



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

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

42 In recent years, there has been efforts to develop FD strategies by analyzing only SCADA data. The
43 use of machine learning techniques has been crucial for this area. For example, in [11] fault prediction
44 and diagnosis for the WT generator is accomplished using real-world SCADA data from two wind
45 power plants located in China based on principal component analysis (PCA) and unsupervised
46 clustering methods. In [12] a FD strategy for WT gearboxes is proposed based on artificial neural
47 networks (ANN) and tested on real-world SCADA data sets of a wind farm in southern Italy. In
48 [13] a strategy to diagnose WT faults from SCADA data using support vector machines (SVM) is
49 advised. Generally, the classification methods that deserve special mention are SVM [11,13–17] and
50 ANN [12,18,19], because of their ability to handle non-linear and noisy data. On one hand, the use
51 of ANN has drawbacks related to their training time and dependability on the optimization of fine
52 tuning their parameters. In particular, in [18] the right number of parameters and their corresponding
53 value must be carefully selected to create a normal behavior model based on ANN. On the other hand
54 SVM is simpler and has successfully proven its suitability in this type of problem. Thus, SVM is the
55 selected classifier in this paper.

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

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

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

83 2. Fault Diagnosis Strategy

84 2.1. Model overview

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

99 The **model** proposes to **simulate** the sensors in **the block diagram environment Simulink** by adding
 100 signals from band limited white noise blocks **that** are parameterized by noise power, to the actual
 101 variables provided by **the FAST software.** These random noise blocks represent measurement noise
 102 either due to **to electrical noise in the system or due to the measuring principle.** The different sensors
 103 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 <i>fore-aft</i> acceleration	$a_{fa,m}$	m/s ²	$5 \cdot 10^{-4}$
S9	Tower top <i>side-to-side</i> acceleration	$a_{ss,m}$	m/s ²	$5 \cdot 10^{-4}$

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

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

112 The stochastic, full-field, turbulent-wind simulator TurbSim —developed by NREL— is used to
 113 generate the wind velocity fields **applied** in the simulations. It **employs a stochastic model**—as opposed
 114 to a physics-based model—to numerically simulate time series of three-component wind-speed vectors.
 115 It provides the ability to drive simulations of **complex** turbine designs with **realistic but** simulated
 116 inflow turbulence environments that **combine** many of the **main** fluid dynamic features known to
 117

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)

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

123 2.2. (Non-) Noise Handling

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

137 2.3. Data Collection

138 In this work, a total of 260 simulations are conducted. In particular 100 with a healthy WT, and
 139 20 simulations for each studied fault. That is (recall there are 8 types of faults, see Table 3), a total of
 140 160 simulations with a faulty WT. All simulations have a duration of 600 seconds. However, only the
 141 last 400 seconds of simulation are used to avoid the transient due to initialization of the numerical

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

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

155 2.4. Data Reshape and Tensor Unfolding

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

- 163 • Fault 8 (F8) is required to fulfill $T_d < 3T_s$. This is the most restrictive detection time as this is the
 164 most severe fault. It is related to the torque actuator and it is noteworthy that the torque rate
 165 limit for the NREL 5-MW WT is 15000 Nm/s [29].
- 166 • Fault 1 (F1) is required to fulfill $T_d < 8T_s$. This fault has a high varying dynamic and is related to
 167 the pitch actuator (high air content in oil). In this case, note that the blade-pitch rate limit for the
 168 NREL 5-MW WT is 8 deg/s as this is speculated to be the blade pitch rate limit of conventional
 169 5-MW machines based on General Electric (GE) Wind's long blade test program [29].
- 170 • Faults 4 to 7 (F4, F5, F6, F7) are required to fulfill $T_d < 10T_s$. These faults are related to the
 171 generator speed sensor and the pitch sensors.
- 172 • Finally, faults 2 and 3 (F2, F3) are required only to satisfy $T_d < 100T_s$ as these are faults with a
 173 very slow dynamic. These faults are related to the pitch actuator (pump wear and hydraulic
 174 leakage).

