HIghly advanced Probabilistic design and Enhanced Reliability methods for high-value, cost-efficient offshore WIND

Title: Novel surrogate modelling approaches for wind turbine reliability assessment

Deliverable nº: D4.1

Delivery date: 31.03.2022
Lead beneficiary: ETH, DTU
Dissemination level: Public

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101006689

Author(s) information (alphabetical):

Name	Organisation	Email
Nikolay Krasimirov Dimitrov	DTU	nkdi@dtu.dk
Stefano Marelli	ETH	marelli@ibk.baug.ethz.ch
Styfen Schär	ETH	styfen.schaer@ibk.baug.ethz.ch

Acknowledgements/Contributions:

Name	Name	Name

Document information:

Version	Date	Description	Prepared by	Reviewed by	Approved by
1	31/3/2022	Report deliverable D4.1	Authors listed above	Prof. B. Sudret	Karen H. Thielsen

Definitions:

Contents

1. Executive Summary 6

2. Introduction 7

2.1. General context 7

2.2. Scope of the work and objectives 8

3. Surrogate modelling of high-dimensional input/output time-series (ETH) 9

3.1. Problem statement and definitions 9

3.2. Validation strategy 10

3.3. Computational model: the NREL 5MW reference turbine 11

3.3.1. Description of the wind turbine 11

3.3.2. Output quantities of interest 11

3.3.3. Wind box generation and discretization (TurbSim) 12

3.3.4. AES modelling (OpenFast) 12

3.4. Surrogate modelling strategy 12

3.4.1. Overview 12

3.4.2. mNARX: a chain of surrogate models 14

3.4.3. Spatial dependence: spectral compression 15

3.4.4. Time dependence: NARX 17

3.4.5. Wind regimes and classification (low/high wind speed regimes) 18

3.4.6. Training strategy 18

3.5. Surrogate model performance 19

3.5.1. Training and validation sets 20

3.5.2. Validation analysis: error plots and error measures 22

3.5.3. Detailed performance analysis on selected QoIs 23

3.5.4. Computational costs 61

3.6. Discussion and conclusions on the time series surrogate model 62

3.6.1. Performance analysis 62

3.6.2. Portability 62

3.6.3. Limitations/further work 62

4. Bayesian Neural Networks (BNN) 64

4.1. Introduction and problem description 64

4.1.1. Probabilistic regression models 65

4.1.2. Deterministic model - Input uncertainty propagation 66

4.1.3. Models with probabilistic outputs (stochastic simulators) ... 67

4.1.4. Considering model uncertainty – by ensemble models and models with probabilistic parameters ... 67

4.2. Study design and data generation ... 68

4.3. Ensemble models of deterministic neural networks ... 72

4.4. BNN models with epistemic and aleatory uncertainty ... 74

4.5. BNN model performance assessment ... 77

4.6. Multivariate models using BNNs ... 81

4.7. Conclusions on the use of BNN as load surrogates ... 82

5. Conclusions ... 83

References ... 83

List of Abbreviations

NARX	non-linear auto-regressive model with exogenous inputs
DEL	damage equivalent load
ASE	aero-servo-elastic
QoI	quantity of interest
NTM	normal turbulence model
ETM	extreme turbulence model
EOG	extreme operating gust
SD	standard data set
RD	replication data set
BNN	Bayesian neural network
FNN	feedforward neural network
KL divergence	Kullback-Leibler divergence
PDF	probability density function
FLS	fatigue limit state
ULS	ultimate limit state

List of figures

Figure 1: the HIPERWind work-package structure. ... 7
Figure 2: Graphical representation of the proposed mNARX surrogate modelling algorithm. ... 13
Figure 3: Magnitude spectrum of wind speed trace at hub position, $\mathbf{V}_x^{(10,10)}(t)$, the rotor speed, $\omega(t)$, and the flapwise blade root moment, $M_y^{\text{Bld}}(t)$. ... 14
Figure 4: Original vs reconstructed wind field after truncating its spectral representation to the first 16 $(n_i = n_j = 4)$ modes only. Left: original 2D turbulence box slice, middle: 2D discrete cosine transform coefficients (coefficients kept after the truncation are highlighted by a black square), right: reconstructed turbulence box slice after truncation. ... 16
Figure 5: Turbulent wind trace at hub position with an extreme operating gust between 455 and 465 s. ... 21
Figure 6: Surrogate model performance on the controller-driven blade pitch $\phi(t)$ on the normal turbulence model (NTM) validation dataset. Top: histogram of the root-mean-square error in degrees. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample. ... 25
Figure 7: Performance of the surrogate model of $\phi(t)$ on a selection of raw traces from the NTM validation set ... 26
Figure 8: Surrogate model performance on the controller-driven blade pitch $\phi(t)$ on the extreme turbulence model (ETM) validation dataset. Top: histogram of the root-mean-square error in degrees. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample. ... 27
Figure 9: Surrogate model performance on a selection of raw traces from the ETM validation set. .. 28
Figure 10: Surrogate model performance for the blade pitch $\phi(t)$ on the extreme operating gust (EOG) validation dataset. Top: histogram of the root-mean-square error in degrees. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample. ... 29
Figure 11: Surrogate model performance on a selection of raw traces from the EOG validation set.. 30
Figure 12: Surrogate model performance on the rotor speed $\omega(t)$ on the normal turbulence model (NTM) validation dataset. Top: histogram of the root-mean-square error in rpm. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample. ... 32
Figure 13: Performance of the surrogate model of $\omega(t)$ on a selection of raw traces from the NTM validation set. ... 33
Figure 14: Surrogate model performance on the rotor speed $\omega(t)$ on the extreme turbulence model (ETM) validation dataset. Top: histogram of the root-mean-square error in rpm. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample. ... 34
Figure 15: Performance of the surrogate model of $\omega(t)$ on a selection of raw traces from the ETM validation set. ... 35
Figure 16: Surrogate model performance on the rotor speed $\omega(t)$ on the extreme operating gust (EOG) validation dataset. Top: histogram of the root-mean-square error in rpm. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample. ... 36
Figure 17: Performance of the surrogate model of $\omega(t)$ on a selection of raw traces from the EOG validation set. ... 37
Figure 18: Surrogate model performance on the power production $P(t)$ on the normal turbulence model (NTM) validation dataset. Top left: scatter plot of true vs. surrogate total energy produced E. Top right: histogram of the energy production discrepancy in MWh. Bottom left: example of an inaccurate energy estimation sample. Bottom right: example of an accurate energy estimation sample. ... 39

Figure 19: Surrogate model performance vs seed-to-seed variability on the total energy produced E on the normal turbulence model (NTM) validation data set. 40
Figure 20: Performance of the surrogate model of the power output P on a selection of raw traces from the NTM validation set. 41
Figure 21: Surrogate model performance on the power production $P(t)$ on the extreme turbulence model (ETM) validation data set. Top left: scatter plot of true vs. surrogate total energy produced E. Top right: histogram of the energy production discrepancy in MWh. Bottom left: example of an inaccurate energy estimation sample. Bottom right: example of an accurate energy estimation sample. 42
Figure 22: Surrogate model performance vs seed-to-seed variability on the total energy produced E on the ETM validation dataset. 42
Figure 23: Performance of the surrogate model of the power output P on the ETM validation set. . 43
Figure 24: Surrogate model performance on the power production $P(t)$ on the extreme operating gust (EOG) validation data set. Top left: scatter plot of true vs. surrogate total energy produced E. Top right: histogram of the energy production discrepancy in MWh. Bottom left: example of an inaccurate energy estimation sample. Bottom right: example of an accurate energy estimation sample. 44
Figure 25: Surrogate model performance vs seed-to-seed variability on the total energy produced E on the EOG validation data set. 44
Figure 26: Performance of the surrogate model of the power output P on a selection of raw traces from the EOG validation data set. 46
Figure 27: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the normal turbulence model (NTM) validation dataset. Top left: scatter plot of true vs. surrogate DEL estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate DEL estimation sample. Bottom right: example of an accurate DEL estimation sample. 48
Figure 28: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the normal turbulence model (NTM) validation dataset. Top left: scatter plot of true vs. surrogate maximum load estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate maximum load estimation sample. Bottom right: example of an accurate maximum load estimation sample. 49
Figure 29: Surrogate model performance vs seed-to-seed variability on the flapwise blade root moment M_y^{Bld} on the normal turbulence model (NTM) validation dataset. Top panel: DEL. Bottom panel: maximum load. 50
Figure 30: Performance of the surrogate model of the flapwise blade root moment M_y^{Bld} on the normal turbulence model (NTM) on a selection of raw traces from the NTM validation set. 51
Figure 31: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the extreme turbulence model (ETM) validation dataset. Top left: scatter plot of true vs. surrogate DEL estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate DEL estimation sample. Bottom right: example of an accurate DEL estimation sample. 52
Figure 32: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the extreme turbulence model (ETM) validation dataset. Top left: scatter plot of true vs. surrogate maximum load estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate maximum load estimation sample. Bottom right: example of an accurate maximum load estimation sample. 53
Figure 33: Surrogate model performance vs seed-to-seed variability on the flapwise blade root moment M_y^{Bld} on the extreme turbulence model (ETM) validation dataset. Top panel: DEL. Bottom panel: maximum load. 54

Figure 34: Performance of the surrogate model of the flapwise blade root moment M_y^{Bld} on the extreme turbulence model (ETM) on a selection of raw traces from the ETM validation set........................... 55
Figure 35: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the extreme operating gust (EOG) validation dataset. Top left: scatter plot of true vs. surrogate DEL estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate DEL estimation sample. Bottom right: example of an accurate DEL estimation sample. 56
Figure 36: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the extreme operating gust (OEG) validation dataset. Top left: scatter plot of true vs. surrogate maximum load estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate maximum load estimation sample. Bottom right: example of an accurate maximum load estimation sample.. 57
Figure 37: Surrogate model performance vs seed-to-seed variability on the flapwise blade root moment M_y^{Bld} on the extreme operating gust (EOG) validation dataset. Top panel: DEL. Bottom panel: maximum load. 58
Figure 38: Performance of the surrogate model of the flapwise blade root moment M_y^{Bld} on the extreme operating gust (EOG) on a selection of raw traces from the EOG validation set. 60
Figure 39: Examples of models considering different uncertainty sources. Top left: deterministic model not considering any uncertainty, top right: model considering all uncertainty as aleatory, bottom left: model taking only epistemic (model) uncertainty into account, bottom right: a model that considers both epistemic and aleatory uncertainties. 66
Figure 40: Scatter plots of the environmental condition distributions – used as load simulation and surrogate model inputs. Top left: wind speed histogram. Top right: turbulence as function of wind speed. Bottom left: wind shear exponent vs. wind speed. Bottom right: Air density vs. wind speed............. 70
Figure 41: Overview of the outcomes from the aerodynamic load simulations. Scatter plots show normalized 10-minute load statistics vs. mean wind speed. 71
Figure 42: Comparison of epistemic (model) uncertainty and aleatory (realization-to-realization) uncertainty effects for extreme blade root flapwise bending moments. Left: scatter plots raw data including aleatory uncertainty and averages over multiple seeds (realizations) at each sample point; Center: histograms of the model uncertainty compared to the realization-to-realization uncertainty; Right: correlation between the model uncertainty and the realization-to-realization uncertainty. 73
Figure 43: Epistemic uncertainty distributions for three different training sample sizes 73
Figure 44: Illustration of a Feedforward Neural Network (FNN) with three hidden layers. 75
Figure 45: Comparison of probabilistic models on the prediction of blade root flapwise extreme bending moments. Top left: deterministic model not considering any uncertainty, top right: model considering all uncertainty as aleatory, bottom left: model taking only epistemic (model) uncertainty into account, bottom right: a model that considers both epistemic and aleatory uncertainties............ 76
Figure 46: Scatter plot of wind turbine blade flapwise minima predictions from a BNN model compared to the true load results for the same input conditions.. 77
Figure 47: Comparison of the probability densities of fatigue damage-equivalent load predictions from a BNN model and an ensemble of deterministic models, for six wind turbine load signals................ 78
Figure 48: Comparison of the tail probability distribution predictions by a BNN model and an ensemble of deterministic models, for four load channels.. 79
Figure 49: Comparison of the probability densities of the predictions from a BNN model and an ensemble of deterministic models, for the mean values of four wind turbine load signals. 80
Figure 50: Comparison of bivariate probability density prediction from a BNN model to the probability density of a reference data set. Left: probability density contours from load simulations; Right: probability density contours from BNN model.. 81

List of tables

Table 1: Properties of the NREL 5-MW baseline wind turbine 11

Table 2: List of the quantities of interest under consideration 11

Table 3: Surrogate modelling dependency chain 15

Table 4: Configuration parameters of the mNARX surrogate 17

Table 5: Summary of datasets (normal turbulence model, NTM; extreme turbulence model, ETM; extreme operating gust (EOG)) 21

Table 6: Composition of training and validation sets 22

Table 7: QoIs and aggregated quantities used for model validation 23

Table 8: Properties of mNARX surrogate for the blade pitch 24

Table 9: Properties of mNARX surrogate for the rotor speed 31

Table 10: Properties of mNARX surrogate for the power output $P(t)$ 38

Table 11: Summary of the surrogate model performance on the total energy produced E. The coefficients of determination R^2 corresponds to the scatter plots in **Figure 18**, **Figure 21** and **Figure 24**. The mean of the surrogate error $\mu_{\varepsilon_{surr}}$, the standard deviation of the surrogate error $\sigma_{\varepsilon_{surr}}$ and the standard deviation of the seed-to-seed uncertainty $\sigma_{\varepsilon_{seed}}$ correspond to the histograms in **Figure 19**, **Figure 22** and **Figure 25**. 45

Table 12: Properties of mNARX surrogate for the flapwise blade root moment $M_y^{\mathrm{Bld}}(t)$ 47

Table 13: Summary of the surrogate model performance on the damage equivalent load of the flapwise blade root moment M_y^{Bld}. The coefficients of determination R^2 corresponds to the scatter plots in **Figure 27**, **Figure 31** and **Figure 35**. The mean of the surrogate error $\mu_{\varepsilon_{surr}}$, the standard deviation of the surrogate error $\sigma_{\varepsilon_{surr}}$ and the standard deviation of the seed-to-seed uncertainty $\sigma_{\varepsilon_{seed}}$ correspond to the histograms in **Figure 29**, **Figure 33** and **Figure 37**. 59

Table 14: Summary of the surrogate model performance on the maximum value of the flapwise blade root moment M_y^{Bld}. The coefficients of determination R^2 corresponds to the scatter plots in **Figure 28**, **Figure 32** and **Figure 36**. The mean of the surrogate error $\mu_{\varepsilon_{surr}}$, the standard deviation of the surrogate error $\sigma_{\varepsilon_{surr}}$ and the standard deviation of the seed-to-seed uncertainty $\sigma_{\varepsilon_{seed}}$ correspond to the histograms in **Figure 29**, **Figure 33** and **Figure 37**. 59

Table 15: Computation costs associated with the preprocessing of a 10 minute turbulence box as performed during the construction and evaluation of the surrogate model for the flapwise blade root moment $M_y^{\mathrm{Bld}}(t)$. 61

Table 16: Computation costs to train the surrogate model for the pitch $\phi(t)$, rotor speed $\omega(t)$ and the flapwise blade root moment $M_y^{\mathrm{Bld}}(t)$. The configuration and number of training samples used corresponds to the low wind speed models listed in the **Table 8**, **Table 9**, and **Table 11**. 61

Table 17: Computation costs to evaluate the surrogate on 10^5 time steps. With a sampling frequency of 160 Hz this is equivalent to a 10 min simulation. 61

Table 18: Overview of FNN and BNN models 75

Table 19: KL divergence of model prediction probability distributions with respect to load simulation data distributions. Green shading indicates that one of the two models has smaller KL divergence for the particular signal, while yellow shading shows cases with less than 5% difference between the two models. 80

1. Executive Summary

Central to the HIPERWIND project is the inclusion of probabilistic design concepts into the design of offshore wind turbine models. These can require large amounts of aero-servo-elastic (ASE) simulations, thus incurring significant computational loads, often intractable with currently available computational resources. To overcome these limitations, WP4.1 is tasked with the design and development of novel surrogate modelling approaches that can deal with the extremely high input- and output-dimensionalities typical of ASE simulations, while retaining acceptable accuracy.

WP4.1 develops two complementary surrogate modelling approaches that can handle and accelerate high-dimensional ASE simulations:

- A full wind-box-input to time-series-output surrogate based on the combination of dimensionality reduction and autoregressive modelling developed by ETH Zurich
- A stochastic simulator based on Bayesian Neural Networks, predicting time-aggregated load quantities while taking into account model and realization-to-realization uncertainty – developed by DTU.

The performance of each surrogate model is benchmarked on an exhaustive database of simulations based on the load cases identified in D1.2, for a number of output quantities of interest (QoIs) and different error measures.

The surrogates presented provide an adequate to excellent approximation of their ASE counterparts, with computational savings of several orders of magnitude, and acceptable modelling errors. As a result, these approaches represent a substantial new asset in the probabilistic design toolchain as they enable large-scale and uncertainty-aware wind turbine design studies.

2. Introduction

2.1. General context

The HIPERWIND project (HIghly advanced Probabilistic design and Enhanced Reliability methods for high-value, cost efficient offshore WIND) aims at reducing the levelized cost of energy (LCOE) through a three-pronged approach:

- by using measured data and advanced physics-based models to improve the probabilistic modelling of environmental conditions and flow on offshore wind-farms
- by using these probabilistic models in conjunction with advanced computational models to reduce uncertainty on turbine structure, component reliability and maintenance requirements
- by deploying both of these, explore turbine designs that both reduce LCOE and increase the system value of wind.

The HIPERWind project comprises five *Specific Objectives*, summarised in **Figure 1**, each outlining a clearly defined work–package (WP) within the project.

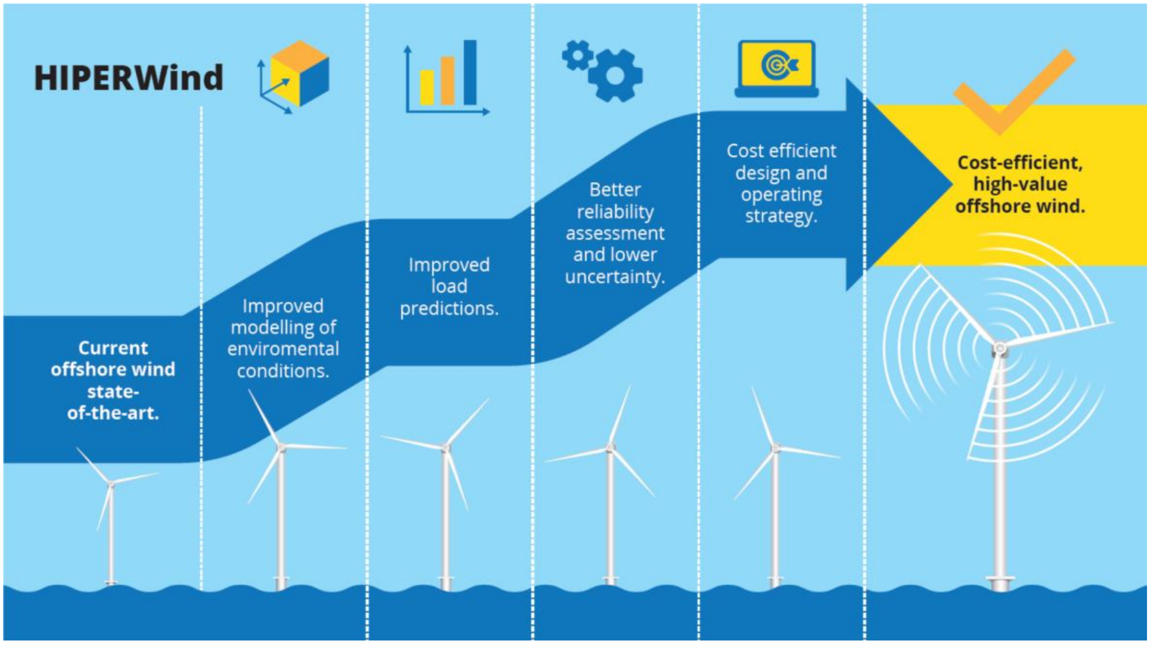
HIPERWind
Current offshore wind state-of-the-art.
Improved modelling of enviromental conditions.
Improved load predictions.
Better reliability assessment and lower uncertainty.
Cost efficient design and operating strategy.
Cost-efficient, high-value offshore wind.

Figure 1: the HIPERWind work-package structure.

- WP2 focuses on the improved probabilistic modelling of the environmental conditions that affect offshore wind farm design and operation, reducing uncertainty in environmental exposure.
- WP3 develops novel load assessment methods for large offshore fixed bottom- and floating-wind turbines, reducing uncertainty in load assessment.
- WP4 develops an efficient framework for the consistent probabilistic assessment of the reliability of offshore turbines, hence enabling full probabilistic design approaches.
- WP5 deals with the longer term modelling of wind turbine degradation, enabling the reduction of expected maintenance costs.
- WP6 provides recommendations based on the outcome of the previous WPs to significantly reduce LCOE and improve the market value of offshore wind energy.

We focus here on WP4, and on the challenges related to the probabilistic reliability assessment of complex offshore wind turbines.

2.2. Scope of the work and objectives

This work is performed in the context of WP4, Task 4.1: *Surrogate models with high-dimensional input and outputs*, led by ETH Zurich, with the participation of DTU. Deliverable contents and measures of success are defined in the following extract from the official SMART deliverable 4.1:

Specific targets:

- Develop a Kriging-based surrogate model which receives turbulent wind inputs and provides load time series outputs including model uncertainty estimates.
- Develop a neural-network based surrogate model with 10-minute load statistics as output, trained with Bayesian inference so that the model parameter uncertainty can be estimated.
- Test and report the efficiency of the surrogate models developed, in terms of their performance on predicting the probability distributions of fatigue damage and extreme loads for selected load cases provided from WP1

Measure of success:

- Surrogate modelling approaches are developed and have at least one order of magnitude better computational efficiency than the original aeroelastic modelling tool
- Surrogate model uncertainty is not greater than the turbulence-induced statistical uncertainty (seed-to-seed uncertainty)

Participant actions:

- Partner A (ETH, LEAD): Develop and evaluate the performance of a Kriging-based surrogate model. Collect results for deliverable report.
- Partner B (DTU): Develop and evaluate the performance of a Neural-Network based surrogate model for predicting loads and power production on major structures in fixed base wind farms. Provide load simulation and turbulence input time series to ETH for surrogate model training purposes.

Because of the independent nature of the work performed by the two partners ETH and DTU, this document includes the report from the two independently developed surrogate modelling techniques as two separate sections, starting from the contribution from ETH. Nevertheless, each section will follow a similar structure, where the problem is first stated, including the description of the reference ASE model considered, the accurate description of the input/output quantities considered, performance metrics, etc., followed by a detailed validation and benchmarking of the proposed methodology.

3. Surrogate modelling of high-dimensional input/output time-series (ETH)

3.1. Problem statement and definitions

From a purely mathematical/abstract viewpoint, every aero-servo-elastic simulation of a specific wind turbine subject to a wind excitation can be abstracted as a generic function (or model $\mathcal{M}$) of the form:

$$\mathbf{y}(t) = \mathcal{M}(\mathbf{x}(\leq t)) \tag{1}$$

where $\mathbf{y}(t) = \left\{ y^1(t), \cdots, y^{N_Y}(t) \right\}$ represents a set of time-dependent output quantities of interest (QoIs, *e.g.* rotor speed, power output, loads, etc.), and $\mathbf{x}(t) = \left\{ x^1(t), \cdots, x^{N_X}(t) \right\}$ is a set of input time-series that may represent both the environmental conditions (*e.g.* wind speed at different coordinates, wave height, etc.), and the state of the wind-turbine itself (*e.g.* orientation, physical properties, etc). The notation $\bullet(\leq t)$ indicates that the time-dependent variable $\bullet$ is considered for times smaller or equal than t. In this representation, the time-dependent set of model responses $\mathbf{y}(t)$ is *deterministic*, *i.e.* evaluating the model $\mathcal{M}$ on the same set of input time series $\mathbf{x}^*(\leq t)$ will result in an identical response $\mathbf{y}^*(t)$.

Due to their inherently stochastic nature, it is often necessary to consider several realizations of the random environmental conditions. To reduce the associated computational burden, we introduce here the concept of surrogate models. Following the same notation as in Eq. (1), we can describe a surrogate model as a mathematical function $\mathcal{M}$ such that:

$$\mathbf{y}(t) = \mathcal{M}(\mathbf{x}(\leq t)) \approx \mathcal{M}(\mathbf{x}(\leq t)) \tag{2}$$

Surrogate models are cheap-to-evaluate functions, that are trained on a training set of time series of size N_{ED} known as the *experimental design* $\mathcal{X} = \{\mathbf{x}^{(1)}(t), \cdots, \mathbf{x}^{(N_{\mathrm{ED}})}(t)\}$ so as to maximize accuracy of the approximation in Eq. (2) according to some suitable error measure.

While surrogate models are very common in a number of fields of applied science and engineering, the construction of surrogates for complex ASE simulations with high dimensional inputs and outputs is as of today a largely unexplored topic.

Our goal is to present here a novel general class of surrogate models that can achieve the approximation in Eq. (2) under the following conditions:

- $\mathbf{x}(t) = \{\mathbf{V}(t), \mathbf{z}(t)\}$ includes both the wind excitation to which a wind turbine is exposed ($\mathbf{V}(t)$), and an additional set of state variables ($\mathbf{z}(t)$) that can affect the turbine (*e.g.* wave height in offshore turbines, pitch angle if available, etc.). In practical computations, $\mathbf{V}(t)$ is a set of three three-dimensional turbulence boxes discretized both in space and in time, where each turbulence box corresponds to one of the wind speed components (longitudinal, transverse and upwards) with respect to the turbine. Typically, each wind field is discretized into $\mathcal{O}(10^{2-3})$ nodes along the spatial coordinates, and $\mathcal{O}(10^{4-5})$ along the time axis for a simulation length of 10 minutes.

- $\mathbf{y}(t)$ represents a vector of output quantities of interest, each discretized along the time similarly to the input coordinates $\mathbf{x}(t)$.
- The size of the training set N_{ED} is maintained manageable in the order of $\mathrm{O}(10^{2-3})$ full ASE simulations.
- The surrogate is entirely non-intrusive and data-driven, *i.e.* no assumptions are made nor knowledge is required about the specific modelling strategy employed to generate the training set.

The last condition adds an important value to the development of this deliverable. In addition to the obvious direct benefits coming from the significant reduction in required computational resources, the availability of a surrogate model can enable different beneficiaries of the HIPERWIND consortium to run calculations without the need to exchange possibly confidential and/or sensitive information, such as computational models.

3.2. Validation strategy

Regardless of the specific technology employed to achieve a surrogate of the form of Eq. (2), its performance needs to be tested on a validation pool of simulations that thoroughly explores the expected environmental conditions considered during the surrogate lifetime. To this end we performed an extensive benchmark study on a model of a standardized wind turbine, with a well-known stack of environmental modelling and ASE simulation technologies. A detailed description of the specific model is given in Section3.3.

We validate the performance of the surrogates using extensive data sets with a total of about 2,000 simulations from different climate scenarios (normal turbulence model (NTM), extreme turbulence model (ETM), extreme operating gust (EOG)), as well as with different random seeds for the generation of the input wind boxes. A detailed description of all the validation sets employed in this study is provided in Section 3.5.1.

For each of the considered wind climate scenarios, a number of different output quantities of interest is benchmarked separately, in fact amounting to a different surrogate model for each. They are described in detail in Section 3.3.2.

Validation is performed at different levels, depending on the specific quantity (e.g. state variables or loads) and wind climate scenario considered, in Section 3.5.3. Whenever possible, a comparison between surrogate-model-induced uncertainty vs. seed-to-seed uncertainty is also performed. Several different error measures, as well as illustrative plots are described in detail in Section 3.5.2.

3.3. Computational model: the NREL 5MW reference turbine

3.3.1. Description of the wind turbine

Due to the unavailability of an offshore turbine model and corresponding simulator codes, as well as the need of a clear probabilistic description of wave conditions, the surrogate model documented in this report was developed and validated on the well-known NREL 5-MW baseline wind turbine (Jonkman, et al., 2009). The turbine model was used with NREL's ROSCO (Reference Open-Source Controller (NREL, 2021) and simulated in the open-source ASE simulator OpenFAST (NREL, 2021) as described in Section 3.3.4.

Since the surrogate modelling approach does not depend strictly on the wind turbine characteristics, but rather aims to learn the dynamics of the turbine from simulation data, it is expected that its performance on offshore turbines would be comparable, as long as sufficient information about the wave loads is provided. A summary of the NREL 5MW reference turbine properties is given in **Table 1**.

Table 1: Properties of the NREL 5-MW baseline wind turbine

Rating	5 MW
Rotor Orientation, Configuration	Upwind, 3 Blades
Control	Variable Speed, Collective Pitch
Drivetrain	High Speed, Multiple-Stage Gearbox
Rotor, Hub Diameter	126 m, 3 m
Hub Height	90 m
Cut-In, Rated, Cut-Out Wind Speed	3 m/s, 11.4 m/s, 25 m/s
Cut-In, Rated Rotor Speed	6.9 rpm, 12.1 rpm

3.3.2. Output quantities of interest

The quantities of interest (QoIs) for the development and validation of the surrogate modelling approach are selected based on standard wind turbine design practices, as well as on the needs of ultimate and fatigue limit state estimation. A complete list of the QoIs included, as well as the acronyms used in this report, are given in **Table 2**.

Table 2: List of the quantities of interest under consideration

QoI	**Acronym**
Blade pitch	$\phi(t)$
Rotor speed	$\omega(t)$
Electrical power	$P(t)$
Flapwise blade root bending moment	$M_y^{\text{Bld}}(t)$

Note that we explicitly highlight the time-dependent behavior of the quantities of interest, as our goal is to provide a surrogate model of the transient response of each quantity.

3.3.3. Wind box generation and discretization (TurbSim)

Turbulent wind was generated using the open-source stochastic turbulence simulator TurbSim (Jonkman, 2009), which constructs random turbulent wind fields from a given spectral and spatial coherence model. Following the IEC61400-1 design standard (IEC, 2019), we used the Kaimal spectrum in conjunction with an exponential coherence model. This provides a temporally coherent wind in the longitudinal ($\mathbf{V}_x(t)$), transverse ($\mathbf{V}_y(t)$), and vertical ($\mathbf{V}_z(t)$) directions. Spatial coherence, however, is only enforced in the longitudinal wind direction. TurbSim does not directly include wind shear in the turbulence generation, but it superimposes it subsequently directly on the generated turbulence box.

A turbulence box by TurbSim consists of the three 3-dimensional wind speed matrices, $\mathbf{V}_x$, $\mathbf{V}_y$ and $\mathbf{V}_z$. Each matrix is discretized along the horizontal and vertical spatial dimensions with the indices κ and ℓ, as well as along the temporal dimension t. Each 2-dimensional slice of a wind speed matrix at a given time discrete time $t = a\delta t, \quad a \in □$, e.g. $\mathbf{V}_x(t)$, is discretized into a grid 19 x 19 elements $V^{\kappa,\ell}(t)$, resulting in 361 data points per time step. The turbulence box is discretized with a sampling frequency of 20 Hz, which corresponds to a time step of 0.05 s. For a 10-minute wind box, this results in $\mathrm{O}(10^6)$ data points for each wind speed matrix and $\mathrm{O}(10^7)$ data points for the full wind box.

3.3.4. AES modelling (OpenFast)

To perform AES simulations, we adopted the publicly available open-source aero-servo elastic simulator OpenFAST (NREL, 2021) to train and validate the proposed surrogate modelling approach. OpenFAST uses InflowWind (Platt, et al., 2016), a module for processing wind inflow data to obtain local wind speeds. The local wind speeds are obtained using Taylor's frozen turbulence hypothesis and spatial and temporal interpolation. The time step of the QoIs output by OpenFAST is $dt = 0.00625\mathrm{s}$, which corresponds to a sampling frequency of 160 Hz. In other words, the turbulence box produced by TurbSim (see previous section) is re-interpolated to a higher sampling frequency of 160 Hz for numerical stability reasons. Spatial discretization is unchanged.

3.4. Surrogate modelling strategy

3.4.1. Overview

We propose a novel surrogate modelling method that takes advantage of the space- and time- coherence of the high dimensional input wind box, and the output time-dependent QoIs. While we re-use existing technology, such as the spectral representation of spatial variability and autoregressive models, the use of these techniques in a surrogate modeling chain to predict the response to entirely out-of-sample turbulence-boxes is completely original.

The overall surrogate modelling algorithm can be in short described as follows (see **Figure 2**):

1. Reduce the spatial dimensionality of the input turbulence boxes by projecting them on a low-dimensional space $\xi(t)$ that is suitable for surrogate modelling. This can be achieved *e.g.* through principal component analysis (PCA) or spectral decomposition (discrete cosine transform) of each spatially coherent 2D timeframe of $\mathbf{V}(t)$. Additional time-dependent time-series related to the environmental/wind conditions (e.g. wave height) can also be included in $\xi(t)$.

2. Train an auto-regressive with exogenous input model (NARX) on the low dimensional features from an available training set (*experimental design*) for those QoIs that are only dependent on the $\xi(t)$ (e.g. pitch angle and in some cases rotor speed).

3. Predict any intermediate quantities that are needed for the emulation of the remaining quantities of interest: pitch angle, blade speed, blade position, etc. (see Section 3.3.2). Combine this new set of additional quantities with $\xi(t)$ into a *manifold* $\zeta(t)$ of manageable dimension.

4. Train a NARX model of each QoI on $\zeta(t)$.

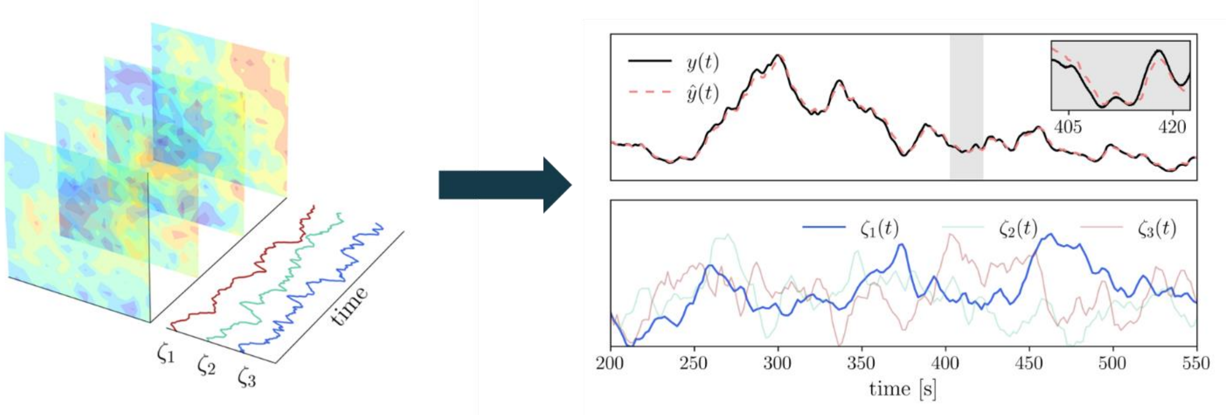

Figure 2: Graphical representation of the proposed mNARX surrogate modelling algorithm.

