Characterizing Damage in Wind Turbine Mooring Using a Data-Driven Predictor Model within a Particle Filtering Estimation Framework

Rohit Kumar[1], Ananay Thakur[2], Shereena OA[3], Arvind Keprate[4] and Subhamoy Sen[5]

[1,2,3,5] *i4S Laboratory, Indian Institute of Technology Mandi, Mandi,175075, HP, India*
rohit373k@gmail.com
thakur07ananay@gmail.com
oa.shereena21@gmail.com
subhamoy@iitmandi.ac.in

[4] *Department of Mechanical, Electrical, and Chemical Engineering, Oslo Metropolitan University, 0166 Oslo, Norway*
arvindke@oslomet.no

ABSTRACT

Floating Offshore Wind Turbines (FOWT) represent a promising solution to renewable energy challenges, yet effective maintenance remains critical for cost management. Traditional machine learning (ML) approaches for detecting FOWT damage often rely on extensive real-world data, which can be impractical and economically unfeasible. Alternatively, stochastic filtering-based time-domain approaches leverage physical understanding through dynamic models, typically finite element models. However, these methods are hindered by excessive simulation calls within the recursive filtering frameworks. This study proposes a novel filtering-based approach that replaces the computationally intensive process model with a Deep Neural Network (DNN) surrogate, addressing the aforementioned limitations. The proposed approach utilizes synthetic data generated from the high-fidelity calibrated OpenFAST model of FOWT dynamics to train a DNN toward learning the dynamic evolution of the FOWT conditioned on the current health state. By offering a computationally efficient representation of system dynamics conditioned on health state, this approach allows for real-time damage detection and interpretable information on damage severity within a stochastic inverse estimation framework, specifically employing Particle Filtering in this study. This approach contrasts with traditional black-box ML-based methods, which typically struggle to provide interpretable information on damage characteristics. Extensive numerical investigations on damaged FOWT mooring lines demonstrate this approach's practical applicability and superiority over traditional ML-based methods. Eventually, integrating explainable ML models within the filtering framework induces promptness in detection without sacrificing transparency.

1. INTRODUCTION

Research on Floating Offshore Wind Turbines (FOWTs) has advanced significantly, reflecting their growing adoption across industries. A key challenge is maintaining safety while minimizing maintenance costs, a critical issue that persists. Structural health monitoring (SHM), particularly data-driven methods, is valued for its noise robustness and cost-effectiveness, as demonstrated by (Azimi, Eslamlou, & Pekcan, 2020).

The complex inverse problems in SHM mandate linking measurements to causes or damages. Machine Learning (ML) approaches have showcased excellent reliability and predictability across applications, yet dependence on data alone raises concerns, especially in mooring line damage detection (Avci, Abdeljaber, & Kiranyaz, 2022). In ML-based system identification for FOWTs, strategies involve extracting damage-sensitive features (DSFs) using supervised or unsupervised learning. While unsupervised methods, such as novelty detection, are effective in damage detection, they face challenges in localization and quantification (Wang, Tian, Peng, & Luo, 2018). Supervised techniques, on the other hand, require large datasets and can function as classifiers or regressors, employing algorithms such as random forest, support vector machine (SVM), and multi-layer perceptron (MLP) (Regan, Beale, & Inalpolat, 2017).

In FOWTs, the selection of appropriate DSFs for mooring line damage detection holds significant importance. Super-

vised algorithms depend on the precise representation of damage scenarios through these selected DSFs. However, it's important to acknowledge that the resulting model is essentially a black box, lacking interpretability to extract additional information not included during training. Consequently, ML-based approaches relying solely on data may not comprehensively address mooring line damage detection, underscoring the necessity for a nuanced monitoring strategy (Malekloo, Ozer, AlHamaydeh, & Girolami, 2022).

