applied sciences

MDPI

Article

Damage Diagnosis for Offshore Wind Turbine Foundations Based on the Fractal Dimension

Ervin Hoxha, Yolanda Vidal and Francesc Pozo *

Campus Diagonal-Besòs (CDB), Control, Modeling, Identification and Applications (CoDAlab), Department of Mathematics, Escola d'Enginyeria de Barcelona Est (EEBE), Universitat Politècnica de Catalunya (UPC), Eduard Maristany, 16, 08019 Barcelona, Spain; ervin.hoxha@upc.edu (E.H.); yolanda.vidal@upc.edu (Y.V.)

* Correspondence: francesc.pozo@upc.edu; Tel.: +34-934-137-316

Received: 11 September 2020; Accepted: 29 September 2020; Published: 5 October 2020

Abstract: Cost-competitiveness of offshore wind depends heavily in its capacity to switch preventive maintenance to condition-based maintenance. That is, to monitor the actual condition of the wind turbine (WT) to decide when and which maintenance needs to be done. In particular, structural health monitoring (SHM) to monitor the foundation (support structure) condition is of utmost importance in offshore-fixed wind turbines. In this work a SHM strategy is presented to monitor online and during service a WT offshore jacket-type foundation. Standard SHM techniques, as guided waves with a known input excitation, cannot be used in a straightforward way in this particular application where unknown external perturbations as wind and waves are always present. To face this challenge, a vibration-response-only SHM strategy is proposed via machine learning methods. In this sense, the fractal dimension is proposed as a suitable feature to identify and classify different types of damage. The proposed proof-of-concept technique is validated in an experimental laboratory down-scaled jacket WT foundation undergoing different types of damage.

Keywords: fractal dimension; structural health monitoring; offshore wind turbine; *k*NN; support vector machines

1. Introduction

Structural health monitoring's (SHM) main purpose is to diagnose in time damage that affects the integrity of a structure and determine whether repair or reinforcement actions are required to avoid or delay its degradation. Generally, SHM strategies consist of the following steps:

(i) the strategic placement of sensors in the overall structure;
(ii) data collection and communication; and
(iii) analysis of the measured data.

It is important to note that, in a wide variety of applications, guided waves, which is a nondestructive approach, is the usual standard. This approach relies on exciting the structure with low frequency ultrasonic waves and then sensing the reflected response waves. Thus, the method relies heavily on the fact that the input excitation is known and also that other perturbations can be filtered or neglected. On the one hand, in civil infrastructures, such as bridges, it is feasible to assume that external perturbations can be neglected or filtered with respect to the induced excitation, see [1,2]. On the other hand, in other applications, as in aerospace, the structure can only be diagnosed with this approach when it is not in service. This strategy is used, for example, in [3] where a multiarea scanning ultrasonic system is built in a hangar to rapidly scan the airplane overall structure. This type of *not in service* diagnose (the airplane can be diagnosed during no-flight conditions, when it is in the hangar) or neglecting the external perturbations (as in SHM for standard civil structures as bridges or buildings) cannot be straightforwardly extrapolated to the main research area of the present work: wind turbines. Online and in-service SHM for wind turbines (WTs) are extremely important. WTs are extremely large structures subject to remarkable external unknown excitations such as wind and waves in the offshore case. Thus, SHM strategies for WTs must be able to cope with unknown significant external excitations hindering the use of the standard exciting-and-sensing approach [4]. To face this challenge, in this work, a vibration-response-only SHM strategy is stated to monitor online and during service a WT offshore fixed foundation by using only the excitation caused by the external and unknown perturbations.

Offshore wind power will expand dramatically in the next two decades, multiplying by 15 by 2040 to a minimum of 345 gigawatts (GW) of installed capacity, according to the Offshore Wind Outlook 2019 report of the International Energy Agency [5]. However, this achievement will only be possible through cost-competitiveness of offshore wind, which depends entirely on SHM capacity to switch preventive maintenance to predictive one [6]. Thus, SHM for offshore assets is imperative to guarantee its exploitability. Hence, in this work, a SHM methodology for offshore fixed foundations is proposed.

Nowadays, the SHM systems for WTs are mostly deployed to blades [7] and tower [8] but research of SHM for offshore support structures is still scarce [9]. The state of the art in this very specific area has three main research lines:

(i) model-based, using, for example, the finite element method as in [10–12];
(ii) data-based using solely experimental and/or real data; and
(iii) a hybrid approach that makes use of real and/or experimental data and numerical models.

Regarding the first option, the work of Stutzamnn et al. is noteworthy [13] where crack detection of monopile offshore foundations is accomplished based on numerical simulations of fatigue cracks. Regarding the second option, a comprehensive review is given in [14] about SHM of offshore WTs through the statistical pattern recognition paradigm. In this review, it is shown that the usual strategy, regarding offshore WT damage detection, is to identify changes in the modal properties. However, this strategy requires detailed attention to take into account the operational and environmental impact, and usually only damage detection (but not classification) is accomplished. For example, in [15] a SHM approach verified on a full-scale foundation is presented. However, dynamic variability between different operational cases only allows the final results to indicate an overall stiffening of the structure but not to conclude whether damage is present or not. Regarding the third approach, the work by Gomez et al. [16] is noteworthy based on acceleration response data and calibrated computer models. However, this work is based in the usual operational modal analysis and holds the difficulties of this type of approach including the fact that only detection (but not classification of damage type) is acquired. In this work, facing the challenge posed by the previous references, different damage types are taken into account and its classification is achieved in an experimental down-scaled jacket WT foundation. It should be noted that the experimental testbed is a reduced model but well-founded for this proof-of-concept work as it is comparable to that employed in the following works: (i) [17], where damage detection is achieved via damage indicators; (ii) [18], where damage detection is obtained via statistical time series analysis; (iii) [19], based on principal component analysis and support vector machines; and (iv) [20], where a deep learning approach based on convolutional neural networks is employed.

It is well known that machine learning requires a feature extraction preprocess. It is a challenge to find suitable features, sensitive to physical characteristics, that lead to the identification of the damage or fault [21]. In this work, the fractal dimension (FD) of the data time series is employed as the main feature. The FD has been used traditionally as a feature for medicine applications. For example, in [22] experiments on intensive care unit data sets show that the FD characterizes the time series better than the correlation dimension; in [23] FD is proven to be discriminant for the detection of epileptic seizures in intracranial electroencephalogram signals; and in [24] glaucomatous eye detection is proposed based on FD estimation. However, it was not until recently that FD has been explored as feature for structural damage detection. It is important to note the recent work by Rezaie et al. [25], where FD for crack pattern recognition is studied. It is also important to note the work by Wen et al. [26] where FD is shown to be effective to realize fault diagnosis of rolling element bearings and cope with the effects of variation in operating conditions. In this work, the FD feature is proposed for the vibration-response signals inspired by the physical insight that the different fractal structures of these signals should be capable to discriminate different types of damage in jacket-type offshore foundations.

The paper is arranged as follows. First, the laboratory test bed and damage scenarios are briefly introduced in Section 2. Section 3 addresses the detailed statement of the developed damage diagnosis strategy that encompasses the following steps:

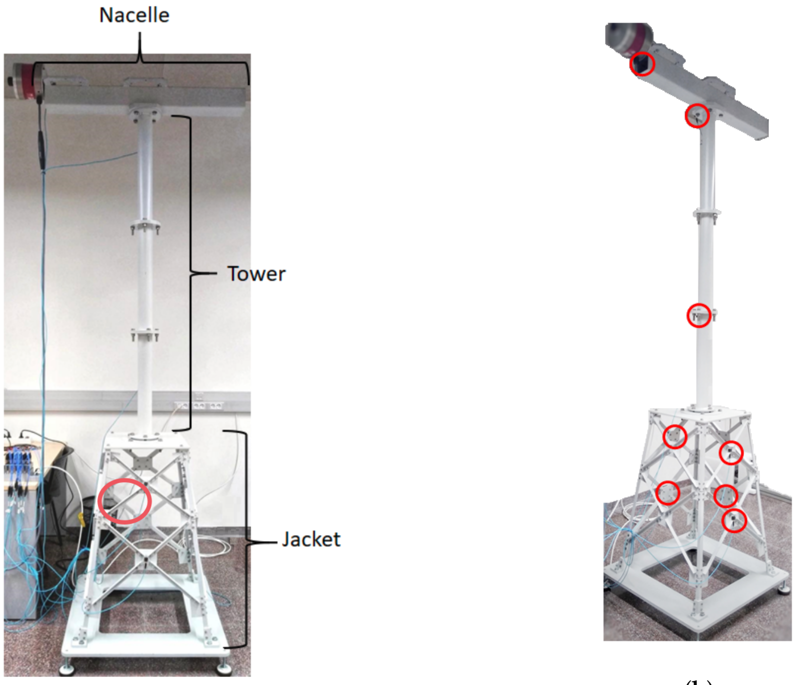
(i) data collection and manipulation;
(ii) fractal dimension feature extraction by means of the Katz's algorithm; and
(iii) normalization and classification tools.

The experimental results are comprehensively stated in Section 4. Finally, conclusions are drawn in Section 5.

2. Experimental Test Bed

The reliability of the damage diagnosis approach presented in this paper is verified using different types of damage in an experimental test bed modeling a jacket-type WT as in [19]. For a very detailed description of the function generator, the amplifier and inertial shaker, the sensor network, the data acquisition system, how the vibration signals are acquired and how the time domain waveforms are processed, readers are referred to [19,20].

A brief characterization of the experimental setup of the small scale wind turbine is described below. First, a function generator (model GW INSTEK AF-2005) is used to produce a white noise signal with four different amplitudes (0.5, 1, 2, and 3) that account for different wind speed regions. This signal is then amplified and used as input to a modal shaker (GW-IV47 from Data Physics) that induces vibration in the structure. The overall description of the test bench is displayed in Figure 1a.



(a)

(b)

Figure 1. (**a**) The test bench detailing the location of the damaged bar (red circle); and (**b**) Location of the sensors.

The structure is 2.7 m high and consists of three parts:

(i) the top beam;
(ii) the tower; and
(iii) the jacket.

