POLITECNICO DI TORINO
Repository ISTITUZIONALE

Diagnosis of the health status of mooring systems for floating offshore wind turbines using autoencoders

*Original*
Diagnosis of the health status of mooring systems for floating offshore wind turbines using autoencoders / Gorostidi, N., Pardo, D., Nava, V.. - In: OCEAN ENGINEERING. - ISSN 0029-8018. - 287:(2023). [10.1016/j.oceaneng.2023.115862]

*Availability:*
This version is available at: 11583/2997026 since: 2025-01-29T14:39:14Z

*Publisher:*
Elsevier Ltd

*Published*
DOI:10.1016/j.oceaneng.2023.115862

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

15 November 2026

Ocean Engineering 287 (2023) 115862



Contents lists available at ScienceDirect

Ocean Engineering

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





# Diagnosis of the health status of mooring systems for floating offshore wind turbines using autoencoders

N. Gorostidi [a,b,*], D. Pardo [b,a,c], V. Nava [d,a]

[a] Basque Center for Applied Mathematics, Alameda Mazarredo 14, 48009 Bilbao, Spain
[b] University of the Basque Country (UPV/EHU), Barrio Sarriena, 48940 Leioa, Spain
[c] Ikerbasque (Basque Foundation for Sciences), Bilbao, Spain
[d] TECNALIA, Basque Research and Technology Alliance (BRTA), Astondo Bidea, Edificio 700, Derio, 48160, Spain

ARTICLE INFO

*Keywords:*
Floating offshore wind
Deep learning
Operation and maintenance
Inverse problem
Autoencoder

ABSTRACT

Floating offshore wind turbines (FOWTs) show promise in terms of energy production, availability, and sustainability, but remain unprofitable due to high maintenance costs. This work proposes a deep learning algorithm to detect mooring line degradation and failure by monitoring the dynamic response of the publicly available DeepCWind OC4 semi-submersible platform. This study implements an autoencoder capable of predicting multiple forms of damage occurring at once, with various levels of severity. Given the scarcity of real data, simulations performed in OpenFAST, recreating both healthy and damaged mooring systems, are used to train and validate the algorithm. The novelty of the proposed approach consists of using a set of key statistical metrics describing the platform's displacements and rotations as input layer for the autoencoder. The statistics of the responses are calculated at 33-minute-long sea states under a broad spectrum of metocean and wind conditions. An autoencoder is trained using these parameters to discover that the proposed algorithm is capable of detecting mild anomalies caused by biofouling and anchor displacements, with correlation coefficients up to 98.51% and 99.16%, respectively. These results are encouraging for the continuous health monitoring of FOWT mooring systems using easily measurable quantities to plan preventive maintenance actions adequately.

## 1. Introduction

Floating offshore wind (FOW) is one of the most rapidly growing forms of green power. As of 2021 though, there were only 113 MW in operation in Europe (REN21, 2021), although this figure is predicted to triple by the end of 2024, and escalate up to 10 GW by 2030 (WindEurope, 2021) and 150 GW by 2050 (IRENA, 2021). The current share of floating offshore wind, however, is residual if compared to bottom-fixed offshore and onshore wind, whose global operating capacities stand at over 35 and 700 GW, respectively (GWEC, 2021). Controversy around onshore wind turbines' noise and visual pollution, as well as potential for higher energy production, has encouraged governments and institutions to issue challenging plans for further development in the offshore sector (WindEurope, 2022). Spain (WindEurope, 2021; OffshoreWind.biz, 2021, 2022), Norway (Reuters, 2019), South Korea (Green Tech Media, 2020b,a), Ireland Offshore Engineer Digital (2021) and the UK (Quarz, 2020; Wind Power Monthly, 2020; CNBC, 2022) are only a few examples of countries already investing heavily in floating wind projects, some of which are estimated to provide a total capacity over 2 GW once completed. Major oil and gas companies are also applying their knowledge of floating offshore technologies on wind power, partially aiming at shifting their public perception heading towards a more sustainable world (Forbes, 2021).

Because of their reduced noise and visual impact, floating offshore wind turbines are designed to be larger in size than their onshore counterparts. This means that their swept area is also larger, in locations where wind is already faster and smoother. All of these factors contribute to higher power output. However, excessive costs still keep floating wind farms unprofitable. Although rapidly decreasing, the levelized cost of energy (LCOE) of floating offshore wind varies between €90/MWh and €120/MWh, that is, more than double the cost of fixed offshore wind power (The Conversation, 2020). Approximately one third of these costs are due to operation and maintenance (O&M) and other associated activities (Nava et al., 2019). New technologies, such as remotely operated vehicles, unmanned aerial vehicles (Sierra-Garcia et al., 2019), and machine learning algorithms (Bishop, 2006), become essential to automate inspection and maintenance tasks, thus reducing operating costs and ultimately making FOW a commercially viable business.

* Corresponding author at: Basque Center for Applied Mathematics, Alameda Mazarredo 14, 48009 Bilbao, Spain.
*E-mail address:* ngorostidi007@ikasle.ehu.eus (N. Gorostidi).

https://doi.org/10.1016/j.oceaneng.2023.115862
Received 14 June 2023; Received in revised form 11 September 2023; Accepted 13 September 2023
Available online 21 September 2023
0029-8018/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

In this work, we propose a method for the structural health monitoring (SHM) of floating wind turbine mooring lines. These systems are one of the most expensive components of FOWTs, and are much less studied than other subsystems, such as their towers (Kim et al., 2019) or blades (Ghoshal et al., 2000). This study formulates an inverse problem, in which samples of measurements from the dynamics of the floating platform are used to train a model, which can then make predictions of the structural health status of mooring lines. Solving inverse problems (Tarantola, 2005) allows for the early identification of damage. This is critical for SHM purposes, as late notification and maintenance lead to performance losses and increased downtime (Farrar and Worden, 2012). These factors crucially affect the overall budget of a FOW project, and thus need to be optimized.

Deep learning (DL) algorithms (Bishop, 2006) are among the most popular methods for solving forecasting and monitoring problems (Penalba et al., 2022). These models are usually represented by deep neural networks (DNNs), and have been widely used for mooring system design and SHM (García et al., 2010; Gordan et al., 2017; Salehi and Burgueño, 2018; Azimi et al., 2020; Chalapathy et al., 2018) because of their potential to approximate complex, highly non-linear functions that go beyond the capabilities of simpler, more traditional approaches. After training, DL algorithms can make instant predictions of the health status of a system, which is crucial in the context of damage detection.

Depending on the DNN's layout, researchers have been able to answer different questions. For instance, multi-layer perceptrons (MLP) are popularly used to detect damage because of their simplicity for implementation. Sidarta et al. (2018) used synthetic data, extracted from 7.5-hour-long sea state simulations, to train an MLP to identify mooring line breakage for the Heave and VIM suppressed (HVS) semi-submersible platform. The implemented DNN had four inputs, specifically long drift periods and means for floater surge and sway; and five outputs detecting damage for five different mooring lines. Because of its network layout and the definition of its damaged scenarios, this study detected ultimate failure rather than predicting mild damage before the system's performance was significantly affected.

Li and Choung (2017), Li et al. (2018) implemented an MLP to detect wide-banded mooring line fatigue for FOWTs. Similarly to the work presented here, they performed FAST (National Renewable Energy Laboratory, 2021) simulations on the semi-submersible platform developed within the DeepCWind project (Robertson et al., 2014; Wang et al., 2021), assuming a mooring system with three catenary lines. Their MLP had four environmental conditions as inputs and 121 outputs representing mooring line tension. They obtained very positive results, especially under severe sea states.

Bjø rni et al. (2021) tested the same turbine model studied in this work, but supported on a spar-type floating platform. They used much shorter simulations than those observed in aforementioned studies, from which they computed floater motions in 6 degrees of freedom (DOF), neglecting aerodynamic loads. Their MLP's input layer consisted of 360 neurons representing 60 s for each DOF. They applied sliding window techniques to predict 30 s of mooring line tension for each mooring line. Their network's size caused high DNN training costs of up to 250 h, which pose a series of challenges when extending the model to different platforms and FOWT designs.

Chung et al. (2020) designed a DNN to detect cross-section losses at different points on mooring lines for a tension leg platform (TLP). Their DNN's inputs were environmental conditions and floater responses for all six DOFs, and its eight output nodes represented damage to each of the TLP's mooring lines. Their study is limited to very severe cross-section changes happening to one individual line at a time. This, in practical terms, implies detecting failure after significant performance losses, and can result in inconsistencies when two or more mooring lines are damaged simultaneously.

MLP-based algorithms are mostly used to detect damage rather than predicting it. Other studies (Saad et al., 2021; Chen et al., 2020, 2021; Wang et al., 2022) apply convolutional neural networks to detect failure using image recognition (Janas et al., 2021) or long short-term memory (LSTM) networks to estimate future mooring damage (Xiang et al., 2021). In particular, Saad et al. (2021) compared MLPs and LSTMs to predict vessel mooring failure by inducing sudden line breakage. Chen et al. (2021) coupled an LSTM network with an autoencoder (AE) to detect anomalies in fixed-bottom offshore wind turbines. Using these algorithms, they predicted which variables were the most relevant for damage detection. Other machine learning techniques used for FOWT condition monitoring are reviewed in Martinez-Luengo et al. (2016), Joshuva et al. (2019), Ciuriuc et al. (2022), Fu et al. (2019) and Hameed et al. (2009).

In this article, we detect mooring failure by implementing an autoencoder whose inputs are time and frequency-domain statistics extracted from the floater's response in six DOFs. This approach, commonly referred to as Frequency Domain Decomposition (FDD) (Oliveira et al., 2021; Peña-Sanchez et al., 2022), allows for a computationally efficient analysis, as we describe the structural integrity of the mooring system using a reduced set of easily measurable variables with known physical meaning. Some examples of applications of FDD include SHM for vessel mooring systems (Low and Langley, 2006), gearboxes for onshore wind turbines (Feng and Liang, 2014; Feng et al., 2016; Antoniadou et al., 2015) and spar-buoy floaters (Ruzzo et al., 2016) using various predictive tools. Previous research by Gorostidi and Nava (2021) and Gorostidi et al. (2022) implemented 1-DOF and 6-DOF models, respectively, to predict one-hot encoded mooring failure using MLPs. This means that only one type of damage, and of a given severity, could be detected for each mooring line.

The main novelty of this study is to design a deep autoencoder, fed by a reduced set of physically meaningful modal inputs describing the response of the floater as in Gorostidi and Nava (2021), Gorostidi et al. (2022). This type of network allows to (a) identify faults under any combination of environmental conditions, and (b) detect mooring line damage at an early stage. This is achieved by building a set of faulty mooring systems with different severities, which are defined by a set of continuous damage coefficients. These coefficients can take any value in the interval [0, 1], and are sampled for each data point using normal distributions centered around zero. This leads to a training set containing mostly undamaged and mildly degraded samples. In that way, the resulting model, whose architecture was optimized using automated machine learning (AutoML) techniques (Shahriari et al., 2022), is capable of detecting low-severity damages, common in early-stage operation, with a much higher rate of accuracy. The network is also capable of predicting more than one type of failure occurring simultaneously, thus increasing its potential practical applicability.

In this initial study, we restrict to only two forms of degradation: biological fouling and anchor shifts. Other forms of degradation will be addressed in further research since they may be sensitive to different input parameters other than the harmonic-like dynamic measurements employed in this work.