1.1. Condition Monitoring of Mooring Lines

Mooring lines are vital components in ensuring the integrity of FOWTs, influencing the optimization of support structures (Altuzarra et al., 2022). Given their significance, monitoring the health of mooring lines is essential due to potential stability implications (Aqdam, Ettefagh, & Hassannejad, 2018). While deep learning (DL) algorithms show promise in detecting damages to wind systems (Choe, Kim, & Kim, 2021), it remains imperative to understand the behavior of coupled systems under extreme conditions (Li, Le, Ding, Zhang, & Zhang, 2019), necessitating the incorporation of physics-based or physics-guided support models. Despite the prevalence of model-based and fuzzy logic approaches for mooring damage diagnosis (Jamalkia, Ettefagh, & Mojtahedi, 2016) in current literature, research on ML and DL in this domain is limited. Vibration measurements facilitate efficient identification of structural damage, aiding in damage diagnosis across various domains, including FOWTs (Farrar, Doebling, & Nix, 2001). Recent studies (Gorostidi, Pardo, & Nava, 2023) highlight the advantages of ML over model-based methods in managing large data and promptness in detection.

Traditional ML models pose significant limitations for real-life SHM due to their lack of interpretability. These black-box models, while effective at processing large amounts of data, often fail to provide meaningful insights into the underlying dynamics of the monitored system. Alternatively, stochastic filtering-based approaches, although capable of incorporating physical understanding through complex models, suffer from the computational burden of these models, making them impractical for real-time applications. However, a compromise between interpretability and efficiency can be achieved by leveraging ML techniques to create a surrogate of the conceptual model. This approach, as proposed in this study, involves replacing the computationally intensive process model with a Deep Neural Networks (DNN) surrogate. By training the DNN with real (/synthetic) data sampled (/generated) from reality (/a high-fidelity dynamic model of the system), the proposed method offers both computational efficiency and interpretability. This surrogate model can then be seamlessly integrated into a stochastic filtering framework, providing real-time damage detection with required promptness while maintaining transparency and accuracy. Thus, the study bridges the gap between conventional ML-based approaches and complex stochastic filtering methods, offering a promising solution for effective SHM in practical applications.

2. METHODOLOGY

A process model plays a major role in filtering-based methods and is often built with a Finite Element (FE) modeling approach. These models, derived from physical systems, simulate and predict the system behavior under diverse operational conditions. Despite their widespread use in evaluating civil structure conditions and detecting damage, FE models face certain challenges, such as numerical convergence issues, memory demands, and complexities in parallelization. Surrogate models such as DNNs are known for their ability to identify complex patterns swiftly and accurately, particularly in one-step-ahead time series forecasting within a data-based time-series modeling framework, and hence are one of the best choices for replacing FE models.

DNN models are faster, more adaptable, and require comparatively less simulation effort than traditional FE models. DNNs also offer other computational advantages like scalability, capturing nonlinear data relationships efficiently, thereby enabling quick and accurate predictions without the need for iterative solutions. This streamlined approach accelerates development and reduces costs associated with model building and simulation.

Typical health assessment problems, addressed with conventional DNN models, require extensive datasets correlating response inputs with damage locations and severity labels to achieve comprehensive accuracy. To mitigate this issue, instead of correlating the response to its corresponding damage labels, the underlying dynamics are learned within a DNN framework utilizing a response time series of consecutive time steps as input-output pairs. The input is additionally augmented with the health states of the system to render the prediction conditional on the health state. Once trained, the unobserved health state can therefore be observed with the DNN network in terms of response. Filter-based estimation methods further leverage this mapping to inversely estimate the health state inferred from the measured response time history. However, before that, exploring the sensitivity of the prepared DNN models compared to traditional FEM-based models is essential to justifying their adoption over costlier FEM-based predictors.

The subsequent discussion focuses on a DNN model trained using simulations from an FEM-based model, synthesized from software like OpenFAST designed for FOWTs. Actual sampled responses will replace simulated ones for real-world implementation.