This approach results in a family of NARX surrogates that can be efficiently run sequentially to provide out-of-sample predictions on completely new (unseen) turbulence-boxes. Because at the core of this strategy lies the construction of a low dimensional manifold suitable for NARX modelling, we name this method mNARX. An interesting aspect of this approach is that the actual manifold used for each output QoI may contain additional information w.r.t. that contained in the input turbulence boxes, due to the highly non-linear transform of the raw input features $\mathbf{x}(t)$ that the auxiliary NARX surrogates imply. This is particularly important in the context of a NARX, which in general fails to perform when the output QoIs contain higher frequencies than the exogenous inputs.

In the following sections, each of the steps necessary to construct and train an mNARX surrogate is described in more detail.

3.4.2. mNARX: a chain of surrogate models

A crucial aspect of the proposed method is the construction of a proper input manifold for each quantity of interest. The manifold itself is obtained as a non-linear transform of the input time-series through a sequence of autoregressive models of intermediate quantities. The manifold itself has in principle lower dimensionality than the raw input turbulence box, and is composed of a number of different *features*. These features may both include a subset of the raw inputs (e.g. their spectral components), but they may also include additional properties that are not part of it, e.g. higher frequency harmonics or other non-linear transforms.

Albeit at first sight counter-intuitive, this is a direct consequence of the physics of a wind turbine system: the frequency content of the input turbulence boxes naturally affects that of the output QoIs from the wind turbine. However, the wind turbine consists of several moving parts and a control system, which may result in cyclic behaviour at frequencies much higher (and lower) than those typical of wind variation. Perhaps the simplest example of this behaviour would be that of a turbine under a constant breeze: while the turbulence box would remain fixed in both time and space, the loads on the wind turbine elements would exhibit a cyclic behaviour at frequencies related to the rotor speed, as well as to the number of blades on the turbine (e.g. due to tower shadowing effects from the blades). A more relevant example is provided in **Figure 3**, where we show the magnitude spectrum of a single turbulence box for the central node $\mathbf{V}_x^{(10,10)}(t)$ (see Section 3.3.3 for more details).

As expected from a Kaimal model, input wind power decays rapidly with frequency, and most of its energy is contained at frequencies well below 1 Hz. Similarly, the power spectrum of the rotor speed ω decays fast, but shows a peak at about 4 Hz. This behaviour is even more evident for $M_y^{\mathrm{Bld}}(t)$, the power spectrum of which remains essentially constant between 0.2 and 1 Hz, and exhibits a strong peak at about 2 Hz.

The input manifold $\zeta(t)$ of the mNARX surrogate for each QoI is therefore a combination of the turbulence box parameters $\mathbf{V}(t)$, and other derived quantities, typically other QoIs that have been surrogated at an earlier stage. This *hierarchical* construction results therefore in a chain (or sequence) of surrogates that may depend on each other. An overview of the cross-dependencies between the surrogates of various QoIs is provided in **Table 3**.

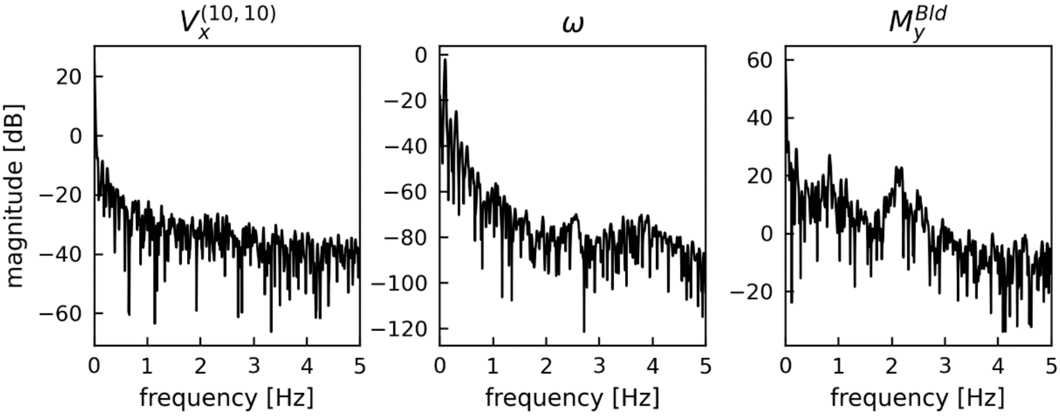

Figure 3: Magnitude spectrum of wind speed trace at hub position, $\mathbf{V}_x^{(10,10)}(t)$, the rotor speed, $\omega(t)$, and the flapwise blade root moment, $M_y^{\mathrm{Bld}}(t)$.

Table 3: Surrogate modelling dependency chain

QoI (Symbol)	**Input manifold components**
Pitch angle ($\phi(t)$)	$\mathbf{V}(t)$
Rotor speed ($\omega(t)$)	$\mathbf{V}(t)$, $\phi(t)$
Azimuth ($\alpha_i(t)$)	$\omega(t)$
Azimuth harmonics[1] ($\alpha_i^k(t)$)	$\alpha_i(t)$
Electrical power ($P(t)$)	$\mathbf{V}(t)$, $\phi(t)$
Blade root flapwise moment ($M_y^{\mathrm{Bld}}(t)$)	$\mathbf{V}(t)$, $\phi(t)$, $\alpha_i^k(t)$

3.4.3. Spatial dependence: spectral compression

According to Eq. (2), the chain of surrogate models $\mathbf{y}(t) = \mathcal{M}(\mathbf{x}(\le t))$ is a function of the (in general time-dependent) environmental and structural parameters contained in $\mathbf{x}(t)$. As described in Section 3.4.2, however, each QoI $\hat{y}_i(t)$ may depend on a quantity-specific set of input parameters (**Table 3**). For notational simplicity, we will now focus on a single output QoI at a time, making their dependence on the input parameters explicit as follows:

$$\hat{y}_i(t) = \mathcal{M}_i(\mathbf{x}(\le t), \hat{\mathbf{y}}_{<i}(\le t)), \qquad (3)$$

where $\mathbf{x}(t)$ is the input time series that describes the time dependent environmental conditions and state variables, and $\mathbf{y}_{<i}(t)$ is a set of QoIs that have been surrogated prior to $\hat{y}_i(t)$ as described in Section 3.4.2.

Because of its high dimensionality, we focus in this section on the $\mathbf{V}(t)$ component of $\mathbf{x}(t)$, or the input turbulence box (see Section 3.3.3). As described in Section 3.3.3, it is typically discretized as three three-dimensional matrices $\mathbf{V}_x(t), \mathbf{V}_y(t)$ and $\mathbf{V}_z(t)$, each with a dimensionality in the order of $\mathrm{O}(10^{2-3})$. Among the three components, by far the dominant one for most QoIs is the longitudinal component
$\mathbf{V}_x(t)$, which is also the only one that exhibits spatial coherence by construction. Because with our specific model configuration (Section 3.3) wind is aligned with the turbine and the other two components are much less significant and have no coherence (white noise), we discard them from the pool of input features. In the general case, all components can in principle be considered.

To be able to use advanced surrogate modelling techniques, and NARX models in particular, we need to reduce the dimensionality of $\mathbf{V}_x(t)$, and to achieve this we capitalize on its spatial coherence (smoothness). There are many approaches available to this end (e.g. principal component analysis, wavelet transforms, etc.), and for this application of mNARX we choose to project $\mathbf{V}_x(t)$ on a truncated discrete cosine basis. In other words, at any timestep t in the discretized turbulence box, we project the

[1] E.g. $\sin(k \cdot \alpha_i(t))$

two-dimensional matrix of longitudinal wind speeds $\mathbf{V}_x(t)$ into its truncated 2D discrete cosine transform (DCT) coefficients $\boldsymbol{\xi}(t)$:

$$V_x^{\kappa\ell}(t) = \sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\xi_{ij}(t)\cos\left[\frac{\pi}{n_i}\left(\kappa+\frac{1}{2}\right)i\right]\cos\left[\frac{\pi}{n_j}\left(\ell+\frac{1}{2}\right)j\right], \tag{4}$$

where κ and ℓ are the indices of the discretized turbulence box along the x and y directions (see Section 3.3.3), $n_i \le n_x$ and $n_j \le n_y$ are the number of discretization nodes along the x and y directions, respectively, and $\xi_{ij}(t) \in □$ are the DCT coefficients. The invertible DCT transform allows us to map $\mathbf{V}_x$ to $\boldsymbol{\xi}$ and vice versa.

Due to the physics of the system, it is expected that only the low spatial frequency modes of $\boldsymbol{\xi}(t)$ in Eq. (4) directly affect the wind turbine response. The expansion is therefore truncated, typically by setting $n_i = n_j = 4$, hence yielding a total number of coefficients in the order $\mathrm{O}(10)$, down from the original $\mathrm{O}(10^{2-3})$ of $\mathbf{V}_x$. A visual comparison of a raw 2D wind velocity profile and the effects of this low-pass filtering compression is provided in **Figure 4**.

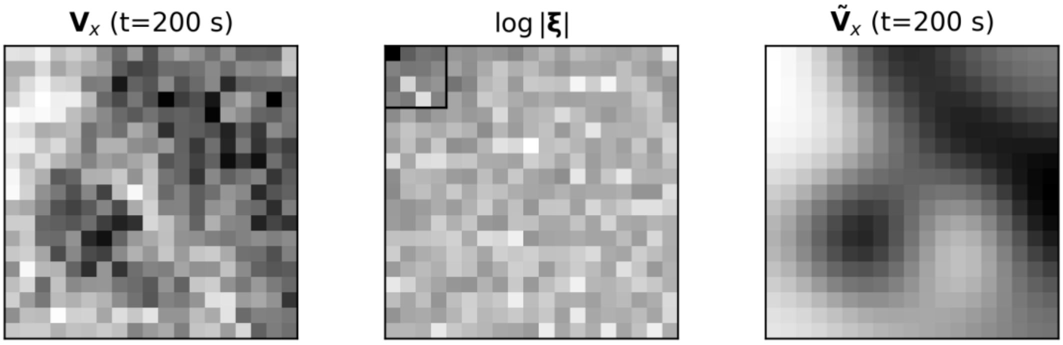

Figure 4: Original vs reconstructed wind field after truncating its spectral representation to the first 16 $(n_i = n_j = 4)$ modes only. Left: original 2D turbulence box slice, middle: 2D discrete cosine transform coefficients (coefficients kept after the truncation are highlighted by a black square), right: reconstructed turbulence box slice after truncation.

After this compression step of the spatially dependent $\mathbf{V}_x(t)$, the total number of the input manifold features is in the order of 4-25, depending on the specific QoI being surrogated, and can be handled by standard auto-regressive modelling methods. More explicitly, the full input manifold can be written as:

$$\boldsymbol{\zeta}(t) = \left\{\boldsymbol{\xi}(t), \mathbf{z}(t), \mathbf{y}_{<i}(t)\right\}. \tag{5}$$

For notational simplicity, we omit here and in the following an explicit reference to the QoI index i on the input manifold $\boldsymbol{\zeta}(t)$, but it is clear from Eq. (5) that it is actually on the specific QoI being surrogated: in general different QoIs will be constructed on different manifolds (see also Section 3.4.2).

3.4.4. Time dependence: NARX

Once the time-dependent input manifold $\zeta(t)$ is constructed, it can be used within a standard NARX modelling framework. In short, we consider an autoregressive model of the following form:

$$\hat{y}_i(t+\delta t) = \mathrm{M}\,(\hat{y}_i(\leq t), \zeta(\leq t+\delta t)) \tag{6}$$

where δt is a small time-increment, which corresponds to the time discretization step in our case. In other words, given an incremental timestep δt, the output quantity of interest $\hat{y}_i(t+\delta t)$ is predicted based on its own value predicted until time t (auto-regressive component), and the input manifold $\zeta(t)$ until time $t+\delta t$ (exogenous input component).

In practical NARX applications, the system is assumed to have finite memory, which means that instead of the full set of autoregressive time-series, only a small subset is considered. This subset is often parameterized in terms of the number of *lags* considered for each quantity. Each lag is normally an integer multiple of the time discretization interval δt, or $\ell_i = k\delta t, \quad k \in \mathbb{N}$, which leads us to rewrite Eq. (6) explicitly as:

$$\hat{y}_i(t+\delta t) = \mathrm{M}\left(\hat{y}_i(t), \hat{y}_i(t-\ell_1^y), \cdots, \hat{y}_i(t-\ell_{n_y}^y), \zeta(t+\delta t), \zeta(t), \zeta(t-\ell_1^\zeta), \cdots, \zeta(t-\ell_{n_\zeta}^\zeta)\right), \tag{7}$$

where ℓ^y and ℓ^ζ are the lags along y_i and ζ (to avoid notational overcrowding we drop the i indices on all instances of ℓ, but in principle they are QoI-dependent), and n_y and n_ζ are the total number of lags considered for y_i and ζ, respectively.

The formulation in Eq. (7) is very general, and it allows for significant freedom: lags can be of different lengths, and they can be different between the autoregressive and the exogenous input components. This is important, because of the multi-modal frequency response spectrum of wind turbine simulations, which requires both short-term lags to reproduce high-frequency behaviour, and long-term lags for its corresponding low-frequency counterpart.

Numerous approaches are available in the literature to construct autoregressive models in the form of Eq. (7), and in this work we use a classic polynomial model available as described in (Billings, 2013, pp. 35-36). For the sake of simplicity, we operate on both input and output time series at the same sampling frequency as the AES simulator of $\nu = 160$ Hz (for more detail, see Section 3.3.4). In this setting, the parameters of the mNARX surrogate are summarized in the following table:

Table 4: Configuration parameters of the mNARX surrogate

Parameter	**Description**
Polynomial degree d	Maximum polynomial order to be considered in the NARX surrogate
Maximum interaction r	Maximum interaction order considered in the polynomial regressor
$\left\{\ell_1^y, \cdots, \ell_{n_y}^y\right\}$	List of the auto-regressive lags ℓ^y
$\left\{\ell_1^\zeta, \cdots, \ell_{n_\zeta}^\zeta\right\}$	List of the exogenous input lags ℓ^ζ
$\zeta(t)$	Definition of the input manifold, including the auxiliary QoIs and the frequency truncation strategy for $\mathbf{V}_x(t)$ (see **Table 3**).

The final complexity of the mNARX model depends of course on the parameters in **Table 4**, because they uniquely determine the number of unknown coefficients that must be inferred from the training set.

3.4.5. Wind regimes and classification (low/high wind speed regimes)

A wind turbine has several mechanisms to operate optimally under a variety of wind conditions, one of which is the controller-driven variable pitch angle. The pitch angle drastically affects the turbine's dynamic response to the wind. This is accounted for by predicting the pitch angle as an intermediate quantity and using it as an additional exogenous input for predict other variables, such as loads. However, it was found that the performance of the proposed surrogate modelling approach is significantly improved when the problem is divided into low and high wind speed regimes. We propose a simple classification based on the rated wind speed of the turbine and the average wind speed of a wind field at hub height V_{hub}. The low wind speed regime includes wind fields with V_{hub} less than the rated wind speed of 11.4 m/s and the high wind speed regime includes wind fields with V_{hub} greater than or equal to the rated wind speed. For either regime a separate surrogate is constructed. New turbulence boxes are first classified according to this simple property and then evaluated with the corresponding surrogate.

3.4.6. Training strategy

With Eq. (7), we introduced a general formulation of a NARX model in which at each future time step $t+\delta t$, the predicted output $\hat{y}_i(t+\delta t)$ of the NARX model is a function of an exogenous input $\boldsymbol{\zeta}(\leq t+\delta t)$ and past predictions $\hat{y}_i(\leq t)$. Subsequently, we refer to the full input vector at a time step τ, consisting of both the exogenous input and the past prediction, as $\boldsymbol{\varphi}_\tau^{(j)}$, where the superscript j refers to the $j-$th simulation in the experimental design:

$$\hat{y}_i(t+\delta t) = \mathrm{M}\underbrace{\left(\hat{y}_i(t), \hat{y}_i(t-\ell_1^y), \cdots, \hat{y}_i(t-\ell_{n_y}^y), \boldsymbol{\zeta}(t+\delta t), \boldsymbol{\zeta}(t), \boldsymbol{\zeta}(t-\ell_1^\zeta), \cdots, \boldsymbol{\zeta}(t-\ell_{n_\zeta}^\zeta)\right)}_{\boldsymbol{\varphi}_\tau^{(j)}}. \tag{8}$$

For the same time step τ and simulation j, we write the true output as $y_\tau^{(j)} = \left\{y_i^{(j)}(t+\delta t)\right\}$. Note that we omit the subscript i hereafter to avoid overcrowding of the notation.

We then model the approximate response $\mathrm{M}(\cdots)$ as a multivariate polynomial on the components of the input vector $\boldsymbol{\varphi}_\tau$, whose coefficients are to be determined from the available training set.

Let $\boldsymbol{\alpha}$ be an integer multi-index belonging to the set:

$$\boldsymbol{\alpha} \in \mathrm{A}^{M,d,r} \quad : \left\{ \left(\left\| \alpha \right\|_1 \le d \right) \cap \left(\left\| \alpha \right\|_0 \le r \right) \right\}, \tag{9}$$

where M is the number of components of the input vector $\boldsymbol{\varphi}_\tau$, d is the maximum degree of the multivariate polynomial and r is the maximum interaction order, while $\left\| \bullet \right\|_p$ is the $p-$norm operator. For every multi-index $\boldsymbol{\alpha}$ we then define a multivariate polynomial $\mathrm{P}_{\boldsymbol{\alpha}}(\boldsymbol{\varphi}_{\boldsymbol{\tau}})$ as:

$$\mathrm{P}_{\boldsymbol{\alpha}}(\boldsymbol{\varphi}_\tau) = \prod_{i=1}^{M} (\varphi_{\tau,i})^{\alpha_i} . \tag{10}$$

Given Eq. (10), we can explicitly rewrite Eq. (8) as:

$$\mathrm{M}\,(\boldsymbol{\varphi}_\tau) = \sum_{\boldsymbol{\alpha} \in \mathrm{A}^{M,d,r}} c_{\boldsymbol{\alpha}} \mathrm{P}_{\boldsymbol{\alpha}}(\boldsymbol{\varphi}_\tau), \tag{11}$$

Where the $c_{\boldsymbol{\alpha}}$ are a set of real polynomial coefficients, which for convenience we gather in a vector $\mathbf{c} = \left\{ c_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathrm{A}^{M,d,r} \right\}^{\text{ú}}$. From Eq. (11) it becomes clear that training the surrogate corresponds to determining the matrix of polynomial coefficients $\mathbf{c}$, which we achieve by ordinary-least-squares (OLS) regression. We construct a regression matrix $\boldsymbol{\psi}$, which contains the samples from all time steps and of the full experimental design:

$$\boldsymbol{\psi} = \left\{ \mathrm{P}_{\tau_0}^{(1)^{\text{ú}}}, \cdots, \mathrm{P}_{\tau_{\max}}^{(1)^{\text{ú}}}, \cdots, \mathrm{P}_{\tau_0}^{(N_{ED})^{\text{ú}}}, \cdots, \mathrm{P}_{\tau_{\max}}^{(N_{ED})^{\text{ú}}} \right\}^{\text{ú}} . \tag{12}$$

Similarly, we also gather the output vector $\mathbf{y}$ as:

$$\mathbf{y} = \left\{ y_{\tau_0}^{(1)}, \cdots, y_{\tau_{\max}}^{(1)}, \cdots, y_{\tau_0}^{(N_{ED})}, \cdots, y_{\tau_{\max}}^{(N_{ED})} \right\}^{\text{ú}} . \tag{13}$$

Using $\boldsymbol{\psi}$ and $\mathbf{y}$, the regression problem can then be written as a linear system $\mathbf{y} = \boldsymbol{\psi}\mathbf{c}$, where $\mathbf{c}$ is a coefficient vector. We solve this linear regression problem by least squares minimization analytically simply as:

$$\mathbf{c} = \left(\boldsymbol{\psi}^{\text{ú}} \boldsymbol{\psi} \right)^{-1} \boldsymbol{\psi}^{\text{ú}} \mathbf{y} . \tag{14}$$

In our problem, we have $N_{ED} \approx \mathrm{O}(10^2)$ simulations in the experimental design, with $\mathrm{O}(10^5)$ time steps per simulation, making it impractical to solve Eq. (14). Therefore, we use only a random subset of the input matrix $\boldsymbol{\psi}_S \subset_R \boldsymbol{\psi}$ and the corresponding output vector $\mathbf{y}_S \subset_R \mathbf{y}$. The size of the random subset N_S depends on the number of coefficients, and we choose $N_S \approx 20 \times |\mathbf{c}|$, to ensure proper conditioning in the matrix inversion in Eq. (14).

3.5. Surrogate model performance

This section provides an extensive analysis of the performance of the surrogate modelling chain described in Section 3.4 on several validation sets that span varying climate conditions, from nominal operation to wind gusts, and provides an estimate of the modelling error introduced by the surrogate at different levels.

3.5.1. Training and validation sets

To benchmark the proposed surrogate modelling strategy just introduced, we consider three different wind condition scenarios identified in the IEC 61400-1 design standard: normal turbulence model (NTM), extreme turbulence model (ETM), and extreme operating gust (EOG). The following sections describe the construction of the turbulent wind datasets for each scenario, and list the number of random input vectors and random seeds used for the performed simulations. An overview of all datasets used can be found in **Table 5**.

Normal turbulence model

The wind climate parameters were chosen in accordance with the IEC 61400-1 design standard. Consequently, two random variables are defined, namely the mean wind speed at hub height V_{hub}, which follows a Rayleigh distribution, and the turbulence standard deviation σ_1, which follows a Weibull distribution and is conditioned on V_{hub}. Sets of random variables with V_{hub} below cut-in wind speed (3 m/s) or above cut-out wind speed (25 m/s) were discarded.

Two datasets were generated for the NTM scenario, subsequently referred to as the standard dataset (SD) and the replication dataset (RD). In the SD each wind field is constructed on a random input vector with a unique random seed different from all others in the dataset. In the RD, multiple wind fields are generated with different random seeds for the same realization of the input random vector of wind parameters. To create the SD, 1,200 simulations were run, of which 1,108 were between cut-in and cut-out wind speed and 903 completed successfully. The RD is based on 720 simulations, of which 660 were within the wind speed bounds and 558 completed successfully. These 720 simulations were based on random turbulent wind fields generated from 24 input vectors and 30 random seeds for each input vector.

Extreme turbulence model

For the ETM the design standard defines V_{hub} as a random variable, while σ_1 is a deterministic value depending on V_{hub}. Only random input vectors with V_{hub} between cut-in and cut-out wind speed were considered.

Similar to the NTM scenario, a SD and RD were created. To create the SD, 240 simulations were run, of which 223 were between cut-in and cut-out wind speed and 182 were completed. The RD is based on wind fields from 24 random input vectors and 30 seeds, resulting in 720 simulations, of which 690 had valid wind speeds and 568 completed.

Extreme operating gust

In the EOG scenario, gusts were generated according to the IEC 61400-1 standard for V_{hub} in the range of 5-25 m/s in steps of 2 m/s. Each gust profile was centred around zero wind speed and superimposed on a random turbulent wind field generated from the corresponding V_{hub} and a fixed turbulence standard deviation $\sigma_1 = 0.1$. An exemplary turbulent wind speed trace at hub position $V_x(10,10)$ with a gust superimposed is shown in **Figure 5**. For each V_{hub}, turbulent wind fields were generated from a set of 30 random seeds, resulting in a total of 330 simulations, of which 189 were completed. In this scenario, all simulations with V_{hub} larger than 19 m/s aborted. This is not unexpected, because of the lack of a proper turbine shutdown mechanic in the OpenFAST controller implementation.

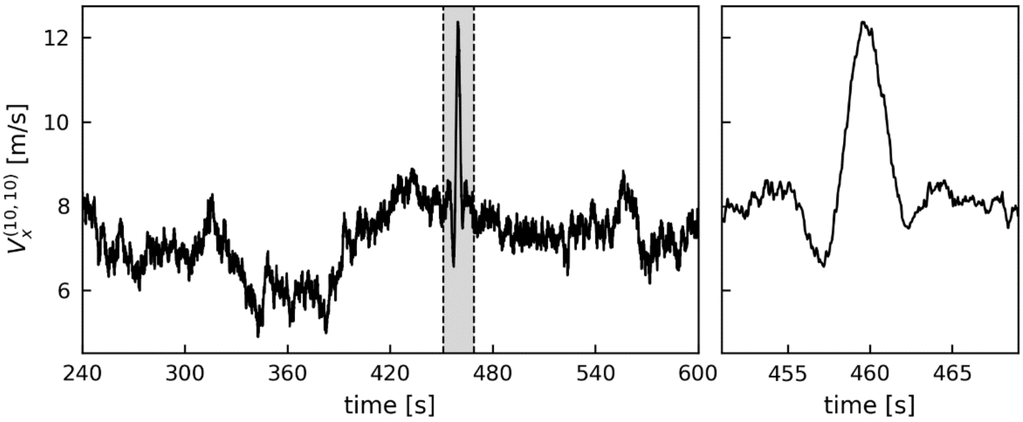

Figure 5: Turbulent wind trace at hub position with an extreme operating gust between 455 and 465 s.

Table 5: Summary of datasets (normal turbulence model, NTM; extreme turbulence model, ETM; extreme operating gust (EOG))

Wind conditions	**Dataset**	**Number of input vectors**	**Number of random seeds**	**Number of simulations (completed/within wind speed bounds/conducted)**
NTM	SD	1200	1	903/1108/1200
NTM	RD	24	30	558/660/720
ETM	SD	240	1	182/223/240
ETM	RD	24	30	568/690/720
EOG	-	11	30	189/330/330

Training set

As described in Section 3.4.5, we build two separate surrogate models at low- and high- wind speeds for each QoI, and we seamlessly integrate them through a simple classifier on V_{hub}. For each regime a surrogate was trained on a total of 100 ASE simulations. The simulations were randomly selected from the valid simulations (wind speed according to classification) across all seven data sets listed in **Table 5**. With a simulated time of 10 min and a sampling frequency of 160 Hz, this results in $\mathrm{O}(10^7)$ time steps for the full training set. Because the surrogate only requires $\mathrm{O}(10^{2-3})$ hyperparameters to be fitted, and due to the temporal coherence of both input and output time series, only a random subset of $\mathrm{O}(10^{5-6})$ time steps was used to train the surrogate. The composition of the training set is provided in **Table 6**.

Validation set

The validation of the mNARX approach was performed with the simulations that remained after the selection of the training sets for the low and high wind speed regime surrogates. Consequently, all simulations presented in our benchmark study (Section 3.5.1) are out-of-sample. The composition of the validation data is given in **Table 6**.

Table 6: Composition of training and validation sets

Wind conditions	**Dataset**	**Number of simulations in training set**		**Number of simulations in validation set**	
		low wind speed regime	**high wind speed regime**	**low wind speed regime**	**high wind speed regime**
NTM	SD	41	35	549	278
NTM	RD	29	22	329	178
ETM	SD	9	5	112	56
ETM	RD	11	31	419	107
EOG	-	10	7	85	87

3.5.2. Validation analysis: error plots and error measures

Due to the underlying physics, different QoIs can have very different spectral characteristics. Most exhibit strong periodicity due to the rotation of the rotor, in conjunction with wind shear. As a result, many QoIs depend on the state of the wind turbine at any given instant.

To address this dependence, we introduced the concept of a surrogate modelling chain as a multilevel approach in Section 3.4. However, errors in the wind turbine state, especially rotor speed and the derived rotor orientation, lead to phase mismatch between actual and predicted outputs. Common time series error measures, such as the root mean square error (RMSE), penalize this phase mismatch and result in errors that do not well represent the performance of the surrogate in emulating the dynamic turbine response. Therefore, RMSE is used only for auxiliary QoIs that do not depend directly on rotor orientation, such as pitch angle or rotor speed.

For the primary QoIs, aggregate quantities common in the wind turbine design process are used instead as meaningful error measures. We focus in particular the peak load values, damage equivalent load (DEL):

$$DEL = \sqrt[m]{\frac{T_{life}}{T_{sim}} \frac{1}{N_{eq}} \sum_{i=1}^{N_A} \frac{n_i A_i^m}{N_{eq}}}\,, \tag{15}$$

with $T_{life} = 20$ years, $T_{sim} = 10$ min, $N_{eq} = 10^7$ and $m = 10$ for the blades, or the integrated energy production over a ten-minute interval (E):

$$E = \int_0^{T_{sim}} P(t)dt \tag{16}$$

with $T_{sim} = 10$ min.

A complete list of aggregated quantities used for each QoI is provided in **Table 2**.

As described in Section 3.5.1, we use two different types of dataset: a standard dataset with no repeating vectors of wind-climate conditions and only unique random seeds on the one hand, and replicate datasets where each dataset is used repeatedly with different random seeds on the other hand. We use the standard dataset (SD) to evaluate the overall accuracy of the mNARX approach, and the replication dataset (RD) to compare the error introduced by the surrogate with the uncertainty associated with the stochasticity of the wind fields, commonly known as seed-to-seed uncertainty.

To represent the results obtained from the standard dataset, we use scatter plots of scalar values aggregated from the full output time series whenever possible. The comparison of the aggregated value from the predicted time series with the value from the true output time series provides a general overview of the performance of the surrogate. For each scatter plot, we highlight with a red and a green circle data points that correspond to bad and good predictions, respectively. We also show the corresponding true and predicted QoI time series. In addition, for each standard dataset and QoI, we show a random selection of five complete true and predicted raw output traces from simulations with very low to very high wind speeds.

We use histograms and box plots to relate the error introduced by the surrogate with the seed-to-seed uncertainty. We define the seed-to-seed variability $\grave{o}_{\text{seed}}$ of an aggregated quantity z as

$$\grave{o}_{\text{seed}}(\mathbf{v},s)=z(\mathbf{v},s)-\mu_{z(\mathbf{v})}\,, \tag{17}$$

where $\mathbf{v}$ is a realization of the random wind parameters (e.g. V_{hub}, σ_1, etc), and s is the random seed and $\mu_{z(\mathbf{v})}=\frac{1}{n_s}\sum_{s=1}^{n_s}z(\mathbf{v},s)$, with n_s the number of available seeds. Analogously, we define the surrogate error $\grave{o}_{\text{surr}}$ as:

$$\grave{o}_{\text{surr}}(\mathbf{v},s)=\hat{z}(\mathbf{v},s)-z(\mathbf{v},s)\,, \tag{18}$$

where $\hat{z}$ is the value aggregated from the predicted time series.

Table 7: QoIs and aggregated quantities used for model validation

QoI	Aggregated quantities
$\phi(t)$	RMSE
$\omega(t)$	RMSE
$P(t)$	RMSE, E
$M_y^{\text{Bld}}(t)$	DEL, $\max(M^{\bullet}(t))$

3.5.3. Detailed performance analysis on selected QoIs

Due to the complex nature of wind turbine models, which include structural physics, wind dynamics, control systems, etc., it is in general not possible to blindly build a surrogate model in the form of Eq. (2) directly from the input model. As introduced in Section 3.4.2, a sequence of intermediate auxiliary QoIs are constructed instead, and they are re-used for surrogating different final quantities of interest.

Two auxiliary quantities of interest are in particular important, as they are re-used by virtually every other QoI that is being surrogated: blade pitch angle (controller-driven) and rotor speed (see Section

3.4.2). In the following, we first provide a detailed analysis of the surrogate performance on these two quantities, followed by a selection of the relevant QoIs introduced in Section 3.3.2.

3.5.3.1. Auxiliary QoI: blade pitch

Rotor speed is one of the most important state variables necessary to assess the state of a wind turbine, as it affects power production, and virtually all the loads each turbine component is subject to. While in principle it is mostly dependent on the input turbulence box (and consequently to its reduced representation $\xi(t)$), it is also dependent on the blade pitch $\phi(t)$, which is electronically determined by the control system. The latter is therefore the first auxiliary QoI that we are going to present here.

Following the surrogate modelling approach introduced in Section 3.4, the mNARX surrogate was calibrated with the set of parameters reported in **Table 8**.

Due to the nature of the quantity, we assess its accuracy in terms of root-mean-square error (RMSE) on each time series on the relevant validation set. A graphical summary of the surrogate performance on the NTM validation dataset (see Section 3.5.2) is provided in **Figure 6**. With an overall RMSE in general below 0.03 rad (< 2°), it is clear that the surrogate can accurately reproduce the behaviour of the ROSCO controller with high accuracy on the entire nominal wind speed dataset. As a visual aid to associate RMSE with performance, two traces are highlighted below the RMSE histogram: a red trace associated to a relatively high RMSE, and a green trace associated to a relatively low one.

To additionally assess the performance of the surrogate, we showcase in **Figure 7** a number of raw traces randomly selected from the NTM validation dataset. In particular, no clear drift/error accumulation is clearly visible over time. This is expected, due to $\phi(t)$ being essentially a function of the exogenous longitudinal wind-component $\mathbf{V}_x$, hence not strongly dependent on the autoregressive part of the mNARX process.

Corresponding validation plots for the extreme turbulence model (ETM) validation dataset are reported in **Figure 8** and **Figure 9**, respectively. The surrogate model performs remarkably well, with a slight increase in RMSE error, mostly due to occasional outliers. One such outliers is provided for reference on the lower left plot of **Figure 8**, where the surrogate pitch $\hat{\phi}(t)$ predicts a large controller response at $t \approx 240$ s that is not present in the real trace. As it is clear from **Figure 9**, the surrogate response is in general more accurate for higher wind speeds.

Table 8: Properties of mNARX surrogate for the blade pitch

Exogenous inputs:	$\mathbf{V}_x(t)$	
	Low wind speed regime	High wind speed regime
Number of turbulence box features $\xi(t)$	9	9
Total number of NARX inputs	9	9
NARX degree (interaction order)	3 (1)	5 (1)
Auto-regressive lags ℓ^y	{1,160}	{1,160}
Exogenous input lags ℓ^ξ	{1}	{1}
Total number of NARX parameters	33	55
Number of training simulations (samples)	100 (10^6)	100 (10^6)

Finally, the surrogate is benchmarked on the extreme operating gust (EOG) dataset in **Figure 10** and **Figure 11**. The performance is once again very good, comparable to the other two regimes (NTM and ETM). The behaviour of the controller at the onset of the gust is also accurately reproduced at all wind speeds (see **Figure 11**).

