



A data-driven model framework for predicting operational and load responses of the IEA 3.4 MW wind turbine

Iasonas Tsaklis¹, Vasilis Pettas¹, Adrien Guilloré², Nicolau Conti Gost¹, David Astrain-Juangarcia⁴, Abhinav Anand^{*2,3}, Andreas Vad², Anik H. Shah², Ion Lizarraga-Saenz⁴, Maria Aparicio-Sanchez⁴, Irene Eguinoa⁴, Carlo L. Bottasso², Nikolay Dimitrov¹, and Tuhfe Göçmen¹

¹DTU Wind and Energy Systems, Technical University of Denmark, Roskilde, Denmark

²Wind Energy Institute, Technical University of Munich, Boltzmannstraße 15, 85748 Garching b. München, Germany

³Nordex Energy SE & Co. KG, Langenhorner Chaussee 600, 22419 Hamburg, Germany

⁴Wind Energy Department, National Renewable Energy Centre (CENER), Spain, Ciudad de la Innovación, 7, 31621 Sarriren (Navarra), Spain

Correspondence: Tuhfe Göçmen (tuhf@dtu.dk)

Abstract.

This paper presents a transparent and reproducible machine-learning framework for developing surrogate models that predict operational variables and structural loads of the IEA 3.4 MW wind turbine. The framework is based on a comprehensive dataset of aeroelastic simulations generated across multiple operational modes and wind conditions using HAWC2. Several surrogate modeling approaches are investigated, including Artificial Neural Networks, Gaussian Process Regression and Random Forests. The models are trained and evaluated using multiple feature engineering strategies and data scaling approaches to assess their predictive performance and robustness across different response variables. The primary objective of this work is not to identify a single optimal surrogate model, but to provide a transparent, accessible, and reproducible workflow for surrogate model development for optimization-friendly structural loads and operational response estimation. The article documents the complete process of dataset generation, feature engineering, model training, and performance evaluation, enabling consistent comparison of machine-learning approaches and facilitating future methodological developments. By providing both the simulation dataset and the associated surrogate-modeling workflow, this work aims to support the development, benchmarking, and adoption of scientific machine-learning-based surrogate models within the wind energy research community.

*The author is currently affiliated with ³Nordex Energy SE Co. KG, however, his contributions to this work were purely carried out as part of his previous affiliation at the ²Wind Energy Institute, Technical University of Munich.



1 Introduction

15 Wind energy has become one of the dominant technologies supporting the global energy transition, and its installed capacity continues to increase worldwide. At the same time, modern wind turbines are becoming larger, and wind farms are increasing in both size and complexity. Because of the large amount of energy produced by utility-scale wind farms, even modest improvements in operational efficiency or structural load mitigation can translate into substantial economic benefits through increased annual energy production (AEP), reduced maintenance costs, and extended turbine lifetime.

20 These challenges have motivated extensive research in wind farm control (WFC), which aims to optimize the collective operation of wind turbines by coordinating control actions across the farm. Early WFC research primarily focused on increasing AEP and reducing structural loading through wake steering and coordinated control strategies (Veers et al. (2019); Knudsen et al. (2015); Kheirabadi and Nagamune (2019); Wingerden et al. (2020)). More recently, the field has evolved towards a broader systems perspective that additionally considers economic objectives, power system services, environmental impact, and overall wind farm performance (Veers et al. (2022); Eguinoa et al. (2021)).

The development and assessment of advanced WFC strategies rely heavily on high-fidelity wind flow and aeroelastic simulations. Although these models provide detailed predictions of turbine behaviour, they are computationally expensive and therefore impractical for applications requiring thousands of repeated evaluations, such as controller optimization, uncertainty quantification, or large-scale design studies. Field measurements provide an alternative source of information but are often 30 proprietary, difficult to access, and limited in their coverage of operating conditions.

Several surrogate-modelling methodologies have been investigated for predicting wind-turbine operational responses and structural loads using either simulation data or field measurements. Polynomial chaos expansions have demonstrated their ability to approximate aeroelastic responses while providing a computationally efficient means of propagating input uncertainty (Dimitrov et al. (2018)). Gaussian Process Regression (GPR) has also been studied extensively, with its predictive accuracy 35 and suitability for relatively small training datasets being highlighted in Singh et al. (2022). Artificial Neural Networks (ANNs) remain among the most widely investigated model families because of their flexibility, scalability, and ability to represent complex nonlinear relationships between inflow conditions and turbine responses (Bay et al. (2026); Nolasco Santos et al. (2023); Vad et al. (2026); Schröder et al. (2018)). Tree-based ensemble methods have received comparatively less attention in this context, although initial studies have reported competitive predictive performance at low training and inference costs, particularly 40 for gradient-boosted decision trees (Tsaklis et al. (2026a)). Bayesian neural networks have likewise shown promising results by combining nonlinear predictive capability with estimates of predictive uncertainty (Dimitrov et al. (2022); Konis et al. (2026); Hlaing et al. (2023)). Mixture Density Networks (MDNs) have also shown promising results, offering the ability to capture complex response distributions and quantify predictive uncertainty, particularly aleatoric uncertainty (Singh et al. (2024)).

Irrespective of the selected model family, the representativeness of the training data remains central to surrogate-model performance. In particular, wake-induced and partially waked inflow conditions introduce spatial variations across the rotor plane 45 that cannot necessarily be represented adequately by conventional inflow descriptors. Previous studies have consequently reported reduced load-prediction accuracy under waked conditions (Vera-Tudela and Kühn (2017); Tsaklis et al. (2026a)). To



address this limitation, Shaler et al. (2022) proposed extracting rotor-plane flow information to provide a more detailed representation of the inflow experienced by the turbine. The value of spatially resolved wake descriptors for surrogate load prediction was subsequently investigated in Guilloré et al. (2024); Ramaswamy et al. (2026). However, the datasets and complete modelling workflows underlying these studies were not made openly available, limiting their reproducibility and reuse.

More broadly, openly available aeroelastic datasets generated under well-defined and controlled conditions remain scarce. One notable contribution is the dataset presented by Pettas and Cheng (2024); however, its scope does not encompass several conditions relevant to realistic turbine operation, including yaw misalignment, multiple operational modes, and waked or partially waked inflow. The present work addresses these limitations by providing an openly documented aeroelastic simulation dataset that covers multiple operational modes, yaw-misalignment and down-regulation settings, and a broad range of free-stream, waked, and partially waked conditions. The dataset is accompanied by a reproducible surrogate-modelling framework implementing ANN, GPR, and Random Forest (RF) models. These model families include both well-established approaches, such as ANN and GPR, and a comparatively less explored tree-based approach whose performance for aeroelastic-response prediction warrants further investigation. Together, the dataset and framework provide a transparent foundation for developing, evaluating, and comparing surrogate models under a common set of controlled operating and inflow conditions.

This paper addresses this need by presenting a benchmark dataset and an accompanying surrogate-model framework based on high-fidelity aeroelastic simulations of the IEA 3.4 MW reference wind turbine (Bortolotti et al. (2019)). The dataset comprises several hundred thousand simulations covering a wide range of inflow conditions, yaw misalignments, and controller operating points, including both free-stream and waked conditions representative of wind farms. Alongside the dataset, a reproducible software framework is provided for data preprocessing, surrogate-model training, evaluation, and post-processing. The framework is organized into distinct components that can be used and adapted independently, allowing researchers, for example, to apply alternative surrogate-model methodologies to the provided dataset or to use the training and evaluation workflow with other compatible datasets.

Rather than proposing a new surrogate methodology, the objective of this work is to provide a transparent foundation upon which future surrogate-model studies can be developed and compared. The framework enables systematic benchmarking of different machine-learning algorithms, feature engineering strategies, and training procedures using a common dataset. It is intended to support future research on wind turbine response prediction, uncertainty quantification, controller development, and data-driven wind farm operation with a fully reproducible workflow and conditional open access for research organisations.

Beyond surrogate-model development, the dataset provides a resource for investigating how different inflow and operating conditions influence the aeroelastic response of the turbine. It can therefore support the identification of response trends, sensitivities, and dependencies across the investigated parameter space. Such insights may inform future developments in areas extending beyond load-surrogate modelling, including turbine design, control strategies, load assessment, and wind-farm operational analysis.

The remainder of this paper is organized as follows. Section 2 describes the design of experiment and the generation of the benchmark dataset through wind field and aeroelastic simulations. Section 3 presents the surrogate modeling framework, including data preprocessing, feature engineering, model training, and evaluation procedures. Section 4 summarizes the per-



formance of the surrogate model families using the adopted evaluation metrics and discusses the results. Finally, Section 5 concludes the paper, outlines its main contributions, and discusses potential directions for future work.

85 2 Design of experiment & Database generation

The general workflow adopted in this study is illustrated in Fig. 1. This section focuses on the first two stages of the workflow: wind-field creation and aeroelastic simulation. Section 2.1 describes the generation of the wind fields, the extraction of the wake slices, and the selection of the inflow conditions. Subsequently, Section 2.2 presents the setup of the HAWC2 aeroelastic model and controller, together with the processing and inspection of the resulting simulation outputs used to construct the

90 dataset.

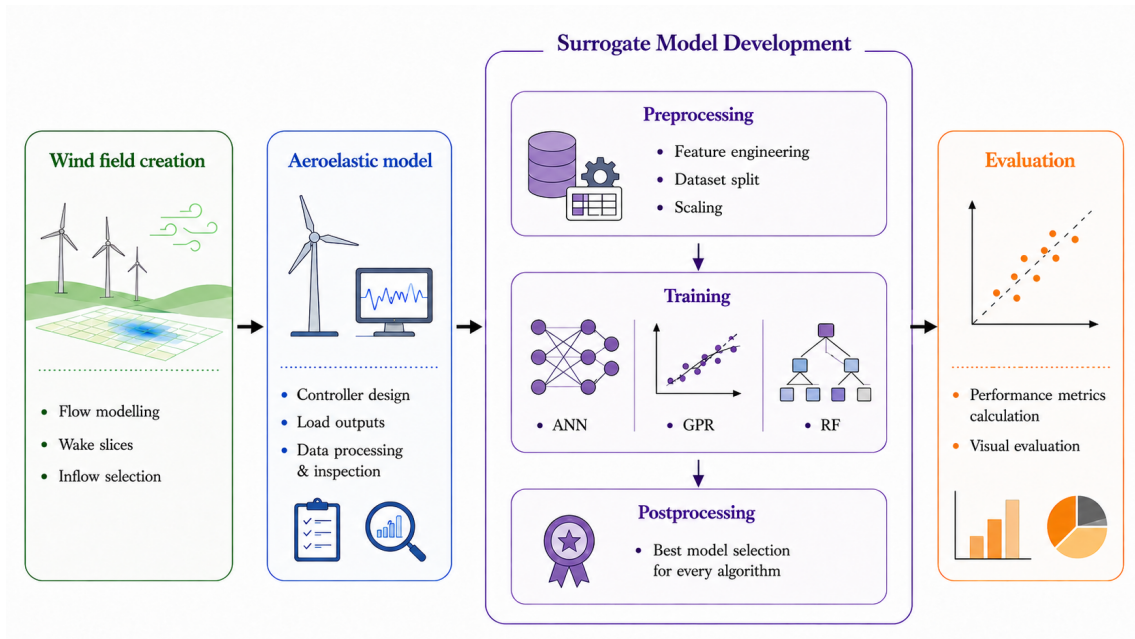


Figure 1. General workflow of the proposed surrogate-model development framework.

2.1 Wind field generation

The wind fields distributed as part of this dataset were generated using the wake-slice methodology introduced and validated by Guillore et al. (2026b), and subsequently applied within the integrated wind-farm response framework presented by Vad et al. (2026). The resulting inflow database is intended for the generation of wind-turbine load surrogate models based on

95 sector-averaged inflow quantities, following the load-surrogate formulation originally proposed by Guillore et al. (2024) and further validated in Ramaswamy et al. (2026). While the methodological developments are described in detail in these earlier



works, the objective of the present paper is to document and disseminate the resulting open-source dataset and the associated surrogate-model resources.

100 The database contains two categories of turbulent inflow conditions: clean inflow profiles and waked inflow profiles. Clean inflow profiles represent ambient free-stream turbulence, whereas waked inflow profiles include the effects of upstream turbine operation, such as velocity deficits, wake meandering and wake-added turbulence. Together, these inflow conditions span the range of operating environments required for the development of load surrogate models applicable to both isolated and wind-farm turbines.

105 For the clean inflow conditions, turbulent wind fields were generated using standard stochastic turbulence generation approaches commonly adopted in aeroelastic simulations. The Kaimal turbulence spectrum implemented in TurbSim (Jonkman, 2009) is used. The database covers a wide range of ambient wind speeds (3–26 m/s) and turbulence intensities (2–47 %) representative of wind-energy applications. Additional meteorological characteristics, such as wind-speed profiles, wind-direction profiles and shear conditions, can subsequently be superimposed depending on the requirements of the downstream aeroelastic simulations.

110 The waked inflow conditions were generated using the Dynamic Wake Meandering (DWM) model (Larsen, 2007) implemented in FAST.Farm (Jonkman et al., 2018). To capture the physical mechanisms driving turbine loading within a wind farm, FAST.Farm v4.0 (National Renewable Energy Laboratory, 2024) was used, which augments the classical DWM framework with wake-added turbulence (Kretschmer et al., 2021; Branlard et al., 2024) and curled-wake models (Branlard et al., 2023).

115 A single instance of the NREL-1.79-100 reference wind turbine (NREL, 2024), which has a diameter of 100 m, was used as a wake-generating turbine. The resulting wake flow fields were sampled at multiple locations downstream of the rotor to create a library of free-stream, partially waked and fully waked inflow conditions. The objective was to generate a diverse dataset representative of the range of wake interactions encountered within wind farms while preserving the spatial and temporal coherence of the turbulent flow field. This strategy allows large databases of waked inflow conditions to be produced from a limited number of wind-farm simulations, as demonstrated in Guilloré et al. (2026b).

120 To ensure broad coverage of relevant environmental conditions while maintaining a tractable computational cost, the ambient conditions used for wake generation were selected using a Latin Hypercube Sampling (LHS) (McKay et al., 1979) following the methodology described in Vad et al. (2026). The sampling was designed to provide representative coverage of the inflow parameter space while minimizing the number of required wind-farm simulations. Since the surrogate methodology relies on local inflow descriptors derived from four sector-averaged wind speeds (SAWS) and four sector-averaged turbulence intensities (SATI), particular attention was given to ensuring that the sampled ambient conditions adequately populated the resulting eight-dimensional inflow space. Additional edge cases were included to limit surrogate extrapolation and improve coverage of extreme wake situations. A total of 83 combinations of ambient wind conditions, defined by mean wind speed and turbulence intensity, were ultimately retained through this sampling procedure to populate the library of inflow profiles.

130 Dynamic operating conditions of the wake-generating turbine were simulated in FAST.Farm across the selected ambient conditions. During the simulations, the three-dimensional flow solution was exported as high-resolution `.vtk` files at regular time intervals. These files contain the local velocity field resolved on the FAST.Farm domain and provide a computationally



efficient basis for extracting large numbers of wake realizations through post-processing without requiring additional wind-farm simulations.

The exported flow fields were subsequently processed to reconstruct reduced wake slices at user-defined locations relative to the wake-generating turbine, following the procedure described in Guilloré et al. (2026b). The selected streamwise locations were

$$X/D = [-0.5, 4, 5, 6, 7, 8], \quad (1)$$

while the lateral positions were

$$Y/D = [-1, -0.75, -0.50, -0.25, 0, 0.25, 0.50, 0.75, 1], \quad (2)$$

where D denotes the rotor diameter of the NREL-1.79-100 turbine (NREL, 2024). This sampling strategy generated a wide variety of inflow conditions, ranging from unwaked free-stream operation to strong wake-impact scenarios. In addition, the intermediate lateral positions capture asymmetric partial-wake conditions, which are commonly encountered in practical wind-farm operation and are particularly relevant for the sector-based load surrogate methodology of Guilloré et al. (2024). In total, the six downstream positions and nine lateral positions yielded 54 wake slices for each FAST.Farm ambient-condition case.

All FAST.Farm simulations were run for 800 s. A 200 s initialization period was included to allow the wake to develop and propagate through the domain, after which the remaining 600 s were retained for load analysis and inflow-field extraction. Each operating condition was simulated using six independent turbulent realizations to ensure statistical convergence of the extracted inflow characteristics and resulting load metrics.

For each extraction location, a square inflow domain of 120×120 m was reconstructed around the wake center. These dimensions were selected to retain the relevant wake structures while keeping the resulting files computationally manageable for large-scale aeroelastic simulation campaigns. The extracted fields were discretized using a spatial resolution of 5 m in both spatial directions and a temporal resolution of 0.1 s. Particular attention was given to these resolutions because accurate representation of fatigue-driving turbulence structures and wake-meandering phenomena is essential for reliable estimation of damage equivalent loads (DELs). As discussed by Liew and Larsen (2022), coarse spatial or temporal discretization may significantly affect the resulting fatigue predictions. The adopted discretization therefore preserves the turbulent fluctuations and wake gradients required for the generation of accurate load-surrogate datasets.