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

- 177 a) In samples of only $J = 3$ time steps (this will lead to a detection time of approximately $3T_s$).
- 178 b) In samples of $J = 8$ time steps (in this case, detection time is close to $8T_s$).
- 179 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:

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

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

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

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

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

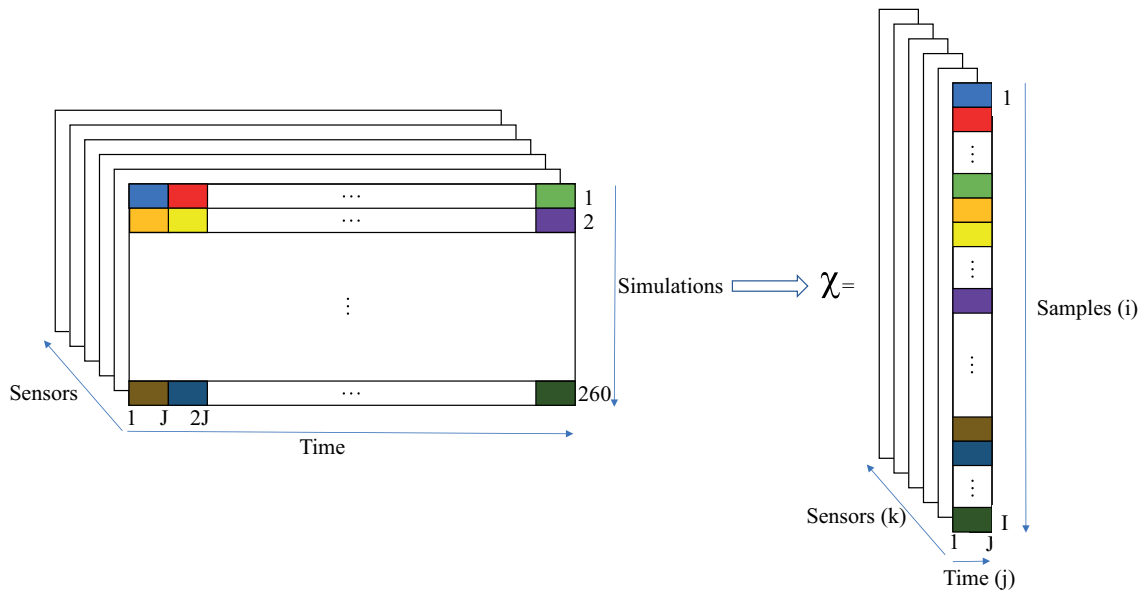


Figure 1. Reshape data from long run simulations (left) into a third-order tensor \mathcal{X} ($I \times J \times K$) with short time samples of J seconds (right).

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

192 Figure 1 illustrates how the available data from the 260 long run simulations, see Equation (1), is
 193 reorganized in a third-order tensor (multidimensional array with three indices) with short time
 194 samples of J time steps, see Equation (2). The first J data-points determine the first sample (represented by
 195 the light blue color box in Figure. 1). Right after, the next J data-points determine the second sample
 196 (red color box), et cetera. After the last J data-points of the first simulation (light green), the first
 197 J data-points of the second simulation (orange box) define the next sample, and so on. In general, let us
 198 consider we have at $j = 1, 2, \dots, J$ time instants, different sensors $k = 1, 2, \dots, K$ stored. Similar data is
 199 generated for a number of samples $i = 1, 2, \dots, I$. This results in the third-order tensor \mathcal{X} ($I \times J \times K$)
 200 as illustrated in Figure 1, where the height (I) gives the number of samples; the width (J) gives the
 201 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

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

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

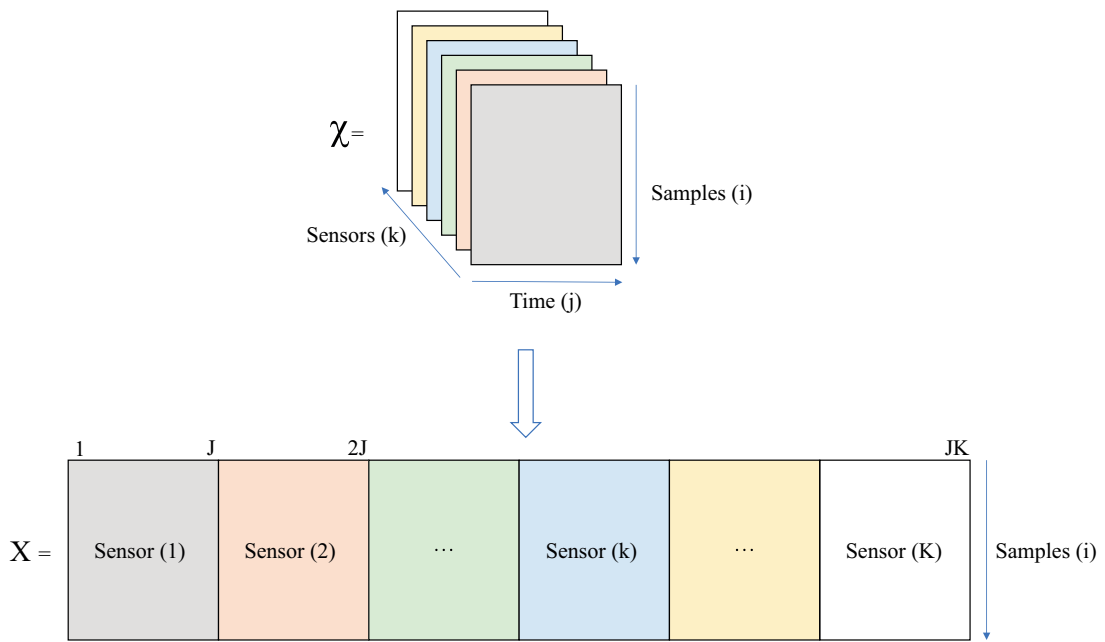


Figure 2. Unfolding of the third-order tensor \mathcal{X} into matrix X .

