

Detecting Bearing Failures in Wind Energy Parks: A Main Bearing Early Damage Detection Method Using SCADA data and a Convolutional Autoencoder

Christian Tutivén^a, Ángel Encalada-Dávila^a, Yolanda Vidal^{*,b,c} and Carlos Benalcázar-Parra^d

^aESPOL Polytechnic University, Escuela Superior Politécnica del Litoral, ESPOL, Faculty of Mechanical Engineering and Production Science (FIMCP), Mechatronics Engineering, Campus Gustavo Galindo Km. 30.5 vía Perimetral, Guayaquil, 09015863, Ecuador

^bControl, Data, and Artificial Intelligence (CoDALab), Department of Mathematics, Escola d'Enginyeria de Barcelona Est (EEBE), Universitat Politècnica de Catalunya (UPC), Campus Diagonal-Besos (CDB), Eduard Maristany, 16, Barcelona, 08019, Spain

^cInstitute of Mathematics (IMTech), Universitat Politècnica de Catalunya (UPC), Pau Gargallo 14, Barcelona, 08028, Spain

^dUniversidad ECOTEC, Km. 13.5 Samborondón, Samborondón, EC092302, Ecuador

ARTICLE INFO

Keywords:
wind turbine
autoencoder
fault detection

ABSTRACT

Wind energy maintenance and operation costs can total millions of dollars each year in an average industrial-size wind park. Therefore, moving from preventive and corrective maintenance to predictive maintenance is imperative in the wind energy sector. This paper contributes to this challenge by providing a main bearing early damage detection technique that exclusively uses standard SCADA data (10-minute average) and a convolutional autoencoder (CAE) with the following contributions. *i*) Entirely semi-supervised (not requiring the labeling of data through work order logs and avoiding the problem of data imbalance between classes) based only on healthy data, thus expanding its range of application (even when the failure of interest has never occurred in the park before). *ii*) Validated using real-world SCADA data and shown to be resistant to seasonality, operational and environmental conditions. *iii*) Reliable predictions with minimum false alarms thanks to a specially designed fault prognosis indicators (FPIs) based on the image mean square error (IMSE) metric. *iv*) The early warning is achieved months in advance, thus providing adequate time for plant operators to plan properly. *v*) The main use of exogenous variables in the model (variables that are not affected by other variables as for example wind speed, wind turbulence, and ambient temperature) guarantees detection of damage directly related only to the low-speed shaft temperature (the only non-exogenous variable used by the stated model). *vi*) Finally, the proposed strategy is validated in a wind park made up of 12 wind turbines.

1. Introduction

The global energy system is undeniably under change. After the Ukrainian invasion, widespread acceptance and use of renewable energy is the fastest and cheapest path to greater energy independence, as well as the key to combating climate change [1]. Renewable-based electrification, according to European Commission forecasts, would be important for Europe to achieve carbon neutrality by 2050. Wind energy is a critical component in achieving this objective, as it is required to account for 50% of the European Union's electricity mix, with 81% coming from renewables, [2]. However, the heart of the issue of the success of the wind industry is the reduction in its levelized cost of energy (LCOE). The LCOE of a wind park is calculated by integrating many factors [3], with operating and maintenance expenses playing a considerable role (20-25% in onshore wind

parks and 25-30% in offshore ones). As a result, optimizing maintenance procedures is a critical aspect.

Energy output losses due to downtime (induced by unscheduled asset repair) and the expenses associated with component replacement can total millions of euros every year in any industrial wind park. As a result, it is critical that the wind sector transitions from corrective and preventive maintenance to predictive maintenance (planned on an as-needed basis depending on asset condition), see [4]. Condition based maintenance is based on actual and timely information gathered to monitor the actual asset through a network of sensors (such as vibration, temperature, oil analysis, and acoustic emissions) and warns operators before the catastrophic damage occurs, thus being able to program maintenance around non-production and replacements availability intervals. Digitalization and artificial intelligence are crucial technologies in this approach for improved exploitation of information in enormous amounts of data [5]. The general goal is to identify changes in condition that represent deviations from normal operation and signal the development of a fault [6].

Predictive maintenance is a broad field of study that has been effectively applied to a wide range of applications. However, applying it to complicated systems such as wind turbines (WTs), which are megastructures that operate under a variety of operational and climatic circumstances, as

*This work is partially funded by the Spanish Agencia Estatal de Investigación (AEI) - Ministerio de Economía, Industria y Competitividad (MINECO), and the Fondo Europeo de Desarrollo Regional (FEDER) through the research project PID2021-122132OB-C21.

*Corresponding author.

E-mail address: yolanda.vidal@upc.edu (Yolanda Vidal)

ORCID(s): 0000-0001-6322-4608 (C. Tutivén); 0000-0001-6259-8464 (Á. Encalada-Dávila); 0000-0003-4964-6948 (Y. Vidal*); 0000-0003-0792-4372 (C. Benalcázar-Parra)

well as in hazardous settings (such as offshore), remains a challenge. Furthermore, the most recent innovations tend to require expensive, especially fitted sensors, which are not economically viable for turbines that are currently in service, much less if they are toward the end of their lives. This is significant since 38 GW of wind parks in Europe are expected to reach the end of their useful lives over the next five years. According to current trends, approximately 2.4 GW will be retired for re-powering [7], and 7 GW will be completely decommissioned. The remaining 28.6 GW will remain operational and will be evaluated for life-extension services. In this area, data-driven maintenance strategies using existing supervisory control and data acquisition (SCADA) data (found in all industrial-sized WTs) are a feasible low-cost solution.

Because the SCADA data was originally designed solely for operation and control, using them for predictive maintenance is a considerable quest [8]. SCADA data can contain more than 200 variables, have a very low sampling rate (usually 10-minute averaged), are dependent on the operational region of the WT as well as the environmental condition, and are temporal series with significant seasonality. Additionally, when SCADA systems were first implemented, the value of maintaining maintenance work order records with comprehensive and standardize comments was unclear (as artificial intelligence was not pictured to aid in this application). On top of this, most of the available data are from healthy operation, making them highly unbalanced data sets. Despite these difficulties, lately, the topic of using SCADA data for early fault detection purposes has gained increased attraction, [9]. However, many challenges are still to be addressed in current and future research. The next paragraph highlights the five main challenges in this research area.

First, supervised algorithms are used in a substantial proportion of papers. Despite their promising performance, [10; 11], in a real application their direct use is almost precluded as they require historical labeled faulty and healthy data. This is a significant disadvantage because getting labeled information from WT operational data is often difficult (as maintenance records are not standardized), time-consuming and prone to errors, and results in a severely imbalanced data set. Although the problem of data imbalance can be addressed with strategies such as few-shot learning [12], supervised approaches cannot be applied in those cases where failure data is not available, that is, they cannot be used directly to wind parks where the fault of interest has not yet happened. Second, a considerable amount of references make use of simulated SCADA data, [13], or experimental data, [14], to validate the stated methodologies. Although this is acceptable, it is an important liability, since depending on these data may not generalize properly to genuine real-world scenarios. Third, the vast bulk of literature derives conclusions based on little data, commonly 1 to 4 WTs, [15]. Therefore, again, it is unclear if these strategies will apply to the entire wind farm. Fourth, some references contribute strategies that lead to a high number of false alarms, [16],

thus making the contribution not convenient in the real application. In this area, it is noteworthy the work in [17] on how to set the alarm triggering control limit. Fifth, but not least, a non-negligible number of papers detect the damage with less than a week's notice, see [15] and [18], being useless in real-world applications.

On this basis, this work proposes a main bearing early fault detection strategy based on a convolutional autoencoder (CAE) and solely on standard WT SCADA data (10-minute average) being its main contribution that addresses, all at the same time, the following seven main challenges found in the literature. *i*) It uses only standard SCADA data (10 minute average); thus, it can be applied to any wind turbine. Installation of additional sensors specifically tailored for condition monitoring can be costly as it involves the purchase and installation of new hardware, as well as the ongoing maintenance and calibration of these sensors. Using data from existing SCADA sensors is a more cost-effective way to predict failures, as it avoids the need to purchase and install new sensors. *ii*) It is a normal behavior model; that is, to be constructed (trained) only requires normal (healthy) data. Since it does not require any faulty data, any wind park (even those where the failure of interest has not yet occurred) can benefit from it (thus expanding its range of application), and it avoids the problem of highly unbalanced datasets. *iii*) It is validated on real (not simulated or experimental) SCADA data and has been shown to be robust to seasonality and operating and environmental conditions (since it is trained using the complete data in one year of operation) and also tested over a whole year of operational SCADA data. *iv*) Mainly exogenous variables are used to construct the model (variables whose cause is external to the wind turbine itself, such as ambient temperature, wind speed, and wind turbulence) together with the low-speed shaft temperature, which is the closest SCADA variable to the main bearing. In this manner, only faults that affect the low-speed shaft temperature will be detected. *v*) The warning is given months in advance of the fatal fault, thus allowing wind park operators to program the maintenance, in contrast to a non-negligible number of papers based on SCADA data that detect the fault less than a week in advance, thus not being helpful in a real application. The conceived methodology allows for this early prediction as the vast majority of different main bearing failure modes are associated with heat release; that is, when a crack initiates or propagates, when friction or wear is present, there is heat release. With the stated methodology, that relies on SCADA data associated with temperature variables, these events (of heat release) can be detected. Note that the low sampling rate of SCADA data (10-minute average) hinders information in variables with a fast dynamic (e.g., vibrations); however, as temperature variables have a slow dynamic, their SCADA data still contain relevant information. *vi*) It advances an indicator based on an exponential weighted moving average filter, depending on the weekly number of anomalies, to reduce the number of false positive alerts contrary to a substantial number of references that result in a significant