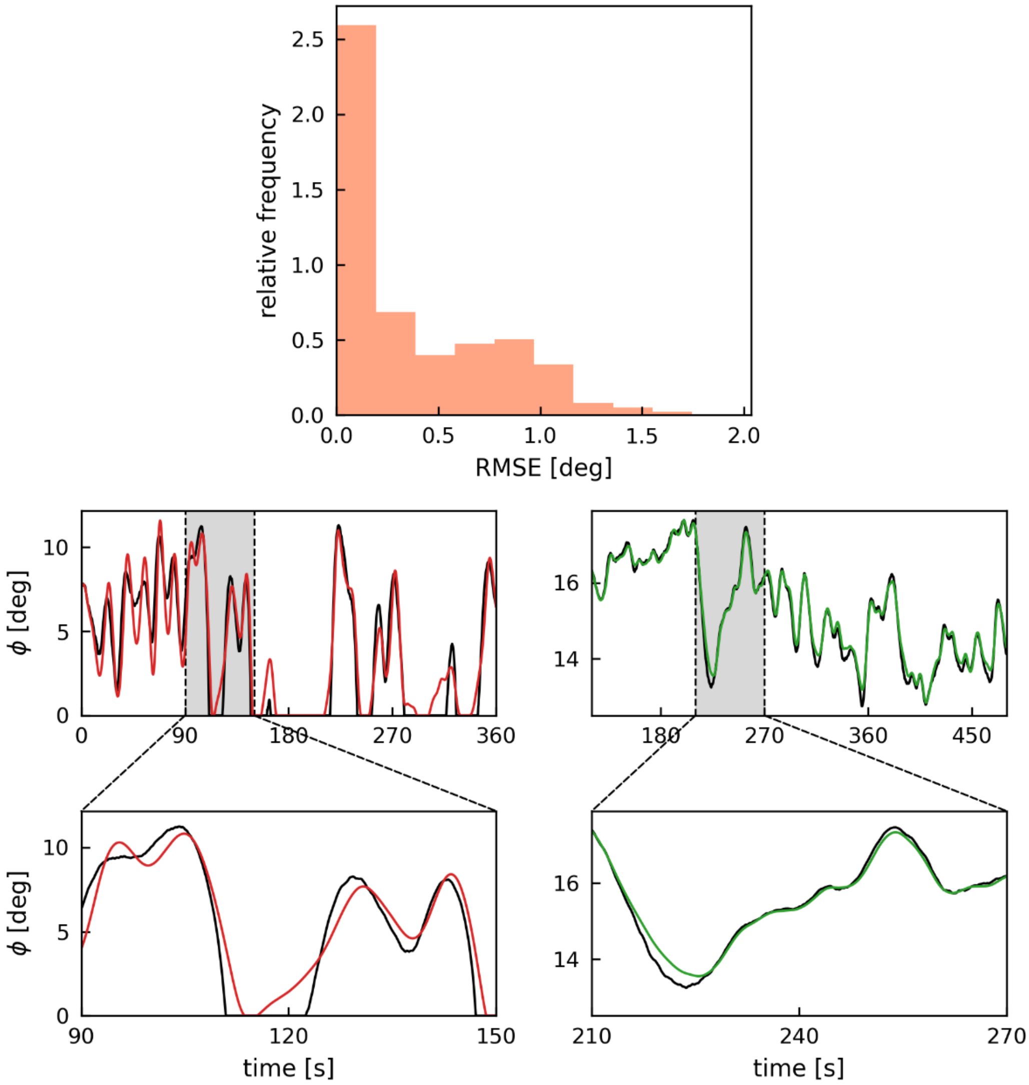

Figure 6: Surrogate model performance on the controller-driven blade pitch $\phi(t)$ on the normal turbulence model (NTM) validation dataset. Top: histogram of the root-mean-square error in degrees. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample.

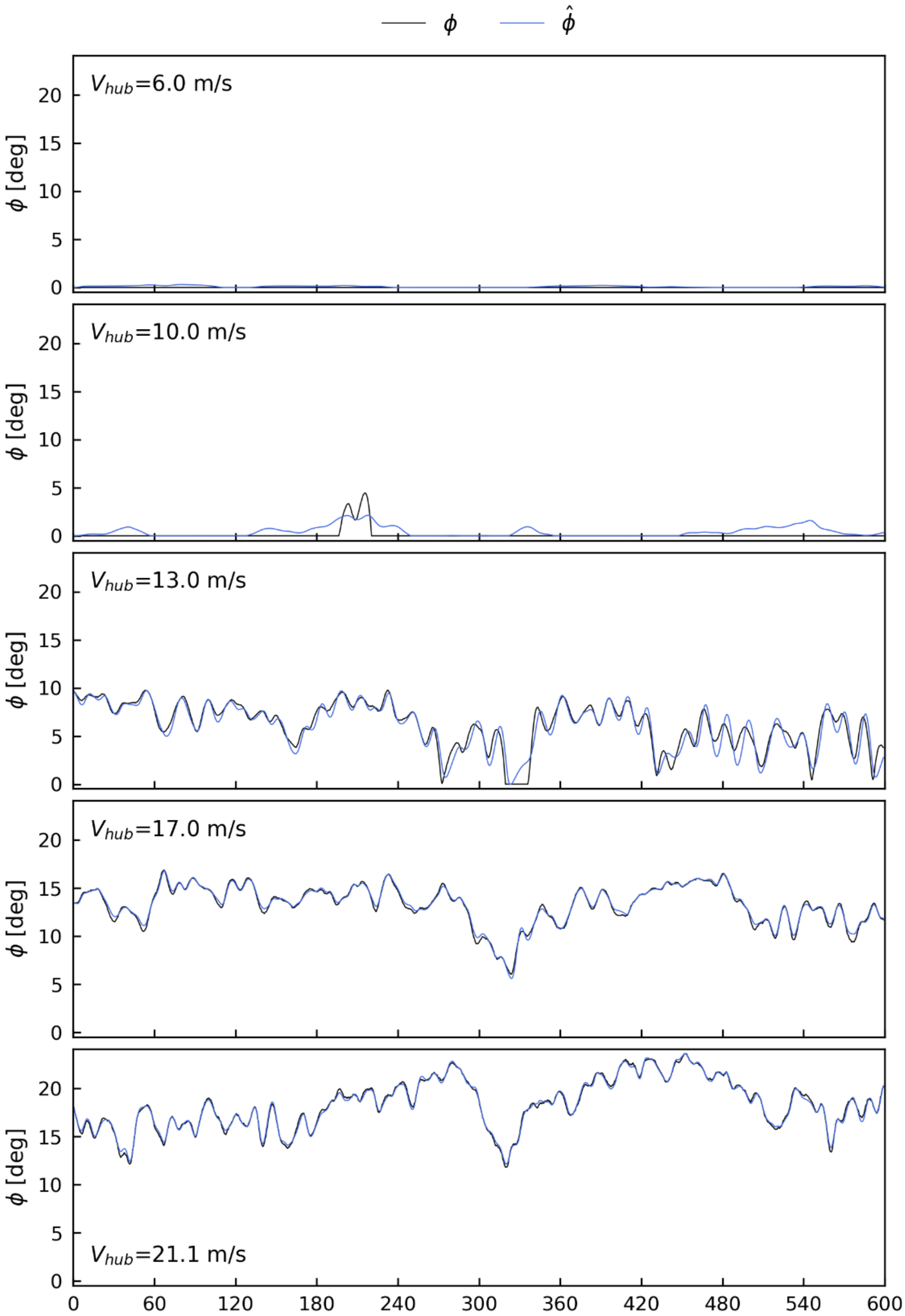

Figure 7: Performance of the surrogate model of $\phi(t)$ on a selection of raw traces from the NTM validation set

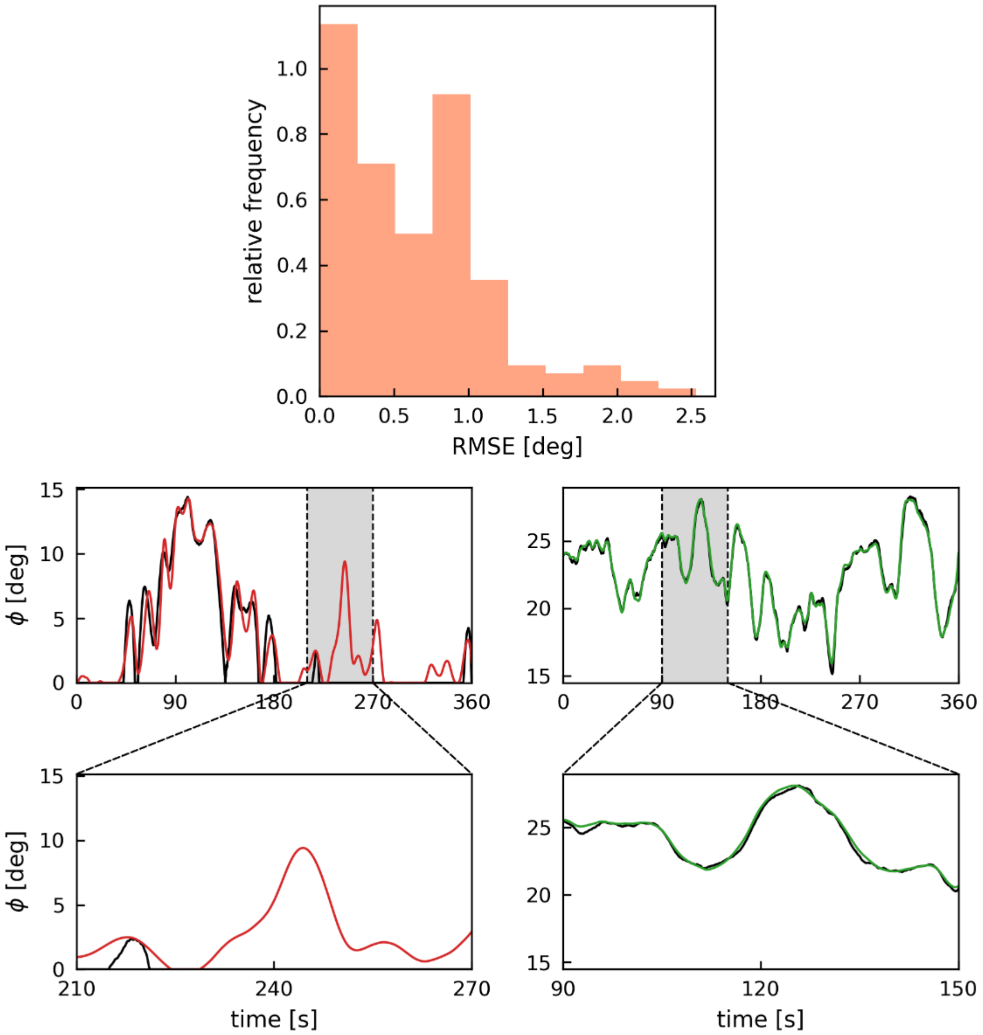

Figure 8: Surrogate model performance on the controller-driven blade pitch $\phi(t)$ on the extreme turbulence model (ETM) validation dataset. Top: histogram of the root-mean-square error in degrees. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample.

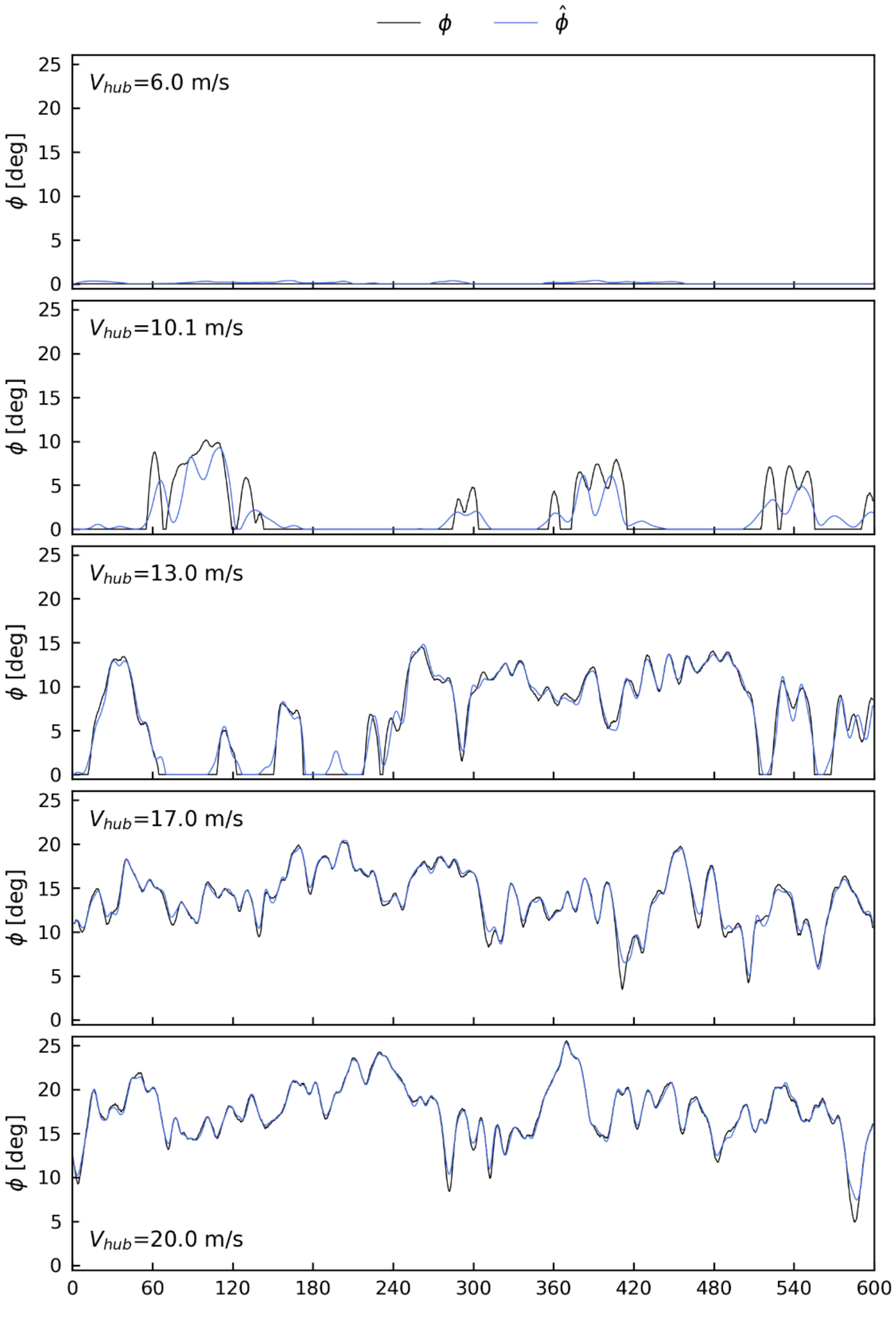

Figure 9: Surrogate model performance on a selection of raw traces from the ETM validation set.

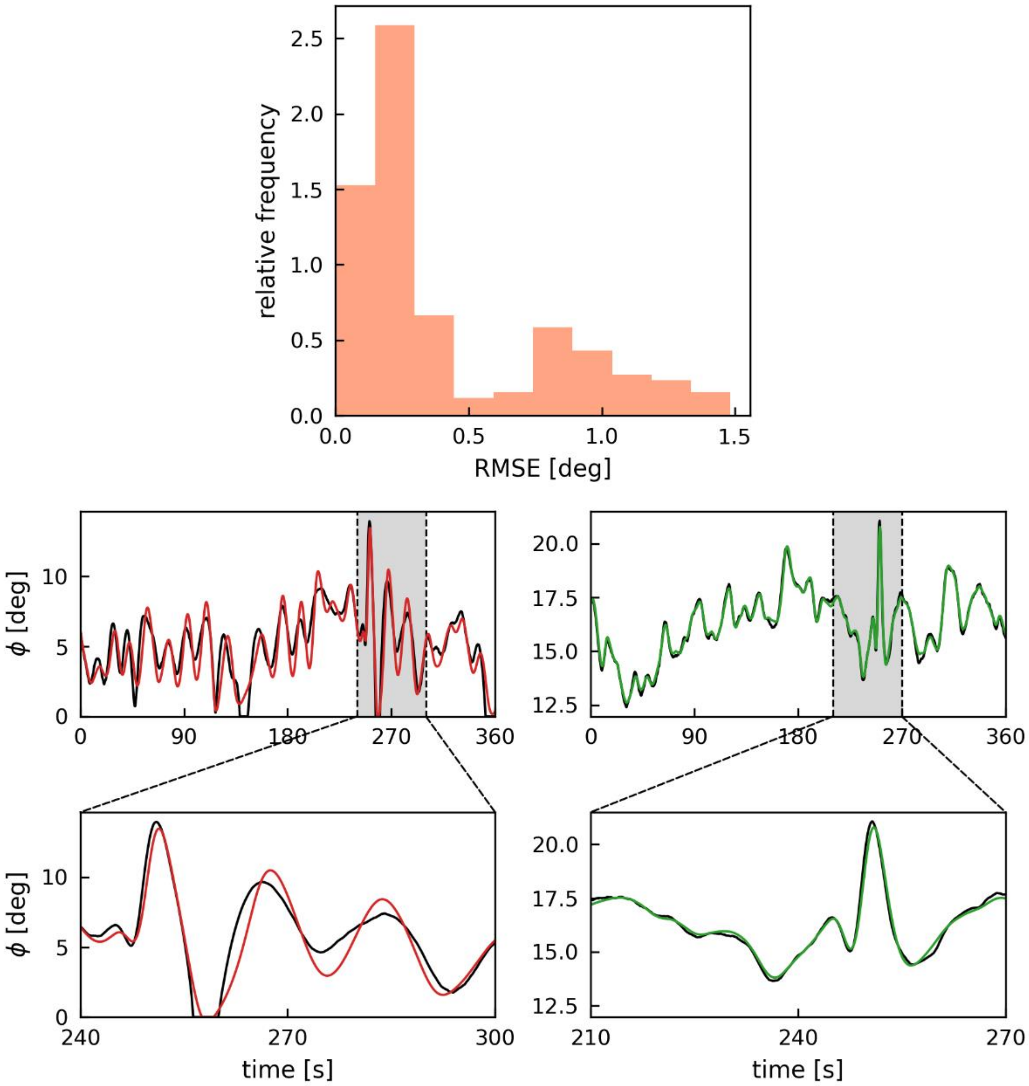

Figure 10: Surrogate model performance for the blade pitch $\phi(t)$ on the extreme operating gust (EOG) validation dataset. Top: histogram of the root-mean-square error in degrees. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample.

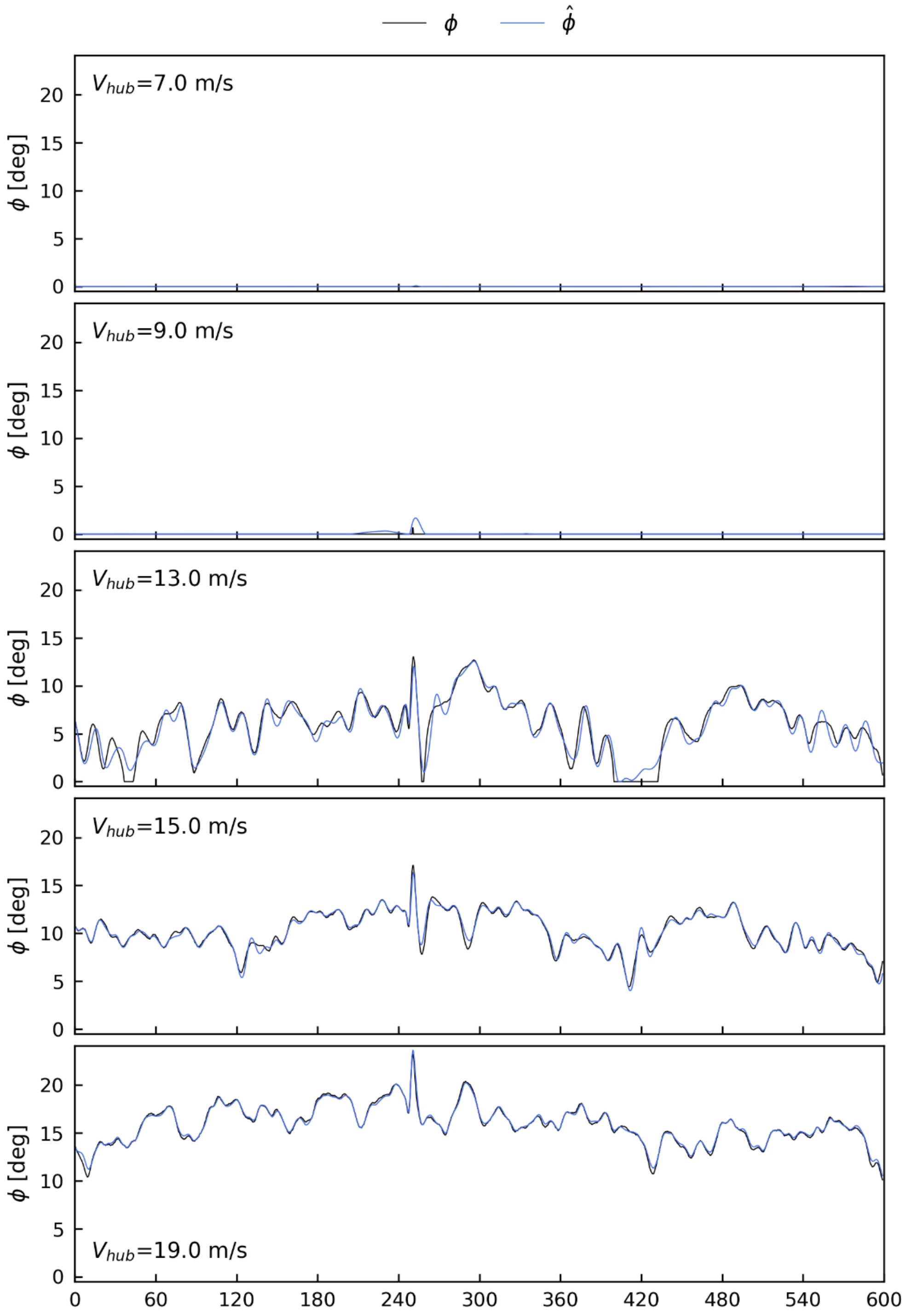

Figure 11: Surrogate model performance on a selection of raw traces from the EOG validation set.

3.5.3.2. Auxiliary QoI: rotor speed

Once the blade pitch is available, it can be used as an additional feature to surrogate the rotor speed. In analogy to **Table 8**, we provide a summary of the surrogate model characteristics in **Table 9**.

Table 9: Properties of mNARX surrogate for the rotor speed

Exogenous inputs:	$\mathbf{V}_x(t), \hat{\phi}(t)$	
	Low wind speed regime	High wind speed regime
Number of turbulence box features $\xi(t)$	9	9
Total number of NARX inputs	10	10
NARX degree (interaction order)	4 (1)	4 (1)
Auto-regressive lags ℓ^y	{1}	{1}
Exogenous input lags ℓ^ζ	{1,2}	{1,2}
Total number of NARX parameters	84	84
Number of training simulations (samples)	$100\ (10^6)$	$100\ (10^6)$

The surrogate of the blade pitch $\hat{\phi}(t)$ is now listed as one of the exogenous inputs, following the surrogate modelling chain described in Section 3.4.2.

An overview of the performance of the mNARX surrogate on the NTM validation set is reported in **Figure 12**. On the top panel, we present a histogram of the RMSE error (see Section 3.5.2) on the entire NTM validation dataset. The overall RMSE is quite low, well below 0.25 rpms for most of the dataset. Two traces with a relatively high and low RMSE, color-coded in red and green, respectively, are shown on the remaining rows to aid with the interpretation of this error measure.

To further assess the accuracy of the surrogate, an additional set of validation trace plots is provided in **Figure 13**. As for the case of the blade pitch $\phi(t)$, the performance over different wind regimes is overall excellent, and no clear performance drift/error accumulation over time is visible at time scales of $\mathrm{O}(6 \cdot 10^2 \mathrm{s})$.

Similar observations also hold for the surrogate model performance on the other two ETM and EOG wind regimes. Their performance are summarized in **Figure 14** and **Figure 15** for the ETM, and on **Figure 16** and **Figure 17** for the EOG validation sets. There is no appreciable decrease in surrogate modelling performance with the different climate scenarios, and even the onset of the wind gust for the EOG dataset is accurately modelled (see **Figure 17**).

The overall performance of the two auxiliary surrogates $\hat{\phi}(t)$ and $\hat{\omega}(t)$ allows us now to proceed in the surrogate modelling chain from Section 3.4.2 to a number of load-related output QoIs.

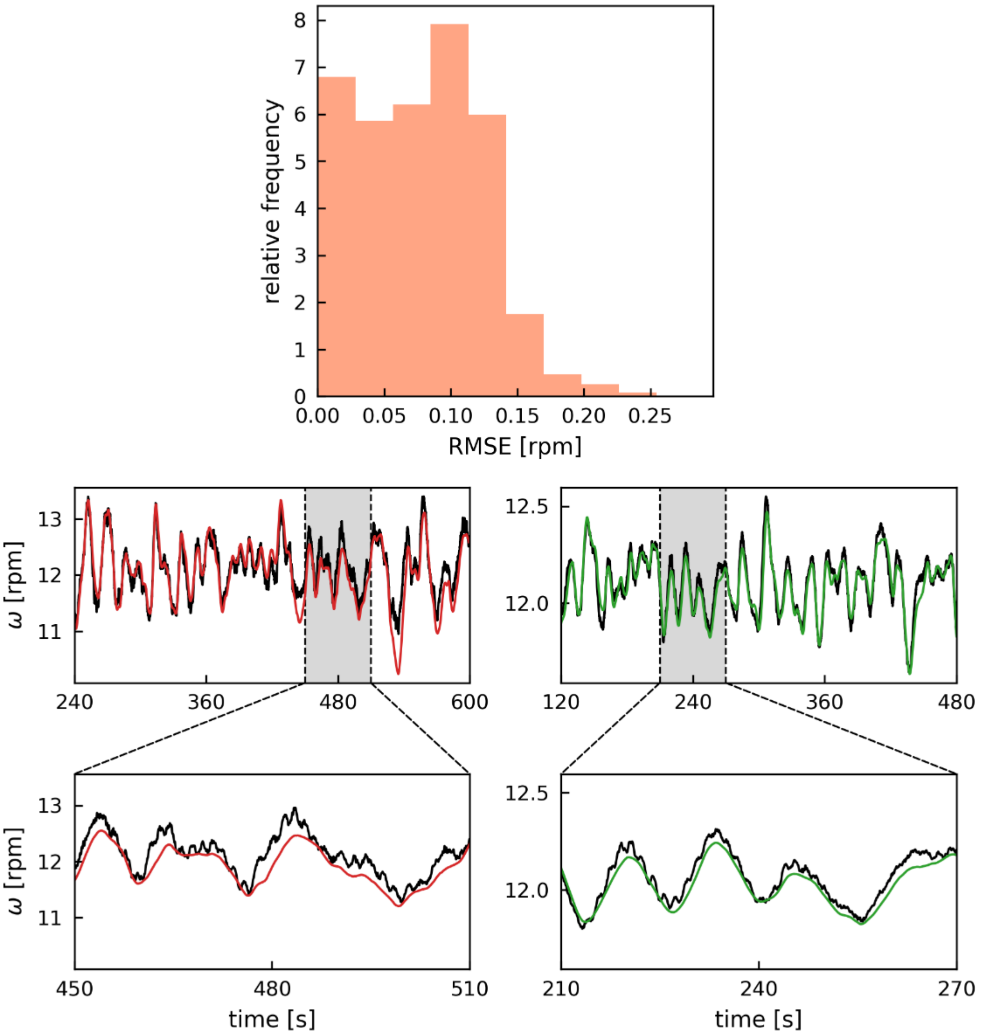

Figure 12: Surrogate model performance on the rotor speed $\omega(t)$ on the normal turbulence model (NTM) validation dataset. Top: histogram of the root-mean-square error in rpm. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample.

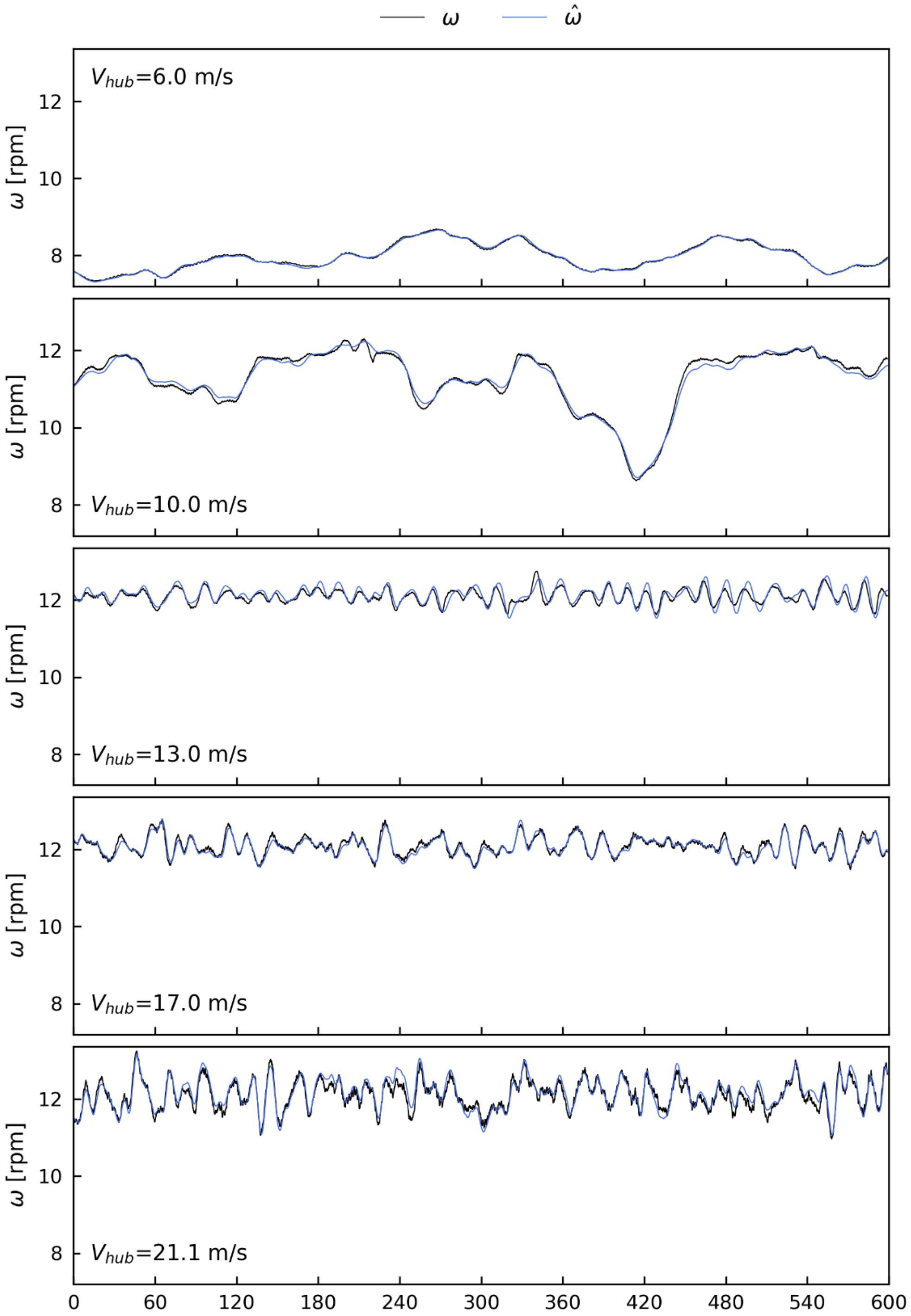

Figure 13: Performance of the surrogate model of $\omega(t)$ on a selection of raw traces from the NTM validation set.

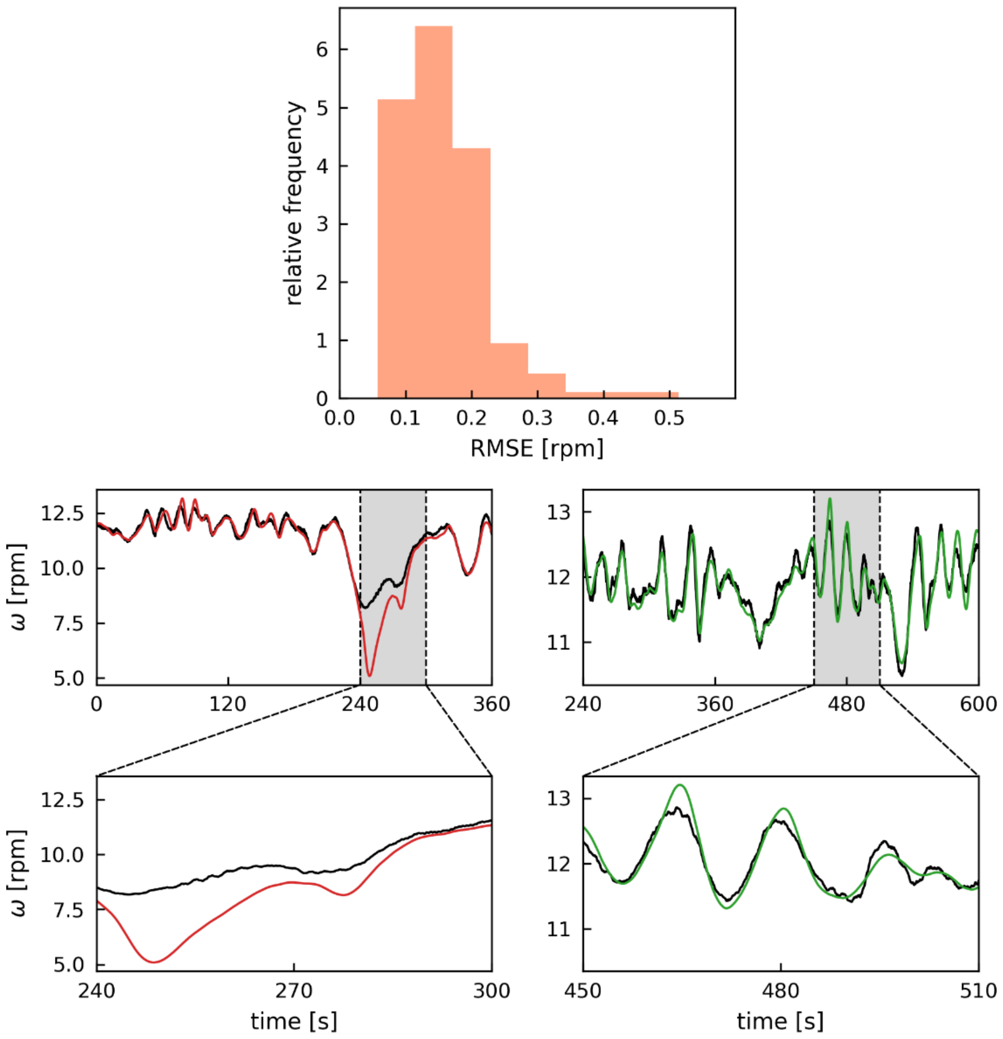

Figure 14: Surrogate model performance on the rotor speed $\omega(t)$ on the extreme turbulence model (ETM) validation dataset. Top: histogram of the root-mean-square error in rpm. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample.

