MDPI

Article

Wind Turbine Multi-Fault Detection and Classification Based on SCADA Data

Yolanda Vidal [1,*], Francesc Pozo[1] and Christian Tutivén [1]

[1] Control, Modeling, Identification and Applications (CoDAlab), Department of Mathematics, Escola d'Enginyeria de Barcelona Est (EEBE), Universitat Politècnica de Catalunya (UPC), Campus Diagonal-Besòs (CDB), Eduard Maristany, 16, Barcelona 08019, Spain.

* Correspondence: yolanda.vidal@upc.edu; Tel.: +34-934-137-309

Academic Editor: name
Version September 20, 2018 submitted to Energies

Abstract: Due to the increasing installation of wind turbines in remote locations, both onshore and offshore, advanced fault detection and classification strategies have become crucial to accomplish the required levels of reliability and availability. In this work, without using specific tailored devices for condition monitoring but only increasing the sampling frequency in the already available (in all commercial wind turbines) sensors of the Supervisory Control and Data Acquisition (SCADA) system, a data-driven multi-fault detection and classification strategy is developed. An advanced wind turbine benchmark is used. The wind turbine we consider is subject to different types of faults on actuators and sensors. The main challenges of the wind turbine fault detection lie in their nonlinearity, unknown disturbances as well as significant measurement noise at each sensor. First, the SCADA measurements are pre-processed by group scaling and feature transformation (from the original high-dimensional feature space to a new space with reduced dimensionality) based on multiway principal component analysis through sample-wise unfolding. Then, 10-fold cross validation support vector machines based classification is applied. In this work, support vector machines were used as a first choice for fault detection as they have proven their robustness for some particular faults but never accomplished, at the same time, the detection and classification of all the proposed faults taken into account in this work. To this end, the choice of the features as well as the selection of data are of primary importance. Simulation results show that all studied faults are detected and classified with an overall accuracy of 98.2%. Finally, it is noteworthy that the prediction speed allows this strategy to be deployed for online (real-time) condition monitoring in wind turbines.

Keywords: wind turbine; fault detection; fault classification; fault diagnosis; principal component analysis; support vector machines; FAST

1. Introduction

Wind energy offers many advantages, which explains why it is one of the fastest growing renewable sources against greenhouse effects. Currently, research efforts are aimed to minimize the overall cost of this energy. The tendency to use larger wind turbines (WTs) in harsh operating environments (e.g. offshore) implies that one of the main cost drivers is directly related to operation and maintenance actions. Thus, fault diagnosis (FD) is crucial for wind power to be cost competitive, and even more for offshore wind farms where bad weather conditions (storms, high tides, etc.) can prevent any repair actions for several weeks.

A variety of surveys on FD taking into account different components of a WT have been recently published. [1–4]. However, the later trend, in this type of literature review, is to focus in a specific WT

sub-assembly: the bearings and planetary gearbox [5,6], the generator and power converter [7,8], the blades [9,10], et cetera. Most of these methods (that focus in a specific part of the WT) require to choose the most appropriate sensors, their advisable position in the sub-assembly, and the most convenient strategy to extract as much information as possible from the obtained data. These are highly localized strategies and each one relies on (costly) extra sensors to be installed. However, it should be possible to retrofit a multi-fault condition monitoring package onto existing WTs without requiring additional sensors and wiring on the machines. In fact, there is a large amount of operational (SCADA) data available (already collected at the WT controller), which can be used to diagnose the turbine condition. This section addresses the state of the art in FD of WT faults using SCADA data.

In recent years, there has been efforts to develop FD strategies by analyzing only SCADA data. The use of machine learning techniques has been crucial for this area. For example, in [11] fault prediction and diagnosis for the WT generator is accomplished using real-world SCADA data from two wind power plants located in China based on principal component analysis (PCA) and unsupervised clustering methods. In [12] a FD strategy for WT gearboxes is proposed based on artificial neural networks (ANN) and tested on real-world SCADA data sets of a wind farm in southern Italy. In [13] a strategy to diagnose WT faults from SCADA data using support vector machines (SVM) is advised. Generally, the classification methods that deserve special mention are SVM [11,13–17] and ANN [12,18,19], because of their ability to handle non-linear and noisy data. On one hand, the use of ANN has drawbacks related to their training time and dependability on the optimization of fine tuning their parameters. In particular, in [18] the right number of parameters and their corresponding value must be carefully selected to create a normal behavior model based on ANN. On the other hand SVM is simpler and has successfully proven its suitability in this type of problem. Thus, SVM is the selected classifier in this paper.

Regarding FD methods based on SVM classifiers, that analyze only SCADA data, considerable research has been done. For example, in [14] different faults are studied but, unfortunately, faults in the pitch actuators could not be detected and, furthermore, the sampling period is unfeasible (0.01 s). Note that SCADA data is typically recorded at 10-minute intervals to reduce transmitted data bandwidth and storage. In [15], SVM could isolate some faults, except for high varying dynamics (this includes a pitch actuator fault), where the use of an observer, which is model-based, was found necessary and, again, the sampling period is 0.01 s. Later references based on SVM are, mainly, specifically tailored for a particular type of fault. For example, in [16] a SVM based method is proposed to classify the misalignment type of fault; in [11] generator faults are diagnosed; in [17] only actuator faults are considered; and in [13] generator and power feeder cables faults are diagnosed. In this paper, we widen the number and type of the studied faults with a unique strategy to cope with them all: three different pitch actuator faults (high air content in oil, pump wear, hydraulic leakage), a generator speed sensor fault (gain factor of 1.2), three different pitch sensor faults (stuck in 5 deg, stuck in 10 deg, and with a gain factor of 1.2), and a torque actuator offset fault.

As it has been noted previously, one of the major drawbacks in using SCADA data is the 10-minute sampling period. This low frequency resolution negatively affects the diagnosis capabilities and may hide short-lived events. On the other hand, high-resolution (but feasible) SCADA data should allow the dynamic turbine behavior to be identified with higher fidelity and thus improve detection efficiency. As stated in [20,21], in this work a research frame work is proposed that takes SCADA data with an add on high but feasible (1 s) frequency from the sensors. That is, the only requirement is to increase the frequency rate in the SCADA data from the already available sensors. Following this idea, in this work, we propose a strategy to detect and classify (through SVM) multiple WT faults using only conventional SCADA data with an add on, but feasible (sampling period of 1 s), high frequency sampling from the sensors and without the added cost of retrofitting additional sensors to the turbine.

The paper is organized as follows. In Section 2, the WT benchmark model is introduced and the proposed FD strategy is described. The obtained results are presented and discussed in Section 3. Section 4 states the conclusions and future work.

2. Fault Diagnosis Strategy

2.1. Model overview

The utmost importance of WT fault diagnosis, as mentioned in the introduction, stimulated the proposal of a model encompassing the prevailing faults accounted in practice (see [22]). This early version of the model described a generic 4.8 MW three-blade horizontal-axis variable-speed WT and it was issued by the company KK Wind Solutions [23] together with MathWorks, Inc. [24] and Aalborg University to release an international competition on fault detection and isolation in WTs. Several teams participated in the contest, and five of the solutions are compared in [25]. Then, a second enhanced model was presented in [26] that incorporated a more realistic WT simulated using the FAST software. This is an aeroelastic WT simulator designed by the U.S. National Renewable Energy Laboratory's (NREL) National Wind Technology Center and widely used by the research community. Several FAST models of WTs of varying sizes are available in the public domain, including NREL's 5 MW baseline turbine which is the one used by the model. It is noteworthy that this simulator is able to consider the WT flexible modes that are present in practice making fault detection more difficult compared to simpler models neglecting these modes (as in [22]). Thus, the second enhanced model, see [26], is the one utilized in this work.

The model proposes to simulate the sensors in the block diagram environment Simulink by adding signals from band limited white noise blocks that are parameterized by noise power, to the actual variables provided by the FAST software. These random noise blocks represent measurement noise either due to to electrical noise in the system or due to the measuring principle. The different sensors provided in the model are shown in Table 1 with the measurement noise modeled as a Gaussian white noise. Finally, a sampling period of 0.0125 s is used in the simulations.