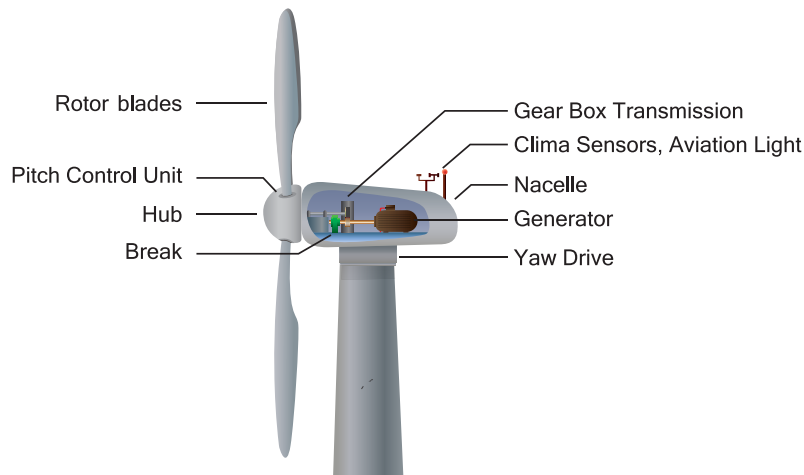


Figure 1: Components and subsystems of a WT.

number of false alerts, making the contribution inconvenient in the real world, as it would result in alarm fatigue for operators. *vii*) The validation is done at the wind farm level that includes 12 WT, unlike the majority of the literature that bases their results on a relatively small amount of data (usually only 1 to 4 wind turbines).

The paper is organized as shown below. Section 2 gives a succinct statement of the wind park under study, as well as introduces the primary forms of bearing damage. Section 3 introduces the available SCADA data and work order records, as well as a full explanation of data preprocessing. In Section 4 the methodology is described in full, as well as the approach of the suggested FPI to minimize false positive alarms. The findings are explained in Section 5. In the end, the conclusions are reached in Section 6.

2. Wind Park and Main Bearing Fault Review

The wind park consists of twelve WTs which are 1.5 MW power and have 77 m rotor diameter. The main WT systems are shown in Figure 1. In [8] more detailed technical specifications for this kind of WTs and the characteristic double spherical main roller bearing that is used on them are explained in depth.

Because of main bearing fault is the fault under study, it is critical to comprehend the many ways a spherical roller bearing might fail. This type of bearing is a significant mechanical component, as it is utilized to provide mobility to other prominent and massive components, such as shafts. SKF, a Swedish bearing and seal manufacturer, has categorized various bearing failure modes using ISO-15243 standard [19]. For more details, see [8] where it is deeply described each bearing failure mode and several illustrative pictures are also included.

3. Real SCADA Data Preprocessing

Wind park's data are gathered from February 2017 to November 2018. Data rely on various WT components,

reaching approximately 160 variables (see [8] for checking a complete classification of these variables). However, as established in Section 1, only exogenous variables are used as part of the proposed methodology that also joins the low speed shaft temperature. Actually, the last one is the most important variable, since it is the most related to the fault under study. Those exogenous variables are closely related to environment, which have a direct impact on the variable of interest. For example, ambient temperature can drastically affect the low-speed shaft temperature due to the season transition, from winter to summer, or vice versa. Actually, there are different possibilities to take into account the effect of ambient temperature. One option is subtracting ambient temperature from all other related-temperature variables to avoid seasonality, like in [20]. Another option is to retain this variable as part of the input variables set to add information to the model about data seasonality, as is proposed in this work.

On the other hand, the operational region of a WT is determined by the wind speed, which is associated with the output power [21]. Considering the sample rate of an SCADA system (10 min), for each variable there are the following statistical measurements available: mean, minimum, maximum, and standard deviation. Next, the environmental variables for the wind park are listed:

- TempAmb: ambient temperature, in ° C.
- VelViento: wind speed, in m/s.
- IndTurbul: turbulence index, which is non-dimensional.

Note that the proposed methodology is based on a WT's normal behavior model (NBM), i.e., it is trained only with healthy (normal) SCADA data to learn the normal behavior of the asset. Aside from SCADA data, information on repair work orders is also provided. The occurrence of failures, when they were resolved, and which components or subsystems were involved are detailed in these records. In detail, over the whole wind park, during 2018 there were two major

Table 1

Range of values for the selected SCADA variables based on the geographic location.

Variable	Description	Range	Units
TempEjeLento	Low-speed shaft temperature (mean)	[0, 120]	°C
IndTurbulMean	Turbulence index (mean)	[0, 40]	-
IndTurbulSdev	Turbulence index (standard deviation)	[0, 40]	-
IndTurbulMin	Turbulence index (min)	[0, 40]	-
IndTurbulMax	Turbulence index (max)	[0, 40]	-
TempAmbMean	Ambient temperature (mean)	[-5, 40]	°C
TempAmbSdev	Ambient temperature (standard deviation)	[-5, 40]	°C
TempAmbMin	Ambient temperature (min)	[-5, 40]	°C
TempAmbMax	Ambient temperature (max)	[-5, 40]	°C
VelVientoMean	Wind speed (mean)	[-5, 60]	m/s
VelVientoSdev	Wind speed (standard deviation)	[-5, 60]	m/s
VelVientoMin	Wind speed (min)	[-5, 60]	m/s
VelVientoMax	Wind speed (max)	[-5, 60]	m/s

faults (including replacement of components) that are listed as follows:

- WT2: main bearing fault –the subject of study in this work–, occurred on May 21, 2018.
- WT8: gearbox fault, occurred on March 22, 2018.

Furthermore, working with real data faces several challenges like missing data and outliers [22]. Handling these tasks determines the final quality of the data, and thus the success of the predictive model. Next, the steps to treat real data are described, that is, pre-processing of real SCADA data.

3.1. Data Cleaning and Imputation

Outliers (extreme values) are not routinely deleted in this study because, as mentioned in [23], it can result in loss of relevant information for damage detection, and even more when considering that time series play the main role in this work and save sequential information, that is, data values do not work only individually, but also as sequences or data series. In contrast, using manually set ranges based on realistic data received by several sensors may be a superior technique. For instance, in environmental variables there are certain conditions based on the weather of a geographic location to decide which measurement is valid or not. In the case of low-speed shaft temperature, this variable is generally known to be above 0°; therefore, any negative values must be eliminated, resulting in missing data that will be addressed later. Following this strategy, in Table 1 are shown the variables employed in this study with their respective ranges. Similarly, Figure 2 shows the outliers presented for the low-speed shaft temperature in a certain WT.

As outliers are eliminated, the amount of missing data increases, requiring the use of a data imputation approach to deal with this issue. In this paper, the piecewise cubic Hermite interpolation polynomial is used [24] to preserve the data’s monotonicity and to guarantee the continuity of the first derivative. Figure 3 shows how this imputation strategy

works in missing data areas in the interior. When missing data occurs towards the beginning or finish of a data set, the closest value before or after the missing values is used to fill in the gaps, correspondingly.

3.2. Data Split

Data split is commonly used to detect if a model is suffering from one of two machine learning’s common problems: underfitting or overfitting. In this work, real SCADA data are split in train, validation, and test data sets. The goal of this work is to develop a defect prediction approach that is insensitive to both operating and environmental variables; therefore, training and test datasets must include data from all working situations (all different regions of operation). Furthermore, to ensure that the detected anomalies are not due to seasonality the training (plus validation) and test datasets were split in such a way that each set had almost one year of data. In particular, the training and validation data sets cover from February 2017 to December 2017. The first 70% (33120 samples) of data is for training, while the remaining 30% (14256 samples) is for validation. Furthermore, data for the almost-full year 2018 (47808 samples) are available for testing.