The top beam is 1 meter wide and 0.6 meters high and the inertial shaker is attached to one of the ends of the beam. Three tubular sections united with bolts form the tower. Finally, the jacket is a pyramidal structure composed of steel bars of different lengths as well as steel sheets.

The vibration of the structure is measured by means of the data acquisition system cDAQ-9188 (National Instruments) and through 8 triaxial accelerometers (model 356A17, PCB Piezotronic) optimally placed following the work by Zugasti (2014) [17], as can be seen in Figure 1b.

In this work we have considered the same 4 different structural states as in the work by Puruncajas et al. [20]. All of the structural states refer to the jacket bar illustrated in Figure 1a. These states are:

(i) the healthy structure with the original healthy steel bar;
(ii) the healthy structure where the original bar is replaced by a replica;
(iii) the structure with a 5 mm crack damaged bar; and
(iv) the structure with an unlocked bolt in the jacket.

3. Damage Diagnose Strategy

In this section the damage diagnosis strategy is stated. First, a detailed description on data collection and manipulation is given. On the one hand, how data is collected and reshaped is of utmost importance in machine learning in general and for this specific application in particular, see [27,28]. On the other hand, it is well known that feature selection allows to improve the classification performance making faster and more profitable the classifiers [21]. In this regard, the fractal dimension feature is introduced for damage classification purposes, as well as a physical insight of its nature for time series and a detailed explanation about the Katz's algorithm used to compute it. Finally, three machine learning classifiers are reviewed and tested for damage classification.

3.1. Data Collection and Manipulation

A total of 100 experimental tests have been conducted that include the four amplitudes that represent the different speed regions. More precisely:

(i) 10 tests with the original healthy bar for each amplitude, i.e., 40 tests;
(ii) 5 tests with the replica bar for each amplitude, i.e., 20 tests;
(iii) 5 tests with the 5 mm crack bar for each amplitude, i.e., 20 tests; and
(iv) 5 tests with the unlocked bolt for each amplitude, i.e., 20 tests.

For each experimental test, the acceleration has been measured through 24 sensors during 59.51636719 s and with a sampling frequency of 275.28 Hz, which leads to 16,384 time instants and a time step of about $\Delta = 0.0036328125$ s.

The raw data of the k-th experimental test, $k = 1, \ldots, 100$, can be arranged as the matrix $\mathbf{X}^{(k)}$ in Equation (1). Each of the 24 columns of matrix $\mathbf{X}^{(k)}$ contain the 16,384 measures of each sensor:

$$\mathbf{X}^{(k)} = \begin{array}{c} \begin{array}{ccccc} \text{sensor \#1} & \text{sensor \#2} & \text{sensor \#3} & \cdots & \text{sensor \#24} \end{array} \\ \left(\begin{array}{ccccc} x_{1,1}^{(k)} & x_{1,2}^{(k)} & x_{1,3}^{(k)} & \cdots & x_{1,24}^{(k)} \\ x_{2,1}^{(k)} & x_{2,2}^{(k)} & x_{2,3}^{(k)} & \cdots & x_{2,24}^{(k)} \\ x_{3,1}^{(k)} & x_{3,2}^{(k)} & x_{3,3}^{(k)} & \cdots & x_{3,24}^{(k)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{16383,1}^{(k)} & x_{16383,2}^{(k)} & x_{16383,3}^{(k)} & \cdots & x_{16383,24}^{(k)} \\ x_{16384,1}^{(k)} & x_{16384,2}^{(k)} & x_{16384,3}^{(k)} & \cdots & x_{16384,24}^{(k)} \end{array}\right) \end{array} \in \mathcal{M}_{16384\times 24}(\mathbb{R}) \tag{1}$$

Each column of the matrix $\mathbf{X}^{(k)}$ in Equation (1) is reshaped into a 64-by-256 matrix to build a new matrix $\mathbf{Y}^{(k)} \in \mathcal{M}_{64\times(256\cdot 24)}(\mathbb{R})$ in Equation (2):

$$\mathbf{Y}^{(k)} = \begin{array}{c} \begin{array}{ccc} \text{sensor \#1} & \cdots & \text{sensor \#24} \end{array} \\ \left(\begin{array}{cccc|c|cccc} x_{1,1}^{(k)} & x_{2,1}^{(k)} & \cdots & x_{256,1}^{(k)} & \cdots & x_{1,24}^{(k)} & x_{2,24}^{(k)} & \cdots & x_{256,24}^{(k)} \\ x_{257,1}^{(k)} & x_{258,1}^{(k)} & \cdots & x_{512,1}^{(k)} & \cdots & x_{257,24}^{(k)} & x_{258,24}^{(k)} & \cdots & x_{512,24}^{(k)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{16129,1}^{(k)} & x_{16130,1}^{(k)} & \cdots & x_{16384,1}^{(k)} & \cdots & x_{16129,24}^{(k)} & x_{16130,24}^{(k)} & \cdots & x_{16384,24}^{(k)} \end{array}\right) \end{array} \tag{2}$$

Two are the main reasons for reshaping the matrix $\mathbf{Y}^{(k)}$ in Equation (2):

(i) on the one hand, for a single experimental test, we create 64 rows. Each one of these rows is what we call a *sample*;

(ii) on the other hand, each sample will contain time-history measures of the whole set of sensors.

We will see in Section 4 that when we want to diagnose whether a wind turbine is healthy or not, we just need to measure these 24 sensors during 256 time instants, that is, during $256\Delta \approx 0.93$ s.

To define the matrix that contains all the data, the matrices $\mathbf{Y}^{(k)}$, $k = 1, \ldots, 100$, from each experiment, are stacked to define

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}^{(1)} \\ \mathbf{Y}^{(2)} \\ \vdots \\ \mathbf{Y}^{(100)} \end{pmatrix} \in \mathcal{M}_{(64\cdot 100)\times(256\cdot 24)}(\mathbb{R}) = \mathcal{M}_{6400\times 6144}(\mathbb{R}). \tag{3}$$

3.2. Fractal Dimension

Fractal geometry was proposed by Benoît Mandelbrot [29] and it is a relatively new mathematics discipline which has found a lot of applications in bio-science [30–32], engineering [33] and many other fields [34].

Euclidean geometry describes common geometric forms like lines, planes, spheres or rectangular volumes. Each of the geometric objects considered so far has an integer dimension (D), either $1, 2$, or 3. However, many natural shapes do not harmonize with the integer-based idea of dimension.

In order to give meaning to noninteger dimensions, a more mathematical description of dimension proposed by P. Bourke [35] is based on "how the *size* of an object behaves as the linear dimension increase". More precisely, consider, for instance, three objects with dimensions $D = 1$ (a line segment); $D = 2$ (a square); and $D = 3$ (a cube). If the line segment, the square and the cube are linearly scaled by a factor of 2, then the results are 2 copies, 4 copies, and 8 copies of the initial objects, respectively. In other words, the length (characteristic size) of the line segment is doubled (Figure 2a), the area (characteristic size) of the square increases by a factor of 4 (Figure 2b), and the volume (characteristic size) of the cube increases by a factor of 8 (Figure 2c).

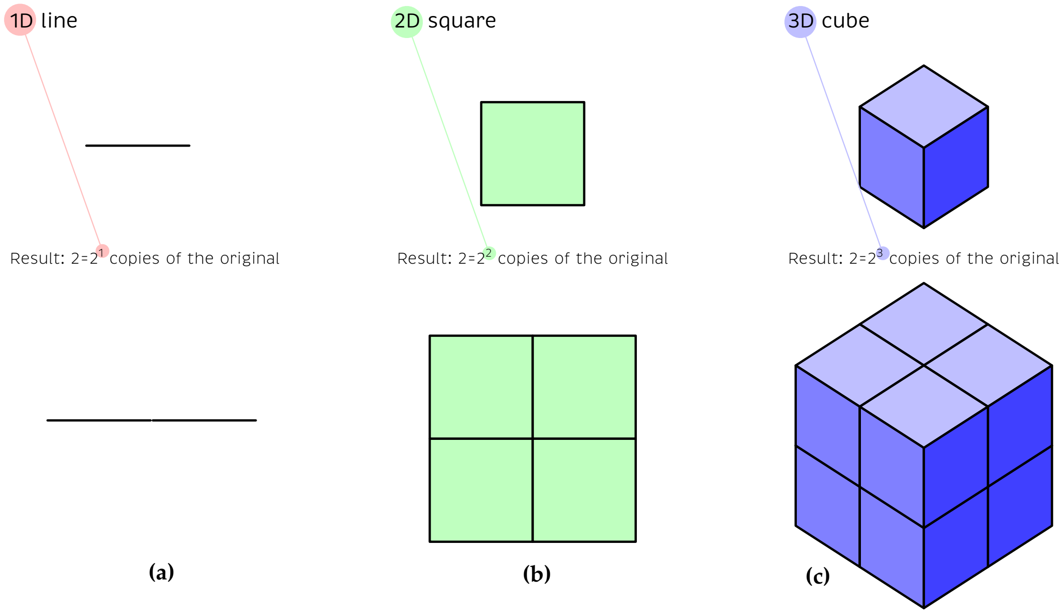

1D line
Result: 2=2^1 copies of the original
2D square
Result: 2=2^2 copies of the original
3D cube
Result: 2=2^3 copies of the original
(a)
(b)
(c)

Figure 2. When objects of different dimension (e.g., (**a**) line, (**b**) square, and (**c**) cube) are linearly scaled by a factor of 2, their characteristic size will have different results associated to their dimension.

