Inter-turbine Modelling of Wind-Farm Power using Multi-task Learning

Simon M. Brealy[a,*], Lawrence A. Bull[b], Pauline Beltrando[c], Anders Sommer[c], Nikolaos Dervilis[a], Keith Worden[a]

[a]*Dynamics Research Group, Department of Mechanical Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK*
[b]*School of Mathematics and Statistics, University of Glasgow, Glasgow, G122 STA, Scotland*
[c]*Vattenfall R&D, Vattenfall AB, Älvkarleby, Sweden*

Abstract

Because of the global need to increase power production from renewable energy resources, developments in the online monitoring of the associated infrastructure is of interest to reduce operation and maintenance costs. However, challenges exist for data-driven approaches to this problem, such as incomplete or limited histories of labelled damage-state data, operational and environmental variability, or the desire for the quantification of uncertainty to support risk management.

This work first introduces a probabilistic regression model for predicting wind-turbine power, which adjusts for wake-effects learned from data. Spatial correlations in the learned model parameters for different tasks (turbines) are then leveraged in a hierarchical Bayesian model (an approach to multi-task learning) to develop a "metamodel", which can be used to make power-predictions which adjust for turbine location—including on previously *unobserved* turbines not included in the training data. The results show that the metamodel is able to outperform a series of benchmark models, while demonstrating a novel strategy for making efficient use of data for inference in populations of structures, in particular where correlations exist in the variable(s) of interest (such as those from wind-turbine wake-effects).

Keywords: Multi-task learning, Hierarchical Bayes, Population-based SHM

*Corresponding author.
Email address: sbrealy1@gmail.com (Simon M. Brealy)

(PBSHM), Wind-power prediction

1. Introduction

1.1. Motivation

Commitments to the expansion of the renewable energy sector equate to tripling the installed capacity globally by 2030 [1]. This growth presents some significant challenges, including how to efficiently manage the increasing cost of operations and maintenance (O&M)—which represent approximately 40% of the lifecycle cost of an offshore wind-farm [2]. Currently, O&M is supported by online monitoring systems [3], utilising sensors from across the turbines for decision-making; gaining the maximum possible insight from these data is therefore of interest to help maximise the economic viability of offshore wind-farms.

1.2. Data-driven Modelling

A common application of data for the online monitoring of wind-farms is in modelling the relationship between wind-speed and power (known as power-curve modelling). These models may be used as a damage-sensitive feature during operation, to inform the need for any remedial repairs [4, 5]. As an additional benefit, these models can also be repurposed to make predictions of future wind-power generation (given forecasted weather inputs), which can help with risk mitigation in both operational and electricity market settings; this includes planning unit dispatch, maintenance scheduling and maximising profit in power trading [6, 7]. A relatively recent review of the modelling approaches developed for power-curve monitoring is provided by Wang *et al.* [8], which is a heavily researched area. However, in many cases the approaches used provide point, deterministic, predictions to predict output power. Inspection of typical power-curve data reveals that it is both heteroscedastic (the variance of the output power changes with wind-speed), and asymmetrically distributed. Deterministic approaches fail to capture these properties; as such, probabilistic approaches have gained greater attention more recently from researchers [4, 5, 9], which provide information on both the mean and the expected variance of the prediction. This additional information gives greater insight into model uncertainty, which could be helpful for managing risk in both engineering and commercial settings.

Further data-driven applications for the online monitoring of offshore wind-farms exist, including focussing on component health such as generator and gearbox-temperature monitoring, blade and bearing fault detection and, pitch and yaw misalignment [10, 11, 12], or more system-level health e.g. detection of scour around the base of turbine monopiles [13]. However, in these applications (and also more generally in the monitoring of engineering systems), significant challenges remain, including (1) a lack of labelled damage-state instances [14], (2) short or incomplete histories of failure data [15], and (3) operational and environmental variability. These challenges can impede the level of sophistication, or even practical implementation of these systems [16].

1.3. Multi-task Learning

As a solution to the problem of limited data, multi-task learning (MTL)—a form of transfer learning (TL)—is a machine-learning approach where the goal is to learn multiple related tasks simultaneously, so that similarities (and differences) can be harnessed between tasks; in this way, model predictive accuracy is improved in domains where data are limited [17, 18, 19]. In practice, a common approach to MTL is by the use of hierarchical Bayesian models, which are able to pool information across tasks via population-level parameters, while accounting for task-specific nuances via task-level parameters. By pooling information in this way, these models can better handle tasks with limited data, and produce robust probabilistic predictions. This model architecture can intuitively be applied to wind-farms, where individual turbines (tasks) may have individual differences (e.g. because of maintenance, damage or physical location), but are ultimately similar to each other as part of the wind-farm population. Hierarchical Bayesian models have recently been applied to other engineering infrastructure; Bull et al. [20] uses a hierarchical Bayesian modelling approach to learn population-level and task-specific parameters in a combined inference, in the setting of a truck-fleet survival analysis. Similarly, Dardeno et al. [21] and Brealy et al. [22] use hierarchical Bayesian models, in order to transfer Frequency Response Function information between helicopter blades and wind-turbine foundation stiffness parameters respectively. In both cases, the uncertainty of parameters in data-sparse domains was reduced as a result of knowledge transfer between structures, as compared to modelling the data-sparse domains independently.

1.4. Research Contribution

In this work, a probabilistic power-curve model is first developed, that considers the heteroscedastic and asymmetric nature of power-curve data, using data from an in-service wind-farm. This model utilises wind-farm (population-level) wind-speed and direction features, with predictions indicating that a level of wind-directionality (from wind-turbine wake-effects), has been captured by the model. Spatial correlations found in the underlying model parameters inspired the incorporation of this model into a hierarchical Bayesian model architecture, where the population-level parameters were designed to capture spatial, inter-task correlations between turbines, which is described here as the "metamodel".

> This model design has a tangible real-world benefit—if one were able to make accurate probabilistic predictions for a variable of interest, at a location without data (and perhaps even without associated telemetry), this may reduce the overall telemetry requirements among a population.

The results show that for predicting wind-turbine output power, the metamodel outperforms a series of benchmark models, both in terms of mean prediction and predictive uncertainty, by making efficient use of data and accounting for environmental variability. The metamodel could also be described as a purely data-based approach to power modelling with adjustment for wake-effects, without the need for physics-based simulation (thus making it more efficient).

1.5. Paper Layout

The layout of the paper is as follows. Section 2 describes the data used and the preprocessing steps. Sections 3 and 4 describe the models. Section 5 describes the model training and scoring process. Results are analysed in Section 6. Conclusions and further work are described in Section 7. Finally, links to the model code used in this study can be found in Section 8.

2. Dataset Overview and Preprocessing

2.1. Description

A Supervisory Control and Data Acquisition (SCADA) dataset was available to the authors of this work, which contains 224 wind-turbines, spanning

three separate wind-farms between the years 2020-2023, at a ten-minute resolution. The measurements within come from a broad range of sensors, but generally fall into electrical, control, component/system temperature measurements, and ambient environmental parameters. This study focuses on one of these wind-farms (under the pseudonym "*Ciabatta*"), for the full year of 2021; this was chosen to maintain a full year of cyclical environmental variation, and to reduce computational overhead with a focus on demonstrating the methods herein.

The raw SCADA data, showing the power-curve relationship for the wind-turbines in wind-farm *Ciabatta* are shown in blue in Figure 1. In these data, a number of patterns typical to wind-farm operation are identifiable, including *curtailments* (whereby turbine power is deliberately limited), *shutdowns* (indicated by a power output of zero, despite significant wind-speeds), and *power-boosting* (where power is raised above nominal turbine capacity). These three (controlled or otherwise) deviations of power away from *normal* operation are typically related to technical and/or commercial constraints, dependent on factors which were considered outside the scope of this work.

2.2. Data Filtering

For the purposes of predicting power output during *normal* operation, data most likely related to *shutdowns* and *curtailments* were filtered out. For the *power-boosted* data, rather than filtering it, it was instead artificially reduced to the rated-power; this was done for two main reasons: (1) For some turbines, it accounted for a significant portion of the data at high wind-speeds, such that filtering it would create data-sparse regions for those turbines, making robust model development and comparison between different models more difficult. (2) From a practical perspective, if a model predicts that rated-power is achievable, and the specific conditions required for *power-boosting* are known, then domain-experts could apply a manual correction if necessary. Additionally, since wake-effects are more prevalent at lower wind-speeds [23], this modification was not expected to significantly interfere with wake-related behaviour in the data. A summary of the main criteria chosen to filter the data are shown in Table 1.

Between low and rated-power, in *normal* operation the blade pitch angle is approximately zero, while during curtailment it increases to control the output power to a set level. When a turbine is *offline* or during a *shutdown*, the pitch angle is also higher than in *normal* operation. Additionally, as the wind-speed increases beyond the cut-out wind-speed, the shaft speed

Filtering step	**Description**	**Filtering target**
1: Thresholding	Filter data based on blade pitch angle and rotor speed (RPM)	Curtailments and shutdowns
2: Rate of change	Filter data where the RPM changes rapidly between time steps	Transient behaviour during startup and shutdown procedures
3: Stationary data	Filter data where the power or wind-speed measurements are stationary	Curtailments and sensor faults
4: Mahalanobis squared-distance outlier detection [24]	Filter data based on the distance in the blade pitch-angle vs. power relationship	Remaining outliers
5: Standard-deviation outlier detection	Filter that retains the middle two standard deviations in power, for the binned *freestream* wind-speed	Remaining outliers

Table 1: Summary of the steps used to filter the raw SCADA data

(RPM) remains high, while the generated power drops considerably; these three sets of behaviours were targeted for filtering in step one, by defining set-thresholds on the blade-pitch-angle and rotor-speed, for differing levels of power. The second filtering step was introduced as it was found that in some cases, the RPM changed quickly; this was considered indicative of *startup* and *shutdown* procedures of the turbines, and not representative of *normal* operation. In this step, data where the RPM increased or decreased by more than a set threshold, compared to the previous value were filtered out. In the data, instances of stationary wind-speed measurements were found, which were deemed likely to be due to sensor or SCADA system error. To combat this, step three was used to remove data where the wind-speed measurements were stationary in time. The same technique was also applied to remove any stationary power data that was not filtered by step one. Step four was used to help remove any remaining outliers in the pitch-angle vs. power relationship, by using a Mahalanobis Squared Distance (MSD)-

based approach [24], which takes account of both correlation and scale in the data. Finally, step five removed remaining outliers that were observed in the output power, by retaining only the middle two standard-deviations for a binned *freestream* wind-speed—defined in Section 2.3.

The filtered data are plotted in orange in Figure 1, compared to the raw unfiltered data in blue. It can be seen that the filtered data appear much more representative of *normal* operation only.

