



# Convolutional Neural Network for Wind Turbine Failure Classification Based on SCADA Data

B. Puruncajas<sup>1,2</sup>, W. Alava<sup>1</sup>, Encalada-Dávila<sup>1</sup>, C. Tutivén<sup>1,2</sup> and Y. Vidal<sup>2,3</sup>

<sup>1</sup>Mechatronics Engineering  
Faculty of Mechanical Engineering and Production Science  
Escuela Superior Politécnica del Litoral  
Guayaquil, Ecuador  
Phone number:+593 967 583900, e-mail: [bpurunca@espol.edu.ec](mailto:bpurunca@espol.edu.ec)  
<sup>2</sup>Control, Modeling, Identification and Applications  
Department of Mathematics  
Escola d'Enginyeria de Barcelona Est  
Universitat Politècnica de Catalunya  
Campus Diagonal-Besós (CDB), 08019 Barcelona, Spain  
<sup>3</sup> Institute of Mathematics (IMTech)  
Universitat Politècnica de Catalunya  
Pau Gargallo 14, 08028 Barcelona, Spain

**Abstract.** As a renewable energy source and an alternative to fossil fuels, the wind power industry is growing rapidly. However, due to harsh weather conditions, wind turbines (WT) still face many failures that raise the price of energy produced and reduce the reliability of wind energy. Hence, the use of reliable monitoring and diagnostic systems of WTs is of great importance. Operation and maintenance expenses represent 30% of the total cost of large wind farms. The installation of offshore and remote wind farms has increased the need for efficient fault detection and condition monitoring systems. In this work, without using specific custom devices for monitoring conditions, but only increasing the sampling frequency in the sensors already available (in all commercial WT) of the supervisory control and data acquisition system (SCADA), data-driven multiple fault detection is performed, and a classification strategy is developed. The data is processed, and subsequently, using a convolutional neural network (CNN), six faults are classified and evaluated with different metrics. Finally, it should be noted that the classification speed allows the implementation of this strategy to monitor conditions online in real under-production WTs.

## Key words

Wind turbine, SCADA data, convolutional neural network, fault detection and classification, FAST.

## 1. Introduction

The growing demand for energy worldwide and the emission of polluting gases caused by conventional fossil fuel resources urges to increase renewable energy sources in the global energy balance. In recent years, the penetration of wind energy throughout the energy market is continuously growing. In 2019, wind power met 15% of the EU's electricity demand [1], and was the primary source of new capacity in the US, and Canada, and the second largest in China. Furthermore, installed capacity worldwide annual increase rate of

more than 10% in the last ten years [2]. To generate large amounts of electricity is necessary to meet the growing demand. Currently, the largest WT produced by Siemens Gamesa generates 14 MW of nominal power, it has a swept area diameter of 222 m and requires an average annual wind speed of 10 m/s [3]. However, this trend towards larger WTs has significantly increased the cost of repair and replacement of its parts.

Condition monitoring is crucial for wind power to be profitable and efficient, even more in offshore wind farms where working conditions are even more extreme than onshore (e.g., oxidation by saltwater, high tides, etc.). The early detection and classification of faults in WTs have become essential to increase their competitiveness since the prevention of faults and timely maintenance reduces the cost of repairs and the downtime of the affected turbine, which can last up to several weeks. Experience from other industries shows that condition monitoring can detect failures before they reach permanent damage, extends service lifetime, operates at initial capacity factors, and enables better planning and maintenance logistics. Failure detection systems are used to provide alarms about the deterioration of the state or failure of different components of a WT based on different types of information (parameters or signals) obtained by different types of sensors. Figure 1 shows that the electrical system, control systems, and blades/pitch have high failure rates.

In the literature, there are several works on WT flaw detection research. In [5] a Concurrent Convolution Neural Network (CeCNN) is proposed, the raw data is entered into the network without any prior knowledge. The characteristics are learned directly and adaptively for the diagnosis of bearing failures in WTs. The results show that the proposed method can extract discriminatory characteristics and classify the bearing data accurately under the disturbance of different rotational speeds, different loads, and random noise. In [6], a drone is used to inspect the blades of the WT through a