The relation between the scaling factor S, the dimension D and the number of generated copies N (increasing size) can be generalized and expressed as:

$$N = S^D, \tag{4}$$

which is equivalent to

$$D = \frac{\log(N)}{\log(S)}. \tag{5}$$

Since D is defined in terms of N and S in Equation (5), it is possible to find the dimension, for instance, of the famous Koch curve [36]. In the case of the Koch curve, at each step, we divide the line segment into $S = 3$ segments of equal length and we draw an equilateral triangle that has the middle as its base and points outward. Therefore, we have created $N = 4$ copies (the two external sides of the original line segment and the two sides of the triangle). Consequently, the fractal dimension D_{Koch} of the Koch curve is:

$$D_{\text{Koch}} = \frac{\log(4)}{\log(3)} \approx 1.2619.$$

As it is very well known, fractals are self-similar subsets of the Euclidean space where the fractal dimension defined in Equation (5) surpasses their topological dimension. Fractals have the same appearance at different scales. In this sense, many time series of different processes can be considered as fractals, since many parts taken from these time series, scaled by proper factors, are similar to the whole series. Considering that the fractal dimension is, somehow, a measure of the complexity that is repeating on each scale, it seems very interesting to compute the fractal dimension of a time series. In this regard, there are several algorithms that can be applied to estimate the fractal dimension of a time series. The approach used in this paper to estimate the fractal dimension is Katz's algorithm, that is summarized in Section 3.2.1.

3.2.1. Katz's Algorithm

For a given sensor $\tau = 1, \ldots, 24$, the time series used in this work are the rows in matrix $\mathbf{Y}$ in Equation (3). More precisely, for a given row $i = 1, \ldots, 6400$ and a given sensor $\tau = 1, \ldots, 24$, the associated time series are composed of a sequence of $\nu = 256$ points, $s^1_{i,\tau}, s^2_{i,\tau}, \ldots, s^{\nu}_{i,\tau} \in \mathbb{R}^2$ where

$$s^j_{i,\tau} = (j, \mathbf{Y}\left[i, j + \nu(\tau - 1)\right]) \in \mathbb{R}^2, \; j = 1, \ldots, \nu,$$

where $\mathbf{Y}[\alpha, \beta]$ represents the element in the α-th row and β-th column of matrix $\mathbf{Y}$.

To estimate the fractal dimension of the time series, Katz [37] defines two magnitudes, L and d, see Figure 3. On the one hand, the total length of the curve L is defined as the sum of the distance between two consecutive points. More precisely, for a given row $i = 1, \ldots, 6400$ and a given sensor $\tau = 1, \ldots, 24$:

$$L_{i,\tau} = \sum_{j=1}^{\nu-1} \left\| s^{j+1}_{i,\tau} - s^j_{i,\tau} \right\|_2 = \sum_{j=1}^{\nu-1} \sqrt{1 + (\mathbf{Y}\left[i, j+1+\nu(\tau-1)\right] - \mathbf{Y}\left[i, j+\nu(\tau-1)\right])^2}. \tag{6}$$

On the other hand, d is the diameter or planar extent of the time series and it is defined as the maximum distance between the first point in the time series and the rest of points. More precisely, for a given row $i = 1, \ldots, 6400$ and a given sensor $\tau = 1, \ldots, 24$:

$$d_{i,\tau} = \max_{j=2,\ldots,\nu} \left\| s^j_{i,\tau} - s^1_{i,\tau} \right\|_2. \tag{7}$$

The last step in the Katz's algorithm is the normalization of both $L_{i,\tau}$ and $d_{i,\tau}$ by the average distance $a_{i,\tau}$ between two consecutive points. More precisely, for a given row $i = 1, \ldots, 6400$ and a given sensor $\tau = 1, \ldots, 24$:

$$a_{i,\tau} = \frac{L_{i,\tau}}{\nu - 1}. \tag{8}$$

Finally, for a given row $i = 1, \ldots, 6400$ and a given sensor $\tau = 1, \ldots, 24$, the formula for the fractal dimension $z_{i,\tau}$ can be represented as:

$$z_{i,\tau} = \frac{\log\left(\frac{L_{i,\tau}}{a_{i,\tau}}\right)}{\log\left(\frac{d_{i,\tau}}{a_{i,\tau}}\right)} = \frac{\log\left(\nu - 1\right)}{\log\left(\frac{d_{i,\tau}(\nu-1)}{L_{i,\tau}}\right)} = \frac{\log(\nu - 1)}{\log(\nu - 1) + \log\left(\frac{d_{i,\tau}}{L_{i,\tau}}\right)}. \tag{9}$$

Note that $d_{i,\tau} \leq L_{i,\tau}$, where both $d_{i,\tau}$ and $L_{i,\tau}$ are positive real numbers. Therefore,

$$0 < \frac{d_{i,\tau}}{L_{i,\tau}} \leq 1,$$

that implies

$$\log\left(\frac{d_{i,\tau}}{L_{i,\tau}}\right) \leq 0.$$

The expression $\log\left(\frac{d_{i,\tau}}{L_{i,\tau}}\right)$ is zero if, and only if, the points $s^j_{i,\tau}$, $j = 1, \ldots, \nu$ are all aligned. In this case, the fractal dimension is exactly 1. When the ratio $\frac{d_{i,\tau}}{L_{i,\tau}}$ decreases from 1 to $\frac{1}{\sqrt{\nu-1}}$, then the fractal dimension of the time series increases up to 2. Even though the fractal dimension of a plane fractal never exceeds 2, the fractal dimension of a time series using the Katz's algorithm may exceed this value when the fraction $\frac{d_{i,\tau}}{L_{i,\tau}}$ is less than $\frac{1}{\sqrt{\nu-1}}$. However, the fractal dimension of a regular time series normally lies within the range $[1, 2]$.

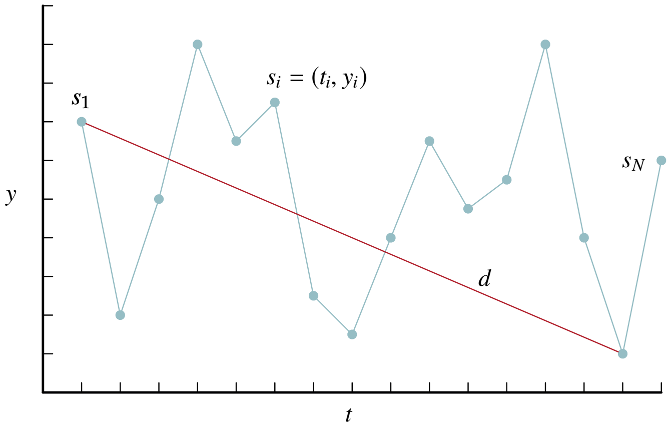


Figure 3. The diameter of a time series d is given by the distance between the first point and the point that provides the maximum distance.

With the fractal dimensions $z_{i,\tau}$ of the time series in matrix $\mathbf{Y}$ in Equation (3), we build a new matrix $\mathbf{Z}$ as:

$$\mathbf{Z} = \begin{array}{c|c|c|c} \text{sensor \#1} & \text{sensor \#2} & \cdots & \text{sensor \#24} \\ z_{1,1} & z_{1,2} & \cdots & z_{1,24} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,24} \\ \vdots & \vdots & \ddots & \vdots \\ z_{6400,1} & z_{6400,2} & \cdots & z_{6400,24} \end{array} \in \mathcal{M}_{6400\times 24}(\mathbb{R}). \tag{10}$$

Specifically:

- $z_{1,1}$ in matrix $\mathbf{Z}$ in Equation (10) is the fractal dimension of the time series

$$x_{1,1}^{(1)}, x_{2,1}^{(1)}, \ldots, x_{\nu,1}^{(1)};$$

- more generally, $z_{i,\tau}$ is the fractal dimension of the time series

$$x_{\alpha,\tau}^{(\beta)}, x_{\alpha+1,\tau}^{(\beta)}, \ldots, x_{\alpha+\nu,\tau}^{(\beta)},$$

where

$$\alpha = [(i \bmod 64) - 1] \cdot \nu + 1, \text{ if } i \bmod 64 \neq 0,$$
$$\alpha = 63\nu + 1, \text{ if } i \bmod 64 = 0,$$
$$\beta = [(i-1) \text{ div } 64] + 1,$$

and div and mod stand for the integer quotient and the remainder of an integer division, respectively.

3.3. Normalization and Classification Tools

Although matrix $\mathbf{Z}$ in Equation (10) is a matrix of elements generally between 1 and 2, the data is normalized by using column-wise scaling. This way, each column, and consequently each sensor, will have the same influence on the posterior analysis. Otherwise, the sensors closest to the source of the excitation and furthest from the structural damage could have a superior influence and make it difficult to detect the damage. Column-wise scaling is performed by subtracting the mean of each column to the elements on that column and dividing the same elements by the standard deviation of the column.

In this work, different classifiers have been used for the classification: k nearest neighbors (kNN) and support vector machines (SVM) with different kernels. In Sections 3.3.1 and 3.3.2 these methods will be briefly reviewed. Finally, it is important to note that 5-fold cross validation has been used to evaluate the classifier models.

3.3.1. k Nearest Neighbor

The k nearest neighbor (kNN) algorithm has been used since 1970. It is a classification algorithm that is used to make a prediction of a new observation based on the category of the k nearest neighbors. Two elements are key to this approach:

(i) the one and only parameter k; and
(ii) the distance measure [38].

The most commonly used distance measures in machine general are, in general, the Hamming distance, Euclidean distance, the Manhattan distance and the Minkowski distance. In this paper, the Euclidean distance is used.

3.3.2. Support Vector Machines (SVM)

SVM is a supervised machine learning algorithm that is used for classification purposes and it has been applied to a large variety of applications [39]. SVM are based on the simple idea of finding the hyperplane (or the decision boundary) that best divides the data into two classes.