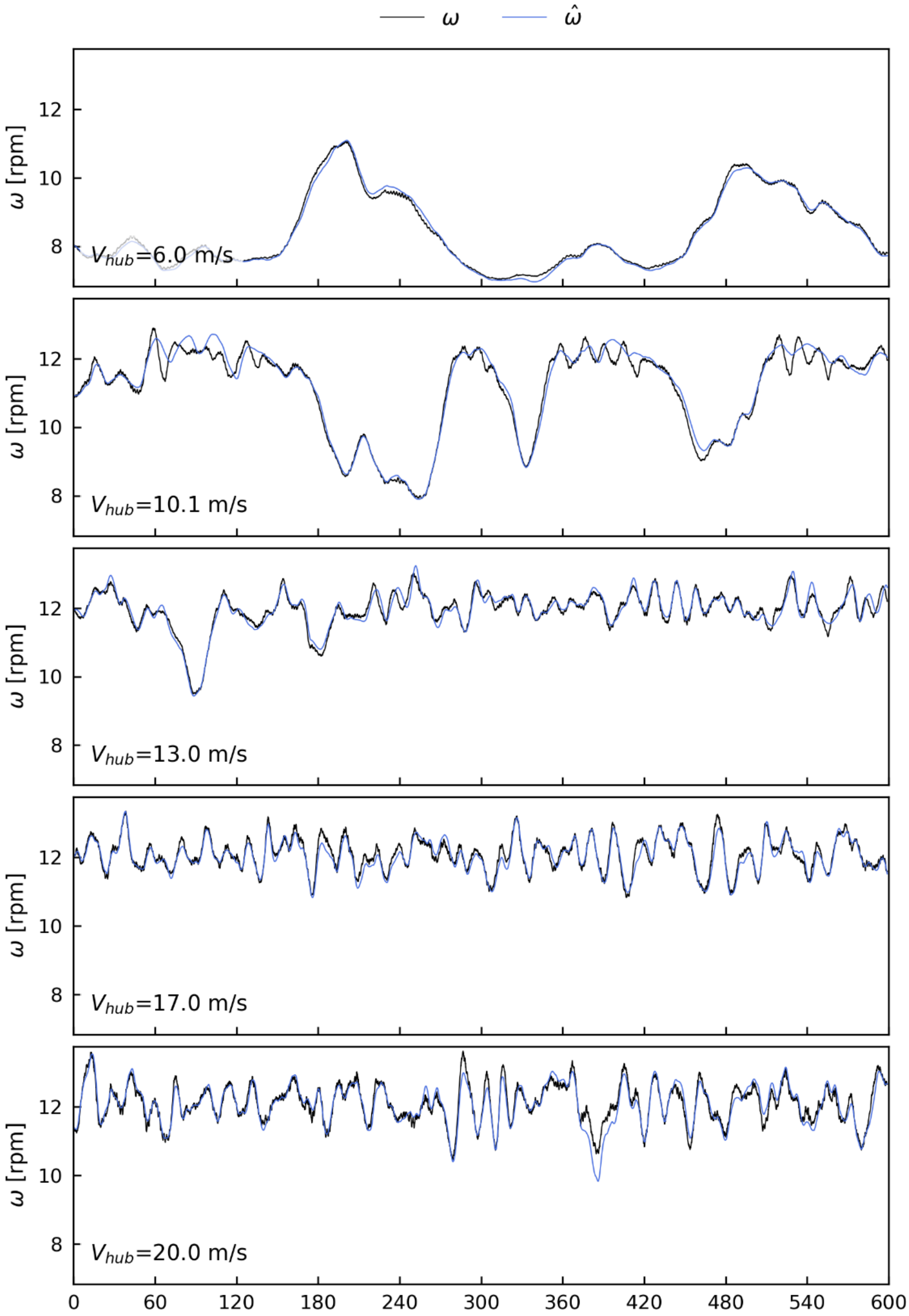

Figure 15: Performance of the surrogate model of $\omega(t)$ on a selection of raw traces from the ETM validation set.

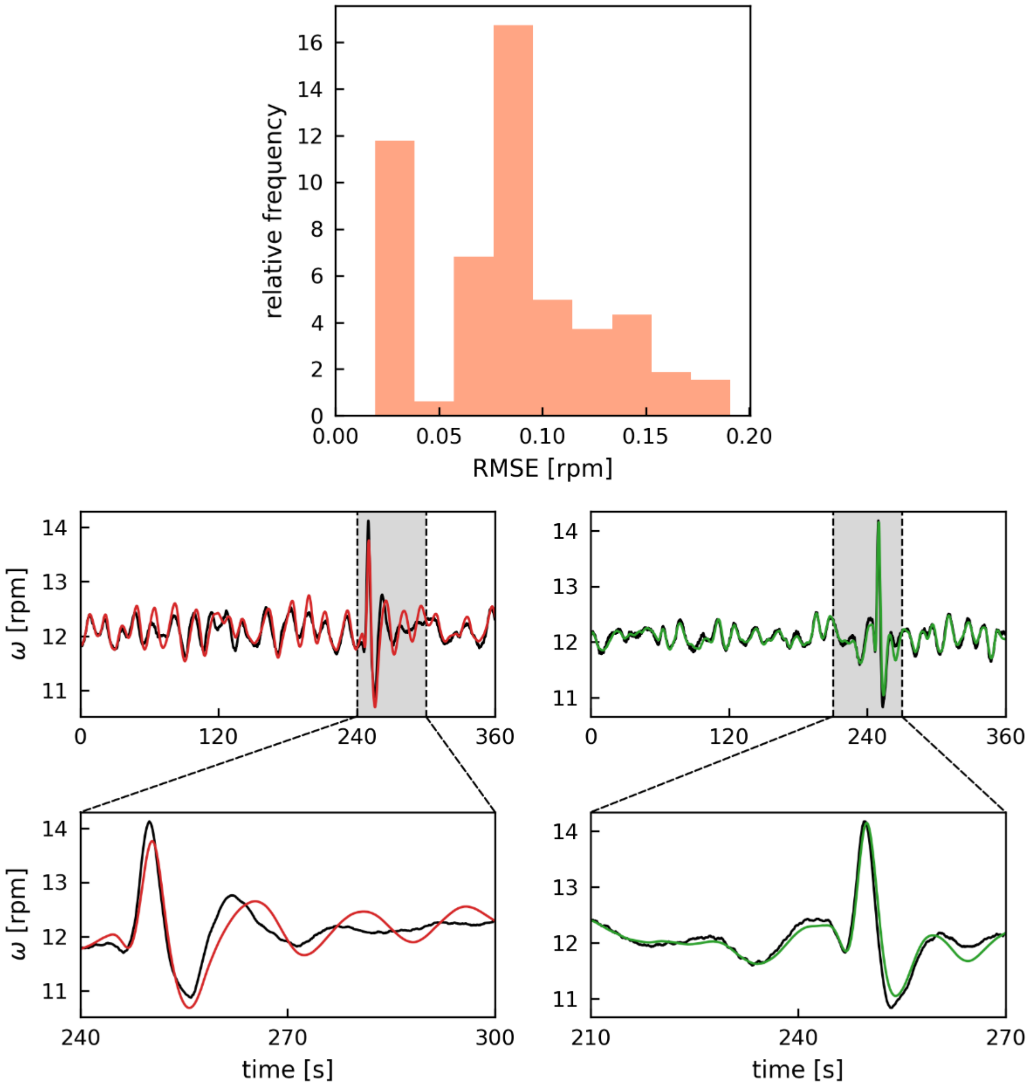

Figure 16: Surrogate model performance on the rotor speed $\omega(t)$ on the extreme operating gust (EOG) validation dataset. Top: histogram of the root-mean-square error in rpm. Bottom left: example of high-RMSE validation sample; bottom right: example of a low RMSE validation sample.

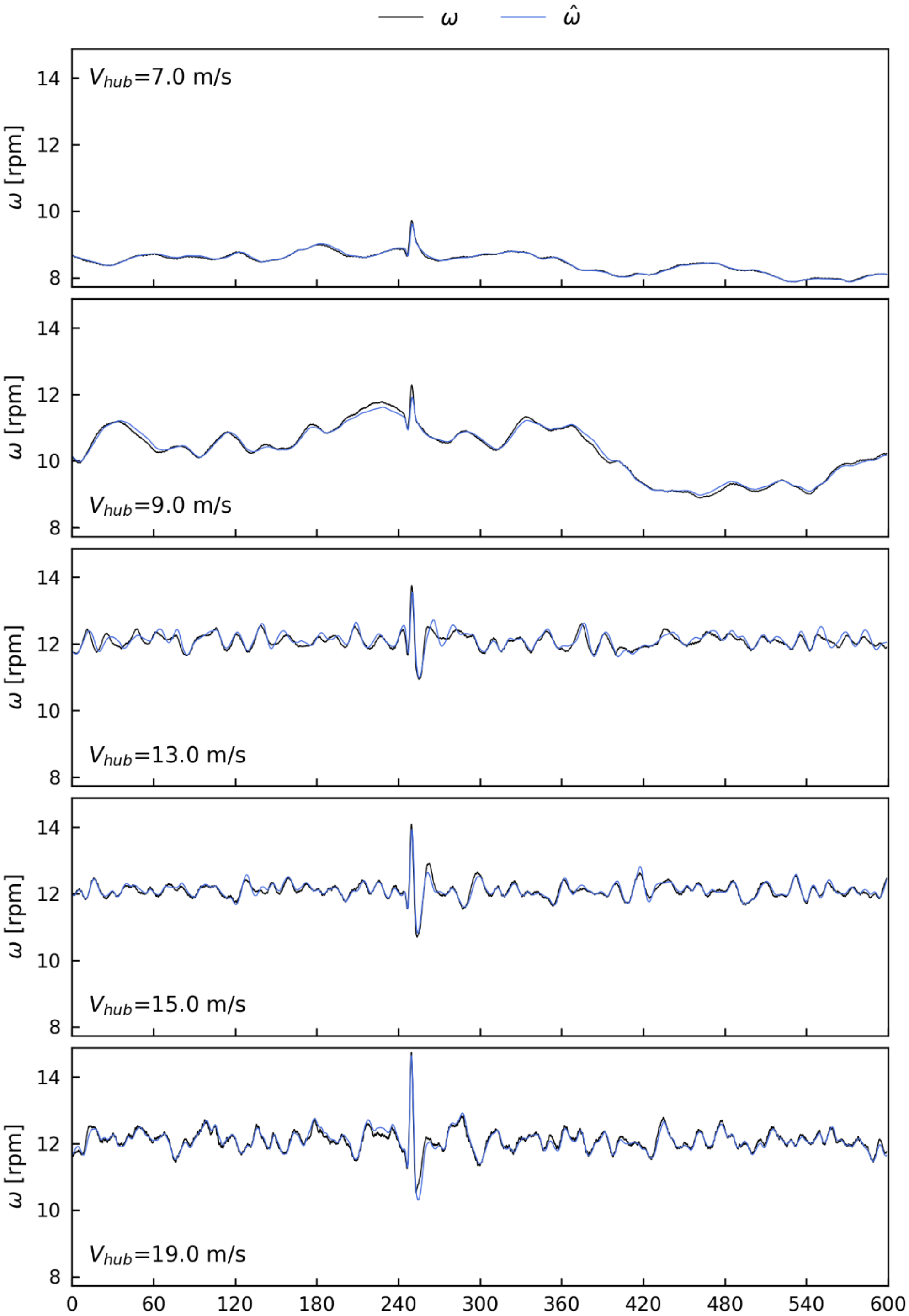

Figure 17: Performance of the surrogate model of $\omega(t)$ on a selection of raw traces from the EOG validation set.

3.5.3.3. Primary QoIs: power production

With both rotor speed and blade pitch available, it is now possible to surrogate most of the primary QoIs described in Section 3.3.2. We start from the electrical power P, using the mNARX configuration provided in **Table 10**.

Table 10: Properties of mNARX surrogate for the power output $P(t)$

Exogenous inputs:	$\mathbf{V}_x(t)$, $\hat{\phi}(t)$, $\hat{\omega}(t)$	
	Low wind speed regime	High wind speed regime
Number of turbulence box features $\xi(t)$	4	4
Total number of NARX inputs	6	6
NARX degree (interaction order)	7 (1)	3 (3)
Auto-regressive lags ℓ^y	{1,30}	{1,30}
Exogenous input lags ℓ^ζ	{1,30}	{1}
Total number of NARX parameters	98	164
Number of training simulations (samples)	100 (10^6)	100 ($5 \cdot 10^5$)

As a suitable error measure for this quantity, we use the integrated energy production over a 10 minutes time interval. We present an overview of the performance of the surrogate $\hat{P}(t)$ on the NTM data set in **Figure 18**. The overall accuracy of the surrogate is quite high, although a clear bias towards underestimating the produced energy is visible, especially close to the high total energy region. This behaviour is more clearly visible in **Figure 19**, where it becomes apparent that this bias occurs mostly in the region with $V_{\text{hub}} \approx 11\ \text{m/s}$. This is the transition region around which the wind turbine often reaches its rated power, hence changing its controller regime. In **Figure 20** we present a random set of raw traces for different values of V_{hub}, making this regime transition region even clearer. It is clear that the mNARX surrogate is very accurate at low and high wind speeds, but additional work is needed to improve its performance in the intermediate wind speeds region.

This behavior is further confirmed in the analogous set of figures for the extreme turbulence model (ETM) validation set: **Figure 21**, **Figure 22** and **Figure 23**. Because of the higher wind speeds, the overall performance of the surrogate is generally much better and less biased than for the normal turbulence model.

Finally, the performance of the mNARX surrogate on the extreme operating gust (EOG) data set is highlighted in **Figure 24**, **Figure 25**, and **Figure 26**. This last dataset unfortunately only has a limited number of simulations available at low high wind speeds, likely due to many of them failing to complete due to the lack of a turbine shutdown mechanism. Nevertheless, the surrogate model performance is once again consistent with the previous findings, with high accuracy at relatively low and relatively high wind speeds, but a significant drop in performance in the region around $V_{\text{hub}} \approx 10\ \text{m/s}$.

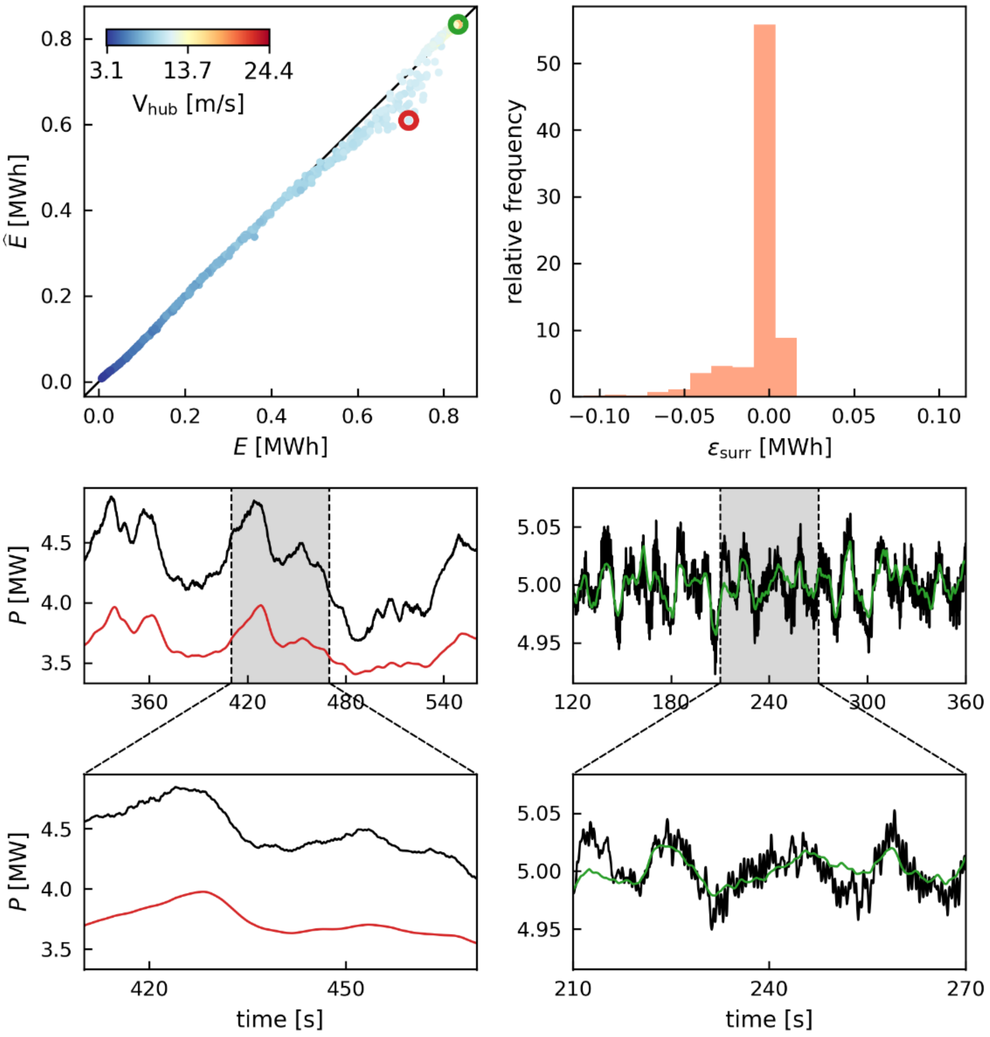

Figure 18: Surrogate model performance on the power production $P(t)$ on the normal turbulence model (NTM) validation dataset. Top left: scatter plot of true vs. surrogate total energy produced E. Top right: histogram of the energy production discrepancy in MWh. Bottom left: example of an inaccurate energy estimation sample. Bottom right: example of an accurate energy estimation sample.

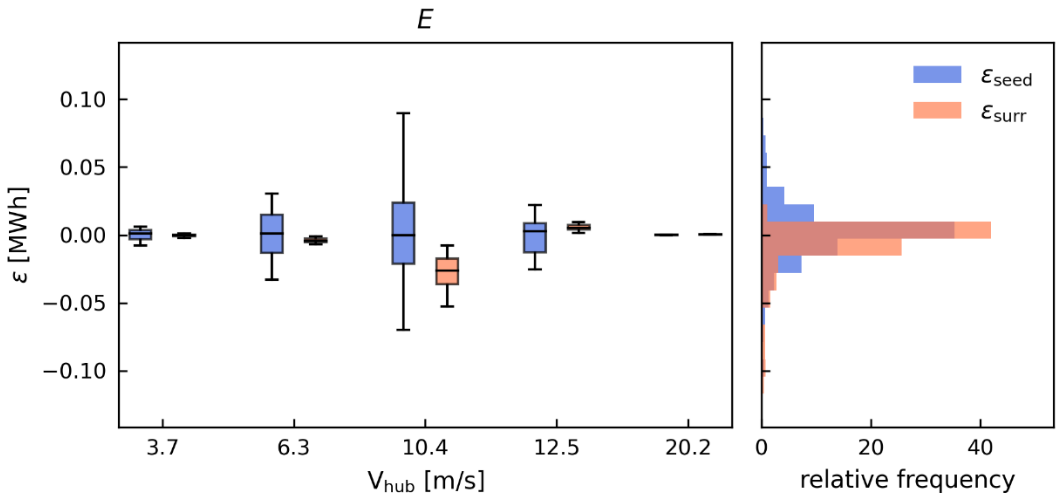

Figure 19: Surrogate model performance vs seed-to-seed variability on the total energy produced E on the normal turbulence model (NTM) validation data set.

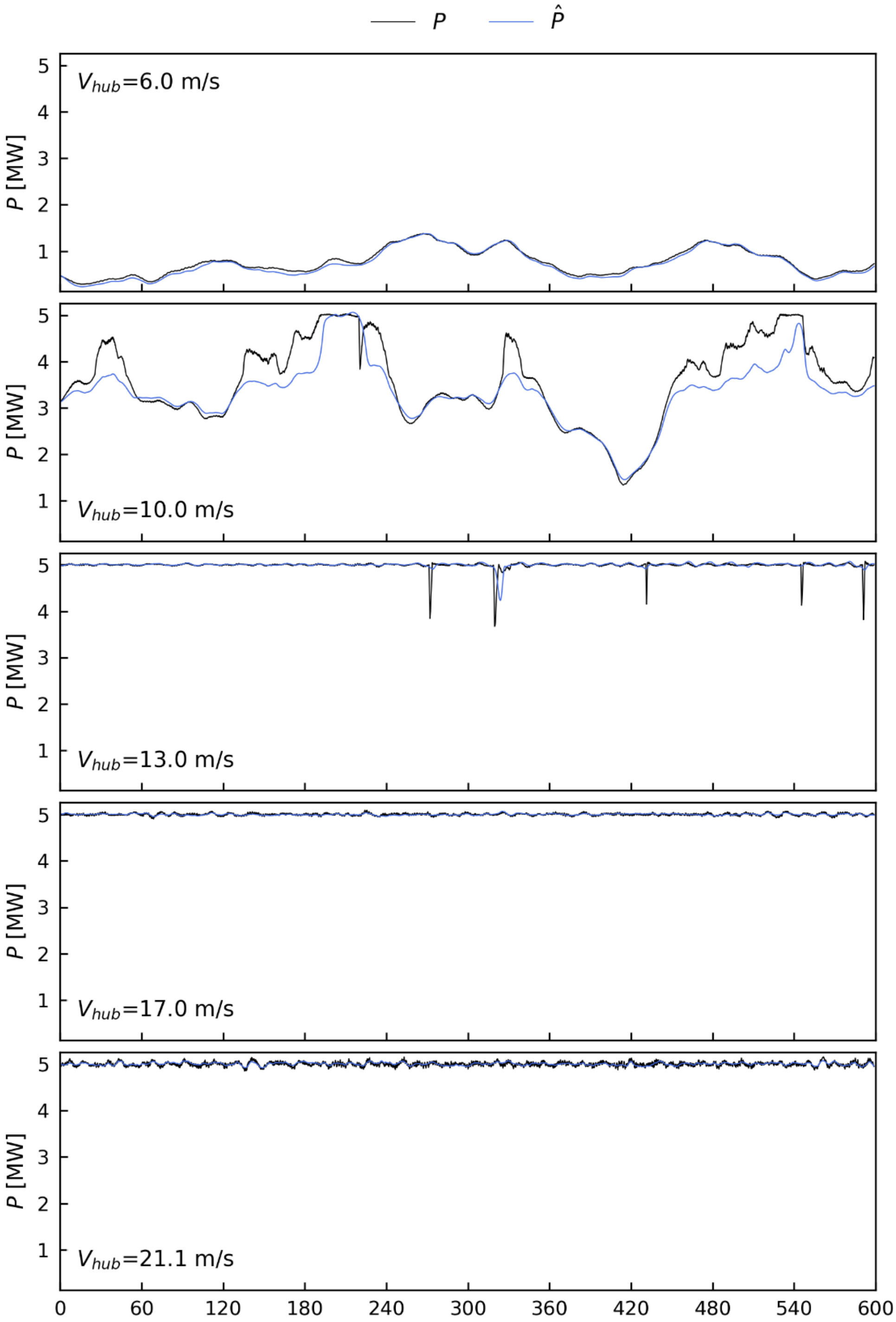

Figure 20: Performance of the surrogate model of the power output P on a selection of raw traces from the NTM validation set.

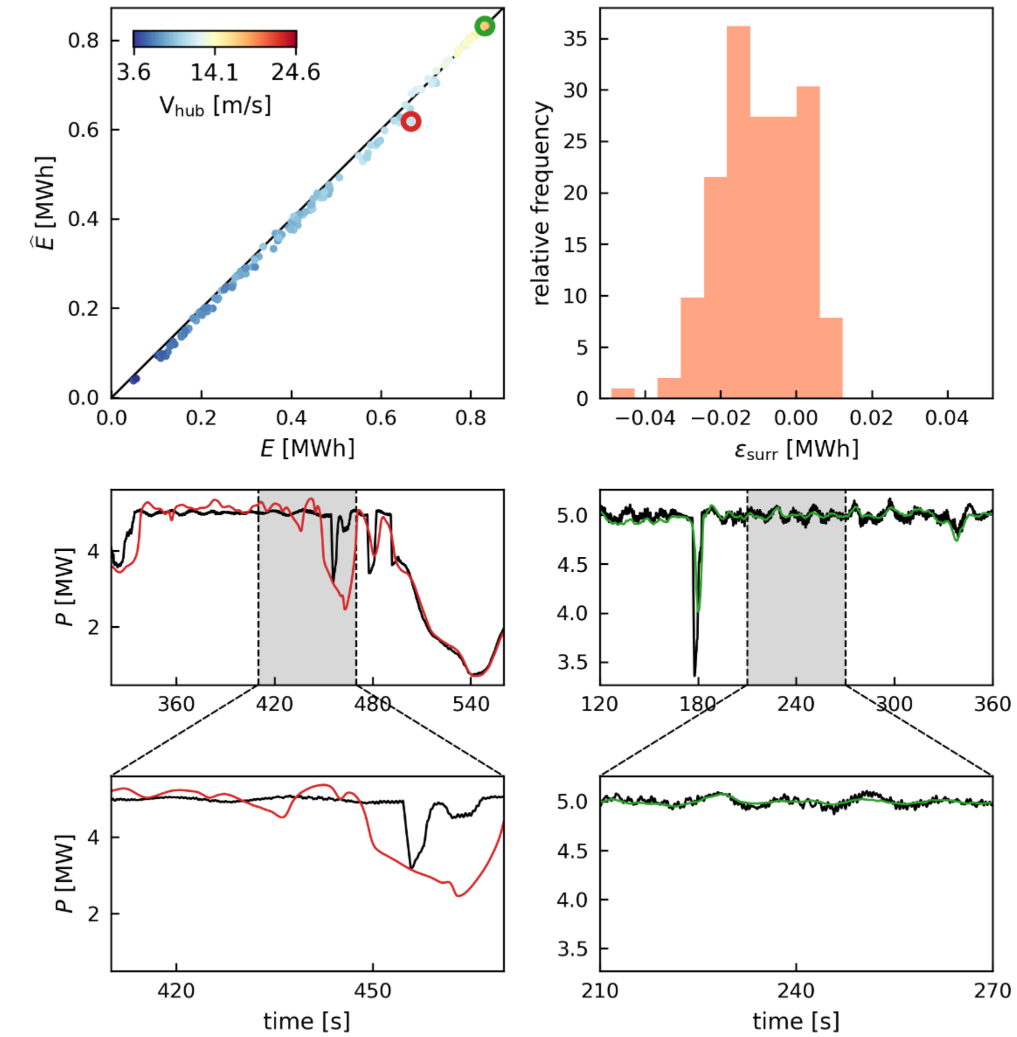

Figure 21: Surrogate model performance on the power production $P(t)$ on the extreme turbulence model (ETM) validation data set. Top left: scatter plot of true vs. surrogate total energy produced E. Top right: histogram of the energy production discrepancy in MWh. Bottom left: example of an inaccurate energy estimation sample. Bottom right: example of an accurate energy estimation sample.

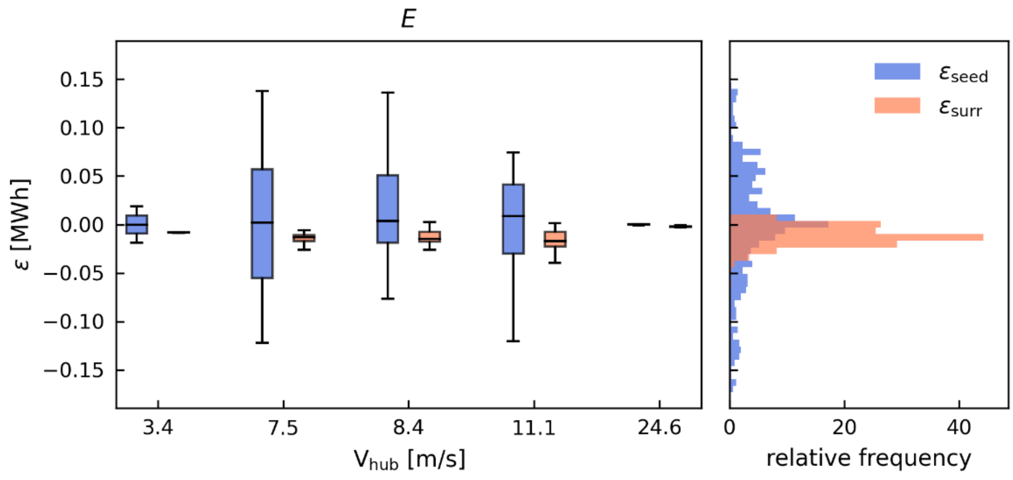

Figure 22: Surrogate model performance vs seed-to-seed variability on the total energy produced E on the ETM validation dataset.

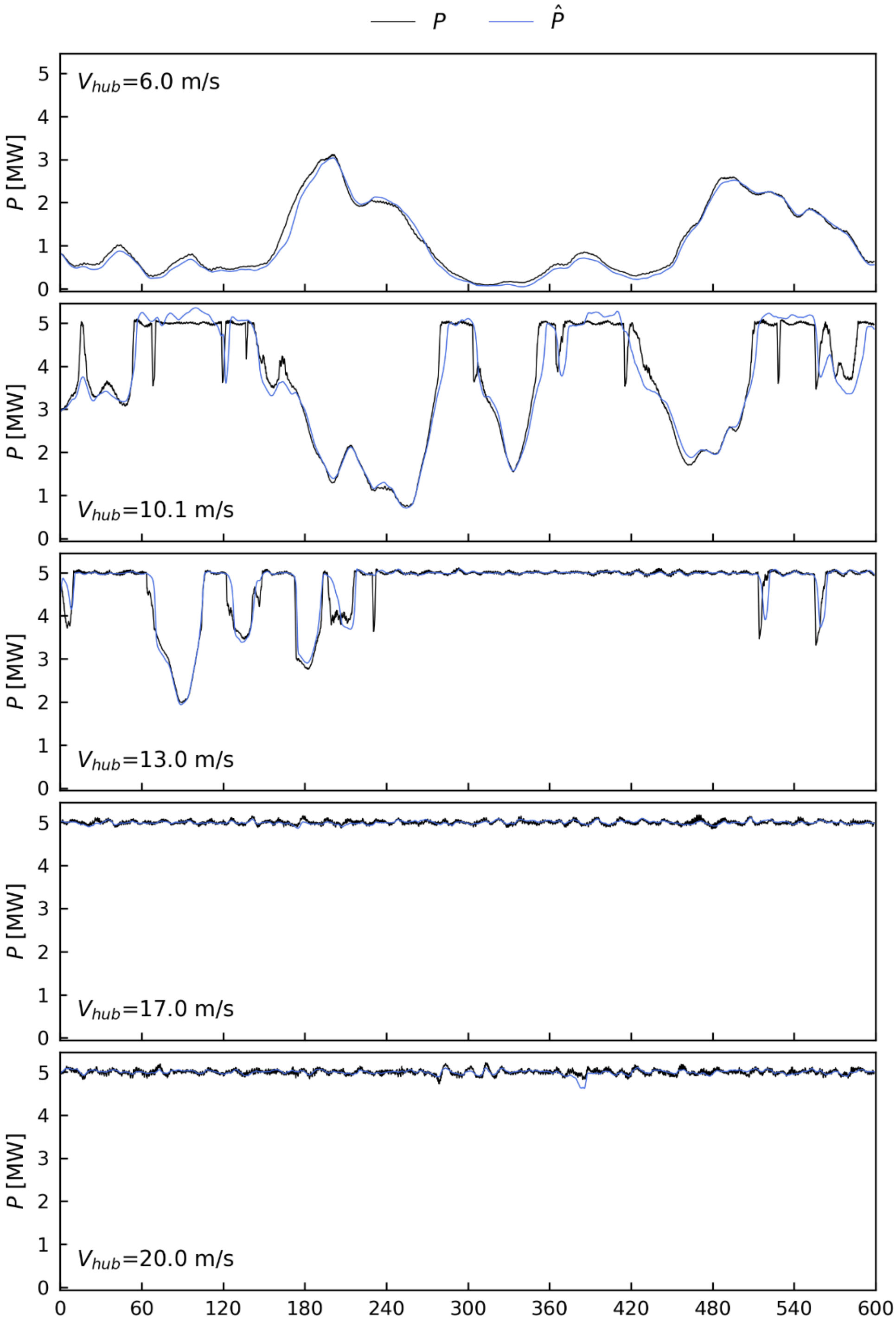

Figure 23: Performance of the surrogate model of the power output P on the ETM validation set.