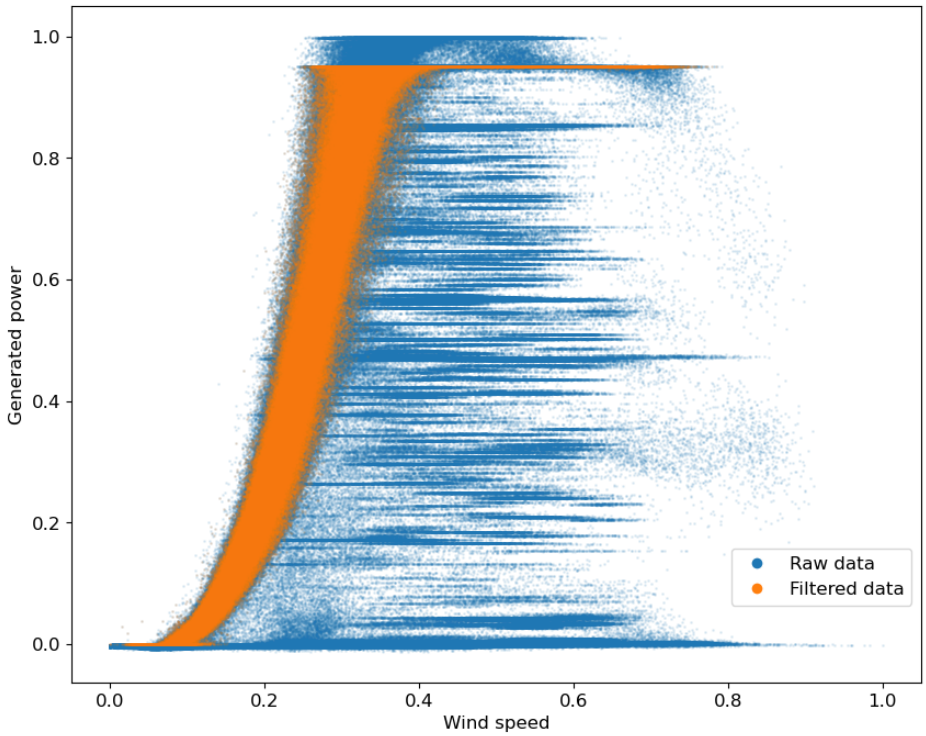
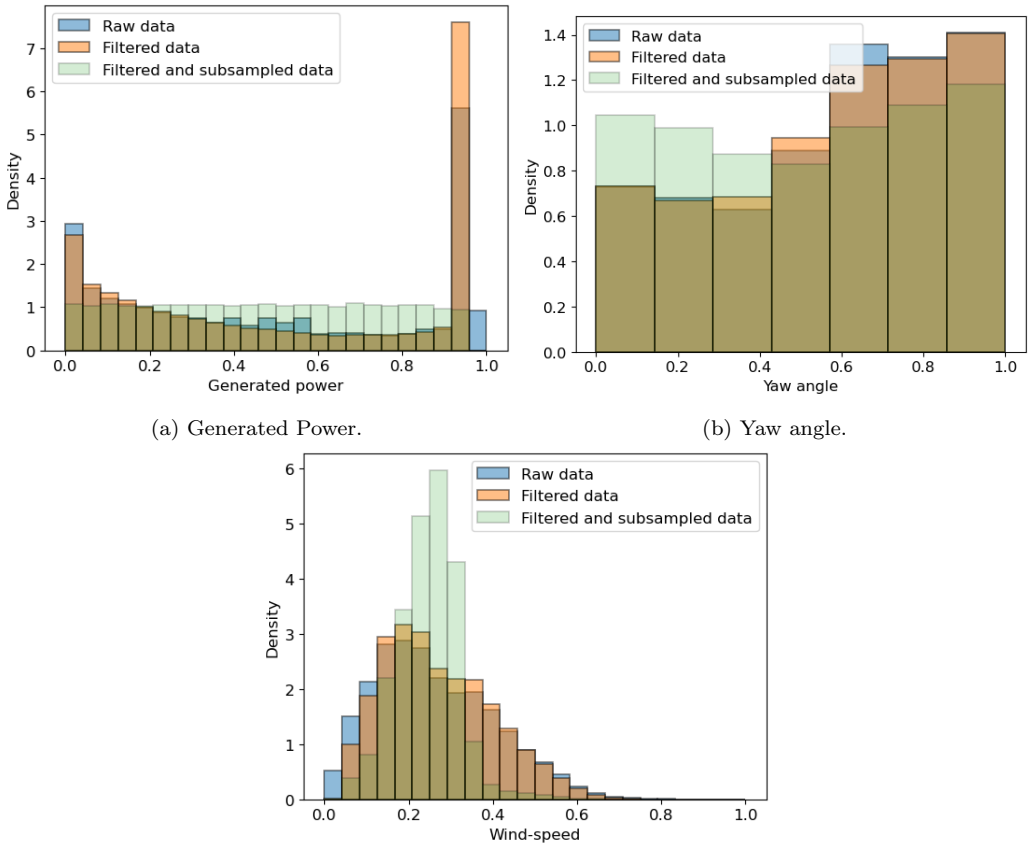


Figure 1: Data from all turbines in wind-farm *Ciabatta* showing the power-curve relationship. Blue markers show the raw data prior to filtering, and orange markers show the data after filtering.

To demonstrate how the filtering process influences the underlying distributions in the data, Figure 2 shows density plots for the generated power (2a), yaw angle (2b), and wind-speed (2c) variables. The raw (unfiltered) data are shown in blue and the filtered data are shown in orange. Here it can be seen that in most cases the general shapes of the distributions after filtering are consistent. One clear difference can be seen at low (or zero)

generated power; this is expected, since a large amount of data was present in this region and targeted for filtering. Some reductions in data density can be seen between 0.4–0.6 in generated power; this is because of the removal of the *curtailment* data. Finally, there is an increase in the quantity of data in the final bin of the filtered generated power; this is because of the capping of the *power-boosted* data. For completeness, the filtered data that had then undergone stratified (sub)sampling on both yaw angle and power features (discussed in Section 5.1.2) are also shown here in green; in this case it can be seen that the desired effect of flattening the density plots for the yaw angle and power features had broadly been achieved.



(a) Generated Power.

(b) Yaw angle.

(c) Wind-speed.

Figure 2: Density plots for the raw data (blue), filtered data (orange) and the subsampled data (green) for the generated power, yaw angle and wind-speed variables.

Considering these filtered data (and power-curve data generally), there are two distinct features that present challenges for modelling: (1) the data are heteroscedastic—as the wind-speed changes, the variance in generated power changes significantly and (2), the skewness changes from being positive at low wind-speeds before cut-in, to negative above the rated wind-speed. These properties should be considered in the model design phase to ensure that the data are modelled appropriately.

2.3. Model Training Features

For all models developed in this study, three features were used for predicting turbine-specific output power. First, came the freestream wind-speed (later referred to as Feature One); since data from a meteorological mast was not available, this was instead approximated as being equivalent to the maximum measured wind-speed of the turbines used in training, for a given timestamp. Since there are turbines on all edges of the wind-farm in the training data sets, this was deemed a reasonable approximation as measurements from "unwaked" turbines were always available. The authors are aware that anemometer sensor error will likely add noise to the data and could reduce model predictive performance as a result, however it is believed it does not detract from the novelty of the method developed in this work.

The second and third features were derived from what was considered a *proxy* for wind-direction—the measured turbine yaw angle; these were assumed to be approximately equal to each other, since for operation in the *normal* regime the control system should orientate the turbines in the direction of the incoming wind. Specifically, Features Two and Three took the sine and the cosine of the yaw angle; this was done to create a smooth transition in the inputs between the equivalent angles of zero and 360 degrees. The median values were used for these features which were taken across the turbines for a given timestamp, to obtain an *averaged* wind-direction, less subject to individual turbine noise. Finally, all features were scaled between zero and one for both numerical efficiency, and data anonymity.

The motivations for choosing these features were as follows: (1) using the freestream wind-speed in combination with wind-directional features, enables the possibility to capture some of the wake-effects that may be present in the data. (2) While not the main focus of this study, population-level features are more representative of forecast data by data providers such as ECMWF [25], since higher resolution turbine-specific forecasts that also account for turbine wake-effects are not readily available.

3. B-spline Beta Regression Model

The heteroscedacity, skewness and bounded nature of power-curve data share similarities with the shape of the beta distribution, which is bounded between zero and one, and whose variance changes with the mean of the distribution. These two factors motivated the development of the B-Spline Beta Regression Model (BSBR), which is a probabilistic regression-based model utilising a beta-likelihood. Mclean et al. [4] similarly used a beta-likelihood, and the description of the BSBR model that follows is very similar to that of Capelletti et al. [26] whom also used a spline-based approach.

3.1. Generalised Linear Models

Suppose one wishes to model the distribution of a real target variable $\mathbf{y} = [y_1, y_2 ... y_N]^T$, given an input design matrix $\mathbf{X}$. Generalised Linear Models (GLMs) are a generalisation of linear regression, which are characterised by three components:

1. a distribution for modelling $\mathbf{y}$ (which must be from the exponential family of distributions);
2. a linear predictor $f = \mathbf{X}\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is a vector of regression parameters;
3. a link function $g(\cdot)$ such that $E(\mathbf{y}|\mathbf{X}) = \boldsymbol{\mu} = g^{-1}(f)$ which allows linear models to be related to a real target variable, $\mathbf{y}$, via the link function.

3.2. Beta Regression Model

Beta regression is one such type of GLM, where the beta distribution is chosen for modelling $\mathbf{y}$. Beta distributions are bounded in the interval between zero and one, while their variance can change with the mean of the distribution; this has clear similarities with power-curve data, which can be sensibly bounded between zero and rated power, with trivial rescaling. The beta distribution for a scalar y is defined as,

$$\begin{aligned} f(y|\mu,\phi) &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} y^{(a-1)}(1-y)^{(b-1)} \\ a &= \mu\phi, \quad b = (1-\mu)\phi \end{aligned} \tag{1}$$

where $\Gamma(\cdot)$ denotes the gamma function, and where $0 < \mu < 1$ and $\phi > 0$. The mean and variance of y are,

$$\mathbb{E}(y) = \mu \tag{2}$$

$$\mathbb{V}(y) = \frac{\mu(1-\mu)}{(1+\phi)} \tag{3}$$

whereby μ can be described as the mean parameter, and ϕ the precision parameter since it is inversely proportional to the variance.

In this case for modelling power-curve data, consider a linear predictor model, $f_1 = \mathbf{X}\boldsymbol{\eta}$ for a GLM, where $\boldsymbol{\eta}$ is a vector of regression parameters. The link function is chosen to be the *logit* function (equivalent to g^{-1} equalling the *expit* function [27]), such that,

$$\boldsymbol{\mu} = expit(\mathbf{X}\boldsymbol{\eta}) \tag{4}$$

where $\boldsymbol{\mu}$ is the mean parameter vector for the beta distribution. The choice of the *logit* link function ensures that $0 < \mathbb{E}(\mathbf{y}) < 1$, which in words, means the predictive mean of the model must lie between zero and one, intuitively matching the behaviour of *normalised* power-curve data. For additional flexibility in the model, the precision parameter vector $\boldsymbol{\phi}$ (in the beta distribution) was also allowed to vary with $\mathbf{X}$ according to an additional linear model $f_2 = \mathbf{X}\boldsymbol{\zeta}$, where $\boldsymbol{\zeta}$ is an additional vector of regression parameters. Since $\boldsymbol{\zeta}$ must be non-negative (to ensure that $\mathbb{V}(\mathbf{y})$ is positive), a natural *logarithm* link function is chosen (equivalent to g^{-1} equalling the *exponential* function), such that,

$$\boldsymbol{\phi} = exp(\mathbf{X}\boldsymbol{\zeta}) \tag{5}$$

This model definition allows both the mean, $\boldsymbol{\mu}$, and precision, $\boldsymbol{\phi}$, parameter vector entries to vary as a function of $\mathbf{X}$, resulting in beta distributions whose shapes vary with the input feature(s); this is similar to the "Variable dispersion beta regression model" described in [26].