Figure 4a shows the illustration of three separating hyperplanes out of many possible. The goal is to choose a hyperplane with the widest margin to separate both classes, see Figure 4b. In this context, the margin is defined as the smallest distance between any of the samples and the hyperplane. The data points closest to the separating hyperplane are called the support vectors. These points will determine how wide the margin is. Let us consider a two-classes example, a training data set $\mathbf{x}_1, \ldots, \mathbf{x}_N$, $N \in \mathbb{N}$ with corresponding binary target values $\{t_1, \ldots, t_N\} \subset \{-1, 1\}$, where one class is labeled as *red* (corresponding to a positive target value 1) and the other one as *blue* (corresponding to a negative target value -1). Commonly, the hyperplane is expressed in the following form:

$$h(\mathbf{x}) = \omega^\top \mathbf{x} + b$$

where ω is the weight vector and b is the bias term. The canonical hyperplane is used in this paper, among all the possible descriptions. The canonical hyperplane satisfies:

$$\begin{aligned} \omega^\top \mathbf{x}^{\mathrm{sv}}_{\mathrm{red}} + b &= 1, \\ \omega^\top \mathbf{x}^{\mathrm{sv}}_{\mathrm{blue}} + b &= -1, \end{aligned}$$

where $\mathbf{x}^{\mathrm{sv}}_{\mathrm{red}}$ and $\mathbf{x}^{\mathrm{sv}}_{\mathrm{blue}}$ represent the so-called support vectors (the closest samples with respect to the hyperplane) on the red and blue classes, respectively. The distance δ from the support vectors to the hyperplane is given by:

$$\delta\left(\mathbf{x}^{\mathrm{sv}}_{\{\mathrm{red,blue}\}}, h\right) = \frac{\left|\omega^\top \mathbf{x}^{\mathrm{sv}}_{\{\mathrm{red,blue}\}} + b\right|}{\|\omega\|} = \frac{1}{\|\omega\|}.$$

Since the margin is twice the distance from the support vectors to the hyperplane, the margin will be $\frac{2}{\|\omega\|}$. As it has been said, the goal is to maximize the margin $\frac{2}{\|\omega\|}$, which is equivalent to minimizing the inverse function $\frac{\|\omega\|}{2}$. This is also equivalent to minimizing

$$\min_{\omega, b} \frac{1}{2}\|\omega\|^2 \qquad \text{subject to} \qquad h(\mathbf{x}_i)t_i \geq 1,\ i = 1, \ldots, N.$$

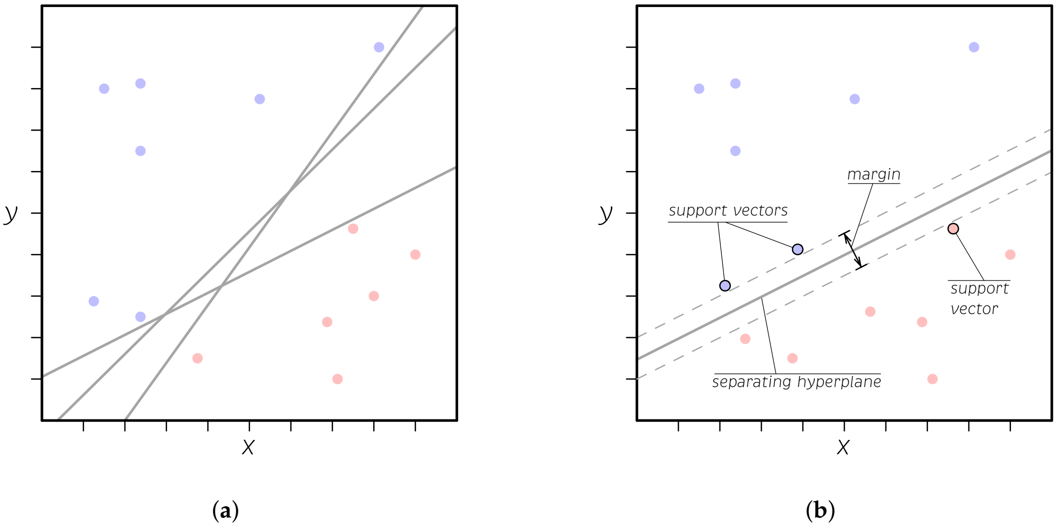


Figure 4. (**a**) There are two classes (blue and red), which are separated by three hyperplanes (in this case lines) out of many possible; (**b**) optimal hyperplane that maximizes the margin between classes.

In order to find the extreme values of a function with multiple constraints, one possible approach is to use the Lagrange multipliers. With this approach, the previous minimization problem is re-expressed as:

$$\min_{\omega,b,\alpha_i} \mathcal{L}(\omega,b;\alpha_i) = \min_{\omega,b,\alpha_i} \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{N} \alpha_i \left[(\omega^\top \mathbf{x}_i + b)t_i - 1\right] \tag{11}$$

where $\alpha_i,\ i = 1,\ldots,N$ are the Lagrange multipliers. To find the extreme values, the partial derivatives with respect to ω and b are computed and equated to zero:

$$\frac{\partial \mathcal{L}(\omega,b;\alpha_i)}{\partial \omega} = \omega - \sum_{i=1}^{N} \alpha_i t_i \mathbf{x}_i = 0 \Rightarrow \omega = \sum_{i=1}^{N} \alpha_i t_i \mathbf{x}_i \tag{12}$$

$$\frac{\partial \mathcal{L}(\omega,b;\alpha_i)}{\partial b} = -\sum_{i=1}^{N} \alpha_i t_i = 0 \tag{13}$$

Equation (12) shows that the weight vector ω is a linear combination of the training data set. Replacing Equations (12) and (13) into Equation (11), the minimization problem is uniquely expressed in terms of $\alpha_i, \mathbf{x}_i$ and t_i:

$$\min_{\alpha_i} \left[\frac{1}{2} \left(\sum_{i=1}^{N} \alpha_i t_i \mathbf{x}_i \right)^\top \left(\sum_{i=1}^{N} \alpha_i t_i \mathbf{x}_i \right) - \sum_{i=1}^{N} \alpha_i t_i \left(\sum_{j=1}^{N} \alpha_j t_j \mathbf{x}_j \right)^\top \mathbf{x}_i - b \sum_{i=1}^{N} \alpha_i t_i + \sum_{i=1}^{N} \alpha_i \right]. \tag{14}$$

After some simple manipulations, Equation (14) is now expressed as:

$$\min_{\alpha_i} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \mathbf{x}_i^\top \mathbf{x}_j \right] \tag{15}$$

As it can be clearly seen, the optimization problem depends only on the dot product of pairs of training data. However, frequently the data are not linearly separable. Therefore, the margin constraint cannot be satisfied for any ω and b. One possible solution is to allow some data points to violate the margin constraints (soft margin), but it is needed to assign them a cost. In this case, a penalty parameter C (box constraint) has to be considered to control the maximum penalty imposed on margin-violating observations, as well as slack variables ε_i that controls the width of the margin. For the case of a linear kernel, dealing with a nonlinearly separable case can be generalized as:

$$\min_{\omega,b,\varepsilon_i} \left[\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N} \varepsilon_i \right] \quad \text{subject to} \quad \begin{cases} h(\mathbf{x}_i)t_i \geq 1 - \varepsilon_i,\ i = 1,\ldots,N; \\ \varepsilon_i \geq 0, i = 1,\ldots,N. \end{cases} \tag{16}$$

The constrained minimization problem in Equation (16) can be rewritten, using Lagrange multipliers, as:

$$\min_{\alpha_i} \left[\sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j t_i t_j \mathbf{x}_i^\top \mathbf{x}_j \right] \quad \text{subject to} \quad \begin{cases} \sum_{i=1}^{N} \alpha_i t_i = 0; \\ 0 \leq \alpha_i \leq C,\ i = 1,\ldots,N. \end{cases} \tag{17}$$

In many cases, even with a soft margin, the space is not linearly separable. In these cases, a transformation ϕ is used to transform the original training data to another space. As it was mentioned before, the optimization depends only in dot products. Therefore, the transformation ϕ is not needed. Instead, only the dot product

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)\phi(\mathbf{x}_j)$$

is needed, renamed as the kernel function. In this work, we will used two kernel functions, quadratic kernel K_q and Gaussian kernel K_G, defined as:

$$K_\mathrm{q}(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \frac{1}{\gamma^2}\mathbf{x}_i^\top \mathbf{x}_j\right),$$

$$K_\mathrm{G}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\gamma^2}\right),$$

where γ is the so-called kernel scale.

4. Results

In this section, the results are organized as follows. First, the evaluation metrics used to assess the classification models are introduced and explained in Section 4.1. As it has been detailed in Sections 3.3.1–3.3.2, the classification models used in this work are *k*NN, quadratic SVM and Gaussian SVM. The results of the present approach using the fractal dimension to build the feature vector and *k*NN, quadratic SVM and Gaussian SVM are presented in Sections 4.2–4.4, respectively.

Figure 5 presents a flowchart summarizing the proposed damage diagnosis strategy. In a nutshell, the fractal dimension is computed and normalized to each time series (per sensor) of the baseline data and machine learning models are trained. Finally, when new data from a structure to be diagnosed comes in, its fractal dimension is computed, normalized and finally the already trained *k*NN or SVM (quadratic or Gaussian) model is applied for the structural state classification.

4.1. Evaluation Metrics

Before the results are presented, in terms of multiclass confusion matrices, it is important to clearly describe the evaluation metrics that are used to assess the performance of each model. One of the most used metrics is the overall accuracy, which is defined as the number of correct predictions out of the total number of predictions. However, the overall accuracy alone does not always tell if a model performs satisfactorily or unsatisfactorily, especially if the test data are comprised of imbalanced classes. However, even in the case of balanced classes, with the information provided by the overall accuracy, it is not possible to completely know how to improve the model. The metrics used in this work are accuracy, precision, recall, F_1-score and specificity. These metrics, for both the binary classification and multiclass classification problem will be defined shortly in the next paragraphs.