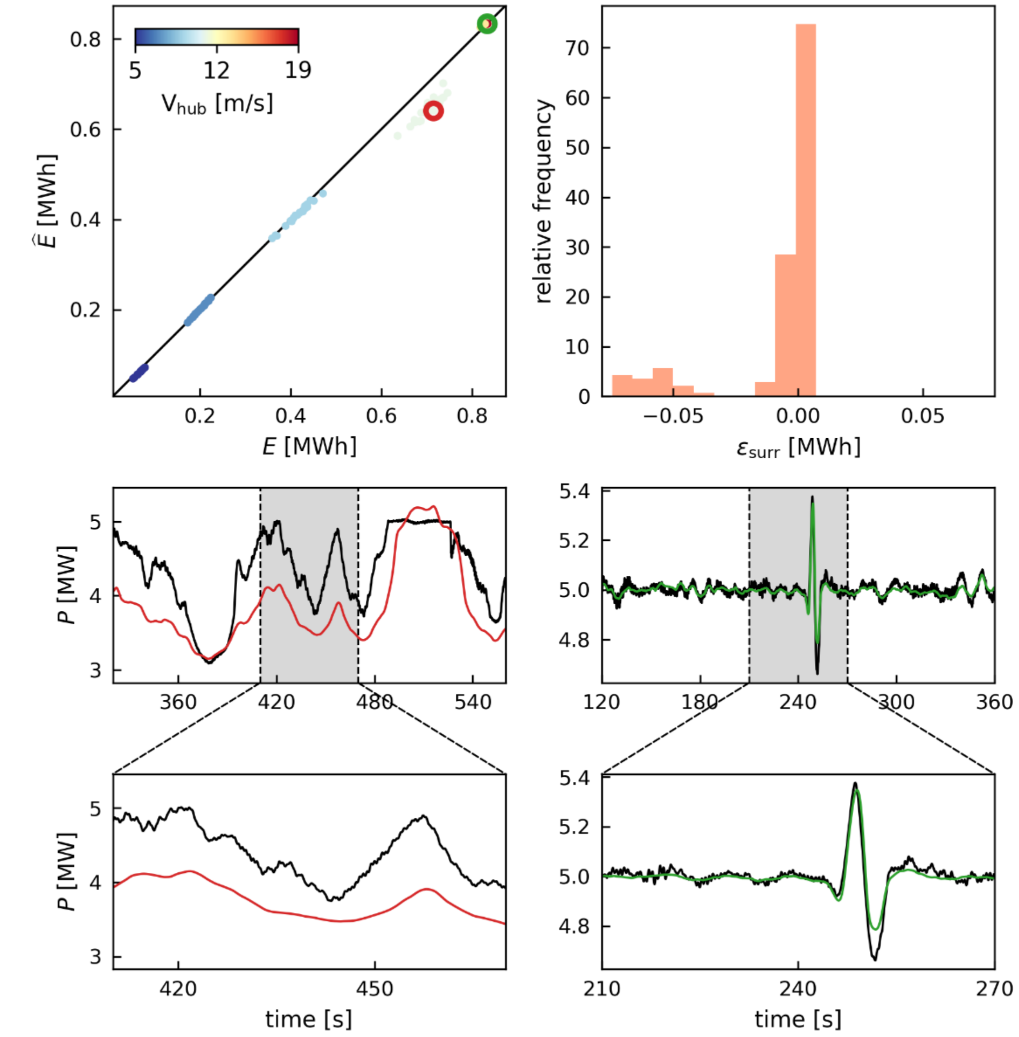

Figure 24: Surrogate model performance on the power production $P(t)$ on the extreme operating gust (EOG) validation data set. Top left: scatter plot of true vs. surrogate total energy produced E. Top right: histogram of the energy production discrepancy in MWh. Bottom left: example of an inaccurate energy estimation sample. Bottom right: example of an accurate energy estimation sample.

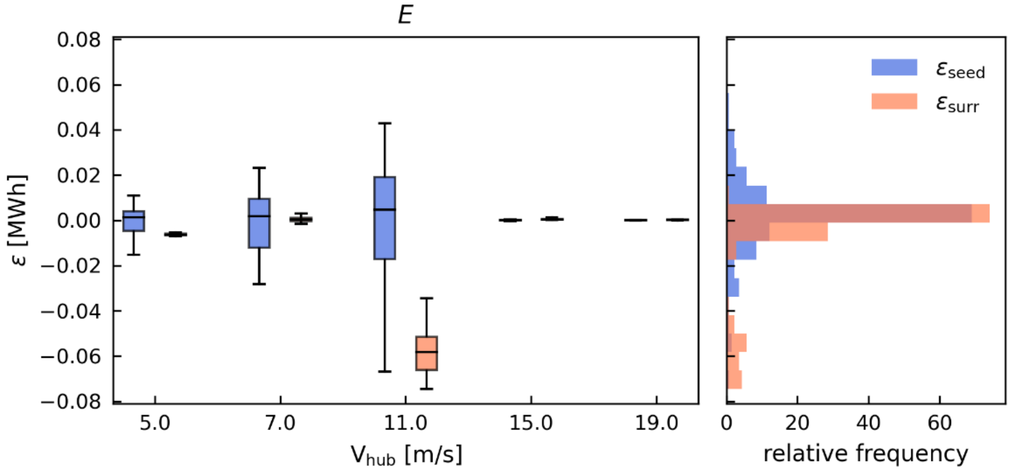

Figure 25: Surrogate model performance vs seed-to-seed variability on the total energy produced E on the EOG validation data set.

Table 11: Summary of the surrogate model performance on the total energy produced E. The coefficients of determination R^2 corresponds to the scatter plots in **Figure 18**, **Figure 21** and **Figure 24**. The mean of the surrogate error $\mu_{\varepsilon_{surr}}$, the standard deviation of the surrogate error $\sigma_{\varepsilon_{surr}}$ and the standard deviation of the seed-to-seed uncertainty $\sigma_{\varepsilon_{seed}}$ correspond to the histograms in **Figure 19**, **Figure 22** and **Figure 25**.

	Energy produced E				
	R^2	$\mu_{\varepsilon_{surr}}$	$\sigma_{\varepsilon_{surr}}$	$\sigma_{\varepsilon_{seed}}$	$\sigma_{\varepsilon_{surr}} / \sigma_{\varepsilon_{seed}}$
NTM	0.996	-0.007	0.019	0.022	0.832
ETM	0.995	-0.012	0.010	0.058	0.170
EOG	0.994	-0.009	0.020	0.015	1.341

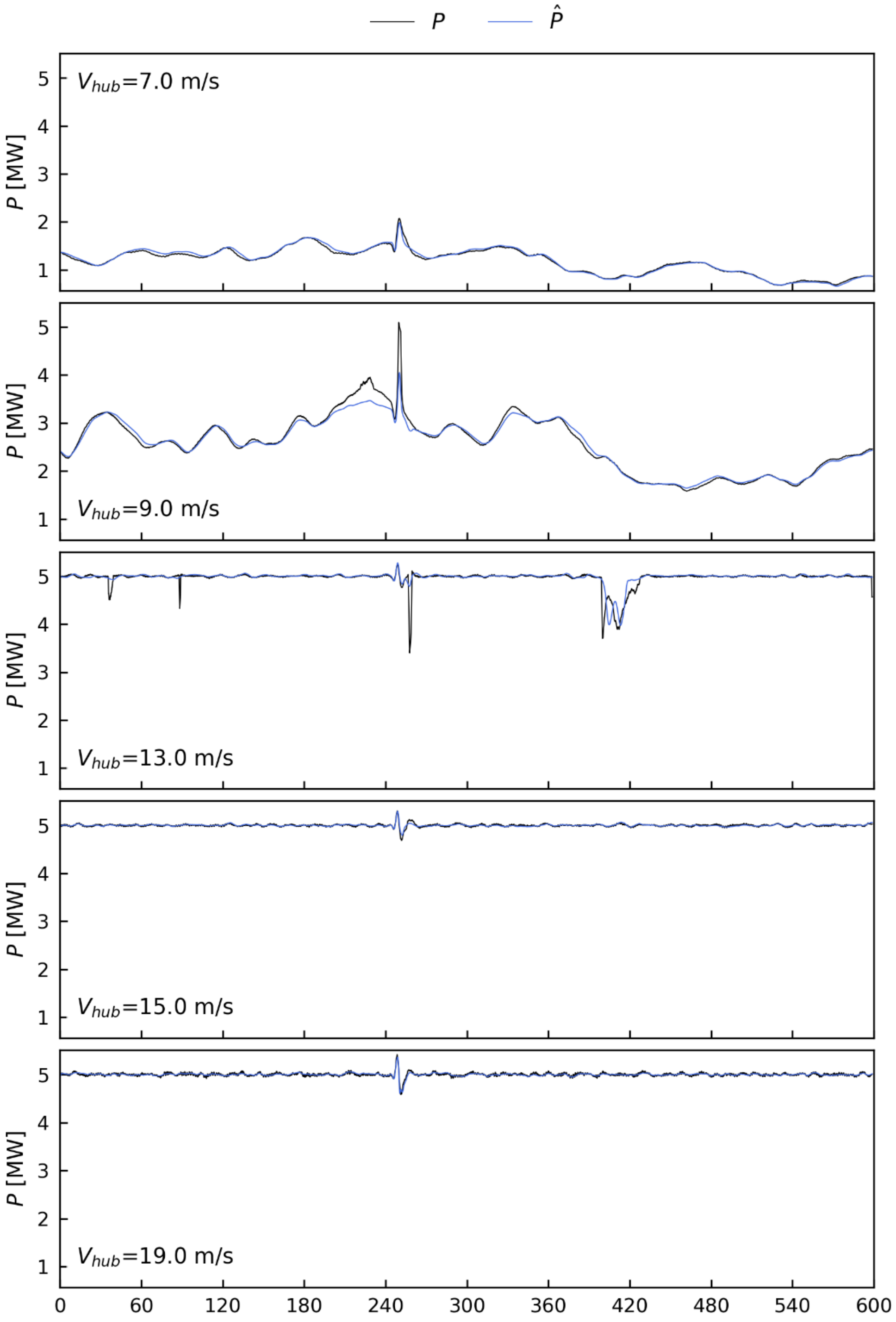

Figure 26: Performance of the surrogate model of the power output P on a selection of raw traces from the EOG validation data set.

3.5.3.4. Primary QoIs: flapwise blade root moment

Next, we benchmark the surrogate performance on the flapwise blade root bending moment M_y^{Bld}, using the mNARX configuration in **Table 12**.

Table 12: Properties of mNARX surrogate for the flapwise blade root moment $M_y^{\text{Bld}}(t)$

Exogenous inputs:	$\mathbf{V}_x(t)$, $\hat{\phi}(t)$, $\hat{\alpha}_i^k(t)$	
	Low wind speed regime	High wind speed regime
Number of turbulence box features $\xi(t)$	25	25
Total number of NARX inputs	34	34
NARX degree (interaction order)	2 (2)	2 (2)
Auto-regressive lags ℓ^y	{1,16,32}	{1,2}
Exogenous input lags ℓ^ζ	{1}	{1}
Total number of NARX parameters	740	702
Number of training simulations (samples)	100 ($2 \cdot 10^5$)	100 ($2 \cdot 10^5$)

As for the auxiliary QoIs presented in Sections 3.5.3.1 and 3.5.3.2, for each of the three validation wind regimes (NTM, ETM and EOG) we present the surrogate model performance at first as a summary/overview, and then on a selection of raw trace plots. Because of the nature of the flapwise blade root moment M_y^{Bld}, we choose to use as a two different performance metrics: RMSE on the damage equivalent load (DEL) for each trace, and RMSE on the maximum load. Please note that these quantities are used only as error measures on the output time series, not as the outputs of the surrogate model themselves.

The surrogate model performance of $\hat{M}_y^{\text{Bld}}(t)$ in terms of DEL is summarized in **Figure 27**, while its performance in terms of maximum load is provided in **Figure 28**. Both plots provide additional insight by means of the scatter plot $M_y^{\text{Bld}}(t)$ vs $\hat{M}_y^{\text{Bld}}(t)$ on the top left, where the true model versus the surrogate are plotted against one another. The heatmap color-coding represents the underlying mean V_{hub} for each simulation point.

Overall, the surrogate model accuracy of $\hat{M}_y^{\text{Bld}}(t)$ in terms of DEL is high, especially for either very low and very high V_{hub}, with a decrease in performance in the intermediate regimes. In particular, a small but significant bias of approximately -0.2 MN is visible across most of the samples, as summarized in the histogram of ò_{surr} on the top right. We interpret this bias as a consequence of an accuracy reduction at higher frequencies, that is visible both in the trace plots in **Figure 27**, and the larger set of raw traces presented in **Figure 30**. Similar observations hold for the performance on the maximum load per trace in **Figure 28**, and the overall surrogate behaviour is comparable.

For a better overview comparison, **Figure 29** provides a deeper insight on the nature of the discrepancy and bias observed in both error measures, with the top row corresponding to DEL, while the bottom corresponding to the maximum load per trace. In each row, we provide in red box-plots of the same discrepancy measure provided in **Figure 27**, but as a function of the V_{hub} (binned in 5 representative speeds for readability). To provide a comparison in scale, we provide a similar error measure based on the dispersion around the mean of the true values in the validation set, due to the natural seed-to-seed variability (see Section 3.5.2 for details). On the right, we provide instead aggregate histograms: the red histogram corresponds to the histogram on the top-right panel of **Figure 27**, while the blue histogram only showcases the dispersion of the true validation dataset around the mean, for each wind speed. Please note that the two error measures are not directly comparable (surrogate modelling error vs seed-to-seed uncertainty), but the seed-to-seed uncertainty is rather provided as a relative scale to assess approximation accuracy.

The main take away of **Figure 29** is that most of the surrogate modelling error is due to its bias at intermediate wind speeds, while low- and high- wind speeds are reproduced correctly and with little to no bias. **Figure 30** showcases a random set of traces from the NTM dataset with increasing V_{hub}, which visually confirms the high accuracy of the surrogate at low and high wind speeds, and its reduced performance at intermediate wind speeds.

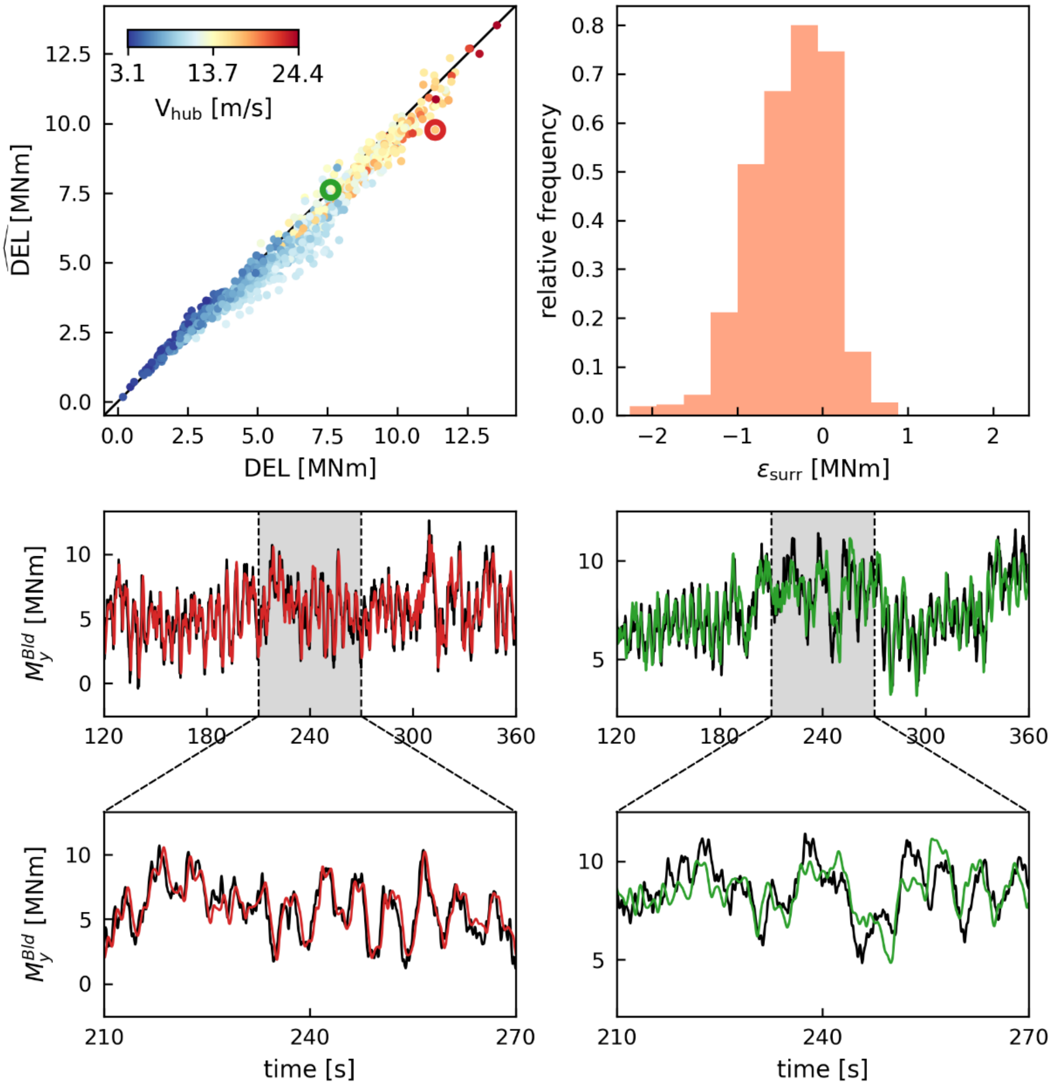

Figure 27: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the normal turbulence model (NTM) validation dataset. Top left: scatter plot of true vs. surrogate DEL estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate DEL estimation sample. Bottom right: example of an accurate DEL estimation sample.

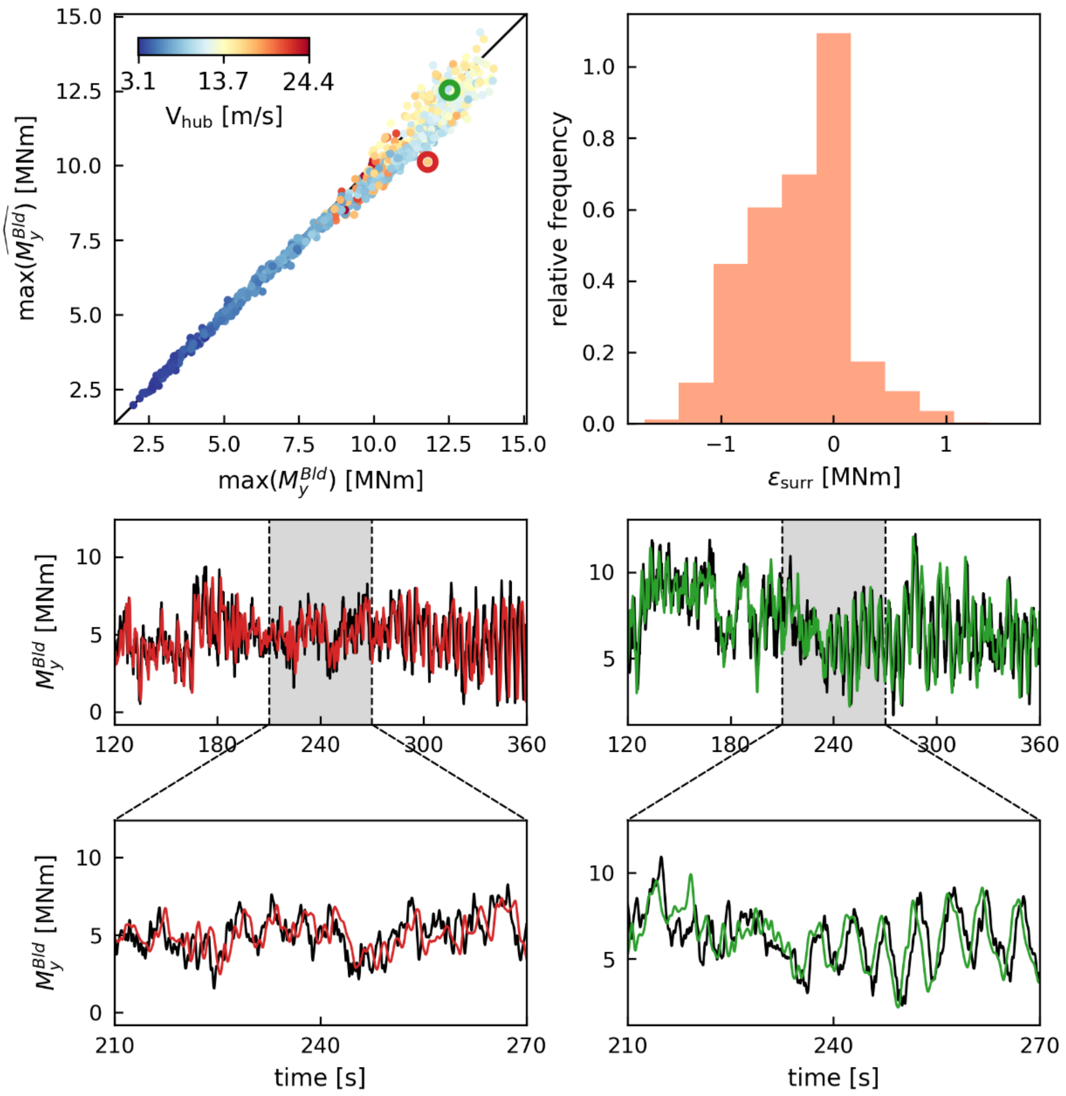

Figure 28: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the normal turbulence model (NTM) validation dataset. Top left: scatter plot of true vs. surrogate maximum load estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate maximum load estimation sample. Bottom right: example of an accurate maximum load estimation sample.

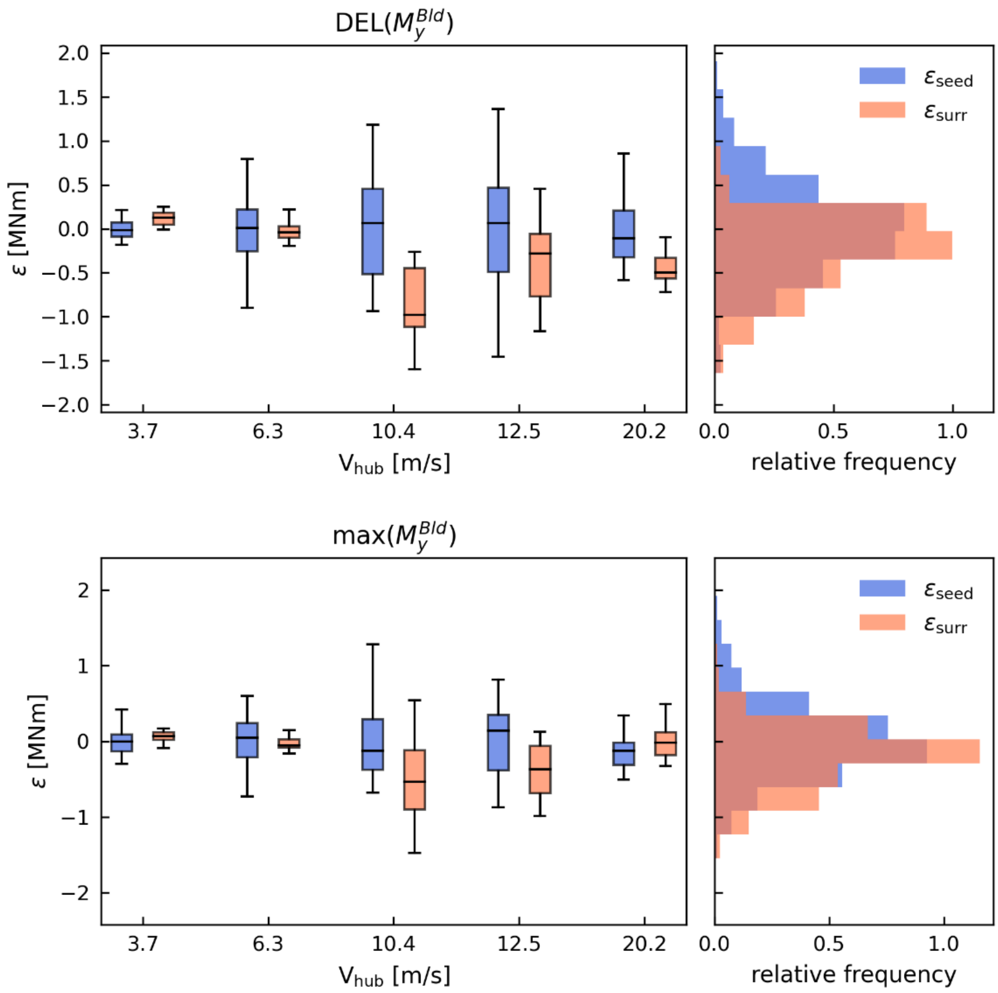

Figure 29: Surrogate model performance vs seed-to-seed variability on the flapwise blade root moment M_y^{Bld} on the normal turbulence model (NTM) validation dataset. Top panel: DEL. Bottom panel: maximum load.

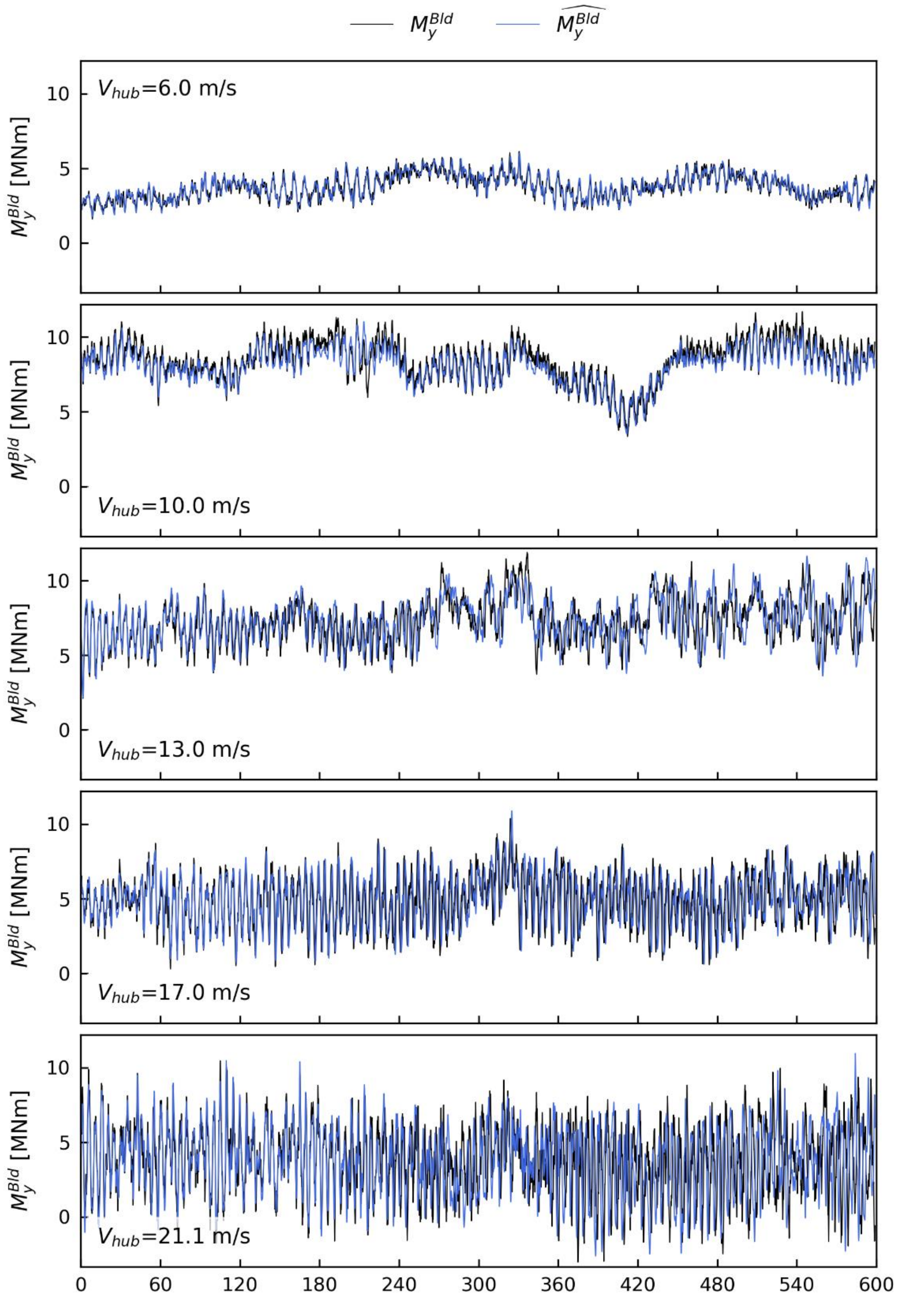

Figure 30: Performance of the surrogate model of the flapwise blade root moment M_y^{Bld} on the normal turbulence model (NTM) on a selection of raw traces from the NTM validation set.

Surrogate performance on the extreme turbulence model (ETM) is showcased by the corresponding set of plots in **Figure 31** (DEL), **Figure 32** (maximum load), **Figure 33** (surrogate error vs seed-to-seed variability) and **Figure 34** (raw trace plots). Consistently with the previous results, at the higher wind speeds typical of the ETM scenario, the overall accuracy remains comparable, but the bias is significantly reduced. This is particularly clear from both the box plots and the histograms in **Figure 33**.

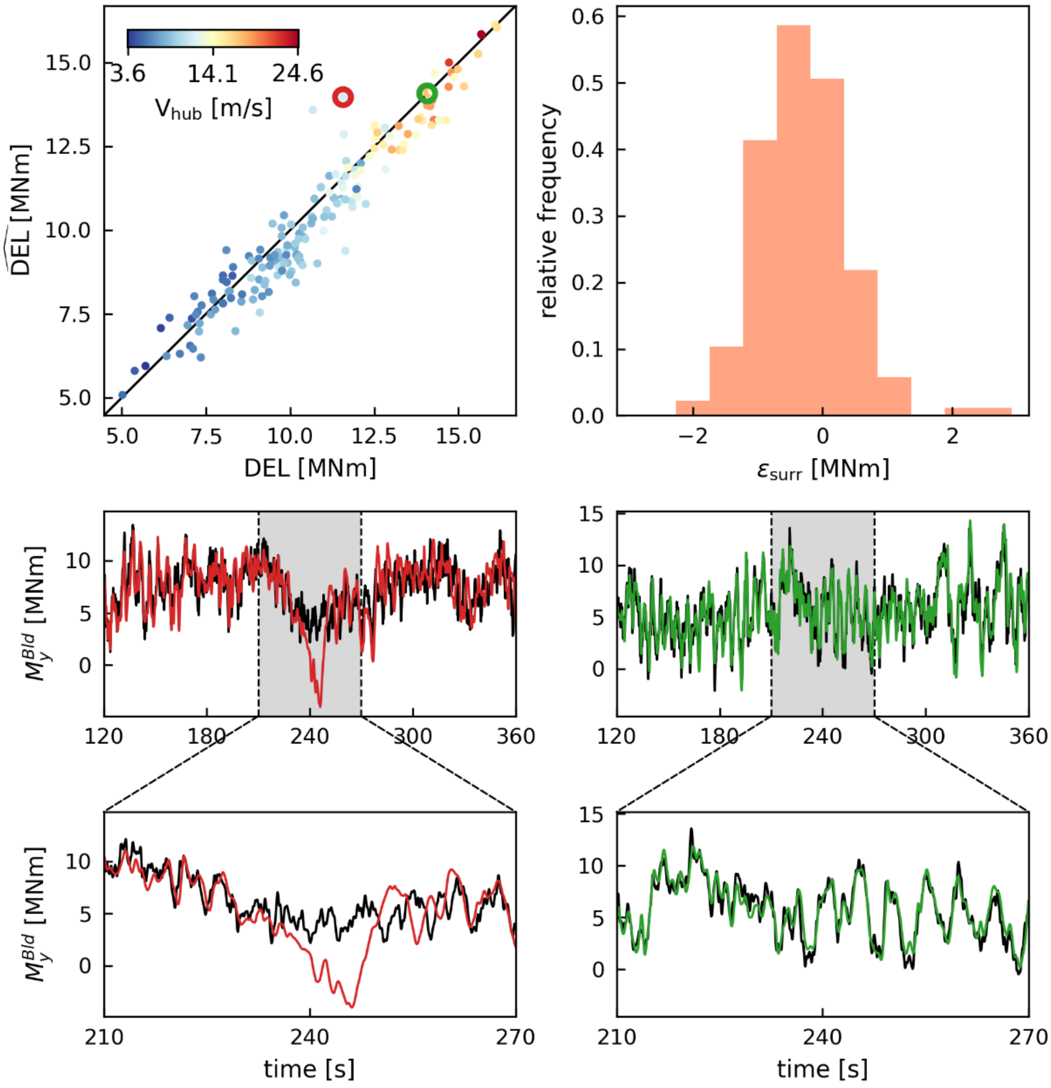

Figure 31: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the extreme turbulence model (ETM) validation dataset. Top left: scatter plot of true vs. surrogate DEL estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate DEL estimation sample. Bottom right: example of an accurate DEL estimation sample.

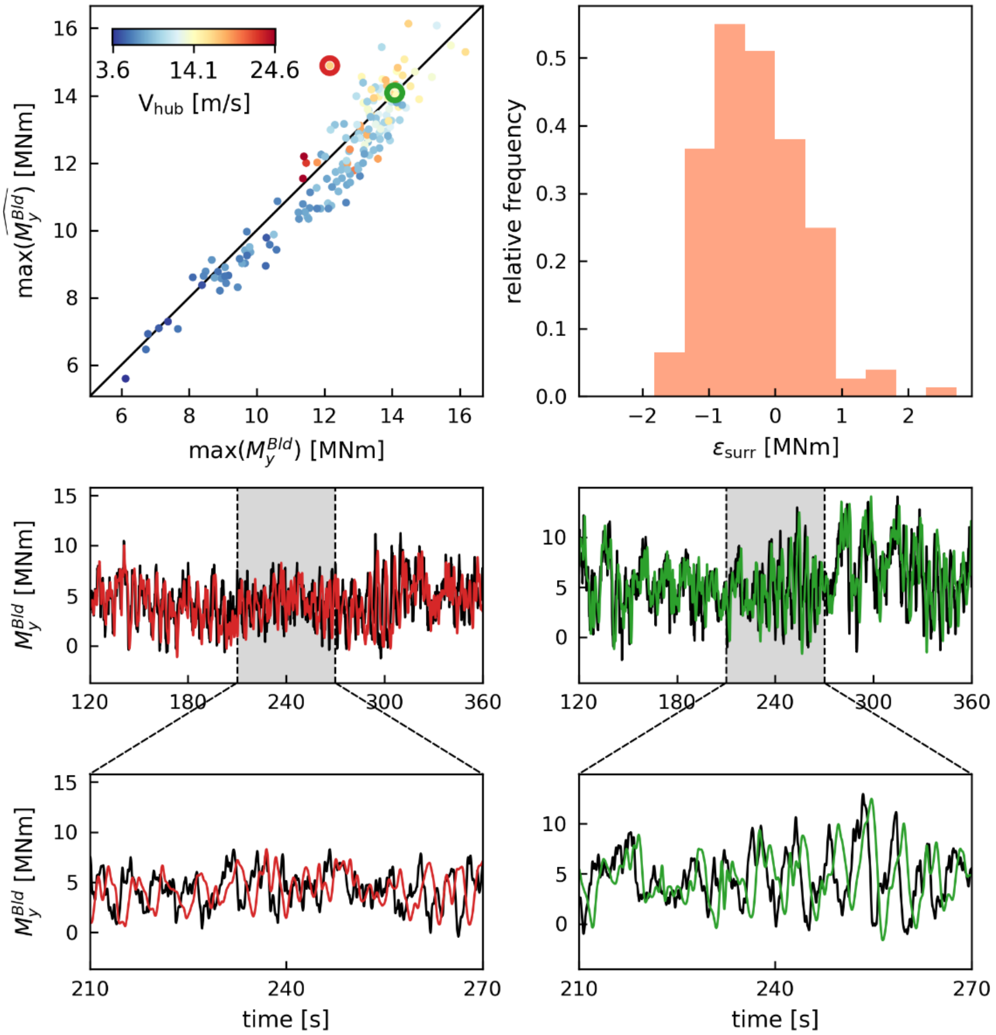

Figure 32: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the extreme turbulence model (ETM) validation dataset. Top left: scatter plot of true vs. surrogate maximum load estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate maximum load estimation sample. Bottom right: example of an accurate maximum load estimation sample.