FOW research stands at a much earlier stage than onshore wind, and hence samples of real data are scarce. We are therefore forced to build a synthetic training set, containing sea states representing different wind and wave conditions. We perform simulations in OpenFAST (National Renewable Energy Laboratory, 2021), a widely popular open-source tool for wind turbine simulation, to recreate the floater's response, and calibrate the solver's settings by comparing the obtained outcomes with computational fluid dynamics (CFD) studies (Benitz et al., 2014; Liu et al., 2019; Zhang and Kim, 2018; Lin and Yang, 2020; Galera-Calero et al., 2021), free decay tests, (Lee et al., 2018; Li et al., 2019; Coulling et al., 2013a; Liu et al., 2018; Stewart et al., 2012) and small-scale experiments (Chen et al., 2018; Coulling et al., 2013b) available in literature. Moreover, damages are synthetically modeled and imposed to the FOWT's mooring system. This means that its structural properties are manually changed to fit a finite set of damage scenarios. For

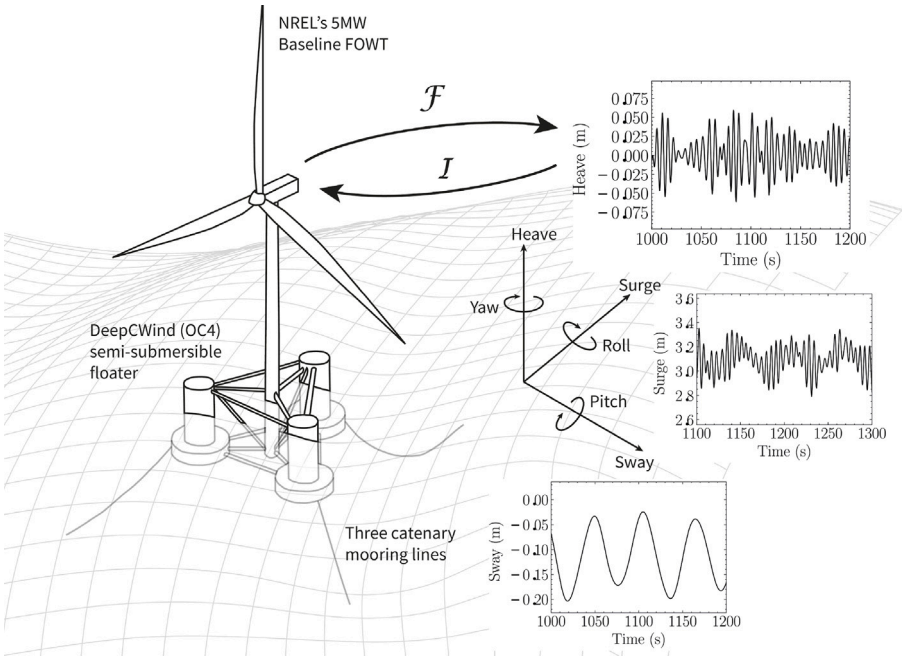



**Fig. 1.** NREL's 5 MW Baseline FOWT mounted on the DeepCWind OC4 floater.

instance, the added mass due to biological fouling degradation was considered to be uniform along the entire length of the mooring lines.

In this article, we considered a single model for our entire range of environmental conditions. Some studies have found that introducing different models for low and high-energy sea states may provide better predictions, as anomalies frequently seen in low-energy scenarios may be common to normal operation under high-energy sea states (Aizpurua et al., 2022). Nonetheless, our algorithm takes all environmental measurements as inputs. Thus, the resulting neural network is expected to provide robust predictions of the health status of mooring lines, even under low-energy sea states. The implementation of condition-informed models or previous clustering steps to discretize energy conditions into groups are left for future work.

The remainder of this article is structured as follows. In Section 2, we detail the case study, we illustrate both the FOWT design we employ and its floating platform, and we list the environmental conditions and structural parameters defining the healthy and degraded mooring systems. In this section, we also dive into the mathematical aspects of the DNN training and validation steps. In Section 3, we address the implementation of the described methodology, we discuss the program we run to generate training and validation data, and we describe the calibration process we carry out so that our results match those we have observed in literature. In this section, we detail the method we used to optimize the network's architecture, and we assess the performance of the model during the training stage. In Section 4, we discuss the outcomes of the simulation process, and we analyze the floater's response both in time and frequency domains. In this section, we also present the accuracy of the DNN after training. We finally extract some key conclusions and propose further work from our research in Section 5.

## 2. Method

### 2.1. Problem description

In this work, we implement a deep learning model for the identification of mooring line damage for floating offshore wind turbines. Let $\mathcal{F}$ be the operator that solves the system of partial differential equations (PDEs) governing the dynamics of the turbine's floater, such that

$$u = \mathcal{F}(P, r), \tag{1}$$

where $u$ describes the floater's response in the form of six degrees of freedom (surge, sway, heave, roll, pitch, and yaw); $P$ denotes the structural integrity of the platform's mooring lines, and $r$ describes the external conditions affecting the platform's behavior, including wave significant height $H_S$, peak period $T_P$, and wind velocity $V$.

We assume that, for any fixed environmental conditions, different structural properties yield different responses, and in that way we intend to find a correlation between the floater's response through time and its structural integrity. This is a typical assumption on supervised deep learning when no damaged data are available (Farrar and Worden, 2012). The approach we used is commonly referred to as *inversion*, as it reverts the classic PDE modeling problem. Here, and as presented in Fig. 1, an inverse operator $\mathcal{I}$ estimates the structural properties of the turbine's mooring system based upon the response of the platform such that

$$P = \mathcal{I}(u, r). \tag{2}$$

### 2.2. Parametrization of the mooring system

We first propose to build a parametrization of the floater's mooring lines. That is, we define the health status of the mooring system using two coefficients ranging from 0 to 1. Each of these describes the severity of a specific form of structural degradation. In particular, we consider two types of mooring line failure: a *biological fouling* coefficient value of 1 represents a simultaneous 10% increase in both the lines' mass per unit length and diameter, and represents the attachment of mussels, algae, and other marine organisms to the mooring lines. An *anchoring* coefficient equal to 1 implies a 20-meter shift of a line's anchor in any direction, assuming a flat seabed. The result of this is a change –either an increase or decrease– in the mooring system' stiffness. Any coefficient between 0 and 1 represents intermediate biofouling or anchoring degradation following a linear interpolation. Because of the current scarcity of experimental samples, we use OpenFAST (National Renewable Energy Laboratory, 2021) simulations that estimate the operator $\mathcal{F}^{FAST}$ describing the dynamic response of the parametrized floater $i$, that is:

$$u_i = \mathcal{F}^{FAST}(P_i, r_i). \tag{3}$$

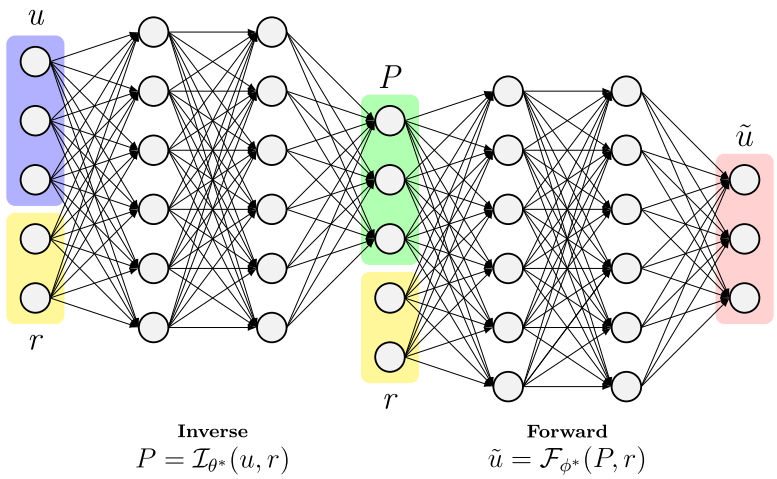



**Fig. 2.** Sketch of an autoencoder deep neural network for our SHM problem.

For an easier read, we hereinafter refer to $\mathcal{F}^{FAST}$ as $\mathcal{F}$, assuming the difference between the actual and estimated operators is negligible. Due to the facts that (a) we do not know the derivatives of $\mathcal{F}$, and (b) we want to detect damage at points not necessarily included in the training set, we use a deep neural network (DNN) to estimate both forward and inverse operators using a two-step loss function (Shahriari et al., 2021):

$$\begin{aligned} \mathcal{F}^*_\phi(P, r) &:= \arg\min_{\phi \in \Phi} \sum_{i=1}^{N} \|\mathcal{F}_\phi(P_i) - \mathcal{F}(P_i)\|_2, \\ \mathcal{I}^*_\theta(u, r) &:= \arg\min_{\theta \in \Theta} \|(\mathcal{F}^*_\phi \circ \mathcal{I}_\theta)(u) - u\|_2. \end{aligned} \tag{4}$$

The training step consists on first finding a set of weights $\phi^*$ that minimize the loss of the forward operator $\mathcal{F}^*_\phi$, then freezing all of its weights and biases, and implementing the autoencoder architecture presented in Fig. 2 to estimate the inverse operator $\mathcal{I}^*_\theta$. The proposed network includes the environmental conditions, $r := \{H_S, T_P, V\}$, as inputs for both encoder and decoder. This makes the inverse problem of detecting damage more unique, thus, helping the model distinguish between samples which may represent similar platform motions. The network calculates the total loss as the norm of the difference between the reconstructed response properties, $\tilde{u}$, and the actual ones obtained from simulations, $u$. The loss of the encoder side inherits the errors obtained in the training of the decoder. Proper training of the forward operator is therefore critical to maximize the accuracy of the inverse operator. This two-step training model (Shahriari et al., 2021) allows for the prediction of the health status of the turbine, $P$, located in the center of the autoencoder.

## 3. Implementation

### 3.1. Data set generation

The data set contains samples of responses obtained from OpenFAST simulations under different wind and wave conditions, as described in Table 1. The ranges of environmental conditions presented therein do not correspond to a specific site, but represent the whole operational region of FOWTs, as we include conditions that result in very low wave energy fluxes (WEF) (under 5kW/m) and very high WEF values (over 40kW/m) (Penalba et al., 2017; Blanco et al., 2021). We recreate the displacements and rotations of the baseline 5 MW FOWT designed by NREL, mounted on the semi-submersible floater developed within the DeepCWind project (Robertson et al., 2014) (see Fig. 1). Data set features include the statistical descriptors of the platform's displacements and rotations presented in Table 2. We described the stable time domain signal, which we assume is stationary, with its mean, $\bar{x}$, and standard deviation, $\sigma$. To sufficiently describe the frequency spectra, we obtain two peak frequencies, $f_1$ and $f_2$, as one of the peaks corresponds to the natural frequency of the system in the translational horizontal DOFs, while the other is affected by environmental conditions (Low and Langley, 2006; Benitz et al., 2014). Plus, we use the zero-th momentum, $m_0$, of the response spectra to assess the magnitude of the frequency peaks and the intensity of all frequencies in the spectra. This modal approach (Ruzzo et al., 2016; Black et al., 2022) reduces the size of the data set to just a few parameters for each sample, minimizing storage requirements and training costs while describing the platform's dynamics within acceptable accuracy levels.

**Table 1**
Setup of our simulation process and ranges for environmental conditions.

| Property | Value | Units |
|---|---|---|
| Simulation time | 2000 | s |
| Time step | 0.025 | s |
| Transient time | 500 | s |
| Significant height, $H_S$ | 2–15 | m |
| Peak period, $T_P$ | 1–15 | s |
| Wind Velocity, $V$ | 1–30 | m/s |

We select a combination of environmental conditions for each simulation by combining values of $H_S$, $T_P$, and $V$, each of them being defined as arrays of evenly-spaced values within the ranges presented in Table 1. Damage is generated by choosing a combination of biofouling ($y_0$) and anchoring ($y_1$) coefficients randomly within the set of sampled points presented in Fig. 3. These combinations arise from coupling normal distributions for both types of damage. These distributions are centered around zero for both $y_0$ and $y_1$. The idea behind this is that, by using this sampling technique instead of a completely random choice, the data set contains many more samples of undamaged and mildly damaged mooring lines. This, once the algorithm is trained, should allow for the identification of damage at an earlier stage. Using a finite number of damage severities, this process produces a surface similar to the one displayed in Fig. 3a, which shows the probability $p$ for the program to select each combination of damage coefficients. Once these coefficients are defined, we use a set of preprocessing scripts to manually adjust the corresponding structural parameters in OpenFAST's mooring dynamics module *MoorDyn* to fit specified mooring system setup, whose baseline properties and coordinates are presented in Tables 3 and 4, respectively.