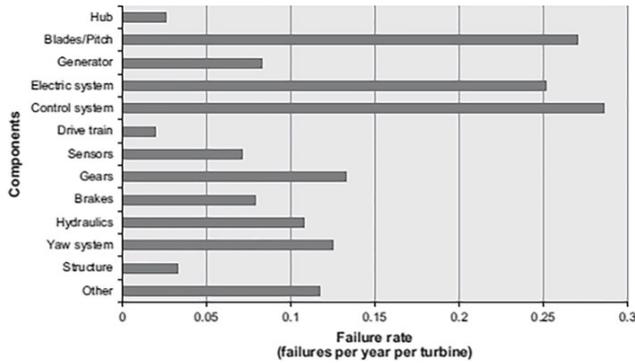


Fig. 1: Average failure rates for WT components. [4]

CNN, obtaining great results in the diagnosis of blade failures.

A SCADA - based condition monitoring system uses data that is already collected in the WT controller and is a cost-effective way to monitor for early warning of faults and performance problems [7]. The SCADA system generally records variable data in 10-minute averaged intervals. SCADA data shows a WT's overall health and can be leveraged to detect when turbine performance is degrading and identify if a fault is developing. Although SCADA data has not been specifically designed for health monitoring, extracting relevant information from it would result in quick implementation and modest setup costs [8]. Some previous studies have emphasized the use of SCADA data for WT condition monitoring. For example, in [9], an automated system based on an artificial neural network (ANN) is proposed. Specialized techniques that can be used to identify faults in the main components of a WT. In [10], a nonlinear state estimation technique is proposed to diagnose faults in WT's gearbox using SCADA data.

In particular, this work aims to detect and classify six types of WT failures (see table III). A CNN is used due to its excellent ability to obtain in-depth features, [11]. This paper is organized as follows: Section 2 shows the description of the data. Section 3 states the methodology used and data preprocessing. The results obtained are presented and discussed in Section 4. In Section 5 the conclusions and future work are drawn.

## 2. Data

The data is generated from a 5 MW WT conventional three-bladed simulation model with variable pitch control and variable speed upwind, whose characteristics are shown in table I. The aeroelastic simulation software FAST [12] is used to model the WT. In the simulation, noise blocks represent the measurement noise caused by the electrical noise of the WTs. A sampling period of 0.0125 s is used in the simulations (a characteristic value of the FAST simulation software). However, the data used for WT failure diagnosis is sampled in a sampling period of 1 s. SCADA data traditionally has a 10-minute sampling rate. Here, it is proposed to use conventional SCADA data with a high-frequency sampling of the sensors of 1 Hz.

To produce more realistic simulations of the WT, wind modeling data generated from the wind flow and stochastic turbulence software of NREL, TurbSim [13], is implemented. For this research, TurbSim simulations are performed under the following wind parameters: Kaimal turbulence model with 10 % intensity, at the height of the hub, with the wind moving logarithmically at an average speed of 18.2 m/s and with a stiffness factor of 0.01 m. Finally, using the given inputs and implementing a SCADA system, which contains several sensors shown in table II, in the FAST software, 260 simulations of 60 seconds each are performed. Each simulation creates a data set, for which a total of 260 data sets are generated. From the 260 simulations, 69 correspond to

TABLE I: Principal features of the WT.

Reference WT	Data
Rated power	5 MW
Number of blades	3
Rotor/ Hub diameter	126 m, 3 m
Hub height	90 m
Cut-in wind speed	3 m/s
Cut-out wind speed	25 m/s
Rated wind speed	11.4 m/s
Rated generator speed	1173.7 rpm
Gearbox ratio	97

healthy simulations and 160 to failure simulations, where each type of failure has 20 associated simulations, as shown in table III. Likewise, each simulation lasts 600 s. However, not all this time is used but only the 400 s to avoid the transient behavior present at the beginning of the simulation.

TABLE II: Description of the available sensors in the WT.

Number	Sensor Type	Symbol	Unit	Noise Power
S1	Generated electrical power	$P_{e,m}$	W	$1.0 \times 10^{-1}$
S2	Rotor speed	$\omega_{r,m}$	rad/s	$1.0 \times 10^{-4}$
S3	Generator speed	$\omega_{g,m}$	rad/s	$2.0 \times 10^{-4}$
S4	Generator torque	$\tau_{e,m}$	Nm	$9.0 \times 10^{-1}$
S5	Pitch angle of first blade	$\beta_{1,m}$	deg	$1.5 \times 10^{-3}$
S6	Pitch angle of second blade	$\beta_{2,m}$	deg	$1.5 \times 10^{-3}$
S7	Pitch angle of third blade	$\beta_{3,m}$	deg	$1.5 \times 10^{-3}$
S8	Tower top fore-aft acceleration	$\alpha_{fa,m}$	m/s <sup>2</sup>	$5.0 \times 10^{-4}$
S9	Tower top side-to-side acceleration	$\alpha_{ss,m}$	m/s <sup>2</sup>	$5.0 \times 10^{-4}$

TABLE III: Experimented WT faults during the simulation.