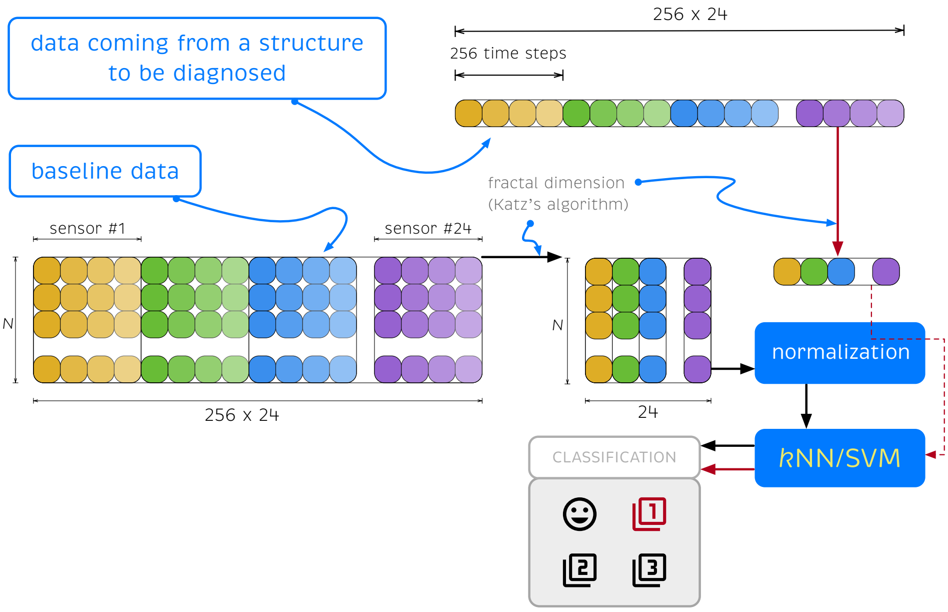


Figure 5. Flowchart summarizing the proposed damage diagnose strategy.

Consider categorical labels when $n \in \mathbb{N}$ observations $x_1, \ldots, x_n$ have to be assigned into predefined classes $C_1, \ldots, C_\ell$, $\ell \in \mathbb{N}$. In a binary classification problem, each observation x_i is to be classified into one, and only one, of two nonoverlapping classes (C_1 and C_2, or *positive* and *negative*). However, in a multiclass classification problem, the input x_i is to be classified into one, and only one, of ℓ nonoverlapping classes.

4.1.1. Metrics for a Binary Classification Problem

A confusion matrix is a table or matrix that summarizes the prediction results of a classification problem. It is not a metric itself but it helps to visually understand the metrics and types of errors the model is making. Table 1 represents the confusion matrix for the case of a binary classification problem, where two classes have been considered: *positive* and *negative*. The observations are distributed in two rows and two columns. The rows represent the actual classes, while the columns represent the predicted classes. The observations in the diagonal represent the correct decisions, while the elements in the antidiagonal represent the misclassifications.

Table 1. Confusion matrix of a binary classification problem.

		Predicted Class	
		Positive	**Negative**
Actual class	**Positive**	True positive (tp)	False negative (fn)
	Negative	False positive (fp)	True negative (tn)

More precisely, the four elements in a confusion matrix of a binary classification problem are:

- True positive (tp): the number of positive observations predicted as positive;
- True negative (tn): the number of negative observations predicted as negative;

- False positive (fp): the number of negative observations wrongly predicted as positive;
- False negative (fn): the number of positive observations wrongly predicted as negative.

The five metrics for the binary classification problem are then defined in Table 2 in terms of the elements of the confusion matrix. The F_1 score is a particular case of the F_β score defined in [40] when $\beta = 1$.

Table 2. Metrics for the evaluation of a binary classification problem.

Metric	Formula
accuracy	$\mathrm{acc} = \dfrac{\mathrm{tp} + \mathrm{tn}}{\mathrm{tp} + \mathrm{fn} + \mathrm{fp} + \mathrm{tn}}$
precision	$\mathrm{ppv} = \dfrac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}}$
recall	$\mathrm{tpr} = \dfrac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}}$
F_1 score	$F_1 = \dfrac{2 \cdot \mathrm{ppv} \cdot \mathrm{tpr}}{\mathrm{ppv} + \mathrm{tpr}}$
specificity	$\mathrm{tnr} = \dfrac{\mathrm{tn}}{\mathrm{tn} + \mathrm{fp}}$

4.1.2. Metrics for a Multiclass Classification Problem

Metrics for a multiclass classification problem are based on a generalization of the metrics in Table 2 for many classes $C_i,\ i = 1, \dots, \ell$ [41,42]. More precisely, with respect to the class C_i, we define:

- tp_i as the true positive for C_i, that is, the number of observations that belong to the class C_i that are correctly labeled as C_i;
- tn_i as the true negative for C_i, that is, the number of observations that do not belong to the class C_i that are not labeled as C_i;
- fp_i as the false positive for C_i, that is, the number of observations that do not belong to the class C_i that are wrongly labeled as C_i; and
- fn_i as the false negative for C_i, that is, the number of observations that belong to the class C_i that are not labeled as C_i.

Table 3 presents the metrics for the evaluation of a multiclass classification problem. Although the quality of the overall multiclass classification is usually assessed in two ways: (i) macroaveraging; and (ii) microaveraging, Table 3 only considers the macroaveraging case, where all classes are treated equally, instead of microaveraging, where bigger classes are favored.

Table 3. Metrics for the evaluation of multiclass classification problems, where ℓ is the number of classes.

Metric	Formula
average accuracy	$\overline{\mathrm{acc}} = \dfrac{1}{\ell} \sum_{i=1}^{\ell} \dfrac{\mathrm{tp}_i + \mathrm{tn}_i}{\mathrm{tp}_i + \mathrm{fn}_i + \mathrm{fp}_i + \mathrm{tn}_i}$
average precision	$\overline{\mathrm{ppv}} = \dfrac{1}{\ell} \sum_{i=1}^{\ell} \dfrac{\mathrm{tp}_i}{\mathrm{tp}_i + \mathrm{fp}_i}$
average recall	$\overline{\mathrm{tpr}} = \dfrac{1}{\ell} \sum_{i=1}^{\ell} \dfrac{\mathrm{tp}_i}{\mathrm{tp}_i + \mathrm{fn}_i}$
average F_1 score	$\overline{F_1} = \dfrac{2 \cdot \overline{\mathrm{ppv}} \cdot \overline{\mathrm{tpr}}}{\overline{\mathrm{ppv}} + \overline{\mathrm{tpr}}}$
average specificity	$\overline{\mathrm{tnr}} = \dfrac{1}{\ell} \sum_{i=1}^{\ell} \dfrac{\mathrm{tn}_i}{\mathrm{tn}_i + \mathrm{fp}_i}$

Finally, it is important to note that in the next subsections all the presented confusion matrices follow the next nomenclature. The rows represent the actual class and the columns represent the predicted class. Label 0 corresponds to the case when the structure is healthy; label 1 corresponds to the structure with a replica bar; label 2 corresponds to the structure with a 5 mm cracked bar; and label 3 corresponds to the structure with an unlocked bolt in the jacket.

4.2. Results of Fractal Dimension and kNN as Classification Method

As it has been said in Section 3.3.1, the one and only parameter of the kNN classifier is k, the number of neighbors. Table 4 shows the performance of the proposed approach using kNN as the classification method, in terms of the number of neighbors k. As described in Section 4.1.2, the metrics for the evaluation of this multiclass classification problem are the average accuracy, the average precision, the average recall, the average F_1 score and the average specificity. The best results for each metric have been highlighted in bold. The same results, as a function of the number of neighbors, are depicted in Figure 6. The case with the best performance corresponds to the case where the number of neighbors is $k = 20$. It can be observed that increasing further the number of neighbors does not increase the indicators' performance and only leads to a higher computational cost. Table 5 represents the confusion matrix for the best case ($k = 20$). In Table 4, the performance measures are presented using macroaveraging. However, in the confusion matrix in Table 5, precision and recall can be extracted for each class, separately. Similarly, Table 5 also presents the false negative rate (fnr)—defined as 1−tpr— and the false discovery rate (fdr)—defined as 1−ppv—. From this confusion matrix, it can be derived all the aforementioned metrics. In particular, it is noteworthy that an average accuracy of 96.9%, an average precision of 94.3% and an average specificity of 97.7% are obtained.

Table 4. Performance measures (per-unit) for the kNN method using different number of nearest neighbors (k). The cases with the best performance of each measure are highlighted in bold.

k	$\overline{\textbf{acc}}$	$\overline{\textbf{ppv}}$	$\overline{\textbf{tpr}}$	$\overline{\textbf{F}}_1$	$\overline{\textbf{tnr}}$
1	0.953	0.899	0.898	0.898	0.968
5	0.964	0.929	0.917	0.922	0.974
10	0.967	0.937	0.922	0.929	0.976
15	0.968	0.940	0.925	0.931	**0.977**
20	**0.969**	0.943	**0.926**	**0.933**	**0.977**
25	**0.969**	0.942	0.925	**0.933**	**0.977**
30	**0.969**	0.943	0.925	**0.933**	**0.977**
35	**0.969**	0.943	0.924	0.932	**0.977**
40	0.968	**0.944**	0.924	0.932	**0.977**
45	0.967	0.941	0.920	0.929	0.976
50	0.967	0.941	0.919	0.928	0.975

Table 5. Confusion matrix for the kNN algorithm when $k = 20$.

	0	**1**	**2**	**3**	**tpr**	**fnr**
0	2527	4	7	22	99%	1%
1	33	1186	60	1	93%	7%
2	6	68	1195	11	93%	7%
3	165	6	13	1096	86%	14%
ppv	93%	94%	94%	97%		
fdr	7%	6%	6%	3%		

4.3. Results of Fractal Dimension and Quadratic SVM as Classification Method

Table 6 summarizes the performance, using macroaveraging, of the proposed approach using quadratic SVM as the classification method, in terms of the box constraint C and the kernel scale γ

hyperparameters. More precisely, we combine the box constraint for $C = 5$, 10, 20, 30, 40 and 50 and the kernel scale for $\gamma = 0.1$, 0.2, 0.5, 1, 2, 5, 10, 15, 20, 30 and 50. The best results for each metric have been highlighted in bold. The same results, for a box constraint $C = 30$ and as a function of the kernel scale γ, are depicted in Figure 7. The case with the best performance corresponds to the case where the box constraint is $C = 30$ and the kernel scale is $\gamma = 1$. Table 7 represents the confusion matrix for this case, where it is worth remarking that an average accuracy of 98.4%, an average precision of 96.5% and an average specificity of 98.9% are obtained.