205 2.5. Autoscaling or Standardization

206 The reason to autoscale the raw data is two-fold: to deal with data that comes from different
 207 sensors and with different magnitudes, and to simplify the computations for the multiway PCA
 208 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

$$\mu_j = \frac{1}{I} \sum_{i=1}^I x_{ij}, \quad j = 1, \dots, JK, \quad (4)$$

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

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

$$\tilde{x}_{ij} := \frac{x_{ij} - \mu_j}{\sigma_j}, \quad i = 1, \dots, I, \quad j = 1, \dots, JK. \quad (6)$$

209 2.6. Multiway PCA

210 Recall that, before using a classifier, the raw data coming from the sensors must be processed to
 211 obtain the most suitable features. In this work, after the autoscaling step, multiway PCA is selected as
 212 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

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

Then the singular value decomposition of S is computed,

$$S = PDP^T \quad (8)$$

where D is a **matrix in diagonal form composed by** the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_{JK}$ 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:

$$\frac{\lambda_1 + \dots + \lambda_d}{\lambda_1 + \dots + \lambda_{JK}}. \quad (9)$$

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

$$Y = \tilde{X}P_d. \quad (10)$$

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

$$h(x) = \omega^T x + b, \quad (11)$$

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

$$\omega^T x_+^{sv} + b = 1, \quad (12)$$

$$\omega^T x_-^{sv} + b = -1, \quad (13)$$

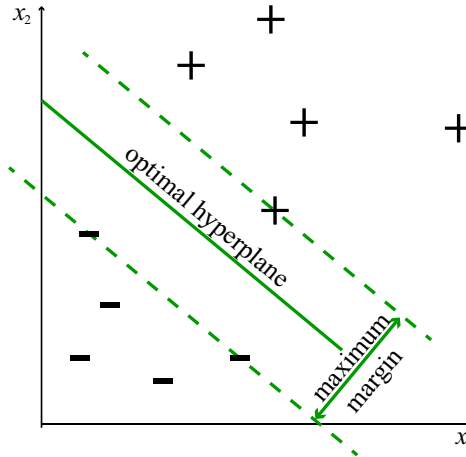


Figure 3. Linear SVM in a two-dimensional example.

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

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

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,

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

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

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

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

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

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

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

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

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

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,

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

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

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

that can be rewritten as

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

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

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

In this case, using Lagrange multipliers, the problem reads

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

223 The final set of restrictions shows why the penalty parameter C is frequently called a box constraint, as
 224 it keeps the admissible values of the Lagrange multipliers in a bounded region. In this work, the box
 225 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,

$$K(x_i, x_j) = \phi(x_i) \phi(x_j). \quad (25)$$

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,

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

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).