The resulting database consists of time-resolved wake slices containing the full spatial variation of the local wind velocity over the extraction plane. Because all slices originate from the same underlying FAST.Farm simulations, they retain the physical consistency of the modeled wake development while enabling a systematic investigation of wake effects as a function of downstream distance and lateral offset. The wake-slice concept allows the expensive wind-farm simulations to be performed only once, while the resulting inflow fields can subsequently be reused for multiple aeroelastic simulation campaigns, turbine models and controller configurations.



Following the framework proposed by Guilloiré et al. (2024) and Ramaswamy et al. (2026), the local inflow contained within each wind field can be decomposed into four rotor sectors, from which sector-averaged wind speeds (SAWS) and sector-averaged turbulence intensities (SATI) are derived. These quantities constitute the principal inflow descriptors used by the load-surrogate models distributed with the present dataset. The use of sector-averaged quantities enables the representation of free-stream, waked, wake-steered and partially waked operating conditions through a common set of local inflow parameters.

The generated wake slices were subsequently converted into turbulent inflow files suitable for aeroelastic simulations. When required by the downstream aeroelastic model, the inflow fields were spatially rescaled to match the dimensions of the target rotor while preserving the underlying wake characteristics and turbulent-flow structures. For larger rotor diameters, the original 120×120 m extraction domain was expanded to 132×132 m to ensure full coverage of the rotor-swept area.

The complete library of clean and waked turbulent inflow files described in this section is openly available on Zenodo at (Guilloiré et al., 2026a). This repository contains the inflow datasets generated within the framework of Vad et al. (2026), including the wake-slice databases used as the basis for the present work. The wind fields generated through this procedure constitute the inflow database used to develop the surrogate models distributed with the present dataset.

Readers interested in the methodological details of the sector-averaged inflow descriptors, the wake-slice generation approach, the associated validation studies, and the broader wind-farm-response framework are referred to Guilloiré et al. (2024), Guilloiré et al. (2026b), and Vad et al. (2026).

2.2 Aeroelastic simulation setup and post-processing

All aeroelastic simulations were performed using the IEA 3.4 MW land-based reference wind turbine, developed within the IEA Wind Task 37 (Bortolotti et al., 2019). The model was developed as a reference representation of a modern land-based wind turbine and was selected for the present work because its aerodynamic and structural definitions are openly available, allowing the simulation database to be documented and reproduced without the confidentiality restrictions associated with commercial turbine models. Furthermore, its multi-megawatt rating and dimensions are representative of the scale of currently deployed onshore turbines and is also relevant to the TWAIN ((European Commission, 2023)) project application studies. The main characteristics of the turbine are summarized in Table 1.

The turbulent wind fields described in Sect. 2.1 are used as inflow to standalone aero-servo-elastic simulations of the IEA 3.4 MW reference wind turbine. The turbine is simulated using HAWC2 (Larsen and Hansen (2007)), together with the DTU Wind Energy Controller (Meng et al. (2020)), which is subsequently adapted to represent the different operational and control modes described below. The complete reference-turbine definition is openly available through the IEA Wind Systems GitHub repository², including its HAWC2 implementation. The HAWC2 implementation of the reference turbine is based on the openly available IEA 3.4 MW model, while the simulation setup follows the general methodology adopted within the TWAIN wind-farm response framework (Vad et al., 2026) and in related applications using the same turbine model (Pettas et al., 2026). Each aeroelastic simulation has a total duration of 800 s. The first 200 s are discarded to remove initialization transients and allow the turbine and inflow response to develop, while the remaining 600 s are retained for the calculation of the operational and

²<https://github.com/IEAWindSystems/IEA-3.4-130-RWT>



Table 1. Main characteristics of the IEA 3.4 MW land-based reference wind turbine used for the aeroelastic simulations Bortolotti et al. (2019).

Parameter	Value
Rated electrical power	3.4 MW
Rotor diameter	130 m
Hub height	110 m
Number of blades	3
Rotor configuration	Upwind, horizontal-axis
Drivetrain configuration	Geared
IEC turbine class	IIIA

structural-response quantities, corresponding to a standard 10-min evaluation period. For each inflow and control condition, six independent turbulent realizations are simulated to account for seed-to-seed variability and provide statistically representative response quantities. The aeroelastic equations are integrated using a simulation time step of 0.01 s. To facilitate reproducibility of the database generation process, the HAWC2 input files (.htc) used for the different simulation configurations are provided together with the accompanying dataset.

The standard DTU Wind Energy Controller (Meng et al., 2020) is used as the baseline controller and adapted to represent the operational and control conditions included in the simulation database. Following the TWAIN wind-farm response framework (Vad et al., 2026), four operational modes are considered: power production (PP), start-up (SU), shut-down (SD), and parked (PK). During PP operation, the turbine response is characterized under normal operation and under two control actions relevant to wind farm control: power down-regulation and static yaw misalignment. These control actions are considered both independently and in combination in order to span the range of turbine operating conditions associated with power-based and yaw-based wind farm control. For SU and SD, power down-regulation is retained as an applicable control input, whereas yaw misalignment is not considered, under the assumption that the turbine is aligned with the incoming wind before performing the corresponding transition. No active control setpoints are prescribed in PK operation.

Power down-regulation is represented by normalized power setpoints of $p_{\text{set}} = \{0.85, 0.90, 0.95, 1.00\}$, corresponding to 85–100% of the baseline power trajectory in increments of 5%. The controller follows the constant tip-speed-ratio (Iso-TSR) down-regulation approach described in (Pettas and Cheng, 2018, 2024), in which the requested reduction in power is achieved primarily through a reduction in generator torque rather than rotor speed. This choice preserves the rotor-speed trajectory and consequently avoids shifting the rotational excitation frequencies of the turbine, while previous aeroelastic analyses have shown that the strategy can provide substantial reductions in structural fatigue loading (Pettas, 2024). The selected 85–100% range was chosen specifically for the wind-farm-control applications targeted by the database. For the considered turbine, the aerodynamic design and the resulting $(C_P)-(C_T)$ characteristics lead to a comparatively steep reduction in thrust over this range for a relatively moderate reduction in power. Consequently, operating within this region can provide meaningful reductions in turbine loading and wake strength while limiting the associated energy penalty. At lower power setpoints, the additional



220 reduction in thrust and structural loading becomes progressively smaller relative to the additional loss of power, making these operating points less attractive for the applications considered here. Yaw-based control is represented by static yaw misalignments between -30° and $+30^\circ$ in increments of 10° , i.e. $\gamma \in \{-30^\circ, -20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ, 30^\circ\}$. The range covers both positive and negative wake-steering directions commonly considered within wind farm control, while the 10° discretization provides sufficient coverage of the control space without excessively increasing the number of aeroelastic simulations.

225 The four operational modes are implemented so that their resulting response quantities can be evaluated consistently over the common 600 s analysis window. In PP mode, the turbine remains in power-producing operation throughout the complete evaluation period. In PK mode, the turbine is non-producing, with the blades pitched to the 90° feather position and the rotor left free to rotate without application of the mechanical brake. SU and SD represent a single transition between these two operating states and are implemented using the corresponding start-up and shut-down functionality of the controller.

230 Rather than evaluating only the comparatively short maneuver itself, the transition is embedded within the 600 s evaluation interval so that the resulting response metrics remain directly comparable with those of the other operational modes (Vad et al. (2026); Pettas et al. (2026)). For SD, the turbine initially operates in power-production mode for 150 s before the shut-down maneuver is initiated, after which it remains parked for the remainder of the evaluation interval. Conversely, for SU, the turbine initially remains parked, performs the start-up maneuver, and subsequently operates in power-production mode for 150 s. This

235 standardized representation retains the contribution of the transient event while avoiding the need to introduce the timing of the maneuver as an additional input dimension of the database.

Following each aeroelastic simulation, the structural-load time series over the retained 600 s interval are post-processed to obtain DELs. For load channels for which a combined response is reported, the resultant load is first calculated at each simulation time step from the corresponding orthogonal components. For two components $M_1(t)$ and $M_2(t)$, the combined

240 signal is defined as

$$M_{\text{comb}}(t) = \sqrt{M_1^2(t) + M_2^2(t)}. \quad (3)$$

The DEL is subsequently calculated from the resulting time series, rather than by combining DELs calculated independently for the individual components. Cycle counting is performed using the rainflow algorithm, and the 1 Hz equivalent DEL for each 10-min simulation is calculated

245
$$M_{\text{eq}} = \left[\frac{1}{N_{\text{eq}}} \sum_i c_i L_i^m \right]^{1/m}, \quad (4)$$

where c_i is the number of cycles with load amplitude L_i , m is the corresponding Wöhler exponent, and $N_{\text{eq}} = 600$ is the reference number of cycles, corresponding to a 1 Hz DEL over the 600 s evaluation interval (Miner (1945)). Wöhler exponents of ($m=10$) are used for the blade-root load channels and ($m=4$) for the remaining steel-dominated structural channels. No mean-load correction is applied during the fatigue calculation, and the reported quantities correspond directly to section loads

250 and moments from the aeroelastic model as they are not converted to local stresses.

In addition to the structural-load DELs, statistical descriptors of the relevant operational quantities are extracted from the retained 600 s simulation time series, including generator power, rotor speed, and blade pitch angle. The blade pitch travel



is additionally calculated as a simplified metric of pitch actuator usage. The total pitch travel over the evaluation interval is defined as:

$$255 \quad \text{ADC} = \sum_{t=t_0}^{t_{\text{end}}-1} |\theta_{t+1} - \theta_t|, \quad (5)$$

where θ_t denotes the blade pitch angle at time step (t) and ADC stands for Actuator Duty Cycle. In addition to these derived quantities, the mean, standard deviation, minimum, and maximum values of the relevant simulation-output channels are retained during post-processing.

260 The version of HAWC2 used for running the aeroelastic simulations corresponds to 13.2.12, which is an intermediate version and not a part of the official HAWC2 release yet. However, the exact executable used in this work is shared via <https://doi.org/10.5281/zenodo.22911878> to ensure full reproducibility of the results. This intermediate version was chosen because it features a new sparse Newmark solver that not only handles sparse constraint matrices but also sparse body matrices, which makes it faster than the previous solver.

270 The main input file to HAWC2 is the .htc file, a text file containing all information regarding the turbine model (structural definition, aerodynamic properties, controller settings etc.) as well as simulation-specific information (time, solver parameters, wind conditions, controller actions etc.) and the output list definition the time series of which will be retrieved. It additionally points to the rest of input files (.st files for Timoschenko beam properties for each main body, .ae file for the aerodynamic definition of the blades planform, .pc file for the profile coefficients of each airfoil, .turb files for the turbulence boxes and .dll files for the pre-compiled dynamic link libraries that will communicate with HAWC2 core during a simulation, such as the controller or the blade pitch and generator servos). One .htc file corresponds to one simulation realization.

The output from HAWC2 consists of .gtsdf files containing the time series of the output sensors defined in the .htc file, which are shown in Tables 5 to 7. Other output formats are available, but gtsdf (General Time Series Data Format) is chosen since it is the only format supported by the post-processing framework available in wetb (DTU Wind Energy), a public Python package developed by DTU Wind containing tools intended for use in the wind community, and especially HAWC2 users.

275 The post-processing of the .gtsdf files is done in steps. First time series are post-processed for each simulation, so a few different statistics are extracted for each output sensor. For doing so, the script `collect_postproc.py` defines and applies the post-processing functions to the time series (shown in Table 2). The outputs of this script are NetCDF files, one for each post-processing function and containing the values for all simulations.



Table 2. Post-processing functions applied to time series.

Name	Description
statistics	Returns min, mean, max and std of all sensors
equivalent_loads	Returns 1Hz DEL of all load sensors for a given list of Wöhler slopes
actuator_duty_cycle	Returns the ADC of the blade pitch sensors
resultant_statistics	Returns min, mean, max and std of combined sensors
resultant_equivalent_loads	Returns 1Hz DEL of combined load sensors for a given list of Wöhler slopes

Secondly the NetCDF files are manipulated by `manipulate_postproc.py`. The simulation parameters shown in Table 3 are added as auxiliar coordinates, and also some fixtures are made (conversion to right units and selection of the outputs of interest).

Table 3. Simulation parameters to be added as auxiliar coordinates.

Name	Description
dlc	Operational mode (PP / SU / SD / PK) [-]
pmode	Controller mode (normal / down-regulation) [-]
pset	Normalized power setpoint [-]
yaw	Yaw misalignment [deg]
wsp	Ambient wind speed [m/s]
ti	Ambient turbulence intensity [-]
x_spacing	Wake-generating turbine streamwise location [D]
y_spacing	Wake-generating turbine lateral location [D]
seed	Turbulence box random number [-]

Independently, the ambient wind speed and turbulence intensity of every inflow file are averaged across each of the four sectors (left, up, right, down) in which the rotor is divided. This can be done by `extract_saiqs.py` and the `.wnd` files, but nevertheless the NetCDF file with these values is also shared.

Finally `create_database.py` creates the databases. It first assembles the different post-processings into a single dimension "statistic" (choosing the right Wöhler slope for the DEL of each load sensor) and adds the previously extracted sector-averaged ambient wind speed and turbulence intensity to their corresponding simulations. Then the statistics across seeds are performed; simulations are grouped into bins with the exact same parameters except for seed, then the statistics shown in Table 4 are computed across seeds. The outputs of this script are one NetCDF file for each statistic. The median p50 is what will be used for training the surrogate models.



Table 4. Statistics applied across seeds.

Name	Description
min	Minimum
mean	Mean
max	Maximum
std	Standard deviation
p25	Percentile 25
p50	Percentile 50
p75	Percentile 75
p90	Percentile 90

In addition to the aforementioned simulation parameters, Tables 5 to 7 summarize the output channels available in each NetCDF file. The user-selected wind points 1–9 correspond to distinct spatial locations at which the three wind-velocity components were evaluated. Their exact coordinates are specified in the accompanying HAWC2 input files. Similarly, tower locations 1–3 used for the local tower load and acceleration channels correspond to 25%, 50%, and 75% of the tower height, respectively.

Table 5: Available operational, inflow, and aerodynamic target channels.

Channel description
Operational variables
Blade pitch angle (position) (for blade 1, 2 & 3)
Rotor azimuth angle (position)
Rotor azimuth angular speed
Electrical generator torque
Electrical generator power
Inflow variables
X, Y, Z components of wind at user selected wind points 1–9
Rotor aerodynamic quantities
Total rotor aerodynamic load (force in x direction)
Total rotor aerodynamic load (moment in x direction)
Rotor aerodynamic power



Table 6: Available drivetrain and tower-response channels.

Channel description
Low-speed shaft loads
Low-speed shaft thrust force (this is constant along the shaft and is equivalent to the rotor thrust force)
Rotating low-speed shaft shear force in the y_a direction (this is constant along the shaft)
Rotating low-speed shaft shear force in the z_a direction (this is constant along the shaft)
Nonrotating low-speed shaft shear force in the y_s direction (this is constant along the shaft)
Nonrotating low-speed shaft shear force in the z_s direction (this is constant along the shaft)
Low-speed shaft torque (this is constant along the shaft and is equivalent to the rotor torque)
Rotating low-speed shaft bending moment about the y_a axis at the shaft tip
Rotating low-speed shaft bending moment about the z_a axis at the shaft tip
Nonrotating low-speed shaft bending moment about the y_s axis at the shaft tip
Nonrotating low-speed shaft bending moment about the z_s axis at the shaft tip
Combined nonrotating low-speed shaft bending moment at the shaft tip (projection in the y_s – z_s plane)
Tower-top loads
Tower-top / yaw bearing yaw moment
Nonrotating tower-top / yaw bearing roll moment
Nonrotating tower-top / yaw bearing pitch moment
Combined tower-top fore-aft and side-to-side bending moment
Tower-base loads
Tower base fore-aft shear force
Tower base side-to-side shear force
Tower base roll (or side-to-side) moment (i.e., the moment caused by side-to-side forces)
Tower base pitching (or fore-aft) moment (i.e., the moment caused by fore-aft forces)
Tower base yaw (or torsional) moment
Combined tower-base fore-aft and side-to-side bending moment
Local tower loads
Local tower roll (or side-to-side) moment (of tower loc 1, 2 & 3)
Local tower pitching (or fore-aft) moment (of tower loc 1, 2 & 3)
Local tower yaw (or torsional) moment (of tower loc 1, 2 & 3)
Tower accelerations

Continued on next page



Table 6 – continued from previous page

Channel description
Tower-top / yaw bearing fore-aft (translational) acceleration (absolute)
Tower-top / yaw bearing side-to-side (translational) acceleration (absolute)
Local tower fore-aft (translational) acceleration (absolute) (of tower loc 1, 2 & 3)
Local tower side-to-side (translational) acceleration (absolute) (of tower loc 1, 2 & 3)

Table 7: Available blade-root load channels.

