Machine Learning-Based Predictive Models for Multi-Hazard Response Analysis of Offshore Wind Turbines

TANMOY CHATTERJEE and SUBHAMOY BHATTACHARYA

ABSTRACT

The growing deployment of offshore wind turbines (OWTs) demands efficient structural health monitoring (SHM) strategies to ensure safety and reliability under multi-hazard conditions. OWTs experience complex dynamic interactions from wind, waves, and seismic forces, making real-time predictive capabilities essential for condition-based maintenance. However, high-fidelity simulations such as finite element models (FEM) are computationally intensive for real-time use. This study presents an ensemble machine learning framework as a fast surrogate for FEM-based simulations, enabling rapid prediction of key structural responses including hub displacement, acceleration, and pile head rotation. The model captures intricate relationships between key system parameters, including wind turbine capacity, offshore environmental conditions (wind speed, wave height, wave period, and ground acceleration), and site-specific factors (water depth, soil properties). Model performance and generalisation are systematically assessed using a statistical learning workflow. The proposed surrogate model significantly reduces computational cost while retaining predictive accuracy, supporting real-time SHM integration and facilitating informed decision-making. This approach advances AI-driven SHM for offshore renewables, providing scalable, data-efficient solutions to enhance resilience, operational efficiency, and sustainability.

INTRODUCTION

Offshore wind energy has become a cornerstone of the global transition to low-carbon power, offering large-scale energy production with minimal visual and acoustic impact [1, 2]. As offshore wind farms expand into deeper waters and harsher environments, structural designs must adapt to increasing turbine size, flexible towers, diverse soil conditions, and elevated environmental loads from wind, waves, and seismic activity [3].

Tanmoy Chatterjee, School of Engineering, University of Surrey, Stag Hill Campus, Guildford GU2 7XH, United Kingdom
Subhamoy Bhattacharya, School of Engineering, University of Surrey, Stag Hill Campus, Guildford GU2 7XH, United Kingdom

These complexities demand high-fidelity simulations, such as finite element models (FEM), to capture critical nonlinear interactions, particularly those involving pile-soil dynamics, hydrodynamic forces, and coupled structural responses [4]. However, generating diverse datasets by executing such simulations is computationally intensive, especially when accounting for the wide variability in site conditions, structural configurations, and multi-hazard scenarios [5].

To reduce reliance on expensive simulations and enable real-time decision-making, machine learning (ML) offers a promising alternative. Trained on FEM-generated data, ML-based surrogate models can rapidly predict structural responses under varying conditions [6]. Yet, their deployment in safety-critical applications like structural health monitoring (SHM) requires careful evaluation of their predictive accuracy, generalization ability, and reliability under diverse operating scenarios.

This study addresses these challenges by developing and validating an ensemble ML framework as a fast surrogate for FEM-based simulations of monopile supported offshore wind turbines under multi-hazard loading. The framework aims to capture the complex relationships between turbine configuration, site-specific factors, and environmental conditions, and to assess the model's generalization across diverse offshore scenarios.

MODELLING AND ANALYSIS OF THE OFFSHORE WIND TURBINE

Three-dimensional finite element modelling of the wind turbine structure was conducted using the commercial software ABAQUS. Steel components were defined using material grade S355, incorporating a full elasto-plastic stress–strain relationship. Major structural components including the blades, nacelle, tower, and monopile were modelled using beam elements.

The blades were represented by B31 beam elements with orthotropic stiffness properties and tapered geometry, based on data from the NREL database. The nacelle was modelled as a rigid beam element, connecting the blade assembly to the tower via discrete springs and viscous dampers. The tower was modelled using beam elements with a tapered cross-section.

The monopile foundation, as defined in this work, is the segment extending from the mid-point of the transition piece to the bottom tip. This segment includes the portion below the sea surface and above the mudline, which is subjected to wave loading.

Soil–structure interaction was modelled using the Winkler foundation approach, where the monopile–soil interaction was represented by nonlinear springs distributed along the embedded length of the pile. The spring stiffness was derived from empirical formulations given in the modified API and DNV codes. Three types of springs were used: p–y springs to model lateral soil resistance, t–z springs to capture vertical frictional resistance, and q–z springs to represent the bearing capacity at the pile tip. These springs were fixed at one end to resist rigid body motion and connected to the pile at the other end via zero-length connector elements.