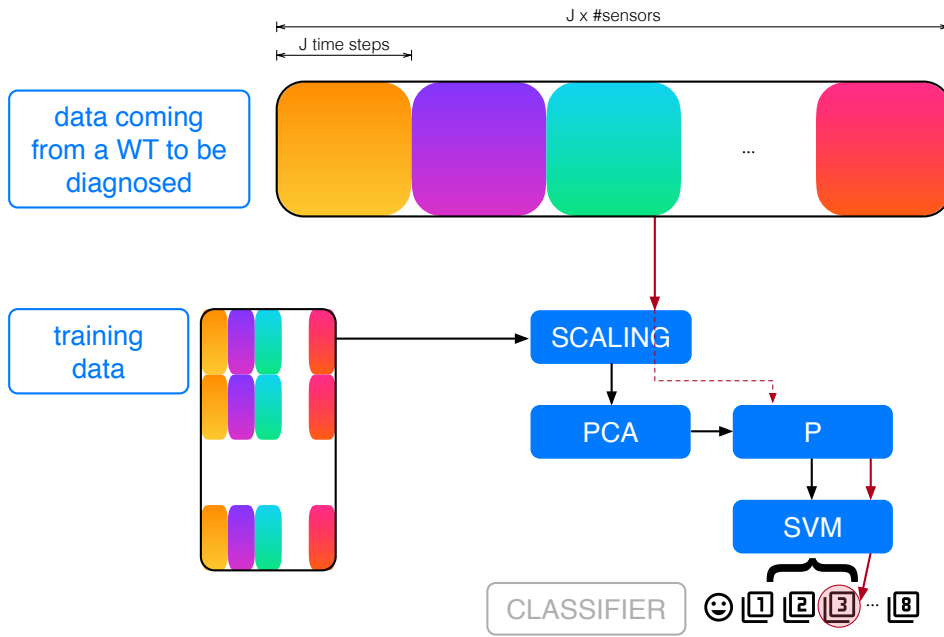


Figure 4. Data coming from a WT to be diagnosed is first scaled, then projected into the vectorial space spanned by the first principal components and finally the projection enters the classifier.

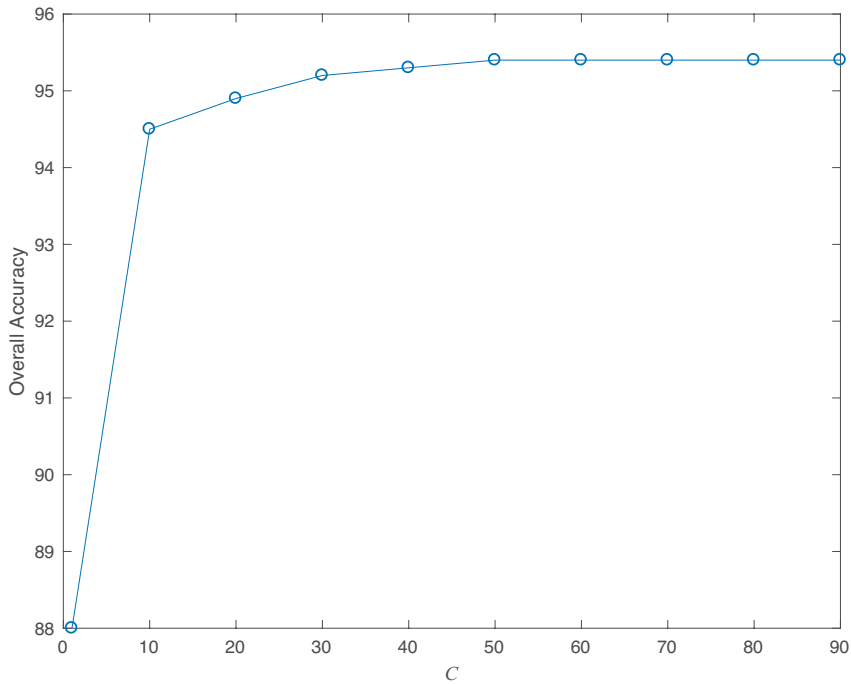


Figure 5. Box constraint value with respect to overall accuracy.

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

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

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

289 When detection time is approximately 8 seconds ($J = 8$), the overall accuracy is 98%. As in the
 290 previous case, the healthy class has a TPR higher than 99%. Fault 1 increases its TPR to 79% (where
 291 16% are missed faults and 5% confusion with Fault 2), and all the other classes increase their TPR to
 292 values higher than 98%. It is noteworthy that Fault 4, related to the generator speed sensor, reaches a
 293 100% TPR. The generator speed measurement from the sensor is used as input in the torque and pitch
 294 controllers, thus being able to correctly diagnose this type of fault is extremely important. As in the
 295 previous case, Fault 8 keeps a 100% TPR.

296 Finally, when $J = 10$ the overall accuracy is 98.2%. In this case, Fault 1 is improved to have a TPR
 297 of 80%. In this case, all misclassifications are 1% or lower, except for Fault 1 that is 15% of the times
 298 misclassified as healthy and 5% of the times misclassified as Fault 2 (recall, also a pitch actuator fault).
 299 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

300 Access to real SCADA datasets is often proprietary. And, therefore, it is not accessible by the
 301 scientific community. To overcome this difficulty, in this work simulated data is obtained by one
 302 of the most widely accepted WT simulators in the scientific community (FAST). The drawbacks of
 303 using simulated data is that there is no possibility to evaluate the proposed method in a full test set
 304 representing the true distribution of real-world data where class imbalance is a challenging problem
 305 [41]. However, there are several references, e.g. [13], where this problem is solved in the training stage
 306 using under/oversampling the training data.