Table 1. Available sensors (measured data).

Number	Sensor type	Symbol	Unit	Noise power
S1	Generated electrical power	$P_{e,m}$	W	10
S2	Rotor speed	$\omega_{r,m}$	rad/s	10^{-4}
S3	Generator speed	$\omega_{g,m}$	rad/s	$2 \cdot 10^{-4}$
S4	Generator torque	$\tau_{c,m}$	Nm	0.9
S5	Pitch angle of first blade	$\beta_{1,m}$	deg	$1.5 \cdot 10^{-3}$
S6	Pitch angle of second blade	$\beta_{2,m}$	deg	$1.5 \cdot 10^{-3}$
S7	Pitch angle of third blade	$\beta_{3,m}$	deg	$1.5 \cdot 10^{-3}$
S8	Tower top *fore-aft* acceleration	$a_{fa,m}$	m/s^2	$5 \cdot 10^{-4}$
S9	Tower top *side-to-side* acceleration	$a_{ss,m}$	m/s^2	$5 \cdot 10^{-4}$

The most important features of the WT are detailed in Table 2. In this paper, we deal with the full load region of operation in the sense that the proposed controller main objective is that the electric power closely follows the rated power.

A set of fault scenarios are defined at the WT model. These scenarios are primarily introduced in sensors and actuators. More precisely, the types of faults are gain factors, offsets, changes in the system dynamics and stuck, as shown in Table 3. These faults are inspired by research in both proprietary and public domain sources [26]. A comprehensive description of these faults and their importance is given in [27].

The stochastic, full-field, turbulent-wind simulator TurbSim —developed by NREL— is used to generate the wind velocity fields applied in the simulations. It employs a stochastic model —as opposed to a physics-based model— to numerically simulate time series of three-component wind-speed vectors. It provides the ability to drive simulations of complex turbine designs with realistic but simulated inflow turbulence environments that combine many of the main fluid dynamic features known to

Table 2. Gross Properties of the Wind Turbine.

Reference wind turbine	
Rated power	5 MW
Number of blades	3
Rotor/Hub diameter	126 m, 3 m
Hub Height	90 m
Cut-In, Rated, Cut-Out Wind Speed	3 m/s, 11.4 m/s, 25 m/s
Rated generator speed (ω_{ng})	1173.7 rpm
Gearbox ratio	97

Table 3. Fault scenarios.

Number	Fault	Type
F1	Pitch actuator - High air content in oil	Change in system dynamics
F2	Pitch actuator - Pump wear	Change in system dynamics
F3	Pitch actuator - Hydraulic leakage	Change in system dynamics
F4	Generator speed sensor	Gain factor (1.2)
F5	Pitch sensor	Stuck value ($\beta_{3,m} = 5$ deg)
F6	Pitch sensor	Stuck value ($\beta_{3,m} = 10$ deg)
F7	Pitch sensor	Gain factor (1.2)
F8	Torque actuator	Offset value (2000 Nm)

negatively affect turbine aeroelastic response and loading. In this work, the generated wind data has the following features: Kaimal turbulence model with intensity set to 10%, mean speed is set to 18.2 m/s and simulated at hub height, logarithmic profile wind type, and the roughness factor is set to 0.01 m. In this work, each simulation is ran with a different wind data set. More precisely, 260 different wind data sets of a duration of 600 seconds each are used.

2.2. (Non-) Noise Handling

To deal with noise in a data set, two broad ways can be considered, in general: (i) it might be filtered out; or (ii) left as it is. Obviously, pros and cons appear when adopting any one of these two approaches. By filtering out the noisy instances from the data, there is a trade-off between the amount of information available for building the classifier and the amount of noise retained in the data set. Robust algorithms do not require preprocessing of the data (the data set is taken as is, with the noisy instances), but a classifier built from a noisy data set may be less predictive and its representation may be less compact that it could have been if the data were not noisy. In this work, this second approach is used. Since multiway PCA (a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components) is used for the pre-treatment of the data, the strategy can be considered as robust. Besides, the 10-fold cross-validation (model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set) is also considered and therefore the impact of a particular noisy subset of data is minimized.

2.3. Data Collection

In this work, a total of 260 simulations are conducted. In particular 100 with a healthy WT, and 20 simulations for each studied fault. That is (recall there are 8 types of faults, see Table 3), a total of 160 simulations with a faulty WT. All simulations have a duration of 600 seconds. However, only the last 400 seconds of simulation are used to avoid the transient due to initialization of the numerical

simulations as in [28]. Measurements are taken from the nine SCADA available sensors, see Table 1. Note that the wind sequence is not used as a known measurement.

It is noteworthy that a time step of 0.0125 seconds is needed in the simulations due to the fixed-step-size time-integration scheme used by the FAST simulation software [29]. However the data used for FD is down-sampled to a sampling period of 1 second. Traditional SCADA data has a 10-minute average sampling frequency. In this paper, following [21], it is proposed to make use of conventional SCADA data with a realistic add on higher frequency sampling from the sensors (1 sample per second). Some condition monitoring systems might surpass the expense of the necessary additional equipment, however may also exhibit high rates of false positive alarms while the diagnosis is dedicated to a unique component or assembly rather than system-wide [30]. In this work, without using specific tailored devices for condition monitoring, only increasing the sampling ratio to a feasible frequency in the already available sensors of the SCADA system, a cost effective multi-fault monitoring tool is obtained.

2.4. Data Reshape and Tensor Unfolding

The main objective is, using all the available SCADA information, to minimize detection time while preserving overall accuracy. Recall that after the classification model is built, in order to diagnose a WT, a sample has to be given as an input to the model. The smaller the needed sample, the smaller will be the detection time (as less time is needed to collect the data from the sensors). Here detection time refers to the time from when the fault occurs to when it is detected. Assuming T_d is the detection time, the fault detection requirements given in the model [26] for the corresponding faults are described in terms of the sampling time for the control system, T_s, which, in this case, is equal to 1 s. In particular:

- Fault 8 (F8) is required to fulfill $T_d < 3T_s$. This is the most restrictive detection time as this is the most severe fault. It is related to the torque actuator and it is noteworthy that the torque rate limit for the NREL 5-MW WT is 15000 Nm/s [29].
- Fault 1 (F1) is required to fulfill $T_d < 8T_s$. This fault has a high varying dynamic and is related to the pitch actuator (high air content in oil). In this case, note that the blade-pitch rate limit for the NREL 5-MW WT is 8 deg/s as this is speculated to be the blade pitch rate limit of conventional 5-MW machines based on General Electric (GE) Wind's long blade test program [29].
- Faults 4 to 7 (F4, F5, F6, F7) are required to fulfill $T_d < 10T_s$. These faults are related to the generator speed sensor and the pitch sensors.
- Finally, faults 2 and 3 (F2, F3) are required only to satisfy $T_d < 100T_s$ as these are faults with a very slow dynamic. These faults are related to the pitch actuator (pump wear and hydraulic leakage).

Using the three most restrictive requirements, it is proposed to organize the available data from the simulations in three different manners:

a) In samples of only $J = 3$ time steps (this will lead to a detection time of approximately $3T_s$).
b) In samples of $J = 8$ time steps (in this case, detection time is close to $8T_s$).
c) In samples of $J = 10$ time steps (for a detection time around to $10T_s$).