Channel description
Blade-root loads
Out-of-plane shear force at the blade root (for blade 1, 2 & 3)
In-plane shear force at the blade root (for blade 1, 2 & 3)
Flapwise shear force at the blade root (for blade 1, 2 & 3)
Edgewise shear force at the blade root (for blade 1, 2 & 3)
In-plane moment (i.e., the moment caused by in-plane forces) at the blade root (for blade 1, 2 & 3)
Out-of-plane moment (i.e., the moment caused by out-of-plane forces) at the blade root (for blade 1, 2 & 3)
Pitching moment at the blade root (for blade 1, 2 & 3)
Edgewise moment (i.e., the moment caused by edgewise forces) at the blade root (for blade 1, 2 & 3)
Flapwise moment (i.e., the moment caused by flapwise forces) at the blade root (for blade 1, 2 & 3)
Combined flapwise and edgewise moment at the blade root (for blade 1, 2 & 3)
Combined in-plane and out-of-plane moment at the blade root (for blade 1, 2 & 3)

3 Surrogate Modeling Methodology

300 This section describes the surrogate-model development framework used to transform the aeroelastic simulation database presented in Section 2 into predictive machine-learning models. The objective of the framework is to provide a transparent, reproducible, and extensible workflow that can serve as a benchmark for developing and evaluating surrogate models using simulation-based wind turbine datasets. The framework is available as a frozen code release in the Zenodo repository accompanying this dataset (Tsaklis et al. (2026b)). Throughout this section, the individual processing steps are described together with the corresponding scripts in the repository to facilitate reproducibility and enable users to adapt the workflow to other datasets or surrogate model architectures. The process of surrogate model development is illustrated in Fig. 2. To facilitate the use of



the framework, a technical code workflow illustrating the execution sequence of the repository scripts is provided in Fig. 3.

305 The channels that are selected for the data-driven model training are presented in Table 8.

Table 8. Input and target channels used for surrogate-model training.

Input channels	Target channels				
	Operational variables	Blade loads	Main shaft loads	Tower top loads	Tower base loads
SAWS right	Generator power	Blade root edgewise moment	Low-speed shaft torque	Yaw bearing roll moment	Tower base side-side moment
SAWS left	Rotor speed	Blade root flapwise moment	Low-speed shaft y bending moment	Yaw bearing pitch moment	Tower base fore-aft moment
SAWS up	Pitch angle ADC	Blade root pitching moment	Low-speed shaft z bending moment	Yaw bearing yaw moment	Tower base torsional moment
SAWS down		Blade root in-plane moment	Shaft combined bending moment	Tower top combined moment	Tower base combined moment
SATI right		Blade root out-of-plane moment			
SATI left		Blade root combined moment			
SATI up		Blade bearing combined moment			
SATI down					
Power setpoint					
Yaw angle					

SAWS and SATI denote sector-averaged wind speed and sector-averaged turbulence intensity, respectively. The listed load target channels correspond to damage equivalent loads derived from the aeroelastic simulations.

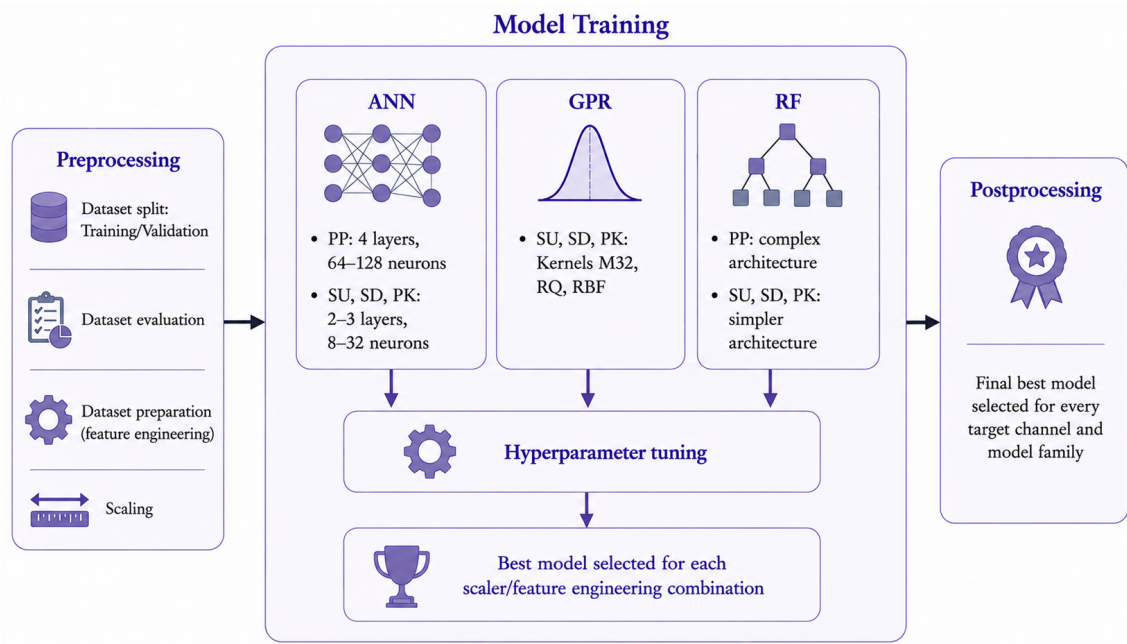


Figure 2. Detailed workflow of the surrogate model development.

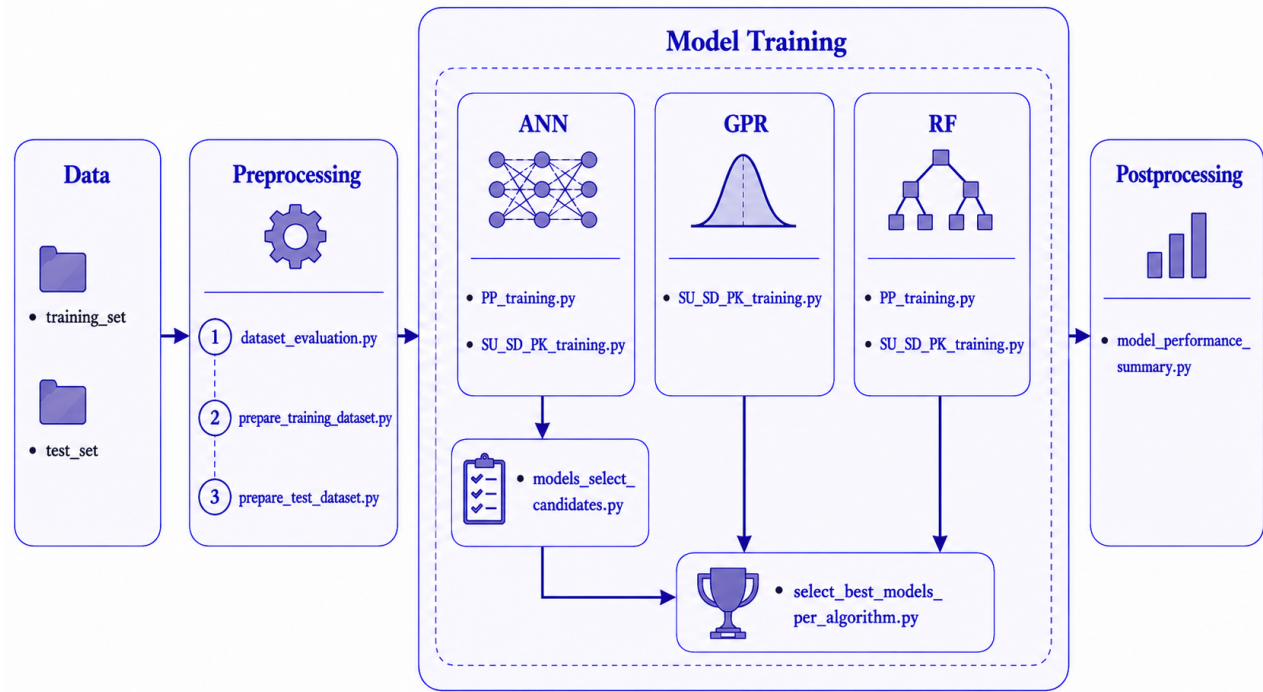


Figure 3. Code workflow.

3.1 Preprocessing

Following the completion of the aeroelastic simulations described in Section 2, the generated datasets require preprocessing before they can be used for surrogate model development. The preprocessing workflow is implemented through three Python scripts located in the `Preprocessing` directory of the accompanying GitLab repository: `dataset_evaluation.py`,
 310 `prepare_training_dataset.py`, and `prepare_test_dataset.py`. Together, these scripts evaluate the simulation output database and generate the inputs required for the subsequent training and evaluation stage.

As discussed in Section 2, each simulation was performed using multiple inflow realizations generated from different random seeds. The resulting NetCDF files therefore contain statistical summaries across all realizations. The surrogate models are trained using the MISO (Multiple-Input Single-Output) approach, that is, a single surrogate model is developed for every target
 315 channel presented in Table 8. For the surrogate model development in this work, the median (p_{50}) response is selected. This choice reflects the objective of training models that represent the typical turbine response under a given operating condition rather than extreme realizations. Furthermore, the median is less sensitive to outliers than the arithmetic mean, making it a more robust representation of the underlying simulation response.

The script `dataset_evaluation.py` provides an initial assessment of the generated datasets. It imports the training
 320 and test NetCDF files for each operational mode and produces histograms of the input and target variables, together with projections of the input space. These visualizations facilitate comparison of the training and test distributions and help verify



that the test samples remain within the domain covered by the training data. This assessment is important because samples outside the training domain require the surrogate models to extrapolate, which generally reduces predictive accuracy because predictions are made beyond the models' validity region. Figs. 4 and 5 present representative distribution plots for two target channels under the PP and SU operational modes, respectively. In all cases, the test samples lie within the ranges covered by the training and validation datasets.

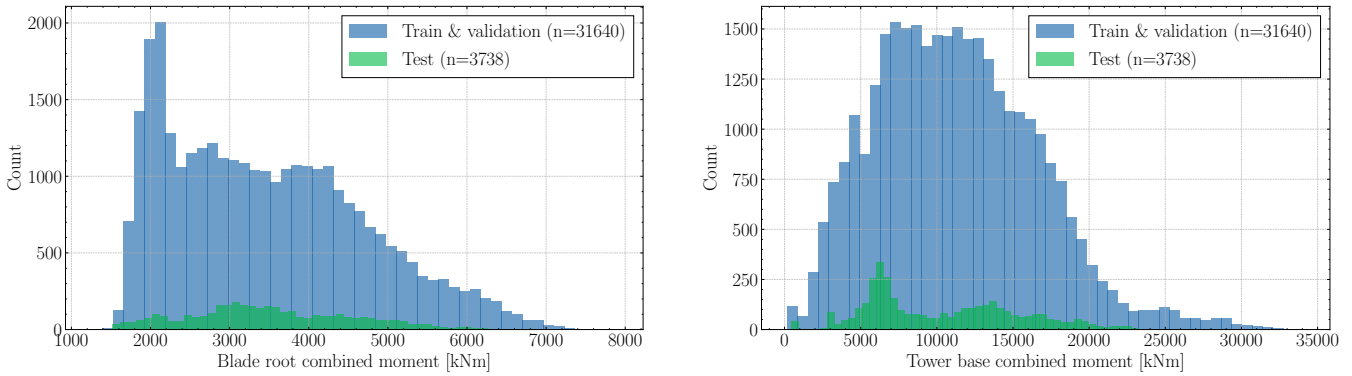


Figure 4. Distributions of the training/validation, and test samples for two representative target channels under the PP operational mode: blade-root combined-moment DEL (left) and tower-base combined-moment DEL (right). Here, n denotes the number of samples.

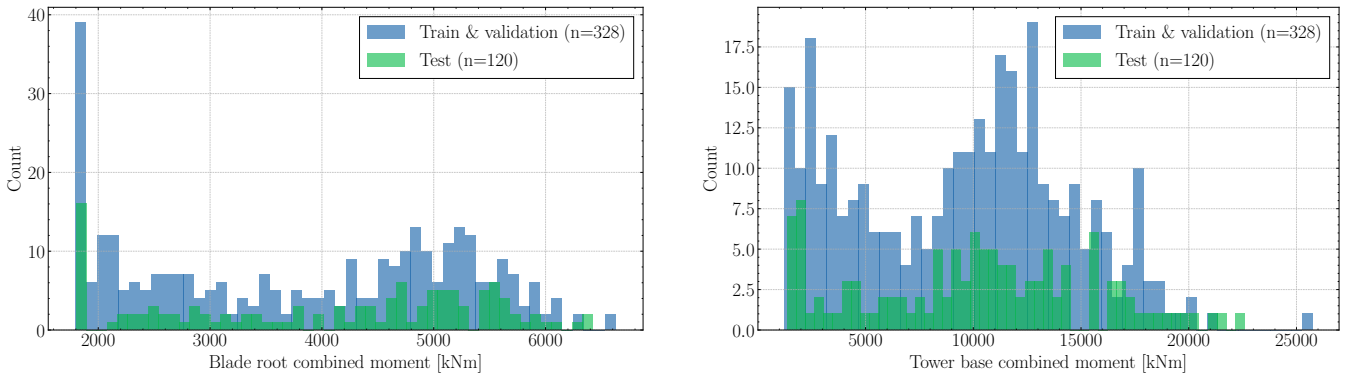


Figure 5. Distributions of the training/validation, and test samples for two representative target channels under the SU operational mode: blade-root combined-moment DEL (left) and tower-base combined-moment DEL (right). Here, n denotes the number of samples.

The preparation of the training dataset is performed by `prepare_training_dataset.py`. The script imports the NetCDF files, extracts the selected input variables and target channels, and creates independent datasets for each predicted output quantity. The data are then randomly divided into training and validation subsets using an 80/20 % split with a fixed random seed to ensure reproducibility.

Two alternative feature engineering strategies are supported. The first, referred to as `no_deltas`, uses the original sector-averaged wind speed and turbulence intensity values together with the available control variables i.e. power & yaw setpoint.



The second strategy, `deltas`, augments the original feature set with the left-right differences in sector-averaged wind speed and turbulence intensity. These additional variables enable an assessment of whether this additional information, regarding
335 asymmetrical inflow conditions, improves the model performance.

The preprocessing script also removes control variables that are not relevant for specific operational modes, such as yaw misalignment for start-up and shut-down, and both yaw misalignment and power setpoint for the parked mode.

The framework further supports three alternative scaling strategies. Standard scaling transforms each feature to have zero mean and unit variance according to

$$340 \quad x_{scaled} = \frac{x - \mu}{\sigma} \quad (6)$$

where μ and σ denote the sample mean and standard deviation, respectively. Min-Max scaling maps each feature to a fixed interval, here $[0, 1]$, according to

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (7)$$

where x_{min} and x_{max} are the minimum and maximum values of the corresponding feature. Both transformations are im-
345 plemented using the `StandardScaler` and `MinMaxScaler` classes available in the `scikit-learn` library (Pedregosa et al. (2011)). In addition, the framework provides an unscaled dataset, which is used for the RF models since tree-based methods are generally insensitive to feature scaling. The fitted input and output scalers are stored and reused during model evaluation to ensure that identical transformations are applied to the test data.

Finally, `prepare_test_dataset.py` follows the same feature extraction and feature engineering procedure as the
350 training data preparation but does not perform a train-validation split or apply feature scaling. Instead, the prepared test datasets are stored in their original form, while any required scaling is applied during model inference using the scalers obtained from the training data. This ensures a consistent preprocessing pipeline while preserving the independence of the test dataset throughout model development.

3.2 Training

355 The training stage of the framework is organized into three independent model families: ANNs, GPR, and RF. Each model family is implemented in a dedicated directory of the accompanying GitLab repository, which contains the corresponding training scripts and model-specific utilities. These model families were selected because of their widespread adoption in research and industrial applications and their complementary characteristics. ANNs offer flexible and scalable approximations of complex nonlinear relationships, particularly for large datasets, while providing efficient inference once trained. GPR is well
360 suited to smaller datasets and provides accurate predictions using comparatively simple model configurations, whereas RFs can achieve robust predictive performance at relatively low computational cost. Their fundamentally different learning principles enable users to select a model according to the dataset size, computational resources, and application requirements, rather than presenting any single algorithm as universally optimal.



Different training strategies are adopted for the PP operational mode and the remaining operational modes (SU, SD, and PK).
365 The primary motivation for this separation is the considerable difference in dataset size. The power production dataset contains approximately 300,000 datapoints for each target channel, whereas the remaining operational modes contain between approximately 1000 and 5000 datapoints. Consequently, different network architectures and hyperparameter optimization strategies are employed to account for the different data regimes.

For the PP operational mode, surrogate models are trained for all 22 target channels, including both operational variables
370 and structural loads. In contrast, only the 19 structural load channels are considered for the SU, SD, and PK operational modes, since operational variables are either constant or not physically meaningful during these transient or parked conditions. Similarly, the two feature engineering strategies introduced in Section 3.1 are evaluated only for the PP operational mode. Since the remaining operational modes consist exclusively of uniform inflow simulations, the left-right wind speed and turbulence intensity differences are negligible and therefore do not provide additional information. Both scaling strategies are evaluated
375 for ANN and GPR models in all operational modes, whereas RF models are trained exclusively on the unscaled datasets. Following hyperparameter optimization, the selected models are stored in the corresponding `models` directory while their training metadata and performance summaries are stored in the corresponding `results` directory.