3.3. Data Normalization

In a data set, variables are common to come from different sources and, therefore, their magnitudes are different. For instance, in this work variables involve speeds, temperatures, indices, etc. If this situation is not handled appropriately, the model output might be influenced by large-scale variables [22]. Thus, one imperative task in data pre-processing is normalization. There are several methods to standardize or normalize data, such as Z-score standardization, robust normalization, Min-Max normalization, among others. The last one is the selected strategy to be applied in this work. Equation 1 details this scaling:

$$\hat{x} = \frac{x - X_{min}}{X_{max} - X_{min}}, \quad (1)$$

where \hat{x} is the normalized value, x is the value to be normalized along the values’ column (each variable), X_{min} is the

Detecting Bearing Failures in Wind Energy Parks

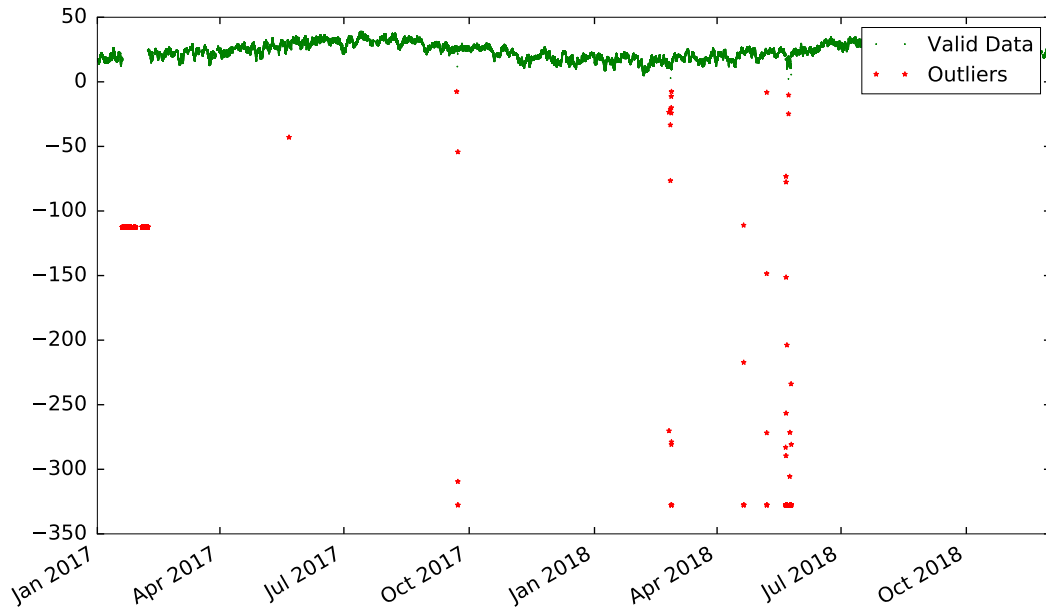


Figure 2: Outliers detected (in red) in the low-speed shaft temperature, where only positive values are valid.

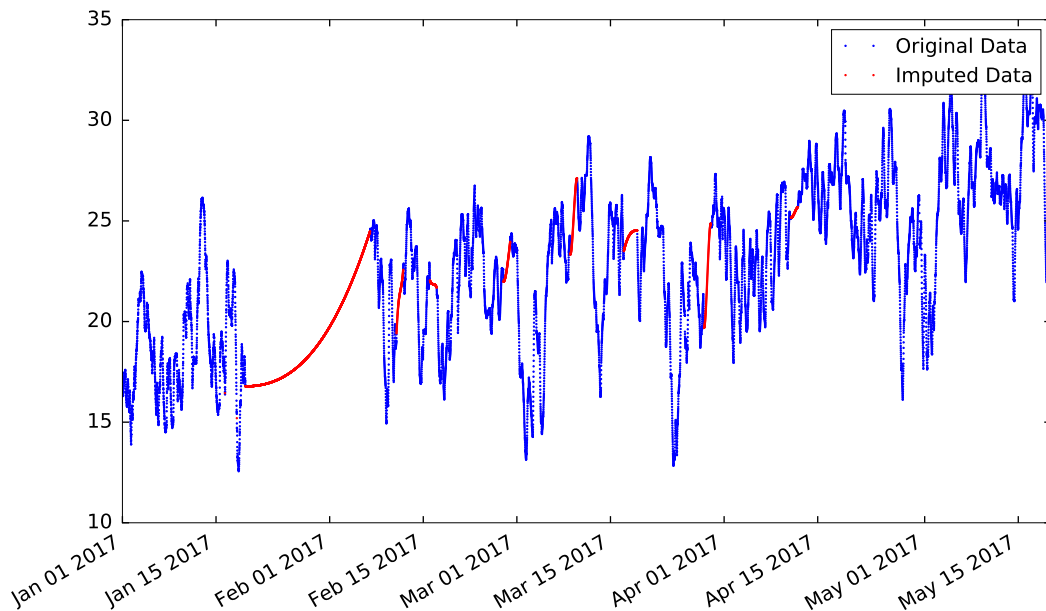


Figure 3: Imputed data (in red) in the low-speed shaft temperature by applying the *pchip* function in internal missing-data areas. X-axis is shortened for a better visualization of the data imputation technique.

minimum value of the variable (values' column) and X_{max} is the maximum one. It is noteworthy to mention that the first normalization is applied to the train data set, then with the maximum and minimum values extracted from the train data set, the validation and test data sets are normalized, i.e., data normalization must be applied after data split to avoid filtering information between data sets and adding bias to the model.

3.4. Feature Selection

When there are several variables which come from a same source like Table 1 shows, feature selection is useful since it is possible to find out those which contribute most to a model [22]. A variance threshold feature picker is employed in this case to eliminate low-variance features from each external variable. Features with higher variance are thought to provide more important information. One key point of this method is that the analysis is made on individual variables, not between features or with respect to any specific variable.

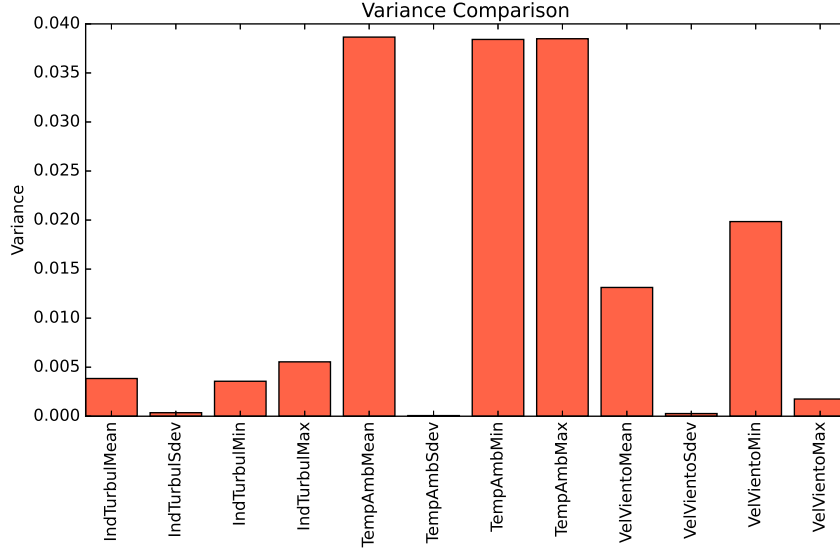


Figure 4: Selection of features based on the variance threshold.

Figure 4 illustrates the variance comparison. The main intention is to select one representative variable for each type of variable, that is, one related to low-speed shaft temperature, another one to wind speed, and so on. Finally, by applying this filtering strategy the variables selected are: TempEjeLento, IndTurbulMax, TempAmbMax, and VelVientoMin. Note that the idea of using also minimum, maximum and standard deviation of the SCADA-collected measurements, and not only the mean, has been demonstrated to be highly beneficial for data-driven modeling in [25]. Figure 5 shows the time series for each variable that has a different behavior according to its source. Then, these data are organized in a matrix (data set) where there are rows as the number of samples in each data set (train, validation, or test) and four columns corresponding to the above-mentioned variables.

3.5. Data Reshaping

One of the main ideas in this work is to convert the time series data at disposal into image information. Recalling that time series save quite important sequential information, it is needed to define the temporal length (like one day, one week, one month) of each time series (i.e., image dimension). In this way, a specially designed CAE can be utilized to capture the temporal features of the image to reconstruct the input. Considering that there are 4 inputs and one day (temporal length chosen) contains 144 samples (as the sample rate is 10 min) the constructed image has a size of $12 \times 12 \times 4$, that is 12×12 images with 4 channels. Figure 6 illustrates the data reshaping process. Finally, the result of data reshaping is 230 matrices for training, 99 for validation, and 332 for testing.

4. Fault Detection Methodology

In this section, two key parts of this work are described: the setup of the proposed CAE model and the logic of the metric used as an FPI. The next subsection will not be devoted to developing or discussing CAE networks in detail; however, certain fundamental concepts will be revisited in order to introduce the nomenclature used.

4.1. Introduction to CAE neural networks

First, it is essential to mention the paradigm of an encoder-decoder structure [26]. Generally, the input is transformed to a lower-dimensional space (encoder) and subsequently reconstructed to duplicate the original input (decoder). Recalling the principles of an autoencoder (AE) [27], it takes an input $x \in \mathbb{R}^s$ and codes it into the so-called latent representation $h \in \mathbb{R}^{s'}$ using a mapping $h = f_{(W,b)}(x) = \sigma(Wx + b)$. Then, this code is used to rebuild the input based on a reverse mapping of f , that is, $y = f_{(W',b')}(h) = \sigma(W'h + b')$, where $W' = W^T$. To sum up, each input x_i is outlined on its code h_i and then reconstructed as y_i . Obviously, during this process, the weights are optimized by trying to minimize the cost function performed.