3.3. B-spline Linear Models

Regression spline models are chosen for functions f_1 and f_2; these provide greater flexibility in modelling nonlinear outputs compared to polynomial regression [28], while still being linear in parameters. Regression splines are piece-wise continuous polynomial functions, joined at *knots*, where the first and second derivatives of adjacent functions are equal; this ensures a smooth transition between polynomials. In practice, polynomials of degree three (cubic splines), balancing smoothness and flexibility, are most commonly used [29] and are chosen here. Specifically, the B-spline basis-function approach

is used for its numerical stability and computational efficiency [30]. Figure 3 shows the B-spline basis-functions for polynomials of degree three, using four interior knots between zero and one. Parameters (weights) are learned for each basis-function during model fitting, with model predictions calculated as a weighted sum of parameters and basis-functions.

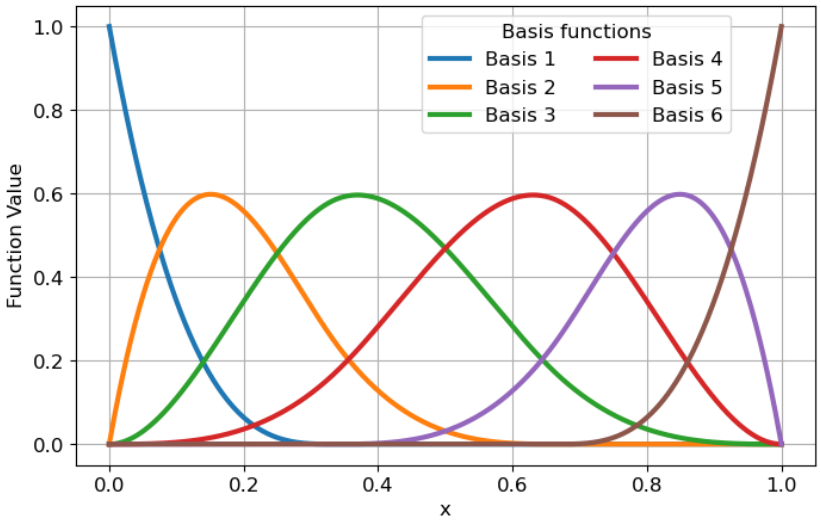


Figure 3: B-spline basis-functions over the interval of zero and one, with four interior knots.

Using the B-spline basis-function approach, the design matrix $\mathbf{X}$ was constructed and used alongside the linear model parameters $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ for the mean ($\boldsymbol{\mu}$) and precision ($\boldsymbol{\phi}$) parameters. B-splines of order four were chosen, with four, uniformly spaced, interior knots per feature. Uniform spacing was chosen for simplicity, while additional knots resulted in no significant improvement in the *normalised* mean squared error (NMSE) (defined in Section 5.3); this is demonstrated in Figure 4. Further optimisation of the number of knots for individual features, and the location of knots was not considered, and is left as an area for further model refinement.

3.4. Graphical Model

A graphical representation of the B-spline beta regression (BSBR) model is displayed in Figure 5. Latent variables (to be estimated) are depicted as unshaded circled nodes, while the shaded circled node represents the observed variable. The dotted lines indicate parameters linked by a transformation,

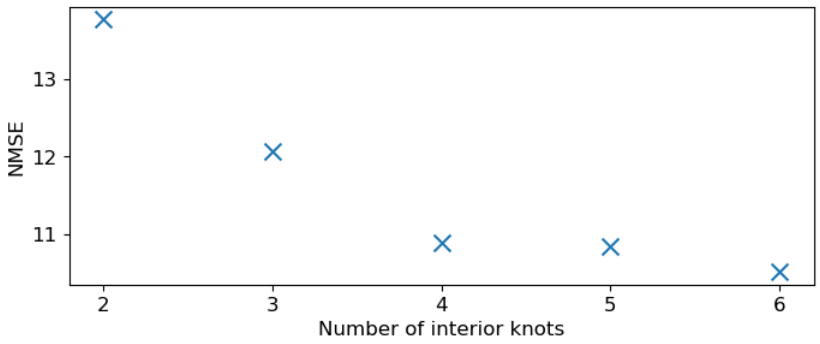


Figure 4: Comparison of NMSE score vs. the number of (evenly-spaced) interior knots per feature, for the no-pooling BSBR model described in Section 5.1.1.

while the arrows indicate that the target parameter is drawn from a distribution using the preceding parameters. The plate represents multiple instances of the contained node.

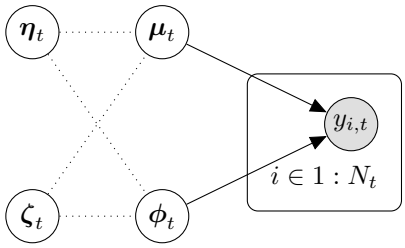


Figure 5: A graphical representation of the B-spline beta regression model.

Here, t relates to the training turbine, while N_t relates to the number of training data points for that turbine. In this initial case, independent models are learned per turbine t, used to model the observed target variable $y_{i,t}$. The column vector $\mathbf{y}_t$, with entries $y_{i,t}$ for $i = 1, 2, \ldots, N_t$, is distributed according to the reparameterised beta distribution,

$$\mathbf{y}_t \sim \text{Beta}(\boldsymbol{\mu}_t, \boldsymbol{\phi}_t) \tag{6}$$

Additionally, *hyperpriors* were placed over the estimated parameters $\boldsymbol{\eta}_t$ and $\boldsymbol{\zeta}_t$, according to a Normal distribution,

$$\{\boldsymbol{\eta_t}, \boldsymbol{\zeta_t}\} \sim \mathcal{N}(0, 3) \quad \text{(element-wise)} \tag{7}$$

where the means and standard deviations of 0 and 3 respectively were found to be weakly informative, while sufficiently aiding convergence.

3.5. Inference

The observed variables in graphical models may be referred to as evidence nodes [31]. In the example in Figure 5, the regression would have the following evidence nodes:

$$\mathcal{E} = \{\mathbf{y}_t\} \tag{8}$$

Latent variables on the other hand are *hidden* nodes,

$$\mathcal{H} = \{\boldsymbol{\eta}_t, \boldsymbol{\zeta}_t\} \tag{9}$$

Bayesian inference relies on finding the posterior distribution of $\mathcal{H}$ given $\mathcal{E}$, that is, the distribution of the unknown parameters given the data,

$$\begin{aligned} p(\mathcal{H} \mid \mathcal{E}) &= \frac{p(\mathcal{E} \mid \mathcal{H})p(\mathcal{H})}{p(\mathcal{E})} \\ &= \frac{p(\mathbf{y_t} \mid \boldsymbol{\eta}_t, \boldsymbol{\zeta}_t)p(\boldsymbol{\eta}_t)p(\boldsymbol{\zeta}_t)}{p(\mathbf{y_t})} \\ &= \frac{p(\mathbf{y_t} \mid \boldsymbol{\eta}_t, \boldsymbol{\zeta}_t)p(\boldsymbol{\eta}_t)p(\boldsymbol{\zeta}_t)}{\int\int p(\mathbf{y_t} \mid \boldsymbol{\eta}_t, \boldsymbol{\zeta}_t)p(\boldsymbol{\eta}_t)p(\boldsymbol{\zeta}_t)d\boldsymbol{\eta}_t d\boldsymbol{\zeta}_t} \end{aligned} \tag{10}$$

As is common in Bayesian inference problems, the integral in this case is intractable, and cannot be solved analytically. In this work, the posterior distribution is approximated using the Monte Carlo Markov Chain (MCMC) method, via the no U-turn (NUTS) implementation of Hamiltonian Monte Carlo (HMC) [32].

4. A Metamodelling Approach

4.1. Hierarchical Bayesian Modelling for Multi-task Learning

In Section 3, the presented BSBR model can be described as being an example of single task learning (STL), since the task-level parameters $\boldsymbol{\eta}_t$ and $\boldsymbol{\zeta}_t$ are learned independently for each task; in this case, there is no-pooling (NP) of information between tasks. By contrast, a model containing parameters solely at the population-level which are shared across all tasks, can be said

to have a complete-pooling (CP) modelling structure. In this case, all tasks are treated as identical, which can lead to poor model generalisation to new tasks. Alternatively, hierarchical Bayesian models (also known as multi-level models) provide a middle-ground, containing both task-specific parameters (allowing for differences between population members), and population-level parameters, which encourage task parameters to be similar; this modelling structure can be described as having partial-pooling (PP) of information. These models are often used as an approach to multi-task learning (MTL), where the goal is to improve the performance and generalisation of a model, by training it on multiple related tasks simultaneously [33].

In the context of power-curve modelling, allowing PP of information makes intuitive sense; wind-turbines within a wind-farm are nominally identical, so they are likely to have similar power-curves (at a population-level), but differences may also arise from varying wake patterns at individual turbine locations (the task-level). An additional benefit of PP of information, is that data-poor tasks in particular are able to borrow statistical strength from data-rich tasks. While in this context data availability is not a significant problem, it is common in the field of data-driven SHM [16], which has led to these modelling approaches being used as a solution in the domain of population-based SHM (PBSHM) research [20, 21].

4.2. Metamodel Design

In the results in Section 6.1, correlations are observed among the learned $\boldsymbol{\eta}_t$ and $\boldsymbol{\zeta}_t$ parameters of the NP BSBR models, in relation to the turbine x and y spatial coordinates. This fact motivated the integration of a metamodel, which can be described as a model over models. The metamodel was designed to capture the inter-turbine relationships in power-curve parameters, which could be used to predict power for previously unobserved turbines, not necessarily in the training data. This model design has a tangible real-world benefit—if one were able to make accurate probabilistic predictions for a variable of interest, at a location without data (and perhaps even without associated telemetry), this may reduce the overall telemetry requirements among a population. While in the use case here data are readily available at each turbine location, it serves as a useful testbed to experiment with the modelling approach.