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

## 3. Methodology

The proposed methodology is carried out as follows: i) the data is previously processed (collected, scaled, reshaped, and converted into a matrix with many channels as sensors); ii) data is divided into training, and validation data sets; iii) the fault classification model is built by designing a deep CNN. The selected SCADA variables that are used to develop the classification model are described in table II. These variables are the input to the CNN to predict the failures shown in table III. Note that six failures will be classified in addition to healthy data.

### A. Data Preprocess

Data is stored in an array  $\mathbf{X} \in \mathcal{M}_{90744 \times 9}(\mathbb{R})$  such that the number of timestamps is given by the number of rows in the data, which is  $I = 90744$ , and the number of columns is equal to the number of sensors (variables),  $J = 9$ .

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,9} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,9} \\ \vdots & \vdots & \ddots & \vdots \\ x_{90746,1} & x_{90746,2} & \cdots & x_{90746,9} \end{pmatrix}, \quad (1)$$

## B. Data Split

The available data is divided into training and validation sets to develop the classification model. This data has been divided as follows: 75 % for training and 25 % for validation. This corresponds to 69253 data for training in the array  $\mathbf{X}_{\text{training}}$  and 21493 data for validation in the array  $\mathbf{X}_{\text{validation}}$ .

$$\mathbf{X}_{\text{training}} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,9} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,9} \\ \vdots & \vdots & \ddots & \vdots \\ x_{69253,1} & x_{69253,2} & \cdots & x_{69253,9} \end{pmatrix}, \quad (2)$$

$$\mathbf{X}_{\text{validation}} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,9} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,9} \\ \vdots & \vdots & \ddots & \vdots \\ x_{21493,1} & x_{21493,2} & \cdots & x_{21493,9} \end{pmatrix}, \quad (3)$$

## C. Data Standardization

The main reason for data standardization is to make the CNN training more efficient, significantly decreasing the number of epochs required for the CNN to learn, and thus leading to a better predictor. In particular, here the data is scaled column-wise for the matrix  $\mathbf{X}$  to be within the specific range  $[0, 1]$ , based with the maximum and minimum values of the training data. Assuming there are  $I$  samples and  $J$  SCADA variables for training, it is computed as follows,

$$M_j = \max(x_{ij}), \quad i = 1, \dots, I \quad (4)$$

$$m_j = \min(x_{ij}), \quad i = 1, \dots, I \quad (5)$$

where  $M_j$  and  $m_j$  are the maximum and the minimum values taking from training data set  $\mathbf{X}_{\text{training}}$ . From here, the scaling is implemented with  $M_j$  and  $m_j$  to create a new matrix  $\mathbf{Y}$  as

$$\mathbf{Y} = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,9} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,9} \\ \vdots & \vdots & \ddots & \vdots \\ y_{90744,1} & y_{90744,2} & \cdots & y_{90744,9} \end{pmatrix}, \quad (6)$$

## D. Data Reshape

The main goal is to detect the failure in a reduced time [14], using the available SCADA data. Note that once the pre-trained classification model needs a sample as input to the model, the smaller the sample, the smaller it will be the time detection due to the data acquisition time in the sensors. Assuming that  $T_d$  is the detection time, the fault detection requirements given in model [15] for the corresponding faults are described in terms of the sampling time  $T_s$ ; which in this case is 1 second.

It has been chosen to comply with  $T_d < 3T_s$  because it is the most restrictive detection time related to the torque actuator. For the other faults, the detection time is higher ( $T_d < 8T_s$ ,  $T_d < 10T_s$ , and  $T_d < 100T_s$ ). It is proposed to organize the available data from the simulations in 3-time steps (this will lead to a detection time of approximately 3  $T_s$ ). The transformation is carried out individually for each of the nine variables used as CNN inputs; the matrices are obtained with three samples (3-time steps). Table IV shows the implementation of this single, unified view approach.

