Fault detection and identification for control systems in floating offshore wind farms: A supervised Deep Learning methodology

Ocean Engineering 310 (2024) 118678

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Fault detection and identification for control systems in floating offshore wind farms: A supervised Deep Learning methodology

Ana Fernandez-Navamuel [a,b,*], Yerai Peña-Sanchez [c], Vincenzo Nava [a,b]

[a] TECNALIA, Basque Research and Technology Alliance (BRTA), 48169, Derio, Spain
[b] Basque Center for Applied Mathematics (BCAM), 48001, Bilbao, Spain
[c] Fluid Mechanics Department, Mondragon University, 20500 Arrasate, Spain

ARTICLE INFO

Keywords:
Floating offshore wind turbines
Inverse problems
Control system fault detection
Fault identification and isolation
Deep neural networks
Wind farm assessment

ABSTRACT

This study employs a data-driven Fault Detection and Isolation (FDI) methodology in Floating Offshore Wind Turbine (FOWT) farms. The main objective of the work lies in classifying faults impacting the components of the control subsystems across multiple turbines. Unlike existing research, the emphasis here is placed specifically on identifying and classifying non-critical faults, which may result in suboptimal farm performance without necessitating a shutdown. From a methodological perspective, a Deep Neural Network has been designed to solve the classification problem by providing a probability vector, the most probable class indicator of the true state. One of the major contributions of this work lies in its applicability to FOWT farms instead of being confined to individual devices, facilitating a comprehensive performance assessment at the global farm level. The integration of this data-driven methodology with tolerant control strategies might enable early intervention, mitigating the impact of these faults and enhancing overall power generation efficiency. The target case study is a three-FOWT farm modeled in a Simulink environment, allowing for the simulation of operational behavior under diverse conditions and various faults affecting sensors and actuators. This work considers ten distinct fault classes, including the healthy condition, and three possible faults for each FOWT: pitch angle sensor, pitch angle actuator, and generator speed sensor. These frequent faults pose challenges to the optimal functioning of the control system managing the FOWTs. The outcomes highlight that the estimated probability of the healthy state serves as a robust indicator for detecting unknown faults. Results also demonstrate the adequate efficacy of the method in pinpointing the fault origin. However, we observe confusion between pitch sensor and actuator faults that require further investigation for comprehensive understanding.

1. Introduction

The energy sector accounts nowadays for more than 75% of the greenhouse gas emissions in Europe. The green deal of achieving a clean, safe, interconnected, and decarbonized energy picture is thus of paramount importance (Hafner and Raimondi, 2021). In this context, wind energy is increasingly gaining weight in transitioning towards a renewable-energy-driven global energy mix. The European commitment to achieve a 45% contribution of renewable energy by 2030 has motivated a ranging increase in the installation of wind farms. In particular, special interest is currently devoted to offshore wind solutions, given their advantages compared to their onshore analogous (possibility to install bigger turbines, steadier wind speeds, etc.) (Gibbs and Jensen, 2022; Durakovic et al., 2023; Talarek et al., 2022).

Wind turbines (WTs) are complex structural systems where many operating mechanisms interact, including electromechanical, aerodynamics, and control devices (Cho et al., 2018). Such systems incur considerable operation and maintenance costs, estimated to be approximately 10–15% (in the case of onshore) and 30% (in the case of offshore) of the total income of the farms (Sørensen and Sørensen, 2012; McMillan and Ault, 2007; Nava et al., 2019). This economic impact poses the main bottleneck for their consolidation as a profitable energy source (Fekih et al., 2021). Among the offshore solutions, Floating Offshore Wind Turbines (FOWTs) are particularly expensive, given the harsh and hazardous environments and the accessibility problems of the sites where they operate (Ciuriuc et al., 2022). For this reason, increasing efforts have been devoted to developing strategies that reduce downtime and maintenance costs. These activities

* Corresponding author at: TECNALIA, Basque Research and Technology Alliance (BRTA), 48169, Derio, Spain.
E-mail address: ana.fernandez-navamuel@tecnalia.com (A. Fernandez-Navamuel).

https://doi.org/10.1016/j.oceaneng.2024.118678
Received 27 March 2024; Received in revised form 10 June 2024; Accepted 4 July 2024
Available online 16 July 2024

Abbeviations

WT	Wind turbine
FOWT	Floating Offshore Wind Turbine
FDI	Fault Detection and Isolation
FTC	Fault Tolerant Control
PCA	Principal Component Analysis
FFT	Fast Fourier Transform
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
DNN	Deep Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
SVM	Support Vector Machine

(e.g., preventive maintenance or fault detection) enable a more cost-effective performance of FOWT farms (Odgaard and Johnson, 2013; Reder et al., 2016).

The control system is one of the most essential subcomponents of FOWTs, given its critical role in their behavior and efficient performance. Control devices (i.e., sensors, actuators, control units) represent the second most frequent failure in WTs, following the electrical system (Rolfes et al., 2014; Scheu et al., 2019; Habibi et al., 2020). They often cause non-critical faults that result in suboptimal performance of the system. However, if these faults remain undetected over time, they may lead to severe failure or unplanned shutdown (Badihi et al., 2022; Reder et al., 2016).

Further, in the case of FOWTs, these faults may originate instability problems on the floating platform due to unexpected aerodynamic forces (Salic et al., 2019; Yu et al., 2018b). The accessibility delays and issues due to safety regulations (daytime hours, adequate weather and sea conditions, reduced wind levels, etc.) on maintenance incursions enforce FOWTs to operate suboptimally for long periods (Badihi et al., 2022). Therefore, the early detection of operative faults is critical for managers and owners of FOWT farms. In this light, real-time monitoring, fault detection and isolation (FDI), and fault tolerant control (FTC) methods that attempt to minimize the impact of such periods of underperformance are becoming more and more popular (Rotondo et al., 2012; Gao and Sheng, 2018; Badihi et al., 2022). FDI can be envisioned as an inverse problem where the health condition of a system (e.g., a wind turbine) is inferred from observations of its measurable response (Isermann, 2005; Gao and Sheng, 2018). In this work, we employ an FDI approach for fault classification in the control devices of a FOWT farm.

FDI techniques detect and identify potential faults in a system from the monitoring data acquired during operation (Isermann, 2005; Gao and Sheng, 2018). Over the past decade, many studies have proposed and analyzed FDI methods in WTs (Tchakoua et al., 2014; Liu et al., 2015; Gao et al., 2015; Sharan and Jain, 2018; Liu et al., 2021; Cho et al., 2021; Wu and Ma, 2022; Saci et al., 2022; Rahimilarki et al., 2022). These techniques can be broadly classified into model-based and data-based strategies (Saci et al., 2022). On the one hand, model-based approaches employ a mathematical (physics-based) representation of the dynamic system to produce redundant signals (Gao and Liu, 2021; Fekih et al., 2022; Habibi et al., 2019).

Model-based approaches have been extensively applied for rapidly detecting faulty behavior (Peña-Sanchez et al., 2023; Ozdemir et al., 2011; Tutivén et al., 2018; Zhang et al., 2019; Cho et al., 2018; Saci et al., 2022). Deep Neural Networks (DNNs), mainly with convolutional and residual architectures, have also been employed to build the signal estimator (Wu and Ma, 2022; Zhang et al., 2019).

On the other hand, data-based approaches directly exploit large volumes of available data from the operating system to detect and identify faulty behavior on different subcomponents (Fekih et al., 2022). These signals may include vibration, acoustic emission, strain, torque, temperature, electrical measurements, and Supervisory Control And Data Acquisition (SCADA) system signals (Dong and Verhaegen, 2011; Qiao and Lu, 2015; Pozo and Vidal, 2016; Fu et al., 2020).

In recent years, data-driven techniques have experienced considerable development, achieving very robust and sophisticated tools. One key driver of this empowerment is Artificial Intelligence (AI), and particularly Machine Learning (ML) techniques, including Deep Neural Networks (DNNs) (Tang et al., 2021; Ng and Lim, 2022). Interested readers are referred to works (Tang et al., 2021) and Stetco et al. (2019) for exhaustive literature reviews on ML methods for wind turbine FDI. Convolutional neural networks are one of the most extensively employed methods for handling space and time-domain dependencies. For example, in work (Rahimilarki et al., 2022), the authors propose a CNN-based methodology to detect and identify minor anomalies in the outputs of a WT by converting the raw time-domain signals into 2D (grayscale) images filled row-wise with a segment of the entire time-series signals. Besides, Yu et al. (2018a) implement a Deep Belief network trained with historical data acquired during wind turbine operation for effective fault detection.

Despite the vast literature on the field, the existing works focus on detecting faults at one specific asset (e.g., the pitch sensors), thus preventing a more general and reliable diagnostic. In particular, most works apply to one isolated wind turbine, and they lack extrapolation to the farm level, where multiple turbines interact. For example, Bach-andersen et al. (2017) employ a convolutional neural network to extract significant features from complex multisensor vibration signals, which are then classified to isolate faults. They validate the method using data from various WTs of the same family but treated as independent assets. Xiao et al. (2021) employ SCADA data to detect faults in the conversion system of a WT via a Deep Convolutional Network. A more sophisticated approach is proposed by Xiang et al. (2021), where authors combine the strengths of CNNs and LSTMs. Cho et al. (2021) address a classification problem for faults in the blade pitch sensor and control valve of a spar-type FOWT using autoencoder-based recurrent neural networks for feature extraction and fault estimation.

After an extensive review, the authors found few works that simultaneously applied FDI to two or more turbines. Jiang et al. (2022) proposed a federated Deep Learning (DL) framework to integrate data from multiple wind turbines, using a multi-scale residual attention network to extract the relevant features from the measured signals. Their approach employs data collected with the SCADA system and attempts to detect electromechanical faults. Aziz et al. (2022) present a multi-turbine strategy to detect faults that cause an underperformance affecting the power curve by modeling a power curve for each WT to calculate a farm-level residual. However, their strategy detects underperformance problems rather than identifying the fault origin.

The proposed approach evaluates the condition of all the FOWTs simultaneously, classifying potential faults, with the main novelty being the use of a farm-based, data-driven method for fault detection in the control system (sensors and actuators) in a FOWT farm. This work aims to leverage the benefits of data synergy by aggregating information from multiple turbines, resulting in a more comprehensive dataset. In this context, the farm-based approach upon which this methodology relies represents a partial response to address the issue of class imbalance in faulty conditions. Indeed, in practical scenarios, when labeled datasets are available from several turbines simultaneously, the machine learning algorithm can capitalize on enriched training data, particularly when the control system shows anomalies. From a data management perspective, the benefits of a farm-wise approach for FDI also rely on its scalability, as a new turbine can be integrated seamlessly with small adjustments, allowing for the wind farm to expand gradually without overhauling the monitoring framework.

The main contribution of the method is its farm-wise orientation to detect faults in multiple contiguous FOWTs, enabling a more manageable assessment for farm owners. Compared to other existing works, we propose a diagnosis tool that considers faults occurring at various sub-components of the control systems of multiple interacting wind turbines.

The methodology employs a DNN with 1-D convolutional operators to extract high-level information from the measured signals in the time domain. As an application case to show the performance of the proposed methodology, an in-house Simulink-based farm model composed of three FOWTs is employed to generate the required signals for training, validation, and test stages of the DNN approach. In this case study, two possible fault modes (offset or drift) can occur on three different subsystems: pitch angle sensor, pitch angle actuator, and generator rotational speed sensor at each FOWT. We analyze the testing results via the main classification metrics and observe that pitch sensor and actuator faults are hardly differentiable given the close-loop effect in the considered variables. We contemplate including memory cells in future work to investigate this limitation. Besides, identifying simultaneous faults is out of the scope of this work and will be regarded as future research.

The remainder of this paper is organized as follows: Section 2 introduces the architecture of the wind farm model considered along with its possible faults, Section 3 describes the FDI strategy proposed in this study, Section 4 provides an application case to show the performance of the proposed FDI strategy and, finally, some conclusions are drawn and the possible future work is discussed in Section 5.

2. Wind farm modeling and control system failures

An offshore wind energy farm contains various (floating or bottom-fixed) wind turbines arranged according to a given layout. Particularly, this study considers an offshore farm composed of floating wind turbines. Each turbine comprises several interconnected subsystems: blade pitch actuator system, drive-train, electric generator, tower, floating platform, and mooring system (Tang et al., 2021; Ciuriuc et al., 2022). The turbine is excited by the forces exerted by both waves f_{wave} and wind f_{wind}. While the former is indirectly inferred, the latter is usually estimated using the wind speed (v_{wind}) measured at different points of the farm using anemometers, often located at the top of the nacelle of each turbine ($v_{\text{wind}}^{\text{nacelle}}$), and on measuring masts ($v_{\text{wind}}^{\text{MM}}$) located at a certain distance from the FOWTs.