First-order linear metamodels were chosen here for both simplicity and to reduce the risk of overfitting to the (potentially noisy) training data. In

matrix form, the parameter vectors $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ are described via the linear metamodels,

$$\boldsymbol{\eta} = \mathbf{C}_{\mathbf{x}}\mathbf{M}_{\boldsymbol{\eta}_{\mathbf{x}}} + \mathbf{C}_{\mathbf{y}}\mathbf{M}_{\boldsymbol{\eta}_{\mathbf{y}}} \tag{11}$$

$$\boldsymbol{\zeta} = \mathbf{C}_{\mathbf{x}}\mathbf{M}_{\boldsymbol{\zeta}_{\mathbf{x}}} + \mathbf{C}_{\mathbf{y}}\mathbf{M}_{\boldsymbol{\zeta}_{\mathbf{y}}} \tag{12}$$

where $\mathbf{C}_{\mathbf{x}}$ and $\mathbf{C}_{\mathbf{y}}$ are the linear regression design matrices for the normalised x and y coordinates for each turbine, and $\mathbf{M}_{\boldsymbol{\eta}_{\mathbf{x}}}$ and $\mathbf{M}_{\boldsymbol{\eta}_{\mathbf{y}}}$ are the metamodel parameter matrices for parameter vector $\boldsymbol{\eta}$. Similarly, $\mathbf{M}_{\boldsymbol{\zeta}_{\mathbf{x}}}$ and $\mathbf{M}_{\boldsymbol{\zeta}_{\mathbf{y}}}$ are the metamodel parameter matrices for parameter vector $\boldsymbol{\zeta}$.

As in the NP BSBR model, *hyperpriors* were used to aid model convergence, and were placed over the metamodel parameter matrices assuming normal distributions,

$$\{\mathbf{M}_{\boldsymbol{\eta}}, \mathbf{M}_{\boldsymbol{\zeta}}\} \sim \mathcal{N}(0, 3) \quad \text{(element-wise)} \tag{13}$$

where $\mathbf{M}_{\boldsymbol{\eta}} = \left\{\mathbf{M}_{\boldsymbol{\eta}_{\mathbf{x}}}, \mathbf{M}_{\boldsymbol{\eta}_{\mathbf{y}}}\right\}$ and $\mathbf{M}_{\boldsymbol{\zeta}} = \left\{\mathbf{M}_{\boldsymbol{\zeta}_{\mathbf{x}}}, \mathbf{M}_{\boldsymbol{\zeta}_{\mathbf{y}}}\right\}$ and the *prior* means and standard deviations were set equal to 0 and 3 respectively, given the distribution of learned parameters seen for the individual NP BSBR models; these were again found to be weakly informative, while sufficiently aiding convergence. Figure 6 shows a graphical representation of the model. As with Figure 5, arrows represent distribution inputs, while dotted lines show calculated inputs, with plates indicating multiple instances of their contained nodes.

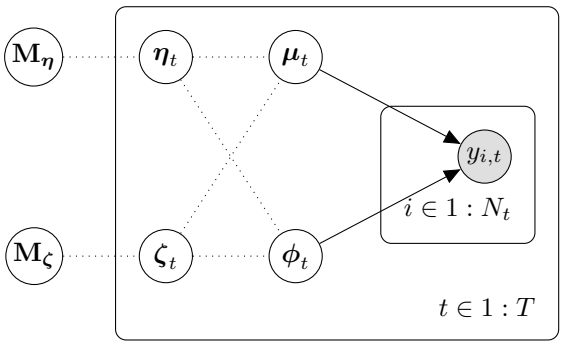


Figure 6: A graphical representation of the metamodel. Arrows leading from parameters indicate they are distribution parameters, whereas dotted lines indicate parameters are associated via a transformation.

Note that the metamodel parameters are located outside the plates, and are therefore learned at the population-level, while the parameters $\boldsymbol{\eta}_t$ and $\boldsymbol{\zeta}_t$ are determined specifically for each turbine.

4.3. Metamodel Predictions

Following posterior estimation of the population-level metamodel parameters, one can make probabilistic predictions at locations of interest, based on spatial coordinates alone,

$$\boldsymbol{\eta}^* = \mathbf{C}_\mathbf{x}^* \hat{\mathbf{M}}_{\boldsymbol{\eta}_\mathbf{x}} + \mathbf{C}_\mathbf{y}^* \hat{\mathbf{M}}_{\boldsymbol{\eta}_\mathbf{y}} \tag{14}$$

$$\boldsymbol{\zeta}^* = \mathbf{C}_\mathbf{x}^* \hat{\mathbf{M}}_{\boldsymbol{\zeta}_\mathbf{x}} + \mathbf{C}_\mathbf{y}^* \hat{\mathbf{M}}_{\boldsymbol{\zeta}_\mathbf{y}} \tag{15}$$

where $\boldsymbol{\eta}^*$ and $\boldsymbol{\zeta}^*$ are the predicted metamodel parameters for all turbine locations of interest, $\mathbf{C}_\mathbf{x}{}^*$ and $\mathbf{C}_\mathbf{y}{}^*$ are their associated first-order linear regression design matrices for the normalised spatial coordinates, and $\hat{\mathbf{M}}_{\boldsymbol{\eta}_\mathbf{x}}$, $\hat{\mathbf{M}}_{\boldsymbol{\eta}_\mathbf{y}}$, $\hat{\mathbf{M}}_{\boldsymbol{\zeta}_\mathbf{x}}$ and $\hat{\mathbf{M}}_{\boldsymbol{\zeta}_\mathbf{y}}$ are the estimated metamodel parameters. The predicted parameters can then be used in line with the population-level input features, $\mathbf{X}$, to make probabilistic power-predictions.

4.4. Metamodel Benchmarking

The model presented in Section 4 is a hierarchical Bayesian model, incorporating a metamodel—this model is simply referred to as the metamodel. To help assess the performance of the metamodel, it is benchmarked against three models which have no metamodel structure or spatial component. The first of these, is the NP BSBR model that was first introduced, which treats each turbine independently—i.e. there is no-pooling of information between turbines, with separate sets of parameters for each one. Secondly the metamodel is compared to a partially-pooled (PP) hierarchical model (shown in Figure 7), where the turbine-specific parameters are drawn from shared population-level distributions. For completeness, the metamodel is compared to a complete-pooling (CP) approach, where all turbines are considered identical and a single set of model parameters are learned for all turbines; this is shown in Figure 8. In the PP model, the η and ζ parameter vectors were assumed to be normally distributed, controlled by the population-level distribution parameters $\left\{\boldsymbol{\mu}_\eta, \boldsymbol{\sigma}_\eta, \boldsymbol{\mu}_\zeta, \boldsymbol{\sigma}_\zeta\right\}$. For further details, including the

details of the *hyperpriors* placed over the parameters in the models, `stan` code containing the four models is provided in the links in Section 8.

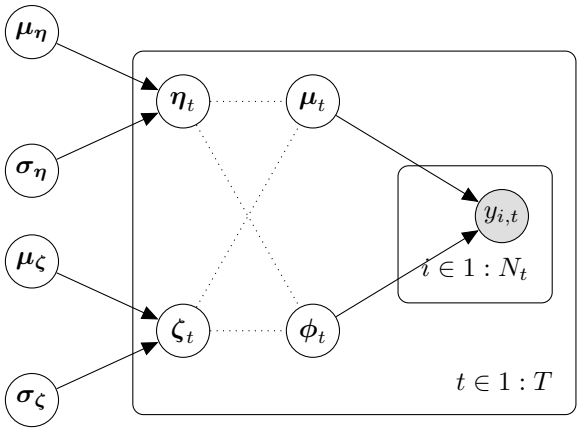


Figure 7: Partially-pooled (PP) BSBR graphical model.

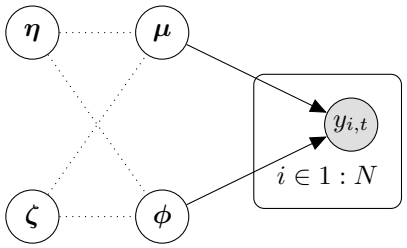


Figure 8: Completely-pooled (CP) BSBR graphical model.

5. Model Training and Testing

5.1. Training and Testing Datasets

5.1.1. BSBR Model Demonstration

In the first instance, to demonstrate the suitability of the BSBR model for capturing the heteroscedastic and asymmetric nature of power-curves, the model is trained and tested on individual turbines (i.e. using the no-pooling model definition). A subset of the filtered data that had undergone stratified sampling by yaw angle is used, to provide the models with an approximately even distribution of data across yaw angles, to try to help make the models more generalisable to all wind-directions. The training and testing datasets consisted of 4000 and 1000 data points per turbine respectively.

5.1.2. Metamodel

In the SCADA data, it was noted that there were significant proportions at zero or maximum power; because of this, additional stratified sampling was applied based on measured turbine output power. The reasons for doing this were twofold: (1) to train and assess models equally for all regions of the power-curve (2) because of the heteroscedastic nature of the power-curve, harmonising the proportions of data in different regions of the power-curve between turbines was considered to make comparisons between predictive metrics more reasonable. Since the metamodel aims to capture spatial correlations induced by wake-effects, reason (1) is particularly important, since wake-effects are more prevalent in the transitional region of the power-curve [23], which is comparatively data-sparse. The distributions of the resulting (sub)sampled data are shown in Figures 2a and 2b.

Additionally, for the metamodel, since a single model is learned for all data simultaneously, computational expense is increased significantly, so a reduced number of training points were needed. The resulting training and testing sets, having undergone stratified sampling on both the yaw angle and output power, consisted of just 250 points per turbine, which were used for the metamodel and benchmark models. Furthermore, since the metamodel and benchmark models are to be trained on a subset of the turbines, as an example, one quarter of the turbines were used, which were approximately equally-spaced throughout the wind-farm. The chosen training and the testing turbines are highlighted in Figure 9. The resulting combined training and test sets consisted of 5000 and 20000 data points respectively.

5.2. Model Training

As previously discussed, MCMC is used to estimate the posterior distributions of our parameters, given observed levels of power per turbine. For all the models in this study, MCMC was implemented using `stan` [34], with 2000 burn-in and 2000 inference samples per chain, and four chains. In all cases, the *maximum* $\hat{R}$ convergence statistics did not exceed 1.01, which suggests proper mixing of the chains [35]. Additionally, the *minimum* effective sample sizes (ESS) were greater than 400, a benchmark considered suitable for stable parameter estimates and proper interpretation of the $\hat{R}$ statistics [35].

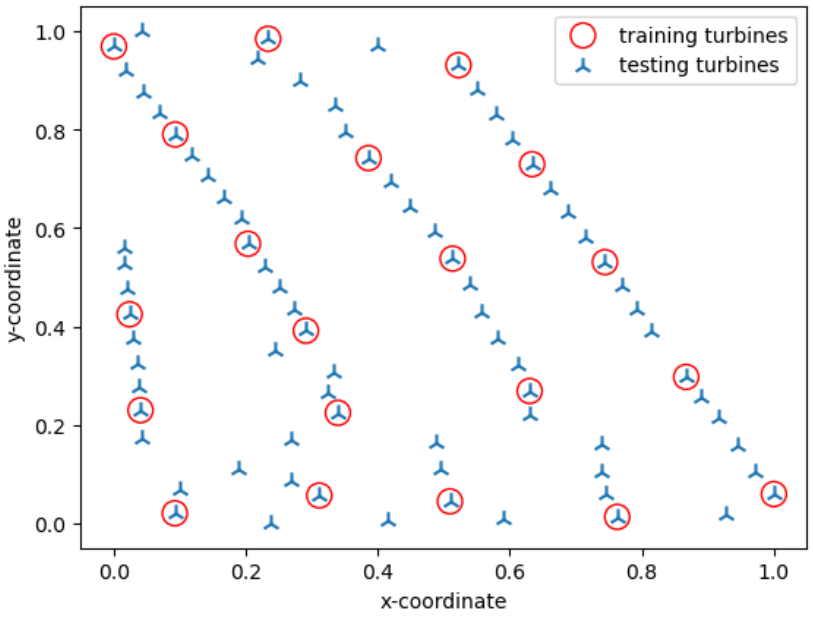


Figure 9: Map showing the training and testing turbines for the metamodel. Red circles show the locations of the turbines used in training, while all turbines (shown by the blue markers) were used in testing.