Algorithm 1 details the process of generating the training, validation, and testing data sets. We performed a total of 102,229 OpenFAST simulations in parallel batches, using 120 Intel Xeon (R) E5-2680, 2.70 GHz CPUs (Donostia International Physics Centre, 2022). We found this number of cases to be sufficient to meet the minimum data

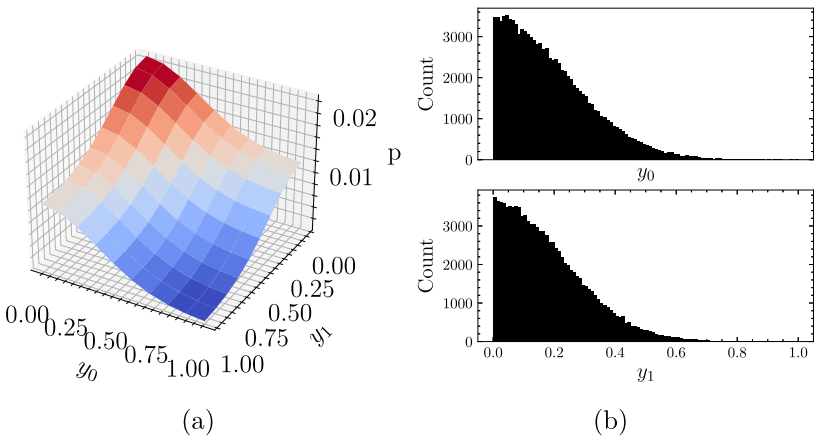



**Fig. 3.** Damage parametrization (a) surface representing the probability of selecting each combination of coefficients $y_0$ and $y_1$, and (b) distributions of coefficients across the entire training set for biofouling and anchoring on top and bottom, respectively.

**Table 2**
Statistical descriptors of the floater's response. The thresholds separating high ($f1$) and low ($f2$) range frequencies are 0.016 Hz for all degrees of freedom.

| Surge | Sway | Heave | Roll | Pitch | Yaw |
|---|---|---|---|---|---|
| $\bar{x}_{surge}$ | $\bar{x}_{sway}$ | $\bar{x}_{heave}$ | $\bar{x}_{roll}$ | $\bar{x}_{pitch}$ | $\bar{x}_{yaw}$ |
| $\sigma_{surge}$ | $\sigma_{sway}$ | $\sigma_{heave}$ | $\sigma_{roll}$ | $\sigma_{pitch}$ | $\sigma_{yaw}$ |
| $f_{1,surge}$ | $f_{1,sway}$ | $f_{1,heave}$ | $f_{1,roll}$ | $f_{1,pitch}$ | $f_{1,yaw}$ |
| $f_{2,surge}$ | $f_{2,sway}$ | $f_{2,heave}$ | $f_{2,roll}$ | $f_{2,pitch}$ | $f_{2,yaw}$ |
| $m_{0,surge}$ | $m_{0,sway}$ | $m_{0,heave}$ | $m_{0,roll}$ | $m_{0,pitch}$ | $m_{0,yaw}$ |

**Table 3**
Baseline mooring properties, assuming a three catenary line system.

| Property | Value | Units |
|---|---|---|
| Number of lines | 3 | – |
| Segments per line | 20 | – |
| Line diameter | $7.66 \cdot 10^{-2}$ | m |
| Mass density | $1.13 \cdot 10^{2}$ | kg/m |
| Line stiffness | $7.54 \cdot 10^{8}$ | N |
| Unstretched length | $8.35 \cdot 10^{2}$ | m |

**Table 4**
Undamaged mooring line coordinates.

| Point | X (m) | Y (m) | Z (m) |
|---|---|---|---|
| Line 1 Anchor | 418.80 | 725.38 | −200.00 |
| Line 1 Fairlead | 20.43 | 35.39 | −14.00 |
| Line 2 Anchor | −837.60 | 0.00 | −200.00 |
| Line 2 Fairlead | −40.87 | 0.00 | −14.00 |
| Line 3 Anchor | 418.80 | −725.38 | −200.00 |
| Line 3 Fairlead | 20.43 | −35.39 | −14.00 |

**Table 5**
Properties of the free decay tests for simulation validation.

| Test | Initial conds. | Sim. length | External conds. |
|---|---|---|---|
| Surge | +22 m | 20 min | No wind, nor waves |
| Sway | +22 m | 20 min | No wind, nor waves |
| Heave | +6 m | 5 min | No wind, nor waves |
| Roll | +8° | 5 min | No wind, nor waves |
| Pitch | +8° | 5 min | No wind, nor waves |
| Yaw | +8° | 15 min | No wind, nor waves |

set size requirements for our problem (Alwosheel et al., 2018). The elapsed time for all simulations was approximately 42 h.

### 3.2. Calibration of the solver

To calibrate our solver's settings and the structural properties of both turbine and floater, we carried out individual free decay tests for each degree of freedom, comparing the obtained curves to reference studies by Liu et al. (2018). In their research, they imposed the initial conditions presented in Table 5 to test the behavior of the DeepCWind floater with a single-point mooring (SPM) arrangement, characterized by a single fairlead for all lines. They compared this design with respect to the more common multi-point mooring (MPM) system, which is the one we implement in this article.

The baseline (Jonkman et al., 2009) and calibrated properties are shown in Table 6. Considering these properties, the results of the free decay tests are presented in Fig. 4, where it can be seen that our numerical results match the reference data with great accuracy. This is especially critical for the surge and pitch degrees of freedom (see Figs. 4a and 4e, respectively), as they are more affected by the assumed $x$-direction of the waves.

**Simulation step**

| Step | Description |
|---|---|
| 1 | Generate as many subdirectories as CPUs to store simulation files |
| 2 | Define number of simulations & copy baseline OpenFAST files |
| 3 | Check for damage (50% of the lines are healthy, 50% are damaged)<br>→ if so, define damage coefficients (random within distribution)<br>→ if not, set damage coefficients as (0, 0) |
| 4 | Define environmental conditions (random within ranges) |
| 5 | Assign combinations of damage coefficients to simulation files |
| 6 | Simulation loop for each CPU: $i = 1$<br>· Rewrite *MoorDyn*, *HydroDyn*, and *InflowWind* files<br>· Run OpenFAST simulation<br>· Obtain time series and frequency spectra<br>· Extract and write modal parameters in a CSV library<br>· $i = i + 1$ |
| 7 | Assemble libraries from all subdirectories into main CSV library |

Algorithm 1: Generation of training and validation samples using OpenFAST.

### 3.3. DNN training

In the training step, we use the data we have generated to minimize the loss function given by Eq. (4), following the process shown on Algorithm Section 3.3. For that purpose, we use TensorFlow 2.5.0 (Abadi et al., 2015). After reading, scaling, and splitting the generated data

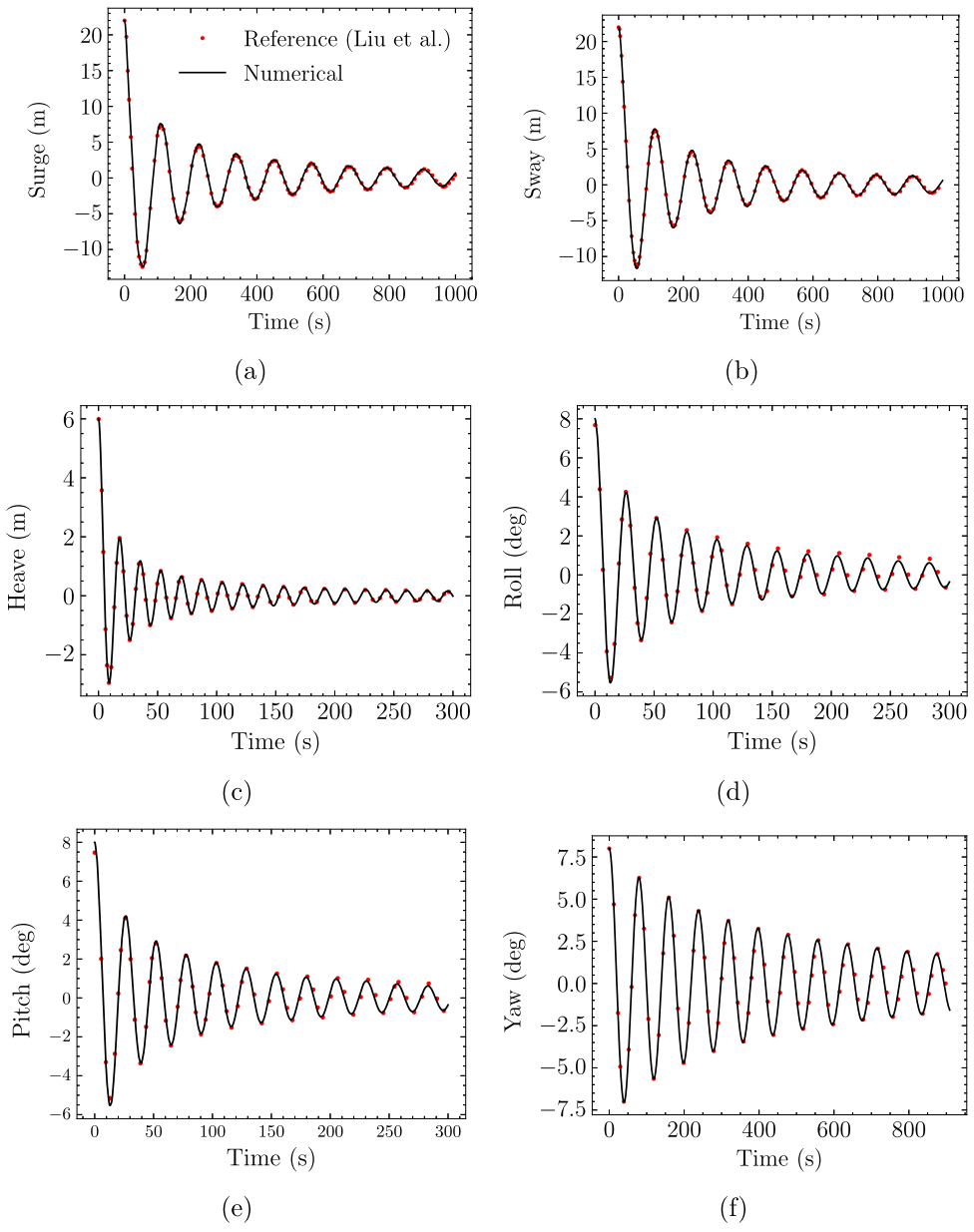



**Fig. 4.** Results of the performed free decay tests with respect to reference studies (Liu et al., 2018) in (a) surge, (b) sway, (c) heave, (d) roll, (e) pitch, and (f) yaw.