To ensure optimal farm performance under varying excitation, the control of the FOWTs is carried out at two different levels: (i) at a farm level, where the actual available power in the farm is estimated based on wind measurements and a given reference power (P_{ref}) is demanded by each device (depending on the grid requirements) (Odgaard et al., 2013); and (ii) at a turbine level, where the blade pitch angles and the generator torque are managed via proportional–integral (PI) controllers to follow the reference from the farm controller (Odgaard et al., 2013; Lan et al., 2018). In a simplified version, the block diagram depicted in Fig. 1 describes the different subsystems of a FOWT, their interactions within the control system, and the most relevant variables involved.

The FOWT controller will force the turbine to operate on a different control regime, depending on the exciting wind speed v_{wind}, to attain an efficient performance. Fig. 2 depicts the power curve that describes the operative regions of WTs depending on the wind speed.

The control regimes are described as follows:

- *Region I*: If $v_{\text{wind}} \leq v_{\text{cut-in}}$, the turbine is stationary.
- *Region II*: If $v_{\text{cut-in}} < v_{\text{wind}} < v_{\text{rated}}$, the controller modifies the generator torque to produce the maximum power. At the rated velocity (v_{rated}), the turbine generates the rated power, given by the nominal power of the generator.
- *Region III*: If $v_{\text{rated}} \leq v_{\text{wind}} < v_{\text{cut-out}}$, the controller regulates the pitch angle to neglect wind power and minimizes the loads affecting the rotating system while generating the rated power.
- *Region IV*: If $v_{\text{cut-out}} \leq v_{\text{wind}}$, the pitch system cannot dissipate enough energy from the wind, and the turbine is shut down to prevent unnecessary strain on the generator.

Note that only the second and third operating regimes are considered here since the study aims to detect faults occurring during operation, which only happens in such regions.

In this work, the farm is numerically modeled using the time-domain FOWLTY simulator (Peña-Sanchez et al., 2023, 2022), which is a wind farm (onshore or offshore) simulation toolbox developed in Matlab/Simulink, based on the existing SimWindFarm simulator (Grunnet et al., 2010), able to recreate faults on the different subsystems of the turbines. The simulator includes four (fully-coupled) modules: (i) the *wind field* module, which has the information of the undisturbed wind field and adds the wake effects of the FOWTs according to their current behavior; (ii) the *wind turbines* module, which describes the individual WTs; (iii) the *farm controller* module, which implements the aforementioned farm-level control assigning a reference power to each turbine depending on the available wind power; and (iv) the *network* module, which simulates the grid behavior.

Since the FOWLTY toolbox is a control-oriented model, it considers a set of assumptions to simplify the model. One of the strongest simplifications is assuming a constant mean wind speed and direction (traveling along the x axis in the positive direction). Such simplification makes modeling the yaw mechanism unnecessary, and the FOWTs are assumed to always face the wind. Additionally, waves are generated from a JONSWAP spectrum (fully characterized by the peak period and the significant wave height) (Hasselmann, 1973) and are assumed to travel in the same direction as the wind, which simplifies the definition of the floating platform. The interested reader is referred to Peña-Sanchez et al. (2023, 2024, 2022), Grunnet et al. (2010) for a more comprehensive description of the simulation toolbox, including the models of the different subsystems and the simplifying assumptions, or to Peña-Sanchez et al. (2024) for validation of FOWLTY against the well-known simulation tool OpenFAST (Jonkman et al., 2022).

Among the different faults that can be recreated using FOWLTY, this study focuses on a set of non-critical faults affecting the control closed-loop governing the FOWTs behavior, i.e., critical faults implying system shutdown are out of the scope of this work. In particular, faults affecting the measured pitch angle (pitch sensor fault), the measured generator speed (generator sensor fault), and the actual blade pitch angle (pitch actuator fault) have been considered.

On the one hand, faults on the pitch subsystem are considered since that is one of the most vulnerable components in FOWTs (Fekih et al., 2021). Faults in the pitch sensor can disrupt the controller and turbine performance, potentially causing structural imbalances due to uneven blade loading (Mazare et al., 2021). The pitch actuator is also susceptible to faults, including hydraulic leakage and air content, which alter the system's dynamic response and compromise angle regulation (Habibi et al., 2019). On the other hand, issues may also arise in the generator subsystem (such as the generator speed sensor or applied generator torque) during operation (Odgaard and Stoustrup, 2014). For instance, a bias in the generator speed sensor might cause pitch angle deviations and instability (Mazare et al., 2021).

In this work, two fault types, frequently observed in control signals of FOWTs, affecting the pitch and generator subsystems are considered: offset and drift (Jana et al., 2022; Singh et al., 2018). The two possible fault modes are introduced in Simulink by altering the original (healthy) time-domain signal[1] as:

[1] Note that the faults are not manually implemented since the considered simulator (FOWLTY) already includes the option to implement the introduced faults at any given time.

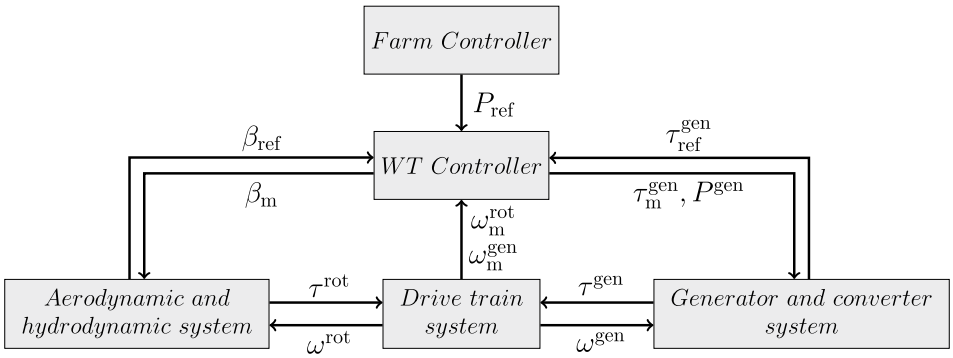


Fig. 1. Block diagram describing wind turbine subsystems interaction and control. P_{ref} indicates the power reference the farm controller demands to each FOWT. β_{ref} and $\tau^{\text{gen}}_{\text{ref}}$ are the reference blade pitch angle and generator torque, respectively, demanded by the controller. β indicates the actual blade pitch angle, τ^{rot} the rotor torque, and ω^{rot} and ω^{gen} the rotor and generator rotational speeds, respectively. The subscript $\{\cdot\}_{\text{m}}$ indicates the measured variables.

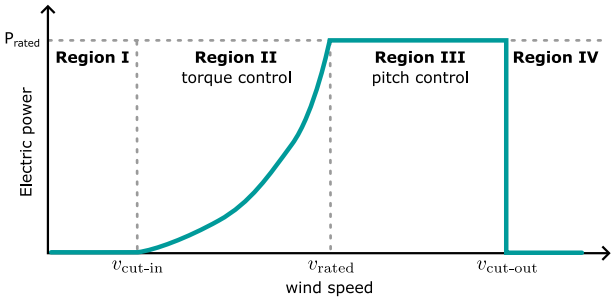


Fig. 2. Power reference curve for a FOWT depending on the wind speed.

Offset fault: the signal presents a constant offset with respect to the original (healthy) value, such that:

$$x^{\text{faulty}}[t] = x^{\text{healthy}}[t] + \delta_{\text{offset}}, \tag{1}$$

where δ_{offset} represents the offset constant, and $x^{\text{healthy}}[t]$ and $x^{\text{faulty}}[t]$ indicate the healthy and faulty observation at time instant t, respectively.

Drift fault: the signal presents a time-dependent offset that follows a linear trend (Jana et al., 2022; Singh et al., 2018), such that:

$$x^{\text{faulty}}[t] = x^{\text{healthy}}[t] + \delta_{\text{drift}} \times t, \tag{2}$$

with δ_{drift} indicating the coefficient of the linear trend.

3. Fault diagnosis methodology

Fault diagnosis is an inverse problem where we employ observations of the system behavior to infer the faults that cause that conduct (Camps Echevarría et al., 2014). The most relevant characteristic of inverse problems is their ill-posedness since they may have multiple solutions (Aster et al., 2005). This property makes their solution a challenging task to solve. This section describes the proposed methodology for implementing an FDI strategy for fault identification in an FOWT farm. Fig. 3 represents a flowchart of the proposed methodology.

Let $\hat{\mathbf{X}} \in \mathbb{R}^{S\times N\times v}$ represent the dataset containing N raw observations from the v variables, including simulations from S fault scenarios. These variables include sensor and actuator signals from the different interacting subsystems described in Section 2 (see Fig. 1). We apply a linear re-scaling variable-wise into the interval $[0.5, 1.5]$ to ensure that the v features have the same order of magnitude. For a certain variable $\hat{\mathbf{x}}$, we define $\hat{x}_{min} := min(\hat{\mathbf{x}})$, and $\hat{x}_{max} := max(\hat{\mathbf{x}})$. We then express the rescaling function as:

$$\mathcal{R}(\hat{\mathbf{x}}) = \frac{\hat{\mathbf{x}} - \hat{x}_{min}}{\hat{x}_{max} - \hat{x}_{min}} + 0.5. \tag{3}$$

We apply Eq. (3) to produce the re-scaled variables, such that $\mathbf{x} = \mathcal{R}(\hat{\mathbf{x}})$. From now one, we will be using the rescaled version of the measured signals, denoted by $\mathbf{X} = \mathcal{R}(\hat{\mathbf{X}})$.

Since this work aims to detect damage early, we must employ short signal segments as the input data to perform the classification task. Thus, we divide the rescaled signals into shorter segments of n measurements. Given the computational constraints (time-demand) of generating the synthetic signals, we employ a data augmentation technique based on overlapping (Shengnan et al., 2020) to increase the number of available samples. Fig. 4 schematically represents the segmentation process of a rescaled signal with n_{ol} overlapping points between consecutive segments. After this process, each sample X_i contains the n-dimensional re-scaled segments for each of the v variables, such that:

$$X_i = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1v} \\ x_{21} & x_{22} & \cdots & x_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nv} \end{bmatrix}. \tag{4}$$

For a certain sample X_i, we express the inverse problem as:

$$\mathbf{y}^{\text{true}}_i = \mathcal{I}(X_i), \tag{5}$$

where $\mathcal{I}$ refers to the inverse operator, and vector $\mathbf{y}^{\text{true}}_i$ describes the system condition diagnostic. We envision $\mathcal{I}$ as a classification task where, given an observation X_i, the fault condition vector $\mathbf{y}^{\text{true}}_i \in [0,1]^K = [y^{\text{true}}_{i,1}, \dots, y^{\text{true}}_{i,K}]$ contains the probability of the system being at any of K possible states. However, in most cases, the true inverse operator is a complex physical process typically unknown. In this work, we propose a DL-based approach to approximate $\mathcal{I}$, and apply it to the case study described in Section 4.

On the other hand, we consider K possible states representing the different fault conditions, including the reference (healthy) state. For any fault scenario $k = 1, 2, \dots, K-1$, each observation j at the ith sample receives a single-value condition label $\hat{y}^{\text{true}}_{i,j} \in \{0, k\}$. Although we consider two failure modes (i.e., offset and drift), this work intends to identify the fault origin regardless of the mode. For this reason, offset and drift signals under the same scenario share the condition label. Hence, the total number of states for the classification problem is $K = 10$, including three faults (see Section 2) for each FOWT and the healthy condition of the farm.