5.3. Model Scoring

In order to quantify the success of each of the models, two performance metrics were used. Firstly, the normalised mean squared error (NMSE) assesses the goodness-of-fit, or how close point predictions are to their test value. The NMSE functions similarly to percentage error, where a score of zero indicates a perfect model, while a score of 100 is equivalent to a model simply predicting the mean value of the data. NMSE is defined as,

$$\text{NMSE} = \frac{100}{N\sigma_y^2}\sqrt{(\mathbf{y}-\widehat{\mathbf{y}})^\intercal(\mathbf{y}-\widehat{\mathbf{y}})} \tag{16}$$

where N is the number of data points, σ_y^2 is the variance of the measured data, $\mathbf{y}$ the measured data, and $\widehat{\mathbf{y}}$ the model predictions.

Since the models are probabilistic and include predictive variance, the joint-log-likelihood (JLL) is also calculated as a measure of how well the uncertainty in the measured data was captured by the model. The JLL is calculated as,

$$JLL = \frac{\sum_{s=1}^{S} \sum_{i=1}^{N_t} \log f(y_{i,t_s} | \mu_{i,t_s}, \phi_{i,t_s})}{S} \tag{17}$$

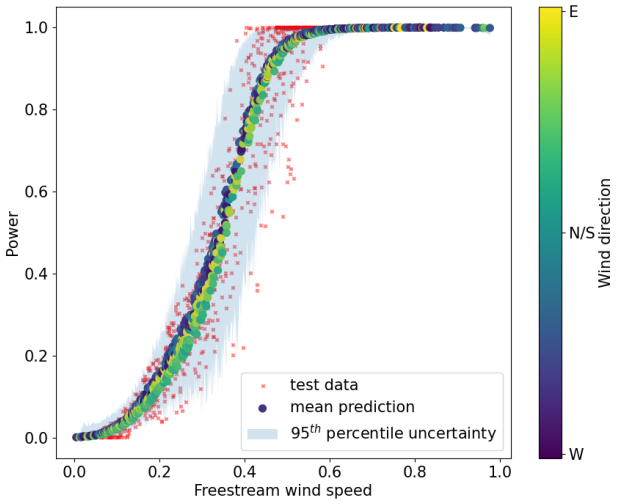
where $f(y_{i,t_s} | \mu_{i,t_s}, \phi_{i,t_s})$ is the probability density function of the beta distribution as defined in equation (1), and S is the number of posterior samples. The larger the JLL, the better the model has captured the true distribution of the test data. Given the prevalence of power-curve modelling in the literature, it is worth stating that since the model metrics are dependant upon both the training features and test data used, they are best used as a comparison of the models within this paper only.

6. Results and Discussion

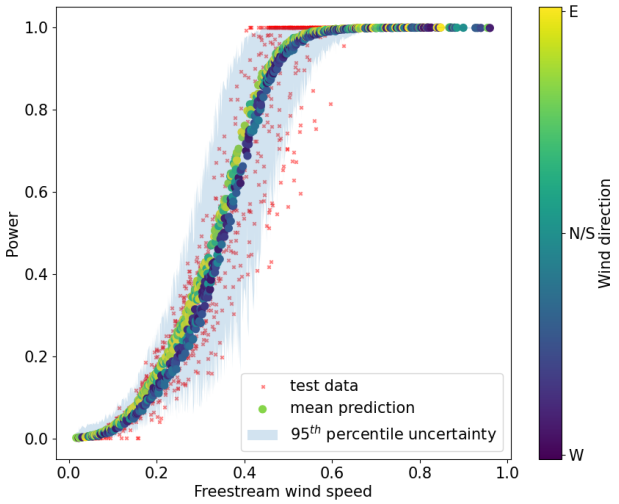
6.1. BSBR Model Demonstration

To first demonstrate probabilistic predictions that can be made with the BSBR models (with no-pooling), Figure 10 shows predictions from models of turbines on both the west and east-side of the wind-farm in subplots (a) and (b) respectively, using the 1000 testing data points in each case. Test data are indicated by red crosses, model mean predictions are indicated by circles (which are coloured by the value of the wind-direction), and model uncertainty is represented by the blue shaded area–indicating the 95^{th}-percentile of predictions.

There are a number of interesting observations. Firstly, the predictive uncertainty appears to match the data, with reduced variance at the lower and upper ends of the curve, and increased variance in the transitional region between turbine minimum and maximum power. Approximately 5% of the test points also lie outside the 95^{th}-percentile of the predictive variance, as should be expected for a correctly-trained model. The "rough" edges of this uncertainty are from data points that, while having a very similar freestream wind-speed, may have significantly different wind-directions, which also contribute to the predictive uncertainty. Secondly, while the majority of the influence on output power appears to be associated with the freestream wind-speed (as expected), the models have allocated some weight to the wind-directional features, resulting in the variance in the mean predictions for a given freestream wind-speed; this is mostly prevalent in the transitional region between minimum and maximum power, and matches intuition since wind-turbine wakes affect this region the most. However, in this region it is clear that significant deviation between model mean predictions



(a) West-side turbine.



(b) East-side turbine.

Figure 10: NP BSBR model predictions for two separate turbines. Test data (1000 points) are shown by red crosses, mean predictions are shown as circles, coloured by the wind-direction. The blue shaded areas show the 95^{th}-percentile of the predictive variance.

and the test data remains. This deviation is likely to be the result of a number of factors, including the oversimplification of the wake-effect in assuming it can be approximated based on the wind-direction. In reality, this fails to account for atmospheric conditions such as wind shear, turbulence intensity,

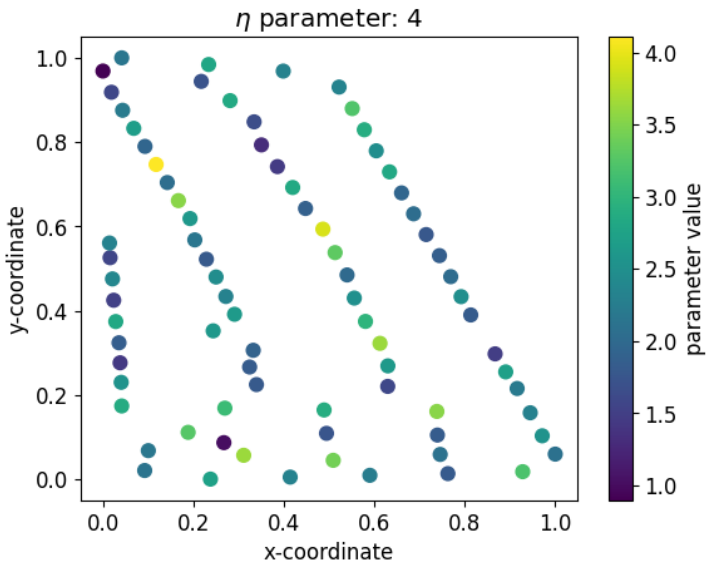
thermal stratification and air-density variations, which are known to have an impact on wake propagation [36, 37]. Furthermore, in calculating the input features, the wind-direction (yaw angle) is assumed to be the same for all turbines—this is not the case, as there is a distribution of localised wind-directions across the wind-farm (yaw angles), which will likely change the wake patterns. There are also additional sources of noise within the data, such as sensor calibration errors in the yaw and wind-speed measurements which are used in calculating the population-level features, as well as possible changes in the underlying behaviour of the turbines, perhaps because of wear or damage during the period of the training data used. Additionally, *curtailed*, *shut-down* or even *power-boosted* turbines upstream of a given turbine will further generate noise in the wake patterns.

Despite these problems, comparison of the west and east-side models yields some evidence that a level of turbine-dependent wind-directionality has been captured. For example, for the west-side turbine, it is not expected to be influenced by wakes when wind approaches from the western direction. Conversely, the east-side turbine is not expected to be influenced by wakes when wind blows from the east. In each case, the models appear to predict higher levels of power when these conditions are true, and less power when they are not, as one would expect to be the case in reality. While not shown here, these characteristics can also be seen for other turbine models around the perimeter of the wind-farm, to similar or lesser degrees.

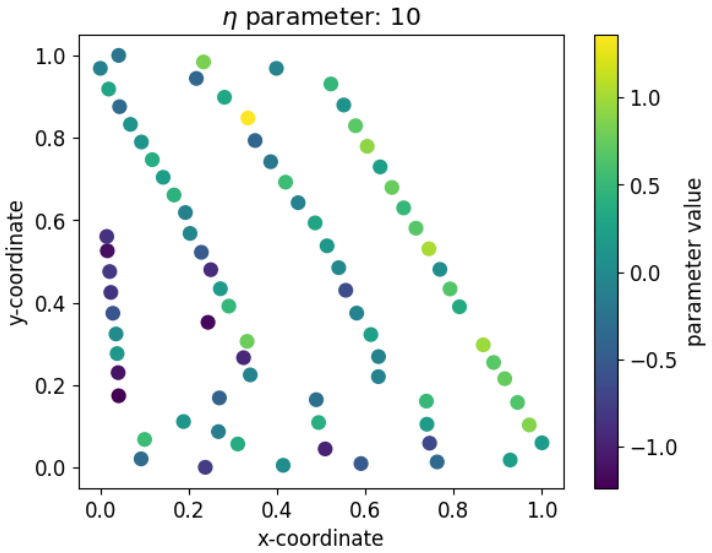
To explore this model behaviour further, the learned model parameter vectors which result in differences in predicted power may be inspected. In this case, each of the vectors $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ consist of 18 parameters, with parameters 1-6 associated with Feature One, 7-12 Feature Two and 13-18 Feature Three. Figure 11 shows the value of selected parameters from the $\boldsymbol{\eta}$ vector plotted for each turbine within the wind-farm. The colour of the points denotes the values of the parameters as indicated by the colour bars.