3.2.1 Artificial Neural Networks

Artificial Neural Networks constitute the most flexible model family included in the framework. Their ability to approximate
380 highly nonlinear relationships makes them particularly suitable for predicting wind turbine operational variables and structural loads. However, the optimal network complexity depends strongly on the amount of available training data. Consequently, two different ANN implementations are adopted within the framework.

For the PP operational mode, the models are implemented using the TensorFlow/Keras framework (Martin et al. (2015)) and optimized using the Optuna hyperparameter optimization library (Akiba et al. (2019)). Hyperparameter tuning is performed
385 using the Tree-structured Parzen Estimator (Bergstra et al. (2011)), a Bayesian optimization algorithm that efficiently explores the search space by learning probabilistic models of promising parameter combinations.

Each Optuna trial defines a complete network architecture consisting of four fully connected hidden layers, the number of neurons in each layer, the activation function, and the learning rate. The candidate models are evaluated using five-fold cross validation on the training dataset with the cross-validated Mean Squared Error (MSE) serving as the optimization objective.
390 After identifying the best hyperparameter configurations, the selected networks are retrained using the original training dataset while monitoring the fixed validation dataset created during preprocessing. Early stopping with restoration of the best weights is employed to prevent overfitting, while the learning rate is automatically reduced when the validation loss reaches a plateau.

Rather than retaining only the single best Optuna trial, the framework stores the five highest-performing unique hyperparameter configurations. This design choice reduces the dependence of the final benchmark on a single optimization run and
395 allows the candidate models to be compared during the subsequent postprocessing stage before selecting the final benchmark model for each target channel.



For the SU, SD, and PK operational modes, a different implementation is adopted. Instead of TensorFlow/Keras, the models are developed using the Skorch library (Tietz et al. (2017)), which provides a scikit-learn compatible wrapper around PyTorch neural networks. Although both implementations are based on feed-forward multilayer perceptrons, the Skorch implementation exhibits considerably lower computational overhead for the relatively small datasets associated with these operational modes.

The reduced dataset size also motivates a simpler network architecture. During hyperparameter optimization, both two- and three-layer networks are considered together with smaller hidden layer sizes. Similar to the PP workflow, Optuna is used to identify the optimal hyperparameter configuration using five-fold cross validation. Following model selection, the final networks are retrained using the combined training and validation datasets in order to maximize the amount of information available for parameter estimation before evaluation on the independent test dataset.

The two different grids for hyperparameter search are presented in Table 9.

Table 9. Hyperparameter search space for the ANN models. Different search spaces are adopted for the power production operational mode and the remaining operational modes to account for the substantial difference in dataset size.

Hyperparameter	Power production	Start-up / Shut-down / Parked
Hidden layers	4	2–3
Neurons per layer	64–128	8–32
Activation function	GELU, LeakyReLU	GELU, Swish, LeakyReLU
Learning rate	0.005, 0.0075, 0.01	0.005, 0.0075, 0.01
Batch size	512	256
Maximum epochs	1000	500
Optuna trials	60	70
Cross-validation	5-fold	5-fold

3.2.2 Gaussian Process Regression

Gaussian Process Regression is included as a probabilistic baseline capable of providing highly accurate predictions together with an estimate of predictive uncertainty. However, the computational complexity of GPR scales cubically with the number of training samples, making the method impractical for the approximately 300,000 datapoints contained in the PP operational mode training dataset. Consequently, GPR models are trained only for the SU, SD, and PK operational modes, where the datasets remain sufficiently small for exact Gaussian process training and inference.

The implementation is based on the `GaussianProcessRegressor` (Rasmussen and Williams (2005)) class provided by the `scikit-learn` library. Three covariance functions are considered during model selection: a Matérn 3/2 kernel with an additional White kernel, a Rational Quadratic kernel, and a Radial Basis Function kernel. For each kernel, the optimizer is restarted five times in order to reduce the likelihood of converging to local optima during marginal likelihood optimization.



Model selection is performed using the fixed training and validation datasets generated during preprocessing. Each kernel is trained on the training subset and evaluated on the validation subset using the MSE. The kernel yielding the lowest validation error is selected as the final model configuration. To maximize the information extracted from the limited amount of available data, the selected kernel is subsequently retrained using the combined training and validation datasets before being stored in the repository. This procedure preserves an unbiased model selection process while allowing the final surrogate model to exploit the complete set of available observations.

3.2.3 Random Forest Regressor

Random Forests provide an ensemble-based regression approach that is robust to nonlinear feature interactions while requiring comparatively limited hyperparameter tuning. The framework employs the `RandomForestRegressor` (Breiman (2001); Geurts et al. (2006)) implementation available in `scikit-learn`. Since decision tree ensembles are largely insensitive to feature scaling, RF models are trained exclusively using the unscaled datasets prepared during preprocessing.

Hyperparameter optimization is performed through an exhaustive grid search using the `ParameterGrid` utility of `scikit-learn`. For the PP operational mode, the larger amount of available training data and the model robustness enable the use of a fixed training-validation split for model selection. Each candidate hyperparameter combination is trained on the training subset and evaluated on the validation subset using the MSE. Once the optimal configuration has been identified, the final RF model is retrained using the combined training and validation datasets before being stored in the repository.

For the SU, SD, and PK operational modes, the considerably smaller datasets motivate a different validation strategy. Instead of relying on a single validation subset, five-fold cross validation is employed during the grid search in order to obtain a more robust estimate of model performance while maximizing the effective use of the available data. The optimal hyperparameter combination is then retrained using the complete combined training and validation dataset. Compared with the PP operational mode, the corresponding hyperparameter grid is also reduced to reflect the lower model complexity required for the smaller datasets. The two different grids are presented in Table 10.

The RF hyperparameters were selected to balance model flexibility, regularization, and computational efficiency while maintaining a limited search space. The number of trees (`n_estimators`) was fixed at 300 to provide stable predictions, as further increases were expected to offer diminishing benefits at additional computational cost. The remaining hyperparameters were tuned to control tree complexity, introduce diversity among trees, and regulate the amount of training information used during tree construction. Together, these choices enabled the models to capture complex relationships in the data while reducing the risk of overfitting.



Table 10. Hyperparameter search spaces for the RF models. Different search spaces and validation strategies are adopted for the power production operational mode and the remaining operational modes to account for the substantial difference in dataset size.

Hyperparameter	Power production	Start-up / Shut-down / Parked
Number of trees	300, 600	300
Minimum samples per split	2	2
Maximum depth	12, 24	None, 6
Minimum samples per leaf	1, 20	1, 4
Maximum features	0.5, 1.0	0.5, 1.0
Maximum samples	None, 0.2	None, 0.8
Validation strategy	Fixed validation set	5-fold cross-validation

445 3.3 Postprocessing

The postprocessing stage completes the data-driven model development framework by identifying the best-performing models obtained during the training stage and organizing them into a benchmark model library. The corresponding scripts are located in the `Postprocessing` directory and consist of `models_select_candidates.py` and `select_best_models_per_algorithm.py`.

450 Within the postprocessing step, the script `models_select_candidates.py` is applied exclusively to the ANN model family. As described in Section 3.2.1, the ANN training workflow retains the five highest-performing candidate architectures identified during the Optuna optimization for every combination of operational mode, target channel, feature engineering strategy, and scaling method. These candidate models are subsequently evaluated on the independent test dataset using the coefficient of determination (R^2), the Mean Absolute Error (MAE), and the Root Mean Squared Error (RMSE). Rather than
 455 selecting the model based on a single performance metric, the framework adopts a majority-voting strategy in which the candidate achieving the best performance in at least two of the three metrics is selected as the final representative of the corresponding preprocessing configuration. The selected models are stored in the `models_candidates` directory.

The final stage is performed by `select_best_models_per_algorithm.py`, which is applied independently to each model family. The script imports all trained models corresponding to the different preprocessing configurations, evaluates them
 460 on the same set of performance metrics (R^2 , MAE, and RMSE), and identifies the best-performing surrogate model for each operational mode and target channel. The selected models are copied to the `models_best` directory together with their associated performance summaries.

Following the execution of these two scripts, the surrogate-model development framework is complete. The resulting benchmark library contains one representative model from each model family for every operational mode and target channel, selected
 465 from all combinations of preprocessing strategies investigated during model development. This curated collection forms the basis for the comparative analysis presented in Section 4 and provides a standardized set of surrogate models that can be directly used in future studies.



In addition to the model selection workflow, the postprocessing stage provides utilities for the systematic analysis and visualization of the resulting surrogate models. The script `model_performance_summary.py` imports the final benchmark models from all three model families, evaluates them on the independent test dataset, and generates a comprehensive summary of their predictive performance. The resulting performance metrics are exported in both CSV and PDF formats, while additional summary figures are produced to visualize the predicted target channels as a function of binned wind speed for the test dataset. These outputs facilitate the comparative assessment of the different surrogate model families and form the basis for the performance evaluation presented in Section 4.

3.4 Surrogate model performance evaluation metrics

Throughout this study, the prediction error is defined as $e_i = y_{\text{pred},i} - y_{\text{actual},i}$. Accordingly, positive errors indicate overprediction, whereas negative errors indicate underprediction. The mean error therefore quantifies the average prediction bias, with positive and negative values corresponding to systematic overprediction and underprediction, respectively. The standard deviation of the error characterizes the dispersion of the prediction errors around their mean. The normalized root mean squared error (NRMSE) is expressed as a percentage and calculated by normalizing the RMSE by the range of the actual values in the test dataset:

$$\text{NRMSE} = \frac{\text{RMSE}}{y_{\text{actual,max}} - y_{\text{actual,min}}} \times 100\%. \quad (8)$$

The remaining performance metrics, namely R^2 , MAE, and RMSE, follow their standard definitions.

To complement the quantitative evaluation, the script `surrogate_aeroelastic_response_summary.py` provides a qualitative analysis of the trained surrogate models. Rather than relying on the test dataset, the script generates user-defined input combinations on the fly and evaluates the corresponding surrogate predictions over the selected input space. The resulting response surfaces, which are exported in PDF format, enable the aeroelastic behaviour learned by the surrogate models to be visualized and facilitate the identification of trends, nonlinearities, or unexpected model behaviour in different regions of the operating space.

4 Results

This section presents the results of the model training described in Section 3.2 and demonstrates complementary approaches for evaluating surrogate-model performance. The aim is not to provide an extensive physical interpretation of the individual model predictions, but to illustrate why multiple evaluation methods should be considered when assessing data-driven models. Performance metrics alone may provide an incomplete representation of model quality, as their values depend strongly on the composition and distribution of the test dataset. They should therefore be complemented by visualizations that reveal model behaviour across different regions of the input and output spaces, as well as potential systematic biases. The evaluation presented here includes performance metrics calculated using the test set defined in Section 2, error-distribution plots, and comparisons across model families and target groups. In addition, surrogate aeroelastic response plots are examined using



datasets generated specifically for this purpose. These datasets are created on the fly by sweeping one input parameter across a prescribed range while holding all other input variables fixed, and are therefore independent of the test set. Representative load channels are selected for these visualizations, while detailed performance tables covering all target channels and operational modes are provided in the Appendix for completeness.

4.1 Performance metric plots

The first set of figures is derived from the performance-metric summary CSV file generated using the `model_performance_summary`. Fig. 6 provides an overview of model performance across the considered operational modes and target channels in terms of the coefficient of determination, R^2 (left), and the NRMSE (right). Each point represents the performance obtained for an individual target channel, while the horizontal black lines indicate the median performance of each model family. The width of each violin represents the estimated density of the corresponding metric values and therefore illustrates how performance is distributed across the target channels. This representation complements the median values by showing the variability in performance within each combination of model family and operational mode.

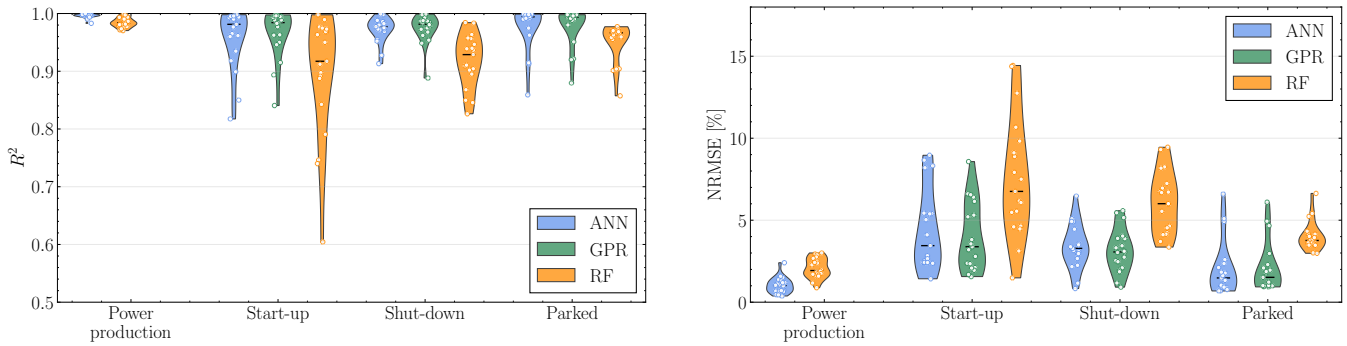


Figure 6. Distribution of R^2 and NRMSE values across target channels within the test dataset for the ANN, GPR, and RF models under the four operational modes. Individual points represent target channels, while the horizontal black lines indicate the median values for each model family.

Overall, the ANN and GPR models exhibit broadly comparable metric distributions, with high median R^2 values and relatively low median NRMSE values across the operational modes for which they were trained. By contrast, the RF models generally exhibit lower R^2 values and higher NRMSE values, particularly for the transient start-up and shut-down cases. The spread of the distributions further demonstrates that model performance is target-dependent: a high median value does not imply equally accurate predictions for every response channel.

The comparatively weaker performance of the RF models should be interpreted in relation to both the properties of the algorithm and the design of the dataset split. RFs partition the input space into a collection of regions and produce piecewise-constant predictions based on the training observations assigned to the terminal leaves of the individual trees. Consequently, they do not provide smooth interpolation between discrete training points in the same manner as the ANN and GPR models. As



described in Section 2.2, the test set was deliberately constructed to include yaw angles and power setpoints located between the discrete values represented in the training and validation sets. This split therefore provides a direct assessment of the models' ability to interpolate across these input dimensions and is particularly challenging for the RF models.

The inclusion of RF models nevertheless serves an important benchmarking purpose by illustrating the interaction between the selected model family, the sampling strategy, and the evaluation dataset. Their performance should therefore not be interpreted as a general indication that RFs are unsuitable for surrogate modelling. For a dataset with denser and more uniformly distributed input samples, RF models may provide a useful baseline because of their straightforward implementation, comparatively low training cost, and limited preprocessing requirements. As the presented framework is reproducible and modular, the same evaluation procedure can be applied to alternative dataset designs to examine how these characteristics affect the relative performance of the model families.

Fig. 7 provides a useful visualization of systematic model bias by decomposing the RMSE into its mean-error and error-spread components. Points close to zero on the x-axis exhibit little systematic bias, while positive and negative values indicate consistent overprediction and underprediction, respectively, based on the adopted error definition. In this case, most of the SD models seem to overpredict several target channels. This distinction is particularly important for subsequent applications such as fatigue-damage accumulation and remaining useful life estimation, where systematic underprediction may result in unsafe lifetime estimates, while mild overprediction is generally acceptable because it provides a conservative assessment.

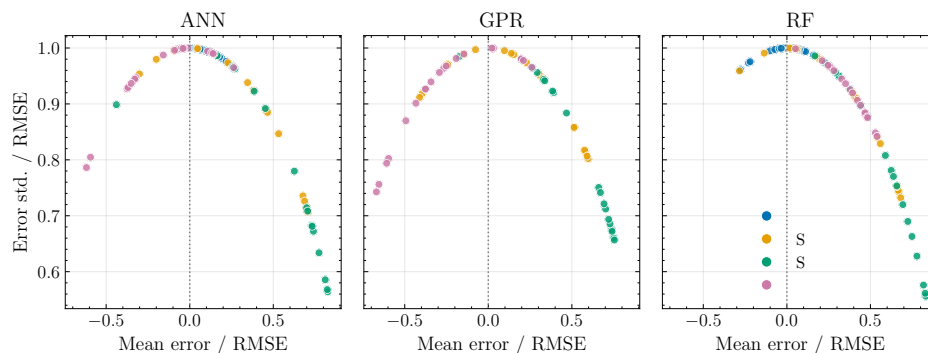


Figure 7. Normalized mean error and error standard deviation for each target channel within the test dataset and model family, coloured by operational mode. The horizontal axis represents the mean error normalized by the RMSE, while the vertical axis represents the standard deviation of the errors normalized by the RMSE. Positive and negative mean errors indicate systematic overprediction and underprediction, respectively.

Fig. 8 provides a direct comparison between the ANN and GPR models for each target channel using the test dataset. Points above the diagonal indicate better ANN performance, whereas points below it indicate better GPR performance. The operational modes are colour-coded, while the target load groups are distinctly marked. This representation helps identify whether the relative performance of the two model families is associated with a particular operational mode or type of structural load.