**Table 6**
Setup of our simulation process and ranges for environmental conditions (RK4: Runge–Kutta 4th order, ABM4: Adams–Bashforth–Moulton 4th order).

| Property | Original | Value | Units |
|---|---|---|---|
| Time step | 0.0125 | 0.025 | s |
| Interpolation | Linear | Quadratic | – |
| Integration method | RK4 | ABM4 | – |
| N blades | 3 | 3 | – |
| Tip radius | 63 | 63 | m |
| Hub radius | 1.5 | 1.5 | m |
| Overhang | −5.02 | −5 | m |
| Tower height | 87.6 | 90 | m |
| Nacelle mass | 2.4E+06 | 2.4E+06 | kg |
| Hub mass | 56,780 | 56,780 | kg |
| Hub inertia | 115,926 | 115,900 | kg m$^2$ |
| Generator inertia | 534.116 | 534.16 | kg m$^2$ |

into training, validation, and testing sets, we define the architecture of the forward side of the autoencoder, as well as a set of hyperparameters for its training. These include optimizer and initial learning rate (ILR). We establish the training routine by defining its batch size and number of epochs, and then we train the decoder.

Once the loss of the forward operator is minimized, we save the decoder's weights and biases. The idea behind this is that we can test multiple architectures for the inverse side of the autoencoder without having to retrain the forward side, thus increasing the computational efficiency of the program.

Before training the complete network, we define the encoder's architecture and the hyperparameters describing its training in the same way as for the decoder. We set the forward side's weights and biases as *non-trainable* and train the autoencoder. Once training is complete, the program predicts both damage coefficients for all testing samples and computes the correlation coefficient $r^2$.

We use AutoML (Shahriari et al., 2022) techniques to train a number of different architectures for both the network's encoder and decoder. We define a series of candidate architectures for $\mathcal{F}$ and $\mathcal{I}$ to find out what combination best describes the data set's variability. The tested architectures vary in both number of hidden layers and neurons per layer. The smaller networks usually underfit the data, behaving well

| Training step | |
|---|---|
| Step | Description |
| 1 | Define architectures for $\mathcal{F}$ and $\mathcal{I}$ |
| 2 | Read data<br>· Assign data set columns to $u$, $P$, and $r$<br>· Split in training (70%), validation (20%), and testing (10%) sets<br>· Scale data |
| 3 | Train forward operator $\mathcal{F}$<br>· Define architecture<br>· Define hyperparameters (Adam optimizer, ILR=0.005)<br>· Define training parameters (Batch size=512, 1,500 epochs)<br>· **Run training**<br>· Save weights<br>· Plot and save losses |
| 4 | Train inverse operator $\mathcal{I}$<br>· Load forward model and weights and set as non-trainable<br>· Define hyperparameters (Adam optimizer, ILR=0.005)<br>· Define training parameters(Batch size=512, 1,000 epochs)<br>· **Run training**<br>· Plot and save losses<br>· Compute correlation coefficient $r^2$ |
| 5 | Select combination of $\mathcal{F}$ and $\mathcal{I}$ with highest $r^2$ |
| 6 | Make predictions of damage coefficients for testing data using $\mathcal{I}$ |

Algorithm 2: DNN training algorithm.

from a numerical point of view but achieving lower accuracy rates. The larger architectures usually cause overfitting, which leads to poor performance against unseen data as the model obtains a function that is too aligned with the specific training data. The architecture presented in Table 7 provides the best performance metrics out of all the tested candidate layouts. The model's encoder consists of six hidden layers, whose inputs are the 30 statistical descriptors calculated from the floater's six displacements and rotations, plus the three parameters defining external conditions. All the encoder's neurons are fully-connected, and the values they store are bounded by the rectified linear unit (ReLU) activation function (Nair and Hinton, 2010). The encoder's output layer estimates the two coefficients describing the structural integrity of the mooring system, $P$, using a sigmoid function (Han and Moraga, 1995). The decoder consists of six hidden layers of increasing width to recreate the displacements and rotations of the floating platform. To address potential overfitting issues, we have added two dropout layers preceding the last two layers of the decoder, each of them dropping a random 10% of their neurons on each iteration. Once again, the hidden layers are fully-connected among one another, and activated using the ReLU function. The sigmoid function bounds the values stored by the decoder's output neurons, which represent the recreated modal measurements $\tilde{u}$. Although there are a few rules of thumb when it comes to hyperparameter settings, e.g. related to activation functions, optimizers, initial learning rates, etc., the usual practice consists on testing the performance of different educated guesses and finding a satisfactory combination of hyperparameters both in terms of computational costs and prediction accuracy (Yang and Shami, 2020).

After defining an adequate DNN architecture, we carry out longer training and validation processes to minimize the network's loss. Proper monitoring of the model's performance as training progresses is key to identify any undesired behaviors, including overfitting, divergences, numerical instabilities, etcetera. Fig. 5a shows the evolution of the decoder's losses as its training takes place. Fig. 5b presents the algorithm's losses after freezing the forward operator's weights and biases and plugging them into the decoding side of the model, as stated in Eq. (4). The results presented in Fig. 5 denote the following characteristics regarding the network's behavior as training progresses:

**Table 7**
Characteristics of the selected encoder–decoder architecture.

| Autoencoder architecture | | |
|---|---|---|
| Architecture design | Input neurons | 33 |
| | Encoder layers | 6 (46, 62, 48, 24, 12, 6) |
| | Bottleneck neurons | 5 |
| | Decoder layers | 6 (7, 18, 36, 60, 90, 85) |
| | Output neurons | 30 |
| | Layer connection | Fully-connected layers |
| Test results | $\mathcal{F}$ training time | 14 min 57 s |
| | $\mathcal{I}$ training time | 11 min 10 s |
| | Training time | 26 min 07 s |
| | $r^2$ | 0.9288 |
| Hyperparams. | Activation functions | ReLU, Sigmoid |
| | Optimizer | Adam |
| | Epochs (forward) | 1,500, 1,000 |
| | Lrn. rate (forward, total) | 0.005, 0.005 |

(a) the expected, asymptotic decrease of the network's losses as training progresses, which indicates a successful loss minimization process for both forward and inverse operators;

(b) a stabilized minimum loss at the end of the training stage, which implies that training is complete; and

(c) an adequately similar behavior of the model's losses for both training and validation sets, which suggests we successfully avoid overfitting issues.

## 4. Results and discussion

Once the autoencoder is trained, we employ a new set of data to predict the health status of the DeepCWind mooring system, $P$, based on the statistical descriptors of the floater's response, $u$. This implies using the weights and biases of the network's encoder only, that is, the left half of the DNN presented in Fig. 2.

Figs. 6a and 6b display crossplots corresponding to predictions of biofouling and anchoring failure, respectively. These represent how well the algorithm understands each kind of structural damage individually, regardless of the presence of other forms of damage in the samples. We observe an adequate correlation between the actual coefficients, usually referred to as *ground truth*, and the predicted values, especially in the case of anchor displacements. The model itself exhibits a squared correlation coefficient $r^2$ of 0.9288, which implies that the model explains nearly a 93% of the data set's variability.

To simplify the interpretation of these results, we categorized each form of damage into five classes. Using this approach, we may observe a minimum severity threshold at which significant performance losses occur and therefore maintenance operations should be performed. Figs. 7a and 7b present the behavior of this model for both biofouling and anchoring issues, respectively. In these figures, we assign the testing set samples to five severity categories according to their predicted and true damage coefficients: very mild (up to and including 0.2), mild (0.4), intermediate (0.6), severe (0.8), and very severe (1.0). The diagonal formed by the white squares shows that the majority of samples are classified correctly, and the fact that lower severity categories include more data proves that we have trained the model primarily with undamaged and mildly damaged scenarios, as presented in Fig. 3. The confusion matrices displayed on Fig. 7 show promising results when dealing with undamaged or severely damaged mooring lines. In the case of biological fouling, the precision for both *very mild* and *very severe* classes stand at 91.2% and 93.2%, respectively. For anchoring issues, predictions are even more precise, with these metrics increasing to 95.9% and 96.3%, respectively. Intermediate classes show a more disperse behavior, with mild, intermediate, and severe damage accuracy rates all under 80% for biofouling damage, and under 90% for anchor shifts.

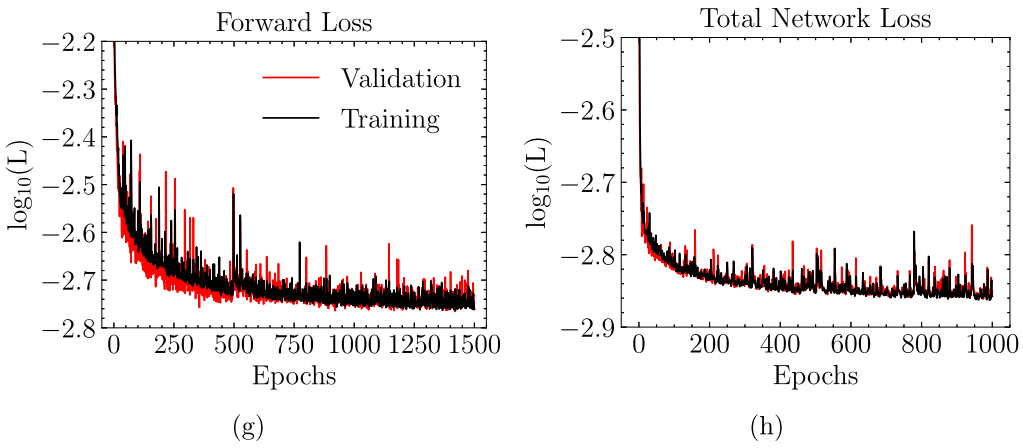



**Fig. 5.** Loss evolution for training and validation data sets as training progresses for both (a) forward operator and (b) total network.

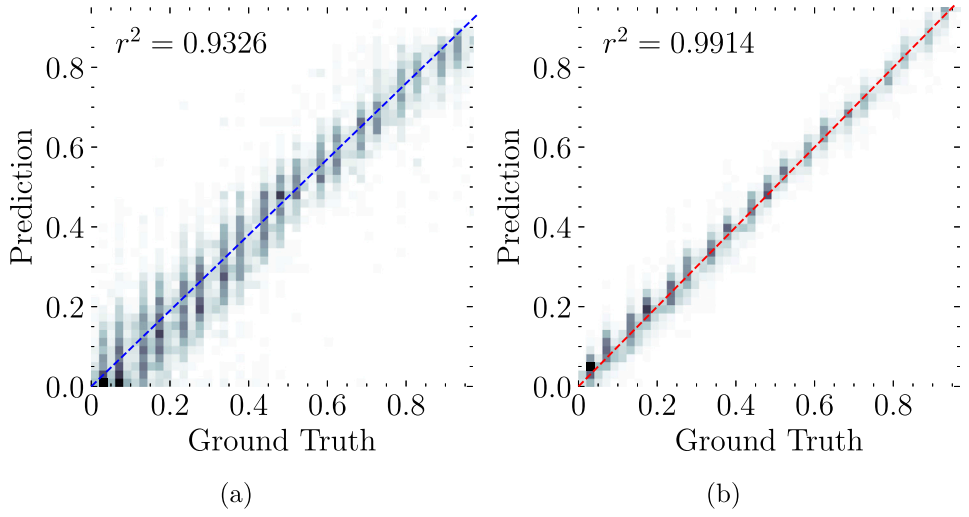



**Fig. 6.** Crossplots for the testing data set for both (a) biological fouling and (b) anchor displacements.

However, there are cases that have been wrongly assigned to different degradation classes. This is more noticeable in the case of biofouling predictions, where it can be seen that, in contrast with estimations of anchor displacements, the percentages of large errors are non-zero for almost all damage classes. More importantly, the coefficients of a 1.13% of samples with severe degradation and a 0.77% of samples with very severe damage were underestimated by at least 0.2. These cases are particularly worrying, as this implies there might be certain circumstances where significant mooring damage goes unnoticed.