## E. Convolutional Neural Network

The proposed CNN architecture is shown in figure 2, and the most relevant characteristics are given in Table V. Initially, the input matrix has a dimension of  $1 \times 3 \times 9$ . This data passes through the first convolution module; This module is made up of 14 filters ( $2 \times 2$  kernel) with padding and stride equal to 1, resulting in an output size of  $2 \times 2 \times 14$ . The second convolution has 21 filters

and the same kernel configuration, padding, and stride as the first convolution, with an output size of  $3 \times 5 \times 21$ . the third convolution has 26 filters and the same configuration of kernel, padding, and stride of the first and second convolutions, with an output size of  $4 \times 6 \times 26$ . Finally, four fully connected layers with sizes 350, 175, 50, and 7. Note that all convolutional layers and connected layers have a ReLU trigger function.

The final layer of the network is a softmax block, which is used to squash the 7-dimensional output into a categorical probability distribution: (0) Healthy; (1) Failure F1, hydraulic leak at the pitch sensor; (2) Failure F2, Gain factor in the generator speed sensor; (3) Failure F3, Stuck value (5 deg) at the pitch sensor; (4) Failure F4, Stuck value (10 deg) at the pitch sensor; (5) Failure F5, Gain factor at the pitch sensor; (6) Failure F6, Offset value on torque actuator.

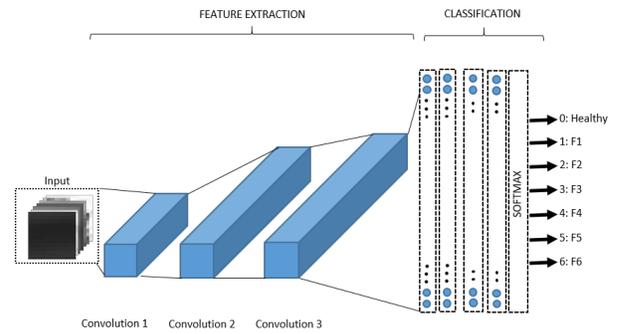


Fig. 2: CNN architecture.

The parameters have been optimized with the Adam optimization algorithm, the values of the selected hyper-parameters are an initial learning rate of  $\alpha_0 = 0.001$ , a gradient decay factor of  $\beta_1 = 0.9$ , a squared gradient decay factor of  $\beta_2 = 0.999$ , and a  $\varepsilon = 1^{-8}$ .

## 4. Results

Multiple metrics can be calculated from a confusion matrix; typically, these metrics evaluate binary classification problems. In a multiclass classification problem, such as the one considered in this paper, these metrics are also applicable using a one-versus-all approach to calculate each metric for each class. This is essential to calculate the different metrics for each label.

Figure 3 illustrates the confusion matrix for the validation data set. The rows represent the true class, while the columns represent the predicted class. The precision and false discovery rate are given in the far right columns. Finally, the recall and false negatives rates are shown in the lower rows. An examination of the confusion matrix reveals that some misclassifications come from the pitch sensor failures and healthy data (labels 3,4,5 and 0).

From the confusion matrix, the different metrics to evaluate the classification model, are computed and presented in Table VII. Table VI shows the average of the results obtained. After 230 epochs of training on a laptop running Windows 10 with an Intel Core i7-9750H, 16GB of RAM, and a 6GB GPU graphics card (GeForce RTX 2060), the overall accuracy performance obtained is 88%. A precision of 100% is reached for fault F2 and fault F6, and to a lesser extent for fault F4 with 85.5%, the lowest precision corresponds to the healthy data. Regarding Recall, the highest result is obtained in failure F2 with 100% and the lowest with failure F5 with 51.5%. For the F1 score metric, 100% is obtained for failures F2 and F6, and the lowest result corresponds to failure F5 with 65.5%. Finally, for specificity, 100% is obtained for the F2 and F6 failure, and to a lesser extent with the healthy data. Thus, the results confirm the viability of the proposed methodology.