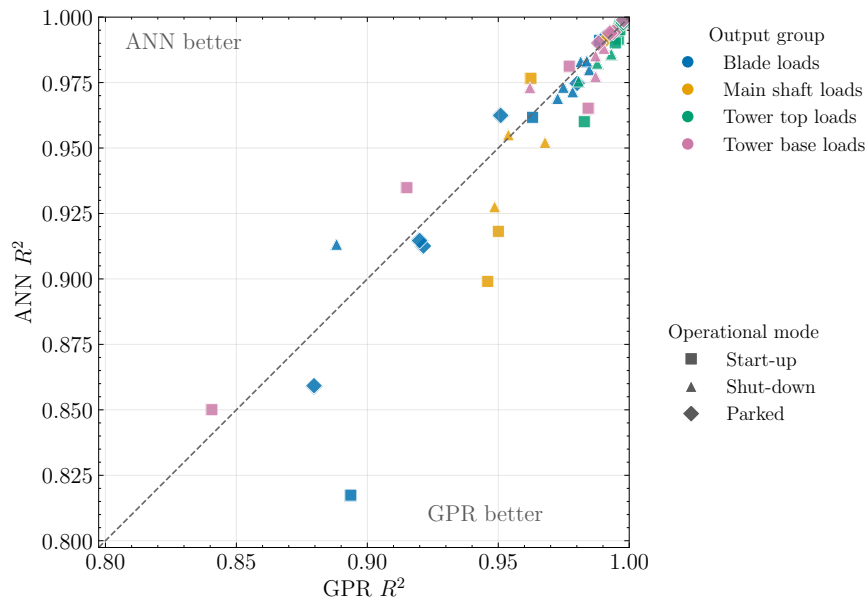


Figure 8. Comparison of ANN and GPR R^2 values for individual target channels within the test dataset, colour-coded by operational mode and distinctly marker by target load group.

Similar to the violin plots presented in Fig. 6, Fig. 9 compares the performance of the three model families, but from a complementary perspective. While the violin plots summarize performance by operational mode using median values and the overall metric distributions, this figure groups the results by target type. Specifically, it presents the mean R^2 and NRMSE values for each target group, calculated across the corresponding channels and operational modes. This comparison helps reveal whether model performance depends on the type of predicted response, such as operational variables or particular load components. The whiskers represent one standard deviation above and below the mean and therefore indicate the variability in performance within each target group. Larger whiskers identify groups for which performance differs more substantially across channels and operational modes.

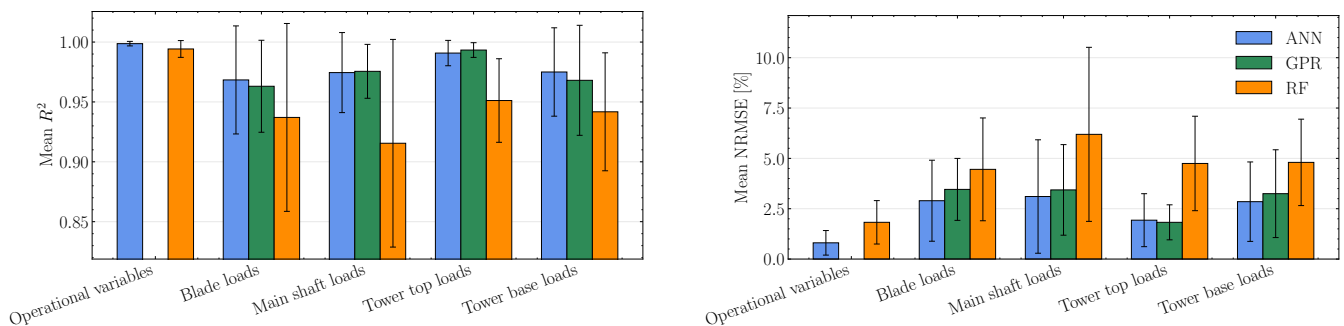


Figure 9. Mean R^2 (left) and NRMSE (right) of the ANN, GPR, and RF models, grouped by target type. Whiskers represent one standard deviation above and below the mean across the evaluated target channels and operational modes.



4.2 Training time

Table 11 summarizes the approximate time required to train a single model from each model family for one target channel, using a single scaling method and feature-engineering strategy. For consistency, the reported times correspond to the same target channel across all model families. The ANN models exhibit the longest training times, primarily because of their more computationally demanding hyperparameter-optimization procedure. This difference is most pronounced for the power-production mode, for which ANN training requires approximately two orders of magnitude more time than the corresponding RF model. Training time may therefore constitute an additional consideration when selecting a suitable surrogate-model family, alongside predictive performance and other model characteristics.

The reported values should, however, be interpreted as indicative estimates for the specific computational setup and model configurations employed in this study, rather than as generally applicable benchmarks. Absolute training times may vary substantially depending on the available computational resources, model architecture, dataset size, and hyperparameter-optimization strategy. Consequently, the relative computational cost of the different model families may also change under alternative implementations and training configurations.

Table 11. Approximate training times for the different surrogate models and operational modes in a standard laptop.

	ANN		GPR		RF	
	PP	SU, SD, PK	PP	SU, SD, PK	PP	SU, SD, PK
Training time	60 min	20–60 s	–	2–40 s	2.5 min	30 s

4.3 Test dataset prediction-based plots

The prediction-based evaluation is presented for four representative load channels: the blade-root combined moment, shaft combined bending moment, tower-top combined moment, and tower-base combined moment. Figs. 10 and 11 show the corresponding percentage-error distributions for the four operational modes (rows) and model families (colour-coded curves). The horizontal-axis limits are kept consistent within each column, enabling comparison of the same target channel across operational modes. A well-performing model should produce a narrow error distribution centred around the vertical zero-error line. A shift towards positive or negative errors indicates overprediction or underprediction, respectively, while a wider distribution indicates greater variability in the prediction errors.

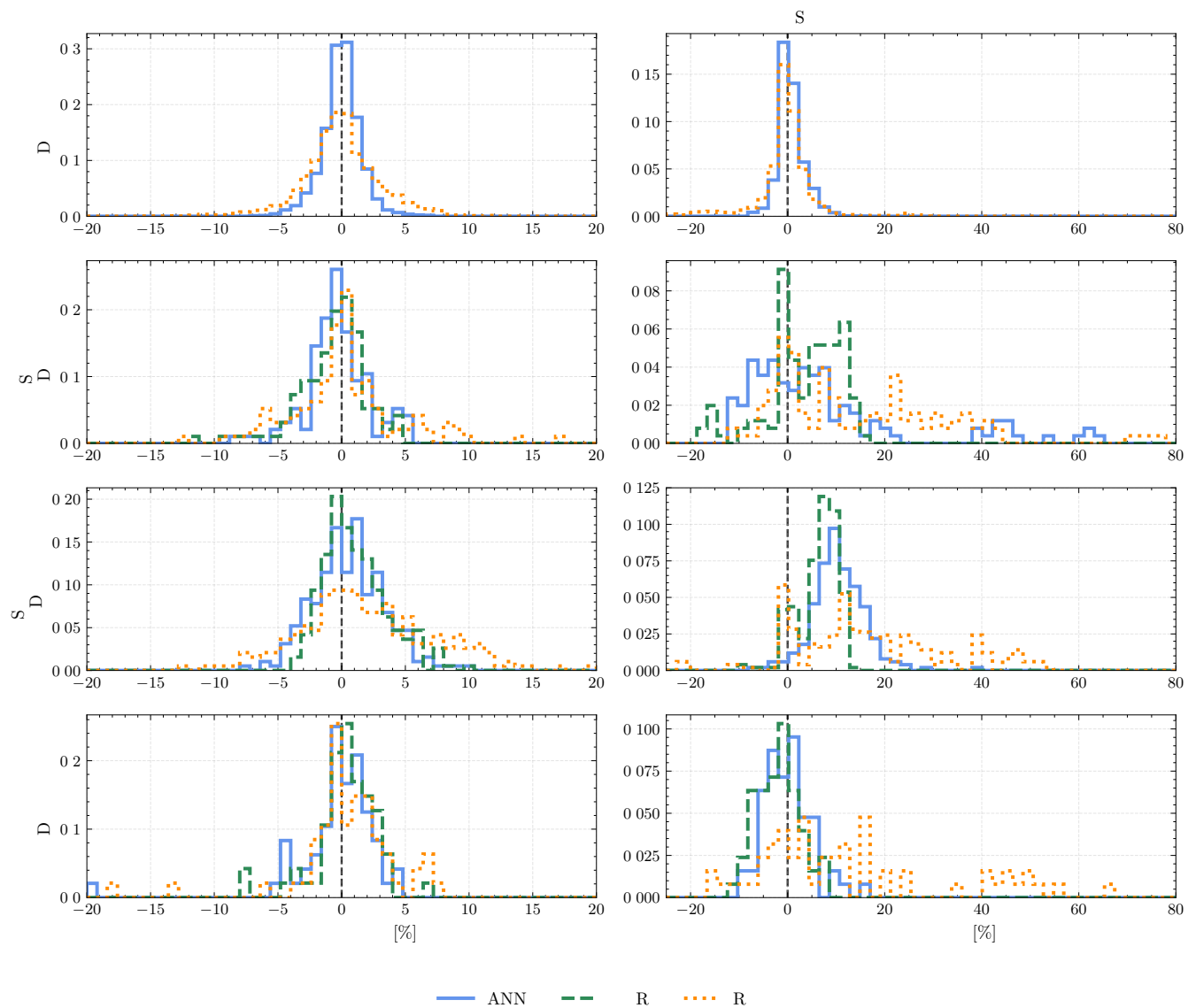


Figure 10. Percentage-error distributions of the ANN, GPR, and RF predictions for the blade root combined and shaft combined bending moments within the test dataset. Columns correspond to target channels and rows to operational modes. The vertical black dashed line indicates zero prediction error, and the horizontal-axis limits are consistent within each column. Curves are normalized to represent probability density.

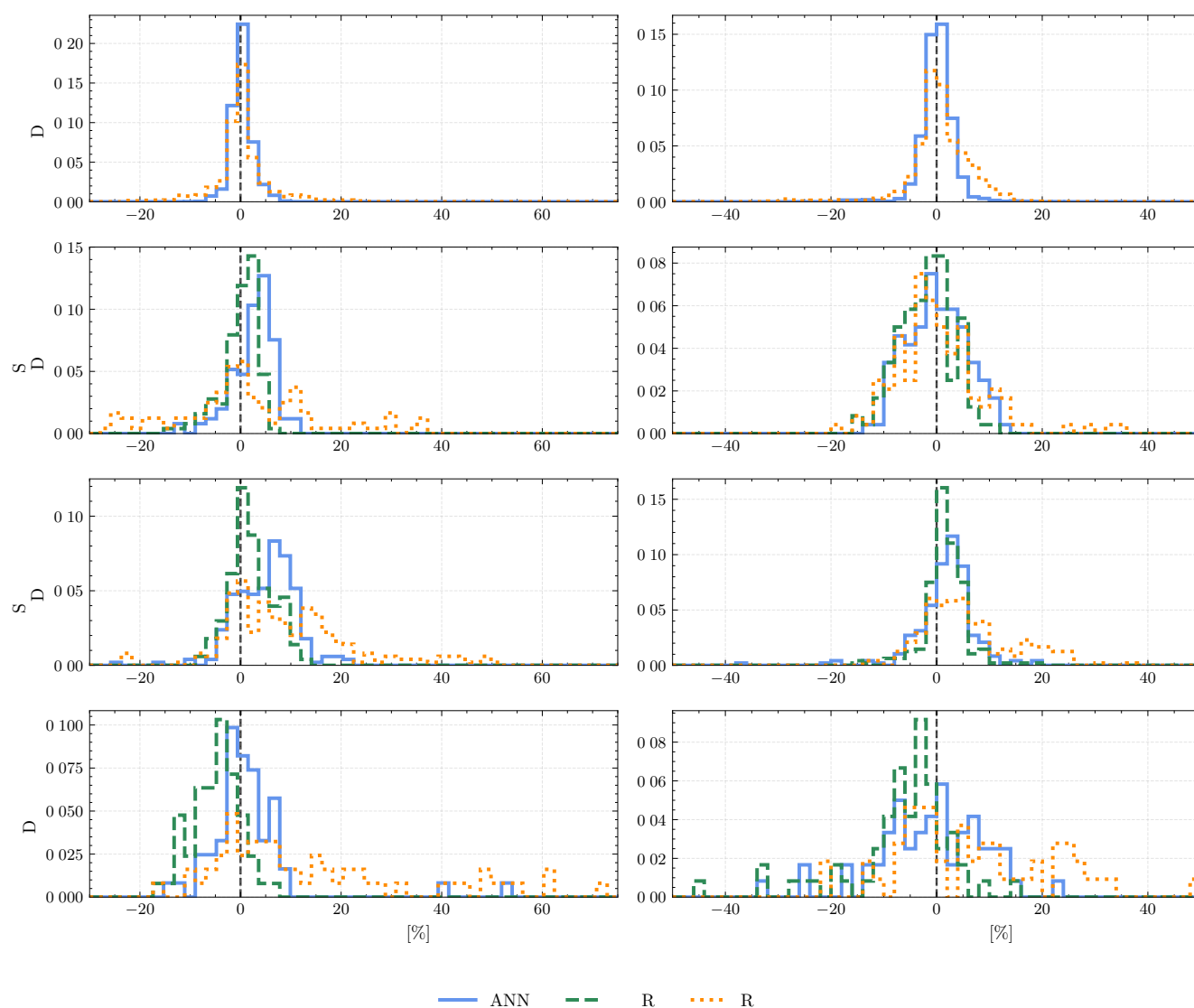


Figure 11. Percentage-error distributions of the ANN, GPR, and RF predictions for the tower-top and tower-base combined moments within the test dataset. Columns correspond to target channels and rows to operational modes. The vertical black dashed line indicates zero prediction error, and the horizontal-axis limits are consistent within each column. Curves are normalized to represent probability density.

Overall, the ANN and GPR models exhibit closely aligned and comparatively narrow error distributions across the evaluated target channels and operational modes. In contrast, the RF models consistently produce wider distributions, indicating greater error variability and supporting the lower predictive performance observed from the quantitative metrics. A systematic tendency towards overprediction is particularly evident for the shaft combined bending moment during SD, for which the error distributions of all three model families are shifted towards positive values relative to the zero-error line.



575 Figs. 12 and 13 show the percentage prediction errors as a function of rotor-averaged wind speed, with operational modes arranged by row and representative target channels by column. This visualization helps identify wind-speed regions in which the models exhibit increased error or systematic bias.

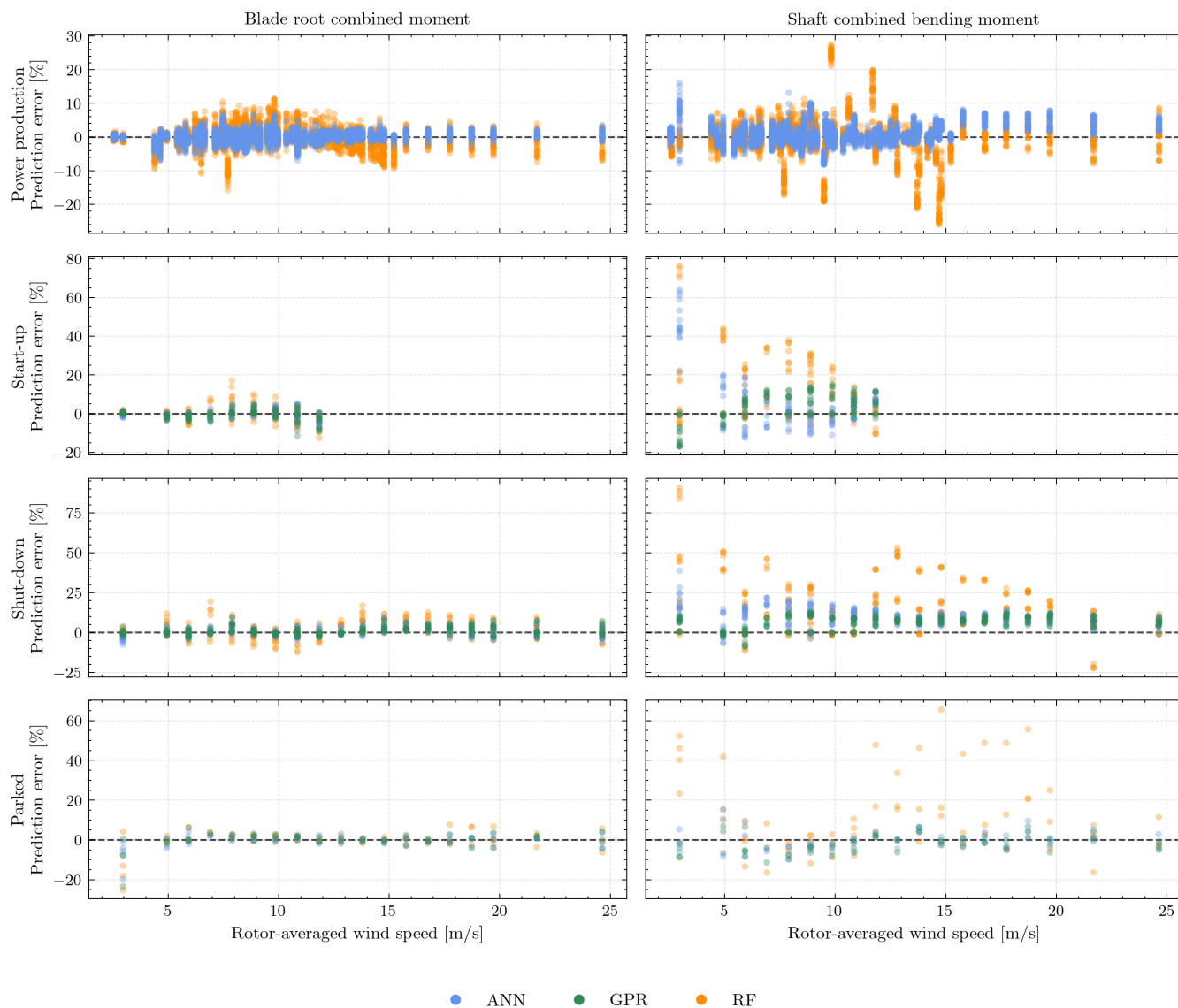


Figure 12. Percentage prediction error as a function of rotor-averaged wind speed for the ANN, GPR, and RF models, evaluated for the blade-root and shaft combined bending moments. Columns represent the selected target channels and rows represent the operational modes. The horizontal dashed line indicates zero prediction error.