To improve the model's interpretation of damage, we had to find out what data points were in this situation. In particular, we found one point for which the model estimated very mild biofouling damage (0.036), while its true coefficient represented very severe conditions (0.96). This sample is highlighted in red in Table 8, which shows test samples with similar environmental conditions.

In order to determine the reason for this, we compared its features with those of a different data point within Table 8. More specifically, we chose the point highlighted in bold, as its health status coefficients are very close to the ones defining our outlier. We plotted every input for both points to find out that, while most features showed little-to-no difference between the two cases, their high-range peak frequencies of heave and low-range frequencies of roll are significantly apart from one another. This is shown in Fig. 8c and 8d, respectively, where the values of all the input features are plotted for both points.

To identify the cause of these discrepancies, we study the time and frequency domain response of the floater in these two cases. in

**Table 8**
Test samples with similar environmental conditions to the outlier found in the classifier ($8.0 \leq H_S \leq 10.0$, $4.0 \leq T_P \leq 6.0$, $5.0 \leq V \leq 9.0$).

| $H_S$ (m) | $T_P$ (s) | $V$ (m/s) | $y_{biof,true}$ | $y_{biof,pred}$ | $y_{anch,true}$ | $y_{anch,pred}$ |
|---|---|---|---|---|---|---|
| 9.333 | 5.441 | 8.000 | 0.00 | 0.000 | 0.00 | 0.001 |
| 9.667 | 5.586 | 9.000 | 0.03 | 0.038 | 0.81 | 0.810 |
| 9.000 | 5.138 | 7.000 | 0.96 | 0.036 | 0.02 | 0.011 |
| 8.000 | 4.241 | 8.000 | 0.49 | 0.447 | 0.43 | 0.416 |
| 9.333 | 5.586 | 9.000 | 0.71 | 0.827 | 0.15 | 0.104 |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |
| 8.000 | 4.690 | 7.000 | 0.33 | 0.360 | 0.41 | 0.386 |
| 9.000 | 4.690 | 8.000 | 0.75 | 0.805 | 0.37 | 0.336 |
| **9.667** | **4.676** | **9.000** | **0.95** | **0.884** | **0.00** | **0.033** |
| 9.000 | 4.294 | 5.000 | 0.05 | 0.057 | 0.40 | 0.377 |
| 8.667 | 4.241 | 7.000 | 0.19 | 0.205 | 0.23 | 0.236 |

In red, the poorly classified sample. In bold, a data point with similar damage coefficients.

Fig. 9, we observe that both heave and roll show similar time domain behavior, with slight differences due to disparities in the external conditions of both cases. The frequency spectrum of heave presents a similar shape for the two simulations, with the poorly classified case showing higher frequency peaks and a slightly lower frequency band, while the spectrum of roll looks essentially identical for both simulations, with a somewhat lower peak for the poorly classified case. The problem is

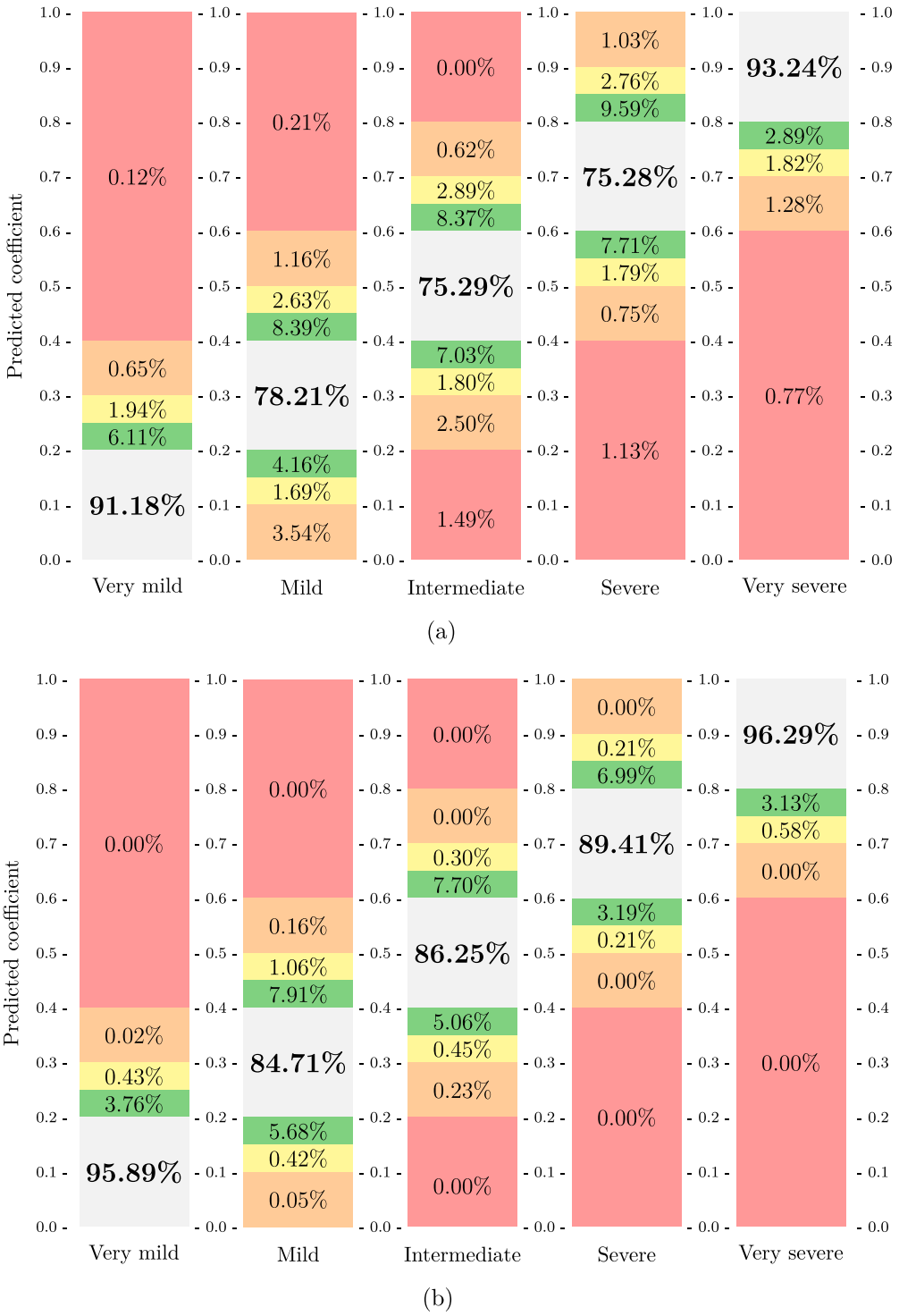



**Fig. 7.** Confusion matrices for the classification model for both (a) biological fouling, and (b) anchor point displacements.

that both frequency spectra show only one significant frequency peak, rather than the two peaks we initially contemplated. In the case of heave, it appears that the frequency bands caused by the structural condition and the external stimuli range around similar values, thus presenting only one distinct peak in the frequency spectra. The high-range roll frequency, however, seems to be uncorrelated to the severity of biofouling, as the range between the minimum and maximum peaks among the cases presented in Table 8 is almost negligible, as later proved by Figs. 8 and 9. The reason for this lies in the fact that we have considered ocean waves to be cylindrical and collinear with the $x$-direction. This, combined with the mechanical symmetry of the system consisting of both platform and mooring configuration, causes some of the degrees of freedom, such as roll or yaw, to be less excited by external conditions than others.

We therefore train a new autoencoder, bringing the total number of features down to 28, as we drop the two recently discussed variables. We ran the architecture optimizer again to find out that the model presented in Table 9 provided the best results. Since we have a

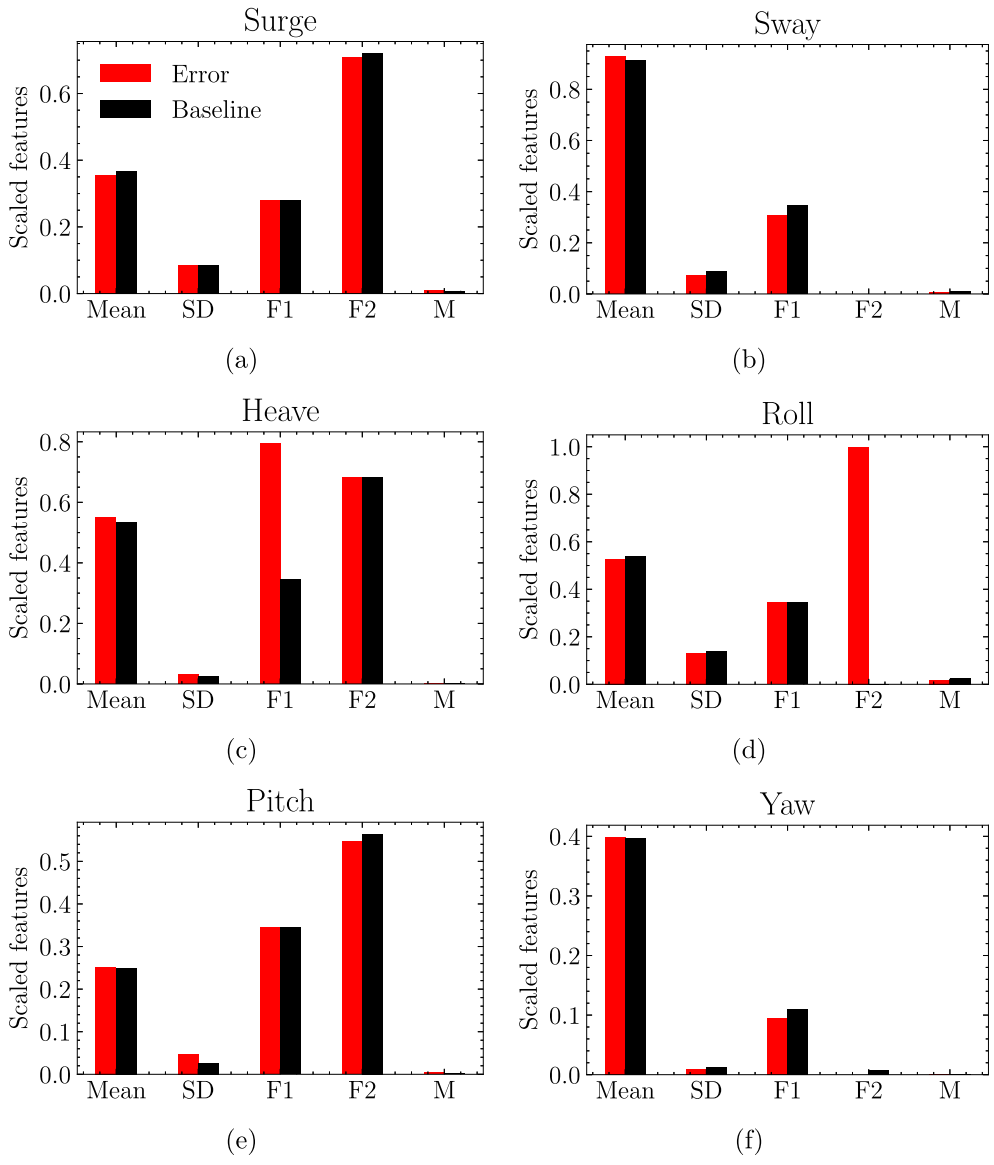



**Fig. 8.** Scaled values of all input features for the poorly classified sample (in red) and the similar, properly classified point (black) for (a) surge, (b) sway, (c) heave, (d) roll, (e) pitch, and (f) yaw.

lower number of input features, it makes sense that the most accurate architecture is simpler than that of the original, 30-feature model. In particular, we employ shallower architectures for both operators, which are now formed by only 5 hidden layers each. These layers are also thinner than those of the original network, with the widest of the encoder's layers having 25 neurons, whereas the 30-feature model's encoder had layers of 46, 48, and even 62 neurons. This is also observed in the decoder, which originally had layers up to 90-neurons wide, whereas the widest layers of the new decoder have 52 and 70 neurons. This simpler architecture also leads to lower training times, with the total training time decreasing from slightly over 26 min to 21 min and 33 s for the original and new networks, respectively.