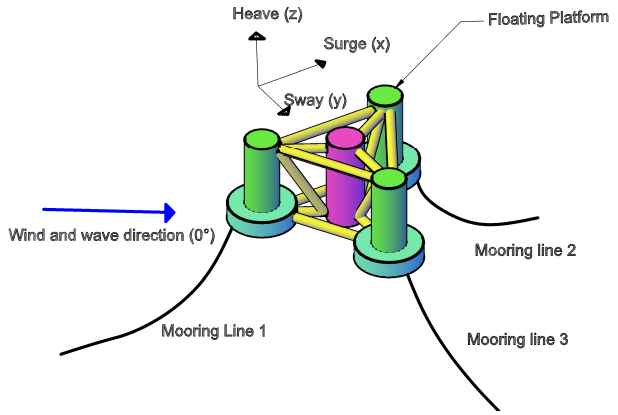

Figure 1. OC4 Semisubmersible floater along with catenary mooring system for NREL's 5MW wind turbine.

2.1. OpenFAST Model

Ensuring the DNN model undergoes thorough training necessitates a substantial volume of data. However, due to the limited prevalence and operational scope of FOWTs, obtaining a satisfactory quantity of authentic data is challenging. Consequently, we address this requirement by generating the requisite data through high-fidelity software capable of multi-physics simulation. In this investigation, we employ an NREL 5 MW Wind turbine model affixed to an OC4 semi-submersible model, as depicted in Figure 1. This configuration incorporates a catenary mooring system with three mooring lines fastened at 120° angles (Robertson et al., 2014); detailed mooring line specifications are provided in Table 1. The specifications for the 5 MW reference turbine are outlined in (Jonkman, Butterfield, Musial, & Scott, 2009). Additionally, data across varying sea states are simulated, characterized by wide-band operational scenarios involving a significant wave height (H_s) of 6 m and a Peak period (T_p) of 10 sec, assuming the turbine operates under a constant, steady wind speed of 8 m/s while in full operational mode.

Table 1. Mooring lines details

Diameter (m)	Mass density (kg m^{-1})	Axial stiffness (N)	Unstretched length (m)
0.0766	113.35	7.5903×10^8	835.35

In OpenFAST, various modules perform distinct functions. For instance, the HydroDyn module adopts a hybrid approach to handle hydrodynamic loads on the platform, merging diffraction theory with the Morison equation. The AeroDyn module uses blade element momentum (BEM) theory to manage aerodynamic loads. MoorDyn oversees loads related to mooring lines through the lumped mass method. ElastoDyn addresses structural and gravitational loads, while the InflowWind module supplies essential wind output. These modules are interconnected, collaborating to simulate the desired responses.

Table 2. Mooring Line's Damage (D) and Healthy (H) scenarios

Cases	Mooring Line's Damage Scenarios		
	Line 1 (k_1)	Line 2 (k_2)	Line 3 (k_3)
Case 1	H	H	H
Case 2	D	H	H
Case 3	H	D	H
Case 4	H	H	D
Case 5	D	D	H
Case 6	H	D	D
Case 7	D	H	D
Case 8	D	D	D

The DNN network underwent training to support the particle filter, capturing six distinct responses: the displacement and velocities of the floating platform in three directional axes. Response data was simulated over a duration of 3600 seconds, sampled at a frequency of 40 Hz. Variations in response resulting from alterations in the material properties of the mooring line were documented, leading to the simulation of different scenarios corresponding to various combinations of mooring line damage. Specifically, damages to the mooring lines were introduced as reductions in axial stiffness. For each scenario, 240 samples were simulated, resulting in a total of 1920 samples across 8 distinct cases as outlined in Table 2. Each case encompasses three discrete levels of damage for each mooring line, representing reductions of 10%, 15%, and 20% in axial stiffness (k_1, k_2, and k_3) of the mooring material.

2.2. Deep Neural Network (DNN)

The DNN model was subsequently developed to learn from data generated by OpenFAST. To predict the displacements and the velocities of a floating platform at the next ($k + 1^{th}$) sampling time step using the current time step (k^{th}) responses along with the health states as the input, the DNN utilizes a technique called back-propagation, wherein the prediction errors are propagated backward through the network, allowing the model to adjust its internal parameters, to improve future predictions. The architecture for the designed DNN model is provided in Table 3.