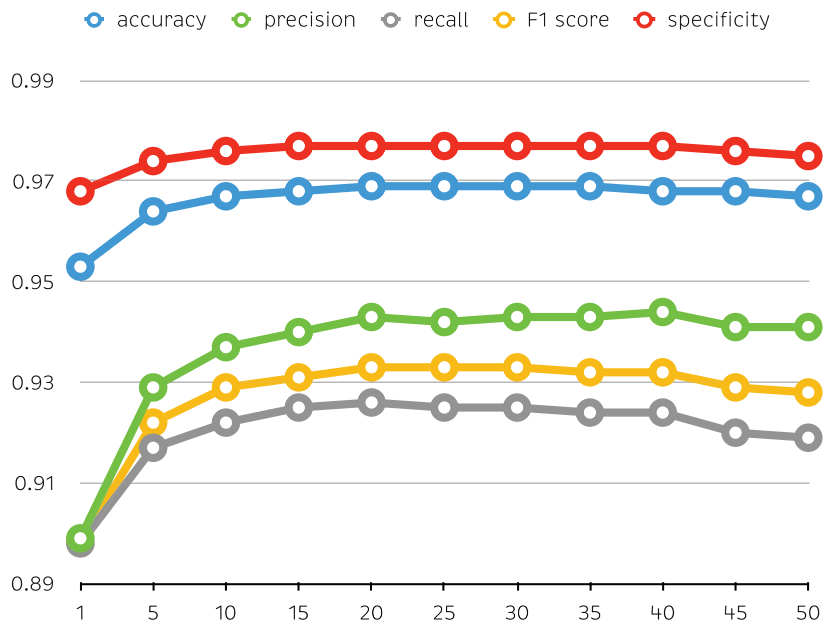


Figure 6. Performance measures (per-unit) corresponding to the *k*NN strategy for the multiclass classification problem with respect to the number of neighbors *k* (horizontal axis).

Table 6. Performance measures (per-unit) corresponding to the quadratic SVM strategy for the multiclass classification problem using different box constraints (C) and different kernel scales (γ). The cases with the best performance of each measure are highlighted in bold.

C	γ	$\overline{\mathbf{acc}}$	$\overline{\mathbf{ppv}}$	$\overline{\mathbf{tpr}}$	$\overline{\mathbf{F}}_1$	$\overline{\mathbf{tnr}}$
5	0.1	0.971	0.935	0.937	0.936	0.981
	0.2	0.981	0.958	0.956	0.957	0.987
	0.5	0.982	0.961	0.959	0.960	0.988
	1	0.981	0.959	0.957	0.958	0.987
	2	0.980	0.958	0.953	0.956	0.986
	5	0.937	0.933	0.844	0.876	0.949
	10	0.924	0.928	0.809	0.850	0.937
	15	0.922	0.924	0.805	0.846	0.935
	20	0.919	0.918	0.798	0.838	0.933
	30	0.893	0.901	0.732	0.777	0.911
	50	0.868	0.887	0.669	0.720	0.890
10	0.1	0.971	0.933	0.935	0.934	0.981
	0.2	0.980	0.956	0.955	0.955	0.987
	0.5	0.982	0.961	0.960	0.960	0.988
	1	0.982	0.961	0.959	0.960	0.988
	2	0.982	0.962	0.959	0.960	0.988
	5	0.963	0.940	0.910	0.922	0.972
	10	0.924	0.928	0.809	0.850	0.937
	15	0.923	0.926	0.808	0.849	0.936
	20	0.922	0.924	0.805	0.846	0.935
	30	0.918	0.916	0.796	0.837	0.933
	50	0.888	0.898	0.720	0.766	0.907

Table 6. *Cont.*

C	γ	$\overline{\mathbf{acc}}$	$\overline{\mathbf{ppv}}$	$\overline{\mathbf{tpr}}$	$\overline{\mathbf{F}}_1$	$\overline{\mathbf{tnr}}$
20	0.1	0.961	0.915	0.912	0.912	0.974
	0.2	0.978	0.951	0.950	0.951	0.951
	0.5	0.982	0.961	0.959	0.960	0.988
	1	**0.984**	0.964	0.962	0.963	**0.989**
	2	0.982	0.962	0.959	0.961	0.988
	5	0.974	0.951	0.939	0.944	0.982
	10	0.926	0.929	0.814	0.854	0.938
	15	0.924	0.927	0.809	0.849	0.937
	20	0.923	0.925	0.808	0.848	0.936
	30	0.921	0.922	0.804	0.844	0.935
	50	0.906	0.908	0.766	0.810	0.923
30	0.1	0.963	0.917	0.919	0.916	0.976
	0.2	0.976	0.947	0.946	0.947	0.984
	0.5	0.983	0.961	0.960	0.960	0.988
	1	**0.984**	**0.966**	**0.964**	**0.965**	**0.989**
	2	0.983	0.962	0.960	0.961	0.988
	5	0.978	0.955	0.948	0.951	0.984
	10	0.934	0.931	0.834	0.869	0.945
	15	0.924	0.928	0.810	0.850	0.937
	20	0.923	0.926	0.808	0.849	0.936
	30	0.923	0.924	0.807	0.847	0.936
	50	0.918	0.916	0.797	0.837	0.933
40	0.1	0.963	0.917	0.919	0.916	0.976
	0.2	0.976	0.947	0.946	0.946	0.984
	0.5	0.982	0.961	0.959	0.960	0.988
	1	**0.984**	0.965	0.963	0.964	**0.989**
	2	0.983	0.962	0.961	0.962	**0.989**
	5	0.980	0.958	0.952	0.955	0.986
	10	0.944	0.932	0.860	0.886	0.954
	15	0.924	0.927	0.810	0.850	0.937
	20	0.924	0.927	0.809	0.849	0.937
	30	0.923	0.924	0.807	0.847	0.936
	50	0.919	0.918	0.799	0.839	0.934
50	0.1	0.963	0.917	0.919	0.916	0.976
	0.2	0.974	0.941	0.941	0.941	0.982
	0.5	0.982	0.959	0.958	0.958	0.988
	1	**0.984**	0.965	0.963	0.964	**0.989**
	2	0.983	0.963	0.961	0.962	**0.989**
	5	0.980	0.958	0.953	0.955	0.986
	10	0.953	0.935	0.883	0.903	0.963
	15	0.925	0.927	0.812	0.852	0.938
	20	0.924	0.927	0.809	0.849	0.937
	30	0.923	0.925	0.807	0.848	0.936
	50	0.920	0.919	0.801	0.841	0.934

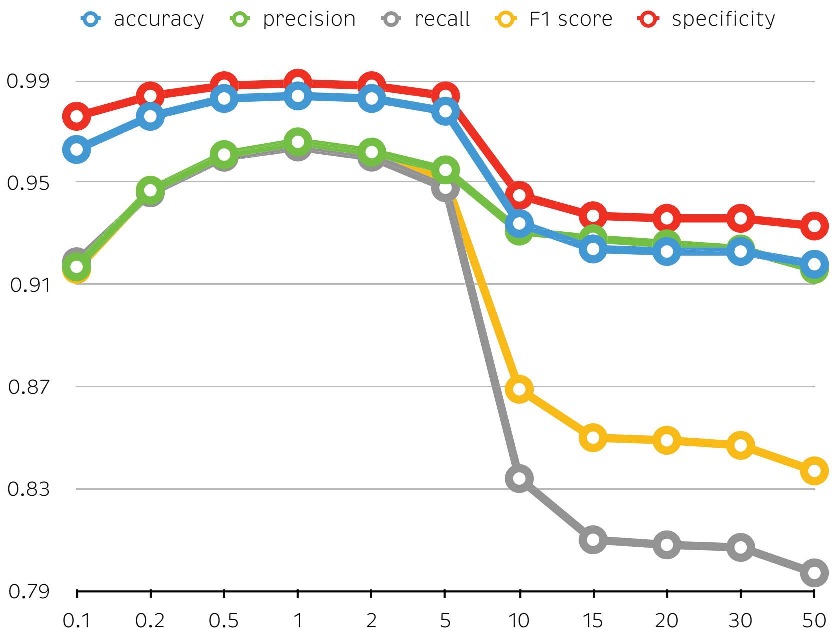


Figure 7. Performance measures (per-unit) corresponding to the quadratic SVM strategy for the multiclass classification problem for a box constraint $C = 30$ and with respect to the kernel scale γ (horizontal axis).

Table 7. Confusion matrix for the quadratic SVM algorithm for the case $C = 30$ (box constraint) and $\gamma = 1$ (kernel scale).

	0	**1**	**2**	**3**	**tpr**	**fnr**
0	2531	7	8	15	99%	1%
1	5	1214	61		95%	5%
2	11	40	1223	6	96%	4%
3	41	10	9	1220	95%	5%
ppv	98%	96%	94%	98%		
fdr	2%	4%	6%	2%		

4.4. Results of Fractal Dimension and Gaussian SVM as Classification Method

As in Section 4.3, Table 8 summarizes the performance of the proposed approach using Gaussian SVM as the classification method, in terms of both the box constraint C and the kernel scale γ. More precisely, we combine the box constraint for $C = 5$, 10, 20, 30, 40 and 50 and the kernel scale for $\gamma = 0.1$, 0.2, 0.5, 1, 2, 5, 10, 15, 20, 30 and 50. The best results for each metric have been highlighted in bold. The same results, for a box constraint $C = 50$ and as a function of the kernel scale γ, are depicted in Figure 8. The case with the best performance corresponds to the case where the box constraint is $C = 50$ and the kernel scale is $\gamma = 1$. Table 9 represents the confusion matrix for this case. From the confusion matrix, it is worth remarking that an average accuracy of 98.7%, an average precision of 97.3% and an average specificity of 99.1% are obtained.

Table 8. Performance measures (per-unit) corresponding to the Gaussian SVM strategy for the multiclass classification problem using different box constraints (C) and different kernel scales (γ). The cases with the best performance of each measure are highlighted in bold.