307 4. Conclusion

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

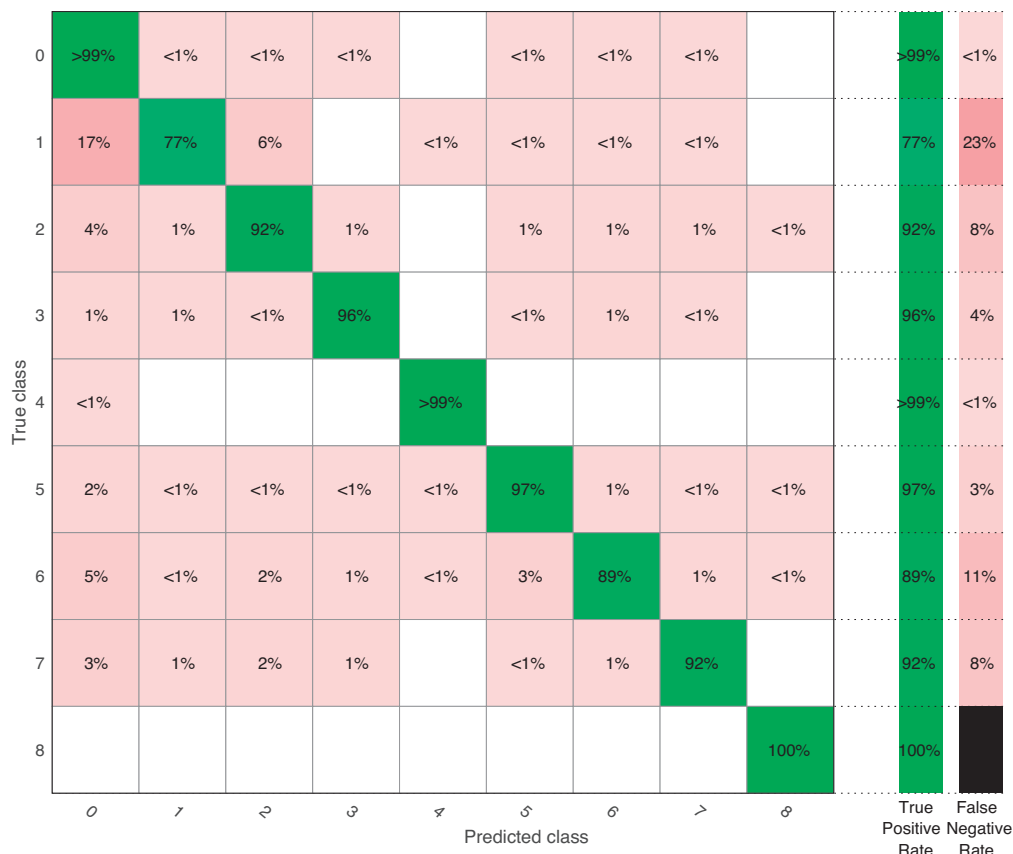


Figure 6. Confusion matrix when $J = 3$.

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

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

321 **Author Contributions:** All authors contributed equally to this work.

322 **Funding:** This work has been partially funded by the Spanish Ministerio de Economía, Industria y Competitividad
 323 (MINECO) through the research project DPI2014-58427-C2-1-R; by the Spanish Agencia Estatal de Investigación
 324 (AEI) - Ministerio de Economía, Industria y Competitividad (MINECO), and the Fondo Europeo de Desarrollo
 325 Regional (FEDER) through the research project DPI2017-82930-C2-1-R; and by the Generalitat de Catalunya
 326 through the research project 2017 SGR 388. We gratefully acknowledge the support of NVIDIA Corporation with
 327 the donation of the Titan Xp GPU used for this research.