Table 3. Characteristics of the DNN model

DNN Architecture		
Layers	**Activation Function**	**Nodes**
Input Layer	-	9
Hidden Layer - 1	ReLU	128
Hidden Layer - 2	ReLU	64
Output Layer	Linear	6
Hyperparameters		
Optimizer		Adam
Epochs		1000
Learning Rate		10^{-6}

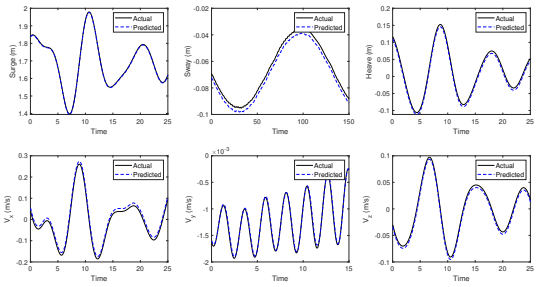

Figure 2. Actual and one-time step ahead predicted response comparison

The DNN architecture comprises an input, an output, and two hidden layers. The model is trained using 70% of the dataset allocated as training data. The remaining, 20% of data is used as a validation set to optimize hyper-parameters (activation function, learning rate, batch size) using RMSE and MAE metrics. Subsequently, testing is conducted using the remaining 10% of data. To ensure the required architecture for datasets, ReLU (Rectified Linear Unit) activation and linear activation functions are used in the hidden layers and output layer, respectively. The Adam optimizer, with a learning rate of 10^{-6} for 1000 epochs, was used for model optimization.

The model is tested on 10% of datasets. The model performed well in the provided regression task with good Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) values, as shown in Table 4. Further, a comparison of one-time step-ahead predictions and actual simulated responses is shown in Figure 2.

Table 4. Trained DNN model's accuracy indices for testing data

	RMSE	**MAE**	**MBE**
Surge (m)	0.0014151	7.7249×10^{-5}	-9.0×10^{-5}
Sway (m)	0.0001885	9.1052×10^{-6}	9.0×10^{-5}
Heave (m)	0.0007831	2.5640×10^{-5}	-1.77×10^{-4}
V_x **(m/s)**	0.0022982	0.0015	-3.245×10^{-3}
V_y **(m/s)**	3.4311×10^{-5}	2.8587×10^{-5}	-1.8×10^{-5}
V_z **(m/s**	0.0006027	0.0003	9.8×10^{-5}

The performance of the DNN model, in predicting various responses, has been evaluated based on the provided accuracy indices. The RMSE values ranging from 3.4311×10^{-5} to 0.0022982 indicate relatively low prediction errors across different parameters, suggesting the model's capability to make accurate one-time step-ahead predictions. Likewise, the MAE values, ranging from 9.1052×10^{-6} to 0.0015, demonstrate the model's ability to predict parameter values with small deviations from the true values. Despite some biases observed in the Mean Bias Error (MBE) values, their magnitudes are relatively small, indicating an unbiased prediction by the ANN model. Overall, these accuracy indices suggest that the ANN model performs well in predicting the parameters of interest, making it useful for particle filters.

2.3. DNN-Particle filter

Particle filter, which typically approaches the Sequential Monte Carlo (SMC) method, is a powerful technique used for state estimation in nonlinear and non-Gaussian systems. The computational difficulties associated with scenarios where the state-space model is complex and the direct analytical solutions are intractable are handled effectively by PF-based algorithms.

The central idea behind a particle filter is that the posterior distribution ($p(x_{k-1}|R_{k-1})$) of the system state could be represented using a set of weighted samples, called particles $\{x_k^{(i)} : i = 1, 2, 3, \ldots, N\}$. These particles evolve according to the system dynamics and are updated based on their likelihood against measurements arriving sequentially in time. The filter approximates the posterior distribution by propagating particles through the system dynamics and adjusting their weights based on the likelihood of observed measurements.