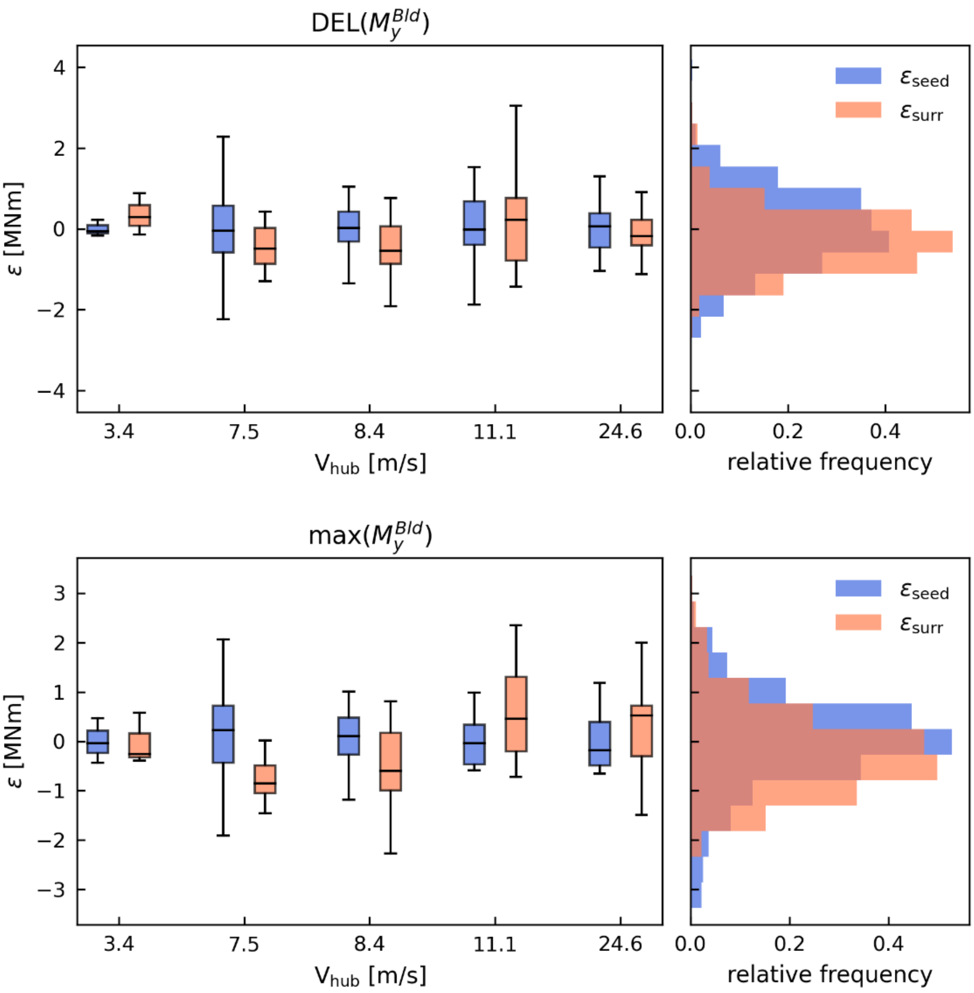

Figure 33: Surrogate model performance vs seed-to-seed variability on the flapwise blade root moment M_y^{Bld} on the extreme turbulence model (ETM) validation dataset. Top panel: DEL. Bottom panel: maximum load.

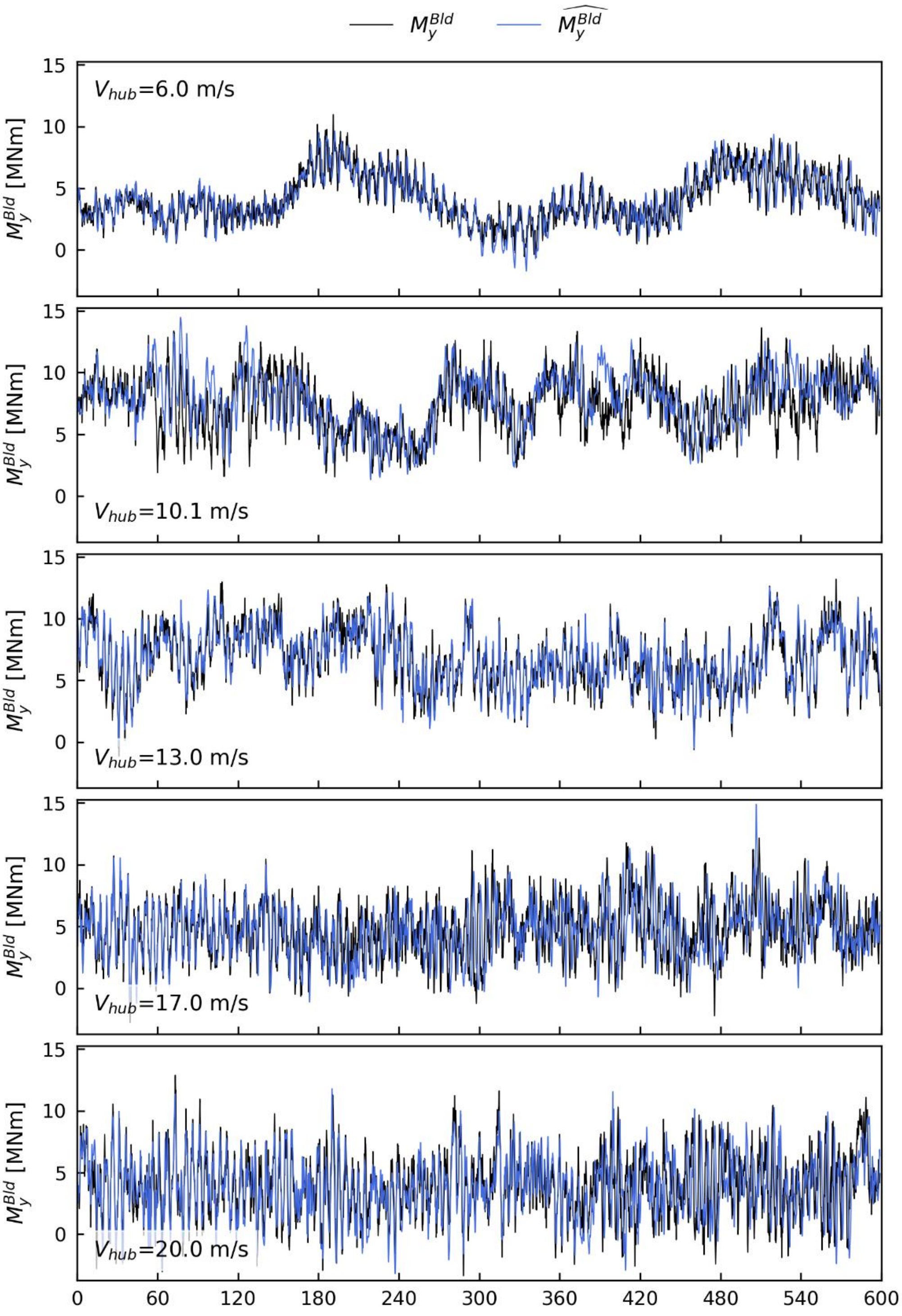

Figure 34: Performance of the surrogate model of the flapwise blade root moment M_y^{Bld} on the extreme turbulence model (ETM) on a selection of raw traces from the ETM validation set.

Finally, the surrogate behavior on the extreme operating gust (EOG) dataset also showcases a similar behavior, but with an overall larger dispersion. We report the benchmarks in **Figure 35** (DEL), **Figure 36** (maximum load), **Figure 37** (surrogate error vs seed-to-seed variability) and **Figure 38** (raw trace plots). Despite the larger variability, probably due to a relative underrepresentation of this type of scenario (extreme winds) in the training dataset, the overall performance of the surrogate is satisfactory, reproducing accurately the effects of the wind gust at its onset on virtually every trace.

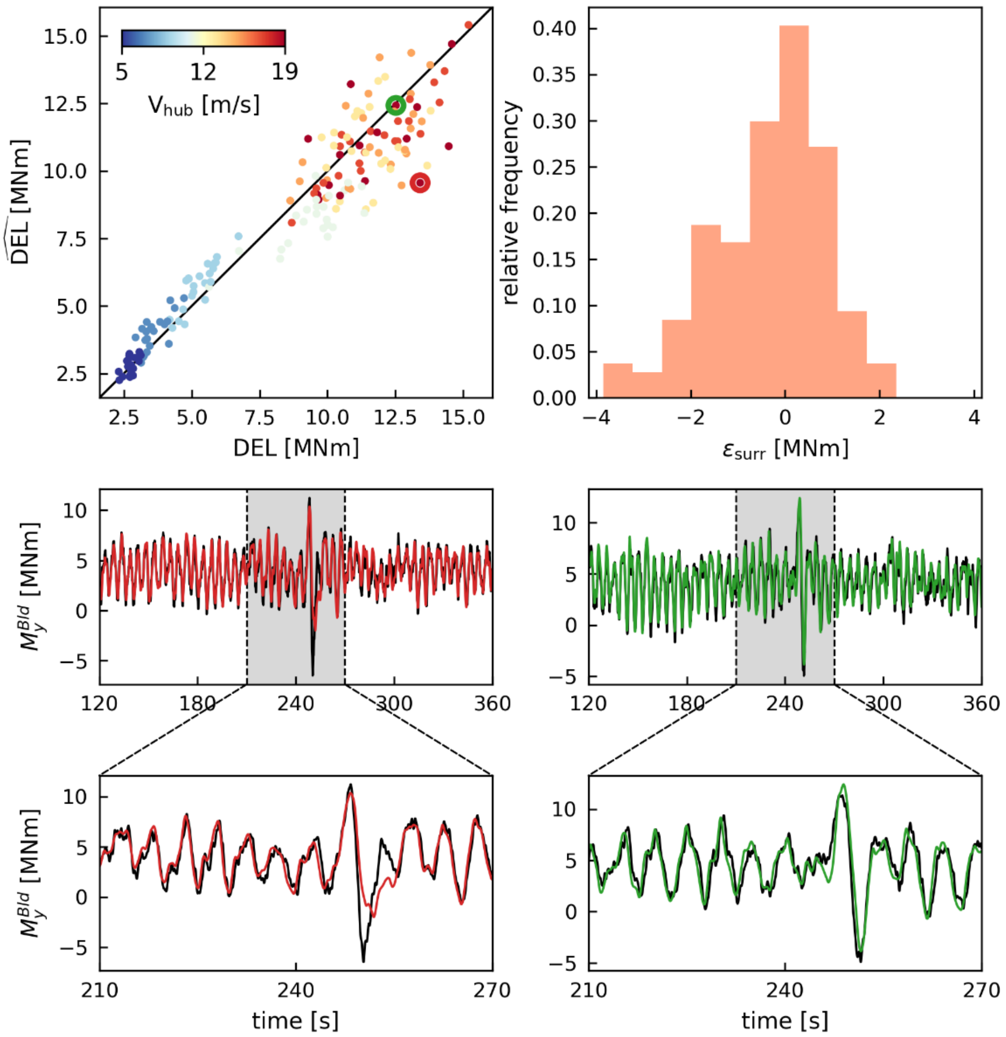

Figure 35: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the extreme operating gust (EOG) validation dataset. Top left: scatter plot of true vs. surrogate DEL estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate DEL estimation sample. Bottom right: example of an accurate DEL estimation sample.

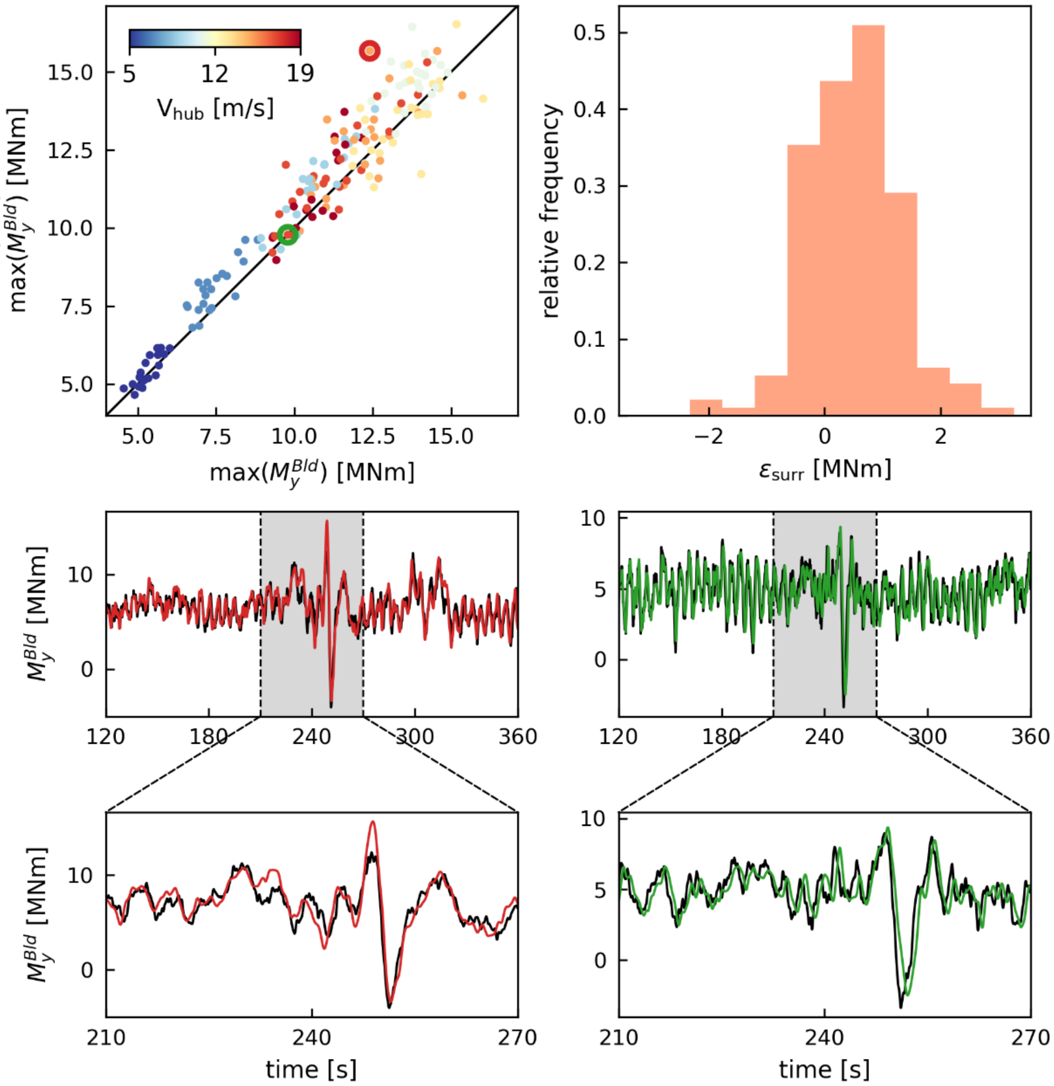

Figure 36: Surrogate model performance on the flapwise blade root moment M_y^{Bld} on the extreme operating gust (OEG) validation dataset. Top left: scatter plot of true vs. surrogate maximum load estimation. Top right: histogram of the root-mean-square error in MN. Bottom left: example of an inaccurate maximum load estimation sample. Bottom right: example of an accurate maximum load estimation sample.

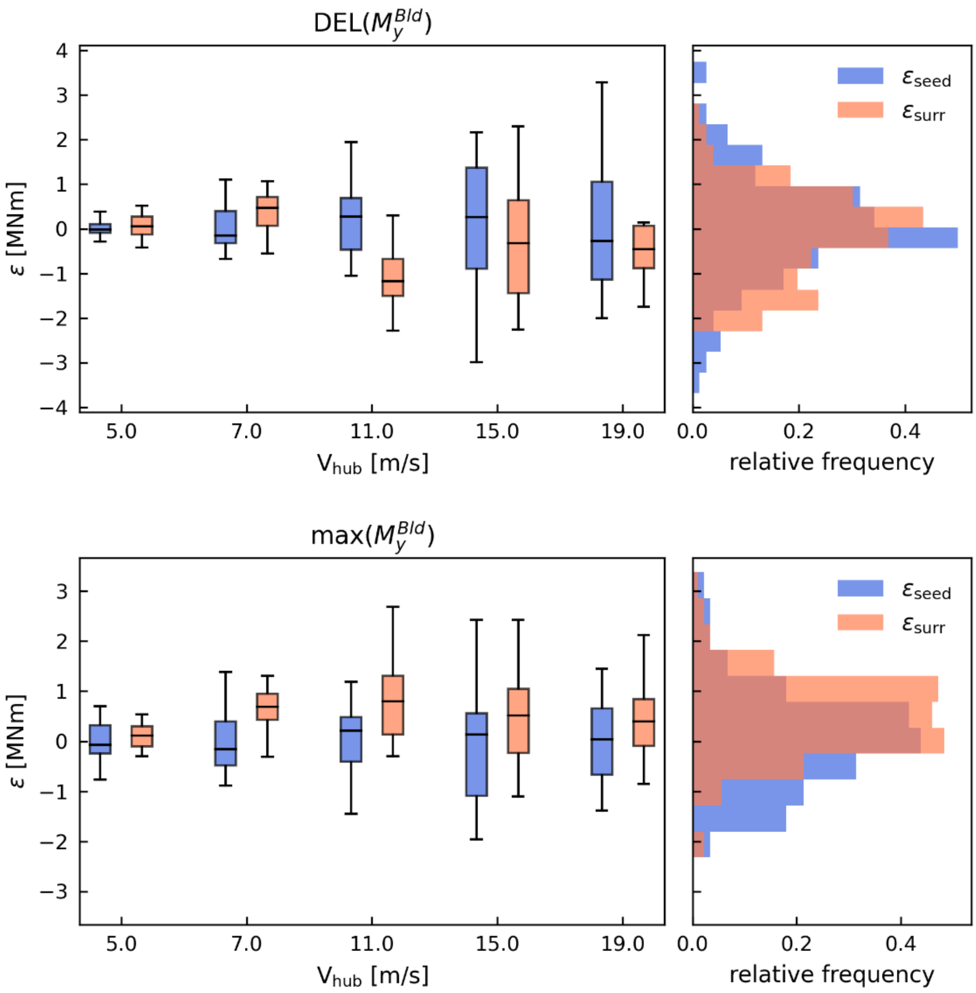

Figure 37: Surrogate model performance vs seed-to-seed variability on the flapwise blade root moment M_y^{Bld} on the extreme operating gust (EOG) validation dataset. Top panel: DEL. Bottom panel: maximum load.

Table 13: Summary of the surrogate model performance on the damage equivalent load of the flapwise blade root moment M_y^{Bld}. The coefficients of determination R^2 corresponds to the scatter plots in **Figure 27**, **Figure 31** and **Figure 35**. The mean of the surrogate error $\mu_{\varepsilon_{surr}}$, the standard deviation of the surrogate error $\sigma_{\varepsilon_{surr}}$ and the standard deviation of the seed-to-seed uncertainty $\sigma_{\varepsilon_{seed}}$ correspond to the histograms in **Figure 29**, **Figure 33** and **Figure 37**.

	DEL M_y^{Bld}				
	R^2	$\mu_{\varepsilon_{surr}}$	$\sigma_{\varepsilon_{surr}}$	$\sigma_{\varepsilon_{seed}}$	$\sigma_{\varepsilon_{surr}} / \sigma_{\varepsilon_{seed}}$
NTM	0.956	-0.276	0.418	0.513	0.815
ETM	0.905	-0.274	0.705	0.944	0.746
EOG	0.898	-0.191	1.007	1.155	0.872

Table 14: Summary of the surrogate model performance on the maximum value of the flapwise blade root moment M_y^{Bld}. The coefficients of determination R^2 corresponds to the scatter plots in **Figure 28**, **Figure 32** and **Figure 36**. The mean of the surrogate error $\mu_{\varepsilon_{surr}}$, the standard deviation of the surrogate error $\sigma_{\varepsilon_{surr}}$ and the standard deviation of the seed-to-seed uncertainty $\sigma_{\varepsilon_{seed}}$ correspond to the histograms in **Figure 29**, **Figure 33** and **Figure 37**.

	$\max(M_y^{\mathrm{Bld}})$				
	R^2	$\mu_{\varepsilon_{surr}}$	$\sigma_{\varepsilon_{surr}}$	$\sigma_{\varepsilon_{seed}}$	$\sigma_{\varepsilon_{surr}} / \sigma_{\varepsilon_{seed}}$
NTM	0.973	-0.221	0.411	0.488	0.843
ETM	0.852	-0.258	0.836	0.938	0.891
EOG	0.894	0.490	0.786	0.985	0.798

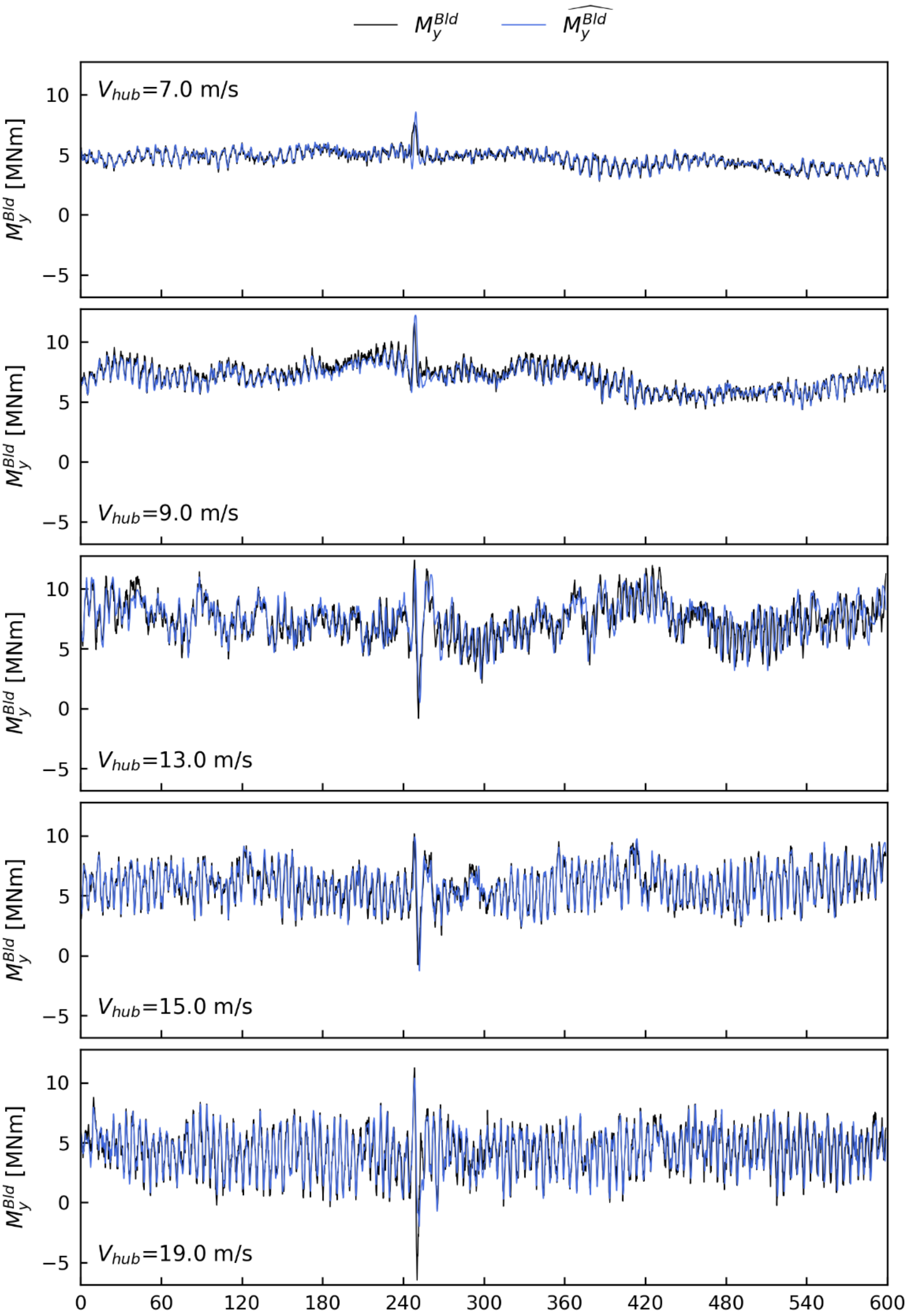

Figure 38: Performance of the surrogate model of the flapwise blade root moment M_y^{Bld} on the extreme operating gust (EOG) on a selection of raw traces from the EOG validation set.

3.5.4. Computational costs

In Section 3.4, we proposed a multi-step surrogate modelling strategy to model the transient response of the output quantities of a wind turbine. In this section we provide an overview of the computational costs of each step involved in constructing the surrogate model, as well as the costs of applying it on new turbulence boxes. One of the primary quantities of interest, $M_y^{\text{Bld}}(t)$, requires the blade pitch $\phi(t)$ and rotor speed $\omega(t)$ as exogenous inputs. For simplicity, we attribute the cost of creating and evaluating the surrogates for $\phi(t)$ and $\omega(t)$ entirely to the costs of the $M_y^{\text{Bld}}(t)$, although they can be reused as exogenous inputs to predict also other QoIs. The same reasoning applies to most of the pre-processing steps involved during the construction of the surrogate. Therefore, the reported computational costs serve as an upper bound and scale well with an increasing number of QoIs.

Table 15 shows the costs associated with preprocessing a single 10-minute simulation as used to create the M_y^{Bld}-model. For the prediction of new turbulence boxes, the steps are identical except that the true values for the QoIs are not loaded from disk. The time taken to train each of the auxiliary surrogates and the primary surrogate is given in **Table 16**

Table 15: Computation time in milliseconds associated with the preprocessing of a 10 minute turbulence box as performed during the construction and evaluation of the surrogate model for the flapwise blade root moment $M_y^{\text{Bld}}(t)$.

Step	**Time (ms)**
Read wind box from disk	164
Read QoIs from disk	40
Discrete cosine transform	73
Scaling	5
Interpolation	64

Table 16: Computation in seconds to train the surrogate model for the pitch $\phi(t)$, rotor speed $\omega(t)$ and the flapwise blade root moment $M_y^{\text{Bld}}(t)$. The configuration and number of training samples used corresponds to the low wind speed models listed in the **Table 8**, **Table 9**, and **Table 11**.

Model	**Time (s)**
ϕ-model	1.5
ω-model	7
M_y^{Bld}-model	14

Table 17: Computation time in milliseconds to evaluate the surrogate on 10^5 time steps (one full trajectory). With a sampling frequency of 160 Hz this is equivalent to a 10 min simulation.

Model	**Time (ms)**
ϕ-model	21
ω-model	57
M_y^{Bld}-model	334

3.6. Discussion and conclusions on the time series surrogate model

3.6.1. Performance analysis

According to the SMART deliverable D4.1 reported in Section 2.2, the two main measures of success for this deliverable are:

1. Surrogate modelling approaches are developed and have at least one order of magnitude better computational efficiency than the original aeroelastic modelling tool
2. Surrogate model uncertainty is not greater than the turbulence-induced statistical uncertainty (seed-to-seed uncertainty)

Based on the extensive benchmarking results presented in Sections 3.5.3 and 3.5.4, we are confident to claim that:

1. The proposed mNARX method is remarkably faster than the one order of magnitude requested by the SMART deliverable. Even in its current prototype/demonstrator stage, the speedup it provides is already $250\times$ faster than the full ASE simulator it is trained on.
2. While more complex to assess, due to the lack of a unique/suitable error measure for seed-to-seed variability of time series, we demonstrated that the accuracy of the proposed mNARX method is already sufficient for a number of applications in the context of probabilistic design of wind turbines. From our experience, we can certainly argue that the modelling uncertainty introduced by the surrogate is lower than that introduced by the discrepancy between different simulators (e.g. OpenFAST, Hawc2, Diego, etc.).
3. The reliance of mNARX on well-known and freely available technologies, such as autoregressive models and DFT makes it an ideal tool to share simulators between Hiperwind partners, without the need of complex technology transfers.

3.6.2. Portability

The entire codebase of mNARX is implemented with Python and based on the modules *NumPy*, *Scipy* and *Numba*, hence not requiring specific/commercial tools to be used. The surrogate model itself is trained on a relatively small set of full ASE simulations ($\mathcal{O}(10^2)$), and it can therefore in principle be easily retrained on different simulation codes (e.g. Hawc2, Diego or DeepLines Wind) and different wind conditions, both offshore and onshore, on a common business laptop.

3.6.3. Limitations/further work

The most obvious limitations of the study presented is that it was benchmarked only on an onshore wind turbine reference model. This limitation is due to the lack of availability to ETH of a proper ASE simulation code, as well as the lack of statistically meaningful models of wind- and wave-environmental conditions during the development stage of mNARX.

Nevertheless, the construction of mNARX is entirely non-intrusive, which means that additional, possibly time-dependent environmental conditions can easily be added to the input manifold $\boldsymbol{\zeta}(t)$ and treated as any of the other quantities. In particular, we expect that the presence of waves and especially that of a floating platform may require additional inputs such as wave elevations, as well as additional intermediate QoIs, such as *e.g.* platform pitch, roll and heave parameters. However, we have no reason to believe that the dynamics of the system could not be handled through the same methodological approach.

Finally, at the current demonstrator stage the mNARX parameters highlighted in **Table 4** are fine-tuned manually, one QoI at a time. However, we plan to fully automatize and optimize this process by means modern optimization techniques based on cross-validation error from the machine-learning and uncertainty quantification literature.

4. Bayesian Neural Networks (BNN)

4.1. Introduction and problem description

An adequate probabilistic design approach needs to consider all major sources of uncertainty, with as accurate quantification as possible. This is especially true for the purpose of reliability analysis, where due to the small target failure probabilities it is not only the bulk of the distributions, but their tails that are potentially of high importance[2]. At the same time, addressing a problem in a probabilistic way means we are dealing with a broad range of scenarios to cover the entire probability space. This will typically require many function calls to numerical models that can be computationally expensive. A common approach to reduce the computational intensity is training surrogate models that map the behavior of the complex models but are significantly faster. This increase in efficiency comes at the expense of introducing additional model uncertainty due to the potential inaccuracies in the surrogate model.

Regression models based on Feedforward Neural Networks (FNN) (Goodfellow, et al., 2016) are a popular surrogate modelling approach (Dimitrov, 2019), (Schröder, et al., 2018) that has some well-known advantages:

- Very efficient training process due to the availability of back-propagation algorithms that utilize the chain rule for computing analytical derivatives;
- Flexibility combined with robustness – FNNs can theoretically approximate any function;
- Compared to other regression model approaches such as spectral surrogates (e.g., polynomial models) or Gaussian process regression, the computational performance of FNNs is less sensitive to the number of input variables.

On the other hand, FNN models have some disadvantages such as:

- They are so-called "black box" models, meaning that their parameters have no physical meaning and it is harder to interpret the model behavior;
- FNN may require large amounts of data to train properly, otherwise they can overfit and interpret noise as patterns in the data;
- Due to the highly connected architecture of the FNNs, it is not possible to perform model reduction by eliminating unnecessary terms as what is the usual practice with simpler models.

The standard FNN-based surrogate models provide a deterministic prediction, and the uncertainty is not a direct output of the models. However, the so-called Bayesian Neural Networks (BNNs) (Blundell, et al., 2015) are a modification of FNNs where the deterministic model parameters are replaced with probabilistic ones, resulting in a model that can give uncertainty estimates. BNNs still have limited use in probabilistic design and reliability analysis problems – especially in the wind energy domain. This means there is lack of experience regarding BNN model performance on specific issues such as tail distribution prediction, model uncertainty when modelling complex regression problems, and other. The work described in this section focuses on answering the following research questions:

1) How can we use BNNs to build accurate wind turbine load and power output surrogates that also represent the uncertainties in an adequate way?
2) How adequate are BNN models for specific probabilistic design applications such as a) predicting the tail distribution of load extremes and b) modelling the full probability distribution of aggregate quantities (such as fatigue loads) including uncertainties?

[2] This is mainly the case when a single or a few random input variables dominate the behavior of the limit state function.

3) What is the performance of BNNs in comparison of other alternatives such as running an ensemble of deterministic models?

4.1.1. Probabilistic regression models

Let us consider a physical system whose behavior can be characterized by random inputs $\mathbf{x}$ and outputs $\mathbf{y}$ (where the bold symbols denote vectors, to indicate that the system can have multiple inputs and multiple outputs as well). The true relationship between $\mathbf{x}$ and $\mathbf{y}$ is represented by a hypothetical function $\tilde{g}(\mathbf{x})$, which however may be too complex to characterize accurately. Therefore, we typically accept that a real-world model will not achieve perfect mapping. The actual model function $g(\mathbf{x})$ will include a finite set of parameters $\boldsymbol{\theta}$, and will have a certain error:

$$\boldsymbol{y} = \tilde{g}(\boldsymbol{x}) = g(\boldsymbol{x}, \boldsymbol{\theta}) + \ \boldsymbol{\varepsilon}_g \tag{19}$$

The term $\boldsymbol{\varepsilon}_g$ is typically called model error and is an example of ***epistemic*** uncertainty - i.e., uncertainty arising due to lack of knowledge that can be reduced by model improvement. In addition to the model, there can be uncertainties in both the input and output variables. These are typically a combination of ***aleatory*** uncertainties (due to random phenomena we cannot control), and epistemic ones such as measurement uncertainties. Let us denote the measured (or predicted) values of the inputs and outputs by $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ respectively. They relate to the true values by $\mathbf{x} = \hat{\mathbf{x}} + \boldsymbol{\varepsilon}_x$ and $\mathbf{y} = \hat{\mathbf{y}} + \boldsymbol{\varepsilon}_y$, leading to the following generic probabilistic model equation:

$$\hat{\boldsymbol{y}} + \boldsymbol{\varepsilon}_y = g(\hat{\boldsymbol{x}} + \boldsymbol{\varepsilon}_x, \boldsymbol{\theta}) + \ \boldsymbol{\varepsilon}_g \tag{20}$$

Within the above definition, multiple model implementations are possible that consider a subset of the uncertainty factors and as a result can serve different purposes. Below, four specific approaches are outlined, where one is the baseline, fully deterministic approach, and each of the remaining three is considering one uncertainty term - $\boldsymbol{\varepsilon}_x$, $\boldsymbol{\varepsilon}_y$, and $\boldsymbol{\varepsilon}_g$, respectively.