Convolutional neural networks (CNNs), on the other hand, are hierarchical models in which convolutional layers shift with subsampling layers, with latent feature representations that retain the links between input's neighborhood and spatial locality [28]. Convolutional layers, max-pooling layers, and the classification layer [29] are the three well-defined blocks of a CNN. Actually, inside a CNN each layers' group has specific functions, for instance, the initial layers learn to detect edges and curves, and further layers combine these to detect geometric shapes, until the detecting section of the image ends up. In general, due to CNN's complexity and capacity, they have a spatial invariance property, which means CNNs learn to recognize image features

Detecting Bearing Failures in Wind Energy Parks

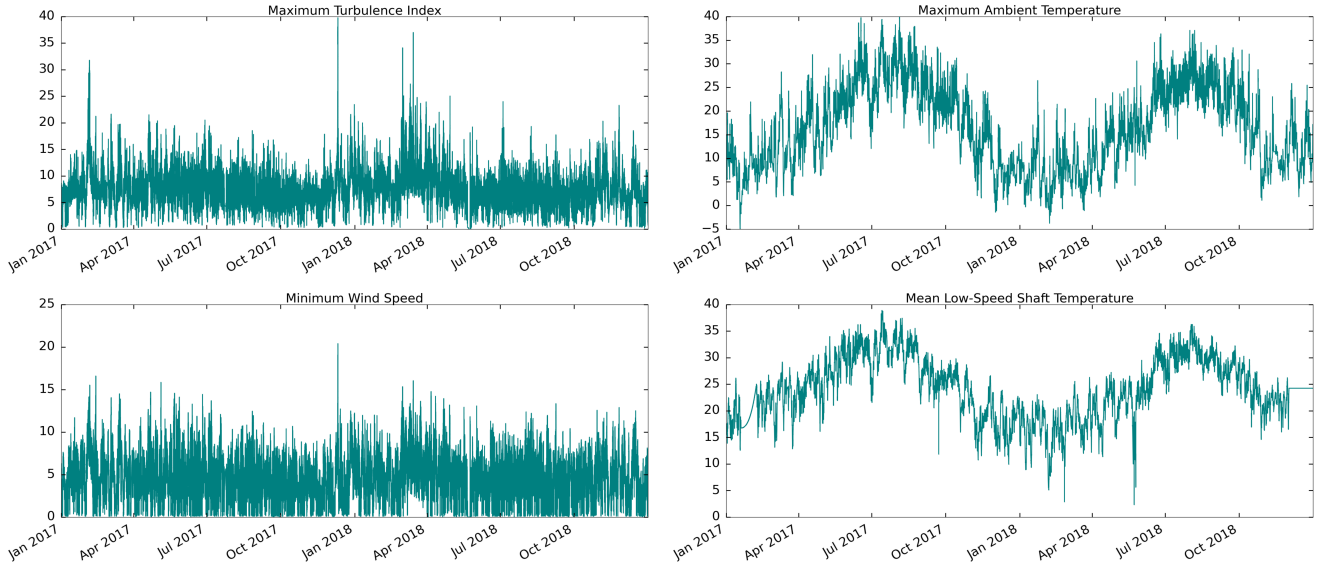


Figure 5: Time series for each selected variable where the imputed data are included. The period of time shown in the x-axis covers train, validation, and test data sets.

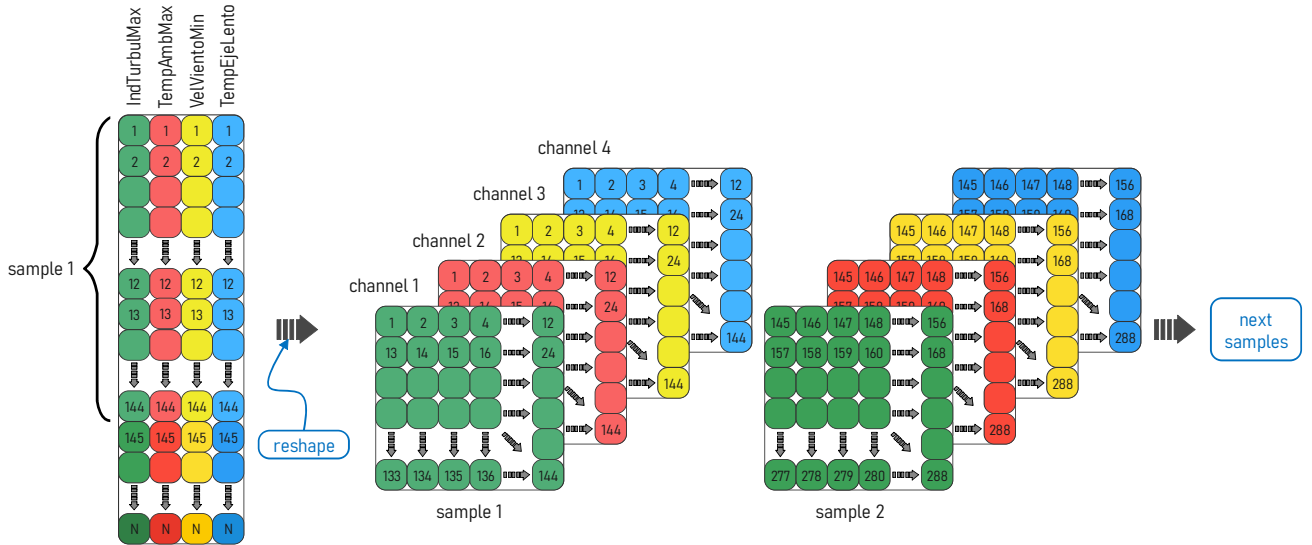


Figure 6: Real SCADA data reshape from a matrix to images with 4 channels.

anywhere in any image. For these and more reasons, CNNs are among the best image classification models and set the standard in various benchmarks [30; 29].

CAEs, which combine an AE and a CNN, do not ignore the 2D picture structure as other architectures such as AEs or denoising autoencoders (DAEs) do. Adding duplication to the parameters and requiring each feature to be global is a major issue. The goal of CNNs is to locate localized features that duplicate themselves across the input [31]. CAEs, unlike AEs, share their weights between all input locations, preserving spatial location. Except for the weights [32], the structure of a CAE is similar to that of an AE, as stated above. For example, considering a one-channel input

x , its latent representation of the j_{th} feature map is,

$$h^j = \sigma(x * W^j + b^j), \quad (2)$$

where b^j is distributed to the whole map, σ is the hyperbolic tangent function (used as an activate function), and $*$ represents the 2D convolution. Each latent map uses a single bias since each filter is intended to specialize in features of the entire input. Then, the reconstruction of the latent representation is shown in Equation 3,

$$y = \sigma \left(\sum_{j \in H} h^j * \hat{W}^j + d \right), \quad (3)$$

where d is the bias per input channel, the latent feature map is represented by H , and the flip operation is defined by \hat{W} .

Finally, the typical cost function $J(\theta)$ used to minimize the error is the mean squared error (MSE) as Equation 4 shows,

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2, \quad (4)$$

where θ are the learned parameter values or weights ($\theta_1, \theta_2, \dots, \theta_k$), n is the number of training samples, \hat{y}_i is the prediction for the i^{th} training sample using the parameters θ , and y_i is the class label for the i^{th} training sample.

On the other hand, recalling, the intention in this work is to perform early fault detection, then the temporal patterns due to seasonality of the environment or failure progress are key points to be taken into account. As stated above, the proposed methodology is based on a CAE to process the real SCADA data, where those data are disposed as matrices (embedding one day of data, that is, 144 samples) with different channels. Moreover, CNNs find spatial patterns, which are similar to finding temporal patterns in time series.

With the approach of CNNs and considering that the model is based on a NBM, that is, only healthy SCADA data are used, CAEs let to learn from that and try to reproduce the input in the output as well as possible. Then, if an image with failure patterns is introduced as input in the trained model, the output will have a higher error being this indicative of a developing fault.

4.2. Setup of the proposed CAE model

Figure 7 shows the suggested CAE model architecture. It includes two 2D convolutional layers and two 2D transpose-convolutional layers. All hidden layers employ the ReLU function, while the output layer uses the Sigmoid function to scale the data in a range of $[0, 1]$. The Adam optimization technique was used to optimize the model parameters, settling some of its hyperparameters. An initial learning rate of $\alpha_0 = 0.001$ is set, which is a common value selected for most training models. Furthermore, a gradient decay factor is used equal to $\beta_1 = 0.9$, while a squared gradient decay factor is also used equal to $\beta_2 = 0.999$, and a $\epsilon = 10^{-8}$.