The goal of the reminder of this section is to show how the data is reshaped in samples of J time steps. As said before, the data comes from 260 simulations of 400 s of duration each (with a time step of 1 s) and nine sensors available. This data is initially stored, for each sensor, in a matrix as follows:

$$\begin{pmatrix} x^{(k)}_{1,1} & x^{(k)}_{1,2} & \cdots & x^{(k)}_{1,\left\lfloor \frac{400}{J} \right\rfloor J} \\ x^{(k)}_{2,1} & x^{(k)}_{2,2} & \cdots & x^{(k)}_{2,\left\lfloor \frac{400}{J} \right\rfloor J} \\ \vdots & \vdots & \ddots & \vdots \\ x^{(k)}_{i,1} & x^{(k)}_{i,2} & \cdots & x^{(k)}_{i,\left\lfloor \frac{400}{J} \right\rfloor J} \\ \vdots & \vdots & \ddots & \vdots \\ x^{(k)}_{260,1} & x^{(k)}_{260,2} & \cdots & x^{(k)}_{260,\left\lfloor \frac{400}{J} \right\rfloor J} \end{pmatrix} \in \mathcal{M}_{260\times \left\lfloor \frac{400}{J} \right\rfloor J}(\mathbb{R}) \tag{1}$$

where the super-index (k) is related to the different sensors $k = 1, 2, \ldots, 9$. That is, there is one of these matrices for each sensor. The matrix has as many rows as simulations (260). The number of columns is taken as $\left\lfloor \frac{400}{J} \right\rfloor J$, where $\lfloor \cdot \rfloor$ is the floor function, to ensure that the matrix can be afterwards reshaped in a matrix of J columns. When $J = 8$ or $J = 10$ this results in using the whole 400 s of each simulation ($\left\lfloor \frac{400}{J} \right\rfloor J = 400$ in these two cases, as 8 and 10 are divisors of 400). However, when $J = 3$ it is obtained that $\left\lfloor \frac{400}{J} \right\rfloor J = 399$, thus in this case only 399 s of each simulation are used.

As said before, when a WT has to be diagnosed it is desirable that with a few seconds of measured data a diagnose can be obtained. Thus, instead of working with the matrices in Equation (1) (where each sample would correspond to 399 or 400 s of data), data is reshaped in a matrix with only J columns (as stated before, in this work $J = 3$, 8, or 10) as follows:

$$\begin{pmatrix} x^{(k)}_{1,1} & x^{(k)}_{1,2} & \cdots & x^{(k)}_{1,J} \\ x^{(k)}_{1,J+1} & x^{(k)}_{1,J+2} & \cdots & x^{(k)}_{1,2J} \\ \vdots & \vdots & \ddots & \vdots \\ x^{(k)}_{1,400-J+1} & x^{(k)}_{1,400-J+2} & \cdots & x^{(k)}_{1,\left\lfloor \frac{400}{J} \right\rfloor J} \\ x^{(k)}_{2,1} & x^{(k)}_{2,2} & \cdots & x^{(k)}_{2,J} \\ x^{(k)}_{2,J+1} & x^{(k)}_{2,J+2} & \cdots & x^{(k)}_{2,2J} \\ \vdots & \vdots & \ddots & \vdots \\ x^{(k)}_{2,400-J+1} & x^{(k)}_{2,400-J+2} & \cdots & x^{(k)}_{2,\left\lfloor \frac{400}{J} \right\rfloor J} \\ \vdots & \vdots & \ddots & \vdots \\ x^{(k)}_{260,1} & x^{(k)}_{260,2} & \cdots & x^{(k)}_{260,J} \\ x^{(k)}_{260,J+1} & x^{(k)}_{260,J+2} & \cdots & x^{(k)}_{260,2J} \\ \vdots & \vdots & \ddots & \vdots \\ x^{(k)}_{260,400-J+1} & x^{(k)}_{260,400-J+2} & \cdots & x^{(k)}_{260,\left\lfloor \frac{400}{J} \right\rfloor J} \end{pmatrix} \in \mathcal{M}_{260\left\lfloor \frac{400}{J} \right\rfloor \times J}(\mathbb{R}) \tag{2}$$

where J defines the number of seconds of each sample, and recall that the super-index (k) is related to the different sensors $k = 1, 2, \ldots, 9$. Note that the total number of samples is given by $I = 260 \cdot \left\lfloor \frac{400}{J} \right\rfloor$, that is

a) $I = 34580$ samples when $J = 3$.
b) $I = 13000$ samples when $J = 8$.

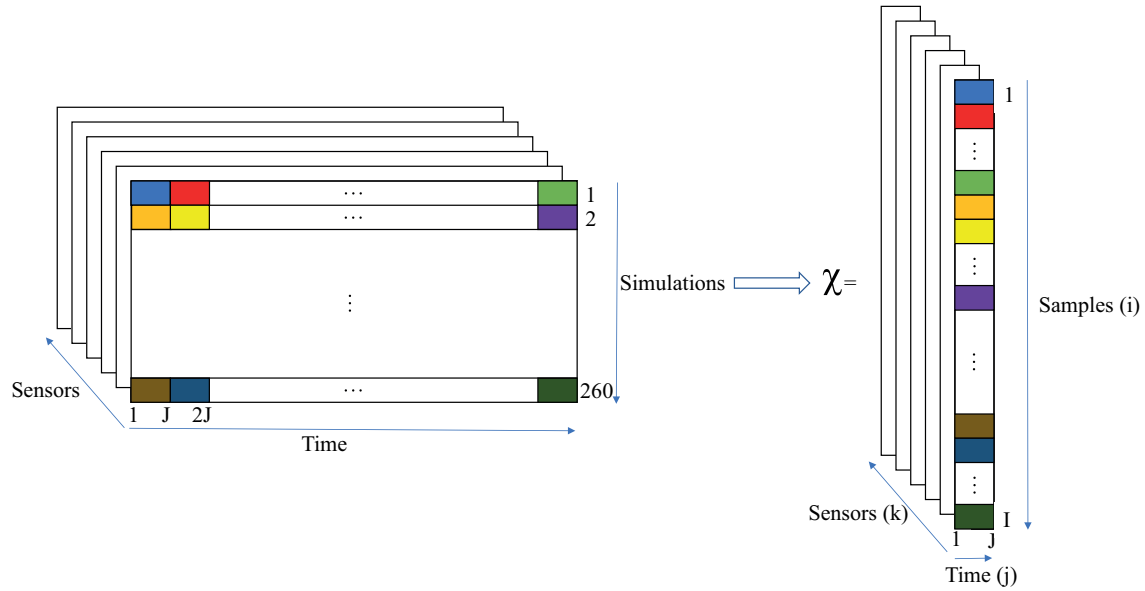

Figure 1. Reshape data from long run simulations (left) into a third-order tensor $\mathcal{X}$ $(I \times J \times K)$ with short time samples of J seconds (right).

c) $I = 10400$ samples when $J = 10$.

Figure 1 illustrates how the available data from the 260 long run simulations, see Equation (1), is reorganized in a third-order tensor (multidimensional array with three indices) with short time samples of J time steps, see Equation (2). The first J data-points determine the first sample (represented by the light blue color box in Figure. 1). Right after, the next J data-points determine the second sample (red color box), et cetera. After the last J data-points of the first simulation (light green), the first J data-points of the second simulation (orange box) define the next sample, and so on. In general, let us consider we have at $j = 1, 2, \ldots, J$ time instants, different sensors $k = 1, 2, \ldots, K$ stored. Similar data is generated for a number of samples $i = 1, 2, \ldots, I$. This results in the third-order tensor $\mathcal{X}$ $(I \times J \times K)$ as illustrated in Figure 1, where the height (I) gives the number of samples; the width (J) gives the number of time instants; and the length (K) gives the number of sensors.

The crux of the matter for fault detection by SVM is the definition of the features to be used for classification [14]. In this work, statistical analysis by multiway PCA is used for pretreatment of the raw data. This is equivalent to implementing basic PCA on a large two-dimensional matrix assembled by unfolding the third-order tensor $\mathcal{X}$, see Figure 1. There are three possible ways of unfolding this tensor as suggested by [31]. In general, sample-wise unfolding facilitates the analysis of the variability among samples by summarizing the information related to the measured variables (sensors) and their variations over time. Thus, in this work, the sample-wise unfolding is used, see Figure 2, where

$$\mathcal{X}(I \times J \times K) \longrightarrow X(I \times JK). \tag{3}$$

That is, the $I \times J$ planes are concatenated into a large two-dimensional matrix X. In summary, multiway PCA of the third-order tensor $\mathcal{X}$ in Figure 1 is implemented considering PCA of the sample-wise unfolded matrix X in Equation (2).