C	γ	$\overline{\mathbf{acc}}$	$\overline{\mathbf{ppv}}$	$\overline{\mathbf{tpr}}$	$\overline{\mathbf{F}}_1$	$\overline{\mathbf{tnr}}$
5	0.1	0.757	0.734	0.396	0.397	0.800
	0.2	0.833	0.833	0.586	0.634	0.863
	0.5	0.939	0.921	0.849	0.876	0.951
	1	0.983	0.965	0.960	0.963	0.988
	2	0.978	0.959	0.946	0.952	0.984
	5	0.925	0.930	0.813	0.853	0.938
	10	0.924	0.929	0.810	0.851	0.937
	15	0.922	0.926	0.806	0.847	0.936
	20	0.919	0.919	0.798	0.839	0.933
	30	0.893	0.901	0.733	0.778	0.912
	50	0.868	0.886	0.669	0.720	0.890
10	0.1	0.756	0.730	0.395	0.396	0.800
	0.2	0.832	0.828	0.584	0.631	0.862
	0.5	0.940	0.923	0.851	0.879	0.952
	1	0.985	0.969	0.965	0.967	0.990
	2	0.977	0.956	0.953	0.954	0.985
	5	0.944	0.936	0.859	0.887	0.954
	10	0.924	0.928	0.810	0.850	0.937
	15	0.924	0.928	0.809	0.850	0.937
	20	0.922	0.925	0.806	0.846	0.935
	30	0.919	0.917	0.797	0.837	0.933
	50	0.888	0.898	0.721	0.767	0.908
20	0.1	0.756	0.730	0.395	0.396	0.800
	0.2	0.831	0.822	0.581	0.627	0.627
	0.5	0.940	0.922	0.852	0.879	0.952
	1	0.986	0.970	0.967	0.968	0.990
	2	0.984	0.967	0.961	0.963	0.988
	5	0.969	0.948	0.922	0.932	0.976
	10	0.925	0.930	0.812	0.853	0.938
	15	0.924	0.928	0.809	0.850	0.937
	20	0.924	0.927	0.809	0.849	0.937
	30	0.921	0.922	0.804	0.844	0.935
	50	0.909	0.910	0.773	0.817	0.925
30	0.1	0.756	0.730	0.395	0.396	0.800
	0.2	0.830	0.821	0.581	0.627	0.862
	0.5	0.940	0.923	0.852	0.879	0.952
	1	**0.987**	0.972	0.969	**0.970**	**0.991**
	2	0.985	0.968	0.963	0.966	0.989
	5	0.976	0.956	0.942	0.948	0.982
	10	0.930	0.930	0.824	0.861	0.942
	15	0.924	0.928	0.809	0.850	0.937
	20	0.924	0.928	0.809	0.850	0.937
	30	0.923	0.924	0.807	0.847	0.936
	50	0.918	0.916	0.796	0.837	0.933
40	0.1	0.756	0.730	0.395	0.396	0.800
	0.2	0.830	0.819	0.580	0.626	0.861
	0.5	0.940	0.923	0.852	0.879	0.952
	1	**0.987**	0.972	**0.970**	**0.971**	**0.991**
	2	0.985	0.969	0.965	0.967	0.990
	5	0.978	0.957	0.946	0.951	0.984
	10	0.936	0.931	0.841	0.873	0.948
	15	0.924	0.928	0.811	0.851	0.937
	20	0.924	0.927	0.809	0.849	0.937
	30	0.923	0.925	0.808	0.848	0.936
	50	0.919	0.918	0.799	0.839	0.934

Table 8. *Cont.*

C	γ	$\overline{\textbf{acc}}$	$\overline{\textbf{ppv}}$	$\overline{\textbf{tpr}}$	$\overline{\textbf{F}}_1$	$\overline{\textbf{tnr}}$
50	0.1	0.756	0.730	0.395	0.396	0.800
	0.2	0.830	0.818	0.579	0.625	0.861
	0.5	0.940	0.922	0.852	0.878	0.952
	1	**0.987**	**0.973**	**0.970**	**0.971**	**0.991**
	2	0.985	0.969	0.965	0.967	0.990
	5	0.979	0.959	0.95	0.954	0.985
	10	0.946	0.933	0.865	0.890	0.956
	15	0.925	0.928	0.812	0.852	0.938
	20	0.924	0.927	0.809	0.849	0.937
	30	0.923	0.926	0.808	0.848	0.936
	50	0.920	0.920	0.802	0.842	0.934

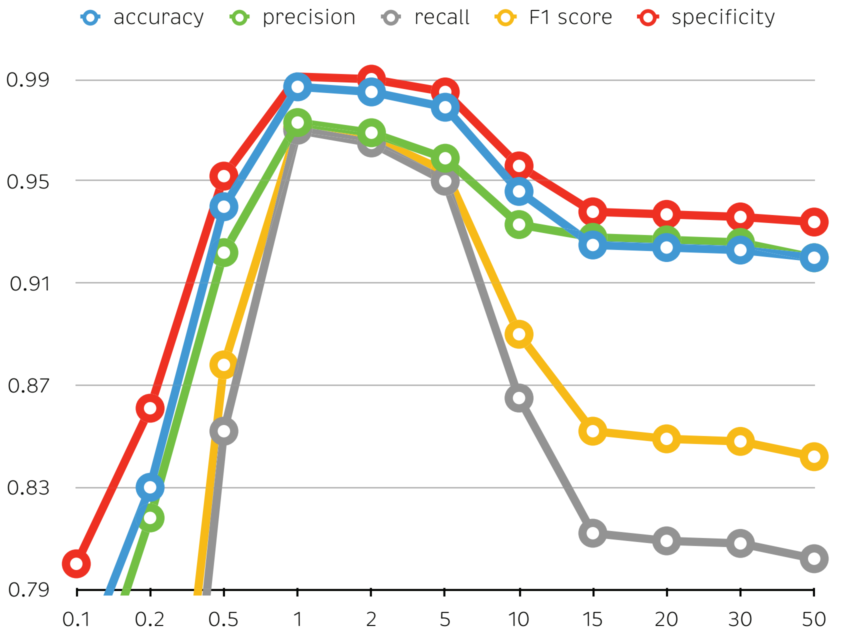


Figure 8. Performance measures (per-unit) corresponding to the Gaussian SVM strategy for the multiclass classification problem for a box constraint $C = 50$ and with respect to the kernel scale γ (horizontal axis).

Table 9. Confusion matrix for the Gaussian SVM algorithm for the case $C = 50$ (box constraint) and $\gamma = 1$ (kernel scale).

	0	**1**	**2**	**3**	**tpr**	**fnr**
0	2542	2	1	15	99%	1%
1	7	1231	40	2	96%	4%
2	1	45	1230	4	96%	4%
3	36	6	3	1235	96%	4%
ppv	98%	96%	97%	98%		
fdr	2%	4%	3%	2%		

4.5. Brief Discussion

Sections 4.2–4.4 present an optimization of the model hyperparameters for the *k*NN, quadratic SVM and Gaussian SVM, respectively. In each subsection, the confusion matrix for the best (optimized) model is presented. In this subsection, the best models are compared among them. That is, a comparison among the *k*NN, quadratic SVM and Gaussian SVM methodologies is given. In particular, Figure 9 shows the accuracy, precision, recall, F_1 score and specificity measures for the best *k*NN, quadratic SVM and Gaussian SVM models. It is noteworthy that the Gaussian SVM accomplishes the highest performance for all the indicators. Thus, it is the recommended approach to be employed with the proposed SHM strategy. Finally, it is also important to note that the quadratic SVM has a close performance to the Gaussian SVM but the *k*NN falls far behind in all the indicators in general, and more markedly for the recall and F_1 score measures. Therefore, its use is inadvisable. As a final remark, the performance of the Gaussian SVM over the quadratic SVM may depend on the nature of the data or even on how this data is preprocessed and what features are extracted. In this sense, the exceeding performance of the Gaussian SVM has been reported in the literature as a machine learning model for the prediction of the viscosity of nanofluids [43] or, in the field of fault diagnosis, to get the operation status of a wind turbine [44].

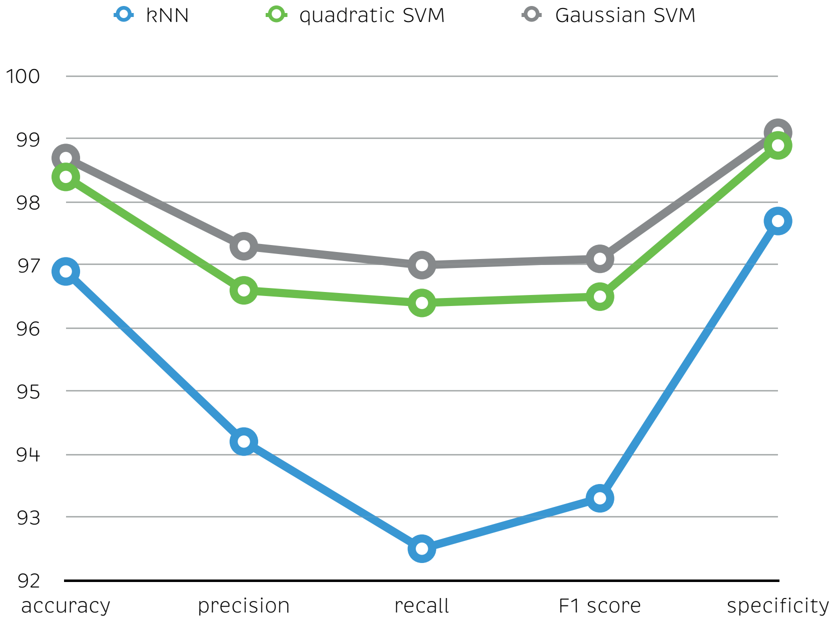


Figure 9. Performance measures (percentage) comparison among the different classifiers.

5. Conclusions

In this work a proof-of-concept damage diagnosis strategy that can be deployed online and during the WT service has been stated. This main contribution of the paper is accomplished by using only the vibration-response accelerometer signals instead of the standard approach based on guided waves. Furthermore, the methodology is based on machine learning techniques. In this regard, the second main contribution of this work is to introduce the FD as a suitable feature to detect and classify different damage scenarios inspired by the physical insight that the different fractal structures of the accelerometer signals should be capable to discriminate different types of damage. Three supervised machine learning classifiers have been studied and optimized for the specific problem. Finally, the proposed methodology has been validated in an experimental laboratory test bed where for the best selected model (Gaussian SVM with box constraint $C = 50$ and kernel scale $\gamma = 1$) all the studied measures (average accuracy, average precision, average recall, average F_1 score and average specificity) have attained values higher than 97%. These results encourage future work in this area of research to develop further this proof-of-concept. More tests including changing the damage location and taking into account and dealing with variable environmental operating conditions, including waves, will be the focus of future work.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially funded by the Spanish Agencia Estatal de Investigación (AEI)-Ministerio de Economía, Industria y Competitividad (MINECO), and the Fondo Europeo de Desarrollo Regional (FEDER) through the research project DPI2017-82930-C2-1-R; and by the Generalitat de Catalunya through the research project 2017 SGR 388.