For Parameter Four (associated with Feature One), there does not appear to be an obvious spatial correlation. However, while their magnitude is smaller than Parameter Four, Parameters Ten and 13 (associated with Features Two and Three), shown in subfigures 11b and 11c, show clear spatial correlation across the wind-farm. For Parameter Ten, there is an approximate negative correlation from east to west, while there is an approximate south to north negative correlation for Parameter 13. While not shown here, similar observations can be made for some of the remaining parameters in the $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ vectors. These spatial differences in parameters represent the

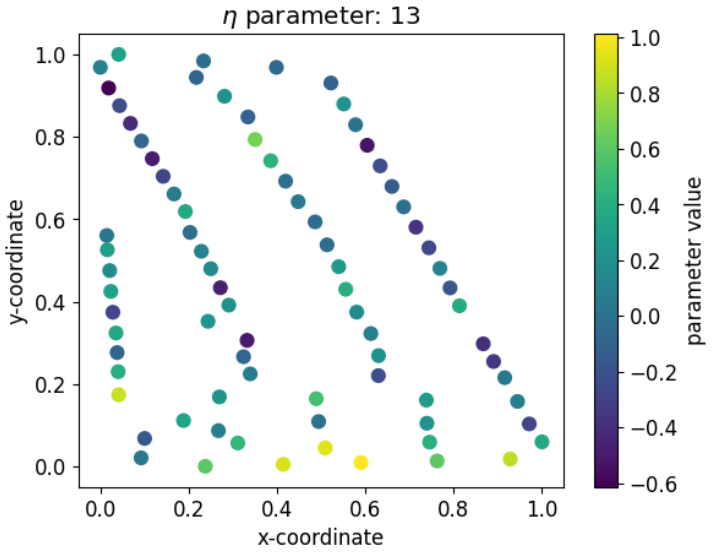
wake-effect captured by the NP models, and motivated the development of the metamodel, which aims to remove the requirement of having data from each individual turbine, by utilising these spatial correlations in the individual model parameters. The use of a hierarchical model in this case also allows "statistical strength" to be shared between turbine-specific parameters, via the population-level parameters.



(a) 4th parameter in the learned $\boldsymbol{\eta}$ vectors.



(b) 10th parameter in the learned $\boldsymbol{\eta}$ vectors.

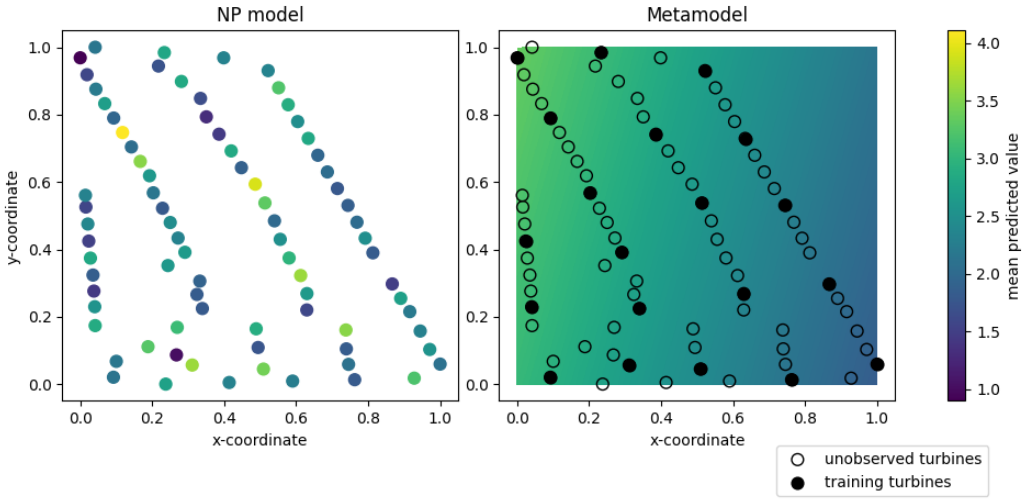


(c) 13th parameter in the learned $\boldsymbol{\eta}$ vectors.

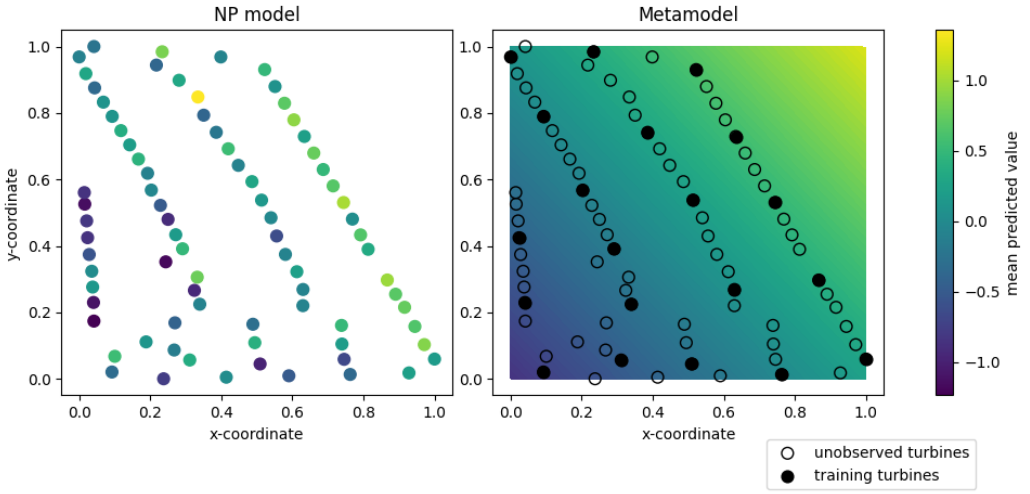
Figure 11: Mean parameter values of selected entries of the learned BSBR model $\boldsymbol{\eta}$ vectors, plotted by turbine location. Colours indicate the value of the parameters.

6.2. Metamodel

The metamodel learns linear models over the parameter vectors, for turbine x and y coordinates. The right-hand plots in sub-Figures 12a and 12b show metamodel mean predicted values by colour, for $\boldsymbol{\eta}$ Parameters Four and Ten respectively, spatially. The black filled circles denote the locations of the turbines used for training the metamodel, while the unfilled black circles represent the locations of the remaining (unobserved) turbines that were also used in testing. These predictions are compared directly with the NP BSBR models, shown in the left-hand plots. As with Figure 11a, there is no clear observable spatial correlation for Parameter Four seen for the NP model; there are also some potential outliers—such as those shaded more yellow in the upper half of the wind-farm. For the same parameter in the metamodel, the regularising effect of the linear model assumption removes the outliers in prediction, while perhaps only a weak correlation in learned values across the farm is observed. For Parameter Eight it can be seen that the metamodel has approximated the more clear observable correlation seen in the NP model parameters. These observations provide confirmation that the metamodel is working as intended.



(a) 4th parameter in the learned $\boldsymbol{\eta}$ vectors.



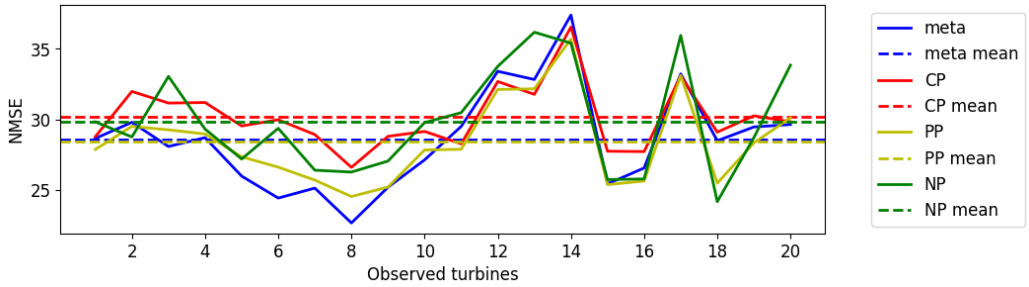
(b) 10th parameter in the learned $\boldsymbol{\eta}$ vectors.

Figure 12: Mean predicted values for the Fourth (subplot (a)) and Tenth (subplot (b)) entries in the $\boldsymbol{\eta}$ parameter vector, plotted spatially for each turbine in wind-farm *Ciabatta*. In each subplot, the left-hand plots show the mean predictions from the NP model, while the right-hand plots show the mean metamodel predictions. Filled circles represent turbines used in training, while unfilled circles represent turbines used in testing only. The points are shaded by their values.

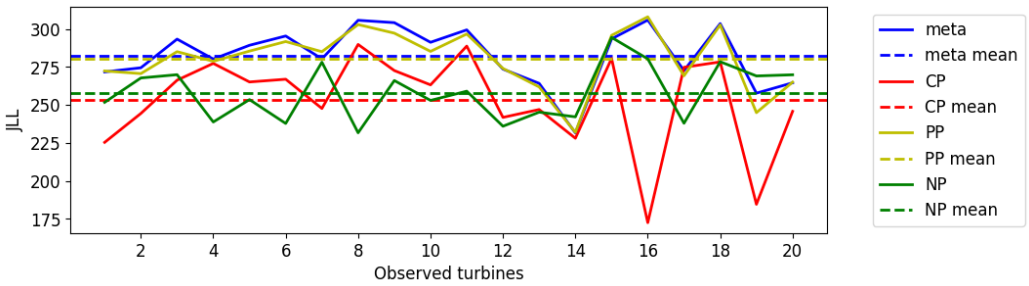
6.2.1. Model Scores

The metamodel is compared against the benchmark NP, PP and CP models, using the scoring metrics. In the case of the NMSE, a lower score indicates improved model performance, while for the JLL a higher score indicates an improved model fit. Firstly, Figure 13a shows the NMSE per turbine, for each model, for the *observed* turbines only (i.e. those used in training). Similarly, Figure 13b shows the JLL per turbine, for each model. For both metrics, it can be seen that on average, the metamodel and PP models score similarly to each other, and marginally better than both the CP and NP models, which themselves score similarly to each another.

Secondly, Figures 14a and 14b show the NMSE and JLL per turbine, for each model, for the *unobserved* turbines; as a reminder, the exception to this is the NP models, which require data from every turbine, and so all turbines are individually observed in this case. For these otherwise unobserved turbines, it can be seen that on average the metamodel performs marginally better than all three of the other models for both the NMSE and JLL metrics, including the NP models despite not having observed data from these



(a) NMSE scores on the observed turbines

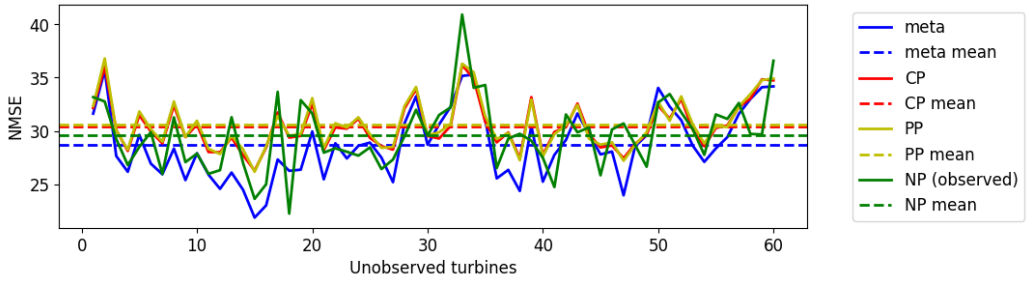


(b) JLL scores on the observed turbines

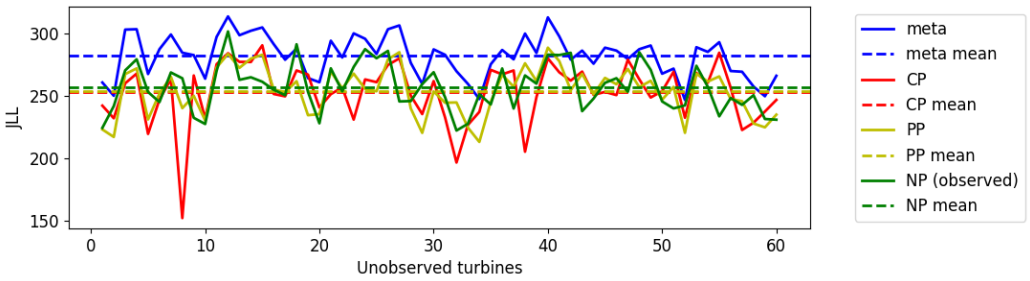
Figure 13: Comparison of model scores for each of the observed turbines in wind-farm *Ciabatta*. Dashed horizontal lines indicate mean scores per model.

turbines in training.