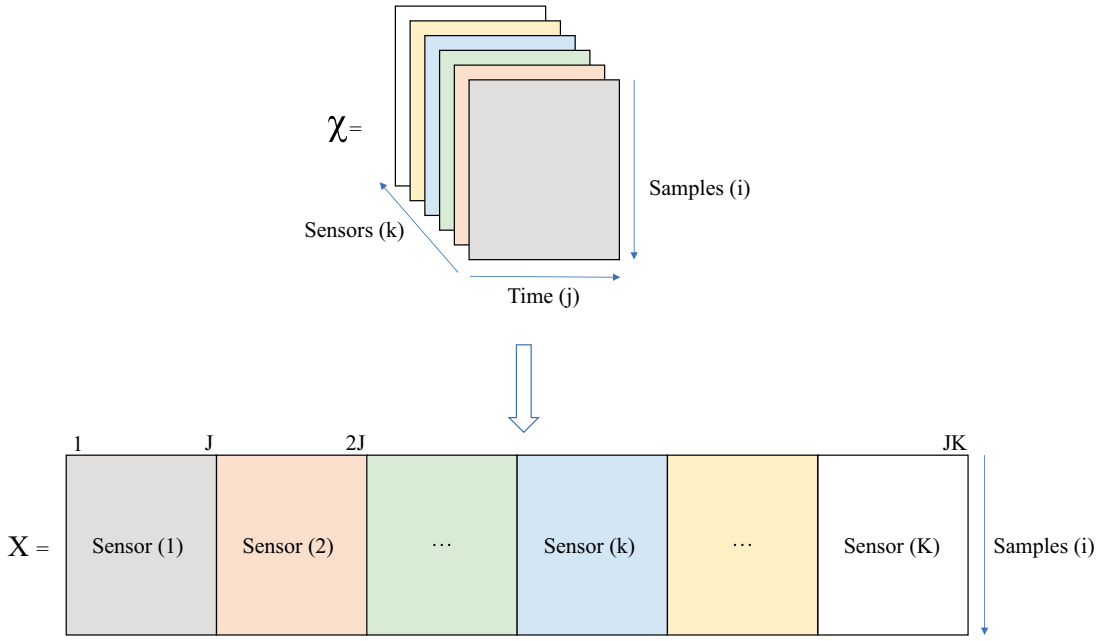

Figure 2. Unfolding of the third-order tensor $\mathcal{X}$ into matrix X.

2.5. Autoscaling or Standardization

The reason to autoscale the raw data is two-fold: to deal with data that comes from different sensors and with different magnitudes, and to simplify the computations for the multiway PCA decomposition.

Autoscaling is a relatively frequent pre-processing method that uses column-wise mean-centering followed by division of each column by the standard deviation of that column of matrix X. The result is that each column of the new autoscaled matrix, $\tilde{X}$, has a mean of zero and a standard deviation of one. This idea is used in this work to rectify for different sensor measurements, magnitudes and units where the prevalent source of variance is due to the signal itself rather than noise. In particular, it is computed

$$\mu_j = \frac{1}{I}\sum_{i=1}^{I} x_{ij},\ j = 1,\ldots,JK, \tag{4}$$

$$\sigma_j = \sqrt{\frac{1}{I}\sum_{i=1}^{I}(x_{ij}-\mu_j)^2},\ j = 1,\ldots,JK, \tag{5}$$

where μ_j and σ_j are the mean and the standard deviation, respectively, of all the measures at column j. Accordingly, the elements of matrix X are normalized to create a new matrix $\tilde{X}$ as

$$\tilde{x}_{ij} := \frac{x_{ij}-\mu_j}{\sigma_j},\ i = 1,\ldots,I,\ j = 1,\ldots,JK. \tag{6}$$

2.6. Multiway PCA

Recall that, before using a classifier, the raw data coming from the sensors must be processed to obtain the most suitable features. In this work, after the autoscaling step, multiway PCA is selected as the main objective is to keep as much information as possible with the minimum amount of data.

Since the input data is given in a mean-centered matrix $\tilde{X}$, the empirical covariance matrix, S, can be computed as

$$S = \frac{1}{I-1}\tilde{X}^T\tilde{X} \in \mathcal{M}_{(JK)\times(JK)}(\mathbb{R}). \tag{7}$$

Then the singular value decomposition of S is computed,

$$S = PDP^T \tag{8}$$

where D is a matrix in diagonal form composed by the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{JK}$ in decreasing order, and $P \in \mathcal{M}_{(JK)\times(JK)}(\mathbb{R})$ is an orthogonal matrix that contains the eigenvectors. Matrix P is usually called the loading matrix. As the main objective is to reduce the overall size of the data set, only a reduced number of $d < JK$ principal components are used. In this work, the number of principal components is selected based on keeping 99.98% of the variance. The proportion of the variance directed along (explained by) the first d components is given by:

$$\frac{\lambda_1 + \cdots + \lambda_d}{\lambda_1 + \cdots + \lambda_{JK}}. \tag{9}$$

In the first case, when $J = 3$, from a total of $J \times K = 3 \times 9 = 27$ components, 99.98% of the variance is accomplished by the first $d = 16$ components. When $J = 8$, from a total of $J \times K = 8 \times 9 = 72$ components, the first $d = 42$ components are needed to keep 99.98% of the variance. Finally, when $J = 10$, from a total of $J \times K = 10 \times 9 = 90$ components, the demanded variance is accomplished by the first $d = 52$ components. Thus, the matrix $P_d \in \mathcal{M}_{(JK)\times(d)}(\mathbb{R})$, with only the first d columns of P is used. Finally, the score matrix $Y \in \mathcal{M}_{(I)\times(d)}(\mathbb{R})$ (transformed coordinates of the $\tilde{X}$ data in the new basis given by the first d principal components), whose columns will be used as features by the SVM strategy, is computed

$$Y = \tilde{X}P_d. \tag{10}$$

2.7. Support Vector Machines

Since their introduction by Vladimir Vapnik [32], SVM have been successfully applied to a number of real world problems such as face detection, object detection, and handwritten digit and character recognition in machine vision. SVM exhibit a remarkable resistance to overfitting and their training is performed by maximizing a convex functional which means that there is a unique solution that can always be found in polynomial time [33]. In this section, basic hints about SVM classification are given.

SVM classification is fundamentally a binary classification technique. Let us consider a training set $\{(x_i, y_i)\}_{i=1}^N$ with d-dimensional data $x_i \in \mathbb{R}^d$ and their corresponding label $y_i \in \{-1, +1\}$. Figure 3 shows these data where one class is labeled as (+) and the other one as (-). The main goal is to find the optimal hyperplane that defines the widest margin to separate both classes, see Figure 3. Formally, the hyperplane is given by

$$h(x) = \omega^T x + b, \tag{11}$$

where b is known as the bias term and ω is the weight vector. The optimal hyperplane can be characterized in an infinite number of different ways by scaling of b and ω. As a matter of agreement, among all the possible descriptions of the hyperplane, it is chosen the so-called *canonical hyperplane* that satisfies

$$\omega^T x_+^{sv} + b = 1, \tag{12}$$
$$\omega^T x_-^{sv} + b = -1, \tag{13}$$

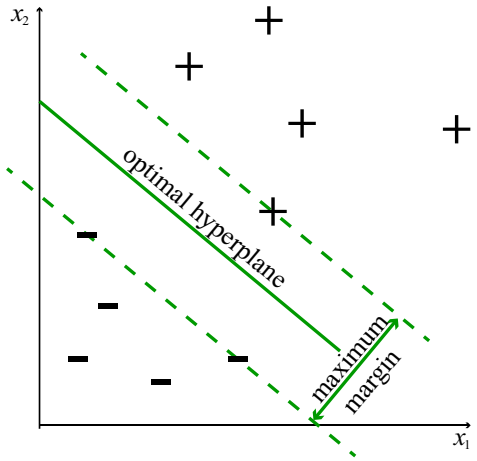

Figure 3. Linear SVM in a two-dimensional example.