The key steps in a particle filter algorithm include prediction, measurement update, and resampling. In the prediction step, particles are propagated forward in time according to the system dynamics, incorporating process noise, if present. In the measurement update step, particles are weighted based on their consistency with the observed measurements, calculated using the likelihood function. Upon receiving measurements, the probability of each sample from the previous time step is evaluated, and the normalized weight of each sample is determined using Eq. (1).

$$a_i = \frac{p(\Phi_k|\widetilde{x}_k^{(i)})}{\sum_{j=1}^{N} p(\Phi_k|\widetilde{x}_k^{(j)})} \tag{1}$$

Each sample $\widetilde{x}_k^{(i)} : i = 1, 2, 3, \ldots, N$ forms a discrete distribution with probability mass a_i associated with element i. Resampling is then performed to prevent particle degeneracy by replicating particles with higher weights and eliminating those with lower weights, redistributing the particle set to high-likelihood areas. The above process helps ensure a diverse representation of the posterior distribution. Resampling the discrete distribution N times creates new samples, weighted based on their likelihood against the observed data. Particle filters can handle nonlinear, non-Gaussian systems without linearization but may suffer in high-dimensional spaces due to the curse of dimensionality, requiring careful parameter tuning like the number of particles to balance efficiency and accuracy.

In essence, particle filters offer a flexible and effective framework for state estimation in nonlinear, non-Gaussian systems, serving as valuable tools in robotics, target tracking, and fi-

nancial modeling.

2.3.1. Parameter Estimation via PF

Within the PF framework, the adopted damage attributes, posed as parameters θ_k, are defined with a set of N_p independent parameter particles $\boldsymbol{\xi} = [\xi_{k-1}^1, \xi_{k-1}^2, \cdots, \xi_{k-1}^{N_p}]$ (?, ?). These particles, each representing a possible state of the system, are used to propagate system uncertainty over time through a process model. Subsequently, the propagated particles are evaluated against available measurements using the measurement model to compute their likelihood. This likelihood, when combined with the prior likelihood of particles, forms the posterior distribution. Finally, new particle samples are drawn from this posterior distribution to be utilized in the next iteration of the process.

The process model for this particle filter demonstrates the evolution of the parameter vector $\theta_k : \mathbb{R}^p$ to be estimated using a random walk model as follows:

$$\theta_{k+1} = \theta_k + u_k \tag{2}$$

θ_k signifies the health states, typically material properties, stiffness, or health indices, through which damage can be characterized. This model allows the parameter states to evolve in time to converge to their respective true values. Within the particle filtering framework, this uncertainty propagation is achieved by the time updating of several particles through the process model, along with the associated uncertainty u_k that has been modeled as a stationary white Gaussian noise (SWGN) of covariance Q_k.

Further, with each time step k, the evolution of the particle, ξ_{k-1}^j, is essentially represented by a random perturbation around its current position. A Gaussian blur ($\mathbb{N}(\delta\xi_k, \sigma_k^\xi)$) is additionally applied to ξ_{k-1}^j with a shift $\delta\xi_k = (1-\alpha)\bar{\xi}_{k-1}$ and a spread of σ_k^ξ [1]. The turbulence in the particle estimation is effectively managed through the implementation of hyperparameter α, which attempts to re-center the particles towards their mean ($\bar{\xi}_{k-1}$) as,

$$\xi_k^j = \alpha\xi_{k-1}^j + \mathcal{N}(\delta\xi_k, \sigma_k^\xi) \tag{3}$$

After propagation, each parameter particle undergoes observation against available measurements utilizing the DNN model. This model can map current responses and parameters to responses at the subsequent time step. The further mapping of the response at the next step to its corresponding available response is not explicitly elaborated here and is collectively incorporated within the measurement function $h(\bullet)$. The measurement model is defined as follows:

$$y_{k+1} = h_k(x_k, \theta_{k+1}, v_k) \tag{4}$$

In this context, h_k therefore utilizes the DNN surrogate of the FEM model trained with simulated synthetic response data. The DNN surrogate predicts the responses at the next time steps, some of which are observed as measurements y_{k+1}. v_k denotes measurement noise, modeled as another SWGN with covariance R_k. In the current FOWT monitoring scenario, the parameter vector encompasses the stiffness parameters of three mooring catenaries (k_1, k_2, and k_3).