Across all observed and unobserved turbines, the CP model scores particularly poorly on the JLL metric in five instances: the observed turbines 16 and 19, and the unobserved turbines eight, 32 and 38. Since the CP model learns a single set of shared population-level parameters which are used for prediction, it is not able to adjust for turbine-specific differences, which may be the case in these instances. For the unobserved turbines, the PP model is limited in the same way as the CP model, since the shared population-level parameters must be used for making predictions; this explains their similar average scores. However unlike the CP model, the PP model does not share the same sharp drops in the JLL metric (turbines eight, 32 and 38), suggesting that partial-pooling of information provides improved adaptation to the model. In the case of the observed turbines, the PP model is able to use the learned turbine-specific weights in prediction, and so can make adjustments which improve the model scores relative to the CP model, which are of a



(a) NMSE scores on the unobserved turbines



(b) JLL scores on the unobserved turbines

Figure 14: Comparison of model scores for each of the unobserved turbines in wind-farm *Ciabatta*. Dashed horizontal lines indicate mean scores per model.

similar performance to the metamodel.

The metamodel combines strengths from all three approaches: information is pooled reducing the effective data-scarcity (like the CP and PP models), whilst turbine-specific adjustments may also be made (like the PP and NP models). Importantly however, the metamodel can additionally make adjustments for unobserved tasks, resulting in a distinct improvement in model performance for these tasks. Compared to the next best performing model for the unobserved turbines (the NP models), the metamodel scores were improved by 3.2% and 9.8% for NMSE and JLL metrics respectively.

A final observation is that there are some relatively-high NMSE scores (and reduced JLL scores) across all models for the observed turbines 12–14, and the unobserved turbines 32–35, compared to the surrounding turbines; to explore this further, the scoring metrics for the metamodel are plotted spatially in Figure 15. Here, in general the scores appear to be worse towards the lower end of the wind-farm; since these turbines follow a less rigid spatial pattern than the turbines at the upper end, they are likely to be a source

of additional noise in the spatial wake patterns of these turbines, which may make accurate prediction of output power using just the freestream wind speed and direction features used in this work more challenging.

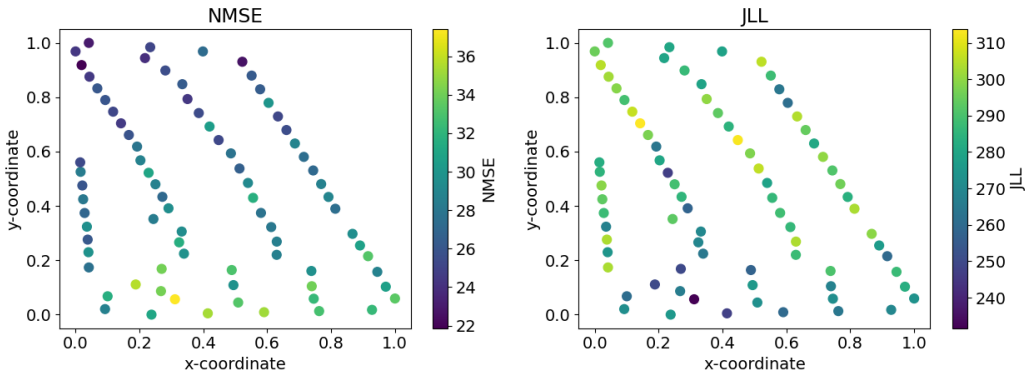
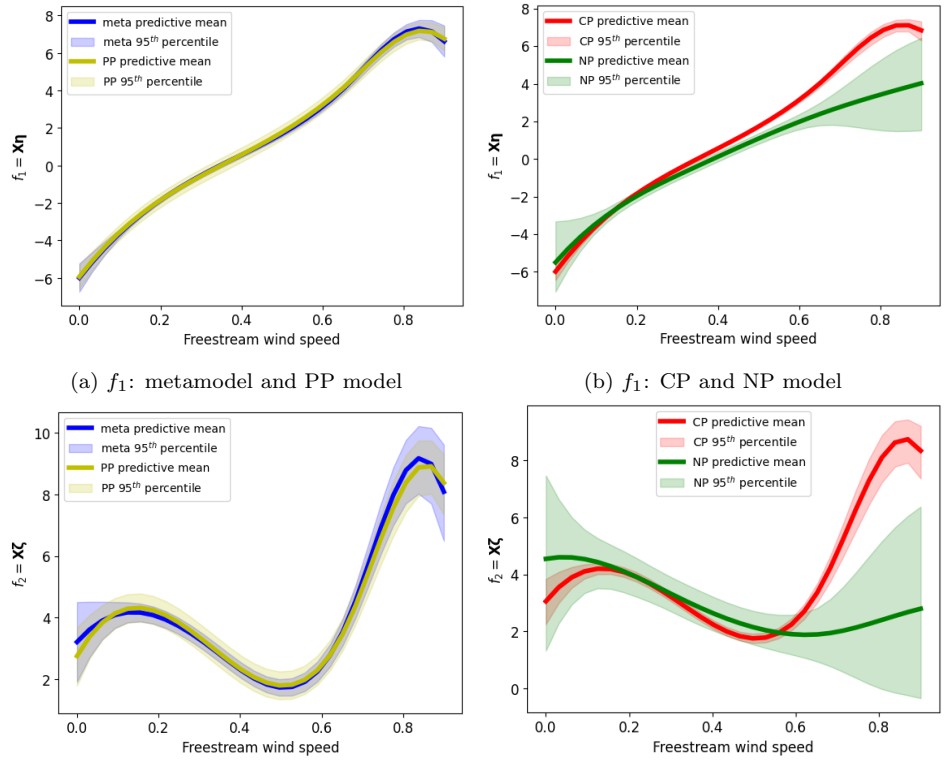


Figure 15: Maps of the metamodel scores across the wind-farm. The left-hand plot shows the NMSE per turbine, while the right-hand plot shows the JLL per turbine.

6.2.2. Predicting Unobserved Turbines

The models are now compared in prediction, for an unobserved turbine—for consistency, the same turbine as previously predicted for in Figure 10b was chosen. Figures 16a–16d first compare the learned linear models f_1 and f_2 across all four models. A gridded input over all three features was used, with averages of the predictive mean and 95^{th} percentile taken over the wind-directional features (Features Two and Three). Mean predictions are shown as solid lines, while 95^{th}-percentiles are shown as shaded areas.

To aid interpretation, f_1 is proportional to the mean prediction as per Equation (4), while f_2 is proportional to the precision (see Equation (5)), which is interpreted as the inverse of the variance. While the CP, PP and metamodels are broadly similar in these plots, there are significant differences in the function shape and 95^{th} percentile that can be seen for the NP model. As a result of the stratified sampling carried out on the training and test data, on a turbine-by-turbine basis there is relative data-scarcity at high freestream wind speeds; for an individual NP model such as this one, this scarcity is likely to result in both increased uncertainty in the trained model weights in this region of the curve, and a tendency for these weights to also be more biased towards their *prior* distribution, which in this case were centred on zero. These factors likely both contribute towards f_1 being pulled



(a) f_1: metamodel and PP model

(b) f_1: CP and NP model

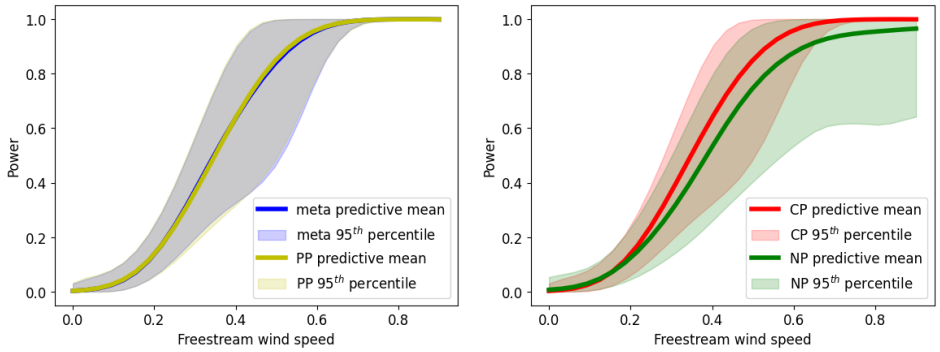
(c) f_2: metamodel and PP model

(d) f_2: CP and NP model

Figure 16: Posterior predictive distributions for functions f_1 and f_2, for the *unobserved* turbine on the east-side of the farm, using a gridded input over all three features. Averages were taken over Features Two and Three. Solid lines represent the mean prediction, while shaded areas show the 95^{th}-percentile.

downwards at higher wind speeds (and with increased uncertainty), and f_2 (proportional to precision) being relatively smaller (resulting in increased variance in the beta distribution). However, for both f_1 and f_2 the CP, PP and metamodels are not completely free of the issues of data-scarcity, as at high freestream wind speeds (beyond 0.8), the curves begin to decrease, which does not match the shape of the power-curve training data—power does not decrease at high wind speeds, nor does its variance increase after rated power is reached (assuming the cut-out wind speed is not reached). This model behaviour is perhaps a limitation of its current design, which could be improved by placing tighter, more positive priors for the higher freestream wind speed weights, which could be reasonably justified based on known behaviour of wind-turbines. More generally for the f_1 and f_2 functions, it can be said that the metamodel, PP and CP models benefit from pooling of information, resulting in narrower 95^{th} percentile predictions for both functions, despite not having observed data from this turbine.

These observations can directly be seen in the full probabilistic power-predictions—shown in Figure 17. Here, the mean prediction of the NP model can be seen to be lower than the other models, with increased uncertainty at higher freestream wind speeds as indicated by the f_1 and f_2 functions. The CP, PP and metamodels all show broadly similar predictions, which are well-matched to the shape of power-curve data. In this space, the effects of the reducing f_1 and f_2 functions at high freestream wind speeds appears to be negligible.



(a) Metamodel and PP model.

(b) CP and NP model.