A combined loading scenario incorporating earthquake, wind, and wave loads was applied in accordance with BS EN IEC 61400-1:2019 standards.

DATA-DRIVEN PREDICTIVE MODELLING VIA MACHINE LEARNING

This study focuses on monopile-supported wind turbines under normal operating conditions using a curated dataset of 31,409 labelled samples, derived from structural analyses of the finite element model described earlier. These analyses were conducted based on input parameter variations generated via Monte Carlo simulations. The normal operational conditions imply that the wind speed at the hub level (reference height of 10 m) will be in the operating range (*i.e.,* < 30 m/s). The data simulates a range of turbine capacities of 2-14 MW considering the variability of the turbine characteristics (*i.e.,* tower height, rotor nacelle mass, tower diameter, tower wall thickness, monopile length), water depth of 10-50 m and soil characteristics based on ground type C as per Eurocode 8. Although not considered in this work, for consistency note that the ground types D and E are less stiff and ground types A and B are stiffer compared to ground type C.

The input parameters considered for training the machine learning (ML) model are (i) peak ground acceleration (PGA) (ii) turbine capacity (iii) water depth (iv) wind speed (v) wave height and (vi) wave period. The above input parameters capture the following three aspects, (a) hazard (i); (b) technology (ii); and (c) environmental loads (iii - vi). The turbine capacity incorporates the structural and functional dimensions implicitly. The output response quantities of interest used for training the ML model are (i) maximum hub rotation (ii) maximum hub displacement (iii) maximum hub acceleration (iv) maximum pile head rotation due to earthquake and (v) maximum pile head rotation due to wind and wave loads. The selected output responses are also referred to as the effective design parameters (EDP). The histograms of the input parameters and the output responses for the ML training/testing are presented in Figures 1 and 2, respectively.

The ML model employed in this study for the supervised learning task is based on ensemble learning. Throughout this work, the term *ensemble learning* refers specifically to methods that aggregate decision tree-based learners. A gentle introduction to decision trees can be found in [7]. As the name suggests, ensemble learning combines multiple simple "building block" models known as *weak learners* into a single, potentially more powerful predictive model. These weak learners, while individually suboptimal, can collectively deliver strong performance through appropriate aggregation. Three popular ensemble methods bagging, random forests, and boosting are employed in this study and briefly outlined below:

Bagging (Bootstrap aggregation) is a general-purpose technique for reducing the variance of a statistical learner—in this case, decision trees. In bagging, B decision trees are trained independently on B bootstrapped samples from the original dataset and predictions are then averaged. However, since the trees are grown independently, they often end up being similar. This can limit the model's ability to explore the broader hypothesis space, potentially leading to local optima.

Random forests improve upon bagging by introducing a randomization step that reduces the correlation among trees. Specifically, each tree split considers only a random subset of m predictors (typically $m \approx \sqrt{p}$), where p is the total number of predictors). By preventing strong predictors from dominating every split, this decorrelation leads to more diverse trees and a more stable ensemble. Notably, if $m = p$, random forests reduce to standard bagging.

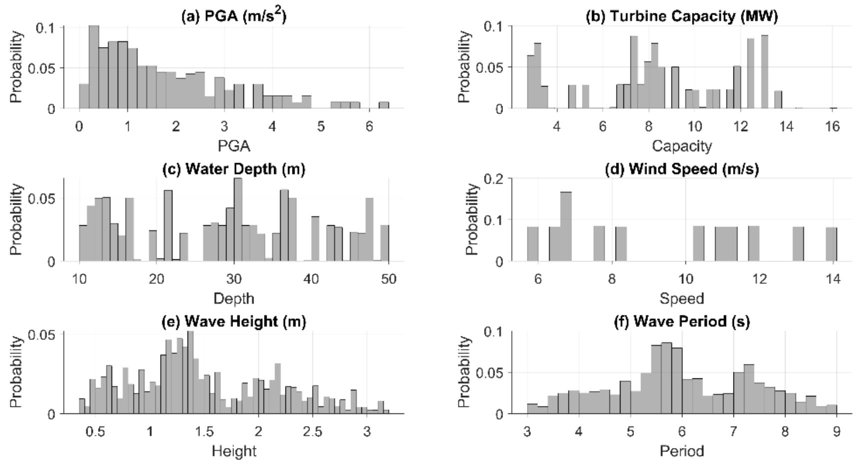

Figure 1. Histograms of the input parameters

Boosting, unlike bagging and random forests, builds trees sequentially. Each tree is trained to correct the residual errors of its predecessors. Rather than relying on bootstrapped samples, boosting fits each tree on a modified version of the original data, with greater focus on previously mispredicted observations. A learning rate parameter shrinks each tree's contribution, facilitating a 'slow learning' process that often improves generalization. For a more rigorous and detailed discussion of these methods, the reader is referred to [8].