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

331 **References**

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

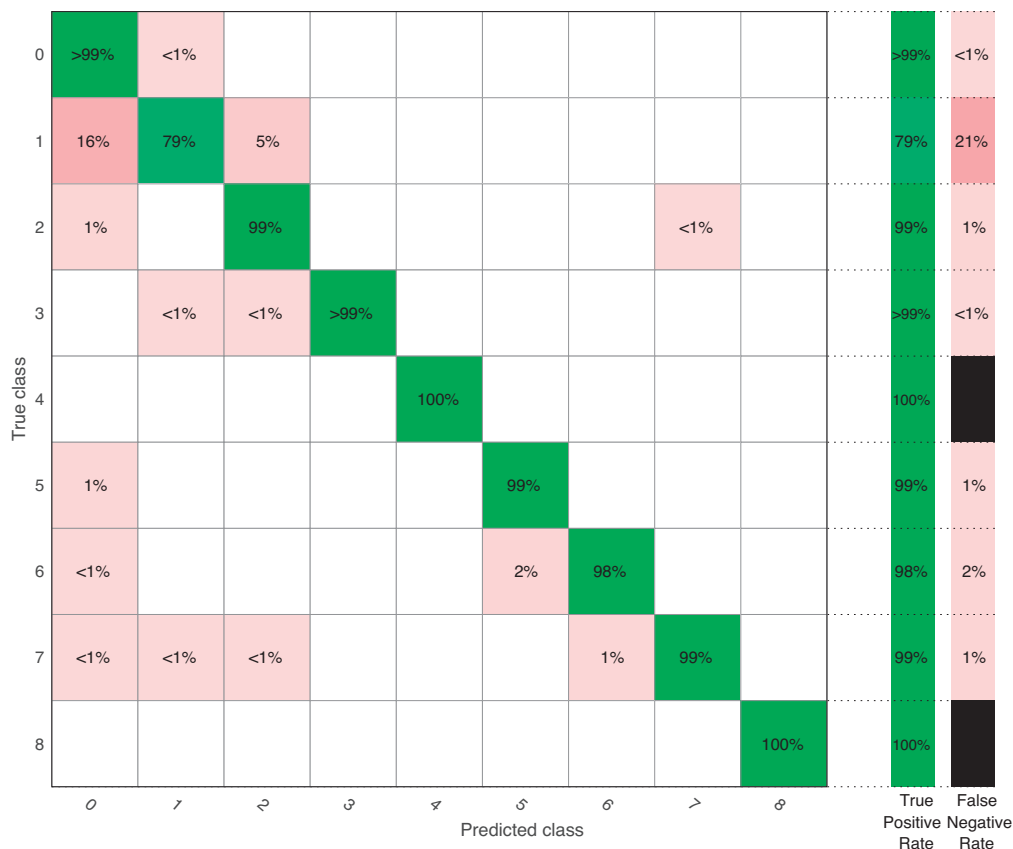


Figure 7. Confusion matrix when $J = 8$.

336 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
 337 monitoring and fault diagnosis methods in wind turbines: A review. *Renewable and Sustainable Energy*
 338 *Reviews* **2015**, *44*, 466 – 472. doi:https://doi.org/10.1016/j.rser.2014.12.005.

339 4. Gao, Z.; Cecati, C.; Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques?Part I: Fault
 340 diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics* **2015**,
 341 *62*, 3757–3767.

342 5. de Azevedo, Henrique Dias Machado and Araújo, Alex Maurício and Bouchonneau, Nadège. A review
 343 of wind turbine bearing condition monitoring: State of the art and challenges. *Renewable and Sustainable*
 344 *Energy Reviews* **2016**, *56*, 368–379.

345 6. Kandukuri, Surya Teja and Klausen, Andreas and Karimi, Hamid Reza and Robbersmyr, Kjell Gunnar. A
 346 review of diagnostics and prognostics of low-speed machinery towards wind turbine farm-level health
 347 management. *Renewable and Sustainable Energy Reviews* **2016**, *53*, 697–708.

348 7. Huang, Shoudao and Wu, Xuan and Liu, Xiao and Gao, Jian and He, Yunze. Overview of condition
 349 monitoring and operation control of electric power conversion systems in direct-drive wind turbines under
 350 faults. *Frontiers of Mechanical Engineering* **2017**, *12*, 281–302.

351 8. Yang, Zhimin and Chai, Yi. A survey of fault diagnosis for onshore grid-connected converter in wind
 352 energy conversion systems. *Renewable and Sustainable Energy Reviews* **2016**, *66*, 345–359.

353 9. Ochieng, Francis Xavier and Hancock, Craig Matthew and Roberts, Gethin Wyn and Le Kernec, Julien.
 354 A review of ground-based radar as a noncontact sensor for structural health monitoring of in-field wind
 355 turbines blades. *Wind Energy* **2018**.

356 10. Shohag, Md Abu S and Hammel, Emily C and Olawale, David O and Okoli, Okenwa I. Damage mitigation
 357 techniques in wind turbine blades: A review. *Wind Engineering* **2017**, *41*, 185–210.

358 11. Zhao, Y.; Li, D.; Dong, A.; Kang, D.; Lv, Q.; Shang, L. Fault Prediction and Diagnosis of Wind Turbine
 359 Generators Using SCADA Data. *Energies* **2017**, *10*, 1210.