The number of epochs is an essential hyperparameter that defines how many times the model trains. If the model trains with a few epochs, it may not select the best workout. However, if it trains a lot the computational cost and training increase, and even the model may fall in overfitting. Thus, to handle a trade-off, 3000 epochs were chosen. However, the 3000th model is not necessarily the best and in consequence is not saved, but to assure that the best one is saved, in each epoch it is checked if the current validation loss is less than the saved last smallest one. If that occurs, the model is saved as the best one. For instance, Figure 8 shows on WT2 the error curve of the training and validation data sets. Note that the validation loss decreases (and it is generally less than the training loss) as the number of epochs increases. The red line marks where that mentioned condition was fulfilled for the last time, that is, in the 2982nd epoch, i.e., when the model was also saved as the best one within the 3000 elapsed epochs.

Thus, recalling that this model is based on an NBM, a good reconstruction indicates that the sample is healthy, but if the reconstruction error is high, it is very likely that the sample represents a fault alarm.

To summarize, Figure 9 resumes the explained subsections that correspond to the training stage.

4.3. FPI based on IMSE Metric

It is well known that an image is basically a matrix of pixel values [33] and there are several methods to measure the distance (how much difference) between images. Thus, in this work, the image mean square error (IMSE) is utilized between the input and output images of the CAE. Equation 5 details the IMSE:

$$\text{IMSE} = \sum_{k=1}^N \frac{(a_k - b_k)^2}{N}, \quad (5)$$

where a_k and b_k are the values of the k_{th} pixel of two images A and B, respectively, while N is the total number of pixels in the matrix. As it was above-mentioned, the input of the model has 4 channels and one of them is the low-speed shaft temperature. Because this variable refers to a sensor closer to the component of interest, this channel is taken in the input and output to apply the IMSE. The fact that the other channels are not included in the metric might cause a certain doubt; however, recall that through convolutions in the model, all four channels affected the reconstruction of the image.

Now, if the computed IMSEs are used directly to establish a threshold, a large number of false positives may be generated, which would represent a decrease in the effectiveness of the methodology and lead to alarm fatigue. To avoid this problem, a technique may be used to compute the persistence over the threshold of the samples over time. To accomplish this, the exponential weighted moving average (EWMA) is proposed, since with this strategy it is possible to smooth the original spiky residual errors and keep the trend of data. In addition, EWMA deals with aging of data by assigning less weight to data as they get older. In Equation 6 is shown how an EWMA value is computed:

$$\hat{T}_{t+1} = \hat{T}_t + \lambda(T_t - \hat{T}_t), \quad (6)$$

where \hat{T}_{t+1} is the estimated value at time $t + 1$, \hat{T}_t is the estimated value at time t , and T_t is the real SCADA measured value at time t . Note that λ is present also in the equation, and this parameter leads to the memory depth of EWMA computation. λ is calculated based on its relation with s , the so-called spans, which is the period of time on which the EWMA is computed. For example, if the sampling rate of the data is 10 min and the EWMA must be calculated in a one-hour window, the value of s is 6. Equation 7 shows how s and λ are related:

$$\lambda = \frac{2}{s + 1}, \quad (7)$$

where $0 < \lambda < 1$, and $s \geq 1$.

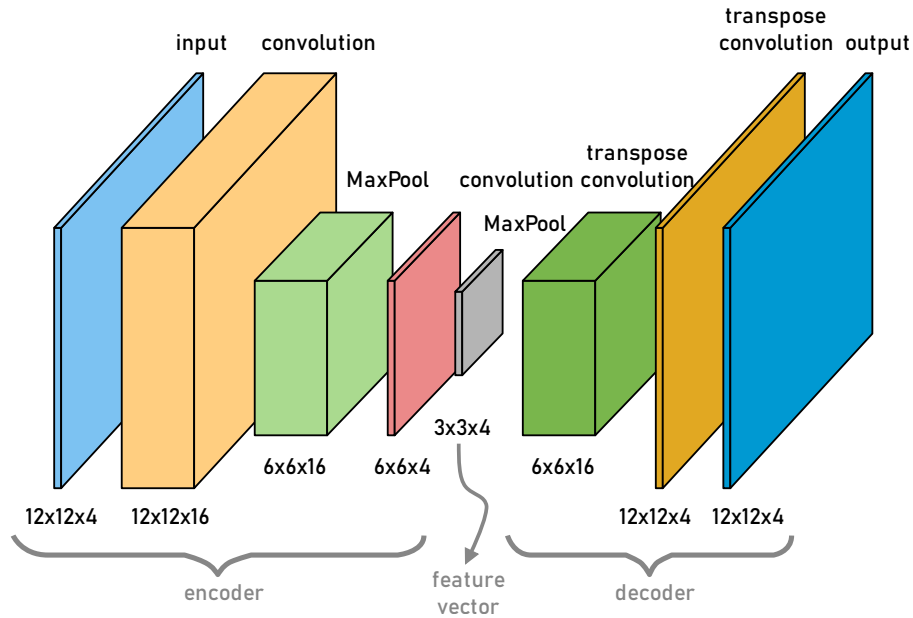


Figure 7: Architecture of the proposed CAE model.

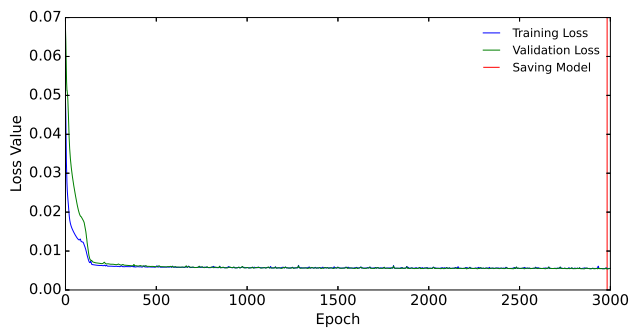


Figure 8: Training and validation loss curve during model training on WT2.

It is desirable to have weekly residual errors even after computing the EWMA where the spans were based on a weekly window, and thus the residual errors were averaged weekly (7 days). Actually, this selection is influenced by the findings of McKinnon et al. [34]. Their research, on the influence of time history on WT failures using SCADA data, tests three distinct moving windows: daily, weekly, and monthly. Compared to the others, the weekly moving window performs best at identifying failures. On the one hand, a daily window contains too much noise, leading to a large percentage of false alarms. On the other hand, a monthly window removes much information and does not allow any specification of when an anomaly occurred. Finally, to define whether a residual error represents a fault alarm or not, another step is required. In this case, with

the computed averaged errors over the train data set, a threshold is calculated by using the mean (μ) of the values and their standard deviation (σ). Equation 8 resumes this computation:

$$\text{threshold} = \mu + \kappa\sigma, \quad (8)$$

where κ represents the spacing of σ with respect to μ . Finally, Figure 10 shows the flow diagram of this stage along with the CAE model testing.

5. Results and Discussion

In this section, the stated methodology is tested on a real wind park. The results of the proposed FPI are shown (in the train, validation, and test datasets) and discussed. Moreover, the results on the test dataset are compared to the ones obtained in [8] where the same wind park was used.

Before to delve in the test-dataset results' discussion of Figure 11, it is essential to explore how to set the threshold. The training error, given in Figure 12, is certainly an important measure related to the future performance of the model. On the one hand, it is desired that similar order training errors are obtained for the different wind turbines. However, some differences are perfectly allowed between training errors of different wind turbines, as they are compensated by the threshold where each wind turbine uses its own mean (bigger training errors lead to higher values of the mean error) and deviation from the training data. On the other hand, when a WT model has a high training error, the data might have some anomalous behavior even if no work orders

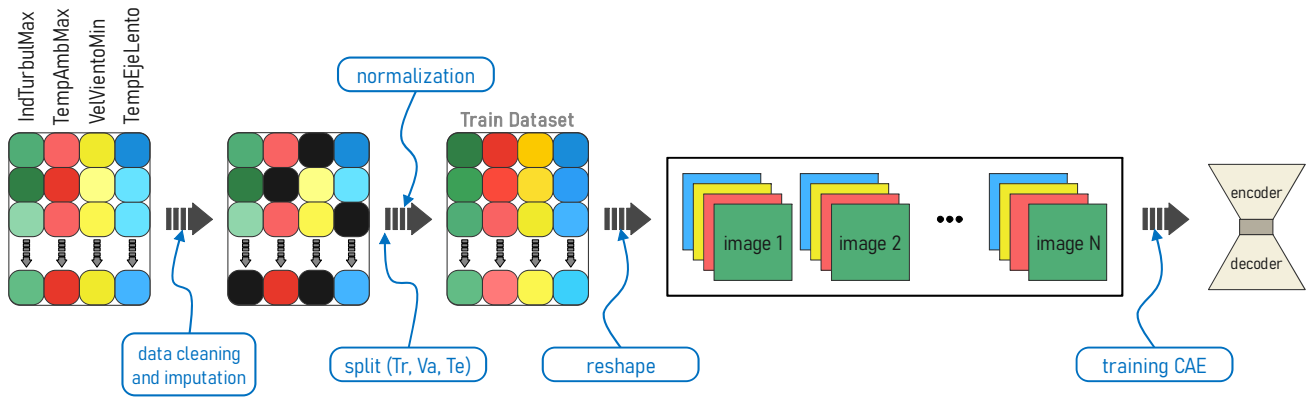


Figure 9: Real SCADA data preprocessing previous to train the CAE model.