Figure 39 shows examples that illustrate the different uncertainty modelling approaches on a toy problem. In each plot, the blue dots represent a reference probabilistic data set following the relationship $y = 0.5x^3 - 2x^2 + 30x - 10 + \varepsilon$, where ε is a random Gaussian variable with mean of 0 and standard deviation of 100. The red lines and circles represent model predictions. The random term ε is representative of aleatory uncertainty, however any model trained on a sample of (x, y) will also have an epistemic (model) uncertainty because the training set is of finite size. The details and properties of each modeling approach are elaborated below.

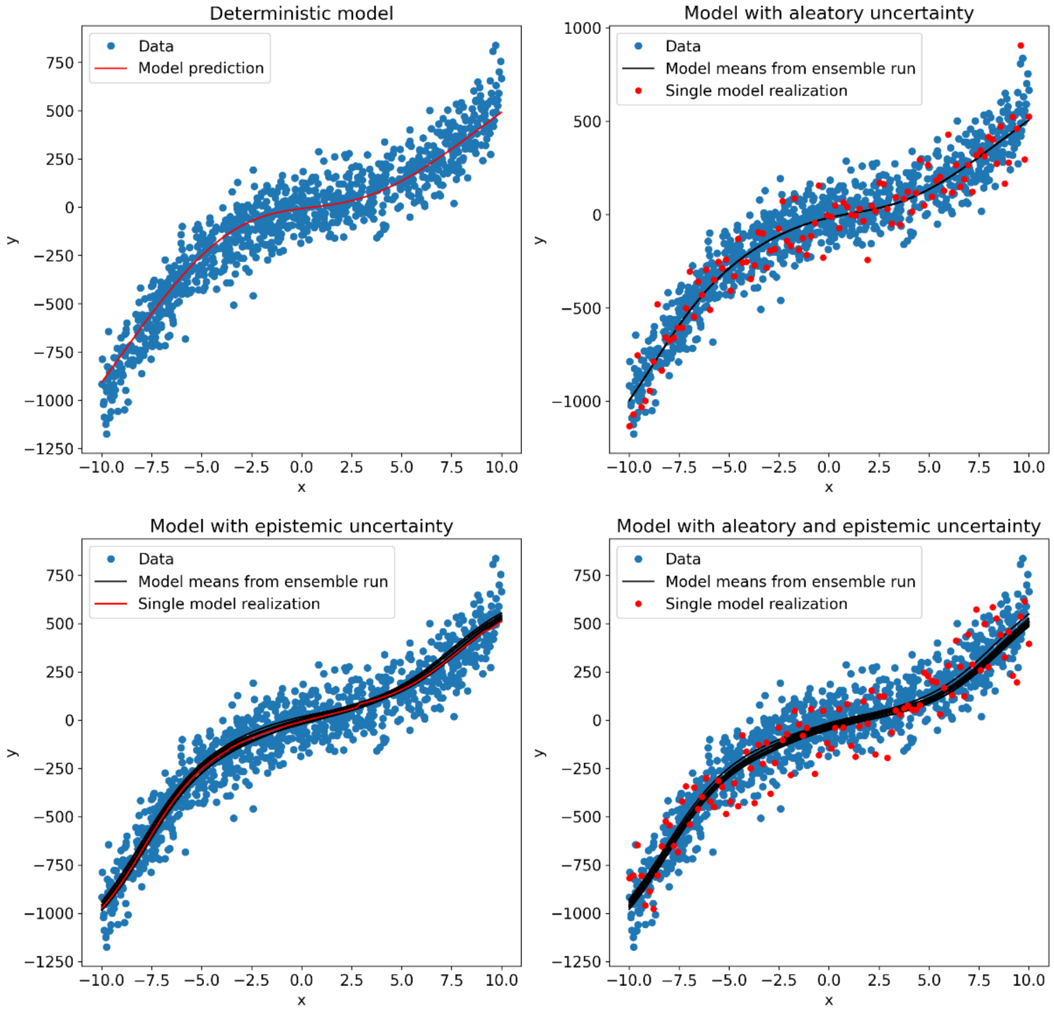

Figure 39: Examples of models considering different uncertainty sources. Top left: deterministic model not considering any uncertainty, top right: model considering all uncertainty as aleatory, bottom left: model taking only epistemic (model) uncertainty into account, bottom right: a model that considers both epistemic and aleatory uncertainties.

4.1.2. Deterministic model - Input uncertainty propagation

For the vast majority of numerical and analytical models, the parameters $\boldsymbol{\theta}$ are deterministic, meaning that the function $g(\hat{\mathbf{x}} + \boldsymbol{\varepsilon}_x, \boldsymbol{\theta})$ is also deterministic, and a given set of input values will always produce the same outputs – i.e., the function $g(\hat{\mathbf{x}}, \boldsymbol{\theta})$ provides unique and invariant mapping from $\hat{\mathbf{x}}$ to $\hat{\mathbf{y}}$. Among many applications of such models, one particular use case is uncertainty propagation, where the inputs $\mathbf{x}$ are sampled in a way representative of the distribution of the uncertainty terms $\boldsymbol{\varepsilon}_x$. As a result, the variation in the outputs represents the effect of the input uncertainty on the outputs. In a case where the uncertainty propagation is the only intended result, the deterministic behavior of $g(\hat{\mathbf{x}}, \boldsymbol{\theta})$ is a requirement as it eliminates the effect of other uncertainty sources.

A deterministic model will only predict the mean values, $\bar{\mathbf{y}} = g(\hat{\mathbf{x}}, \boldsymbol{\theta})$ where $\bar{\mathbf{y}} = E(\mathbf{y}\,|\mathbf{x})$ and will not capture the variability due to the aleatory uncertainty. The overall uncertainty can be estimated by analysis of the model residuals (the unexplained variance), but the source of uncertainty and its dependence on the inputs cannot be identified.

Training such a deterministic model involves minimizing the difference between the model predictions and the true outputs. This amounts to finding the values of $\boldsymbol{\theta}$ that lead to the model error $\boldsymbol{\varepsilon}_g$ having approximately zero mean and the smallest possible variance. The most commonly used training approach is least-squares minimization. An example of the modelling outcome is shown in **Figure 39,** top right plot.

4.1.3. Models with probabilistic outputs (stochastic simulators)

Repeated experiments (e.g. repeatedly sampling a numerical model or acquiring measurements) at the same input values will give a variation in $\mathbf{y}$ which is representative of the output (aleatory) uncertainty, $\boldsymbol{\varepsilon}_y$. A simple way to characterize this uncertainty is to consider $\mathbf{y} = \hat{\mathbf{y}} + \boldsymbol{\varepsilon}_y$ as a random variable following a given (joint) probability distribution, and let the model predict the distribution parameters of $\mathbf{y}$ rather than the actual values (Zhu & Sudret, 2020), (Zhu & Sudret, 2021). As the outputs are the only probabilistic element, models with the discussed definition treat both the model and the output uncertainties as a single factor and it is not possible to directly distinguish between the uncertainty sources. However, if the training sample becomes large and the model has adequate predictive capability, the model uncertainty will get smaller (as seen on e.g. **Figure 39** and **Figure 42**) and the aleatory uncertainty will dominate. We list two practical approaches to train such a model:

- Run repeated experiments at each sample point in $\hat{\mathbf{x}}$, thus obtaining a sample of the distribution of $\hat{\mathbf{y}}|\hat{\mathbf{x}}$. Train a deterministic model for $\boldsymbol{\theta}_{\mathbf{y}}|\hat{\mathbf{x}}$ that are the distribution parameters of $\hat{\mathbf{y}}|\hat{\mathbf{x}}$, as function of the inputs $\hat{\mathbf{x}}$. Probabilistic sampling from the surrogate model at a given point $\tilde{\mathbf{x}}$ takes place by 1) calling the model to obtain an estimate of the conditional distribution parameters $\boldsymbol{\theta}_{\mathbf{y}}|\tilde{\mathbf{x}}$, and 2) generating a random sample from that conditional distribution. With this approach, the same workflow as with the completely deterministic models can be maintained because the training can be done using a least-squares minimization approach.
- Alternatively, define a likelihood function on the probability distribution of $\mathbf{y}|\hat{\mathbf{x}}$, which under the assumption that we want to train an unbiased model (i.e., the mean of $\boldsymbol{\varepsilon}_g$ is zero), will be equivalent to the probability distribution of the model error $\boldsymbol{\varepsilon}_g|\hat{\mathbf{x}}$. Model training aims at finding the set of parameters that maximize the likelihood function. Note that the total set of parameters is the combination of $(\boldsymbol{\theta}, \boldsymbol{\theta}_{\mathbf{y}})$, i.e. both the model parameters and the distribution parameters of $\mathbf{y}|\hat{\mathbf{x}}$. With this approach, repeated experiments at each sample point are not necessary since the distribution parameters $\boldsymbol{\theta}_{\mathbf{y}}$ do not need to be explicitly observed beforehand, they are determined within the training process. In a Bayesian Neural Network (BNN) architecture, this modelling approach can be achieved by introducing a distribution layer (a layer with coefficients corresponding to the parameters of a probability distribution) as the output layer of the network.

Figure 39, top-right, shows an example of the behavior of a model with probabilistic outputs representing predominantly aleatory uncertainty.

4.1.4. Considering model uncertainty – by ensemble models and models with probabilistic parameters

Using different datasets to train multiple deterministic regression models with identical input and output variables results in a model ensemble that can give an estimate of the model uncertainty. This estimate would mainly account for the model (statistical) uncertainty due to the limited amount of data used for model training – but will not account for other factors such as the model functions not being sufficient to represent the patterns in the data. Consequently, this estimate of the model uncertainty will tend to zero when the amount of training data tends to infinity and is drawn from a stationary distribution. One way to construct a model ensemble from relatively limited amount of data is the bootstrapping technique. Further, in order to isolate the model uncertainty effect from that of aleatory (output) uncertainties, the training data may consist of multiple realizations at the same input value combinations. Then the actual model training uses the mean of all realizations at each sample point. This model formulation reads:

$$\bar{\mathbf{y}} = g_i(\hat{\mathbf{x}}, \boldsymbol{\theta}_i) + \boldsymbol{\varepsilon}_{g,i}, \quad i = 1 \ldots N_{ensemble}$$

where $\boldsymbol{\varepsilon}_{g,i}$ are the residuals of the i^{th} model in predicting the outputs averaged over all realizations, $\hat{\mathbf{y}}$.The estimate of model uncertainty can then be obtained by characterizing the distribution of $\boldsymbol{\varepsilon}_g$.

Another way of taking model uncertainty into account is making the model parameters, $\boldsymbol{\theta}$, probabilistic, represented by a joint distribution. The parameters of this joint distribution can be determined by likelihood maximization. For smaller problems (a few computational parameters) it would be the full distribution of $\boldsymbol{\theta}$ that could be inferred, however this is intractable for large neural networks where the large number of parameters would among other things require inferring a very large matrix of cross-covariance terms. A common solution (as proposed by (Blundell, et al., 2015)) is to only infer the mean values of the parameters as well as the main diagonal of the covariance matrix – meaning that all parameters are treated as independent, which is a fair assumption for neural networks where the parameters have no physical meaning. This is the approach used with BNNs, where probabilistic parameters can be included by introducing “variational” layers in the network.

Figure 39, bottom-left, shows an example of a model ensemble that is used to demonstrate the effect of model uncertainty.

Finally, it is also possible to have BNN models that simultaneously take into account epistemic and aleatory uncertainty. This is achieved by introducing both variational layers and probabilistic output layers in the network. This model implementation is explained in more details in the following sections.

4.2. Study design and data generation

In order to study the behavior of BNNs as probabilistic regression surrogate models, we define a use case based on wind turbine aeroelastic load simulations (hereafter called load simulations for brevity). Load simulations are used for wind turbine design and site suitability assessments, which requires simulating a broad range of design scenarios (typically in the order of thousands of 10-minute time series simulations per design iteration). Since the load simulations are relatively costly from a computational point of view, there are situations where it is desirable and feasible to replace them with a surrogate model. Examples include site suitability assessment, wind farm layout and control optimization. Since the wind turbine design criteria include both fatigue limit state (FLS) and ultimate limit state (ULS) criteria, it means that both aggregated statistics (such as damage-equivalent fatigue loads, DEL) and extremes need to be quantified. A useful surrogate model (or models) would need to have high accuracy in predicting the full distribution of loads in order to get accurate numerical integrations over the joint probability distribution of environmental input conditions - which are needed to evaluate design criteria over the turbine lifetime. In addition, for adequate assessment of the ultimate limit states and potentially using those in a reliability analysis, the models need a good prediction of the tails of the load distributions.

The IEC 61400-1 design standard provides a reference definition of the scenarios and the environmental conditions to be used for wind turbine design. For onshore sites, the major driving environmental factors are the wind speed and the turbulence, characterized by a joint probability distribution. Wind shear is also considered - defined as an exponential function with a deterministic exponent (either 0.11 or 0.2). When addressing site-specific problems, additional variables or more detailed modelling of a given variable may come into play depending on the specific conditions at the site. As a realistic use case, we choose four environmental variables: wind speed, turbulence, wind shear and air density. The joint distribution of wind speed vs. turbulence is as defined by the IEC61400-1 standard, while the wind shear is taken as conditional on wind speed using the formula defined in (Dimitrov, et al., 2015). Air density is considered independent and following a Gaussian distribution. The exact distribution parameters are outlined below.

- Wind speed u: a truncated Weibull distribution with parameters $A = 11.28$ and $k = 2$, and bounded between the cut-in and cut-out wind speed of the turbine under consideration. The Weibull distribution parameters correspond to IEC wind speed class I (mean wind speed of 10m/s).
- Turbulence σ_u: the standard deviation of wind speed, σ_u, is considered Lognormally distributed, conditional on the wind speed. The mean and standard deviation of the lognormal distribution are defined with the formulas $\mu_{\sigma_u}|u = 0.14(0.75u + 3.8)$ and $\sigma_{\sigma_u} = 1.4 \cdot 0.14$, respectively, which corresponds to IEC turbulence class B.
- Wind shear exponent α: considered Gaussian, with mean and standard deviation conditional on the wind speed: $\mu_\alpha|u = 0.088(\ln(u) - 1)$ and $\sigma_\alpha|u = 1/u$, respectively. This is the expression recommended by (Dimitrov, et al., 2015), (Kelly, et al., 2014).
- Air density ρ: independent Gaussian with mean 1.225 and standard deviation of 0.05

A total of 2,000 random samples are generated from the above distributions. The sample generation is done in a two steps process: 1) a set of uniform, i.i.d. variables ranging from 0 to 1 is drawn from a Halton quasi-random sequence, and 2) the physical values with the correct joint distribution are obtained by a Rosenblatt transformation. The resulting dataset is visualized in **Figure 40**.

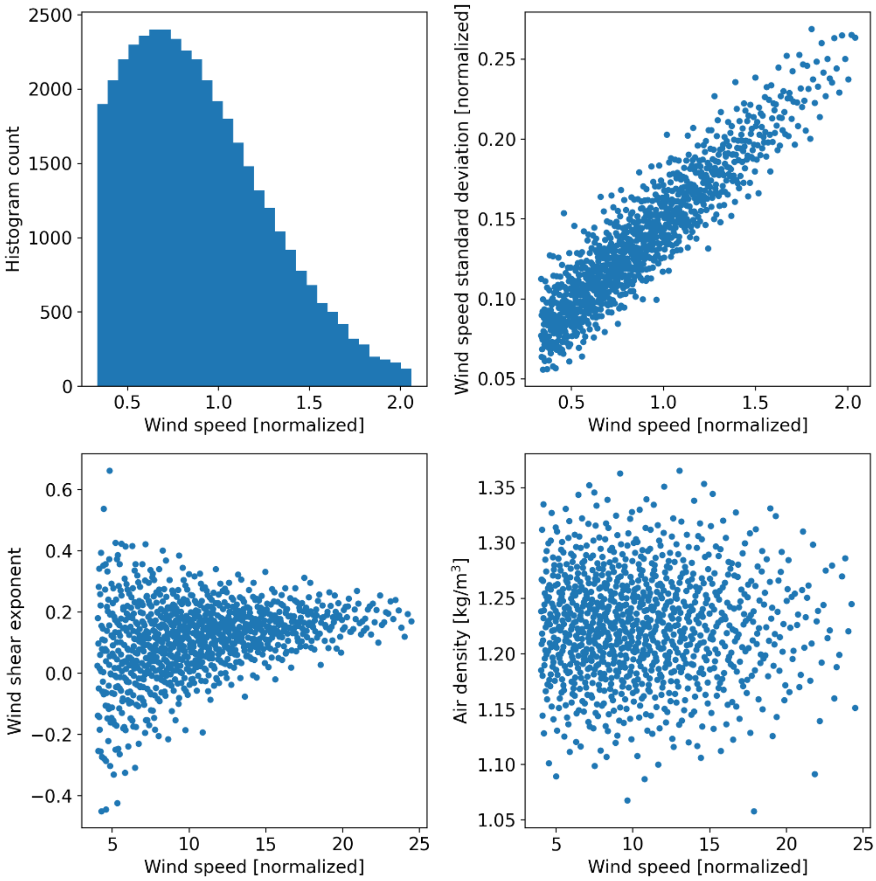

Figure 40: Scatter plots of the environmental condition distributions – used as load simulation and surrogate model inputs. Top left: wind speed histogram. Top right: turbulence as function of wind speed. Bottom left: wind shear exponent vs. wind speed. Bottom right: Air density vs. wind speed.

An advantage of using numerical simulations in the present problem is that there is no measurement uncertainty, meaning that the "true" values of the inputs and outputs are known, and $\mathbf{y} \equiv \hat{\mathbf{y}}$ and $\mathbf{x} \equiv \hat{\mathbf{x}}$. This eliminates the $\boldsymbol{\varepsilon}_x$ term from Eq. (2), as well as the part of $\boldsymbol{\varepsilon}_y$ that is due to measurement uncertainty.

Load simulations use turbulent wind field realizations (so-called "turbulence boxes") as inputs. Since turbulence boxes are random realizations and external to the actual wind turbine system that we are modelling, their effect can be considered as aleatory uncertainty (appearing as part of the uncertainty on the output, $\boldsymbol{\varepsilon}_y$). This gives the possibility to design studies where the effect of aleatory uncertainty can be quantified and/or eliminated, and considered separately from the effect of model uncertainties. In the present study, 20 different turbulence realizations (seeds) are used at each sampling point. Since the variation in the load outputs with respect to seed-to-seed uncertainty follows the law of large numbers, using 20 seeds means that the standard deviation of the mean load estimates is reduced by a factor of $\sqrt{20} = 4.47$.

Based on the above design of experiment, 40,000 load simulations (20 seeds per sample point over 2,000 sample points) are carried out on a multi-megawatt offshore wind turbine model using the Hawc2 aeroelastic simulation tool. All simulations are with 10-minute duration and represent operation under normal (power production) conditions. **Figure 41** shows an overview of the statistics from the load simulation results for several channels of interest.

The surrogate modelling effort focuses on three groups of signals divided according to the type of statistics that are of interest:

1) Damage-equivalent fatigue loads (DELs) where quality of representation of the entire probability distribution is of interest are considered for 6 signals: tower base fore-aft (M_x) and side-to-side (M_y) bending moments, blade root flapwise (M_x) and side-to-side (M_y) bending moments, tower top yaw (torsion) moment M_z, and shaft torsion moment, M_z;
2) Extremes, where the prediction of the distribution tails is of highest importance, for 4 signals: tower base fore-aft (M_x) and side-to-side (M_y) bending moments, blade root flapwise (M_x) and side-to-side (M_y) bending moments;
3) Mean values of the shaft torsion M_z and for the electrical power, blade pitch, and rotor rotational velocity. Just as with the DELs, for the mean values it is the full probability distribution over the full range of environmental inputs that is of importance.

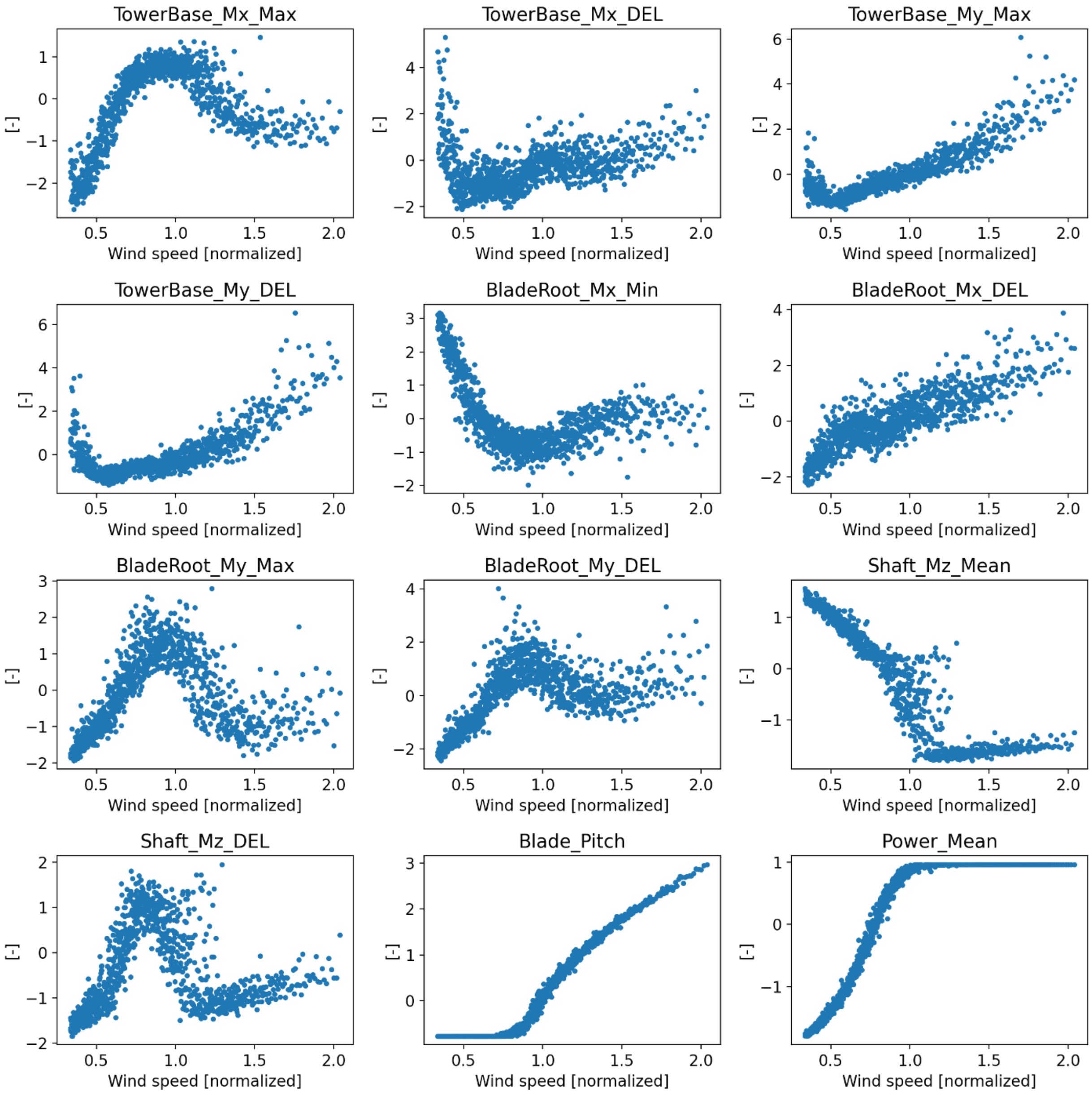

Figure 41: Overview of the outcomes from the aerodynamic load simulations. Scatter plots show normalized 10-minute load statistics vs. mean wind speed.

4.3. Ensemble models of deterministic neural networks

As discussed in Section 4.1, one way to consider the statistical uncertainty due to limited training data is to train an ensemble of deterministic models. In order to avoid the effect of aleatory uncertainty (which in our load simulation example is represented by the realization-to-realization uncertainty in the turbulence boxes), the ensemble model has to be trained on results averaged over all realizations at each sample point. **Figure 42**, left illustrates the effect the averaging over multiple realizations on the variation in blade root flapwise bending moments as function of wind speed. The scatter in the averaged loads (red dots) is less than the scatter from single realizations (black dots) showing that some of the uncertainty has been reduced. A large part of the remaining scatter is due to the influence of other variables (turbulence, wind shear and air density) that are not in the plot axes.

Taking the average $\bar{\mathbf{y}}$ over the 20 realizations simulated at each sample point $\mathbf{x}$, we obtain a dataset of 2,000 input/output pairs. 1,600 of the data points are designated as a training set. An ensemble of $N_{ensemble} = 15$ FNN models are trained, by drawing 15 bootstrapped samples from the training set. Each sample is drawn by generating a set of random integers in the range of 1 to 1,600, followed by selecting the $(\mathbf{x}, \bar{\mathbf{y}})$ data pairs whose location in the training set corresponds to the randomly generated integers. Each sample has a size of 1,600 which is equal to the entire training set size, however the samples are not identical as the data are drawn while allowing repetitions of the indexes. The ensemble model is implemented with the Hipersim package (Dimitrov, 2022), that includes FNN model capabilities. The individual FNN models have two hidden layers with 20 units each. According to earlier experience (Schröder, et al., 2018), this model size is adequate for the current purpose, and changes within a certain range do not have a significant influence on the model performance. The activation function is tanh (hyperbolic tangent). The training algorithm used is adam (Kingma & Ba, 2017), with learning rate of 0.001, and training takes place over 100 epochs.

Model testing is carried out on the $N_{test} = 400$ data points that are not used in the training. Predictions are obtained from each of the models in the ensemble, and the model residuals are computed as:

$$\varepsilon_{g,ij} = \bar{\mathbf{y}}_j - g_i(\mathbf{x}_{ij}, \boldsymbol{\theta}_i), \qquad i = 1 \dots N_{ensemble}, j = 1 \dots N_{test}$$

For each sample point $\mathbf{x}_j$, the model uncertainty is characterized by the standard deviation $\sigma_{\varepsilon_{g,j}}$ of the residuals $\varepsilon_{g,ij}$ over all indexes $i = 1 \dots N_{ensemble}$ and for the same value of j. A histogram of these uncertainties over the test data set is shown in **Figure 42**, center plot, with orange bars, for the blade root flapwise bending moment extremes.

It is relevant to compare the magnitude of the epistemic uncertainty with that of the aleatory uncertainty caused by the realization-to-realization variations in the load outputs. For that purpose, we evaluate the uncertainty in the individual simulation outcomes $\mathbf{y}$ as well as for the averaged values, $\bar{\mathbf{y}}$:

$$\sigma_{\varepsilon_y} = \sqrt{\frac{\sum_{k=1}^{N_k} (\mathbf{y}_k - \bar{\mathbf{y}})^2}{N_k}}, \qquad \sigma_{\varepsilon_{\bar{y}}} = \frac{\sigma_{\varepsilon_y}}{\sqrt{N_k}},$$

where N_k is the number of realizations. The histograms of σ_{ε_y} and $\sigma_{\varepsilon_{\bar{y}}}$ are plotted in **Figure 42**, center plot, with blue and green bars respectively. The realization-to-realization uncertainty is seen to cause much larger variation in the loads compared to the model uncertainty, however averaging over multiple realizations bring the model uncertainty and the aleatory uncertainty to similar magnitudes. This outcome however is just a snapshot of the uncertainty magnitudes for this specific experimental design. As mentioned earlier, the realization-to-realization uncertainty in the mean $\bar{\mathbf{y}}$ decreases with increasing number of independent realizations, while the model uncertainty evaluated here is dependent on the training sample size and decreases with increased number of training points. The latter is illustrated in **Figure 43**, which shows the distribution of the model uncertainty for three different training set sizes. Finally, given that the aleatory and epistemic uncertainty discussed in the present problem have different sources, it is expected that they are uncorrelated. This is confirmed by the right-hand-side plot in **Figure 42**, which does not indicate any dependence between the two uncertainty types. The overall significance of these observations is that both increasing the sample size and the number of realizations per sampling point should be beneficial in terms of uncertainty reduction.

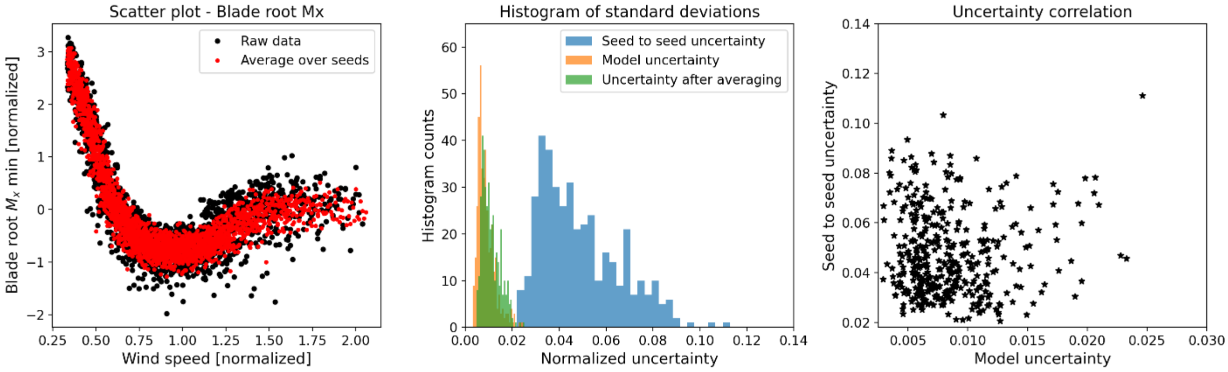

Figure 42: Comparison of epistemic (model) uncertainty and aleatory (realization-to-realization) uncertainty effects for extreme blade root flapwise bending moments. Left: scatter plots raw data including aleatory uncertainty and averages over multiple seeds (realizations) at each sample point; Center: histograms of the model uncertainty compared to the realization-to-realization uncertainty; Right: correlation between the model uncertainty and the realization-to-realization uncertainty.

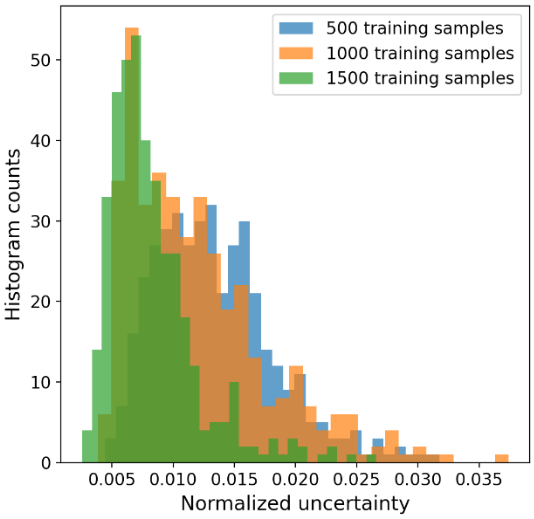

Figure 43: Epistemic uncertainty distributions for three different training sample sizes

4.4. BNN models with epistemic and aleatory uncertainty

Bayesian inference (Tiao & Box, 1992) is a statistical model calibration technique where the main goal is to determine (infer) the joint probability distribution of the model parameters $\boldsymbol{\theta}$ that can best explain a set of observations. As the name suggests, this approach makes use of the Bayes theorem, and the posterior (updated according to observations) distribution $p(\boldsymbol{\theta}|\mathbf{y})$ is obtained by maximizing a likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ combined with a prior assumption about the marginal distribution $p(\boldsymbol{\theta})$:

$$p(\boldsymbol{\theta}|\mathbf{y}) = cp(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

where c is a normalization coefficient needed to ensure that $p(\boldsymbol{\theta}|\mathbf{y})$ is a valid probability distribution with $\int p(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} = 1$.

Bayesian inference is especially useful for applications where the models are based on analytical functions or numerical models with only a few tunable parameters (Conti, et al., 2021), (Dimitrov, et al., 2016). For such problems it is relatively straightforward to infer the complete posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ including terms that describe the dependence between model parameters such as e.g. the off-diagonal terms of the covariance matrix $\boldsymbol{\Sigma}_{\theta\theta}$. However, in many function approximation models such as Neural Networks, the number of model parameters is very large which makes the inference of a full parameter covariance matrix (and hence the joint distribution) an untenable task. As an example, the FNN model described earlier in Section 4.3 with four inputs, one output and two hidden layers with 20 units each, has a total of 541 trainable parameters in its deterministic version. Applying standard Bayesian inference to establish the joint distribution of these parameters would require determining a total of 146,882 additional quantities. However, obtaining an approximate probabilistic representation $p(\boldsymbol{\theta}|\mathbf{y})$ for a Neural Network model is still possible under one key assumption: since the dependence between model parameters is expected to be weak, the covariance matrix $\boldsymbol{\Sigma}_{\theta\theta}$ can be considered as a diagonal matrix (Blundell, et al., 2015). The weak dependence assumption is supported by the nature of the Neural Network parameters that have no concrete physical meaning – and hence no conditions to imply dependence. With this simplifying assumption, the total number of trainable parameters is approximately double that of a deterministic Neural Network model.

Several practical implementations of BNNs are included in publicly available software libraries. For the present study we use an implementation from the tensorflow-probability package in Python (Sountsov, et al., 2019). The probabilistic parameters are introduced to the model by replacing one or more of the standard fully connected hidden layers (also referred to as "dense" layers), **Figure 44**, with so-called "variational" layers. In the variational layers, the weights and biases are represented by their means and standard deviations following the assumptions discussed above. As discussed in Section 4.1, having variational layers in the model (i.e., having a model with probabilistic parameters) allows capturing the epistemic uncertainty. In addition, the BNN architecture can also be tailored to capture aleatory uncertainty, by defining the output layer as probabilistic – where the output quantities are considered realizations from a (joint) probability distribution with parameters conditional on the model coefficients in previous layers. This is an example of what is referred to as "Models with probabilistic outputs" in Section 4.1.