We label each sample according to the rate of healthy/faulty observations. We define n_{healthy} as the number of healthy observations in a sample, $n_{\text{healthy}} = \sum_{i=1}^{N}[\hat{y}_i = 0]$. Analogously, we obtain the number of faulty observations as $n_{\text{faulty}} = \sum_{i=1}^{n}[\hat{y}_i \neq 0]$. We build the probability vector $\mathbf{y}^{\text{true}}_i$ such that for fault k:

$$y^{\text{true}}_{i,j} = \begin{cases} \frac{n_{\text{healthy}}}{n} & \text{if } j = 0 \\ \frac{n_{\text{faulty}}}{n} & \text{if } j = k \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

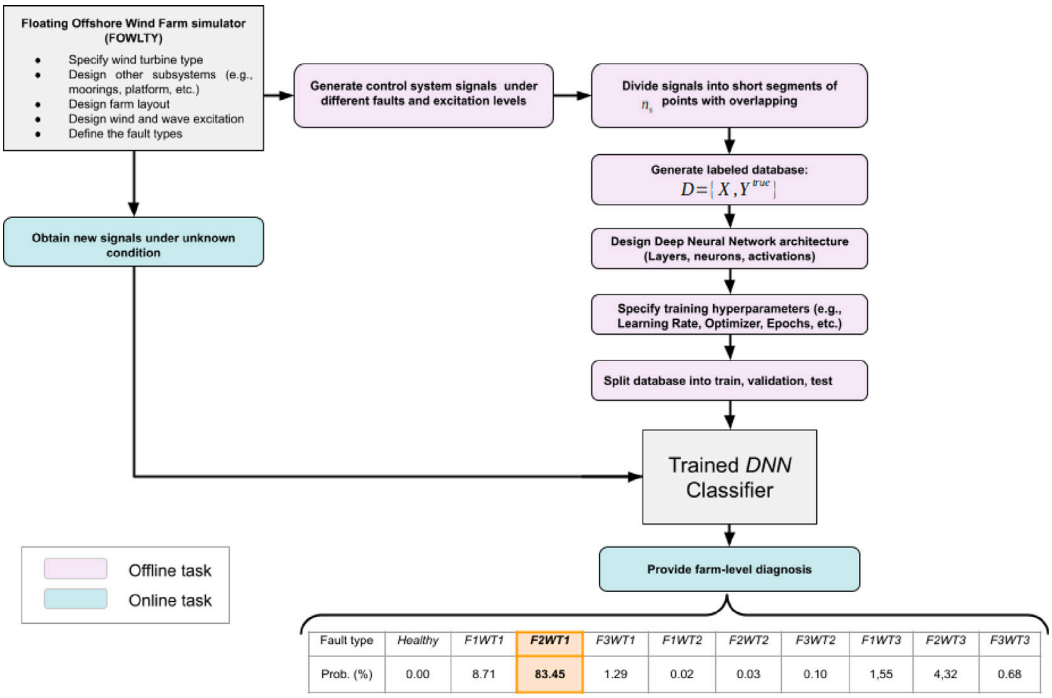


Fig. 3. Flowchart describing the proposed methodology. The offline tasks cover the farm model configuration, the dataset generation and pre-processing to feed the Deep Neural Network classifier, and the training–validation stage. In the online task, once trained, the classifier receives newly acquired signals and provides a diagnostic in the form of probability values associated with each class.

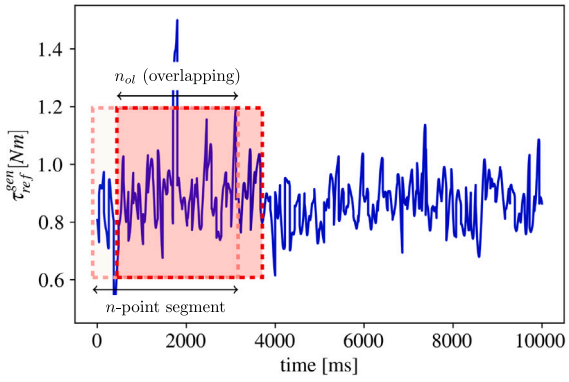


Fig. 4. Generation of short-time segments from the available simulated signals after rescaling. Each segment contains n data points, and an overlapping of n_{ol} observations is selected between consecutive segments for data augmentation purposes.

As an example, the true probability vector associated with a sample that contains two fault observations of type $k = 7$ is:

$$\mathbf{y}^{\text{true}} = [0.8\ 0\ 0\ 0\ 0\ 0\ 0\ 0.2\ 0\ 0], \tag{7}$$

where $n_{\text{healthy}} = 8$ since no simultaneous faults are considered.

Finally, we denote by $D = [\mathbf{X}, Y^{\text{true}}]$ to the dataset, where array $\mathbf{X} = [X_1\ X_2\ \ldots\ X_N]$ collects the N observed samples, and matrix Y^{true} contains the corresponding condition probability vectors.

In this work, we approximate the inverse operator $\mathcal{I}$ that discovers faults from the signals measured in the farm FOWTs using Deep Neural Networks (DNNs). We support this decision by some singular advantages of DNNs, including (*i*) it has demonstrated strong potential in approximating extremely complex functions as they found on the theorem of universal approximation (Hornik et al., 1989); (*ii*) the use of 1D convolutional transformations is beneficial to extract high-level features from multivariate time series; and (*iii*) it enables the incorporation of physical knowledge by imposing constraints and regularization terms that allow restraining the solution space to physically meaningful outcomes, which we consider future work to enhance the results (Rojas et al., 2024). However, without loss of generality, the classification problem can be addressed via other methods.

Any DNN is structured in layers, and each layer applies a composition of an affine transformation followed by a nonlinear activation function (Goodfellow et al., 2016; Caterini and Chang, 2018). When dealing with signal segments from multiple features, often convolutional layers are included in the network architecture (Zhao et al., 2017). A convolution is a specific type of affine transformation that applies filters to extract local patterns and structures in the data (Goodfellow et al., 2016; Zhao et al., 2017). Here, we approximate $\mathcal{I}$ by a DNN denoted as $\mathcal{I}_\theta$, where θ gathers all the network parameters that define the affine transformations:

$$\mathcal{I} \approx \mathcal{I}_\theta := g \circ A_l \circ g \circ A_{l-1} \circ \ldots \circ g \circ A_1, \tag{8}$$

where A_k applies an affine transformation at the lth layer, and g indicates a nonlinear transformation (e.g., ReLU, Sigmoid, Softmax (Goodfellow et al., 2016)). The parameter vector $\theta \in \Theta$ includes the entire network parameter set.

We employ a fully-connected architecture that includes convolutional layers followed by standard fully-connected layers. Convolutional layers considerably reduce the number of parameters compared to a fully connected network with the same depth. The batch normalization layers prevent overfitting and gain robustness against data variability (Ioffe and Szegedy, 2015). The activation function in the hidden layers is "ReLU" (Goodfellow et al., 2016). We employ softmax (Goodfellow et al., 2016) activation at the output layer to constrain the outputs into the interval $[0, 1]$ and their sum up to one. This activation function is commonly employed to solve multiclass classification problems. Fig. 5 shows the selected architecture, which contains four convolutional layers in charge of extracting relevant high-level features from the multivariate time series, and five additional dense layers to map with the class label of the different damage scenarios.

The proposed architecture provided adequate results while avoiding overfitting problems. Although different schemes and types could

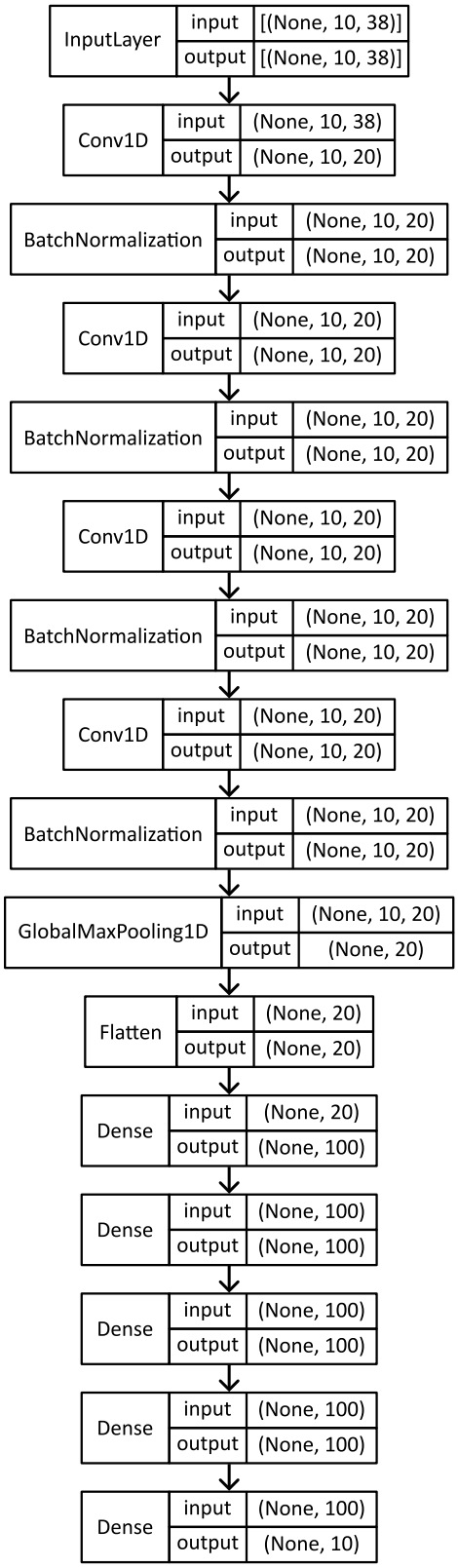


Fig. 5. Architecture of $\mathcal{I}_\theta$.

be employed, optimizing the architecture is out of the scope of this work (Elsken et al., 2019).

We evaluate the performance of $\mathcal{I}_\theta$ in approximating $\mathcal{I}$ by means of a loss function $\mathcal{L}$ (Goodfellow et al., 2016). The loss function $\mathcal{L}$ adopts

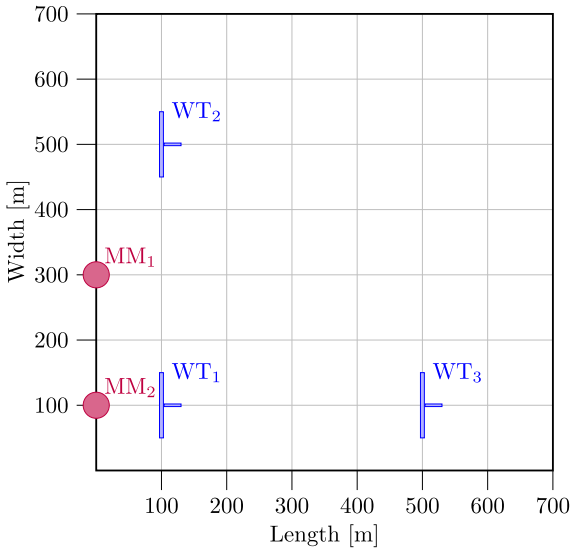


Fig. 6. Layout of the case study wind farm. The farm covers a 700×700 m^2 extension. The three wind turbines, indicated as WT_i, are 400 meters away in both directions (length and width). The measuring masts for wind monitoring are indicated as MM_j.

different forms according to the target problem. A commonly employed loss function in regression is the Mean Squared Error (MSE) (Ciampiconi et al., 2023). By contrast, in classification problems, we employ the categorical Cross-Entropy (Ciampiconi et al., 2023) as the loss function, for a certain sample $\{\mathbf{x}_i, \mathbf{y}_i^{\text{true}}\}$ and K possible classes, is expressed as:

$$\mathcal{L}(\mathcal{I}_\theta(X), Y) = -\sum_{j=1}^{K} y_{i,j}^{\text{true}} \log(\mathcal{I}_{\theta^*}(\mathbf{x}_i)), \tag{9}$$

where, $\mathbf{y}_i^{\text{true}} = [y_{i,1}^{\text{true}}, \dots, y_{i,K}^{\text{true}}]$ contains the true probabilities of each class representing the possible system conditions. Analogously, the predicted probabilities are included in $\mathbf{y}_i^{pred} \in [0,1]^K = \mathcal{I}_{\theta^*}(\mathbf{x}_i)$. In both vectors, it holds that $\sum_{j=1}^{K} y_{i,j} = 1$.

Learning an adequate DNN for the specific problem requires a training phase that learns an optimal parameter set iteratively according to some input–output samples. We split the available data samples in $\mathcal{D}$, devote 70% for training and 20% for validation, and preserve a 10% for testing. We denote $\mathcal{D}^{\text{train}} = [\mathbf{X}^{\text{train}}, Y^{\text{train}}]$ the data subset devoted for training. We describe the training process as a minimization of the loss function, such that:

$$\theta^* := \arg\min_{\theta \in \Theta} \mathcal{L}(\mathcal{I}_\theta(\mathbf{X}^{\text{train}}), Y^{\text{train}}). \tag{10}$$

where θ^* gathers the optimal set of parameters defining the transformations of the DNN. The training process requires a minimization algorithm that is often gradient-based, and the gradients are calculated via backpropagation using the chain rule (Werbos, 1990). After training, we determine the system condition by directly evaluating the DNN ($\mathcal{I}_{\theta^*}$) for any new measured sample:

$$\mathbf{y}_{new}^{pred} = \mathcal{I}_{\theta^*}(\mathbf{x}_{new}). \tag{11}$$

4. Case study

This section provides an example case study to show the effectiveness of the proposed FDI methodology. To this end, Section 4.1 introduces the considered wind farm case, Section 4.2 the variables of the farm provided to the FDI strategy, Section 4.3 analyzes and discusses the classification results, and Section 4.4 performs a condition assessment test using additional faulty signals.

4.1. Wind farm

A wind farm composed of three FOWTs and two wind-measuring masts is considered, arranged as shown in Fig. 6. We employ this layout since it includes two opposite configurations of wind turbine pairs with respect to the wind direction: parallel-wise (WT_1 and WT_3) and row-wise (WT_1 and WT_3). Adding extra turbines would not increase the complexity of the farm layout, as it would just involve another combination of the already considered situations (parallel or in-line turbines). However, additional turbines would increase computational costs due to the extra measurements required and a higher number of classes. Therefore, the current scenario represents the simplest (yet general) case, encompassing all possible interactions between turbines.