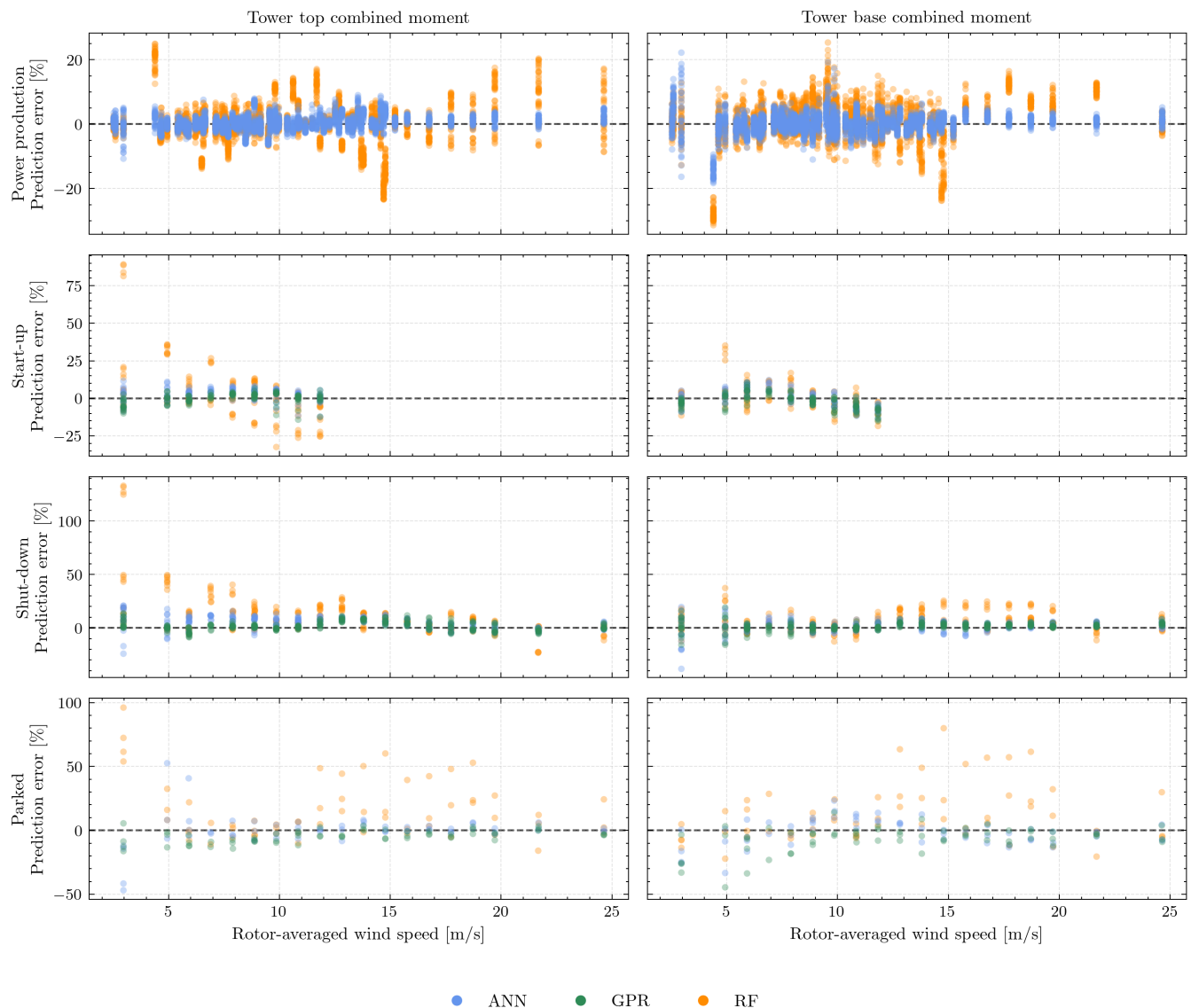


Figure 13. Percentage prediction error as a function of rotor-averaged wind speed for the ANN, GPR, and RF models, evaluated for the tower top and tower base combined moments. Columns represent the selected target channels and rows represent the operational modes. The horizontal dashed line indicates zero prediction error.

In the present results, no wind-speed region appears to consistently produce higher errors across all targets and operational modes. However, since the dataset is multivariate, an apparent concentration of errors within a particular wind-speed range cannot necessarily be attributed to wind speed alone. It may also result from other input variables, such as turbulence-intensity values, that occur predominantly within that region. Therefore, this visualization provides an initial assessment of regions associated with high prediction errors. However, further analysis is required to determine the underlying causes of potential



errors. Users can select different variables for the x-axis, such as rotor-averaged turbulence intensity, power setpoint, or yaw angle, to examine how the prediction errors vary across the input space and identify regions of reduced model accuracy.

585 4.4 Surrogate aeroelastic response

The surrogate aeroelastic response plots provide a complementary means of visualizing the behaviour of the data-driven models under prescribed operating and ambient conditions. An input dataset can be generated on the fly by fixing selected variables and sweeping another variable, as demonstrated here for rotor-averaged wind speed. Alternatively, multiple input combinations representing specific ambient conditions may be generated and passed through the models to estimate the corresponding DELs,
590 which could support applications such as fatigue-damage accumulation and remaining-useful-life estimation. This approach also allows the models to be inspected in regions that are sparsely represented, or not represented, in the test set.

Since these inputs are generated independently by sweeping the wind speed over a continuous range, no corresponding aeroelastic simulations are available for direct comparison at every evaluated point. However, the primary purpose of this analysis is not to assess predictive accuracy against individual ground-truth simulations, which is addressed in Sections 4.1
595 and 4.3, but rather to examine the qualitative behaviour of the surrogate models across a continuous range of inputs. This representation provides insight into how the models respond to systematic changes in the input parameters and helps identify potential limitations arising from the characteristics of the training dataset.

Fig. 14 presents the predicted blade-root and main-shaft DEL responses for the four operational modes. Correspondingly, Fig. 15 presents the predicted tower-top and tower-base DEL responses.

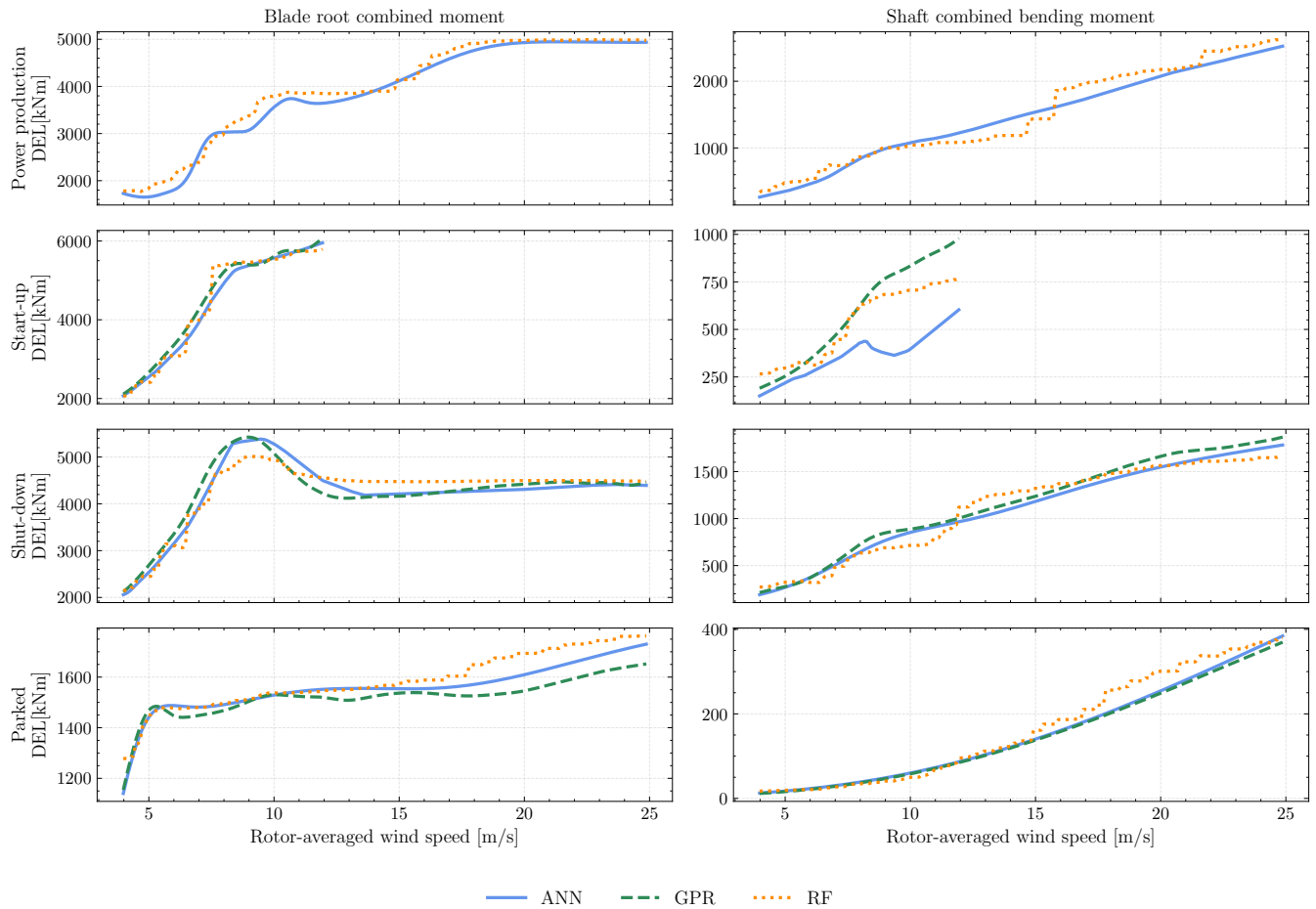


Figure 14. Surrogate aeroelastic responses of the blade-root combined DEL and shaft combined bending DEL as functions of rotor-averaged wind speed. The input dataset is generated on the fly by varying wind speed from 4 to 25 m/s, except for start-up, where the range is 4–12 m/s. Turbulence intensity is fixed at 0.10, the shear exponent at 0.20, the yaw angle at 0° where applicable, and the power setpoint at 1.0 where applicable.

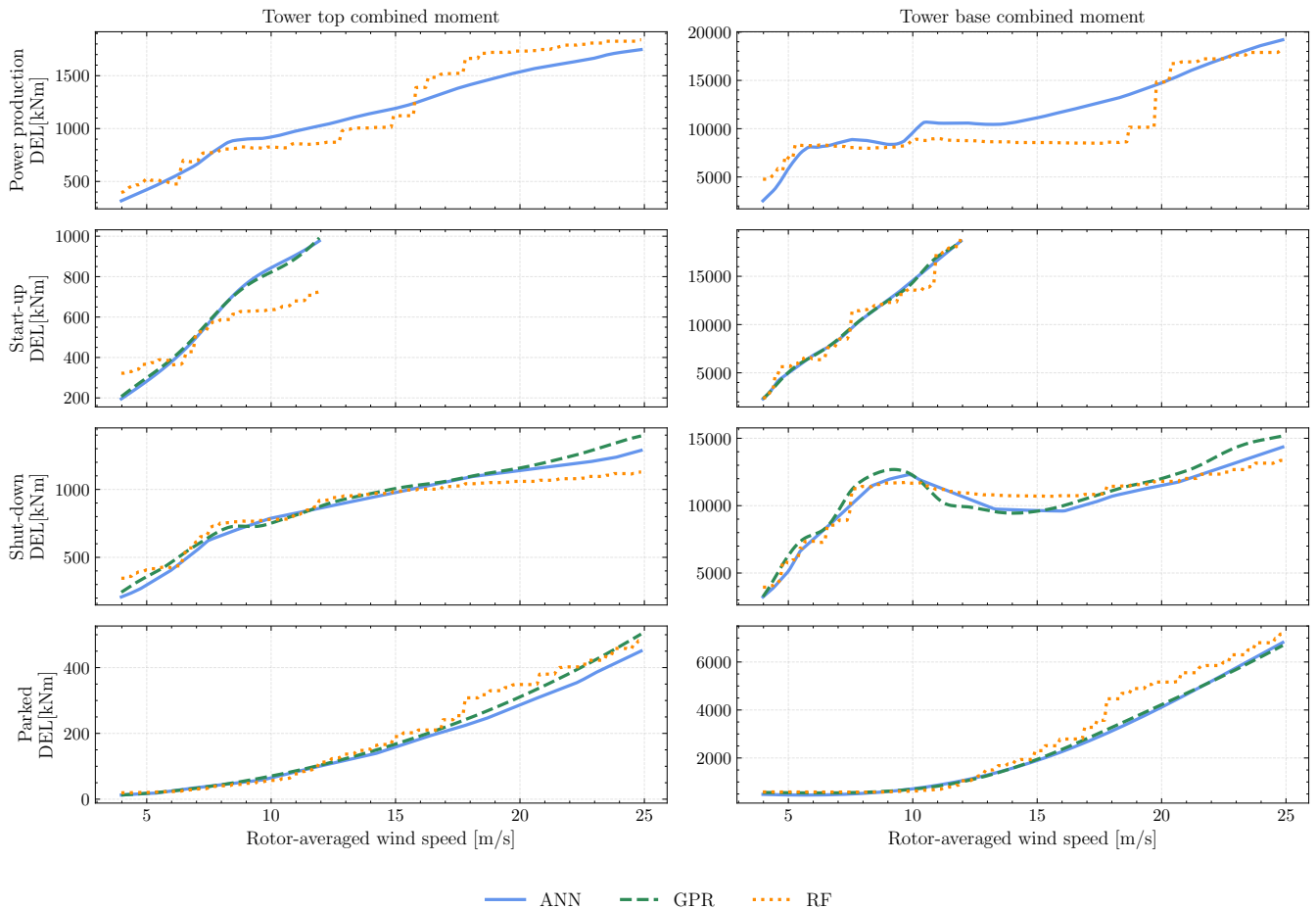


Figure 15. Surrogate aeroelastic responses of the tower-top and tower-base combined DELs as functions of rotor-averaged wind speed. The input dataset is generated on the fly by varying wind speed from 4 to 25 m/s, except for start-up, where the range is 4–12 m/s. Turbulence intensity is fixed at 0.10, the shear exponent at 0.20, the yaw angle at 0° where applicable, and the power setpoint at 1.0 where applicable.

600 The stepwise RF responses reflect the inherent partitioning of the input space by decision trees, which assign constant predictions within each terminal region. Consequently, RF models do not provide smooth interpolation between training samples, even when the underlying physical response is expected to vary continuously. This characteristic is clearly exposed by the present wind-speed sweeps, particularly in operational modes with sparse or non-uniformly distributed training data. The resulting abrupt changes and plateaus are unlikely to represent the physical variation of turbine loads with wind speed and

605 indicate a limitation of RF models for reproducing continuous aeroelastic response curves. By contrast, the ANN and GPR models generally produce smooth and mutually consistent load responses across the evaluated wind-speed ranges. Agreement between these independently trained model families increases confidence that the predicted trends reflect patterns present in the underlying simulation data, although it does not establish their physical accuracy in the absence of corresponding aeroelastic simulations. A notable exception is the shaft combined bending DEL during start-up, for which the ANN and GPR predictions



610 differ substantially in both magnitude and response trend. This discrepancy may indicate that one of the models is overfitting the available training data. With insufficient training samples to adequately constrain the response, the models may learn different approximations of the underlying relationship, leading to substantial differences in the predicted magnitude and trend.

Determining which model provides the more plausible response would require examining the distribution and observed behaviour of the relevant training and test samples. Additional sweeps over multiple combinations of turbulence intensity, shear exponent, yaw angle, and other applicable inputs would also be necessary to establish whether the discrepancy is persistent or specific to the selected input conditions. This example demonstrates how surrogate aeroelastic response plots can reveal model-specific behaviour and regions of disagreement that may not be apparent from aggregate performance metrics alone.