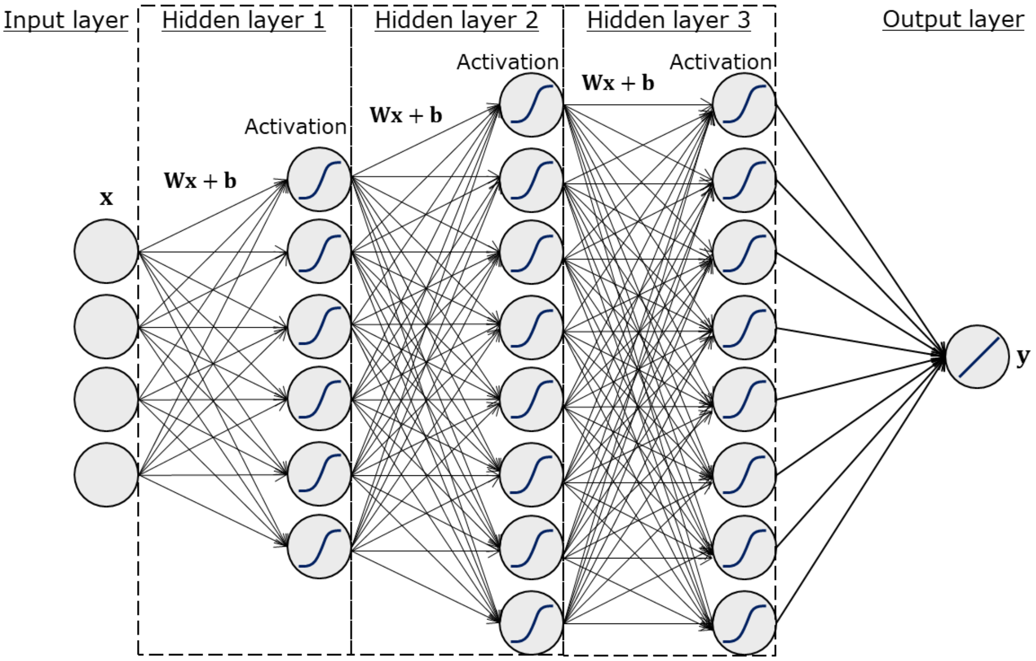

Figure 44: Illustration of a Feedforward Neural Network (FNN) with three hidden layers.

Using the tensorflow-probability BNN implementation, we train three model examples where different parts of the model are considered probabilistic and as a result the uncertainty is treated in a different way. The model architecture in terms of number of layers and units is retained from the deterministic FNN model considered earlier – with 2 hidden layers and 20 units in each layer. Model training is done using the Adam solver, and the cost function is a negative log-likelihood function proportional to $p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. For all models, the parameter probability distributions as well as the output probability distributions are considered jointly Gaussian. An overview of the three BNN models is provided in **Table 18**, compared to the setup of a deterministic model with the same general architecture.

Table 18: Overview of FNN and BNN models

Model #	Inputs	Hidden layer 1	Hidden layer 2	Output layer	Uncertainty captured
1	$u, \sigma_u, \alpha, \rho$	Dense (linear + activation)	Dense	Linear	None (epistemic in case an ensemble model is trained)
2		Dense	Dense	Gaussian	Aleatory $(\boldsymbol{\varepsilon}_y)$
3		Variational	Variational	Linear	Epistemic $(\boldsymbol{\varepsilon}_g)$
4		Variational	Variational	Gaussian	Aleatory and epistemic $(\boldsymbol{\varepsilon}_y + \boldsymbol{\varepsilon}_g)$

BNN model training is carried out on the full aeroelastic load data sets, without averaging over different realizations. In this way the aleatory uncertainty is retained in the output data and can be captured by the BNN model. The data are split into a training set consisting of 20 realizations at each of 1,600 sample points, a total of 32,000 data points. The test set covers 400 sample points with 20 realizations each, or a total of 8,000 data points.

Figure 45 illustrates the behavior of the FNN and BNN models depending on the types of uncertainty taken into account, for one selected load quantity, the blade root flapwise bending moment extremes. The blue dots show data points representing the outcomes from aeroelastic simulations, as function of mean wind speed. The scatter is due to realization-to-realization uncertainty as well as due to the effect of other variables (turbulence, wind shear and density) that are not considered in the plot. The red dots and lines show model predictions for wind speed inputs varying from cut-in to cut-out speed, with the other three input variables held constant (meaning we can expect somewhat lower variance in the model predictions compared to the full data set). The top left plot in **Figure 45** shows the outputs of an ensemble of deterministic models trained as described in Section 4.3 (model #1 in **Table 18**). The top right plot shows the outputs of model #2 where just the output layer is probabilistic, treating all uncertainty as aleatory. The bottom left plot shows model #3 where only one hidden layer is probabilistic, thus capturing part of the epistemic uncertainty. Finally, the bottom right plot shows the predictions from model #4, a BNN with both a variational hidden layer and a probabilistic output layer. There is both an estimate of epistemic uncertainty and the aleatory uncertainty, and the variation in the output predictions is the combination of the two uncertainty sources.

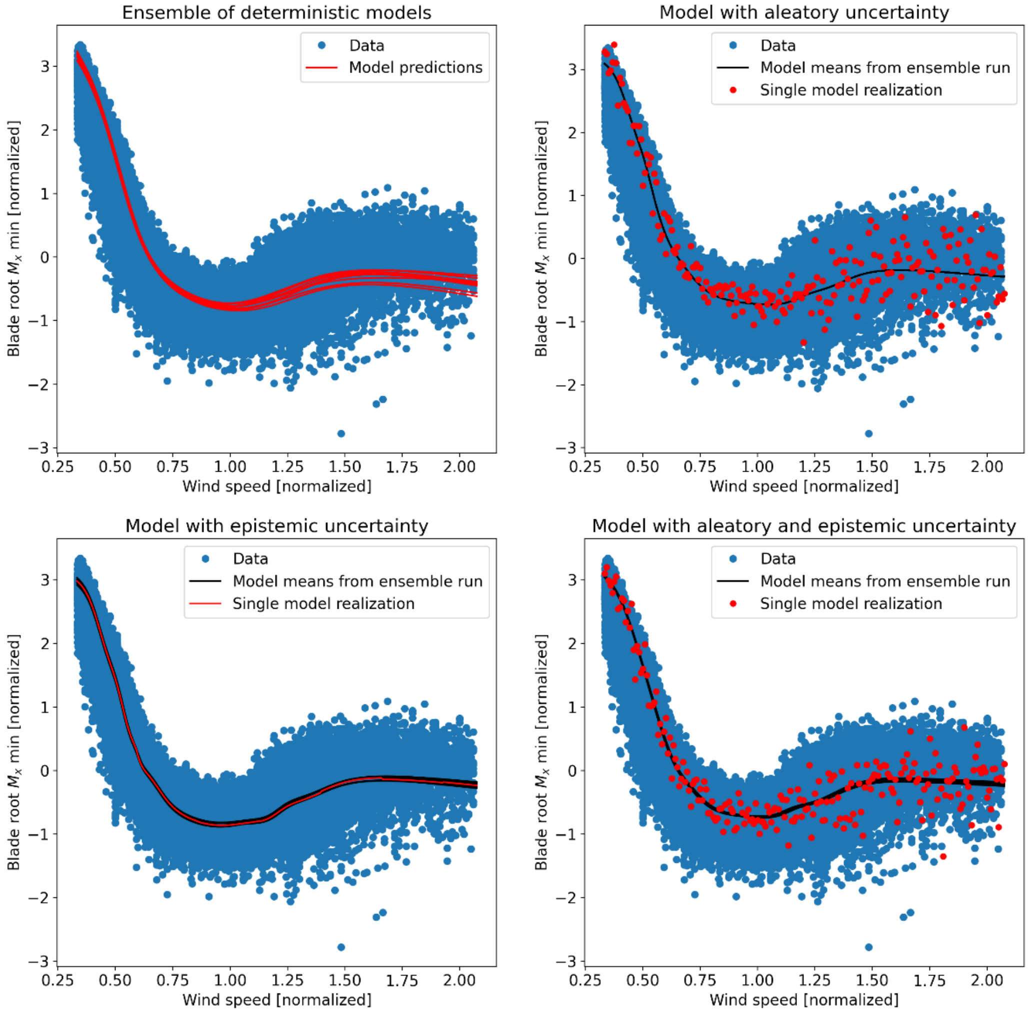

Figure 45: Comparison of probabilistic models on the prediction of blade root flapwise extreme bending moments. Top left: deterministic model not considering any uncertainty, top right: model considering all uncertainty as aleatory, bottom left: model taking only epistemic (model) uncertainty into account, bottom right: a model that considers both epistemic and aleatory uncertainties.

Figure 46 shows a scatter plot of the predictions of model #4 for the entire test data set, compared to the actual load outcomes of the same data set. Since in this example the inputs to the BNN model are exactly the same as the inputs to the load simulations, we should expect that with an adequate model the distribution of the model predictions will overlap with the distribution of the full data. We can see that this is the case for the blade root flapwise extreme bending moments. In the following section, the BNN model performance assessment is extended to other relevant load channels.

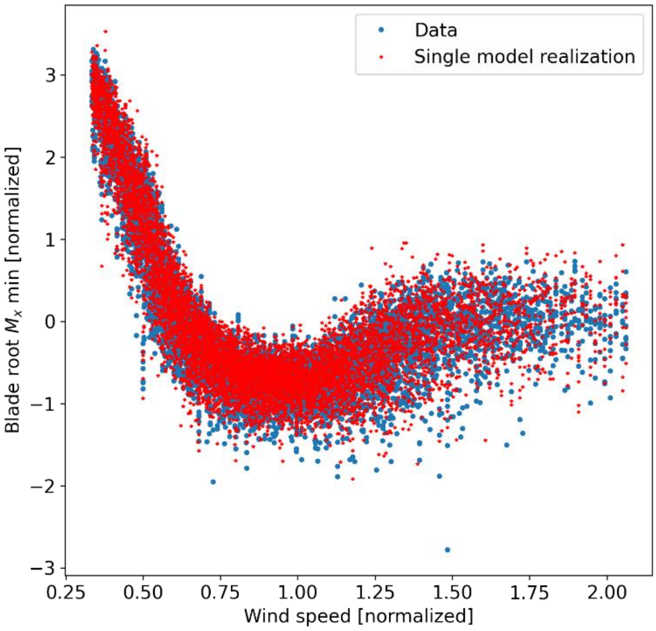

Figure 46: Scatter plot of wind turbine blade flapwise minima predictions from a BNN model compared to the true load results for the same input conditions.

4.5. BNN model performance assessment

As discussed in Section 4.2, the wind turbine design load assessment considers multiple limit states, and the performance of a load surrogate model needs to be assessed in different ways depending on the quantity of interest. Further, some standard performance measures like coefficient of determination (R-squared) and Root Mean Squared Error (RMSE) are not directly applicable for probabilistic models. As a result, the model performance assessment is based on comparison of the output probability distributions using the following criteria:

- Exceedance probabilities (distribution tails) for channels representing 10-minute extremes; and
- Probability densities (pdfs) as well as Kullback-Leibler divergence (KL divergence, describing the difference between probability distributions (Kullback & Leibler, 1951)) for channels representing 10-mean values as well as those representing Damage-Equivalent Loads (DELs).

We compare the BNN model with variational layers and distribution output (model #4 in **Table 18**) against an ensemble model where both the epistemic and aleatory uncertainty have been taken into account using the approach described as “model with probabilistic outputs” in Section 4.1. With this approach, the output is considered to follow a Gaussian probability distribution due to the aleatory uncertainty, $\boldsymbol{\varepsilon}_y$, and a separate model is trained to predict each of its distribution parameters conditional on the inputs. In order to account for the epistemic uncertainty, $\boldsymbol{\varepsilon}_g$, the model that predicts the conditional mean is actually an ensemble of multiple deterministic models. Sampling from the model constructed with this approach (simply referred to as “ensemble model” below), happens as following: 1) at a given input combination, $\mathbf{x}_j$, run the N_k ensemble models to obtain the mean predictions $\bar{\mathbf{y}}_k|\mathbf{x}_j, k = 1 \dots N_k$. Compute $\bar{\mathbf{y}}|\mathbf{x}_j$ and $\boldsymbol{\sigma}_{\boldsymbol{\varepsilon}_g}|\mathbf{x}_j$ as the mean and standard deviation of the predictions, respectively. 2) call the separate model that estimates $\boldsymbol{\sigma}_{\boldsymbol{\varepsilon}_y}|\mathbf{x}_j$. 3) Draw two random numbers (or vectors in the case of multiple outputs), $\mathbf{F}_{\boldsymbol{\varepsilon}_g}$ and $\mathbf{F}_{\boldsymbol{\varepsilon}_y}$, from a uniform distribution in the range (0,1). 4) Compute the final estimate:

$$\mathbf{y}_j = \bar{\mathbf{y}}|\mathbf{x}_j + (\boldsymbol{\sigma}_{\boldsymbol{\varepsilon}_g}|\mathbf{x}_j)\,\Phi^{-1}\left(\mathbf{F}_{\boldsymbol{\varepsilon}_g}\right) + (\boldsymbol{\sigma}_{\boldsymbol{\varepsilon}_y}|\mathbf{x}_j)\Phi^{-1}\left(\mathbf{F}_{\boldsymbol{\varepsilon}_y}\right)$$

An ensemble model based on the above procedure as well as a BNN model are trained on each of the 14 signals listed in Section 4.2. The training uses the same 32,000 data points as defined in Section 4.4. Since the comparisons involve tail probabilities, it is important to include as many rare events as possible, therefore the evaluation of the model performance are based on all the 40,000 data points available. **Figure 47** shows the fatigue damage-equivalent load (DEL) probability distribution predictions for six load channels, while **Figure 48** shows a comparison of the exceedance probabilities in the tail distributions of the extremes for four load channels. The agreement in the density functions is good except for the shaft torsion, where both surrogate model approaches have some difficulties. A potential explanation is that at around rated wind speed the shaft torsion exhibits some scatter (see **Figure 41**) due to changing controller regimes, which is poorly correlated with the input conditions and hence difficult to explain by a surrogate model. In all the plotted cases (DEL and extremes) the BNN shows similar or even superior performance than the ensemble model approach.

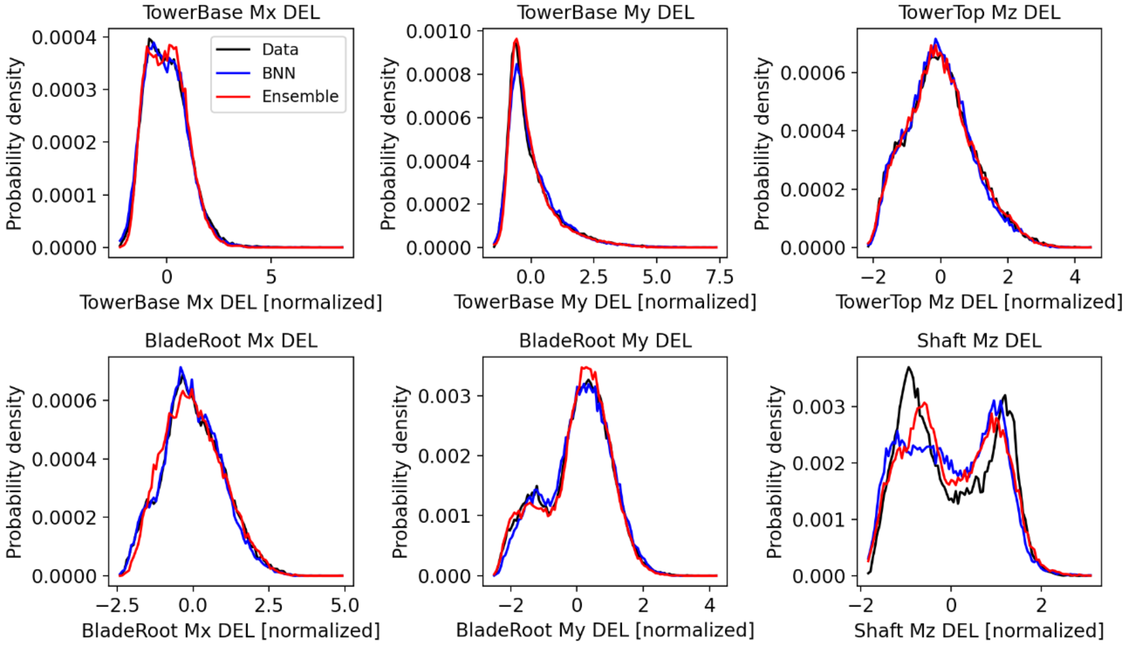

Figure 47: Comparison of the probability densities of fatigue damage-equivalent load predictions from a BNN model and an ensemble of deterministic models, for six wind turbine load signals.

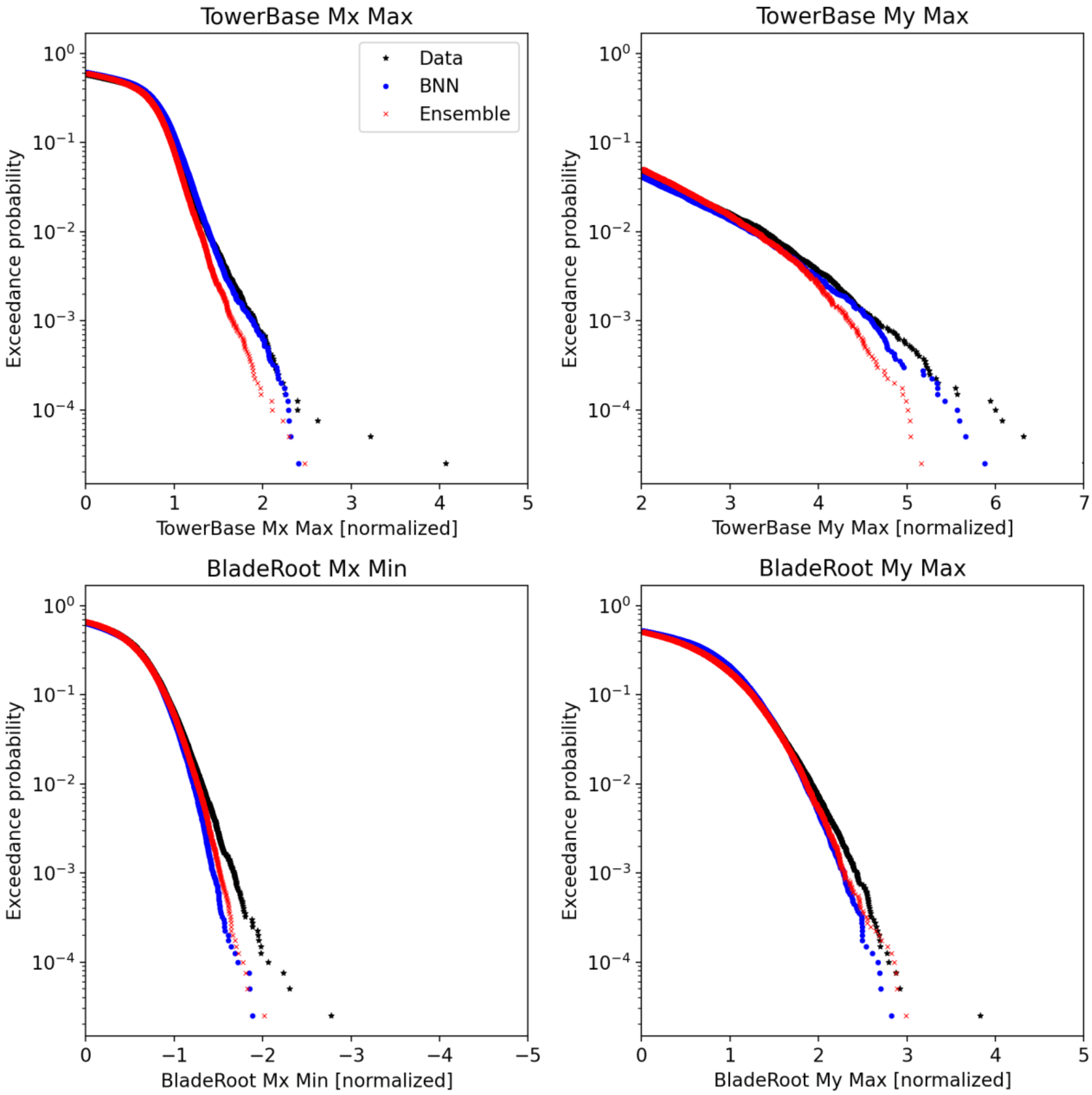

Figure 48: Comparison of the tail probability distribution predictions by a BNN model and an ensemble of deterministic models, for four load channels.

The shapes of the probability distributions shown in **Figure 47** and **Figure 48** are governed by two factors: one is the conditional dependence of the loads on the input conditions, and the second is the distributions of the uncertainties (epistemic and aleatory). For the extreme load cases as those shown on **Figure 48**, it is likely that the most critical response can happen within a limited range of the input variables. In such a situation, the tail of the extremes distribution may just contain a few events within a narrow range of conditions, and the shape of the tail will then be dominated by the distribution of the uncertainties $\boldsymbol{\varepsilon}_g$ and $\boldsymbol{\varepsilon}_y$. In the present study all models assume Gaussian distributions of the uncertainties, meaning that the shape of the tails predicted by the surrogate models will resemble that of a Gaussian distributed variable. The plots shown in **Figure 48** indicate that for some load channels the distribution of the largest extremes can deviate from Gaussian.

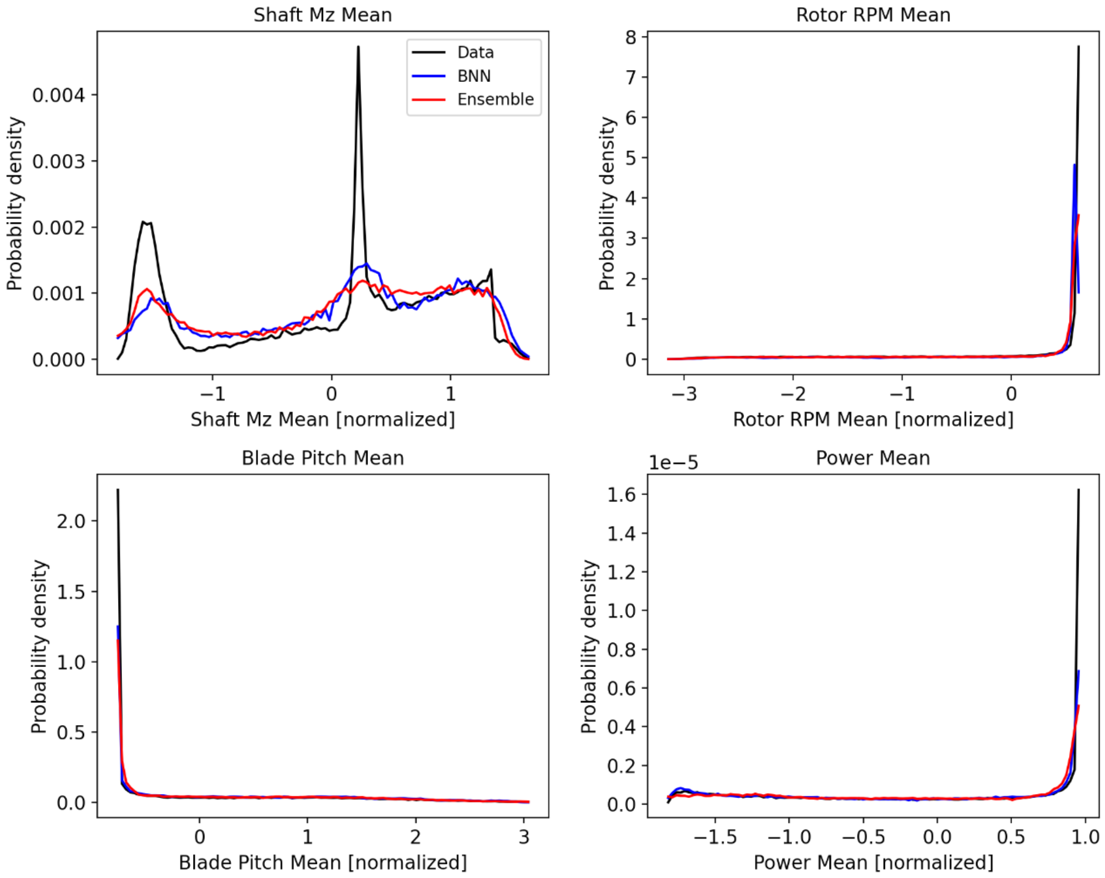

Figure 49: Comparison of the probability densities of the predictions from a BNN model and an ensemble of deterministic models, for the mean values of four wind turbine load signals.

Figure 49 compares the probability density predictions for the mean values of four signals. Just as with the damage-equivalent loads, the prediction of the mean shaft loads probability density is less accurate than for the other channels, and this is likely due to the same reasons. Regarding the power output, rotor rpm, and blade pitch signals, the model predictions seem to agree well with data, however interpretation of these signals is difficult because they all have a certain fixed (nominal) value for a broad range of input conditions, which creates a large delta function-like peak in the probability distributions.

As means of quantitative comparison between distributions, we consider the Kullback-Leibler (KL) divergence that is a measure of the difference between the true distribution of the data, $P(\mathbf{y})$, and the distribution of the model predictions, $Q(\mathbf{y})$. In a discretized form, the KL divergence is computed as:

$$D_{KL}(P||Q) = \sum_{\mathbf{y}} P(\mathbf{y}) \log\left(\frac{P(\mathbf{y})}{Q(\mathbf{y})}\right)$$

Due to the specifics of the mean results for power, blade pitch and rotor rpm discussed above, these signals are not included in the assessment of their KL divergence. For the remaining signals under consideration, the KL divergence of the model predictions with respect to the data are listed in **Table 19**.

Table 19: KL divergence of model prediction probability distributions with respect to load simulation data distributions. Green shading indicates that one of the two models has smaller KL divergence for the particular signal, while yellow shading shows cases with less than 5% difference between the two models.

Channel name	D_{KL}**, ensemble model**	D_{KL}**, BNN model**
Tower base M_x DEL	$8.18 \cdot 10^{-5}$	$4.08 \cdot 10^{-5}$
Tower base M_yDEL	$1.27 \cdot 10^{-4}$	$1.94 \cdot 10^{-4}$
Tower top M_z DEL	$8.86 \cdot 10^{-5}$	$1.44 \cdot 10^{-4}$
Blade root M_x DEL	$2.21 \cdot 10^{-4}$	$1.66 \cdot 10^{-4}$
Blade root M_y DEL	$4.97 \cdot 10^{-4}$	$6.17 \cdot 10^{-4}$
Shaft M_z DEL	$5.35 \cdot 10^{-3}$	$3.92 \cdot 10^{-3}$
Tower base M_x MAX	$4.27 \cdot 10^{-5}$	$4.38 \cdot 10^{-5}$
Tower base M_y MAX	$3.00 \cdot 10^{-4}$	$4.70 \cdot 10^{-5}$
Blade root M_x MIN	$1.49 \cdot 10^{-4}$	$1.57 \cdot 10^{-4}$
Blade root M_y MAX	$9.31 \cdot 10^{-4}$	$4.40 \cdot 10^{-4}$
Shaft M_z Mean	$9.22 \cdot 10^{-3}$	$9.25 \cdot 10^{-3}$

For many of the investigated channels, the KL divergence of the BNN models is smaller than that of the ensemble model. The differences in the distributions are not significant enough to claim that the BNN model is decisively superior, however the results clearly show that BNNs could be at least as accurate as their alternatives and can be a promising approach for running probabilistic wind turbine load surrogates.

4.6. Multivariate models using BNNs

Throughout the description of the present study we have maintained the vector notation for the model outputs **y**. This is to remind that while the majority of the discussed models have scalar outputs, the entire setup is easily extendable to models with multiple outputs – as long as there is a parametric probability distribution that can describe the joint probability of these multiple outputs. Given that the present modelling setup only uses Gaussian distributions, predicting multiple outputs is straightforward by assigning a multivariate Gaussian distribution to the outputs. During the training process the means and the variances of each of the output signals will be inferred, as well as their covariance. A demonstration of the outcome of such a model choice is shown in **Figure 50**.

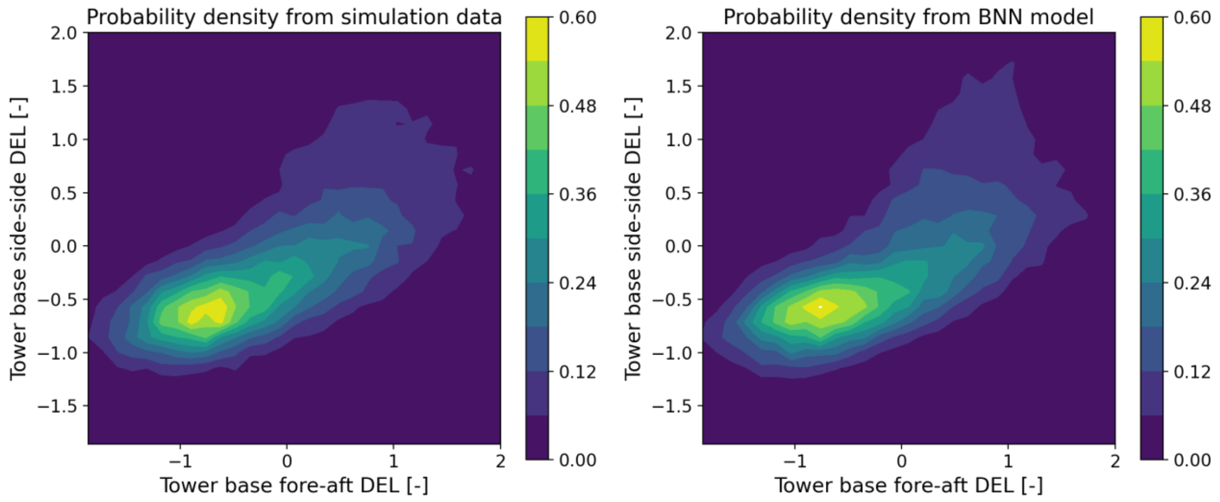

Figure 50: Comparison of bivariate probability density prediction from a BNN model to the probability density of a reference data set. Left: probability density contours from load simulations; Right: probability density contours from BNN model.

4.7. Conclusions on the use of BNN as load surrogates

This study demonstrated how Bayesian Neural Networks (BNNs) can be used as probabilistic surrogate models for wind turbine load and power prediction. Their performance was evaluated with the aeroelastic model of a commercial multi-megawatt turbine, and compared against the performance of an alternative probabilistic surrogate based on an ensemble of deterministic Feedforward Neural Networks (FNNs). The main conclusions are the following:

- BNNs are a viable option for building a probabilistic surrogate model that can both take epistemic and aleatory uncertainties into account;
- Compared to an ensemble FNN model, the BNN were shown to have equal or superior performance in modelling the probability distribution of load and power outputs – while being computationally more efficient. This shows the potential of BNNs as surrogates for fatigue load and power prediction;
- The way the tail behavior is modelled is driven by the choice of the posterior distribution functions – using a joint Gaussian prescribes a Gaussian-like tail. As a result, the tail probability predictions are very similar to those from an ensemble model where the aleatory uncertainty is modelled as a Gaussian distribution too.
- The realization-to-realization uncertainty in the load simulations seems to be non-Gaussian, hence a more complex model with a non-Gaussian posterior may be required (but not guaranteed) to improve the tail prediction capabilities.

For the type of problems presently discussed, BNNs may be an efficient alternative to Gaussian models as they show high computational performance and uncertainty prediction capabilities, while lacking the limitations of the Gaussian models in terms of deteriorating computational efficiency for large sample sizes.

5. Conclusions

Within this report, we documented two complementary approaches to perform surrogate modelling of complex aero-servo-elastic simulations: transient response surrogates based on the newly developed manifold-autoregressive with exogenous inputs modelling (mNARX) methodology, and stochastic simulation of 10-minute aggregate quantities (DEL and Max Load) with Bayesian neural networks (BNN).

Both metamodeling approaches exhibit robust and accurate performance according to various accuracy metrics. They significantly outperform the original requirement of one order of magnitude speedup with respect to full numerical simulations, with training costs in the order of $N_{ED} \approx \mathrm{O}(10^{2-4})$.

We therefore conclude that both surrogate modelling methods can be used to significantly speedup ASE simulations, and therefore enable fully quantitative and uncertainty-aware wind turbine design practices.

References

Billings, S. A., 2013. *Nonlinear Systems Identification.* s.l.:John Wiley & Sons.

Blundell, C., Cornebise, J., Kavukcuoglu, K. & Wierstra, D., 2015. Weight Uncertainty in Neural Networks. *Proceedings of Machine Learning Research,* Volume 37, pp. 1613-1622.

Conti, D., Dimitrov, N. K., Peña, A. & Herges, T., 2021. Probabilistic estimation of the Dynamic Wake Meandering model parameters using SpinnerLidar-derived wake characteristics. *Wind Energy Science,* 6(5), pp. 1117-1142.

Dimitrov, N. K., 2019. Surrogate models for parameterized representation of wake-induced loads in wind farms. *Wind Energy,* 22(10), pp. 1371-1382.

Dimitrov, N. K., 2022. *Hipersim Homepage.* [Online]
Available at: https://pypi.org/project/hipersim/
[Accessed 2022].

Dimitrov, N. K., Der Kiureghian, A. & Berggreen, C., 2016. Bayesian inference model for fatigue life of laminated composites. *Journal of Composite Materials,* 50(2), pp. 131-143.

Dimitrov, N. K., Natarajan, A. & Kelly, M. C., 2015. Model of wind shear conditional on turbulence and its impact on wind turbine loads. *Wind Energy,* 18(11), pp. 1915-1931.

Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep learn.* s.l.:MIT Press.

IEC, 2019. *International Standard, Wind energy generation systems - Part 1: Design requirements,* s.l.: International Electrotechnical Commission.

Jonkman, B., 2009. *TurbSim User's Guide: Version 1.50.*

Jonkman, J., Butterfield, S., Musial, W. & Scott, G., 2009. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*, National Renewable Energy Laboratory.

Kelly, M. C., Larsen, G. C., Dimitrov, N. K. & Natarajan, A., 2014. Probabilistic Meteorological Characterization for Turbine Loads. *Journal of Physics: Conference Series*, 524(1), p. 012076.

Kingma, D. P. & Ba, J., 2017. Adam: A Method for Stochastic Optimization. *Arxiv Preprint*, p. 1412.6980.

Kullback, S. & Leibler, R., 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1), pp. 79-86.

NREL, 2021. *OpenFAST Documentation, Release v2.5.0*, s.l.: National Renewable Energy Laboratory.

NREL, 2021. *ROSCO. Version 2.3.0.* s.l.:GitHub.

Platt, A., Jonkman, B. & Jonkman, J., 2016. *InflowWind User's Guide*, s.l.: National Wind Technology Center.

Schröder, L., Dimitrov, N. K., Verelst, D. R. & Sørensen, J. A., 2018. Wind turbine site-specific load estimation using artificial neural networks calibrated by means of high-fidelity load simulations. *Journal of Physics: Conference Series*, 1037(6), p. 062027.

Sountsov, P., Suter, C., Burnim, J. & Dillon, J. V., 2019. *Regression with Probabilistic Layers in TensorFlow Probability*. [Online]
Available at: https://blog.tensorflow.org/2019/03/regression-with-probabilistic-layers-in.html
[Accessed 2022].

Tiao, G. C. & Box, G. E. P., 1992. *Bayesian inference in statistical analysis.* Chicago: Wiley.

Zhu, X. & Sudret, B., 2020. Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions. *International Journal for Uncertainty Quantification*, 10(3), pp. 249-275.

Zhu, X. & Sudret, B., 2021. Emulation of stochastic simulators using generalized Lambda models. *SIAM-ASA Journal on Uncertainty Quantification*, 9(4), pp. 1345-1380.