In this work, we implemented these ensemble learning models using MATLAB's built-in function '*fitrensemble'*. Model hyperparameters were automatically tuned using Bayesian optimization [9] with the Automatic Relevance Determination (ARD) Matérn 5/2 kernel in Gaussian Process regression [10]. The search space included the choice of aggregation algorithm (bagging, random forest, or least-squares boosting), the number of learning cycles (ranging from 10 to 500), and the learning rate (ranging from 1e-3 to 1).

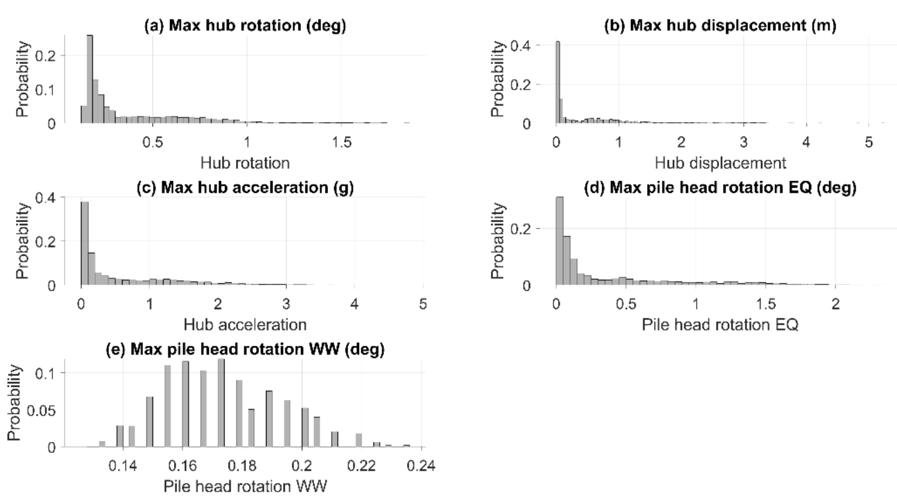

Figure 2. Histograms of the output response quantities

The optimization was guided by the average 5-fold cross-validation (CV) error using the *expected improvement* acquisition function (AF) [11]. For time-weighting, preventing over-exploitation of local minima and ensure global exploration, the '*expected-improvement-per-second-plus*' AF variant was employed [12]. A generalized workflow for the training and evaluation of the ensemble learning setup is provided in the next section (see Table I and Algorithm 1).

RESULTS AND DISCUSSION

The performance of the ensemble learning model in capturing the data trends has been assessed. The methodology adopted has been linked to the key model attributes and performance in Table I to provide diagnostic insights and enable real-time ML deployment. Furthermore, a systematic and generalized workflow of the ML model training and evaluation has been presented in Algorithm 1. Note that the workflow presented is completely generalized to accommodate any ML model, sample size, number of seeds, splitting scheme of train/test datasets, number of outputs, hyperparameter optimization strategy and acquisition function type, performance evaluation metrics and analysis as relevant to the user preferences and application requirements.

TABLE I. PERFORMACE ASSESSMENT OF THE ML MODEL ACROSS CATEGORIES

Category	**Methodology**
Robustness/efficiency	Multiple train/test splits
Optimization/tuning	Hyperparameter optimization on training data
Generalization	k-fold cross validation
Epistemic uncertainty quantification	Seed-based variance analysis
Error analysis	Train/CV vs test error bars with sample sizes
Evaluation metrics	Root mean squared error: RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$; Coefficient of determination: $R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$
Model reliability/generalization	Bias variance trade-off

Following the generalized workflow presented in Algorithm 1, the convergence of the CV and test RMSE and test R^2 with increase in the training sample size in Figure 3 clearly indicate the strong potential of the ensemble learning model in capturing the nonlinear trends of the effective design parameters.