The measuring masts are usually located upwind on wind farms (deploying one or more depending on the main wind direction at the farm emplacement) to estimate the available power for the farm controller (Held, 2019). In this work, we consider two measuring masts, both 100 m up-wind on the farm, one aligned in the center of the farm and the second one in line with WT_1 and WT_3. Given the absence of specific information regarding the optimal location for one measuring mast, these two locations are considered.

The wind turbine model is the well-known NREL 5MW baseline turbine first introduced in Jonkman et al. (2009), and the chosen floating platform is a DeepCWind-like structure as defined in Robertson et al. (2014). As shown in Fig. 6, the wind field covers a 700 × 700 m area, with a grid size of 15 m, and the wind is generated with a turbulence intensity of 0.1, a sampling frequency of 20 Hz, and an average wind speed ranging from 9 to 23 m/s. Regarding the wave excitation, since it scarcely affects the behavior of the turbine for our purposes, we consider one single sea state computed according to a JONSWAP distribution (Hasselmann, 1973) with 2 m of significant wave height, 10 s of peak period, and a peak-enhancement factor of 3.3.

4.2. Dataset construction

This section describes the synthetic database generated with the wind farm simulator described in the previous section. The validation of the FOWLTY wind farm simulator employed in this work (Peña-Sanchez et al., 2023, 2024, 2022; Grunnet et al., 2010) supports the designed simulator's reliability and the quality of the dataset generated in the present work.

We select 12 control variables to represent the behavior of each FOWT, including the pitch angles from the three blades, rotor and generator speeds, generated power, generator reference torque, and wind at the nacelle top. Together with the two anemometers in the measuring masts (see Fig. 6), these result in $v = 38$ variables, as summarized in Table 1. These variables correspond to the measurable features in an actual operative wind farm.

In order to develop an effective FDI strategy that promptly identifies faults, it is essential to capture the dynamic changes from healthy to faulty states. To this end, 300 simulations of 1.2 s duration (and a sampling frequency of 20 Hz) are stacked for each fault scenario. The simulations include eight increasing wind speed levels – from 9 m/s to 23 m/s – to cover the operative conditions where the control system is active (regions *II* and *III* in Fig. 2). By way of example, Fig. 7 displays the measured wind speed at MM_1 for the entire dataset, with the wind excitation generated as introduced in Section 4. Each 1.2-s snapshot contains ten healthy measurements (0.5 s) followed by fourteen faulty measurements (0.7 s). Such snapshots result from cropping the end of a 35-s long simulation that ensures the farm reaches its steady state before inducing the fault. By generating the data this way, it is possible to ensure that the observations include enough shifts from healthy to the considered faulty states for the FDI strategy to identify the changing dynamics.

Table 1
Summary and description of the considered variables, where $i = 1, 2, 3$ and $j = 1, 2, 3$ indicate the turbine and blade number, respectively.

Variable ID	Symbol	Units	Description
j+3(i-1)	$\beta_{\mathrm{m}}^{i,j}$	deg	Reference pitch at blade j of WT_i
9+j+3(i-1)	$\beta_{\mathrm{ref}}^{i,j}$	rad/s	Measured pitch at blade j of WT_i
18+i	P_{Gen}^{i}	W	Generated power of WT_i
21+i	P_{ref}^{i}	W	Reference power of WT_i
24+i	$\omega_{\mathrm{m}}^{\mathrm{gen},i}$	rad/s	Measured generator speed of WT_i
27+i	$\omega_{\mathrm{m}}^{\mathrm{rot},i}$	rad/s	Measured rotor speed of WT_i
30+i	$\tau_{\mathrm{ref}}^{\mathrm{gen},i}$	Nm	Reference generator torque of WT_i
33+i	$v_{\mathrm{wind}}^{\mathrm{nacelle},i}$	m/s	Measured wind velocity at WT_i
37	$v_{\mathrm{wind}}^{\mathrm{MM}_1}$	m/s	Measured wind velocity at MM_1
38	$v_{\mathrm{wind}}^{\mathrm{MM}_2}$	m/s	Measured wind velocity at MM_2

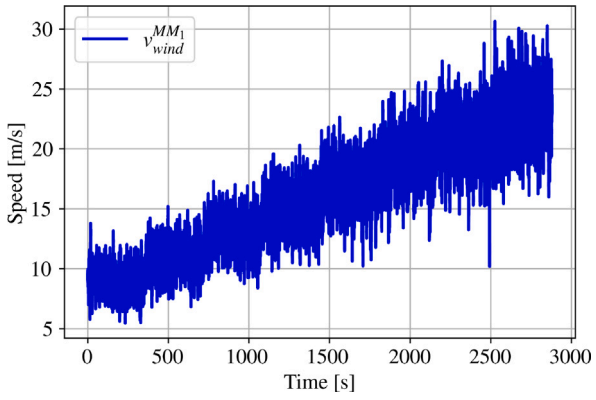


Fig. 7. Representation of the increasing wind excitation measured at MM_1 during the fault simulations. This interval covers the operating regimes where the control system takes action to optimize the farm performance.

For each wind turbine, three fault conditions are contemplated: pitch angle sensor, generator speed sensor, and pitch angle actuator as described in Section 2. We consider two fault effects for each fault, i.e., offset and drift, defined in Eqs. (1) and (2), respectively.

Table 2 details the value of the constants δ_{offset} and δ_{drift} for each fault type. Such values are selected following other existing works to represent small faults, such as (Cho et al., 2016), where authors employ pitch offset values of 3°, or (Odgaard et al., 2013) and Pozo and Vidal (2016), where fixed values of 5 degrees are employed. In Odgaard et al. (2013), gain factors of 1.1 and 0.9 affect the two generator speed sensors, and in Zhang et al. (2018), a factor of 0.95 is considered, which equates to approximately a 6 rad/s offset. Note that, for the sake of simplicity, the same drift and offset constant values (δ_{drift} and δ_{offset}) are applied for both the pitch sensor and actuator subsystems. The same scenarios are considered for all the FOWTs in the farm, producing a total of $S = 18$ fault simulations. These simulations form the dataset $\hat{X}$ to be rescaled and divided into segments. Here, we consider a segment length of $n = 10$ measurements, with an overlapping of $n_o = 9$ observations. This segment duration is the shortest segment length that provides adequate classification results after a trial and error test. Shorter segments lose the time dependencies among different features and hinder the classification task. We consider this duration sufficiently short to provide timely fault detection while preserving the time dependencies. As a final step in the preprocessing, we dismiss the segments containing transitions from faulty to healthy states as they represent unreal situations.

Fig. 8 shows the measured pitch angle for the healthy and the two pitch sensor fault modes at WT_1 during a 25-s frame corresponding

Table 2
Description of the considered fault magnitudes.

Fault type	Magnitude
Pitch angle offset	$\delta_{\text{offset}} = 3$ [deg]
Pitch angle drift	$\delta_{\text{drift}} = 8$ [deg / s]
Generator speed offset	$\delta_{\text{offset}} = -5$ [rad / s]
Generator speed drift	$\delta_{\text{drift}} = 3$ [rad / s^2]

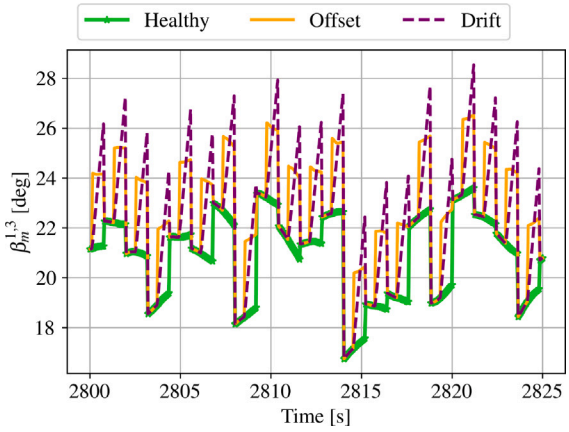


Fig. 8. Visualization of a 25-s long segment of the measured pitch angle (third blade) signal at WT_1. The green line indicates the healthy signal, whereas the blue and orange contain offset and drift faults, respectively.

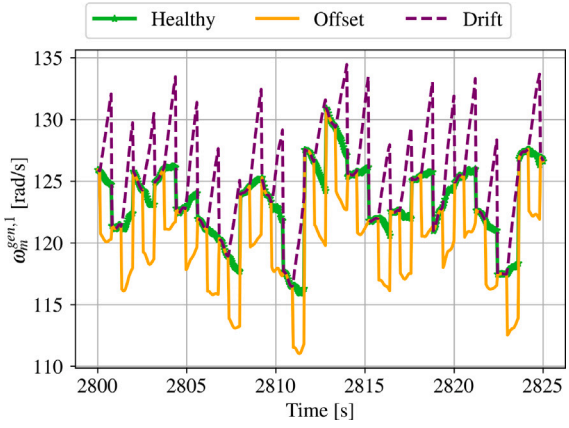


Fig. 9. Visualization of a 25-second-length segment of the measured generator speed signal at WT_1. The green line indicates the healthy signal, whereas the blue and orange contain offset and drift faults at the sensor, respectively.

to a mean wind speed of 23 m/s. It should be noted that, due to the interconnection of the WT subsystems, when the fault occurs in the pitch actuator, the associated reference and measured values are directly affected.

Similarly, Fig. 9 illustrates the generator sensor signal in the healthy state and under offset and drift faults at the same time frame as Fig. 8. Note that, even though Figs. 8 and 9 only display the variables directly impacted by the faults, additional variables (e.g., the generated power, P_{Gen}) are also influenced.

For the sake of clarity, Table 3 associates a descriptive name to each class, which will be used henceforth to refer to the different faults considered.

Fig. 10 represents the condition value $\hat{y}^{\text{true}}$ for the same 25-s time frame depicted in Fig. 8.

Given that we generated the data synthetically, it is balanced in terms of samples from each different class. Although the available experimental data would be imbalanced in real practice (mostly healthy

Table 3
Description of the fault scenarios for the classification problem. Each type of fault includes offset and drift effects.

Name	Value ($\hat{y}^{\text{true}}$)	Description
Healthy	0	No fault
F_1WT_1	1	Pitch sensor fault at WT_1
F_2WT_1	2	Generator sensor fault at WT_1
F_3WT_1	3	Pitch actuator fault at WT_1
F_1WT_2	4	Pitch sensor fault at WT_2
F_2WT_2	5	Generator sensor fault at WT_2
F_3WT_2	6	Pitch actuator fault at WT_2
F_1WT_3	7	Pitch sensor fault at WT_3
F_2WT_3	8	Generator sensor fault at WT_3
F_3WT_3	9	Pitch actuator fault at WT_3

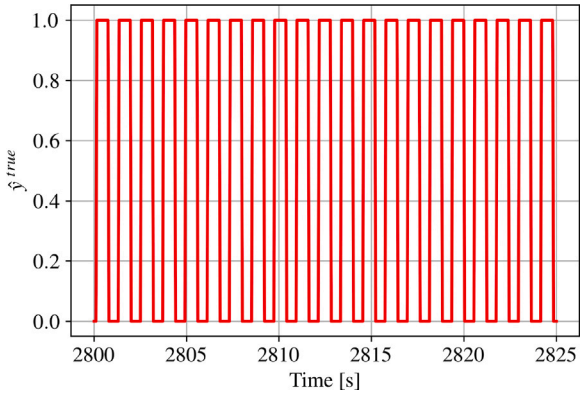


Fig. 10. Visualization of a 25-second-length fraction of the condition value $\hat{y}$ under the first scenario (pitch sensor fault at WT_1 represented in Fig. 8). Every 1.2 s there is a cycle of 0.5 s of healthy measurements followed by 0.7 s of faulty measurements corresponding to $\hat{y}^{\text{true}} = 1$.

signals would be measured), synthetic samples from a representative farm model must be produced to complement them and include damage scenarios. Using domain adaptation techniques aids in approximating synthetic and experimental samples to generate the entire database and keep it balanced (Wang et al., 2018).

4.3. DNN classifier training and validation

We first split the designed dataset into 70% training, 20% validation, and 10% test. The specifications for the architecture and the hyperparameters are selected according to previous experience of the authors and other studies on gradient-based training of Neural Networks (Bengio, 2000, 2012). We use Adam optimizer (Kingma and Ba, 2015) since it prevents getting trapped in local minima. Regarding the learning rate, a small value may result in excessively slow training or stacking into a local minimum (Glorot and Bengio, 2010). If the learning rate is too high, it may originate instability during training or divergence of the loss function (Gupta et al., 2015). After exploring different learning rates, we select a value of 2×10^{-05} as it yields adequate training and validation results. A larger value resulted in the loss function value wiggling considerably, turning into a non-smooth and unstable convergence curve. On the other hand, when using smaller values, the convergence slowed down, requiring many iterations to find the optimum.