where x_+^{sv} and x_-^{sv} symbolize the $(+)$ and $(-)$ training samples closest to the hyperplane, that is the so called support vectors, see Figure 3. The distance between a point x and the hyperplane h is given by

$$\mathrm{d}(x,h) = \frac{|\omega^T x + b|}{||\omega||}. \tag{14}$$

In particular, for the canonical hyperplane, when x is a support vector, the numerator $|\omega^T x + b|$ is equal to one and the distance to the support vector is,

$$\mathrm{d}(x_\pm^{sv},h) = \frac{1}{||\omega||}. \tag{15}$$

The width of the margin is twice this distance, that is $\frac{2}{||\omega||}$. Thus, to maximize the margin is equivalent to minimize the expression $\frac{||\omega||}{2}$, which is equivalent to the following minimization problem

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 \quad \text{subject to} \quad \begin{cases} h(x_i) \geq 1, \ \forall y_i = 1 \text{ samples;} \\ h(x_i) \leq -1, \ \forall y_i = -1 \text{ samples.} \end{cases} \tag{16}$$

The two previous restrictions can be rewritten in one single equation by taking the product $h(x)y$,

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 \quad \text{subject to} \quad h(x_i)y_i \geq 1, \ i = 1,\ldots,N. \tag{17}$$

This problem, to find the extrema of a function with constraints, can be solved using Lagrange multipliers, thus leading to

$$\min_{\omega,b} L(\omega,b) = \frac{1}{2}||\omega||^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(\omega^T x_i + b) - 1 \right], \tag{18}$$

where α_i are the Lagrange multipliers. Taking partial derivative with respect to ω equal to zero,

$$\frac{\partial L(\omega,b)}{\partial \omega} = \omega - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \Leftrightarrow \omega = \sum_{i=1}^{N} \alpha_i y_i x_i. \tag{19}$$

This equation states that the decision vector, ω, is a linear combination of the data samples. Taking partial derivative with respect to b equal to zero,

$$\frac{\partial L(\omega, b)}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0. \tag{20}$$

Finally, substitution of Equations (19) - (20) into Equation (18) leads to

$$\min_{\alpha_i} \left[\frac{1}{2} \left(\sum_{i=1}^{N} \alpha_i y_i x_i \right)^T \left(\sum_{j=1}^{N} \alpha_j y_j x_j \right) - \sum_{i=1}^{N} \alpha_i y_i \left(\sum_{j=1}^{N} \alpha_j y_j x_j \right)^T x_i - b \underbrace{\sum_{i=1}^{N} \alpha_i y_i}_{=0} + \sum_{i=1}^{N} \alpha_i \right], \tag{21}$$

that can be rewritten as

$$\min_{\alpha_i} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \right]. \tag{22}$$

If the data does not admit a separating hyperplane, SVM can use a soft margin, meaning a hyperplane that separates many, although not all data points. Consequently, the previous problem is generalized by means of slack variables, ε_i, and a penalty parameter, C. The general formulation for the linear kernel is in this case:

$$\min_{\omega, b, \varepsilon_i} \frac{1}{2} ||\omega||^2 + C \sum_{i=1}^{N} \varepsilon_i \quad \text{subject to} \quad \begin{cases} h(x_i) y_i \geq 1 - \varepsilon_i, \ i = 1, \ldots, N; \\ \varepsilon_i \geq 0, \ i = 1, \ldots, N. \end{cases} \tag{23}$$

In this case, using Lagrange multipliers, the problem reads

$$\min_{\alpha_i} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \right] \quad \text{subject to} \quad \begin{cases} \sum_{i=1}^{N} \alpha_i y_i = 0; \\ 0 \leq \alpha_i \leq C, \ i = 1, \ldots, N. \end{cases} \tag{24}$$

The final set of restrictions shows why the penalty parameter C is frequently called a box constraint, as it keeps the admissible values of the Lagrange multipliers in a bounded region. In this work, the box constraint value has been tuned to optimize the performance of the SVM, as is shown in Section 4.

From Equations (22) and (24) is obvious that optimization depends only on dot products of pairs of samples. Also, the decision rule depends only on the dot product. Furthermore, the optimization problem is solved in a convex space (in contrast to neural networks), thus it never obtains a local extrema but the global one. When the space is not linearly separable (the classification problem does not have a simple hyperplane as a useful separating criterion even using a soft margin), a transformation to another space can be used, $\phi(\cdot)$. In fact, the transformation itself is not needed, but just the dot product, the so called kernel function,

$$K(x_i, x_j) = \phi(x_i)\phi(x_j). \tag{25}$$

The kernel function permits the computation of the inner product between the mapped vectors without expressly calculating the mapping. This is advantageous as it implies that if data is transformed into a higher dimensional space (which helps to better classification) there is no need to compute the exact transformation of the data, but only the inner product of the data in that higher dimensional space (which is computationally cheaper). This fact is known as the kernel trick [34]. Different kernels can be used, namely polynomial, hyperbolic tangent or Gaussian radial basis function. On one hand, the feature space mapping of the Gaussian kernel has infinite dimensionality. On the other hand, the

Gaussian kernel has a ready interpretation as a similarity measure as its value decreases with distance and ranges between zero and one. For these reasons, in this work the Gaussian kernel is used, namely,

$$K(x_i, x_j) = e^{-\gamma(||x_i - x_j||^2)}, \tag{26}$$

where γ is a free parameter, hereafter denoted as kernel scale, related to the Gaussian kernel width. In this work, the kernel scale is computed as the inverse of the square root of the number of features. It is noteworthy that, in this work, the same features and the same kernel scale value for the Gaussian kernel are used to detect all faults. In other words, a unique trained SVM is able to classify among all the studied classes (eight faulty classes and one healthy class). That is not the case in the previous literature related to WT fault detection (e.g., [14,35]) where the features and the variance were adjusted case by case to detect each different fault, thus leading to a much more complex strategy that needed as many different SVM classifiers as faults to detect. Regarding computational effort, there is a clear advantage related to the feature computation, as only one set of features is needed in our proposed approach.

As it has been mentioned earlier, SVM classification is essentially a binary (two-class) classification technique, which has to be modified to deal with the multi-fault classification. Two of the most common methods to enable this adaptation include the one-vs-one and one-vs-all approaches. The one-vs-all technique represents the earliest and most common SVM multiclass approach [36] and comprises the division of an N class dataset into N two-class cases and it chooses the class which classifies the test with greatest margin. The one-vs-one strategy comprises constructing a machine for each pair of classes, thus resulting in $N(N-1)/2$ machines. When this approach is applied to a test point, each classification gives one vote to the winning class and the point is labeled with the class having most votes. The one-vs-one strategy is more computationally demanding since the results of more SVM pairs ought to be computed. In this work, the one-vs-all approach is used.

2.8. k-Fold Cross Validation

Normally, a data-based classifier is inferred based on training data and considering a classifier learning algorithm. A prediction error —also known as true error— is associated to each classifier. However, this prediction error is usually unknown, cannot be computed and must be estimated based on data. Different estimators of the prediction error can be considered, from the simple hold-out [37] and resubstitution [38] to the more sophisticated bootstrap [39]. One of these techniques, and possibly the most popular, is k-fold cross validation [40]. In k-fold cross validation, the data set is distributed into k folds, the classifier is then learned using $k-1$ folds, and the prediction error is computed by testing the classifier in the fold that is not used in the learning step. In the end, the estimation of the error is the numerical mean of the errors committed in each fold. In this paper, 10-fold cross validation is used to estimate the performance of the proposed FD strategy.

3. Results, Analysis and Discussion

The results of the proposed multi-fault diagnosis strategy introduced in Section 2 in the dataset under study are presented in this section.