The low CV error even at small sample sizes reveals that the model can learn relevant patterns from the data quickly and indicates low bias (probability of underfitting). The variance (chance of overfitting) is controlled as indicated by the stable nature of the test error and its proximity to the CV error. Even though the test RMSE is slightly lower than that of the CV RMSE values, their small difference across the training fractions (indicated by the dashed line) suggests stable learning dynamics and balanced bias variance tradeoff illustrating good generalization ability. This is crucial for deploying reliable ML models in safety-critical offshore scenarios, where robustness to unseen conditions is key.

ALGORITHM 1. PSEUDOCODE FOR MODEL TRAINING AND EVALUATION

For each training size T in predefined set:
    For each random seed S in $\{1, 2, ..., N\}$:
        Randomly split dataset into training (**X**_train, **Y**_train)
        and testing (**X**_test, **Y**_test) sets using seed S

        For each output variable o in $\{1, 2, ..., n\}$:
            Extract ***y***_train and ***y***_test for the o^{th} output

            Perform **Bayesian optimization** to tune model hyperparameters (HP):
                Use **k-fold cross-validation** (*e.g.*, 5-fold) on (**X**_train, ***y***_train)
                Evaluate HP combinations via the iterative search on the updated design space
                Select the best hyperparameters based on average CV performance

            Store the mean and standard deviation of CV errors for the optimal model

            Train the final model using full (**X**_train, ***y***_train) and optimal hyperparameters

            Evaluate the model on (**X**_test, ***y***_test)
            Store the test performance metrics (e.g., RMSE, R^2)
        End for output o
    End for seed S
End for training size T

Aggregate and summarize CV, test performance metrics across seeds for each training size and output.

The true and the predicted test EDP with varying PGA, grouped by turbine capacity ranges have been compared in Figure 4. These have been smoothed using a centered moving average to enhance the visualization of de-noised global response trends with respect to PGA across different turbine capacities. The predicted responses were obtained using the ensemble model trained with a sample size/fraction of 0.5. The predicted (dotted) and true (in solid) test responses are in proximity with each other consistently across the PGA and turbine capacity range.

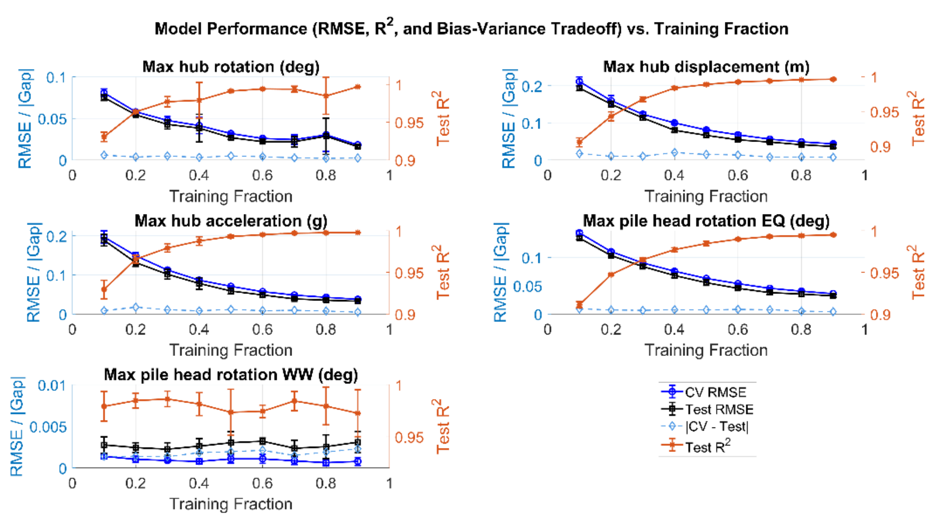

Figure 3. Error convergence and bias-variance tradeoff with varying train/test split