We fix the number of epochs to a value after which no significant improvement is observed in the validation loss, preventing overfitting phenomena. Here, we attain non-improvable results after 1,500 epochs. The batch size specifies the number of samples that conform the batches (subsets of data) employed during training. By dividing the training data into batches, we enhance the efficiency of the minimization process and enable the generalization capability of the neural network. The minimum batch size is one, corresponding to a number of batches

Table 4
Summary of architecture and training specifications.

Property	Value
ID	$\mathcal{I}_\theta$
Input dimension	(10, 38)
Parameters	50,787
optimizer	Adam
Epochs	1,500
Batch size	512
Learning rate	2×10^{-5}

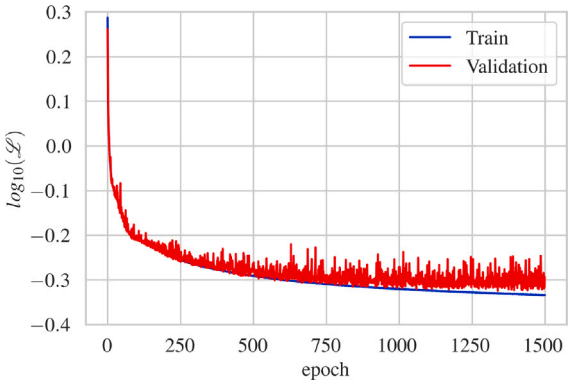


Fig. 11. Evolution of the loss function during training (logarithmic scale).

equal to the total number of available samples, which results in a computationally prohibitive training when handling large amounts of data. Contrarily, a very large value can compromise the memory limits depending on the computational resources of the employed GPU. A common practice in the computer science community is to choose the batch size as a power of two to align with the processing operations of GPUs. We employ a batch size of 512 samples, which provides an adequate balance between training speed and memory usage constraints.

Table 4 summarizes the main specifications. For illustrative purposes, Fig. 11 shows the evolution of the loss function during training for the train and validation datasets. The computational resources employed to train the NNs correspond to a workstation with the following characteristics: GPU: NVIDIA Quadro GV100; 32 GB VRAM CPU: Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz Memory: 258GiB. The computational time devoted to the training and validation stages was approximately five hours.

Regarding the validation of the algorithm, we compare the validation results against two standard classification algorithms, i.e., Random Forests and Gradient Boosting (Frank et al., 2020; Fawaz et al., 2018). Tables 5–7 gather the main classification metrics (Precision, Recall, and F1-score) (Sokolova and Lapalme, 2009).

Results reveal that the benchmark classification techniques provide similar results (the three methods fail in separating pitch sensor and actuator faults). On average, the proposed DNN approach slightly overcomes the two comparison techniques, with Gradient Boosting providing relatively suboptimal results. Besides, we demonstrate the robustness and generalizability of the method, we perform a cross-validation technique. We divide the dataset into $f = 5$ folds with stratification to ensure that a similar number of samples from each class participate in each fold (Kim et al., 2023). Table 8 shows the average accuracy in the classification task for each fold.

Finally, we evaluate $\mathcal{I}_{\theta^*}$ on the test dataset, which contains 10% of the available data in D. These data are unseen during the training and cross-validation tasks and contribute to demonstrating the generalization capability of the proposed method. The report in Table 9 summarizes the results according to the standard classification metrics

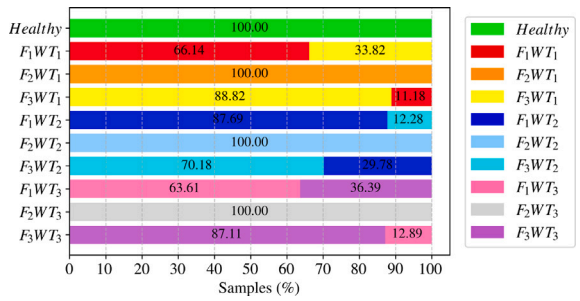


Fig. 12. Bar plot representation of the confusion matrix during testing.

and the support value, which indicates the number of testing samples in each class.

Note that, due to how the dataset is generated (see Section 4.2), the amount of healthy samples exceeds those from the different fault conditions, as shown in Table 9. One could notice that the healthy condition is successfully classified, which indicates that false positives are unlikely to occur. The same happens with generator torque faults of the three FOWTs (i.e., F_2WT_1, F_2WT_2, and F_2WT_3), which are unmistakably classified with a 100% success. However, considerably worse outcomes for the three FOWTs are obtained when analyzing pitch sensor and actuator faults. We explain this confusion as the pitch sensor and actuator operate in a closed loop, where the sensor measures the signal and sends it to the controller, and subsequently, the controller adjusts the reference value provided to the actuator for the pitch system. When a fault occurs in the sensor, it introduces an incorrect measurement to the controller, which modifies the reference fed into the actuator, even if the actuator is healthy. When a fault occurs in the actuator, it automatically induces an incorrect pitch angle in the affected blade, which the sensor measures afterward. Thus, both fault types end up affecting the reference and measured signals, misleading the classification.

Confusion matrices are one of the most common methods to evaluate the performance of multi-class classifiers (Grandini et al., 2020). Here, we depict them more accurately by building the color bar plot shown in Fig. 12. Each horizontal bar in the figure represents the amount of test samples associated with a particular class. It corresponds to the support value in Table 9. The bars are colored according to the portion of samples the DNN assigns (prediction) to each possible class. This representation highlights the underperformance perceived in the classification report. We observe that faults from each FOWT are well distinguished. Confusion occurs between each pair of pitch sensor and actuator faults.

The confidence of the predictions for each fault is analyzed as a final test. The Confidence Score (CS) is the value of the probability vector $\mathbf{y}_i$ for the true class. Hence, for a given class c ($c = 0, 1, \ldots, K$), the CS metric is defined as:

$$\mathrm{CS}_i = y^{pred}_{i,j=c}. \tag{12}$$

In the best scenario, any sample would receive a CS = 1, indicating a perfect classification. Fig. 13 represents the prediction confidence in the form of a histogram for each class in the testing data set. In these plots, it is possible to observe that most fault types present a strongly supported confidence score around the unit value. This means that a high probability value is assigned to the true class over the test samples. As expected, pitch sensor and actuator faults exhibit the most spread histograms, coinciding with the results shown, for example, in Figs. 13(b) or 13(g). Fig. 13 shows that the three turbines yield nearly identical classification results. Thus, with the proposed methodology, the orientation of deploying the two turbines — either in parallel, perpendicular to the wind direction, or in line with the wind direction — shows no significant difference, suggesting that the classifications of the three turbines are carried out independently.

Table 5
Comparison in terms of Precision for the proposed DNN against Random Forest and Gradient Boosting.

Classifier	Class									
	Healthy	F_1WT_1	F_2WT_1	F_3WT_1	F_1WT_2	F_2WT_2	F_3WT_2	F_1WT_3	F_2WT_3	F_3WT_3
Proposed DNN	1.00	0.85	1.00	0.73	0.75	1.00	0.85	0.83	1.00	0.70
Random Forest	1.00	0.72	0.99	0.77	0.71	0.99	0.76	0.70	0.99	0.74
Gradient Boosting	0.83	0.73	1.00	0.69	0.73	0.99	0.68	0.62	1.00	0.70

Table 6
Comparison in terms of Recall for the proposed DNN against Random Forest and Gradient Boosting.

Classifier	Class									
	Healthy	F_1WT_1	F_2WT_1	F_3WT_1	F_1WT_2	F_2WT_2	F_3WT_2	F_1WT_3	F_2WT_3	F_3WT_3
Proposed DNN	1.00	0.66	1.00	0.89	0.88	1.00	0.70	0.64	1.00	0.87
Random Forest	1.00	0.79	0.97	0.69	0.78	0.97	0.69	0.77	0.98	0.67
Gradient Boosting	0.99	0.53	0.85	0.61	0.52	0.83	0.62	0.64	0.83	0.44

Table 7
Comparison in terms of F1-Score for the proposed DNN against Random Forest and Gradient Boosting.

Classifier	Class									
	Healthy	F_1WT_1	F_2WT_1	F_3WT_1	F_1WT_2	F_2WT_2	F_3WT_2	F_1WT_3	F_2WT_3	F_3WT_3
Proposed DNN	1.00	0.74	1.00	0.80	0.81	1.00	0.77	0.72	1.00	0.78
Random Forest	1.00	0.76	0.98	0.73	0.74	0.98	0.72	0.73	0.98	0.70
Gradient Boosting	0.91	0.62	0.92	0.65	0.61	0.90	0.65	0.63	0.91	0.54

Table 8
Cross-validation results with $f = 5$ folds.

Fold	Avg. Precision	Avg. Recall	Avg. F1-Score
Fold 1	0.86	0.83	0.85
Fold 2	0.87	0.86	0.87
Fold 3	0.87	0.85	0.86
Fold 4	0.85	0.87	0.86
Fold 5	0.85	0.86	0.85

Table 9
Test classification report.

Class	Precision	Recall	F1-Score	Support
Healthy	1.00	1.00	1.00	4209
F1WT1	0.85	0.66	0.74	2794
F2WT1	1.00	1.00	1.00	2916
F3WT1	0.73	0.89	0.80	2925
F1WT2	0.75	0.88	0.81	2859
F2WT2	1.00	1.00	1.00	2812
F3WT2	0.85	0.70	0.77	2827
F1WT3	0.83	0.64	0.72	2894
F2WT3	1.00	1.00	1.00	2887
F3WT3	0.70	0.87	0.78	2856
Accuracy			0.87	29 979
Macro Avg	0.87	0.86	0.86	29 979
Weighted Avg	0.88	0.87	0.87	29 979

In general, the results reveal that identifying the faulty subsystem (pitch or generator) and the affected wind turbine is possible. However, distinguishing sensor from actuator faults exceeds the capability of the proposed method as they might be twisted.

4.4. Early fault classification test

The performance of the FDI strategy in the task of early fault classification is further explored by evaluating a short dataset where different faults are randomly introduced. This testing process attempts to represent the acquisition of new signals that would be received by the trained DNN and automatically evaluated to assess the farm condition.

Table 10 describes the introduced faults, including the time interval when they occur.

Additionally, Fig. 14 illustrates, by way of example, some of the variables in the dataset. Note that, for visualization purposes, the first 15 s of simulation that correspond to the transient state are removed from Figs. 14(a) and (b).

Table 10
Description of the testing fault scenarios introduced in a short-tem testing dataset.

Fault ID	Description	Occurrence interval
1	F_2WT_2 offset	80–82 s
2	F_1WT_1 offset	94–96 s
3	F_2WT_1 drift	108–110 s
4	F_3WT_2 drift	125–127 s
5	F_3WT_3 offset	134–136 s
6	F_1WT_3 drift	148–150 s

To analyze the performance of the FDI strategy, the probability vector assigned to each sample during this test is studied here. To this end, Fig. 15 compares the true probability (solid line) and the predicted probability (dotted line) of each class in the probability vector. In the figure, each line represents the evolution of each element in the probability vector $\mathbf{y}^{true}$, whereas the dots represent the elements in $\mathbf{y}^{pred}$. Note that the plots are constrained to show just the regions of the dataset where each fault occurs. The figure preserves the time scale given that we obtain a 0.5-s duration sample every 0.05 s given the overlapping of $n - 1 = 9$ observations (see Section 4.2). However, there is a delay equal to the sample length (0.5 s) as the first sample can only be taken once this time has elapsed.

One could notice that, for all the cases, before the fault occurs, the probability associated with the healthy state reaches the highest value (close to one). Then, when the fault appears, the healthy probability drops as the probability of the fault increases. This can easily be seen in, for example, Fig. 15(a), which covers the occurrence interval of the first fault (see Table 10). The figure shows that, after around 0.25 s, the faulty label exceeds a probability of 0.5, permitting a rapid identification. Note that the lack of prediction samples in transitioning from faulty to healthy states was justified in Section 4.2: they correspond to an unrealistic scenario in an operative FOWT farm since faults are not expected to disappear by themselves. This fault was successfully classified during the test, as expected from the results that had been analyzed previously.