Table 2

Assessment of false-positive alarms (checkmarks) on the training and validation data sets (varying κ values in the threshold definition).

WT ID	$\mu + 3\sigma$	$\mu + 6\sigma$	$\mu + 9\sigma$
WT1	✓		
WT2	✓		
WT3	✓		
WT4	✓	✓	
WT5			
WT6	✓		
WT7	✓		
WT8			
WT9			
WT10			
WT11			
WT12			

are reported. In this case, it is highly recommended to double check whether this turbine had issues during the year used as training. Table 2 resumes false positive alarms triggered in the model assessment using only the training and validation data set. It should be noted that there is no information on the test data set during that process. During calibration, several values of κ such as 3, 6, and 9, were tested because κ is related to μ and σ , so this parameter is typically changing as a multiple of 3. Finally, the κ value that fits better is the smallest one (since it is not suitable to have larger deviations from μ) and that triggers the least number of false positives (alarms), thus in this case κ is set to 6.

Note that the selected value for κ is empirically derived just based on the observation of the training and validation dataset where the WTs are healthy. The value of κ is set to minimize the number of false alarms over these datasets (training and validation). Therefore, there is no information from the test set (or from the knowledge of the occurred fault on the test set) to decide the κ value. Once the κ value is selected and the FPI's threshold is defined, the results on the wind park test set are shown in Figure 11.

In Figure 11, the red dotted lines correspond to the thresholds, while the green solid lines outline the weekly residual errors. Of the 12 tested WTs, eight of them are correctly predicted as healthy over the test data sets (which cover almost the entire year 2018), and four of them (WT2, WT6, WT8 and WT11) show prominent peaks above the threshold that would indicate (not yet conclusive) a fault alarm corresponding to the main bearing. In the following paragraphs, each of these fault alarms is analyzed in a comprehensive way.

On the one hand, for WT2, this alarm is correct, since, as detailed in Section 3.2, on this WT a main bearing fault occurred on 21 May 2018 and, precisely at the beginning of February 2018 two large peaks (spanning two continuous weeks) are triggered, indicating the onset of an abnormal behavior. This alarm must be used to alert maintenance team and plan ahead an inspection on the main bearing component. Also, note that between February and May, there is a decreasing trend in the residuals until they remain below the threshold. It does not mean that after the great temperature's peak everything returns to normal; by contrast, it is a typical behavior in bearing degradation. When a bearing failure mode starts, there is normally a short (but quite significant) heat release (referred as an unexpected temperature increase) [19]. Following that, the temperature returns to normal, i.e., the crack does not grow, but the bearing degradation slowly continues if it is not inspected in time. Thus, the relevance of the proffered methodology is to detect those fault onsets typically rendered as heat release or temperature increases, several months before the bearing is entirely damaged.

On the other hand, in the case of WT8 and in inspection of work orders, there is no record of a main bearing fault, but there is a record of a gearbox replacement (see Section 3.2 for more details). Thus, this fault alarm (which persists for three consecutive weeks) is correctly associated with the occurrence of an abnormal behavior, but not related to the main bearing, but to the gearbox. Although the main use of exogenous variables in the model guarantees detection of damages directly related only to the variable of interest (low-speed shaft temperature), note that the gearbox is connected

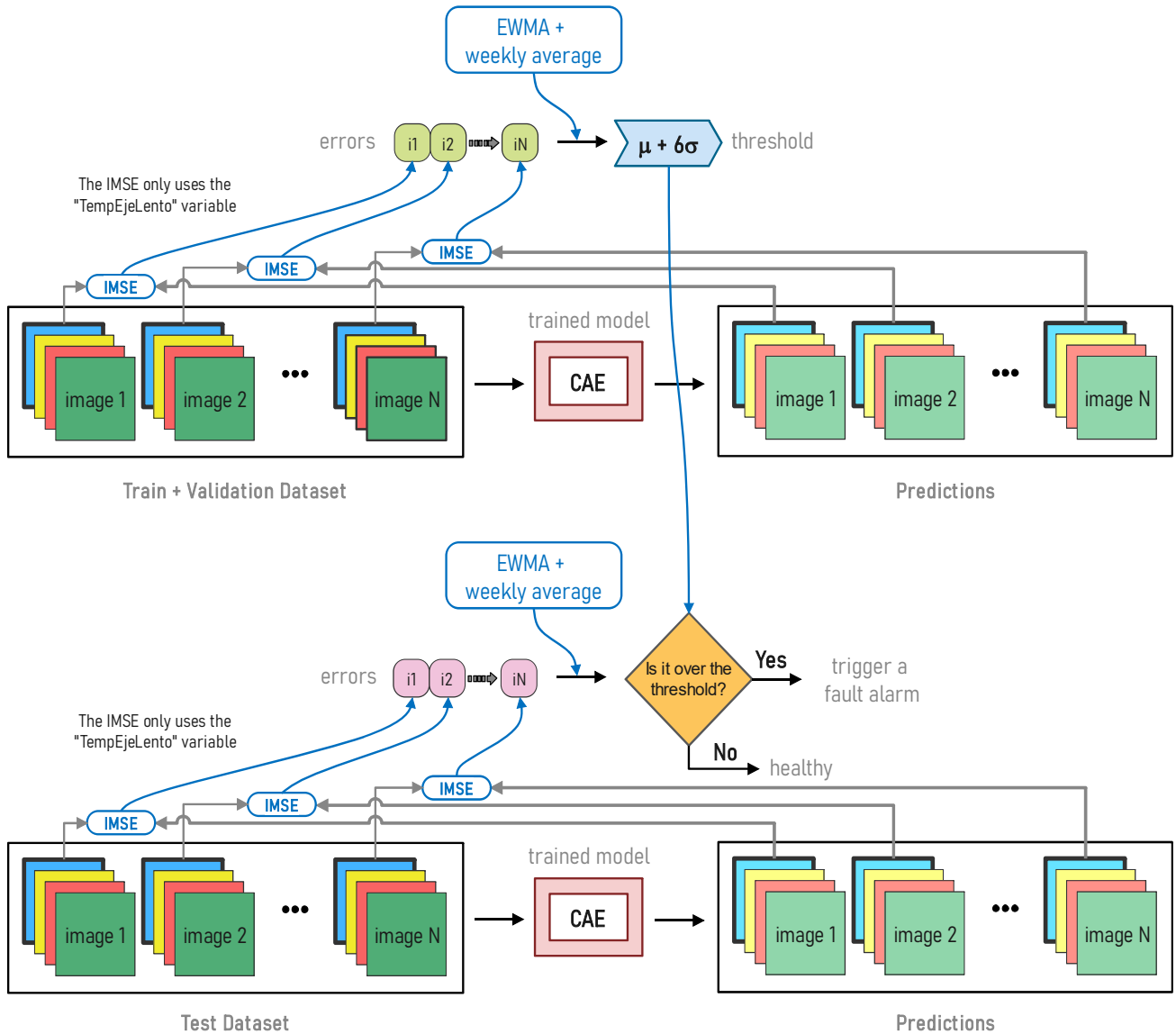


Figure 10: Fault detection methodology in the testing stage using the trained CAE model.

to the main shaft and thus the method also detects this abnormal behavior.

Finally, for WT6 and WT11, the triggered peaks would represent false positives (false alarms), since there is no record of any faults on those WTs during 2018 period. However, two false alarms represent a low false alarm rate considering that the method is validated on 12 WTs over an almost full year. Furthermore, it is significant to consider that on WT11 there is only one peak (on February 5, 2018), that is, there is no persistence (at least for two continuous weeks) for a long time, as it occurs with the analyzed WTs mentioned above. Likewise, making-decision process for triggering fault alarms based on peak's persistence relies also on wind park's holder company, since the methodology would be quite strict or moderate.

As introduced in this section, Encalada-Dávila et al. [8] worked with the same wind park by applying a methodology

based on ANN (artificial neural networks) to detect early the main bearing fault in WT. Their proposed model employs SCADA data's variables related to generated power, rotor speed, and temperatures (gearbox, generator, bearing coupling side, and bearing non-coupling side), i.e., non-exogenous variables. To sum up, Table 3 resumes the results for both works, where hits and mistakes for prediction (if the WT is healthy or faulty) on test data sets are detailed.