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.

Abbreviations

The following abbreviations are used in this manuscript:

acc	accuracy
FD	fractal dimension
fdr	false discovery rate
fn	false negative
fnr	false negative rate
fp	false positive
GW	gigawatts
*k*NN	*k* nearest neighbor
ppv	precision
SHM	structural health monitoring
SVM	support vector machine
tn	true negative
tnr	specificity
tp	true positive
tpr	recall
WT	wind turbine

References

1. Na, W.S.; Baek, J. A review of the piezoelectric electromechanical impedance based structural health monitoring technique for engineering structures. *Sensors* **2018**, *18*, 1307. [CrossRef] [PubMed]
2. Chen, H.P. *Structural Health Monitoring of Large Civil Engineering Structures*; John Wiley & Sons: Hoboken, NJ, USA, 2018.
3. Shin, H.J.; Lee, J.R. Development of a long-range multi-area scanning ultrasonic propagation imaging system built into a hangar and its application on an actual aircraft. *Struct. Health Monit.* **2017**, *16*, 97–111. [CrossRef]
4. Ozbek, M.; Meng, F.; Rixen, D.J. Challenges in testing and monitoring the in-operation vibration characteristics of wind turbines. *Mech. Syst. Signal Process.* **2013**, *41*, 649–666. [CrossRef]
5. Outlook, I.O.W. *World Energy Outlook Series*; International Energy Agency: Paris, France, 2019.
6. Martinez-Luengo, M.; Shafiee, M. Guidelines and Cost-Benefit Analysis of the Structural Health Monitoring Implementation in Offshore Wind Turbine Support Structures. *Energies* **2019**, *12*, 1176. [CrossRef]
7. Arnold, P.; Moll, J.; Mälzer, M.; Krozer, V.; Pozdniakov, D.; Salman, R.; Rediske, S.; Scholz, M.; Friedmann, H.; Nuber, A. Radar-based structural health monitoring of wind turbine blades: The case of damage localization. *Wind Energy* **2018**, *21*, 676–680. [CrossRef]
8. Nguyen, T.C.; Huynh, T.C.; Yi, J.H.; Kim, J.T. Hybrid bolt-loosening detection in wind turbine tower structures by vibration and impedance responses. *Wind Struct* **2017**, *24*, 385–403. [CrossRef]
9. Mieloszyk, M.; Ostachowicz, W. An application of Structural Health Monitoring system based on FBG sensors to offshore wind turbine support structure model. *Mar. Struct.* **2017**, *51*, 65–86. [CrossRef]
10. Li, M.; Kefal, A.; Oterkus, E.; Oterkus, S. Structural health monitoring of an offshore wind turbine tower using iFEM methodology. *Ocean. Eng.* **2020**, *204*, 107291. [CrossRef]
11. Zhao, X.; Lang, Z. Baseline model based structural health monitoring method under varying environment. *Renew. Energy* **2019**, *138*, 1166–1175. [CrossRef]
12. Kim, H.C.; Kim, M.H.; Choe, D.E. Structural health monitoring of towers and blades for floating offshore wind turbines using operational modal analysis and modal properties with numerical-sensor signals. *Ocean. Eng.* **2019**, *188*, 106226. [CrossRef]

13. Stutzmann, J.; Ziegler, L.; Muskulus, M. Fatigue crack detection for lifetime extension of monopile-based offshore wind turbines. *Energy Procedia* **2017**, *137*, 143–151. [CrossRef]
14. Martinez-Luengo, M.; Kolios, A.; Wang, L. Structural health monitoring of offshore wind turbines: A review through the Statistical Pattern Recognition Paradigm. *Renew. Sustain. Energy Rev.* **2016**, *64*, 91–105. [CrossRef]
15. Weijtjens, W.; Verbelen, T.; De Sitter, G.; Devriendt, C. Foundation structural health monitoring of an offshore wind turbine—A full-scale case study. *Struct. Health Monit.* **2016**, *15*, 389–402. [CrossRef]
16. Gomez, H.C.; Gur, T.; Dolan, D. Structural condition assessment of offshore wind turbine monopile foundations using vibration monitoring data. Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2013. *Int. Soc. Opt. Photonics* **2013**, *8694*, 86940B.
17. Zugasti Uriguen, E. Design and Validation of a Methodology for Wind Energy Structures Health Monitoring. Ph.D. Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 16 January 2014.
18. Spanos, N.A.; Sakellariou, J.S.; Fassois, S.D. Vibration-response-only statistical time series structural health monitoring methods: A comprehensive assessment via a scale jacket structure. *Struct. Health Monit.* **2019**, *19*, 736–750. [CrossRef]
19. Vidal, Y.; Aquino, G.; Pozo, F.; Gutiérrez-Arias, J.E.M. Structural Health Monitoring for Jacket-Type Offshore Wind Turbines: Experimental Proof of Concept. *Sensors* **2020**, *20*, 1835. [CrossRef]
20. Puruncajas, B.; Vidal, Y.; Tutivén, C. Vibration-Response-Only Structural Health Monitoring for Offshore Wind Turbine Jacket Foundations via Convolutional Neural Networks. *Sensors* **2020**, *20*, 3429. [CrossRef]
21. Ruiz, M.; Mujica, L.E.; Alferez, S.; Acho, L.; Tutiven, C.; Vidal, Y.; Rodellar, J.; Pozo, F. Wind turbine fault detection and classification by means of image texture analysis. *Mech. Syst. Signal Process.* **2018**, *107*, 149–167. [CrossRef]
22. Sarkar, M.; Leong, T.Y. Characterization of medical time series using fuzzy similarity-based fractal dimensions. *Artif. Intell. Med.* **2003**, *27*, 201–222. [CrossRef]
23. El-Kishky, A. Assessing entropy and fractal dimensions as discriminants of seizures in EEG time series. In Proceedings of the 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), Montreal, QC, Canada, 2–5 July 2012; pp. 92–96.
24. Kolář, R.; Jan, J. Detection of glaucomatous eye via color fundus images using fractal dimensions. *Radioengineering* **2008**, *17*, 109–114.
25. Rezaie, A.; Mauron, A.J.; Beyer, K. Sensitivity analysis of fractal dimensions of crack maps on concrete and masonry walls. *Autom. Constr.* **2020**, *117*, 103258. [CrossRef]
26. Wen, W.; Fan, Z.; Karg, D.; Cheng, W. Rolling element bearing fault diagnosis based on multiscale general fractal features. *Shock Vib.* **2015**, *2015*, 167902. [CrossRef]
27. Vidal, Y.; Pozo, F.; Tutivén, C. Wind turbine multi-fault detection and classification based on SCADA data. *Energies* **2018**, *11*, 3018. [CrossRef]
28. Pozo, F.; Vidal, Y.; Serrahima, J.M. On real-time fault detection in wind turbines: Sensor selection algorithm and detection time reduction analysis. *Energies* **2016**, *9*, 520. [CrossRef]
29. Mandelbrot, B.B. *The Fractal Geometry of Nature*; WH freeman: New York, NY, USA, 1983; Volume 173,
30. Sevcik, C. A procedure to estimate the fractal dimension of waveforms. *arXiv* **2010**, arXiv:1003.5266.
31. Raghavendra, B.; Dutt, D.N. A note on fractal dimensions of biomedical waveforms. *Comput. Biol. Med.* **2009**, *39*, 1006–1012. [CrossRef]
32. Higuchi, T. Approach to an irregular time series on the basis of the fractal theory. *Phys. D Nonlinear Phenom.* **1988**, *31*, 277–283. [CrossRef]
33. Russ, J.C. Fractal dimension measurement of engineering surfaces. *Int. J. Mach. Tools Manuf.* **1998**, *38*, 567–571. [CrossRef]
34. Breslin, M.; Belward, J. Fractal dimensions for rainfall time series. *Math. Comput. Simul.* **1999**, *48*, 437–446. [CrossRef]
35. Bourke, P. *An Introduction to Fractals*; The University of Western of Australia: Perth, Australia, 1991; Volume 5.
36. Addison, P.S. *Fractals and Chaos: An Illustrated Course*; CRC Press: Boca Raton, FL, USA, 1997.
37. Katz, M.J. Fractals and the analysis of waveforms. *Comput. Biol. Med.* **1988**, *18*, 145–156. [CrossRef]
38. Mulak, P.; Talhar, N. Analysis of distance measures using k-nearest neighbor algorithm on kdd dataset. *Int. J. Sci. Res.* **2015**, *4*, 2101–2104.

39. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 112.
40. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
41. Krüger, F. Activity, Context, and Plan Recognition with Computational Causal Behaviour Models. Ph.D. Thesis, University of Rostock, Rostock, Germany, December 2016.
42. Hameed, N.; Hameed, F.; Shabut, A.; Khan, S.; Cirstea, S.; Hossain, A. An Intelligent Computer-Aided Scheme for Classifying Multiple Skin Lesions. *Computers* **2019**, *8*, 62. [CrossRef]
43. Shateri, M.; Sobhanigavgani, Z.; Alinasab, A.; Varamesh, A.; Hemmati-Sarapardeh, A.; Mosavi, A.; Shamshirband, S.S. Comparative Analysis of Machine Learning Models for Nanofluids Viscosity Assessment. *Nanomaterials* **2020**, *10*, 1767. [CrossRef] [PubMed]
44. Wu, Z.; Wang, X.; Jiang, B. Fault Diagnosis for Wind Turbines Based on ReliefF and eXtreme Gradient Boosting. *Appl. Sci.* **2020**, *10*, 3258. [CrossRef]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).