First, a flowchart of the proposed approach and how it is applied is given in Figure 4. When a WT has to be diagnosed, data coming from the WT sensors is scaled and then, using the already computed PCA projection, the features are computed. Then the already trained SVM classifies the data.

Note that the box constraint value is tuned to optimize the SVM performance. Making this value large increases the weight of misclassification, see Equation (23), which leads to a stricter separation. However, increasing its value leads to longer training times. The value $C = 50$ is used in this work because, as shown in Figure (5), with smaller values the overall accuracy is degraded and with larger values similar results are obtained (with longer training times).

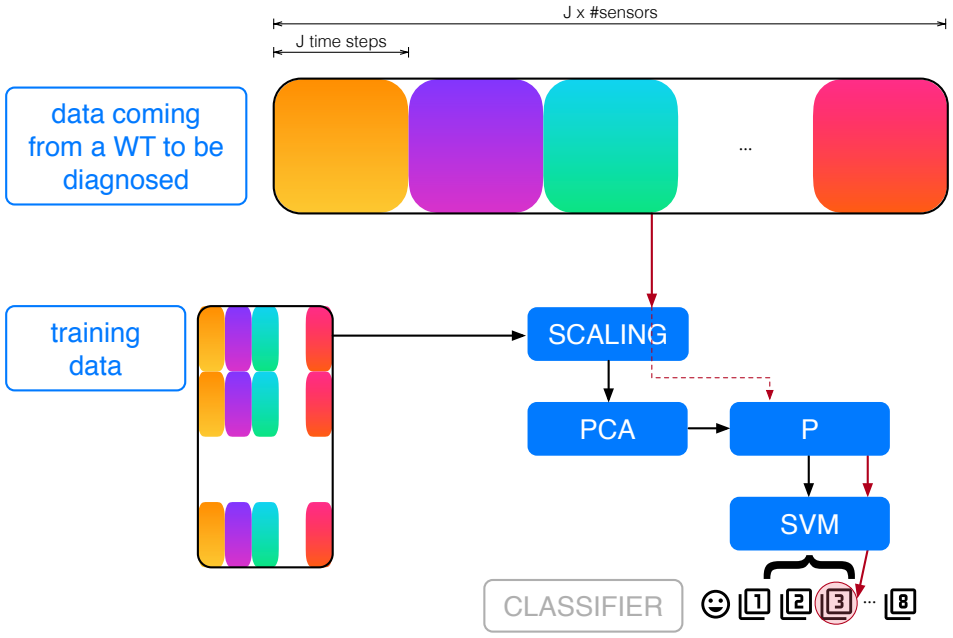

Figure 4. Data coming from a WT to be diagnosed is first scaled, then projected into the vectorial space spanned by the first principal components and finally the projection enters the classifier.

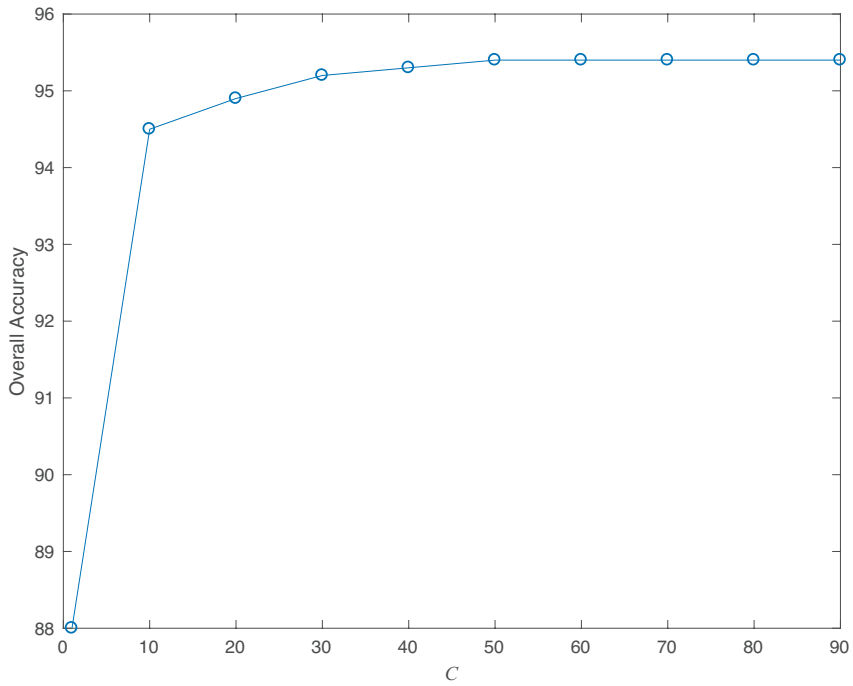

Figure 5. Box constraint value with respect to overall accuracy.

Table 4 summarizes the results obtained from the proposed strategy. It presents not only the overall accuracy, but also the training time and prediction speed, as both parameters are critical in real application. Notice that prediction speed, in all cases, allows this strategy to be deployed for online (real-time) condition monitoring in WTs. Besides, a comprehensive decomposition of the error between the true classes and the predicted classes is shown by means of the so called confusion matrices, see

Figures 6, 7, and 8 (note that an empty blank square means 0%). In these matrices, each row represents the instances in a true class while each column represents the instances in a predicted class (by the classifier). In particular, first row (and first column) is labeled as 0 and corresponds to the healthy case. Next labels (for rows and columns) correspond to each fault (from Fault 1 to Fault 8). From the confusion matrices and Table 4 the following issues can be highlighted.

When detection time is approximately 3 seconds ($J = 3$), the overall accuracy is 95.5%. In this case, the healthy class has a true positive rate (TPR), that is percentage of correctly classified instances, higher than 99% and a false negative rate (FNR), that is percentage of incorrectly classified instances, smaller than 1%. Fault 1 (the most difficult to classify in previous references and related to the pitch actuator fault with high dynamics) has a TPR of 77% and a FNR of 23%. This FNR percentage is mainly obtained from 17% missing faults and 6% confusion with Fault 2, which is also a fault located in the pitch actuator. Fault 6, related to a stuck value (10 deg) of the pitch sensor measurement, is 5% of the times misclassified as healthy, 3% of the times confused with the same type of fault but with only a 5 deg stuck value (Fault 5), and 2% misclassified as Fault 2 (pitch actuator fault). The other faults have a TPR higher than 92%. It is noteworthy that Fault 8, the most severe one and related to the torque actuator, has a 100% TPR with this most restrictive detection time.

When detection time is approximately 8 seconds ($J = 8$), the overall accuracy is 98%. As in the previous case, the healthy class has a TPR higher than 99%. Fault 1 increases its TPR to 79% (where 16% are missed faults and 5% confusion with Fault 2), and all the other classes increase their TPR to values higher than 98%. It is noteworthy that Fault 4, related to the generator speed sensor, reaches a 100% TPR. The generator speed measurement from the sensor is used as input in the torque and pitch controllers, thus being able to correctly diagnose this type of fault is extremely important. As in the previous case, Fault 8 keeps a 100% TPR.

Finally, when $J = 10$ the overall accuracy is 98.2%. In this case, Fault 1 is improved to have a TPR of 80%. In this case, all misclassifications are 1% or lower, except for Fault 1 that is 15% of the times misclassified as healthy and 5% of the times misclassified as Fault 2 (recall, also a pitch actuator fault). Note that Fault 1, 4, and 8 obtain a remarkable 100% TPR.

Table 4. Summary of the obtained results.

	$J = 3$	$J = 8$	$J = 10$
Accuracy (%)	95.5	98	98.2
Training time (s)	2990	202	181
Prediction speed (obs/s)	3000	3500	3600

Access to real SCADA datasets if often proprietary. And, therefore, it is not accessible by the scientific community. To overcome this difficulty, in this work simulated data is obtained by one of the most widely accepted WT simulators in the scientific community (FAST). The drawbacks of using simulated data is that there is no possibility to evaluate the proposed method in a full test set representing the true distribution of real-world data where class imbalance is a challenging problem [41]. However, there are several references , e.g. [13], where this problem is solved in the training stage using under/oversampling the training data.