When comparing the results, the ANN-based model triggered two apparent false positives, while the CAE-based model triggered three. However, both methodologies agree that on WT8 there is a notable alarm that effectively corresponds to the gearbox component and needs assistance on site. For the rest of false positives, each methodology has independently detected those, and thus they are not completely conclusive. Moreover, observing the results' comparison, to minimize false positives, it would be pertinent to work on

Detecting Bearing Failures in Wind Energy Parks

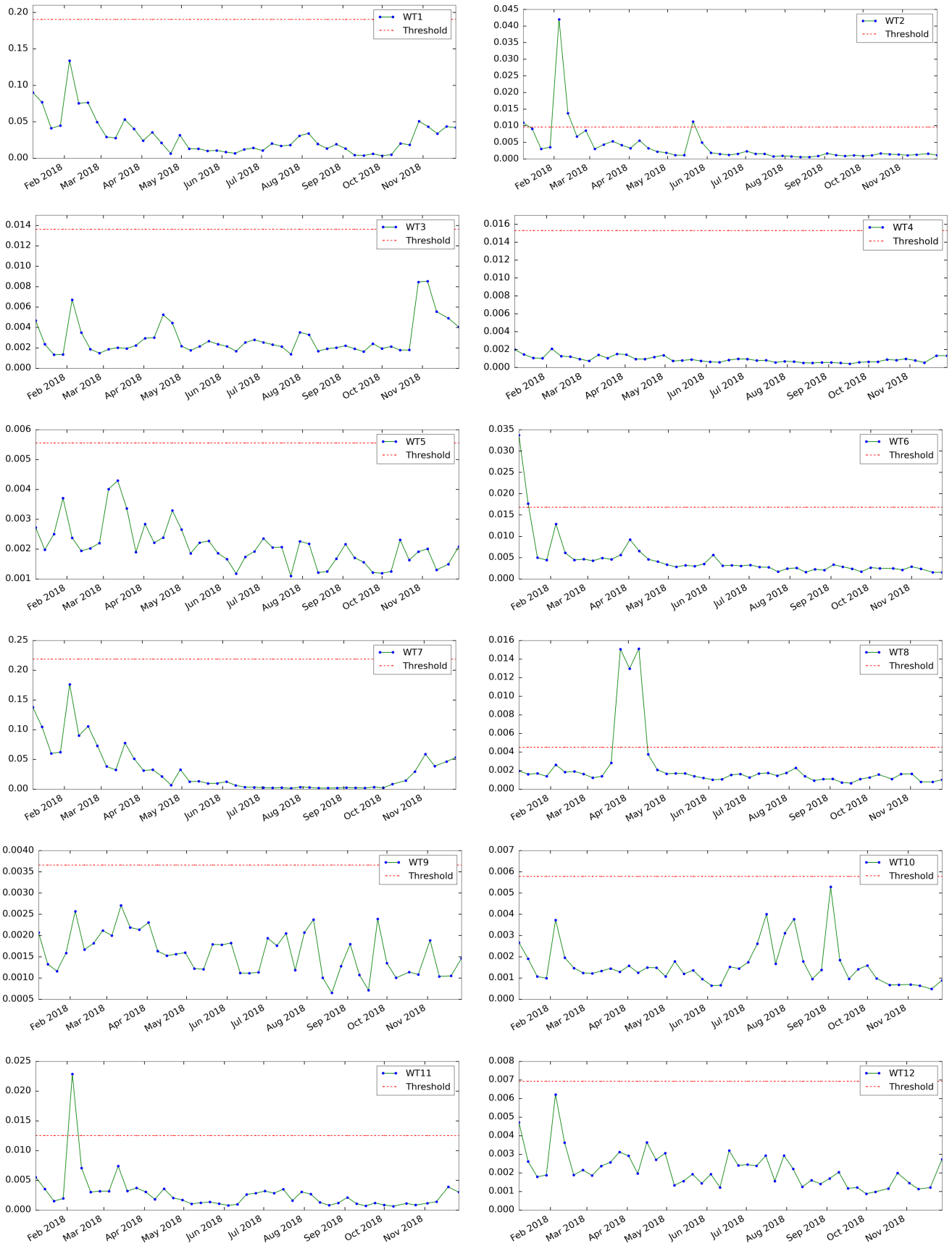


Figure 11: Results of the proposed FPI on the test dataset for the entire wind park made up of 12 WTs.

Detecting Bearing Failures in Wind Energy Parks

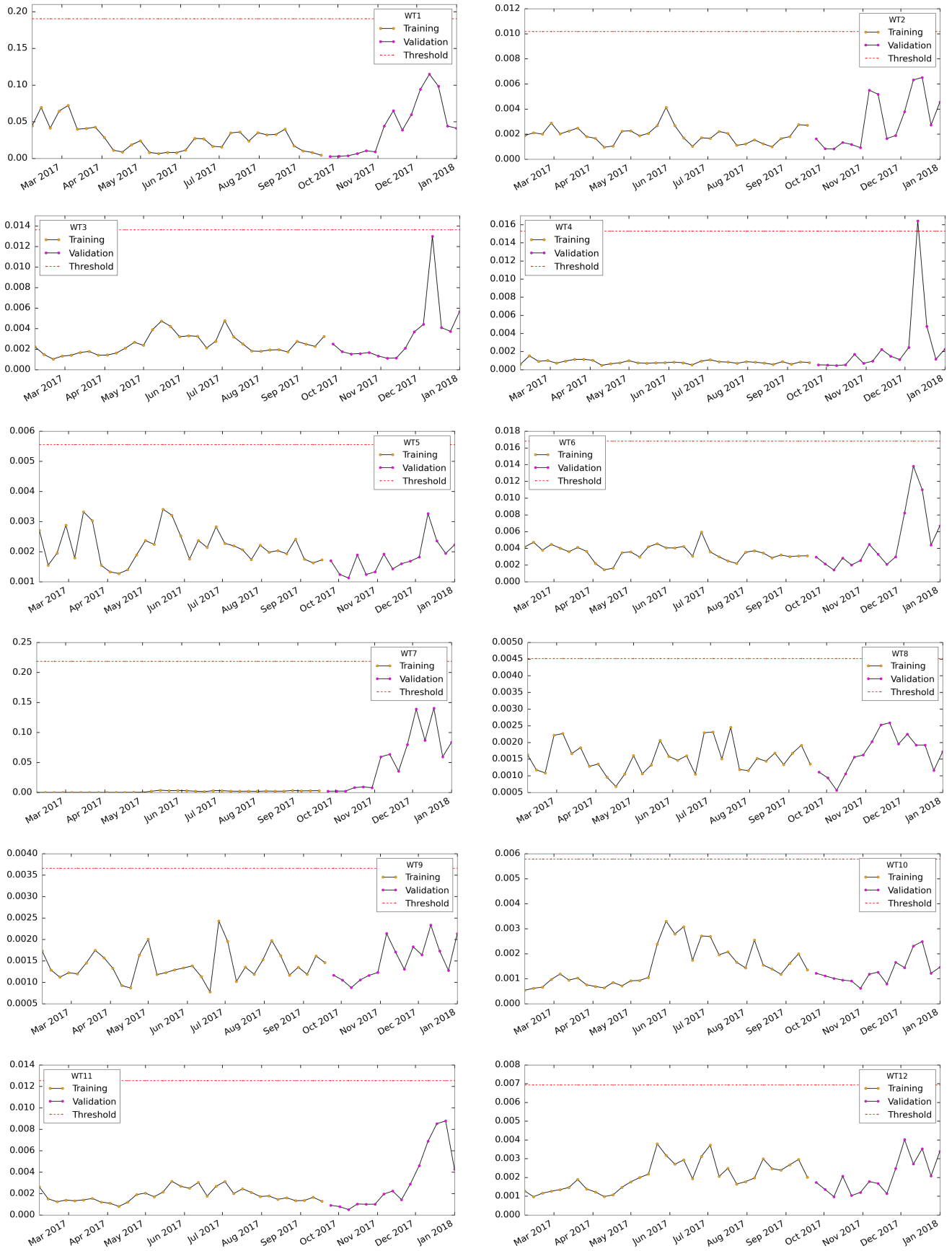


Figure 12: Results of the proposed FPI on the train and validation datasets for the entire wind park made up of 12 WTs.

Table 3

Comparison of testing results between the current methodology and the ANN-based one. Note that hits and mistakes of the prediction are marked as ticks and crosses, respectively.

WT ID	Methodology	
	CAE	ANN [8]
WT1	✓	✓
WT2	✓	✓
WT3	✓	✓
WT4	✓	✓
WT5	✓	✓
WT6	x	✓
WT7	✓	✓
WT8	x	x
WT9	✓	x
WT10	✓	✓
WT11	x	✓
WT12	✓	✓

an ensemble model which does a making-decision process before trigger or not fault alarms.

Finally, the proposed methodology could be used to detect high-speed bearing faults (by selecting SCADA temperature variables close to this bearing), which are more difficult to isolate with a good advance. As the stated approach focuses on the detection of a heat release and is not based on vibration, the bearing speed does not directly affect the methodology. This is because the methodology relies on SCADA data associated with temperature variables to detect failure. Note that the low sampling rate of SCADA data (10-minute average) hinders information in variables with a fast dynamic (e.g., vibrations); however, as temperature variables have a slow dynamic, their SCADA data still contain relevant information. Finally, as indicated previously, the vast majority of different main bearing failure modes are associated with heat release, allowing the proposed methodology to be widely applicable.

6. Conclusions

In this paper, a fault early detection strategy for WTs is addressed via a CAE. The main aim of this methodology is to give an early fault alarm to let operators plan maintenance operations in advance to minimize WT's downtime. Furthermore, the model is trained and validated only from healthy SCADA data, and the test data set contains healthy and faulty data to properly validate the methodology and the proposed FPI to detect abnormal behavior. It is noteworthy that for training and validation the data come from all possible regions of operation of the WT, and also from different year seasons (that highly affect temperature related variables). In consequence, it guarantees that the model is robust to operational and environmental variations. Finally, recall that mainly exogenous variables are used, apart from the low-speed shaft temperature, thus guaranteeing that the

strategy is focused on only detecting faults related to this last mentioned variable.