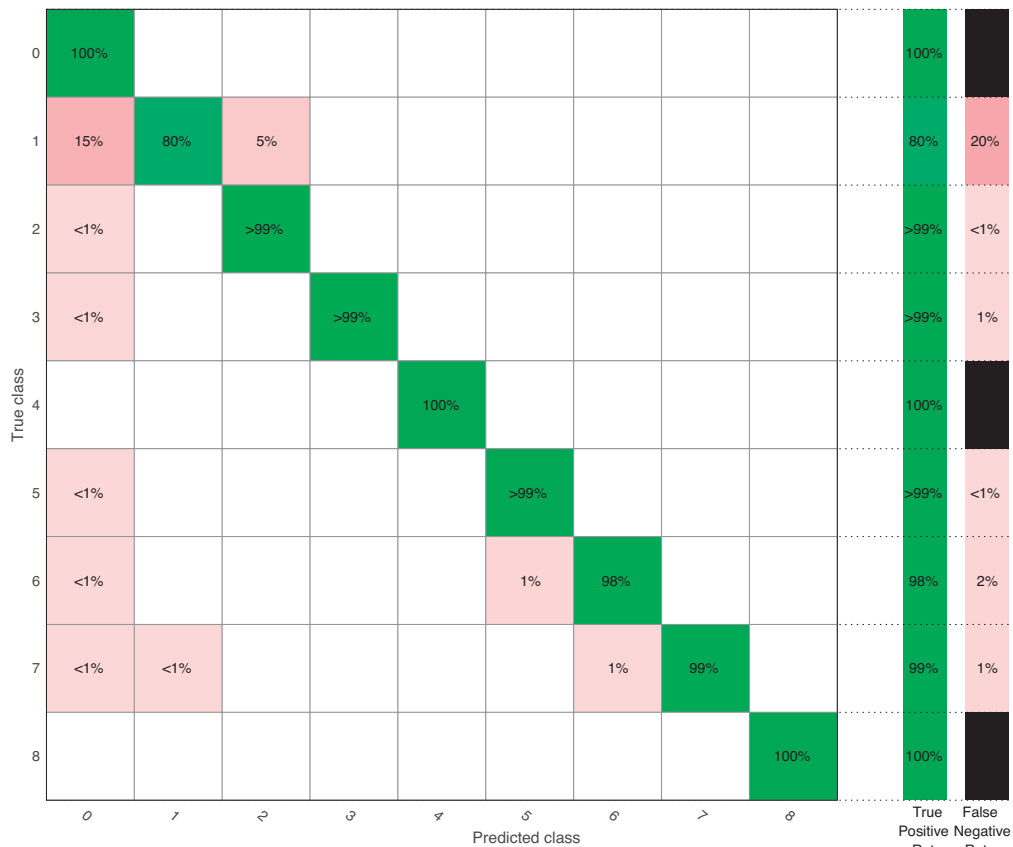


Figure 8. Confusion matrix when $J = 10$.

- 360 12. Astolfi, Davide and Castellani, Francesco and Scappaticci, Lorenzo and Terzi, Ludovico. Diagnosis of
 361 wind turbine misalignment through SCADA data. *Diagnostyka* **2017**, *18*.
- 362 13. Leahy, K.; Hu, R. L.; Konstantakopoulos, I. C.; Spanos, C. J.; Agogino, A. M.; O'Sullivan, D. T. J. Diagnosing
 363 and predicting wind turbine faults from SCADA data using support vector machines. *International Journal*
 364 *of Prognostics and Health Management* **2018**, *9*, 1–11.
- 365 14. Laouti, N.; Sheibat, N.; Othman, S. Support vector machines for fault detection in wind turbines.
 366 Proceedings of IFAC World Congress, 2011, Vol. 2, pp. 7067–707.
- 367 15. Laouti, Nassim and Othman, Sami and Alamir, Mazen and Sheibat-Othman, Nida. Combination
 368 of model-based observer and support vector machines for fault detection of wind turbines. *International*
 369 *Journal of Automation and computing* **2014**, *11*, 274–287.
- 370 16. Xiao, Y.; Hong, Y.; Chen, X.; Chen, W. The application of dual-tree complex wavelet transform (DTCWT)
 371 energy entropy in misalignment fault diagnosis of doubly-fed wind turbine (DFWT). *Entropy* **2017**, *19*, 587.
- 372 17. Abdelkrim, S.; Djamel, M.M.; Samia, A.; Hayet, M.; Mawloud, T. The MAED and SVM for fault diagnosis
 373 of wind turbine system. *International Journal of Renewable Energy Research (IJRER)* **2017**, *7*, 758–769.
- 374 18. P. Mazidi and M. Du and L. Bertling Tjernberg and M. A. S. Bobi. A performance and maintenance
 375 evaluation framework for wind turbines. 2016 International Conference on Probabilistic Methods Applied
 376 to Power Systems (PMAPS), 2016, pp. 1–8. doi:10.1109/PMAPS.2016.7763931.
- 377 19. Peyman Mazidi and Lina Bertling Tjernberg and Miguel A Sanz Bobi. Wind turbine prognostics and
 378 maintenance management based on a hybrid approach of neural networks and a proportional hazards
 379 model. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* **2017**,
 380 *231*, 121–129. doi:10.1177/1748006X16686899.
- 381 20. Wang, K.S.; Sharma, V.S.; Zhang, Z.Y. SCADA data based condition monitoring of wind turbines. *Advances*
 382 *in Manufacturing* **2014**, *2*, 61–69.