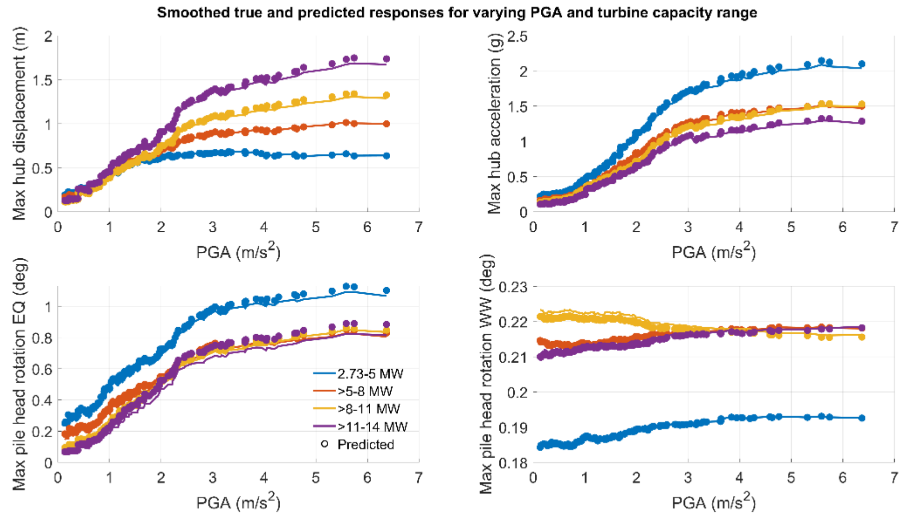

Figure 4. Comparison of the true and predicted responses with peak ground acceleration for different turbine capacity ranges. The training fraction of the ML model used is 0.5.

This demonstrates that ML predictions can offer valuable insights for assessing risks associated with monopile-supported OWTs of a given rated capacity (or range), and for anticipating long-term operational and maintenance costs based on parameters such as water depth, soil type, and PGA.

CONCLUSION

This study demonstrated the effectiveness of an ensemble learning-based predictive modelling framework for estimating critical structural responses (EDP) of offshore wind turbines under multi-hazard loading conditions. The trained models achieved a satisfactory level of accuracy while exhibiting robust generalization capabilities across unseen datasets. This highlights their potential for deployment in real-time structural health monitoring (SHM) systems, where rapid response prediction is essential for informed decision-making. The proposed framework offers a scalable and data-efficient alternative for risk assessment and damage prediction. Such a shift not only accelerates predictive maintenance workflows but also supports proactive asset management strategies in offshore renewable energy infrastructures.

Future work will explore the integration of transfer learning techniques to extend the applicability of these models across diverse offshore environments and configurations. Additionally, efforts will be made to enhance the models' extrapolation capabilities and incorporate domain adaptation methods, thereby improving their reliability under previously unencountered operational scenarios. Overall, this research contributes to the development of AI-driven, real-time digital twin frameworks for resilient and sustainable offshore energy systems.

REFERENCES

1. GWEC, Global Wind Energy Council 2017. "*Global wind report, 2016 annual market update.*" Brussels, Belgium, p. 76.
2. Bhattacharya, S. and Nikitas, G. 2025. "Wind energy: status and outlook with focus on offshore wind", in *Energy and Climate Change*, Ch. 8:197-224.
3. Sunday, K. and Brennan, F. 2021. "A review of offshore wind monopiles structural design achievements and challenges," *Ocean Eng.,* 235:109409.
4. Sahu, S. K., Kumar, V., Dutta, S. C., Sarkar, R., Bhattacharya, S. and Debnath, P. 2024. "Structural safety of offshore wind turbines: present state of knowledge and future challenges," *Ocean Eng*., 309: 118383.
5. Zhang, Z., Risi, R.D. and Sextos, A. 2024. "Multi-hazard fragility assessment of monopile offshore wind turbines under earthquake, wind and wave loads," *Earthquake Engng Struct Dyn.*, 52:2658-2681.
6. Chatterjee, T., Friswell, M.I., Adhikari, S. and Chowdhury, R. "A global two-layer meta-model for response statistics in robust design optimization," *Eng. Opt.*, 54(1): 153-169.
7. James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J. 2023. "Tree-Based Methods," in *An Introduction to Statistical Learning*, Springer Texts in Statistics. Cham: Springer.
8. Hastie, T., Tibshirani, R., and Friedman, J. 2008. *The Elements of Statistical Learning*, Second Edition. New York: Springer.
9. Jones, D. R., Schonlau, M., and Welch, W. J. 1998. "Efficient Global Optimization of Expensive Black-Box Functions," *J. Global Optim.*, 13(4):455–492.
10. Rasmussen, C. E., and Williams, C. K. I. 2006. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.
11. Jones, D. R. 2001. "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *J. Global Optim.*, 21(4):345–383.
12. Bull, A. D. 2011. "Convergence Rates of Efficient Global Optimization Algorithms," *J. Machine Learning Research*, (3–4):2879–2904.