However, it is possible to see in Fig. 15 the aforementioned faulty pitch sensor–actuator confusion. For example, let us focus on the case of Fig. 15(d), where fault F_3WT_2 occurs. During the transition from

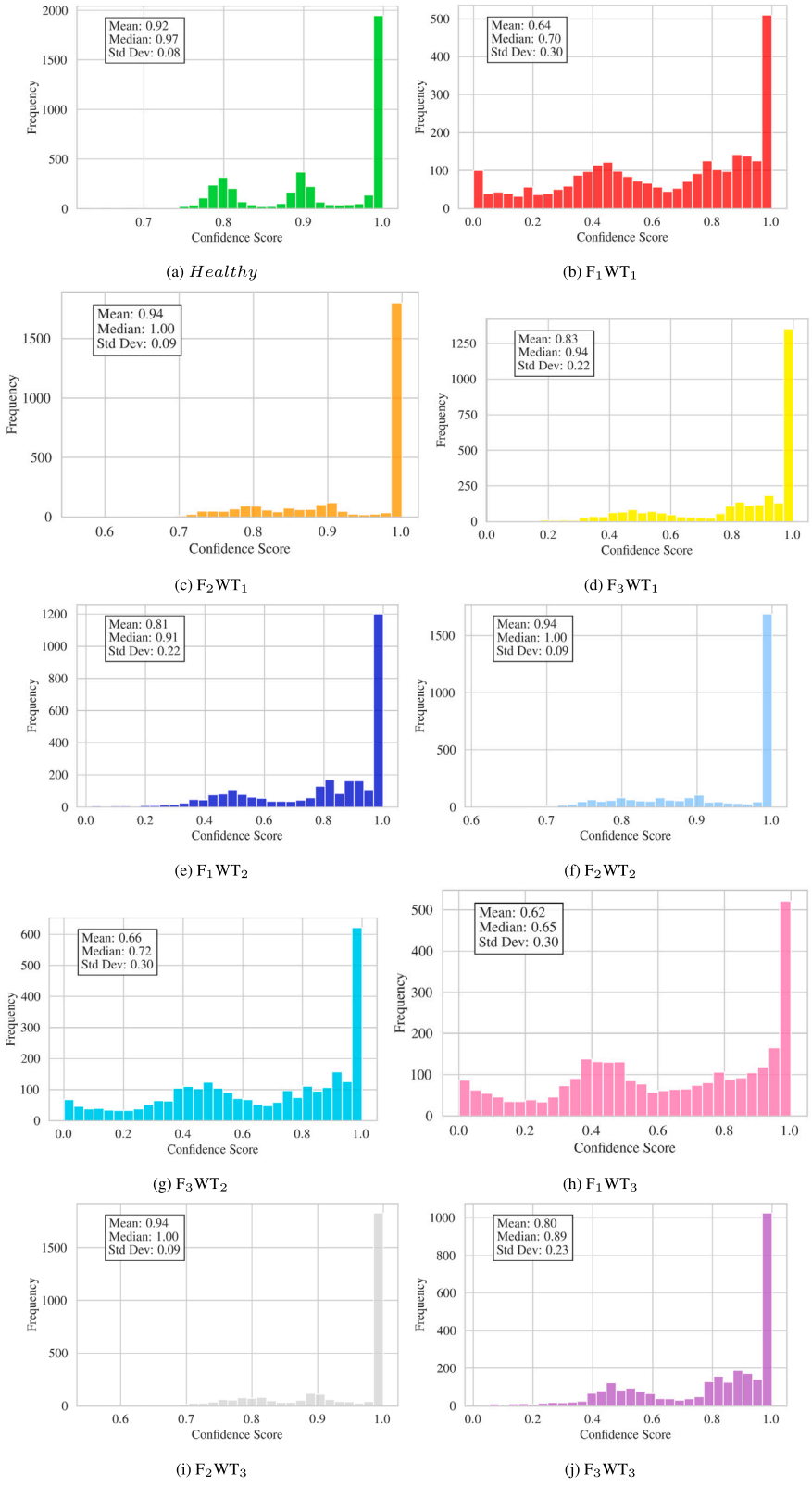
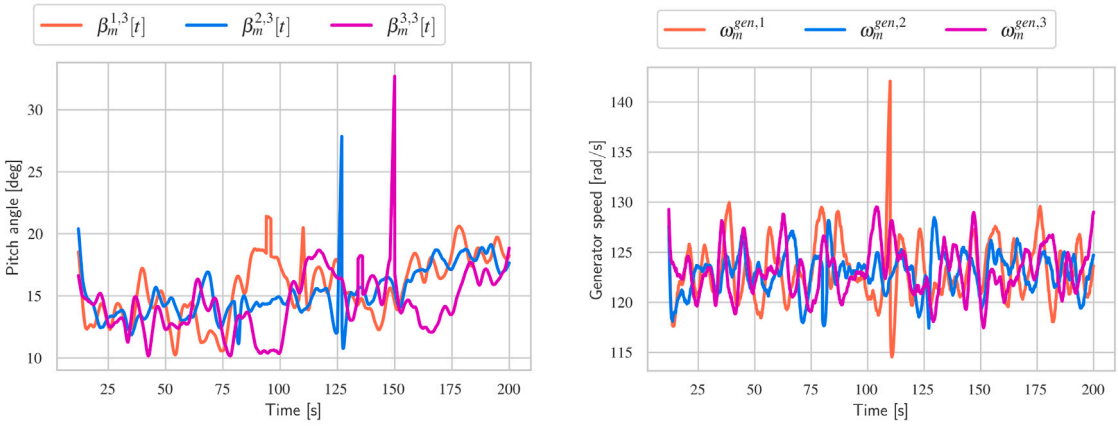


Fig. 13. Confidence scores for each class.

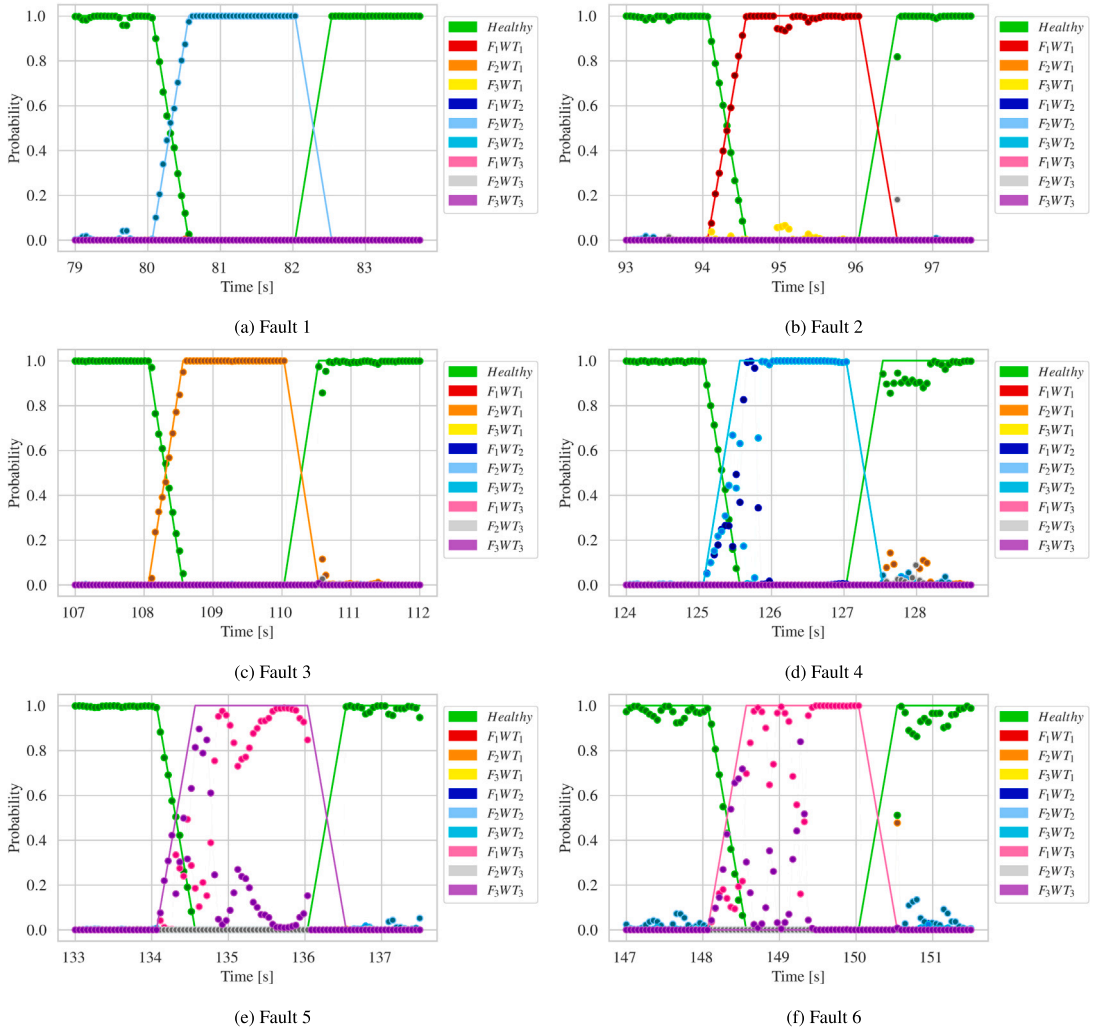
the healthy to the faulty state, both F_1WT_2 and F_3WT_2 states reach high values, misleading the classification. The DNN detects the fault (the healthy probability drops to zero), but it is unable to classify it successfully. The prediction indicates a fault in the pitch control system, but whether it comes from the sensor or the actuator is unknown. An interesting behavior is observed shortly after the fault occurs, where the true class (F_3WT_2) reaches a value of one with no further confusion with F_1WT_2. Note that similar results can be observed in Fig. 15(f) when fault F_1WT_3 appears. However, fault F_3WT_3 breaks this pattern and remains misclassified. This behavior requires deeper investigation



(a) Measured pitch angle signals from the three FOWTs.

(b) Measured generator speed signals from the three FOWTs.

Fig. 14. Measured pitch angle (a) and generator speed (b) signals from the three FOWTs.



(a) Fault 1

(b) Fault 2

(c) Fault 3

(d) Fault 4

(e) Fault 5

(f) Fault 6

Fig. 15. Probability vector plots for the six test faults.

to understand the physics governing the turbines under each fault and to successfully separate faults from sensors and actuators. This behavior is probably caused by the overlapping effects of a fault in the pitch sensor or in the actuator, given the interconnection between the actuator and sensor systems. Both fault types affect the measured quantities, causing the neural network to struggle to distinguish between the two. On the other hand, sensor faults also impact the actuator systems since the measured values are fed back to adjust the control actions. The results suggest that this closed-loop interconnection leads to the confusion shown in Fig. 15. A deeper analysis of the feature

selection may lead to identifying additional quantities that capture the characteristics of pitch sensor and actuator failures. Also, effectively accounting for the time dependencies in the affected signals under each type of fault is considered for future research in order to address this limitation.

In this work, we considered no simultaneous faults. However, in real practice, we may find scenarios with coexisting or cascade-induced faults that can hinder the classification task. When using the DNN classifier trained with single-fault scenarios and tested with multiple-fault scenarios, we obtained unsuccessful results. However, this is expected since the DNN is unable to extrapolate beyond the framework considered for training. The training samples only contain two coexisting scenarios, i.e., healthy and faulty, at those transition segments where the fault appears. For the rest of the training samples, a 100% probability is assigned to the true fault. Generating a complete training database, including multiple simultaneous faults, is the first step towards identifying such situations adequately. Besides, time-delayed concurring factors (cascade effect) need to be explored, where capturing time dependencies among the measured features is paramount.

5. Conclusions and future work

5.1. Conclusions

This work proposes a Deep Learning approach to classify non-critical faults at the different components of the control subsystems governing the behavior of FOWTs on a farm. We generate synthetic signals using a Simulink-based simulator representing a three-FOWT farm case study. We design a Deep Neural Network with 1D convolutional operations that receives short-term segments from the available signals and estimates the state of the farm as a probability vector. The probability vector indicates the probability of each possible state among ten possible classes, including the healthy state, and three faults for each FOWT: pitch sensor and actuator faults, and generator sensor. The results obtained during testing demonstrate the ability of the method to rapidly detect any fault by continuously evaluating the healthy state probability value.

The proposed work contributes to implementing rapid assessment strategies at the farm level, combining the signals from all the operating turbines interacting with each other during operation. The methodology provides a continuous diagnostic that complements the punctual information provided by in-situ inspections, which is even more relevant in the hardly accessible environment where floating offshore wind turbines operate.

However, successfully isolating the faults, i.e., identifying their origin, is challenging given the closed-loop interrelations among the control signals. The classification results reveal that the proposed method successfully identifies the faulty FOWT in the farm and the affected subsystem (pitch or generator). However, distinguishing pitch sensor and actuator faults has yet to be achieved and requires future research. This misclassification poses the main limitation of the proposed work.

Besides, this work handles only one fault at a time, but various coexisting faults or cascade effect situations may also occur and hinder the classification problem if not adequately handled. Faults occurring at other control system assets (e.g., the rotor) were not included in this work, focusing on pitch and generator signals as the most frequent fault origins. This poses another limitation that requires extending the work to the entire control system.

5.2. Future work

Based on the existing limitations of the present work, we identify various research lines to enhance the current approach. We consider a crucial future task to study the closed-loop time dependency between the signals under sensor and actuator faults. One of the main ways to achieve this goal is by implementing active-learning techniques (e.g., temporary Q-learning) to capture the interrelations between these two fault types effectively.