Using this process and measurement model, the particle filter estimates the posterior of parameter particles, and the particle mean leads to the estimate of the particle filters. Due to space constraints, a detailed description of the particle filter is not provided here. Interested readers are encouraged to refer to the extensive literature available in this field.

2.3.2. Particle update and particle approximation

Next, the likelihood, $\mathcal{L}(\xi_k^j)$ for each j^{th} particle is computed using the corresponding innovation mean and covariance. These likelihoods are further convoluted with the prior weights $w(\xi_{k-1}^j)$ to estimate the corresponding posterior $w(\xi_k^j)$.

$$w(\xi_k^j) = \frac{w(\xi_{k-1}^j)\mathcal{L}(\xi_k^j)}{\sum_{j=1}^{N_p} w(\xi_{k-1}^j)\mathcal{L}(\xi_k^j)} \quad \text{with}$$
$$\mathcal{L}(\xi_k^j) = \left((2\pi)^n \sqrt{|\mathbf{R}_k|} \right)^{-1} e^{-0.5 i_k^{j\,T} \mathbf{S}_k^{-1} i_k^j} \tag{5}$$

With these updated weights, the particle approximations for the parameters are then estimated as:

$$\xi_{k|k} = \sum_{j=1}^{N_p} w(\xi_k^j)\xi_k^j \tag{6}$$

3. Results and Discussion

The integration of a DNN-based process model within a particle filtering environment underwent testing for two numerical case studies under two different operational conditions (refer Case 2 and Case 8). Under operating condition case 2, the numerical experiment additionally considers 10% damage in the first mooring line alone ($k_1 = 0.9k$), and no damage in other mooring lines, while under operating condition case 8, the numerical experiment assumes 20% damage in each of all the three mooring lines. To simulate real-world conditions, the observation vector was contaminated with 1% and 5% Gaussian noise. The objective was to assess the efficiency and robustness of the proposed detection approach in estimat-

[1] $A + B\mathbb{N}(\mu, \sigma)$ means $A + Bz$, where z follows $\mathbb{N}(\mu, \sigma)$

Algorithm 1 DNN-Particle Filter

Inputs:
N: Number of particles
$x(0)$: Initial state estimate
$p(x(0))$: Initial state probability distribution
$f(x_k, u_k)$: System dynamics function
$h_k(x_k, \theta_k)$: Measurement model function
$DNN(x_k, \theta_k)$: DNN surrogate model for measurement update STATE σ_ξ: Standard deviation for particle perturbation
α: Hyper-parameter for particle re-centering
Q_k: Process noise covariance
R_k: Measurement noise covariance
Outputs:
Estimated state posterior: $p(x_k|z_1 : k)$ (represented by particles)
Estimated parameter vector: θ_k
Initialization:
for $i = 1$ **to** N **do**
  Sample initial state: $x_0^{(i)} = \text{Sample\_From}(p(x(0)))$
  Initialize parameter particles: $\theta_0^{(i)} = \text{Random\_Vector}()$
  Initialize weights: $w_0^{(i)} = 1/N$
end for
Main Loop:
for $k = 1$ **to** T (number of time steps) **do**
  Prediction Step:
  for $i = 1$ **to** N **do**
    Propagate particle state: $x_k^{(i)} = f(x_{(k-1)}^{(i)}, u_k)$
    Perturb parameter particle: $\theta_k^{(i)} = \alpha\theta_{(k-1)}^{(i)} + \text{N}(0, \sigma_\xi)$
  end for
  Measurement Update Step:
  for $i = 1$ **to** N **do**
    Calculate innovation: $\text{innovation} = y_k - h_k(x_k^{(i)}, \theta_k^{(i)})$
    Calculate likelihood: $\mathcal{L}(\xi_k^j) = \left((2\pi)^n\sqrt{|\mathbf{R}_k|}\right)^{-1} \exp\left(-0.5(\boldsymbol{i}_k^j)^T\mathbf{S}_k^{-1}\boldsymbol{i}_k^j\right)$
    Update weight based on likelihood: $w_k^{(i)} = w_{(k-1)}^{(i)} \cdot \mathcal{L}(\xi_j^k)$
  end for
  Normalize weights: $w_i = \frac{w_i}{\sum_{j=1}^N w_j}$
  Resampling Step:
  Perform resampling to generate new particles and parameters based on weights
  Update particles and parameters based on resampling results
end for