- 383 21. Gonzalez, E.; Stephen, B.; Infield, D.; Melero, J. On the use of high-frequency SCADA data for improved
384 wind turbine performance monitoring. *Journal of Physics: Conference Series*. IOP Publishing, 2017, Vol.
385 926, p. 012009.
- 386 22. Odgaard, P.F.; Stoustrup, J.; Kinnaert, M. Fault tolerant control of wind turbines—a benchmark model. *IFAC*
387 *Proceedings Volumes* **2009**, *42*, 155–160.
- 388 23. **KK Wind Solutions**. <http://www.kkwindsolutions.com/>. Accessed: 10-09-2018.
- 389 24. **The MathWorks, Inc.** <http://www.mathworks.com/>. Accessed: 10-09-2018.
- 390 25. Odgaard, P.F.; Stoustrup, J.; Kinnaert, M. Fault-tolerant control of wind turbines: A benchmark model.
391 *IEEE Transactions on Control Systems Technology* **2013**, *21*, 1168–1182.
- 392 26. Odgaard, P.; Johnson, K. Wind Turbine Fault Diagnosis and Fault Tolerant Control - an Enhanced
393 Benchmark Challenge. *Proc. of the American Control Conference*, 2013, pp. 1–6.
- 394 27. Ruiz, M.; Mujica, L.E.; Alférez, S.; Acho, L.; Tutivén, C.; Vidal, Y.; Rodellar, J.; Pozo, F. Wind turbine fault
395 detection and classification by means of image texture analysis. *Mechanical Systems and Signal Processing*
396 **2018**, *107*, 149–167.
- 397 28. Lackner, M.A.; Rotea, M.A. Passive structural control of offshore wind turbines. *Wind energy* **2011**,
398 *14*, 373–388.
- 399 29. **Jonkman, Jason and Butterfield, Sandy and Musial, Walter and Scott, George**. Definition of a 5-MW
400 reference wind turbine for offshore system development. *National Renewable Energy Laboratory, Golden, CO,*
401 *Technical Report No. NREL/TP-500-38060* **2009**.
- 402 30. May, A.; McMillan, D.; Thöns, S. Economic analysis of condition monitoring systems for offshore wind
403 turbine sub-systems. *IET Renewable Power Generation* **2015**, *9*, 900–907.
- 404 31. Hong, X.; Xu, Y.; Zhao, G. LBP-TOP: a Tensor Unfolding Revisit. *Asian Conference on Computer Vision*.
405 Springer, 2016, pp. 513–527.
- 406 32. Vapnik, V. *The nature of statistical learning theory*; Springer Verlag, 1995.
- 407 33. Yang, C.H.; Chin, L.C.; Hsieh, S.C. Morse code recognition using support vector machines. *Biomedical*
408 *Engineering*, 2003. *IEEE EMBS Asian-Pacific Conference on*. IEEE, 2003, pp. 220–222.
- 409 34. **Sergios Theodoridis and Konstantinos Koutroumbas**. *Pattern Recognition*; Elsevier, 2009.
- 410 35. Santos, P.; Villa, L.F.; Reñones, A.; Bustillo, A.; Maudes, J. An SVM-based solution for fault detection in
411 wind turbines. *Sensors* **2015**, *15*, 5627–5648.
- 412 36. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector
413 machines. *IEEE Transactions on geoscience and remote sensing* **2004**, *42*, 1778–1790.
- 414 37. McLachlan, G. *Discriminant analysis and statistical pattern recognition*; Vol. 544, John Wiley & Sons, 2004.
- 415 38. Devroye, L.; Wagner, T. Distribution-free performance bounds with the resubstitution error estimate
416 (Corresp.). *IEEE Transactions on Information Theory* **1979**, *25*, 208–210.
- 417 39. Efron, B.; Tibshirani, R.J. *An introduction to the bootstrap*; CRC press, 1994.
- 418 40. Stone, M. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical*
419 *society. Series B (Methodological)* **1974**, pp. 111–147.
- 420 41. **Leahy, Kevin and Gallagher, Colm and O'Donovan, Peter and Bruton, Ken and O'Sullivan, Dominic**. **A**
421 **Robust Prescriptive Framework and Performance Metric for Diagnosing and Predicting Wind Turbine**
422 **Faults Based on SCADA and Alarms Data with Case Study**. *Energies* **2018**, *11*, 1738.