After the short-term simulation test, we obtained that the proposed methodology detects the faults with a time delay equal to the required frame to form the input sample. Thus, we consider future work designing a predictive approach that permits a timely detection of the faults. Such a methodology would precede a fault-tolerant control strategy to intervene in the affected subsystem and maintain optimal performance even in the presence of the fault. Designing extended databases that incorporate simultaneous faults (with delays in occurrence) and allow for the identification of multiple sources of failure is another important challenge to which we expect to contribute.

To demonstrate the effectiveness of the proposed method and its potential interest for offshore farm managers and owners, we will devote efforts to finding collaborations that give access to experimental monitoring data from instrumented wind farms. The combination of synthetic and experimental data will aid in ensuring the robustness and reliability of the assessments.

In relation to the real field assessment, we contemplate as a future line the extrapolation of the methodology from instrumented assets to additional operative assets of similar characteristics that lack a sufficient amount of data via transfer learning methods. This strategy might help generalize the assessment for analogous wind farms even if they are subject to data scarcity. In this context, the issue of class imbalance due to scarcity of data in faulty conditions cannot be fully solved via the data synergy among data from different turbines in a farm. An approach to solve this limitation is to couple the dataset with generative algorithms as Generative Adversarial Networks and conditional Variational Autoencoders. These approaches have been successfully and very recently applied to other domains (e.g. Liu et al. (2024)), resulting promising to support the fault diagnosis of mechanical systems. Regarding the strategy for acquiring and processing experimental field data, the application of the proposed method depends on the dataset used for training and validation. While fine-tuning the DL methods is mandatory when using real data, extending the present methodology to field data or experimental datasets is straightforward. The approach relies on quantities, such as pitch angles, power, torque, rotor speed, and wind velocity — typically measured, collected, and transferred by the SCADA and condition monitoring systems. Thus, implementing this method does not require modifications to the existing monitoring and sampling system, nor the installation of novel sensors. The simulator has undergone code-to-code validation against another simulator, OpenFast (Peña-Sanchez et al., 2022, 2024) and OpenFast has been also validated experimentally. Consequently, we can reasonably expect that the quality of the data underpinning our method's implementation is realistic. To quantify the uncertainty in the outcomes of the inverse problem when dealing with incomplete noisy data from real instrumentation systems, distributional learning approaches (e.g., Bayesian Variational Autoencoders (Rodriguez et al.)) are to be explored. All these considerations provide an optimistic outlook on the applicability of the present approach to real-world field data.

CRediT authorship contribution statement

Ana Fernandez-Navamuel: Writing – original draft, Validation, Supervision, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Yerai Peña-Sanchez:** Writing – original draft, Methodology, Funding acquisition, Data curation. **Vincenzo Nava:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Ana Fernandez-Navamuel would like to acknowledge the Basque Government funding within the HAZITEK program (ERROTAID project (ZL-2022/00317)) co-financed by the European Regional Development Fund (FEDER)), the ELKARTEK program (TCRINI project (KK-2023-0029)), and the BERC 2022–2025 program; the European Horizon (HE) with LIASON project (GA 101103698), FUTURAL project (101083958), and Marie Sklodowska-Curie Staff Exchange project RECHARGED (101086413).

Yerai Peña-Sanchez is funded by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreements N° 10103429.

Vincenzo Nava's work is funded by the Spanish Ministry of Science and Innovation projects with references TED2021-132783B-I00, PID2019-108111RB-I00 MCIN/AEI/10.13039/501100011033 (FEDER/AEI); the Spanish Ministry of Economic Affairs and Digital Transformation project IA4TES (MIA.2021.M04.0008); the "BCAM Severo Ochoa" accreditation of excellence (CEX2021-001142-S/MICIN/AEI/10.13039/501100011033); and the Basque Government, Spain through the BERC 2022–2025 program.

References

Aster, R.C., Borchers, B., Thurber, C.H., 2005. Parameter Estimation and Inverse Problems, Elsevier A ISBN: 0120656043, p. 316.

Aziz, U., Charbonnier, S., Berenguer, C., Lebranchu, A., Prevost, F., 2022. A Multi-Turbine Approach for Improving Performance of Wind Turbine Power-Based Fault Detection Methods. Energies 15 (8), http://dx.doi.org/10.3390/en15082806.

Bach-andersen, M., Rømer-odgaard, B., Winther, O., 2017. Deep learning for automated drivetrain fault detection. Wind Energy 21 (January), 1–13. http://dx.doi.org/10.1002/we.2142.

Badihi, H., Zhang, Y., Jiang, B., Pillay, P., Rakheja, S., 2022. A Comprehensive Review on Signal-Based and Model-Based Condition Monitoring of Wind Turbines: Fault Diagnosis and Lifetime Prognosis. Proc. IEEE 110 (6), 754–806. http://dx.doi.org/10.1109/JPROC.2022.3171691.

Bengio, Y., 2000. Gradient-based optimization of hyperparameters. Neural Comput. 12, 1889–1900. http://dx.doi.org/10.1162/089976600300015187.

Bengio, Y., 2012. Practical Recommendations for Gradient-Based Training of Deep Architectures. In: Neural Networks: Tricks of the Trade: Second Edition. Springer Berlin Heidelberg, pp. 437–478. http://dx.doi.org/10.1007/978364235289826.

Camps Echevarría, L., De Campos Velho, H.F., Becceneri, J.C., Da Silva Neto, A.J., Llanes Santiago, O., 2014. The fault diagnosis inverse problem with Ant Colony Optimization and Ant Colony Optimization with dispersion. Appl. Math. Comput. 227, 687–700. http://dx.doi.org/10.1016/j.amc.2013.11.062.

Caterini, A.L., Chang, D.E., 2018. Introduction and motivation. In: Deep Neural Networks in a Mathematical Framework. Springer International Publishing, ISBN: 978-3-319-75304-1, pp. 1–10. http://dx.doi.org/10.1007/978-3-319-75304-1_1.

Cho, S., Choi, M., Gao, Z., Moan, T., 2021. Fault detection and diagnosis of a blade pitch system in a floating wind turbine based on Kalman filters and artificial neural networks. Renew. Energy 169, 1–13. http://dx.doi.org/10.1016/j.renene.2020.12.116.

Cho, S., Gao, Z., Moan, T., 2016. Model-based fault detection of blade pitch system in floating wind turbines. J. Phys. Conf. Ser. 753 (9), http://dx.doi.org/10.1088/1742-6596/753/9/092012.

Cho, S., Gao, Z., Moan, T., 2018. Model-based fault detection, fault isolation and fault-tolerant control of a blade pitch system in floating wind turbines. Renew. Energy 120, 306–321. http://dx.doi.org/10.1016/j.renene.2017.12.102.

Ciampiconi, L., Elwood, A., Leonardi, M., Mohamed, A., Rozza, A., 2023. A survey and taxonomy of loss functions in machine learning. arXiv:2301.05579.

Ciuriuc, A., Rapha, J.I., Guanche, R., Domínguez-García, J.L., 2022. Digital tools for floating offshore wind turbines (FOWT): A state of the art. Energy Rep. 8, 1207–1228. http://dx.doi.org/10.1016/j.egyr.2021.12.034.

Dong, J., Verhaegen, M., 2011. Data driven fault detection and isolation of a wind turbine benchmark. In: IFAC Proceedings Volumes (IFAC-PapersOnline), vol. 44, IFAC, ISBN: 9783902661937, pp. 7086–7091. http://dx.doi.org/10.3182/20110828-6-IT-1002.00546.

Durakovic, G., del Granado, P.C., Tomasgard, A., 2023. Powering europe with north sea offshore wind: The impact of hydrogen investments on grid infrastructure and power prices. Energy 263, 125654. http://dx.doi.org/10.1016/j.energy.2022.125654.

Elsken, T., Metzen, J.H., Hutter, F., 2019. Neural architecture search: A survey. arXiv:1808.05377.

Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A., 2018. Deep learning for time series classification: A review. Data Min. Knowl. Discov. 33, 917–963, URL https://api.semanticscholar.org/CorpusID:52195012.

Fekih, A., Habibi, H., Simani, S., 2022. Fault diagnosis and fault tolerant control of wind turbines: An overview. Energies 15 (19), http://dx.doi.org/10.3390/en15197186.

Fekih, A., Mobayen, S., Chen, C.C., 2021. Adaptive robust fault-tolerant control design for wind turbines subject to pitch actuator faults. Energies 14 (6), http://dx.doi.org/10.3390/en14061791.

Frank, M., Drikakis, D., Charissis, V., 2020. Machine-learning methods for computational science and engineering. Computation 8 (1), 1–35. http://dx.doi.org/10.3390/computation8010015.

Fu, Y., Gao, Z., Liu, Y., Zhang, A., Yin, X., 2020. Actuator and sensor fault classification for wind turbine systems based on fast fourier transform and uncorrelated multi-linear principal component analysis techniques. Processes 8 (9), http://dx.doi.org/10.3390/pr8091066.

Gao, Z., Cecati, C., Ding, S.X., 2015. A survey of fault diagnosis and fault-tolerant techniques-Part I: Fault diagnosis with model-based and signal-based approaches. IEEE Trans. Ind. Electron. 62 (6), 3757–3767. http://dx.doi.org/10.1109/TIE.2015.2417501.

Gao, Z., Liu, X., 2021. An Overview on Fault Diagnosis, Prognosis and Resilient Control for Wind Turbine Systems. Processes 9 (2), http://dx.doi.org/10.3390/pr9020300.

Gao, Z., Sheng, S., 2018. Real-time monitoring, prognosis, and resilient control for wind turbine systems. Renew. Energy 116, 1–4. http://dx.doi.org/10.1016/j.renene.2017.10.059.

Gibbs, D., Jensen, P.D., 2022. Chasing after the wind? Green economy strategies, path creation and transitions in the offshore wind industry. Reg. Stud. 56 (10), 1671–1682. http://dx.doi.org/10.1080/00343404.2021.2000958.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. In: Proceedings of Machine Learning Research, vol. 9, PMLR, Chia Laguna Resort, Sardinia, Italy, pp. 249–256, URL https://proceedings.mlr.press/v9/glorot10a.html.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, p. 785, URL www.deeplearningbook.org.

Grandini, M., Bagli, E., Visani, G., 2020. Metrics for multi-class classification: an overview. arXiv:2008.05756.

Grunnet, J.D., Soltani, M., Knudsen, T., Kragelund, M.N., Bak, T., 2010. Aeolus toolbox for dynamics wind farm model, simulation and control. In: Proceedings of the European Wind Energy Conference and Exhibition. (EWEC), Warsaw, Poland.

Gupta, S., Agrawal, A., Gopalakrishnan, K., Narayanan, P., 2015. Deep learning with limited numerical precision. arXiv:1502.02551.

Habibi, H., Howard, I., Habibi, R., 2020. Bayesian fault probability estimation: Application in wind turbine drivetrain sensor fault detection. Asian J. Control 22 (2), 624–647. http://dx.doi.org/10.1002/asjc.1973, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/asjc.1973, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/asjc.1973.

Habibi, H., Howard, I., Simani, S., 2019. Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control : A review. Renew. Energy 135, 877–896. http://dx.doi.org/10.1016/j.renene.2018.12.066.

Hafner, M., Raimondi, P.P., 2021. Priorities and challenges of the EU energy transition: From the European Green Package to the new Green Deal. Russ. J. Econ. 6 (4), 374–389. http://dx.doi.org/10.32609/J.RUJE.6.55375.

Hasselmann, K., 1973. Measurements of wind wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP). Deutches Hydrographisches Inst. 8, 95.

Held, D.P., 2019. Inflow Measurements by Nacelle Mounted Lidars for Wind Turbine and Farm Control (Ph.D. thesis). Ph.D. thesis, DTU Wind Energy, https://orbit.dtu.dk/en/publications/inflow ....

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Netw. 2 (5), 359–366. http://dx.doi.org/10.1016/0893-6080(89)90020-8.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: 32nd International Conference on Machine Learning, ICML 2015, vol. 1, ISBN: 9781510810587, pp. 448–456, arXiv:1502.03167.

Isermann, R., 2005. Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. Springer Science & Business Media.

Jana, D., Patil, J., Herkal, S., Nagarajaiah, S., Duenas-Osorio, L., 2022. CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction. Mech. Syst. Signal Process. 169 (December 2021), 108723. http://dx.doi.org/10.1016/j.ymssp.2021.108723.

Jiang, G., Fan, W.P., Li, W., Wang, L., He, Q., Xie, P., Li, X., 2022. DeepFedWT: A federated deep learning framework for fault detection of wind turbines. Measurement: J. Int. Meas. Confed. 199 (June), http://dx.doi.org/10.1016/j.measurement.2022.111529.

Jonkman, J., Butterfield, S., Musial, W., Scott, G., 2009. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

Jonkman, B., Mudafort, R., Platt, A., Branlard, E., Sprague, M., Jonkman, J., Hayman, G., Vijayakumar, G., Buhl, M., Ross, H., et al., 2022. OpenFAST/openfast: OpenFAST v3. 1.0. Zenodo [code] 10.