TABLE IV: Data reshaped

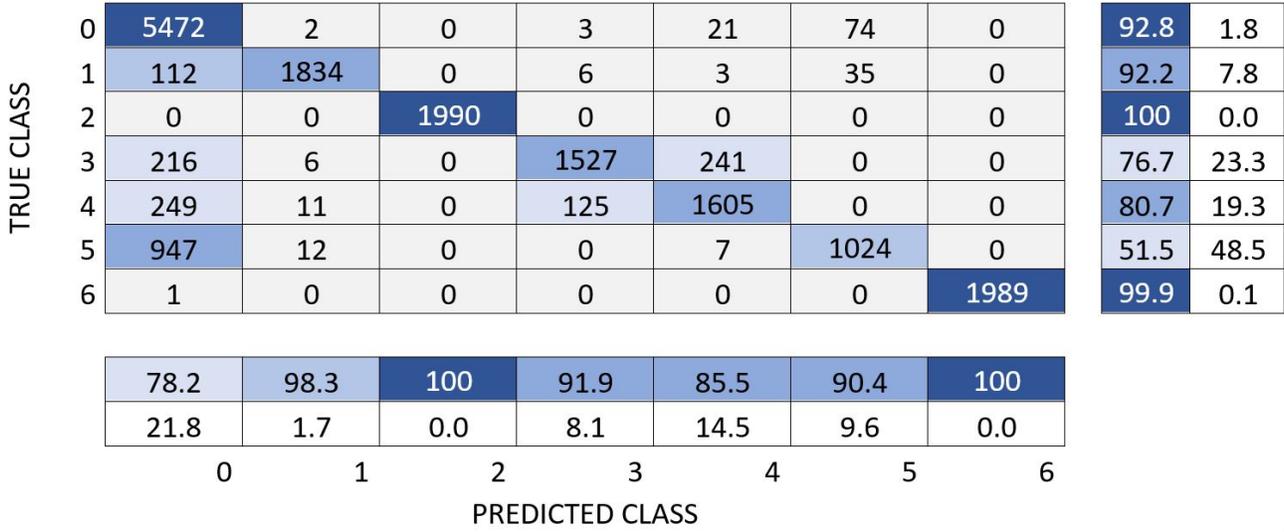
$$V = \begin{pmatrix} \begin{array}{ccc|c|ccc} \text{SCADA variable 1} & & & \dots & & \text{SCADA variable 9} \\ \hline x_{1,1} & x_{2,1} & x_{3,1} & & x_{1,9} & x_{2,9} & x_{3,9} \\ x_{2,1} & x_{3,1} & x_{4,1} & & x_{2,9} & x_{3,9} & x_{4,9} \\ \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ x_{90744,1} & x_{90744,1} & x_{90744,1} & & x_{90744,9} & x_{90744,9} & x_{90744,9} \end{array} \end{pmatrix}$$


Fig. 3: Validation confusion matrix

TABLE V: Characteristics of the designed CNN.

Layer	Output size	Parameters	# of Parameters
Input	$1 \times 3 \times 9$	-	0
Convolution#1 kernel size (2, 2), stride (1, 1), padding(1, 1) ReLU#1	$2 \times 4 \times 14$	Weights $2 \times 2 \times 9 \times 14$ Bias $1 \times 1 \times 14$	518
Convolution#2 kernel size (2, 2), stride (1, 1), padding(1, 1) ReLU#2	$3 \times 5 \times 21$	Weights $2 \times 2 \times 14 \times 21$ Bias $1 \times 1 \times 21$	1197
Convolution#3 kernel size (2, 2), stride (1, 1), padding(1, 1) ReLU#2	$4 \times 6 \times 26$	Weights $2 \times 2 \times 21 \times 26$ Bias $1 \times 1 \times 26$	2210
Fully connected layer#1	$1 \times 350$	Weights $624 \times 350$ Bias $350 \times 1$	218750
Fully connected layer#2	$1 \times 175$	Weights $350 \times 175$ Bias $175 \times 1$	61425
Fully connected layer#3	$1 \times 50$	Weights $175 \times 50$ Bias $50 \times 1$	8800
Fully connected layer#4	$1 \times 7$	Weights $50 \times 7$ Bias $7 \times 1$	357
Softmax	-	-	0
Classoutput	-	-	0

TABLE VI: Average of metrics

Precision	Recall	F1 score	Specificity
92.1	85.6	87.8	97.6

## 5. Conclusions

Due to its low standard sampling frequency, there is a lack of knowledge about the potential of SCADA data for condition monitoring. In this work, a promising strategy to detect and classify

multiple WT faults is presented using only conventional SCADA data with additional, but feasible, high-frequency sampling of the sensors (1 Hz). In other words, the strategy does not involve complementary installation of expensive data detection equipment specially designed for WT condition monitoring.