The evolution of training and validation losses when we train the new model are presented in Fig. 10, showing equally appropriate behavior, with the total network loss this time reaching values under $10^{-3}$, which were not obtained with the original, 30-feature model. We observe crossplots for predictions of both biofouling and anchoring damage using the new model in Fig. 11, with a complete model squared correlation coefficient $r^2$ of 0.9459, almost a 2% increase with respect to the original network. It can be appreciated that, while the prediction of anchoring damage shows similar results with respect to

**Table 9**
Characteristics of the selected encoder–decoder architecture.

| Autoencoder architecture | | |
|---|---|---|
| Architecture design | Input neurons | 31 |
| | Encoder layers | 5 (25, 20, 16, 16, 12) |
| | Bottleneck neurons | 5 |
| | Decoder layers | 5 (6, 11, 30, 52, 70) |
| | Output neurons | 28 |
| | Layer connection | Fully-connected layers |
| Test results | $\mathcal{F}$ training time | 11 min 27 s |
| | $\mathcal{I}$ training time | 10 min 06 s |
| | Training time | 21 min 33 s |
| | $r^2$ | 0.9459 |
| Hyperparams. | Activation functions | ReLU, Sigmoid |
| | Optimizer | Adam |
| | Epochs (forward, total) | 1,500, 1,000 |
| | Lrn. rate (forward, total) | 0.005, 0.005 |

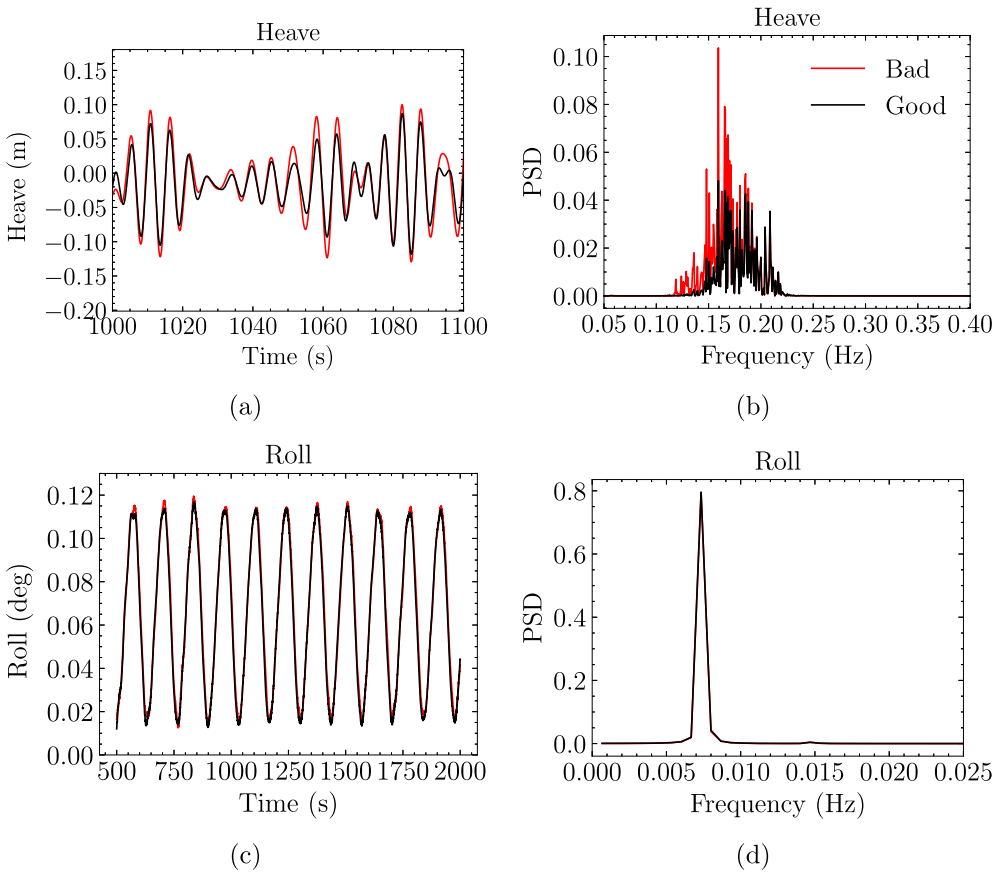



**Fig. 9.** Response of the floater for the misclassified sample (in red) and the similar case (black) for (a) heave in time domain, (b) heave in frequency domain, (c) roll in time domain, and (d) roll in frequency domain.

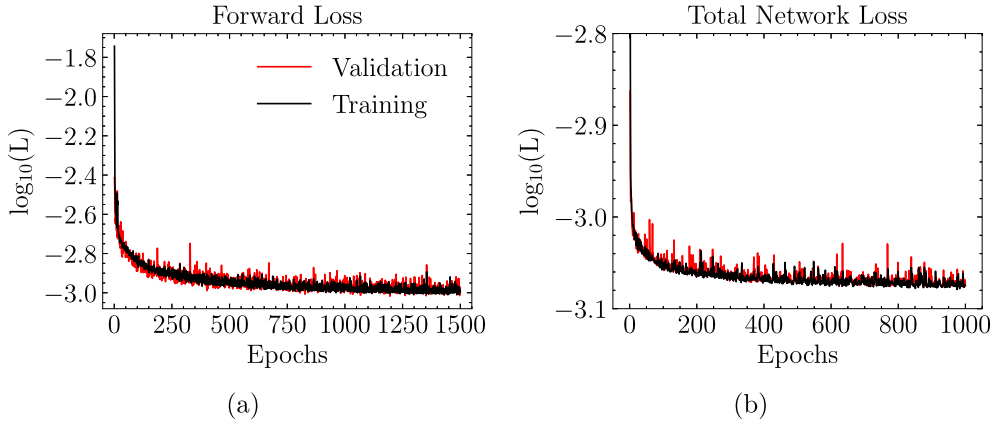



**Fig. 10.** Loss evolution for training and validation data sets as training progresses for both (a) forward operator and (b) total network using the 28-feature dataset.

the initial model, the correlation coefficient between predicted and true biofouling damage increases to 0.9851, an increase of over 5% with respect to the original network.

Fig. 12 shows the classification into five classes according to the severity of each kind of damage. We observe that this time, in contrast with the results of the 30-feature model, there are no cases with large over or underestimations of the damage coefficients for either form of failure. This implies that the removed features introduced some randomness in the network's interpretation of damage, primarily due to overlapping frequency peaks or independence between a feature and degradation coefficient. The new figures show improved results, with precision values up to 92.8% and 95.4% for very mild and very severe biofouling issues, respectively, and 94.3% and 97.8% for very mild and very severe damage caused by anchor shifts, respectively. In fact, all damage classes are very accurately represented, with precision metrics over 90% for all categories in the case of biofouling, and over 94% in the case of anchor displacements. These are much more promising results compared to those obtained with the original model, which in some intermediate cases dropped down to the mid 70s and mid 80s.

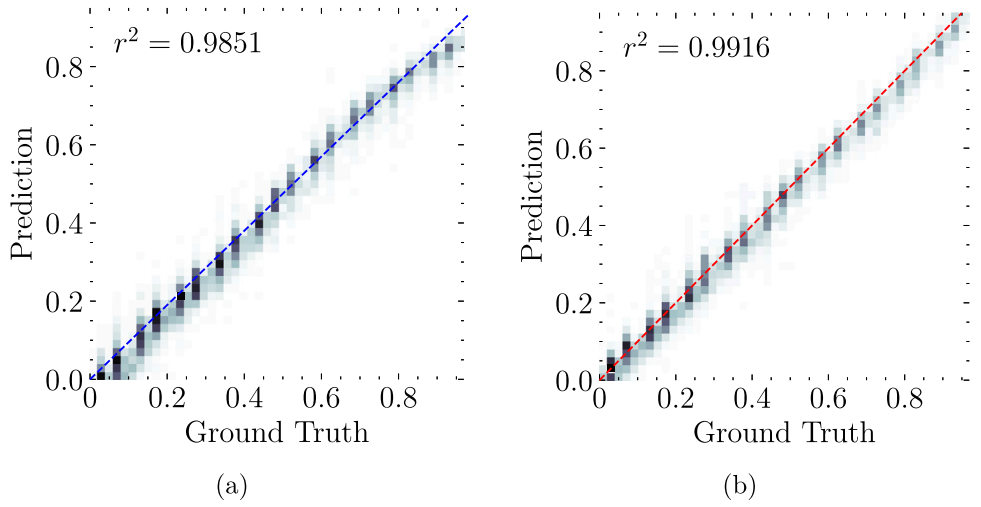



**Fig. 11.** Crossplots obtained using the 28-feature dataset for (a) biological fouling, and (b) anchor point displacements.

**Table 10**
Predictions on the testing data set after training the 28-feature model.

| Biofouling, pred. | | Biofouling, true | | Match? | Error | Anchoring, pred. | | Anchoring, true | | Match? | Error |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.191 | VM | 0.20 | M | NO | 0.009 | 0.550 | I | 0.56 | I | YES | |
| 0.286 | M | 0.28 | M | YES | | 0.622 | S | 0.67 | S | YES | |
| 0.472 | I | 0.46 | I | YES | | 0.288 | M | 0.28 | M | YES | |
| 0.019 | VM | 0.00 | VM | YES | | 0.001 | VM | 0.00 | VM | YES | |
| 0.448 | I | 0.45 | I | YES | | 0.032 | VM | 0.00 | VM | YES | |
| 0.000 | VM | 0.00 | VM | YES | | 0.000 | VM | 0.00 | VM | YES | |
| 0.208 | M | 0.20 | M | YES | | 0.502 | I | 0.50 | I | YES | |
| 0.809 | VS | 0.80 | VS | YES | | 0.313 | M | 0.30 | M | YES | |
| 0.155 | VM | 0.14 | VM | YES | | 0.624 | S | 0.70 | S | YES | |
| 0.077 | VM | 0.06 | VM | YES | | 0.003 | VM | 0.06 | VM | YES | |
| 0.000 | VM | 0.00 | VM | YES | | 0.000 | VM | 0.00 | VM | YES | |
| 0.795 | S | 0.80 | VS | NO | 0.005 | 0.677 | S | 0.70 | S | YES | |
| 0.597 | I | 0.60 | S | NO | 0.003 | 0.057 | VM | 0.05 | VM | YES | |
| 0.307 | M | 0.30 | M | YES | | 0.238 | M | 0.26 | M | YES | |
| 0.596 | I | 0.60 | S | NO | 0.004 | 0.399 | M | 0.40 | I | NO | 0.001 |
| 0.000 | VM | 0.00 | VM | YES | | 0.000 | VM | 0.00 | VM | YES | |
| 0.067 | VM | 0.00 | VM | YES | | 0.000 | VM | 0.00 | VM | YES | |
| 0.000 | VM | 0.00 | VM | YES | | 0.000 | VM | 0.00 | VM | YES | |
| 0.211 | M | 0.25 | M | YES | | 0.190 | VM | 0.20 | M | NO | 0.010 |
| 0.000 | VM | 0.00 | VM | YES | | 0.000 | VM | 0.00 | VM | YES | |
| 0.678 | S | 0.72 | S | YES | | 0.949 | VS | 0.97 | VS | YES | |
| 0.123 | VM | 0.10 | VM | YES | | 0.277 | M | 0.25 | M | YES | |
| 0.755 | S | 0.75 | S | YES | | 0.386 | M | 0.35 | M | YES | |
| 0.064 | VM | 0.00 | VM | YES | | 0.000 | VM | 0.00 | VM | YES | |
| 0.541 | I | 0.58 | I | YES | | 0.475 | I | 0.49 | I | YES | |
| 0.000 | VM | 0.00 | VM | YES | | 0.002 | VM | 0.00 | VM | YES | |
| 0.470 | I | 0.45 | I | YES | | 0.745 | S | 0.75 | S | YES | |
| 0.892 | VS | 1.00 | VS | YES | | 0.020 | VM | 0.00 | VM | YES | |
| 0.197 | VM | 0.20 | M | NO | 0.003 | 0.117 | VM | 0.10 | VM | YES | |
| 0.291 | M | 0.20 | M | YES | | 0.926 | VS | 0.95 | VS | YES | |
| 0.059 | VM | 0.00 | VM | YES | | 0.229 | M | 0.26 | M | YES | |
| 0.137 | VM | 0.05 | VM | YES | | 0.408 | I | 0.40 | I | YES | |
| 0.236 | M | 0.30 | M | YES | | 0.039 | VM | 0.00 | VM | YES | |
| 0.920 | VS | 0.85 | VS | YES | | 0.359 | M | 0.25 | M | YES | |
| 0.161 | VM | 0.15 | VM | YES | | 0.242 | M | 0.25 | M | YES | |
| 0.193 | VM | 0.15 | VM | YES | | 0.104 | VM | 0.10 | VM | YES | |
| 0.190 | VM | 0.10 | VM | YES | | 0.988 | VS | 1.00 | VS | YES | |
| 0.000 | VM | 0.00 | VM | YES | | 0.000 | VM | 0.00 | VM | YES | |

