\_TrnTmp\_Brg2$  in the case of our target temperature  $wtrm\_avg\_TrnTmp\_Brg1$ ), which makes it more difficult to predict the target temperature. However, more importantly is that despite the reduction on the performance, it is again observed that the models trained with 60-min data are the ones that perform better in terms of RMSE, both in the training and in the test step.

While it has been already proved in [4] that the regression models are able to detect the deterioration of the systems and in a quasi identical way as in [15], in the experimental results it is shown that averaging the SCADA system results beyond the default 5-min data provided by modern WTs, which allows for models that better fit the data. This happens persistently when calculating the projections over a longer period of time and for all the experiments, based on the signal representation described, for time scales of 5, 10, 15, 30, and 60 min.

Although it is likely that some of the improvements in prognosis systems that will be seen in the near future will come from the possibility of using high-frequency SCADA data, as that will prevent the loss of information on some of the most important variables in the WT systems, what is currently happening is that there are a large number of wind farms in operation around the world that need to be properly operated and maintained in order to be as competitive as possible to have a long useful life. Most of these parks operate on SCADA systems that perform 5-min prognosis, or even 10-min prognosis in the

oldest ones, and therefore these are the data available for the design and exploitation of the prognosis models. This work suggests that models developed and operated at 60 min fit the data better than models developed and operated at 5 min, detecting anomalous situations without losing performance, but on the contrary, even improving them. This is not a minor issue because wind farms can have tens or hundreds of WTs and at the same time each turbine consists of a large number of systems and subsystems. The fact of monitoring them at 60 min reduces the computational requirements by a factor of 12 compared to working at 5 min, while also reducing the requirements associated with the time and transmission of these data.

## 5. Conclusions

High-frequency sampling is a preferable option when designing wind farm prognosis systems, as it allows to better capture the dynamics of the variables involved in wind turbine monitoring [8,10]. Low frequency sampling (5 min) commonly used in the SCADA system removes much of the information from the measured signals, so that information is lost. This is the case, for example, of the power signal, in which the information provided by the SCADA system contains already less information compared to the original one because the medium and high frequency components have already been lost due to the long sampling period. However, this does not imply that these variables cannot be used for wind turbine prognosis, as has been shown in other studies [4,9,12,17].

When high-frequency sampling is not an option or not available, or if a simpler system has to be designed, to process, store, and send smaller amounts of data, aggregation to longer time window frames is a possible and useful option. This option has been explored in this work, and experimental results show that models using 60-min aggregated data outperform results obtained with the original 5-min data. Three different experiments demonstrate that better predictions were obtained using 60-min data, which simplifies the model and the amount of data to store, process, send, and/or display.

To our knowledge, this is the first work investigating this option and the results make it interesting and potentially useful. Our experimental results suggest that averaging over long periods of time improves performance because the variability of a signal is usually reduced. In future work we will deeply investigate that, and will consider other subsystems and target variables, along with the possibility of extending the models beyond normality models.

**Author Contributions:** Conceptualisation: P.M.-P., A.B.-M., and J.S.-C.; methodology: P.M.-P. and J.S.-C.; software: A.-B.M., P.M.-P., and A.B.-S.; validation: P.M.-P. and J.S.-C.; formal analysis: A.B.-S., P.M.-P. and J.S.-C.; investigation: P.M.-P., A.B.-M., and J.S.-C.; resources: J.S.-C.; data curation: A.-B.M.; writing—original draft preparation: J.S.-C. and P.M.-P.; writing—review and editing: A.B.-S., A.B.-M., P.M.-P., and J.S.-C.; visualisation: P.M.-P. and J.S.-C.; 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.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Restrictions apply to the availability of these data. Data was obtained from Smartive.

**Acknowledgments:** The authors would like to thank Smartive (<http://smartive.eu/> (accessed on 1 June 2021)) for providing the data used in the experimental part.

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

## Abbreviations

The following abbreviations are used in this manuscript:

CMS	condition monitoring systems
DANN	deep artificial neural networks
ELM	extreme learning machines
LCOE	levelised cost of energy
NBM	normal behaviour model
O&M	operation and maintenance
OPC	Open Platform Communications
RMSE	root-mean-square error
PLS	partial least squares
RMSE	root mean square error
SCADA	supervisory control and data acquisition data
SLFN	single-hidden layer feedforward neural networks
SVM	support vector machines
WT	wind turbine

## References

1. Wang, H. A survey of maintenance policies of deteriorating systems. *Eur. J. Oper. Res.* **2002**, *139*, 469–489. [CrossRef]
2. 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]
3. Zhang, P.; Lu, D. A survey of condition monitoring and fault diagnosis toward integrated O&M for wind turbines. *Energies* **2019**, *12*, 2801.
4. Marti-Puig, P.; Serra-Serra, M.; Solé-Casals, J. Wind Turbine Prognosis Models Based on SCADA Data and Extreme Learning Machines. *Appl. Sci.* **2021**, *11*, 590. [CrossRef]
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. *Renew. Power Gener. IET* **2014**, *8*, 390–397. [CrossRef]
7. Conroy, N.; Deane, J.; Ó Gallachóir, B.P. Wind turbine availability: Should it be time or energy based?—A case study in Ireland. *Renew. Energy* **2011**, *36*, 2967–2971. [CrossRef]
8. Gonzalez, E.; Stephen, B.; Infield, D.; Melero, J.J. Using high-frequency SCADA data for wind turbine performance monitoring: A sensitivity study. *Renew. Energy* **2019**, *131*, 841–853. [CrossRef]
9. Fischer, K.; Coronado, D. *Condition Monitoring of Wind Turbines: State of the Art, User Experience and Recommendations*; Fraunhofer-IWES: Bremerhaven, Germany, 2015.
10. Gonzalez, E.; Stephen, B.; Infield, D.; Melero, J. On the use of high-frequency SCADA data for improved wind turbine performance monitoring. *J. Phys. Conf. Ser.* **2017**, *926*, 012009. [CrossRef]
11. Yang, W.; Tavner, P.J.; Crabtree, C.J.; Feng, Y.; Qiu, Y. Wind turbine condition monitoring: Technical and commercial challenges. *Wind Energy* **2014**, *17*, 673–693. [CrossRef]
12. 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]
13. Blanco-M, A.; Solé-Casals, J.; Marti-Puig, P.; Justicia, I.; Cárdenas, J.J.; Cusidó, 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 (IWOB), Funchal, Portugal, 10–12 July 2017; pp. 1–7.
14. 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.
15. Encalada-Dávila, Á.; Puruncajas, B.; Tutivén, C.; Vidal, Y. Wind Turbine Main Bearing Fault Prognosis Based Solely on SCADA Data. *Sensors* **2021**, *21*, 2228. [CrossRef] [PubMed]
16. Stehly, T.; Beiter, P.; Duffy, P. *Cost of Wind Energy Review*; National Renewable Energy Laboratory: Golden, CO, USA, 2020; NREL/TP-5000-78471. Available online: <https://www.nrel.gov/docs/fy21osti/78471.pdf> (accessed on 1 June 2021).
17. 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]