ing the stiffness parameters (k_1, k_2, and k_3) of the mooring lines across varying noise levels.

Results indicate that, regardless of the noise level, the integrated model effectively estimated the states and stiffness parameters with sufficient accuracy. Figures 3, 4, 5, and 6 demonstrate the proposed framework's parameter estimation capabilities even in the presence of noise. However, an increase in noise led to decreased accuracy in parameter estimation. Higher noise levels introduced greater ambiguity into the observation vector, resulting in increased inaccuracy in the estimation process. The accuracy of the estimated parameters is quantified in terms of RMSE, with lower RMSE values indicating better accuracy.

Although all 8 scenarios in Table 2 were studied with the proposed estimation framework, only the results corresponding to Case 2 and Case 8 are presented in the paper for the numerical demonstration due to technical limitations.

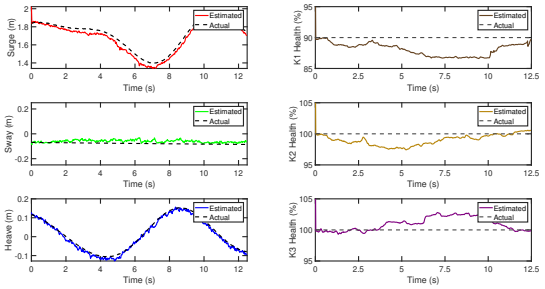

Figure 3. Actual and estimated states (left) and stiffness (right) parameters for Case 2 with 1% noise.

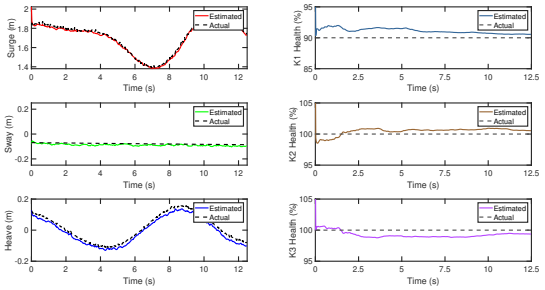

Figure 4. Actual and estimated states (left) and stiffness (right) parameters for Case 2 with 5% noise.

4. Conclusion

The initial findings suggest that the DNN-particle filter holds promise as a means to enhance the reliability and efficiency of mooring line monitoring by integrating an ML-based predictor model instead of a high-fidelity FEM model. By leveraging synthetic data generated from a calibrated OpenFAST model of FOWT dynamics, the DNN-particle filter offers a

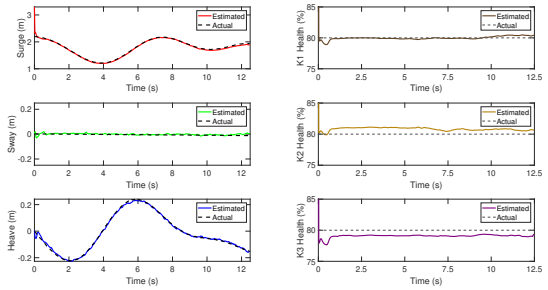

Figure 5. Actual and estimated states (left) and stiffness (right) parameters for Case 8 with 1% noise.