A subset of the model's predictions is displayed in Table 10, wherein we observe the algorithm's behavior when predicting both damage coefficients and classes. Mismatches between the predicted and true classes, as it can be seen in the table, always occur when the true damage coefficients are equal to the threshold separating consecutive categories. Small errors can therefore cause the model to assign a sample to the wrong class. We therefore recommend to observe both the predicted damage classes, as well as the predicted damage coefficients and the error estimate, which in Table 10 never even goes past 0.01.

## 5. Conclusions and future work

This article studies the use of predictive models for the inversion of complex engineering problems, focusing on structural health monitoring. In particular, we propose an algorithm for the early identification of damage in mooring systems for floating offshore wind turbines. This approach could help cut down costs massively by optimizing maintenance operations, e.g. minimizing unnecessary commissioning and decommissioning tasks, measurement tools, and trial-and-error




**Fig. 12.** Confusion matrices for the classification model for both (a) biological fouling, and (b) anchor point displacements.

inspection methods. In this work, we implement a deep neural network structure known as encoder–decoder, or autoencoder, to recreate a series of modal parameters describing the DeepCWind floater's response to a set of external conditions, including significant wave height, peak period and wind speed. With this approach, we intend to estimate two damage coefficients, which represent the severity of two different kinds of mooring line damage, including biological fouling and anchor displacement issues. Since experimental and real data are very scarce, we perform OpenFAST simulations to generate a set of training data. We then employ a portion of this data set to approximate the weights and biases of our algorithm, and another subset to test the model's performance when facing previously unseen samples. The training process includes two stages, the first of which covers the training of the decoder side, i.e. the forward operator. This step predicts the floating platform's response from the target damage coefficients, much like a classic partial differential equation modeling problem. We then freeze the decoder and plug it right after the encoder, which represents the inverse operator. Then, we train the entire model to optimize the parameters of the encoder, which predicts the structural integrity of the platform based on the statistical descriptors extracted from its response in 6 degrees of freedom. After designing and training an optimized encoder–decoder layout, we observe a correlation coefficient of nearly

93% between the predicted and true damages, with better accuracy for failure due to anchoring displacements. We propose a categorization approach to classify damage according to its severity, which could help users decide at which point maintenance operations should be performed. We analyze the features causing the algorithm to misclassify some samples to find out that the low range frequency of heave and high range frequency of roll introduce uncertainty in the network's prediction. We drop these features and retrain the network, obtaining improved 98.51% and 99.16% squared correlation coefficients and no relevant mismatches in the classifier.

As further research, we intend to generalize our tool by including a wider range of failure modes, such as line loss, mooring entanglement, mooring misalignment, fatigue due to cyclic loading, and corrosion. We also contemplate extending our predictor to more designs for both FOWT and floater, as well as to other components of FOWTs, such as their towers or blades, which might affect the dynamics of the FOWT and floater assembly. Another potential study consists on studying the influence of faulty sensors in the network's predictions. We also aim at implementing probabilistic deep learning algorithms, such as Gaussian mixture models or Bayesian neural networks (BNNs), to estimate uncertainty when predicting the severity of each kind of damage. This would be of great use when extending our data set to cover not only synthetic but also experimental and real data samples.

## Code availability

The code we used to train the neural networks is available in GitHub (Gorostidi et al., 2023).

## CRediT authorship contribution statement

**N. Gorostidi:** Software, Visualization, Validation, Writing – original draft, Writing – review & editing. **D. Pardo:** Methodology, Validation, Writing – review & editing, Supervision. **V. Nava:** Conceptualization, Validation, Writing – review & editing, Supervision.

## 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.

## Data availability

Data will be made available on request.

## Acknowledgments

This work has been funded by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 777778 (MATHROCKS); the Spanish Ministry of Science and Innovation projects with references TED2021-132783B-I00, PID2019-108111RB-I00 (FEDER/AEI) and PDC2021-121093-I00 (AEI/Next Generation EU); the Spanish Ministry of Economic Affairs and Digital Transformation project with reference 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, the three Elkartek projects 3KIA (KK-2020/00049), EXPERTIA (KK-2021/00048), and SIGZE (KK-2021/00095), Euskampus (project DIA), and the Consolidated Research Group MATHMODE (IT1456-22) given by the Department of Education. The authors would like to acknowledge the team at Donostia International Physics Center (DIPC) for their collaboration and support with HPC resources for the data generation stage.

## References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/, Software available from tensorflow.org.

Aizpurua, J.I., Penalba, M., Kirillova, N., Lekube, J., Marina, D., 2022. Context-informed conditional anomaly detection approach for wave power plants: The case of air turbines. Ocean Eng. 253, 111196.

Alwosheel, A., van Cranenburgh, S., Chorus, C.G., 2018. Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. J. Choice Model. 28, 167–182.

Antoniadou, I., Manson, G., Staszewski, W., Barszcz, T., Worden, K., 2015. A time–frequency analysis approach for condition monitoring of a wind turbine gearbox under varying load conditions. Mech. Syst. Signal Process. 64, 188–216.

Azimi, M., Eslamlou, A.D., Pekcan, G., 2020. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. Sensors 20 (10), 2778.

Benitz, M.A., Schmidt, D.P., Lackner, M.A., Stewart, G.M., Jonkman, J., Robertson, A., 2014. Comparison of hydrodynamic load predictions between reduced order engineering models and computational fluid dynamics for the OC4-deepCWind semi-submersible. In: International Conference on Offshore Mechanics and Arctic Engineering, Vol. 45547. American Society of Mechanical Engineers, V09BT09A006.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg.

Bjø rni, F.A., Lien, S., Midtgarden, T.A., Kulia, G., Verma, A., Jiang, Z., 2021. Prediction of dynamic mooring responses of a floating wind turbine using an artificial neural network. In: IOP Conference Series: Materials Science and Engineering. 1201, IOP Publishing, 012023.

Black, I.M., Cevasco, D., Kolios, A., 2022. Deep neural network hard parameter multi-task learning for condition monitoring of an offshore wind turbine. J. Phys. Conf. Ser. 2265 (3).

Blanco, M., Santos-Herran, M., Navarro, G., Torres, J., Najera, J., Villaba, I., Lafoz, M., Ramírez, D., 2021. Meta-heuristic optimisation approach for wave energy converter design by means of a stochastic hydrodynamic model. IET Renew. Power Gener. 15 (3), 548–561.

Chalapathy, R., Menon, A.K., Chawla, S., 2018. Anomaly detection using one-class neural networks. arXiv preprint arXiv:1802.06360.

Chen, J., Hu, Z., Duan, F., 2018. Comparisons of dynamical characteristics of a 5 MW floating wind turbine supported by a spar-buoy and a semi-submersible using model testing methods. J. Renew. Sustain. Energy 10 (5), 053311.

Chen, J., Li, J., Chen, W., Wang, Y., Jiang, T., 2020. Anomaly detection for wind turbines based on the reconstruction of condition parameters using stacked denoising autoencoders. Renew. Energy 147, 1469–1480.

Chen, H., Liu, H., Chu, X., Liu, Q., Xue, D., 2021. Anomaly detection and critical SCADA parameters identification for wind turbines based on LSTM-AE neural network. Renew. Energy 172, 829–840.

Chung, M., Kim, S., Lee, K., Shin, D.H., 2020. Detection of damaged mooring line based on deep neural networks. Ocean Eng. 209, 107522. http://dx.doi.org/10.1016/j.oceaneng.2020.107522.

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.

CNBC, 2022. Plans for floating wind energy projects off UK's coastline get millions in funding boost. URL https://www.cnbc.com/2022/01/25/plans-for-floating-wind-energy-projects-off-uks-coastline-get-funding-boost.html. (Accessed 18 July 2022).

Coulling, A.J., Goupee, A.J., Robertson, A.N., Jonkman, J.M., 2013a. Importance of second-order difference-frequency wave-diffraction forces in the validation of a fast semi-submersible floating wind turbine model. In: International Conference on Offshore Mechanics and Arctic Engineering, Vol. 55423. American Society of Mechanical Engineers, V008T09A019.

Coulling, A.J., Goupee, A.J., Robertson, A.N., Jonkman, J.M., Dagher, H.J., 2013b. Validation of a FAST semi-submersible floating wind turbine numerical model with DeepCwind test data. J. Renew. Sustain. Energy 5 (2), 023116.

Donostia International Physics Centre, 2022. Atlas FDR – technical documentation. URL http://dipc.ehu.es/cc/computing_resources/systems/atlas-fdr/. (Online; Accessed 19 October 2022).

Farrar, C.R., Worden, K., 2012. Structural Health Monitoring: A Machine Learning Perspective. John Wiley & Sons.

Feng, Z., Liang, M., 2014. Fault diagnosis of wind turbine planetary gearbox under nonstationary conditions via adaptive optimal kernel time–frequency analysis. Renew. Energy 66, 468–477.

Feng, Z., Qin, S., Liang, M., 2016. Time–frequency analysis based on Vold-Kalman filter and higher order energy separation for fault diagnosis of wind turbine planetary gearbox under nonstationary conditions. Renew. Energy 85, 45–56.

Forbes, 2021. Why are oil majors investing in offshore wind? URL https://www.forbes.com/sites/woodmackenzie/2021/04/06/why-are-oil-majors-investing-in-offshore-wind/?sh=74c168663ffa. (Accessed 20 July 2022).

Fu, J., Chu, J., Guo, P., Chen, Z., 2019. Condition monitoring of wind turbine gearbox bearing based on deep learning model. Ieee Access 7, 57078–57087.

Galera-Calero, L., Blanco, J.M., Iglesias, G., 2021. Numerical modelling of a floating wind turbine semi-submersible platform. Appl. Sci. 11 (23), 11270.

García, F.P., Pedregal, D.J., Roberts, C., 2010. Time series methods applied to failure prediction and detection. Reliab. Eng. Syst. Saf. 95 (6), 698–703.