5 Conclusions

This study presented an openly accessible benchmark dataset and a modular data-driven modelling framework for predicting operational variables and DELs of the IEA 3.4 MW reference wind turbine. The dataset was generated through aeroelastic simulations covering multiple operational modes and control setpoints, while the accompanying framework supports the complete surrogate-modelling workflow, including data preprocessing, model training for ANNs, GPR and RF models, performance evaluation, and postprocessing.

The objective of this work was to provide a transparent and reproducible foundation for future model-development and benchmarking studies, rather than to introduce a novel surrogate-modelling methodology. The proposed resources support several complementary applications. The aeroelastic simulation dataset can be used to train alternative surrogate-model families or to investigate different preprocessing, feature-engineering, training, and postprocessing strategies, thereby enabling the comparison of new modelling approaches using a common underlying dataset. Beyond surrogate modelling, the dataset can also be used to investigate trends in the turbine's aeroelastic response across a broad range of operating conditions characterized by inflow and control variables. It may therefore provide a useful reference for researchers seeking to understand the response of a representative wind turbine before developing other types of data-driven, reduced-order, or physics-based models.

Conversely, the surrogate-training framework can be applied to other aeroelastic simulation datasets, provided that they follow a compatible file structure or can be readily adapted to the required structure. In this context, the implemented models and evaluation procedures provide a reference workflow for conducting systematic performance comparisons similar to those presented in Section 4.



Appendix

Table 12. Model performance comparison for the power production operational mode.

Target channel	ANN			RF		
	R^2 [-]	Mean error $\pm$ std	NRMSE [%]	R^2 [-]	Mean error $\pm$ std	NRMSE [%]
Operational variables						
Generator power	1.000	-0.6 $\pm$ 14.9	0.45	0.998	-11.7 $\pm$ 51.9	1.59
Rotor speed	1.000	-0.0 $\pm$ 0.0	0.46	0.999	-0.0 $\pm$ 0.1	0.89
Pitch angle ADC	0.996	0.4 $\pm$ 2.9	1.51	0.986	0.6 $\pm$ 5.7	3.00
Blade loads						
Blade root edgewise moment	0.997	1.7 $\pm$ 16.1	1.06	0.991	-0.7 $\pm$ 28.1	1.84
Blade root flapwise moment	0.997	4.0 $\pm$ 75.1	0.76	0.984	-17.1 $\pm$ 167.9	1.70
Blade root pitching moment	0.997	0.1 $\pm$ 0.4	0.71	0.984	-0.0 $\pm$ 0.9	1.74
Blade root in-plane moment	0.999	0.8 $\pm$ 11.9	0.38	0.993	-10.2 $\pm$ 35.4	1.17
Blade root out-of-plane moment	0.997	7.8 $\pm$ 40.0	0.68	0.984	5.7 $\pm$ 95.1	1.59
Blade root combined moment	0.997	0.1 $\pm$ 60.9	1.20	0.985	-1.6 $\pm$ 126.3	2.50
Blade bearing combined moment	0.998	3.7 $\pm$ 30.3	0.80	0.987	-5.0 $\pm$ 75.4	1.97
Main shaft loads						
Low-speed shaft torque	0.998	0.6 $\pm$ 5.4	1.03	0.990	-0.1 $\pm$ 12.6	2.39
Low-speed shaft y bending moment	0.997	6.2 $\pm$ 54.0	0.64	0.981	3.4 $\pm$ 143.5	1.70
Low-speed shaft z bending moment	0.997	10.7 $\pm$ 59.3	0.71	0.982	2.7 $\pm$ 140.4	1.65
Shaft combined bending moment	0.994	10.2 $\pm$ 40.3	1.19	0.974	-2.9 $\pm$ 84.8	2.43
Tower top loads						
Yaw bearing roll moment	0.997	1.1 $\pm$ 6.6	1.11	0.994	-0.1 $\pm$ 10.6	1.74
Yaw bearing pitch moment	0.996	9.2 $\pm$ 41.2	1.00	0.979	2.3 $\pm$ 96.3	2.28
Yaw bearing yaw moment	0.990	10.2 $\pm$ 62.7	1.58	0.972	-4.9 $\pm$ 107.3	2.67
Tower top combined moment	0.997	3.5 $\pm$ 26.4	0.68	0.977	5.0 $\pm$ 73.3	1.89
Tower base loads						
Tower base side-side moment	0.983	30.8 $\pm$ 488.2	2.41	0.976	44.4 $\pm$ 585.0	2.89
Tower base fore-aft moment	0.996	31.0 $\pm$ 349.2	1.34	0.982	79.6 $\pm$ 705.4	2.72
Tower base torsional moment	0.993	7.6 $\pm$ 53.4	1.30	0.970	-8.3 $\pm$ 114.1	2.76
Tower base combined moment	0.996	12.1 $\pm$ 320.0	1.38	0.981	108.5 $\pm$ 674.1	2.94

Units of the mean error and standard deviation: generator power in kW, rotor speed in RPM, pitch angle ADC in degrees, and all load channels in kNm.



Table 13. Model performance comparison for the start-up operational mode.

Target channel	ANN			GPR			RF		
	R^2 [-]	Mean error $\pm$ std	NRMSE [%]	R^2 [-]	Mean error $\pm$ std	NRMSE [%]	R^2 [-]	Mean error $\pm$ std	NRMSE [%]
Blade loads									
Blade root edgewise moment	0.994	11.5 $\pm$ 22.4	2.37	0.995	7.4 $\pm$ 21.2	2.11	0.989	18.0 $\pm$ 27.9	3.12
Blade root flapwise moment	0.962	104.1 $\pm$ 282.0	5.41	0.963	67.6 $\pm$ 287.3	5.31	0.950	86.2 $\pm$ 332.2	6.17
Blade root pitching moment	0.817	0.7 $\pm$ 0.8	8.66	0.894	0.5 $\pm$ 0.6	6.61	0.604	1.0 $\pm$ 1.1	12.75
Blade root in-plane moment	0.998	0.8 $\pm$ 17.0	1.43	0.998	5.7 $\pm$ 17.7	1.56	0.998	3.5 $\pm$ 17.3	1.48
Blade root out-of-plane moment	0.991	16.3 $\pm$ 66.5	2.54	0.989	-18.3 $\pm$ 73.1	2.79	0.969	22.5 $\pm$ 123.2	4.64
Blade root combined moment	0.991	-12.2 $\pm$ 130.4	2.83	0.989	-35.9 $\pm$ 143.5	3.20	0.977	3.2 $\pm$ 212.4	4.60
Blade bearing combined moment	0.993	2.1 $\pm$ 44.7	2.43	0.993	6.7 $\pm$ 43.2	2.37	0.963	40.4 $\pm$ 92.4	5.48
Main shaft loads									
Low-speed shaft torque	0.996	-0.6 $\pm$ 6.8	2.42	0.996	-0.5 $\pm$ 6.6	2.35	0.973	1.0 $\pm$ 17.1	6.08
Low-speed shaft y bending moment	0.918	116.0 $\pm$ 116.1	8.20	0.950	74.6 $\pm$ 104.3	6.41	0.746	196.8 $\pm$ 211.5	14.43
Low-speed shaft z bending moment	0.899	124.5 $\pm$ 131.5	8.96	0.946	76.3 $\pm$ 108.3	6.55	0.740	193.7 $\pm$ 216.7	14.38
Shaft combined bending moment	0.977	12.3 $\pm$ 52.3	4.11	0.962	40.2 $\pm$ 54.9	5.21	0.843	91.8 $\pm$ 104.9	10.67
Tower top loads									
Yaw bearing roll moment	0.995	3.0 $\pm$ 7.4	2.42	0.996	0.7 $\pm$ 6.8	2.08	0.975	5.2 $\pm$ 17.5	5.55
Yaw bearing pitch moment	0.992	20.3 $\pm$ 38.6	2.39	0.996	10.0 $\pm$ 29.1	1.69	0.917	76.4 $\pm$ 113.2	7.50
Yaw bearing yaw moment	0.960	61.5 $\pm$ 62.6	5.43	0.983	29.5 $\pm$ 49.4	3.56	0.888	57.3 $\pm$ 135.6	9.11
Tower top combined moment	0.990	19.8 $\pm$ 31.5	2.65	0.995	3.8 $\pm$ 27.1	1.95	0.912	6.9 $\pm$ 110.9	7.92
Tower base loads									
Tower base side-side moment	0.850	-32.4 $\pm$ 539.1	8.32	0.841	-83.2 $\pm$ 550.5	8.57	0.791	12.8 $\pm$ 637.8	9.83
Tower base fore-aft moment	0.935	-526.7 $\pm$ 1669.4	5.39	0.915	-790.6 $\pm$ 1835.0	6.15	0.898	-622.4 $\pm$ 2105.4	6.76
Tower base torsional moment	0.965	60.7 $\pm$ 60.7	5.03	0.984	29.6 $\pm$ 49.5	3.38	0.892	62.6 $\pm$ 138.1	8.89
Tower base combined moment	0.981	-146.5 $\pm$ 717.4	3.45	0.977	-333.0 $\pm$ 738.4	3.82	0.969	-128.4 $\pm$ 939.8	4.47

The mean error and standard deviation for all load channels are expressed in kNm.



Table 14. Model performance comparison for the shut-down operational mode.

Target channel	ANN			GPR			RF		
	R^2 [-]	Mean error $\pm$ std	NRMSE [%]	R^2 [-]	Mean error $\pm$ std	NRMSE [%]	R^2 [-]	Mean error $\pm$ std	NRMSE [%]
Blade loads									
Blade root edgewise moment	0.969	11.6 $\pm$ 44.0	3.54	0.973	20.0 $\pm$ 37.7	3.32	0.947	28.6 $\pm$ 52.4	4.64
Blade root flapwise moment	0.983	23.8 $\pm$ 141.8	2.93	0.981	49.7 $\pm$ 141.8	3.06	0.929	110.9 $\pm$ 272.7	6.00
Blade root pitching moment	0.913	0.4 $\pm$ 1.6	4.92	0.888	0.5 $\pm$ 1.8	5.58	0.826	0.7 $\pm$ 2.2	6.95
Blade root in-plane moment	0.980	35.5 $\pm$ 33.2	3.16	0.985	31.8 $\pm$ 28.0	2.75	0.964	53.5 $\pm$ 37.7	4.25
Blade root out-of-plane moment	0.973	75.4 $\pm$ 77.0	3.55	0.975	69.0 $\pm$ 78.3	3.44	0.904	158.4 $\pm$ 127.8	6.71
Blade root combined moment	0.983	31.5 $\pm$ 116.8	3.09	0.984	46.5 $\pm$ 109.4	3.03	0.937	73.2 $\pm$ 223.7	6.01
Blade bearing combined moment	0.971	61.6 $\pm$ 61.8	4.45	0.978	53.3 $\pm$ 54.1	3.87	0.902	112.3 $\pm$ 116.4	8.25
Main shaft loads									
Low-speed shaft torque	0.999	0.6 $\pm$ 3.8	0.83	0.999	1.4 $\pm$ 4.0	0.92	0.984	2.9 $\pm$ 16.9	3.70
Low-speed shaft y bending moment	0.928	208.0 $\pm$ 150.2	6.47	0.949	155.7 $\pm$ 149.9	5.45	0.846	311.3 $\pm$ 208.3	9.45
Low-speed shaft z bending moment	0.955	147.4 $\pm$ 137.1	5.09	0.954	141.3 $\pm$ 147.2	5.16	0.850	304.6 $\pm$ 206.7	9.31
Shaft combined bending moment	0.952	104.8 $\pm$ 72.3	4.92	0.968	78.7 $\pm$ 68.5	4.03	0.868	158.3 $\pm$ 140.2	8.17
Tower top loads									
Yaw bearing roll moment	0.998	2.6 $\pm$ 6.3	1.16	0.998	4.6 $\pm$ 5.1	1.17	0.984	8.7 $\pm$ 17.7	3.36
Yaw bearing pitch moment	0.976	97.1 $\pm$ 66.3	3.49	0.981	77.4 $\pm$ 70.3	3.11	0.895	176.4 $\pm$ 168.1	7.24
Yaw bearing yaw moment	0.982	73.5 $\pm$ 66.8	2.96	0.988	60.3 $\pm$ 56.7	2.47	0.911	142.8 $\pm$ 172.4	6.68
Tower top combined moment	0.986	35.9 $\pm$ 44.8	2.66	0.993	10.6 $\pm$ 38.6	1.86	0.939	74.6 $\pm$ 93.3	5.54
Tower base loads									
Tower base side-side moment	0.973	-325.2 $\pm$ 666.1	3.28	0.962	-153.1 $\pm$ 865.6	3.89	0.958	270.9 $\pm$ 889.5	4.12
Tower base fore-aft moment	0.985	69.4 $\pm$ 542.5	2.25	0.987	101.2 $\pm$ 499.7	2.10	0.940	251.9 $\pm$ 1073.9	4.54
Tower base torsional moment	0.977	90.9 $\pm$ 74.5	3.38	0.987	65.2 $\pm$ 59.7	2.54	0.909	154.3 $\pm$ 176.7	6.75
Tower base combined moment	0.988	194.9 $\pm$ 384.1	2.17	0.990	149.2 $\pm$ 357.9	1.95	0.956	484.9 $\pm$ 664.6	4.14

The mean error and standard deviation for all load channels are expressed in kNm.



Table 15. Model performance comparison for the parked operational mode.

Target channel	ANN			GPR			RF		
	R^2 [-]	Mean error $\pm$ std	NRMSE [%]	R^2 [-]	Mean error $\pm$ std	NRMSE [%]	R^2 [-]	Mean error $\pm$ std	NRMSE [%]
Blade loads									
Blade root edgewise moment	0.992	-8.4 ± 10.6	1.77	0.990	-9.8 ± 11.3	1.95	0.977	4.9 ± 22.5	3.01
Blade root flapwise moment	0.859	33.3 ± 227.8	6.59	0.880	41.9 ± 208.8	6.10	0.857	60.3 ± 223.7	6.63
Blade root pitching moment	0.915	0.2 ± 1.2	5.10	0.920	0.2 ± 1.1	4.94	0.904	0.5 ± 1.2	5.41
Blade root in-plane moment	0.913	7.9 ± 74.2	4.93	0.921	18.4 ± 68.3	4.67	0.901	26.0 ± 74.8	5.23
Blade root out-of-plane moment	0.998	-2.7 ± 3.7	0.90	0.998	-3.2 ± 3.5	0.93	0.967	8.1 ± 17.4	3.76
Blade root combined moment	0.962	-3.3 ± 43.0	2.60	0.951	1.5 ± 49.3	2.97	0.904	3.5 ± 68.7	4.14
Blade bearing combined moment	0.975	-3.7 ± 23.3	2.36	0.980	0.4 ± 20.9	2.09	0.959	7.4 ± 28.9	2.98
Main shaft loads									
Low-speed shaft torque	0.998	-0.0 ± 0.0	1.04	0.998	-0.0 ± 0.0	1.02	0.967	0.0 ± 0.0	3.85
Low-speed shaft y bending moment	0.992	-4.5 ± 11.9	1.83	0.991	-7.9 ± 10.4	1.87	0.970	6.8 ± 23.2	3.48
Low-speed shaft z bending moment	0.995	-0.9 ± 9.4	1.33	0.996	-1.3 ± 8.6	1.22	0.966	11.9 ± 21.6	3.47
Shaft combined bending moment	0.998	-0.2 ± 5.5	0.92	0.998	-1.8 ± 5.8	1.02	0.968	11.4 ± 20.7	3.94
Tower top loads									
Yaw bearing roll moment	0.998	-0.3 ± 0.8	0.98	0.998	-0.3 ± 0.8	1.00	0.970	1.4 ± 2.6	3.63
Yaw bearing pitch moment	0.998	-2.4 ± 7.0	0.89	0.998	-3.4 ± 7.1	0.95	0.967	16.1 ± 25.8	3.69
Yaw bearing yaw moment	0.999	-0.2 ± 6.5	0.69	0.998	-3.5 ± 8.6	1.00	0.972	15.6 ± 29.3	3.54
Tower top combined moment	0.999	0.9 ± 6.3	0.77	0.997	-5.3 ± 7.1	1.07	0.968	16.1 ± 25.1	3.62
Tower base loads									
Tower base side-side moment	0.994	-3.3 ± 309.5	1.63	0.993	-65.6 ± 334.2	1.79	0.963	300.0 ± 701.7	4.01
Tower base fore-aft moment	0.994	32.1 ± 118.1	1.48	0.994	-33.5 ± 120.8	1.52	0.955	150.5 ± 306.2	4.13
Tower base torsional moment	0.999	-0.4 ± 6.5	0.69	0.998	-3.2 ± 8.7	0.97	0.971	16.3 ± 30.0	3.58
Tower base combined moment	0.990	-79.2 ± 233.1	2.12	0.988	-131.8 ± 232.4	2.30	0.959	177.1 ± 471.7	4.33

The mean error and standard deviation for all load channels are expressed in kNm.

Data availability. The database containing the synthetic inflow profiles used in this study is available on Zenodo at <https://doi.org/10.5281/zenodo.18634031> (Guilloré et al. (2026a)). The HAWC2 input files and the simulation datasets used to train and evaluate the surrogate models are available at <https://doi.org/10.5281/zenodo.22911878>. The surrogate-model library containing the trained models developed using the proposed framework is available at <https://doi.org/10.5281/zenodo.22670615>.