4. Conclusion

Because of its standard low sampling rate, there is a lack of knowledge on the potential of SCADA data for condition monitoring. In this work, a promising strategy to detect and classify multiple WT faults has been presented using only conventional SCADA data with an add on, but feasible, high frequency sampling from the sensors (one sample per second). That is, the FD strategy does not involve supplementary installation of costly purpose-built data sensing equipment for wind power plants.

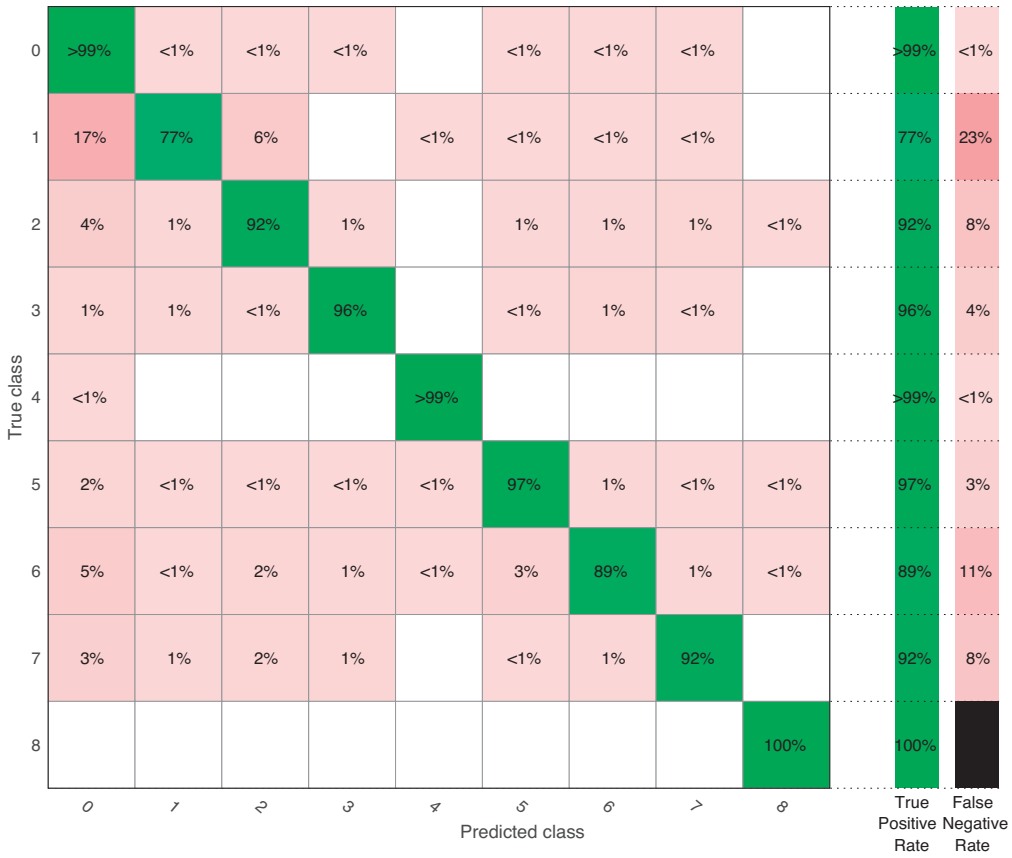

Figure 6. Confusion matrix when $J = 3$.

It is noteworthy that in this work, in contrast to the previous literature, the same features and the same variance for the Gaussian kernel are used to detect all the faults detailed in the benchmark. Thus, leading to a unique trained classifier capable to cope with all the studied faults by computing only one set of features from the data to diagnose.

As future work, other faults will be included involving misalignment, ice accumulation, and tower damage. Finally, it will be studied the contribution of an effective predictive maintenance strategy based on this same principle in order to further optimize operation and maintenance in WTs.

Author Contributions: All authors contributed equally to this work.

Funding: This work has been partially funded by the Spanish Ministerio de Economía, Industria y Competitividad (MINECO) through the research project DPI2014-58427-C2-1-R; 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 DPI2017-82930-C2-1-R; and by the Generalitat de Catalunya through the research project 2017 SGR 388. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Hossain, Md Liton and Abu-Siada, Ahmed and Muyeen, S. M.. Methods for Advanced Wind Turbine Condition Monitoring and Early Diagnosis: A Literature Review. *Energies* **2018**, *11*.
2. Ahadi, Amir. Wind turbine fault diagnosis techniques and related algorithms. *International Journal of Renewable Energy Research (IJRER)* **2016**, *6*, 80–89.

Figure 7. Confusion matrix when $J = 8$.

3. W.Y. Liu and B.P. Tang and J.G. Han and X.N. Lu and N.N. Hu and Z.Z. He. The structure healthy condition monitoring and fault diagnosis methods in wind turbines: A review. *Renewable and Sustainable Energy Reviews* **2015**, *44*, 466 – 472. doi:https://doi.org/10.1016/j.rser.2014.12.005.
4. Gao, Z.; Cecati, C.; Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques?Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics* **2015**, *62*, 3757–3767.
5. de Azevedo, Henrique Dias Machado and Araújo, Alex Maurício and Bouchonneau, Nadège. A review of wind turbine bearing condition monitoring: State of the art and challenges. *Renewable and Sustainable Energy Reviews* **2016**, *56*, 368–379.
6. Kandukuri, Surya Teja and Klausen, Andreas and Karimi, Hamid Reza and Robbersmyr, Kjell Gunnar. A review of diagnostics and prognostics of low-speed machinery towards wind turbine farm-level health management. *Renewable and Sustainable Energy Reviews* **2016**, *53*, 697–708.
7. Huang, Shoudao and Wu, Xuan and Liu, Xiao and Gao, Jian and He, Yunze. Overview of condition monitoring and operation control of electric power conversion systems in direct-drive wind turbines under faults. *Frontiers of Mechanical Engineering* **2017**, *12*, 281–302.
8. Yang, Zhimin and Chai, Yi. A survey of fault diagnosis for onshore grid-connected converter in wind energy conversion systems. *Renewable and Sustainable Energy Reviews* **2016**, *66*, 345–359.
9. Ochieng, Francis Xavier and Hancock, Craig Matthew and Roberts, Gethin Wyn and Le Kernec, Julien. A review of ground-based radar as a noncontact sensor for structural health monitoring of in-field wind turbines blades. *Wind Energy* **2018**.
10. Shohag, Md Abu S and Hammel, Emily C and Olawale, David O and Okoli, Okenwa I. Damage mitigation techniques in wind turbine blades: A review. *Wind Engineering* **2017**, *41*, 185–210.
11. Zhao, Y.; Li, D.; Dong, A.; Kang, D.; Lv, Q.; Shang, L. Fault Prediction and Diagnosis of Wind Turbine Generators Using SCADA Data. *Energies* **2017**, *10*, 1210.