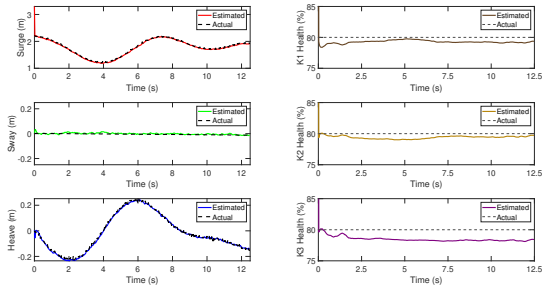

Figure 6. Actual and estimated states (left) and stiffness (right) parameters for Case 8 with 5% noise.

computationally efficient representation of system dynamics, enabling real-time damage detection and interpretable information on damage severity within a stochastic inverse estimation framework. This contrasts with traditional black-box ML-based methods, which often struggle to provide interpretable information on damage characteristics.

The DNN-particle filter's data processing capabilities maximize resource allocation by focusing efforts on critical areas identified by the model. By doing so, it enhances the promptness of the detection algorithm without sacrificing accuracy and transparency. This development has the potential to usher offshore operations into a new era of durability and resilience by ensuring the integrity of mooring lines and streamlining operating procedures. Ultimately, it has the potential to enhance the safety and longevity of offshore operations by effectively managing damage and noise.

REFERENCES

Altuzarra, J., Herrera, A., Matías, O., Urbano, J., Romero, C., Wang, S., & Guedes Soares, C. (2022). Mooring system transport and installation logistics for a floating offshore wind farm in lannion, france. *Journal of marine science and engineering*, *10*(10), 1354.

Aqdam, H. R., Ettefagh, M. M., & Hassannejad, R. (2018). Health monitoring of mooring lines in floating structures using artificial neural networks. *Ocean Engineering*, *164*, 284–297.

Avci, O., Abdeljaber, O., & Kiranyaz, S. (2022). An overview of deep learning methods used in vibration-based damage detection in civil engineering. In *Dynamics of civil structures, volume 2: Proceedings of the 39th imac, a conference and exposition on structural dynamics 2021* (pp. 93–98).

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.

Choe, D.-E., Kim, H.-C., & Kim, M.-H. (2021). Sequence-based modeling of deep learning with lstm and gru networks for structural damage detection of floating offshore wind turbine blades. *Renewable Energy*, *174*, 218–235.

Farrar, C. R., Doebling, S. W., & Nix, D. A. (2001). Vibration–based structural damage identification. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *359*(1778), 131–149.

Gorostidi, N., Pardo, D., & Nava, V. (2023). Diagnosis of the health status of mooring systems for floating offshore wind turbines using autoencoders. *Ocean Engineering*, *287*, 115862.

Jamalkia, A., Ettefagh, M. M., & Mojtahedi, A. (2016). Damage detection of tlp and spar floating wind turbine using dynamic response of the structure. *Ocean Engineering*, *125*, 191–202.

Jonkman, J., Butterfield, S., Musial, W., & Scott, G. (2009, 2). Definition of a 5-mw reference wind turbine for offshore system development. Retrieved from `https://www.osti.gov/biblio/947422` doi: 10.2172/947422

Li, Y., Le, C., Ding, H., Zhang, P., & Zhang, J. (2019). Dynamic response for a submerged floating offshore wind turbine with different mooring configurations. *Journal of Marine Science and Engineering*, *7*(4), 115.

Malekloo, A., Ozer, E., AlHamaydeh, M., & Girolami, M. (2022). Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. *Structural Health Monitoring*, *21*(4), 1906–1955.

Regan, T., Beale, C., & Inalpolat, M. (2017). Wind turbine blade damage detection using supervised machine learning algorithms. *Journal of Vibration and Acoustics*, *139*(6), 061010.

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* (Tech. Rep.). National Renewable Energy Lab.(NREL), Golden, CO (United States).

Wang, P., Tian, X., Peng, T., & Luo, Y. (2018). A review

of the state-of-the-art developments in the field monitoring of offshore structures. *Ocean Engineering*, *147*, 148–164.