Code availability. The source code used to preprocess the data and train and evaluate the surrogate models is available in Zenodo at Tsaklis et al. (2026b).

Author contributions. IT developed and implemented the surrogate-model training and evaluation framework, coordinated the overall study, analysed and interpreted the results, and wrote the majority of the manuscript. VP contributed to the conceptualization of the study, wind-field dataset generation, methodological development of the framework, aeroelastic model development and controller tuning, as well as the writing and review of the manuscript. TG contributed to the methodological development of the framework, provided guidance on the study



conceptualization and interpretation of the results, and reviewed the manuscript. NCG contributed to the development of the aeroelastic model and post-processing framework, the generation of the aeroelastic simulation database used as input to the surrogate-model training framework and the writing of the manuscript. ND contributed to the methodological development and provided guidelines on the conceptualization and interpretation of results. DAJ contributed to surrogate framework conceptualization, generation and optimization of the RF models and wrote corresponding parts of the manuscript; ILS, MAS & IE contributed to database design, surrogate framework conceptualization and reviewed and completed the manuscript. AG led the methodological development and implementation of the wake slices approach, generated and shared the library of inflow profiles, wrote the corresponding section of the manuscript, and contributed to the development of location-agnostic load surrogate methodology. AA contributed to the methodological development and overall framework implementation, led the standardization of data preparation and sharing procedures, and reviewed the manuscript. AHS generated the library of clean inflow profiles and implemented the LHS-based selection of inflow conditions. AV led the integration of steady-state wake models into the surrogate framework used for LHS-based design-space exploration and inflow profile selection, and contributed to the implementation of the load surrogates. CLB conceived and developed the underlying concepts for location-agnostic load surrogates and the efficient generation of wake slices based on similarity laws, provided scientific guidance throughout the project, and supervised the research.

Competing interests. The authors have the following competing interests: At least one of the (co-)authors is a member of the editorial board of Wind Energy Science.

Acknowledgements. The authors acknowledge the support of the research project collaborators and the computational resources used for generating the aeroelastic simulation dataset. This work has been supported by the TWAIN project, which receives funding from the European Union's Horizon Europe Programme under the grant agreement No. 101122194.



665 References

- Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.
- Bay, C. J., Guilloré, A., Bortolotti, P., Frontin, C., Sinner, M., Thedin, R., and Bottasso, C. L.: Impacts on damage equivalent loads from plant optimization using artificial neural network surrogate models, *Journal of Physics: Conference Series*, 3224, 032125,
 670 <https://doi.org/10.1088/1742-6596/3224/3/032125>, 2026.
- Bergstra, J., Bardenet, R., and Kégl, B.: Algorithms for Hyper-Parameter Optimization, 2011.
- Bortolotti, P., Tarrés, H., Dykes, K., Merz, K., Sethuraman, L., Verelst, D., and Zahle, F.: IEA Wind TCP Task 37: Systems Engineering in Wind Energy-WP2.1 Reference Wind Turbines, National Renewable Energy Laboratory (NREL), <https://doi.org/10.2172/1529216>, 2019.
- Branlard, E., Martínez-Tossas, L. A., and Jonkman, J.: A time-varying formulation of the curled wake model within the FAST.Farm frame-
 675 [work](https://doi.org/10.1002/we.2785), *Wind Energy*, 26, 44–63, <https://doi.org/10.1002/we.2785>, 2023.
- Branlard, E., Jonkman, J., Platt, A., Thedin, R., Martínez-Tossas, L. A., and Kretschmer, M.: Development and Verification of an Improved Wake-Added Turbulence Model in FAST.Farm, *Journal of Physics: Conference Series*, 2767, 092036, <https://doi.org/10.1088/1742-6596/2767/9/092036>, 2024.
- Breiman, L.: Random Forests, *Machine Learning*, 45, 5–32, <https://doi.org/10.1023/a:1010933404324>, 2001.
- 680 Dimitrov, N., Kelly, M., Vignaroli, A., and Berg, J.: From wind to loads: wind turbine site-specific load estimation with surrogate models trained on high-fidelity load databases, *Wind energy science*, 3, 767–790, <https://doi.org/10.5194/wes-3-767-2018>, 2018.
- Dimitrov, N. K., Marelli, S., and Schär, S.: Novel surrogate modelling approaches for wind turbine reliability assessment, HIPERWIND Deliverable D4.1, ETH Zurich and Technical University of Denmark (DTU), 2022.
- DTU Wind Energy: Wind Energy Toolbox (WETB), <https://github.com/DTUWindEnergy/WindEnergyToolbox>.
- 685 Eguinoa, I., Göçmen, T., Garcia-Rosa, P. B., Das, K., Petrović, V., Kölle, K., Manjock, A., Koivisto, M. J., and Smailes, M.: Wind farm flow control oriented to electricity markets and grid integration: Initial perspective analysis, *Advanced Control for Applications*, 3, e80, <https://doi.org/https://doi.org/10.1002/adc2.80>, 2021.
- European Commission: TWAIN: Integrated, Value-based and Multi-objective Wind Farm Control Powered by Artificial Intelligence, <https://doi.org/10.3030/101122194>, horizon Europe, Grant Agreement No. 101122194, 2023.
- 690 Geurts, P., Ernst, D., and Wehenkel, L.: Extremely randomized trees, *Machine Learning*, 63, 3–42, <https://doi.org/10.1007/s10994-006-6226-1>, 2006.
- Guilloré, A., Shah, A. H., and Anand, A.: Datasets as supplement for the journal publication "Modeling wind farm response: a modular, integrated, and multi-stakeholder approach", <https://doi.org/10.5281/zenodo.18634031>, data set, 2026a.
- Guilloré, A., Campagnolo, F., and Bottasso, C. L.: A control-oriented load surrogate model based on sector-averaged inflow quantities: capturing damage for unwaked, waked, wake-steering and curtailed wind turbines, *Journal of Physics: Conference Series*, 2767, 032019,
 695 <https://doi.org/10.1088/1742-6596/2767/3/032019>, 2024.
- Guilloré, A., Chaudhary, A., Anand, A., Vad, A., Shah, A. H., Pettas, V., Göçmen, T., and Bottasso, C. L.: Efficient generation of location-agnostic wind turbine load surrogate models using wake slices, *Journal of Physics: Conference Series*, 3224, 032106, <https://doi.org/10.1088/1742-6596/3224/3/032106>, 2026b.



- 700 Hlaing, N., Morato, P. G., Santos, F. d. N., Weijtjens, W., Devriendt, C., and Rigo, P.: Farm-wide virtual load monitoring for offshore wind structures via Bayesian neural networks, *Structural Health Monitoring*, 23, 1641–1663, <https://doi.org/10.1177/14759217231186048>, 2023.
- Jonkman, B. J.: *TurbSim User's Guide: Version 1.50*, Tech. Rep. NREL/TP-500-46198, National Renewable Energy Laboratory (NREL), <https://doi.org/10.2172/965520>, 2009.
- 705 Jonkman, J., Doubrawa, P., Hamilton, N., Annoni, J., and Fleming, P.: Validation of FAST.Farm Against Large-Eddy Simulations, *Journal of Physics: Conference Series*, 1037, 062 005, <https://doi.org/10.1088/1742-6596/1037/6/062005>, 2018.
- Kheirabadi, A. C. and Nagamune, R.: A quantitative review of wind farm control with the objective of wind farm power maximization, *Journal of Wind Engineering and Industrial Aerodynamics*, 192, 45–73, <https://doi.org/https://doi.org/10.1016/j.jweia.2019.06.015>, 2019.
- Knudsen, T., Bak, T., and Svenstrup, M.: Survey of wind farm control—power and fatigue optimization, *Wind Energy*, 18, 1333–1351, <https://doi.org/https://doi.org/10.1002/we.1760>, 2015.
- 710 Konis, P., Hlaing, N., Noppe, N., Weijtjens, W., and Devriendt, C.: Fleet-wide fatigue assessment of offshore wind turbines using artificial and Bayesian neural networks, *J. Phys. Conf. Ser.*, 3224, 062 028, 2026.
- Kretschmer, M., Jonkman, J., Pettas, V., and Cheng, P. W.: FAST.Farm load validation for single wake situations at alpha ventus, *Wind Energy Science*, 6, 1247–1262, <https://doi.org/10.5194/wes-6-1247-2021>, 2021.
- 715 Larsen, G. C.: *Dynamic Wake Meandering Modeling*, Tech. Rep. Risø-R-1607(EN), Risø National Laboratory for Sustainable Energy, Technical University of Denmark, Roskilde, Denmark, <https://orbit.dtu.dk/en/publications/dynamic-wake-meandering-modeling>, 2007.
- Larsen, T. and Hansen, A.: *How 2 HAWC2, the user's manual, no. 1597(ver. 3-1)(EN)* in Denmark. Forskningscenter Risoe. Risoe-R, Risø National Laboratory, ISBN 978-87-550-3583-6, 2007.
- Liew, J. and Larsen, G. C.: How does the quantity, resolution, and scaling of turbulence boxes affect aeroelastic simulation convergence?, *Journal of Physics: Conference Series*, 2265, 032 049, <https://doi.org/10.1088/1742-6596/2265/3/032049>, 2022.
- 720 Martin, A., Ashish, A., Paul, B., Eugene, B., Zhifeng, C., Craig, C., Greg, S., C., Andy, D., Jeffrey, D., Matthieu, D., Sanjay, G., Ian, G., Andrew, H., Geoffrey, I., Michael, I., Jia, Y., Rafal, J., Lukasz, K., Manjunath, K., Josh, L., Dan, M., Rajat, M., Sherry, M., Derek, M., Chris, O., Mike, S., Jonathon, S., Benoit, S., Ilya, S., Kunal, T., Paul, T., Vincent, V., Vijay, V., Fernanda, V., Oriol, V., Pete, W., Martin, W., Martin, W., Yuan, Y., and Xiaoqiang, Z.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, <http://tensorflow.org/>, 2015.
- 725 McKay, M. D., Beckman, R. J., and Conover, W. J.: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, *Technometrics*, 21, 239–245, <https://doi.org/10.1080/00401706.1979.10489755>, 1979.
- Meng, F., Lio, W. H., and Barlas, T.: DTUWEC: an open-source DTU Wind Energy Controller with advanced industrial features, *Journal of Physics: Conference Series*, 1618, 022 009, <https://doi.org/10.1088/1742-6596/1618/2/022009>, 2020.
- 730 Miner, M. A.: Cumulative Damage in Fatigue, *Journal of Applied Mechanics, Transactions ASME*, 12, A159–A164, <https://doi.org/10.1115/1.4009458>, 1945.
- National Renewable Energy Laboratory: FAST.Farm, <https://github.com/OpenFAST/OpenFAST>, 2024.
- Nolasco Santos, F., D'Antuono, P., Robbelein, K., Noppe, N., Weijtjens, W., and Devriendt, C.: Long-term fatigue estimation on offshore wind turbines interface loads through loss function physics-guided learning of neural networks, *Renewable Energy*, 205, 461–474, <https://doi.org/10.1016/j.renene.2023.01.093>, 2023.
- 735 NREL: <https://github.com/NREL/openfast-turbine-models/tree/main/IEA-scaled>, GitHub repository, accessed March 12, 2025, 2024.



- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, 12, 2825–2830, 2011.
- 740 Pettas, V.: Wind Turbine Operational Optimization Considering Revenue and Fatigue Objectives, Doctoral dissertation, University of Stuttgart, Stuttgart, Germany, <https://doi.org/10.18419/opus-13959>, 2024.
- Pettas, V. and Cheng, P. W.: Down-regulation and individual blade control as lifetime extension enablers, *Journal of Physics: Conference Series*, 1102, 012 026, <https://doi.org/10.1088/1742-6596/1102/1/012026>, 2018.
- Pettas, V. and Cheng, P. W.: Surrogate Modeling and Aeroelastic Analysis of a Wind Turbine with Down-Regulation, Power Boosting, and
 745 IBC Capabilities, *Energies*, 17, 1284, <https://doi.org/10.3390/en17061284>, 2024.
- Pettas, V., Göçmen, T., Duc, T., and Vad, A.: On the effects of bat protection strategies on energy production and structural loads of wind farms, *Wind Energy Science*, <https://doi.org/10.5194/wes-2026-56>, 2026.
- Ramaswamy, H., Guilloré, A., Riva, R., Quick, J., Pettas, V., Murcia, J., Campagnolo, F., Bottasso, C., and Réthoré, P.: Comparison of wind turbine load surrogate model performances with multi-scale local inflow parameterization, *Journal of Physics: Conference Series*, 3224,
 750 032 073, <https://doi.org/10.1088/1742-6596/3224/3/032073>, 2026.
- Rasmussen, C. E. and Williams, C. K. I.: Gaussian processes for machine learning, *Adaptive Computation and Machine Learning series*, MIT Press, London, England, 2005.
- Schröder, L., Krasimirov Dimitrov, N., Verelst, D. R., and Sørensen, J. A.: Wind turbine site-specific load estimation using artificial neural networks calibrated by means of high-fidelity load simulations, *Journal of Physics: Conference Series*, 1037, 062 027,
 755 <https://doi.org/10.1088/1742-6596/1037/6/062027>, 2018.
- Shaler, K., Jasa, J., and Barter, G. E.: Efficient Loads Surrogates for Waked Turbines in an Array, *Journal of Physics: Conference Series*, 2265, 032 095, <https://doi.org/10.1088/1742-6596/2265/3/032095>, 2022.
- Singh, D., Dwight, R. P., Laugesen, K., Beaudet, L., and Viré, A.: Probabilistic surrogate modeling of offshore wind-turbine loads with chained Gaussian processes, *Journal of Physics: Conference Series*, 2265, 032 070, <https://doi.org/10.1088/1742-6596/2265/3/032070>,
 760 2022.
- Singh, D., Dwight, R., and Viré, A.: Probabilistic surrogate modeling of damage equivalent loads on onshore and offshore wind turbines using mixture density networks, *Wind Energy Science*, 9, 1885–1904, <https://doi.org/10.5194/wes-9-1885-2024>, 2024.
- Tietz, M., Fan, T. J., Nouri, D., Bossan, B., and skorch Developers: skorch: A scikit-learn compatible neural network library that wraps PyTorch, <https://skorch.readthedocs.io/en/stable/>, 2017.
- 765 Tsaklis, I., Gräfe, M. J., and Pettas, V.: Data-driven models for predicting structural loads in offshore wind turbines using SCADA data, *Journal of Physics Conference Series*, 3224, 062 042, <https://doi.org/10.1088/1742-6596/3224/6/062042>, 2026a.
- Tsaklis, I., Göçmen, T., Pettas, V., Lizarraga-Saenz, I., and Astrain-Juangarcia, D.: Surrogate model development IEA 3.4 MW, <https://doi.org/10.5281/zenodo.22685724>, 2026b.
- Vad, A., Guilloré, A., Anand, A., Pettas, V., Shah, A. H., Lizarraga-Saenz, I., Aparicio-Sanchez, M., Eguinoa, I., Conti Gost, N., Tsaklis, I.,
 770 Frère, A., Hermans, K., Kamau, J. K., Dimitrov, N., Göçmen, T., and Bottasso, C. L.: Modeling wind farm response: a modular, integrated, and multi-stakeholder approach, <https://doi.org/10.5194/wes-2026-45>, 2026.
- Veers, P., Dykes, K., Lantz, E., Barth, S., Bottasso, C. L., Carlson, O., Clifton, A., Green, J. B., Green, P. F., Holttinen, H., Laird, D., Lehtomäki, V., Lundquist, J. K., Manwell, J. F., Marquis, M., Meneveau, C., Moriarty, P., Munduate, X., Muskulus, M., Naughton, J., Pao,



- 775 L. Y., Paquette, J., Peinke, J., Robertson, A., Rodrigo, J. S., Sempreviva, A. M., Smith, J. C., Tuohy, A., and Wiser, R.: Grand challenges in the science of wind energy, *Science*, 366, <https://doi.org/10.1126/science.aau2027>, 2019.
- Veers, P., Dykes, K., Basu, S., Bianchini, A., Clifton, A., Green, P. F., Holttinen, H., Kitzing, L., Kosović, B., Lundquist, J. K., Meyers, J., O'Malley, M., Shaw, W. J., and Straw, B.: Grand Challenges: wind energy research needs for a global energy transition, *Wind energy science*, 7, 2491–2496, <https://doi.org/10.5194/wes-7-2491-2022>, 2022.
- 780 Vera-Tudela, L. and Kühn, M.: Analysing wind turbine fatigue load prediction: The impact of wind farm flow conditions, *Renewable Energy*, 107, 352–360, <https://doi.org/10.1016/j.renene.2017.01.065>, 2017.
- Wingerden, J., Fleming, P., Göçmen, T., Eguinoa, I., Doekemeijer, B., Dykes, K., Lawson, M., Simley, E., King, J., Astrain, D., Iribas, M., Bottasso, C. L., Meyers, J., Raach, S., Kölle, K., and Giebel, G.: Expert Elicitation on Wind Farm Control, *Journal of Physics Conference Series*, 1618, 022 025, <https://doi.org/10.1088/1742-6596/1618/2/022025>, 2020.