Ghoshal, A., Sundaresan, M.J., Schulz, M.J., Pai, P.F., 2000. Structural health monitoring techniques for wind turbine blades. J. Wind Eng. Ind. Aerodyn. 85 (3), 309–324.

Gordan, M., Razak, H.A., Ismail, Z., Ghaedi, K., 2017. Recent developments in damage identification of structures using data mining. Latin Am. J. Solids Struct. 14, 2373–2401.

Gorostidi, N., Nava, V., 2021. A deep learning model for the structural health monitoring of floating offshore wind turbine mooring lines based on modal parameters. In: Book of Proceedings of the 17th EAWE PhD Seminar, Vol. 1. pp. 149–156.

Gorostidi, N., Nava, V., Aristondo, A., Pardo, D., 2022. Predictive maintenance of floating offshore wind turbine mooring lines using deep neural networks. J. Phys.: Conf. Ser. 2257, 012008.

Gorostidi, N., Pardo, D., Nava, V., 2023. Deep learning for the structural health monitoring of mooring lines. https://github.com/ngoros/DeepLearningMooringFOWT. (Accessed 12 June 2023).

Green Tech Media, 2020a. Equinor: Floating wind farms a natural fit for oil and gas companies. URL https://www.greentechmedia.com/articles/read/floating-wind-is-cutting-costs-faster-than-regular-offshore-wind. (Accessed 20 July 2022).

Green Tech Media, 2020b. Total and Macquarie invest in 2.3GW portfolio of floating wind projects in South Korea. URL https://www.greentechmedia.com/articles/read/total-and-macquarie-partner-on-worlds-first-full-scale-floating-wind-projects. (Accessed 20 July 2022).

GWEC, 2021. Global Wind Energy Report 2021. Tech. Rep., Global Wind Energy Council, Brussels, Belgium.

Hameed, Z., Hong, Y., Cho, Y., Ahn, S., Song, C., 2009. Condition monitoring and fault detection of wind turbines and related algorithms: A review. Renew. Sustain. Energy Rev. 13 (1), 1–39.

Han, J., Moraga, C., 1995. The influence of the sigmoid function parameters on the speed of backpropagation learning. In: International Workshop on Artificial Neural Networks. Springer, pp. 195–201.

IRENA, 2021. Offshore renewables: an action agenda for deployment. International Renewable Energy Agency, Abu Dhabi.

Janas, K., Milne, I., Whelan, J., 2021. Application of a convolutional neural network for mooring failure identification. Ocean Eng. 232, 109119.

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

Joshuva, A.D., Aslesh, A., Sugumaran, V., 2019. State of the art of structural health monitoring of wind turbines. Int. J. Mech. Sci. 9, 95–112. http://dx.doi.org/10.24247/ijmperdoct201910.

Kim, H.-C., Kim, M.-H., Choe, D.-E., 2019. Structural health monitoring of towers and blades for floating offshore wind turbines using operational modal analysis and modal properties with numerical-sensor signals. Ocean Eng. 188, 106226.

Lee, S., Churchfield, M., Driscoll, F., Sirnivas, S., Jonkman, J., Moriarty, P., Skaare, B., Nielsen, F.G., Byklum, E., 2018. Load estimation of offshore wind turbines. Energies 11 (7), 1895.

Li, C.B., Choung, J., 2017. Fatigue damage analysis for a floating offshore wind turbine mooring line using the artificial neural network approach. Ships Offshore Struct. 12 (sup1), S288–S295.

Li, C.B., Choung, J., Noh, M.-H., 2018. Wide-banded fatigue damage evaluation of catenary mooring lines using various artificial neural networks models. Mar. Struct. 60, 186–200.

Li, Y., Le, C., Ding, H., Zhang, P., Zhang, J., 2019. Dynamic response for a submerged floating offshore wind turbine with different mooring configurations. J. Mar. Sci. Eng. 7, 115. http://dx.doi.org/10.3390/jmse7040115.

Lin, Y.-H., Yang, C.-H., 2020. Hydrodynamic simulation of the semi-submersible wind float by investigating mooring systems in irregular waves. Appl. Sci. 10 (12), 4267.

Liu, Z., Fan, Y., Wang, W., Qian, G., 2019. Numerical study of a proposed semi-submersible floating platform with different numbers of offset columns based on the DeepCwind prototype for improving the wave-resistance ability. Appl. Sci. 9 (6), 1255.

Liu, Y., Yoshida, S., Yamamoto, H., Toyofuku, A., He, G., Yang, S., 2018. Response characteristics of the deepCWind floating wind turbine moored by a single-point mooring system. Appl. Sci. 8 (11), 2306.

Low, Y., Langley, R., 2006. Time and frequency domain coupled analysis of deepwater floating production systems. Appl. Ocean Res. 28 (6), 371–385.

Martinez-Luengo, M., Kolios, A., Wang, L., 2016. Structural health monitoring of offshore wind turbines: A review through the statistical pattern recognition paradigm. Renew. Sustain. Energy Rev. 64, 91–105.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. In: Icml. pp. 807–814.

National Renewable Energy Laboratory, 2021. OpenFAST documentation. URL https://openfast.readthedocs.io/en/main/. (Online; Accessed 10 February 2022).

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. Institution of Engineering and Technology, pp. 397–424. http://dx.doi.org/10.1049/PBPO129E_ch11.

Offshore Engineer Digital, 2021. Shell joins floating wind project offshore Ireland. URL https://www.oedigital.com/news/484925-shell-joins-floating-wind-project-offshore-ireland. (Accessed 20 July 2022).

OffshoreWind.biz, 2021. Iberdrola to invest over EUR 1 billion in floating wind farm offshore Spain. URL https://www.offshorewind.biz/2021/02/16/iberdrola-to-invest-over-eur-1-billion-in-floating-wind-farm-offshore-spain/. (Online; Accessed 26 July 2022).

OffshoreWind.biz, 2022. Saitec unveils plans for 50 MW floating wind farm offshore Spain. URL https://www.offshorewind.biz/2022/06/02/saitec-unveils-plans-for-50-mw-floating-wind-farm-offshore-spain/. (Accessed 18 July 2022).

Oliveira, G., Magalhães, F., Cunha, A., Caetano, E., 2021. Modal decomposition of the dynamic response of wind turbine during one year of continuous monitoring. Struct. Control Health Monit. 28 (8), 1–20.

Peña-Sanchez, Y., Faedo, N., Ringwood, J.V., 2022. Frequency-domain identification of radiation forces for floating wind turbines by moment-matching. Float. Offshore Energy Dev. 1, 66–73.

Penalba, M., Aizpurua, J.I., Martinez-Perurena, A., Iglesias, G., 2022. A data-driven long-term metocean data forecasting approach for the design of marine renewable energy systems. Renew. Sustain. Energy Rev. 167, 112751.

Penalba, M., Touzón, I., Lopez-Mendia, J., Nava, V., 2017. A numerical study on the hydrodynamic impact of device slenderness and array size in wave energy farms in realistic wave climates. Ocean Eng. 142, 224–232.

Quarz, 2020. Floating wind farms just became a serious business. URL https://qz.com/1650433/hywind-scotland-makes-floating-wind-farms-a-serious-business/. (Accessed 20 July 2022).

REN21, 2021. Renewables 2021 Global Status Report. Tech. Rep., REN21, Paris, France.

Reuters, 2019. Equinor to invest nearly $550 million in floating wind power off Norway. URL https://www.reuters.com/article/us-equinor-windfarm/equinor-to-invest-nearly-550-million-in-floating-wind-power-off-norway-idUSKBN1WQ0DZ. (Accessed 20 July 2022).

Robertson, A., Jonkman, J., Masciola, M., Song, H., 2014. Definition of the Semisubmersible Floating System for Phase II of OC4. Tech. Rep. 1, National Renewable Energy Laboratory (NREL), Denver, CO.

Ruzzo, C., Failla, G., Collu, M., Nava, V., Fiamma, V., Arena, F., 2016. Operational modal analysis of a spar-type floating platform using frequency domain decomposition method. Energies 9 (11), 870.

Saad, A.M., Schopp, F., Barreira, R.A., Santos, I.H.F., Tannuri, E.A., Gomi, E.S., Costa, A.H.R., 2021. Using neural network approaches to detect mooring line failure. IEEE Access 9, 27678–27695.

Salehi, H., Burgueño, R., 2018. Emerging artificial intelligence methods in structural engineering. Eng. Struct. 171, 170–189.

Shahriari, M., Pardo, D., Kargaran, S., Teijeiro, T., 2022. Automated machine learning for borehole resistivity measurements. http://dx.doi.org/10.48550/ARXIV.2207.09849, URL https://arxiv.org/abs/2207.09849.

Shahriari, M., Pardo, D., Rivera, J.A., Torres-Verdín, C., Picon, A., Del Ser, J., Ossandón, S., Calo, V.M., 2021. Error control and loss functions for the deep learning inversion of borehole resistivity measurements. Internat. J. Numer. Methods Engrg. 122 (6), 1629–1657.

Sidarta, D.E., O'Sullivan, J., Lim, H.-J., 2018. Damage detection of offshore platform mooring line using artificial neural network. In: International Conference on Offshore Mechanics and Arctic Engineering, Vol. 51203. American Society of Mechanical Engineers, V001T01A058.

Sierra-Garcia, J.E., Santos, M., Victores, J.G., 2019. Neural controller of UAVs with inertia variations. In: International Conference on Intelligent Data Engineering and Automated Learning. Springer, pp. 169–177.

Stewart, G.M., Lackner, M.A., Robertson, A., Jonkman, J., Goupee, A.J., 2012. Calibration and validation of a FAST floating wind turbine model of the DeepCwind scaled tension-leg platform. In: The Twenty-Second International Offshore and Polar Engineering Conference. OnePetro, pp. 1–10.

Tarantola, A., 2005. Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM.

The Conversation, 2020. Floating wind farms: how to make them the future of green electricity. URL https://theconversation.com/floating-wind-farms-how-to-make-them-the-future-of-green-electricity-142847. (Accessed 18 July 2022).

Wang, Z., Qiao, D., Yan, J., Tang, G., Li, B., Ning, D., 2022. A new approach to predict dynamic mooring tension using LSTM neural network based on responses of floating structure. Ocean Eng. 249, 110905.

Wang, L., Robertson, A., Jonkman, J., Yu, Y.-H., Koop, A., Nadal, A.B., Li, H., Bachynski-Polić, E., Pinguet, R., Shi, W., et al., 2021. OC6 phase ib: Validation of the CFD predictions of difference-frequency wave excitation on a FOWT semisubmersible. Ocean Eng. 241, 110026.

Wind Power Monthly, 2020. MHI Vestas installs 'most powerful' floating offshore wind turbine. URL https://www.windpowermonthly.com/article/1699782/mhi-vestas-installs-most-powerful-floating-offshore-wind-turbine. (Accessed 20 July 2022).

WindEurope, 2021. Spain issues plan for up to 3 GW offshore wind by 2030 – in perfect time for WindEurope 2022 in Bilbao! URL https://windeurope.org/newsroom/news/spain-issues-plan-for-up-to-3-gw-offshore-wind-by-2030-in-perfect-time-for-windeurope-2022-in-bilbao/. (Accessed 18 July 2022).

WindEurope, 2022. Europe can expect to have 10 GW of floating wind by 2030. URL https://windeurope.org/newsroom/news/europe-can-expect-to-have-10-gw-of-floating-wind-by-2030/. (Accessed 18 July 2022).

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, 109094.

Yang, L., Shami, A., 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing 415, 295–316.

Zhang, Y., Kim, B., 2018. A fully coupled computational fluid dynamics method for analysis of semi-submersible floating offshore wind turbines under wind-wave excitation conditions based on OC5 data. Appl. Sci. 8 (11), 2314.