The contribution of this work is the design of a deep CNN, whose architecture and hyperparameters play a key role in the specific application of fault classification. Furthermore, the proposed method does not require hand-designed functions in advance because CNN learns the representative features automatically. In particular, a remarkable overall precision of 92.1% is obtained. These results show that large (deep) CNNs are promising to develop WT fault diagnosis strategies. Future work will focus on two main areas. First, an investigation will be conducted to gain the ability to separate the unknown faults from the known and specific studied fault types. Finally, to face the validation of the proposed strategy in a more realistic environment, a down-scaled real WT in a laboratory will be used.

## Acknowledgments

This work 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. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

TABLE VII: Metrics for each label of the multiclassification problem.

label	Precision	Recall	F1 score	Specificity
0: Healthy data	78.2	98.2	87.1	86.7
1: Failure F1, hydraulic leak at the pitch sensor	98.3	92.2	95.1	99.8
2: Failure F2, Gain factor in generator speed sensor	100.0	100.0	100.0	100.0
3: Failure F3, Stuck value (5deg) at the pitch sensor	91.9	76.7	83.6	99.0
4: Failure F4, Stuck value (10deg) at the pitch sensor	85.5	80.7	83.0	98.1
5: Failure F5, Gain factor at the pitch sensor	90.4	51.5	65.5	99.2
6: Failure F6, Offset value on torque actuator	100.0	99.9	100.0	100.0

## References

- [1] W. Europe, Wind energy in europe in 2019—trends and statistics, Wind Europe: Brussels, Belgium (2020).
- [2] C. Xiao, Z. Liu, T. Zhang, X. Zhang, Deep learning method for fault detection of wind turbine converter, *Applied Sciences* 11 (3) (2021) 1280.
- [3] Siemens gamesa renewable energy, <https://www.siemensgamesa.com/product-and-services/offshore/wind-turbine-sg-14-222-dd>, accessed: 2021-02-24.
- [4] E. I. Konstantinidis, S. Katsavounis, P. N. Botsaris, Design structure matrix (dsm) method application to issue of modeling and analyzing the fault tree of a wind energy asset, *Wind Energy* 23 (3) (2020) 731–748.
- [5] M. A. Ozgur, Review of turkey’s renewable energy potential, *Renewable Energy* 33 (11) (2008) 2345–2356.
- [6] A. Reddy, V. Indragandhi, L. Ravi, V. Subramaniaswamy, Detection of cracks and damage in wind turbine blades using artificial intelligence-based image analytics, *Measurement* 147 (2019) 106823.
- [7] J. Tautz-Weinert, S. J. Watson, Using scada data for wind turbine condition monitoring—a review, *IET Renewable Power Generation* 11 (4) (2016) 382–394.
- [8] B. Badrzadeh, M. Bradt, N. Castillo, R. Janakiraman, R. Kennedy, S. Klein, T. Smith, L. Vargas, Wind power plant scada and controls, in: 2011 IEEE Power and Energy Society General Meeting, IEEE, 2011, pp. 1–7.
- [9] A. Zaher, S. McArthur, D. Infield, Y. Patel, Online wind turbine fault detection through automated scada data analysis, *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 12 (6) (2009) 574–593.
- [10] Y. Wang, D. Infield, Supervisory control and data acquisition data-based non-linear state estimation technique for wind turbine gearbox condition monitoring, *IET Renewable Power Generation* 7 (4) (2013) 350–358.
- [11] B. Puruncajas, Y. Vidal, C. Tutivén, Vibration-response-only structural health monitoring for offshore wind turbine jacket foundations via convolutional neural networks, *Sensors* 20 (12) (2020) 3429.
- [12] J. Jonkman, National wind technology center computer-aided engineering tools (fast).
- [13] N. Kelley, B. Jonkman, National wind technology center computer-aided engineering tools (turbsim), last modified 30 may 2013.
- [14] F. Pozo, Y. Vidal, J. M. Serrahima, On real-time fault detection in wind turbines: Sensor selection algorithm and detection time reduction analysis, *Energies* 9 (7) (2016) 520.
- [15] P. F. Odgaard, K. E. Johnson, Wind turbine fault detection and fault tolerant control—an enhanced benchmark challenge, in: 2013 American Control Conference, IEEE, 2013, pp. 4447–4452.