Kim, M.-C., Lee, J.-H., Wang, D.-H., Lee, I.-S., 2023. Induction motor fault diagnosis using support vector machine, neural networks, and boosting methods. Sensors 23 (5), http://dx.doi.org/10.3390/s23052585, URL https://www.mdpi.com/1424-8220/23/5/2585.

Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. pp. 1–15, arXiv:1412.6980.

Lan, J., Patton, R.J., Zhu, X., 2018. Fault-tolerant wind turbine pitch control using adaptive sliding mode estimation. Renew. Energy 116, 219–231. http://dx.doi.org/10.1016/j.renene.2016.12.005, Real-time monitoring, prognosis and resilient control for wind energy systems.

Liu, Y., Ferrari, R., Wu, P., Jiang, X., Li, S., 2021. Fault diagnosis of the 10MW Floating Offshore Wind Turbine Benchmark : A mixed model and signal-based approach. Renew. Energy 164, 391–406. http://dx.doi.org/10.1016/j.renene.2020.06.130.

Liu, Y., Fu, S., Lin, L., Zhang, S., Suo, S., Xi, J., 2024. DECVAE: Data augmentation via conditional variational auto-encoder with distribution enhancement for few-shot fault diagnosis of mechanical system. Meas. Sci. Technol. 35 (4), 046104. http://dx.doi.org/10.1088/1361-6501/ad197c.

Liu, W.Y., Tang, B.P., Han, J.G., Lu, X.N., Hu, N.N., He, Z.Z., 2015. The structure healthy condition monitoring and fault diagnosis methods in wind turbines: A review. Renew. Sustain. Energy Rev. 44, 466–472. http://dx.doi.org/10.1016/j.rser.2014.12.005.

Mazare, M., Taghizadeh, M., Ghaf-Ghanbari, P., 2021. Fault tolerant control of wind turbines with simultaneous actuator and sensor faults using adaptive time delay control. Renew. Energy 174, 86–101. http://dx.doi.org/10.1016/j.renene.2021.04.077.

McMillan, D., Ault, G.W., 2007. Quantification of condition monitoring benefit for offshore wind turbines. Wind Eng. 31 (4), 267–285.

Nava, V., Ruiz-Minguela, P., Perez-Moran, G., Rodriguez-Arias, R., Lopez-Mendia, J., Villate-Martinez, J.-L., 2019. Installation, operation and maintenance of offshore renewables. In: Renewable Energy from the Oceans: From Wave, Tidal and Gradient Systems To Offshore Wind and Solar. In: Energy Engineering, Institution of Engineering and Technology, pp. 397–424. http://dx.doi.org/10.1049/PBPO129E_ch11.

Ng, E.Y.K., Lim, J.T., 2022. Machine Learning on Fault Diagnosis in Wind Turbines. Fluids 7 (12), http://dx.doi.org/10.3390/fluids7120371.

Odgaard, P.F., Johnson, K.E., 2013. Wind turbine fault detection and fault tolerant control - An enhanced benchmark challenge. In: 2013 American Control Conference. pp. 4447–4452. http://dx.doi.org/10.1109/ACC.2013.6580525.

Odgaard, P.F., Stoustrup, J., 2014. Frequency based fault detection in wind turbines. In: IFAC Proceedings Volumes (IFAC-PapersOnline), vol. 19, (3), IFAC, ISBN: 9783902823625, pp. 5832–5837. http://dx.doi.org/10.3182/20140824-6-za-1003.00955.

Odgaard, P.F., Stoustrup, J., Kinnaert, M., 2013. Fault-tolerant control of wind turbines: A benchmark model. IEEE Trans. Control Syst. Technol. 21 (4), 1168–1182. http://dx.doi.org/10.1109/TCST.2013.2259235.

Ozdemir, A.A., Seiler, P., Balas, G.J., 2011. Wind turbine fault detection using counter-based residual thresholding. IFAC Proc. Vol. 44 (1), 8289–8294. http://dx.doi.org/10.3182/20110828-6-IT-1002.01758, 18th IFAC World Congress.

Peña-Sanchez, Y., Penalba, M., Knudsen, T., Nava, V., Pardo, D., 2024. Development and validation of a health-aware floating offshore wind farm simulation platform (FOWLTY) for fault detection and mitigation. Wind Energy Eng. Res. submitted for publication.

Peña-Sanchez, Y., Penalba, M., Nava, V., 2022. Faulty wind farm simulation: An estimation/control-oriented model. Trends Renew. Energies Offshore 679–685.

Peña-Sanchez, Y., Penalba, M., Nava, V., Puig, V.P., 2023. Fault diagnosis of floating offshore wind farms, a benchmark case study. In: Proceedings of the IFAC World Congress 2023, Yokohama, Japan. pp. 2–9.

Pozo, F., Vidal, Y., 2016. Wind turbine fault detection through principal component analysis and statistical hypothesis testing. Energies 9 (1), http://dx.doi.org/10.3390/en9010003.

Qiao, W., Lu, D., 2015. A survey on wind turbine condition monitoring and fault diagnosis—Part II: Signals and signal processing methods. IEEE Trans. Ind. Electron. 62, 6546–6557.

Rahimilarki, R., Gao, Z., Jin, N., Zhang, A., 2022. Convolutional neural network fault classification based on time-series analysis for benchmark wind turbine machine. Renew. Energy 185, 916–931. http://dx.doi.org/10.1016/j.renene.2021.12.056.

Reder, M.D., Gonzalez, E., Melero, J.J., 2016. Wind Turbine Failures - Tackling current Problems in Failure Data Analysis. J. Phys. Conf. Ser. 753 (7), http://dx.doi.org/10.1088/1742-6596/753/7/072027.

Robertson, A., Jonkman, J., Masciola, M., Song, H., Goupee, A., Coulling, A., Luan, C., 2014. Definition of the Semisubmersible Floating System for Phase II of OC4. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

Rodriguez, O., Taylor, J.M., Pardo, D., Multimodal variational autoencoder (MVAE) for inverse problems in geophysics : Application to a 1D magnetotelluric problem.

Rojas, S., Maczuga, P., Muñoz-Matute, J., Pardo, D., Paszyński, M., 2024. Robust variational physics-informed neural networks. Comput. Methods Appl. Mech. Engrg. 425, 116904. http://dx.doi.org/10.1016/j.cma.2024.116904.

Rolfes, R., Tsiapoki, S., Häckell, M.W., 2014. Sensing solutions for assessing and monitoring wind turbines. Sensor Technologies for Civil Infrastructures, vol. 1, ISBN: 9781782422433, pp. 565–604. http://dx.doi.org/10.1533/9781782422433.2.565.

Rotondo, D., Nejjari, F., Puig, V., Blesa, J., 2012. Fault tolerant control of the wind turbine benchmark using virtual sensors/actuators. 45, http://dx.doi.org/10.3182/20120829-3-MX-2028.00185,

Saci, A., Cherroun, L., Hafaifa, A., Mansour, O., 2022. Effective fault diagnosis method for the pitch system, the drive train, and the generator with converter in a wind turbine system. Electr. Eng. 104 (4), 1967–1983. http://dx.doi.org/10.1007/s00202-021-01446-8.

Salic, T., Charpentier, J.F., Benbouzid, M., Boulluec, M.L., 2019. Control strategies for floating offshore wind turbine: Challenges and trends. Electronics (Switzerland) 8 (10), 1–14. http://dx.doi.org/10.3390/electronics8101185.

Scheu, M., Tremps Bolsa, L., Smolka, U., Kolios, A., Brennan, F., 2019. A systematic failure mode effects and criticality analysis for offshore wind turbine systems towards integrated condition based maintenance strategies. Ocean Eng. 176, 118–133. http://dx.doi.org/10.1016/j.oceaneng.2019.02.048.

Sharan, B., Jain, T., 2018. Actuator and Sensor Fault Diagnosis for Wind Energy Conversion Systems. In: 2018 15th International Conference on Control, Automation, Robotics and Vision. ICARCV 2018, IEEE, ISBN: 9781538695821, pp. 955–959. http://dx.doi.org/10.1109/ICARCV.2018.8581339.

Shengnan, T., Yuan, S., Zhu, Y., 2020. Data preprocessing techniques in convolutional neural network based on fault diagnosis towards rotating machinery. IEEE Access PP, http://dx.doi.org/10.1109/ACCESS.2020.3012182, 1–1.

Singh, S., Shashank, S., Hegde, S.S., Paul, T.K., Reddy, R., 2018. Drift fault accommodation system of a transport aircraft using Neural Network models. In: 2018 International Conference on Advances in Computing, Communications and Informatics. ICACCI 2018, IEEE, ISBN: 9781538653142, pp. 2464–2470. http://dx.doi.org/10.1109/ICACCI.2018.8554748.

Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manage. 45 (4), 427–437. http://dx.doi.org/10.1016/j.ipm.2009.03.002.

Sørensen, J.D., Sørensen, J.N. (Eds.), 2012. Wind Energy Systems: Optimising Design and Construction for Safe and Reliable Operation, vol. 1, Woodhead Publishing Series in Energy.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J., Nenadic, G., 2019. Machine learning methods for wind turbine condition monitoring: A review. Renew. Energy 133, 620–635. http://dx.doi.org/10.1016/j.renene.2018.10.047.

Talarek, K., Knitter-Piatkowska, A., Garbowski, T., 2022. Wind parks in Poland:New challenges and perspectives. Energies 15 (19), http://dx.doi.org/10.3390/en15197004.

Tang, M., Zhao, Q., Wu, H., Wang, Z., Meng, C., Wang, Y., 2021. Review and perspectives of machine learning methods for wind turbine fault diagnosis. Front. Energy Res. 9, http://dx.doi.org/10.3389/fenrg.2021.751066.

Tchakoua, P., Wamkeue, R., Ouhrouche, M., Slaoui-Hasnaoui, F., Tameghe, T.A., Ekemb, G., 2014. Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges. Energies 7 (4), 2595–2630. http://dx.doi.org/10.3390/en7042595.

Tutivén, C., Vidal, Y., Acho, L., Rodellar, J., 2018. Fault detection and isolation of pitch actuator faults in a floating wind turbine. IFAC-PapersOnLine 51 (24), 480–487. http://dx.doi.org/10.1016/j.ifacol.2018.09.620, 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2018.

Wang, D., Cui, P., Zhu, W., 2018. Deep asymmetric transfer network for unbalanced domain adaptation. In: AAAI Conference on Artificial Intelligence. URL https://api.semanticscholar.org/CorpusID:19219599.

Werbos, P., 1990. Backpropagation through time: what it does and how to do it. Proc. IEEE 78 (10), 1550–1560. http://dx.doi.org/10.1109/5.58337.

Wu, Y., Ma, X., 2022. A hybrid LSTM-KLD approach to condition monitoring of operational wind turbines. Renew. Energy 181, 554–566. http://dx.doi.org/10.1016/j.renene.2021.09.067.

Xiang, L., Wang, P., Yang, X., Hu, A., Su, H., 2021. Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism. Measurement 175 (February), 109094. http://dx.doi.org/10.1016/j.measurement.2021.109094.

Xiao, C., Liu, Z., Zhang, T., Zhang, X., 2021. Deep learning method for fault detection of wind turbine converter. Appl. Sci. (Switzerland) 11 (3), 1–22. http://dx.doi.org/10.3390/app11031280.

Yu, D., Chen, Z.M., Xiahou, K.S., Li, M.S., Ji, T.Y., Wu, Q.H., 2018a. A radically data-driven method for fault detection and diagnosis in wind turbines. Int. J. Electr. Power Energy Syst. 99 (November 2017), 577–584. http://dx.doi.org/10.1016/j.ijepes.2018.01.009.

Yu, W., Lemmer, F., Schlipf, D., Cheng, P.W., Visser, B., Links, H., Gupta, N., Dankemann, S., Counago, B., Serna, J., 2018b. Evaluation of control methods for floating offshore wind turbines. J. Phys. Conf. Ser. 1104 (1), http://dx.doi.org/10.1088/1742-6596/1104/1/012033.

Zhang, D., Qian, L., Mao, B., Huang, C., Huang, B., Si, Y., 2018. A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost. IEEE Access 6 (April 2018), 21020–21031. http://dx.doi.org/10.1109/ACCESS.2018.2818678.

Zhang, Y., Wang, K., Qian, X., Gendeel, M., 2019. Robust fault-detection based on residual K–L divergence for wind turbines. IET Renew. Power Gener. 13 (13), 2400–2408. http://dx.doi.org/10.1049/iet-rpg.2018.6190.

Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D., 2017. Convolutional neural networks for time series classification. J. Syst. Eng. Electron. 28 (1), 162–169. http://dx.doi.org/10.21629/JSEE.2017.01.18, arXiv:1603.06995.