This model is tested on an entire wind park of twelve WTs. The results indicate that the detection system triggers minimal false alarms, and in some cases those alarms are related to faults in other components (e.g., gearbox). Note that after triggering the alarm, there is a distinct downward trend in residuals. Because when bearing failure begins (or worsens), there is normally a transient heat escape as temperature rises. According to [19], practically all bearing failure mechanisms are caused by unanticipated heat release. After that, the temperature goes back to normal (e.g., when the crack does not advance). Thus, the objective of the strategy is to forecast the usual heat liberation in advance, months ahead of complete breakdown. When there is no heat release, the methodology returns to small residuals. As a result, even if the residuals are returned below the threshold, the triggered alert must be maintained operational.

Unfortunately, the analyzed wind park data contains only one significant bearing failure, which is insufficient for statistical analysis. To conduct a more thorough investigation and derive findings such as a forecast time and confidence level, the model must be applied to more situations involving this issue.

The application of predictive maintenance to complex systems, such as wind turbines, remains an open challenge. In addition, the latest developments tend to use expensive specifically tailored sensors, which is not economically viable for turbines already in operation, and even less in case they are close to reaching the end of their lifespan. This is relevant, as 38 GW of wind farms in Europe are expected to reach their life expectancy in the next five years. In this context, this work has proposed a data-driven predictive maintenance methodology based only on existing supervisory control and data acquisition (10-minute averaged SCADA) data available in all industrial-sized wind turbines that is a promising and cost-effective solution for life-extension services. In addition, the stated strategy does not need faulty data or labeling data as it is based on a normal behavior model (not supervised), thus extending its range of application to truly all industrial-size wind turbines. Furthermore, the methodology has been tested on real SCADA data from a wind farm in production, ensuring a high level of technological readiness of the proposed strategy, obtaining a very low rate of false alarms and a very early prediction of the fault (months in advance), thus really allowing wind park operators to program the maintenance in weather windows and when replacements are available.

References

- [1] P. Żuk, P. Żuk, National energy security or acceleration of transition? energy policy after the war in Ukraine, *Joule* 6 (4) (2022) 709–712.
- [2] K. Hainsch, H. Brauers, T. Burandt, L. Goeke, C. R. von Hirschhausen, C. Kemfert, M. Kendzioriski, K. Löffler, P.-Y. Oei, F. Präger, et al., Make the European Green Deal real: Combining climate neutrality and economic recovery, no. 153, DIW Berlin: Politikberatung kompakt, 2020.

- [3] M. Bruck, P. Sandborn, N. Goudarzi, A leveled cost of energy (lcoe) model for wind farms that include power purchase agreements (ppas), *Renewable Energy* 122 (2018) 131–139.
- [4] P. Zhou, P. Yin, An opportunistic condition-based maintenance strategy for offshore wind farm based on predictive analytics, *Renewable and Sustainable Energy Reviews* 109 (2019) 1–9.
- [5] E. G. Nabati, K. D. Thoben, Big data analytics in the maintenance of off-shore wind turbines: A study on data characteristics, in: *Dynamics in Logistics*, Springer, 2017, pp. 131–140.
- [6] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. A. Tameghe, G. Ekemb, Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges, *Energies* 7 (4) (2014) 2595–2630.
- [7] F. Safaei, N. Tazi, E. Châtelet, Y. Bouzidi, When and how to repower energy systems? a four strategies-based decision model, *ISA transactions*.
- [8] Á. Encalada-Dávila, B. Puruncajas, C. Tutivén, Y. Vidal, Wind turbine main bearing fault prognosis based solely on scada data, *Sensors* 21 (6) (2021) 2228.
- [9] A. Lebranchu, S. Charbonnier, C. Bérenguer, F. Prevost, A combined mono-and multi-turbine approach for fault indicator synthesis and wind turbine monitoring using scada data, *ISA transactions* 87 (2019) 272–281.
- [10] H. Zhao, Y. Gao, H. Liu, L. Li, Fault diagnosis of wind turbine bearing based on stochastic subspace identification and multi-kernel support vector machine, *Journal of Modern Power Systems and Clean Energy* 7 (2) (2019) 350–356.
- [11] Y. Fu, Z. Gao, A. Zhang, X. Liu, Fault classification for wind turbine benchmark model based on hilbert-huang transformation and support vector machine strategies, in: *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, IEEE, 2021, pp. 1–8.
- [12] X. Liu, W. Teng, Y. Liu, A model-agnostic meta-baseline method for few-shot fault diagnosis of wind turbines, *Sensors* 22 (9) (2022) 3288.
- [13] S. Zare, M. Ayati, Simultaneous fault diagnosis of wind turbine using multichannel convolutional neural networks, *ISA transactions* 108 (2021) 230–239.
- [14] W. Touti, M. Salah, K. Bacha, A. Chaari, Condition monitoring of a wind turbine drivetrain based on generator stator current processing, *ISA transactions*.
- [15] J. Tautz-Weinert, S. J. Watson, Using scada data for wind turbine condition monitoring—a review, *IET Renewable Power Generation* 11 (4) (2017) 382–394.
- [16] J. L. Godwin, P. Matthews, Classification and detection of wind turbine pitch faults through scada data analysis, *IJPHM Special Issue on Wind Turbine PHM* (2013) 90.
- [17] E. Latiffianti, S. Sheng, Y. Ding, Wind turbine gearbox failure detection through cumulative sum of multivariate time series data, *Frontiers in Energy Research* 10 (NREL/JA-5000-82585).
- [18] X. Jin, Z. Xu, W. Qiao, Condition monitoring of wind turbine generators using scada data analysis, *IEEE Transactions on Sustainable Energy* 12 (1) (2020) 202–210.
- [19] SKF. Group, Bearing damage and failure analysis, *SKF Explorer*.
- [20] Encalada-Dávila, L. Moyón, C. Tutivén, B. Puruncajas, Y. Vidal, Early fault detection in the main bearing of wind turbines based on gated recurrent unit (gru) neural networks and scada data, *IEEE/ASME Transactions on Mechatronics* (2022) 1–11 doi:10.1109/TMECH.2022.3185675.
- [21] P. Tavner, C. Edwards, A. Brinkman, F. Spinato, Influence of wind speed on wind turbine reliability, *Wind Engineering* 30 (1) (2006) 55–72.
- [22] M. Kang, J. Tian, Machine learning: Data pre-processing, Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things (2018) 111–130.
- [23] P. Marti-Puig, A. Blanco-M, J. J. Cárdenas, J. Cusidó, J. Solé-Casals, Effects of the pre-processing algorithms in fault diagnosis of wind turbines, *Environmental Modelling & Software* 110 (2018) 119–128. doi:https://doi.org/10.1016/j.envsoft.2018.05.002.
- [24] S. Lu, Y. Wang, Y. Wu, Novel high-precision simulation technology for high-dynamics signal simulators based on piecewise hermite cubic interpolation, *IEEE Transactions on Aerospace and Electronic Systems* 54 (5) (2018) 2304–2317. doi:10.1109/TAES.2018.2814278.
- [25] D. Astolfi, F. Castellani, A. Lombardi, L. Terzi, Multivariate scada data analysis methods for real-world wind turbine power curve monitoring, *Energies* 14 (4) (2021) 1105.
- [26] M. Ranzato, F. J. Huang, Y.-L. Boureau, Y. LeCun, Unsupervised learning of invariant feature hierarchies with applications to object recognition, in: *2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007*, pp. 1–8. doi:10.1109/CVPR.2007.383157.
- [27] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: *Proceedings of the 19th International Conference on Neural Information Processing Systems, NIPS'06*, MIT Press, Cambridge, MA, USA, 2006, p. 153–160.
- [28] D. H. Hubel, T. N. Wiesel, Receptive fields and functional architecture of monkey striate cortex, *The Journal of physiology* 195 (1) (1968) 215–243.
- [29] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86 (11) (1998) 2278–2324. doi:10.1109/5.726791.
- [30] A. Krizhevsky, Learning multiple layers of features from tiny images, Tech. rep., University of Toronto (2009).
- [31] T. Serre, L. Wolf, T. Poggio, Object recognition with features inspired by visual cortex, Vol. 2, 2005, pp. 994 – 1000 vol. 2. doi:10.1109/CVPR.2005.254.
- [32] J. Masci, U. Meier, D. C. Ciresan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: *ICANN, 2011*.
- [33] H. B. Mitchell, Image fusion: theories, techniques and applications, Springer Science & Business Media, 2010.
- [34] C. McKinnon, A. Turnbull, S. Koukoura, J. Carroll, A. McDonald, Effect of time history on normal behaviour modelling using scada data to predict wind turbine failures, *Energies* 13 (18) (2020) 4745.