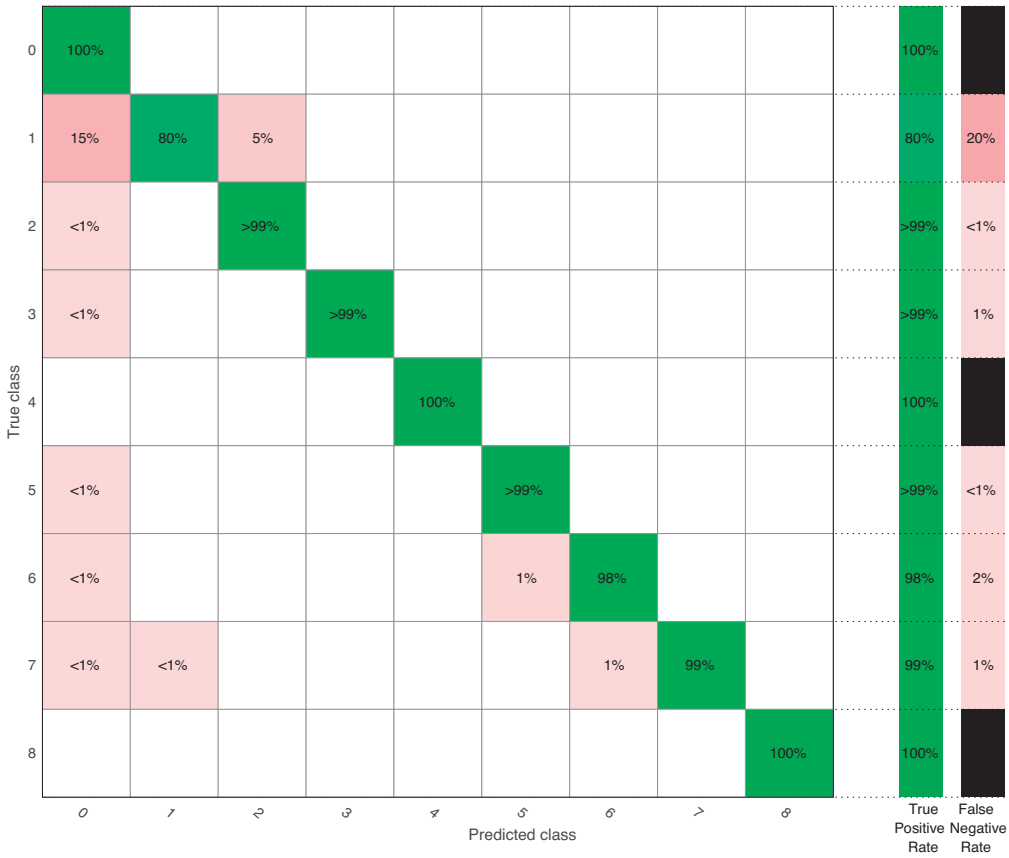

Figure 8. Confusion matrix when $J = 10$.

12. Astolfi, Davide and Castellani, Francesco and Scappaticci, Lorenzo and Terzi, Ludovico. Diagnosis of wind turbine misalignment through SCADA data. *Diagnostyka* **2017**, *18*.
13. Leahy, K.; Hu, R. L.; Konstantakopoulos, I. C.; Spanos, C. J.; Agogino, A. M.; O'Sullivan, D. T. J. Diagnosing and predicting wind turbine faults from SCADA data using support vector machines. *International Journal of Prognostics and Health Management* **2018**, *9*, 1–11.
14. Laouti, N.; Sheibat, N.; Othman, S. Support vector machines for fault detection in wind turbines. Proceedings of IFAC World Congress, 2011, Vol. 2, pp. 7067–707.
15. Laouti, Nassim and Othman, Sami and Alamir, Mazen and Sheibat-Othman, Nida. Combination of model-based observer and support vector machines for fault detection of wind turbines. *International Journal of Automation and computing* **2014**, *11*, 274–287.
16. Xiao, Y.; Hong, Y.; Chen, X.; Chen, W. The application of dual-tree complex wavelet transform (DTCWT) energy entropy in misalignment fault diagnosis of doubly-fed wind turbine (DFWT). *Entropy* **2017**, *19*, 587.
17. Abdelkrim, S.; Djamel, M.M.; Samia, A.; Hayet, M.; Mawloud, T. The MAED and SVM for fault diagnosis of wind turbine system. *International Journal of Renewable Energy Research (IJRER)* **2017**, *7*, 758–769.
18. P. Mazidi and M. Du and L. Bertling Tjernberg and M. A. S. Bobi. A performance and maintenance evaluation framework for wind turbines. 2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), 2016, pp. 1–8. doi:10.1109/PMAPS.2016.7763931.
19. Peyman Mazidi and Lina Bertling Tjernberg and Miguel A Sanz Bobi. Wind turbine prognostics and maintenance management based on a hybrid approach of neural networks and a proportional hazards model. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* **2017**, *231*, 121–129. doi:10.1177/1748006X16686899.
20. Wang, K.S.; Sharma, V.S.; Zhang, Z.Y. SCADA data based condition monitoring of wind turbines. *Advances in Manufacturing* **2014**, *2*, 61–69.

21. Gonzalez, E.; Stephen, B.; Infield, D.; Melero, J. On the use of high-frequency SCADA data for improved wind turbine performance monitoring. Journal of Physics: Conference Series. IOP Publishing, 2017, Vol. 926, p. 012009.
22. Odgaard, P.F.; Stoustrup, J.; Kinnaert, M. Fault tolerant control of wind turbines–a benchmark model. *IFAC Proceedings Volumes* **2009**, *42*, 155–160.
23. KK Wind Solutions. http://www.kkwindsolutions.com/. Accessed: 10-09-2018.
24. The MathWorks, Inc. http://www.mathworks.com/. Accessed: 10-09-2018.
25. Odgaard, P.F.; Stoustrup, J.; Kinnaert, M. Fault-tolerant control of wind turbines: A benchmark model. *IEEE Transactions on Control Systems Technology* **2013**, *21*, 1168–1182.
26. Odgaard, P.; Johnson, K. Wind Turbine Fault Diagnosis and Fault Tolerant Control - an Enhanced Benchmark Challenge. Proc. of the American Control Conference, 2013, pp. 1–6.
27. Ruiz, M.; Mujica, L.E.; Alférez, S.; Acho, L.; Tutivén, C.; Vidal, Y.; Rodellar, J.; Pozo, F. Wind turbine fault detection and classification by means of image texture analysis. *Mechanical Systems and Signal Processing* **2018**, *107*, 149–167.
28. Lackner, M.A.; Rotea, M.A. Passive structural control of offshore wind turbines. *Wind energy* **2011**, *14*, 373–388.
29. Jonkman, Jason and Butterfield, Sandy and Musial, Walter and Scott, George. Definition of a 5-MW reference wind turbine for offshore system development. *National Renewable Energy Laboratory, Golden, CO, Technical Report No. NREL/TP-500-38060* **2009**.
30. May, A.; McMillan, D.; Thöns, S. Economic analysis of condition monitoring systems for offshore wind turbine sub-systems. *IET Renewable Power Generation* **2015**, *9*, 900–907.
31. Hong, X.; Xu, Y.; Zhao, G. LBP-TOP: a Tensor Unfolding Revisit. Asian Conference on Computer Vision. Springer, 2016, pp. 513–527.
32. Vapnik, V. *The nature of statistical learning theory*; Springer Verlag, 1995.
33. Yang, C.H.; Chin, L.C.; Hsieh, S.C. Morse code recognition using support vector machines. Biomedical Engineering, 2003. IEEE EMBS Asian-Pacific Conference on. IEEE, 2003, pp. 220–222.
34. Sergios Theodoridis and Konstantiinos Koutroumbas. *Pattern Recognition*; Elsevier, 2009.
35. Santos, P.; Villa, L.F.; Reñones, A.; Bustillo, A.; Maudes, J. An SVM-based solution for fault detection in wind turbines. *Sensors* **2015**, *15*, 5627–5648.
36. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on geoscience and remote sensing* **2004**, *42*, 1778–1790.
37. McLachlan, G. *Discriminant analysis and statistical pattern recognition*; Vol. 544, John Wiley & Sons, 2004.
38. Devroye, L.; Wagner, T. Distribution-free performance bounds with the resubstitution error estimate (Corresp.). *IEEE Transactions on Information Theory* **1979**, *25*, 208–210.
39. Efron, B.; Tibshirani, R.J. *An introduction to the bootstrap*; CRC press, 1994.
40. Stone, M. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society. Series B (Methodological)* **1974**, pp. 111–147.
41. Leahy, Kevin and Gallagher, Colm and O'Donovan, Peter and Bruton, Ken and O'Sullivan, Dominic. A Robust Prescriptive Framework and Performance Metric for Diagnosing and Predicting Wind Turbine Faults Based on SCADA and Alarms Data with Case Study. *Energies* **2018**, *11*, 1738.

© 2018 by the authors. Submitted to *Energies* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).