Figure 17: Probabilistic power-predictions for the *unobserved* turbine, over gridded input data, averaged for the wind-direction features

Finally, to demonstrate how the metamodel is able to make location-specific adaptations in prediction for different wind-directions (which the CP and PP models cannot), Figure 18 shows power predictions for the same unobserved turbine, for easterly (90°), southerly (180°), westerly (270°) and northerly (360°) winds. As was shown for the model predictions in Figure 10b, the model mean prediction is greater in the wake-effected region of the power-curve during easterly winds (as compared to westerly winds), which matches intuition.

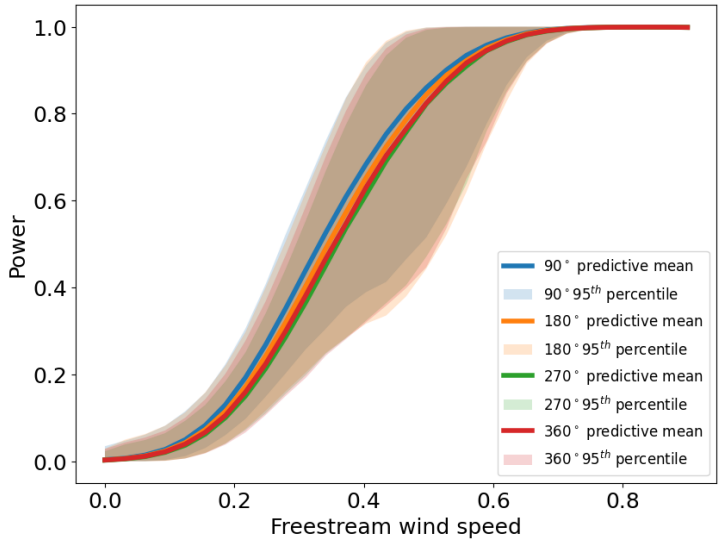


Figure 18: Probabilistic power-predictions for the metamodel for the *unobserved* turbine 35, over a gridded input of freestream wind-speeds, for easterly (90°), southerly (180°), westerly (270°) and northerly (360°) wind-directions.

6.3. Model Computational Complexity

To compare the computational complexity of the models, Table 2 compares both the computational complexity of the likelihoods, and the number of learned parameters in each case.

Model	T	B	N	**Likelihood complexity**	**Number of parameters**
No-pooling	80	18	250	$\mathcal{O}(NT) = \mathcal{O}(20000)$	$2B \times T = 36 \times 80 = 2880$
Complete-pooling	20	18	250	$\mathcal{O}(NT) = \mathcal{O}(5000)$	$2B = 36$
Partial-pooling	20	18	250	$\mathcal{O}(NT) = \mathcal{O}(5000)$	$2BT + 2B + 2 = 758$
Meta model	20	18	250	$\mathcal{O}(NT) = \mathcal{O}(5000)$	$8B = 144$

Table 2: Comparison of the model likelihood complexity, and the number of parameters for each model. T: number of training turbines, B: width of the design matrix, N: data points per training turbine.

While the number of iterations, and the model architecture influence overall computational complexity, the likelihood calculation is repeated every HMC iteration, and so should contribute significantly to the overall computational cost; in this case, since the CP, PP and metamodels are trained on 20 of the turbines, their likelihood complexity is one quarter of the overall cost of the 80 NP models. Additionally, the total number of parameters learned in each model varied significantly; the NP and PP models learn parameters for every turbine used in training (totalling 2880 and 758 respectively), while the CP and metamodels only learn parameters at the population-level (36 and 144 respectively). A higher number of parameters are also likely to increase the per-iteration cost, while also increasing the complexity of the *posterior* geometry, which affects how efficiently HMC can move [31]. In practice, model training varied in the order of hours for the CP, PP and metamodels, while it was in the order of a day for all 80 of the NP models, which roughly correlates with the increased total likelihood complexity, and number of parameters for these models.

7. Conclusions

A probabilistic, multivariate wind-turbine power-prediction model was developed, utilising a beta regression model with B-spline basis-functions (named the BSBR model). This BSBR model was shown to be successful in capturing both the heteroscedastic and asymmetric properties of power-curve data. When training the BSBR model on each turbine within a wind-farm, correlations were observed in the learned model parameters with respect to turbine locations, which were deemed to be associated with the directional dependency of wind-turbine wakes.

These spatial correlations in parameters were then leveraged via a "meta-model", which imposed functional constraints over turbine-specific parameters, as part of a hierarchical Bayesian model. This model design is able to learn from all turbines (tasks) simultaneously, and via the metamodel parameters, can be used to make adaptive probabilistic predictions for turbines not necessarily included in the training data, using their spatial coordinates alone. Across an 80-turbine wind-farm, the metamodel was found to outperform three benchmark models that utilised no-pooling, partial-pooling and complete-pooling of information, both in terms of normalised mean squared error (NMSE) and joint log-likelihood (JLL) in the majority of cases, while having a relatively low number of model parameters.

Since the models are trained on farm-level input features to predict turbine-level output power, they are tuned (to a degree) on the specific environmental conditions (including wake-effects) of the wind-farm used in this work, which are also related to the specific layout of the wind-farm, as well as the turbine-type. If one wishes to apply this model to another wind-farm of interest, the model parameters should therefore be re-learned on data directly from that wind-farm in order to obtain valid results.

While in practice, training data for all turbines in a wind-farm are often available for power-curve prediction, the results suggest that the metamodel design is an efficient modelling solution for situations where data may be limited, and correlations could be expected in a variable of interest among a population of structures. Potential further applications in this domain include (but are not limited to): structural load assessment for fatigue-life estimation, condition monitoring and fault detection, and design for wind-farm layouts. As another perspective, suppose one wishes to measure a new variable of interest in a population of structures; this modelling approach could justify instrumenting just a subset of this population. By leveraging shared information across the subset of the population, the model may estimate the variable of interest with sufficient precision for the remaining structures without sensors, reducing the overall cost of sensor deployment.

As highlighted in Section 6.1, the complexity of wake-effects cannot be fully captured by the wind-direction alone, which likely limits the performance of all the models in predicting power in the region between the cut-in and the rated wind-speeds. Future work could consider incorporating additional explanatory variables, such as wind-shear, turbulence-intensity, thermal-stratification and air-density variations, which may improve predictive performance. For the turbulence-intensity, if the standard-deviation of

the wind-speed was available in the SCADA data (which is sometimes the case), it may be estimated as the ratio of the standard-deviation to the mean wind-speed [38]. The incorporation of weather forecast (or hindcast) data may also help in gaining information for the additional variables, which could be obtained from a provider such as ECMWF [25]. The additional uncertainty associated with forecasted data does need to be considered however, as well as how to combine the 10-minute SCADA data with the weather data, which is likely to be coarser. If turbine-specific data are used, then the incorporation of computational fluid dynamics-based wake modelling—or a surrogate model with lower computational cost—may offer further improvements. The *PyWake* wind-farm simulation tool [39] may be a practical option for such surrogate modelling.

A number of simplifying assumptions were made in relation to the training features, to help capture wake-effects from the data; these features may be improved if meteorological mast data were available, which would likely be a more accurate reflection of freestream wind conditions. Given the complexity of turbine wake physics, future work could also consider using physics-based inputs, to develop a hybrid data and physics-based approach. Datasets labelled with turbine operating regimes, and instances of damage or repair that may indicate a change in underlying turbine behaviour, may also help with data preprocessing to reduce possible sources of noise. Model design may also be improved by optimising the number of B-spline knots per feature, or their locations—perhaps by increasing the density of knots in the regions of the cut-in wind-speed and where rated power is reached, since these regions have increased curvature. Alternative spline-modelling approaches could also be considered. The use of nonlinear metamodels may also be worth considering, particularly if parameter spatial correlations are complex.

If one wishes to scale the model for larger wind-farms, or perhaps with more data, it may be practical to reduce the computational cost. One approach could be the use of variational inference (instead of HMC), where the *posterior* is approximated as one of a family of known distributions [31]; this however may come at the cost of accuracy. Keeping the total number of model parameters as low as reasonably practicable is also likely to reduce the computational burden.

8. Data availability

For the purpose of reproducibility, the `stan` code used in this work has been made available online at: `https://github.com/smbrealy/wind_mtl`. The data used are part of a Non Disclosure Agreement and cannot be shared.

9. Acknowledgements

The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) and the Natural Environment Research Council (NERC), via grant references EP/W005816/1, EP/R003645/1 and EP/S023763/1. The authors also gratefully acknowledge Vattenfall R&D for their time and for providing access to the data used in this study. For the purpose of open access, the author(s) has/have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

References

[1] COP28, Global Renewables And Energy Efficiency Pledge, 2023.

[2] Guide to an Offshore Wind Farm, Technical Report, BVG Associates, 2019.

[3] J. Tautz-Weinert, S. J. Watson, Using SCADA data for wind turbine condition monitoring – a review, IET Renewable Power Generation 11 (2017) 382–394.

[4] J. H. Mclean, M. R. Jones, B. J. O'Connell, E. Maguire, T. J. Rogers, Physically meaningful uncertainty quantification in probabilistic wind turbine power curve models as a damage-sensitive feature, Structural Health Monitoring 22 (2023) 3623–3636.

[5] T. J. Rogers, P. Gardner, N. Dervilis, K. Worden, A. E. Maguire, E. Papatheou, E. J. Cross, Probabilistic modelling of wind turbine power curves with application of heteroscedastic Gaussian process regression, Renewable Energy 148 (2020) 1124–1136.

[6] C. Gilbert, Topics in high dimensional energy forecasting, Ph.D. thesis, University of Strathclyde, Glasgow, 2021.

[7] S. Hanifi, X. Liu, Z. Lin, S. Lotfian, A critical review of wind power forecasting methods—past, present and future, Energies 13 (2020) 3764.

[8] Y. Wang, Q. Hu, L. Li, A. M. Foley, D. Srinivasan, Approaches to wind power curve modeling: A review and discussion, Renewable and Sustainable Energy Reviews 116 (2019) 109422.

[9] G. P. Bazionis I.K., Review of deterministic and probabilistic wind power forecasting: Models, methods, and future research, Electricity **2** 13–47 (2021).

[10] A. Stetco, F. Dinmohammadi, X. Zhao, V. Robu, D. Flynn, M. Barnes, J. Keane, G. Nenadic, Machine learning methods for wind turbine condition monitoring: A review, Renewable Energy 133 (2019) 620–635.

[11] J. Maldonado-Correa, S. Martín-Martínez, E. Artigao, E. Gómez-Lázaro, Using SCADA data for wind turbine condition monitoring: A systematic literature review, Energies 13 (2020) 3132.

[12] R. Pandit, D. Astolfi, J. Hong, D. Infield, M. Santos, SCADA data for wind turbine data-driven condition/performance monitoring: A review on state-of-art, challenges and future trends, Wind Engineering 47 (2023) 422–441.

[13] S. Jawalageri, R. Ghiasi, S. Jalilvand, L. J. Prendergast, A. Malekjafarian, A data-driven approach for scour detection around monopile-supported offshore wind turbines using Naive Bayes classification, Marine Structures 95 (2024) 103565.

[14] A. Rytter, Vibrational Based Inspection of Civil Engineering Structures, Ph.D. thesis, University of Aalborg, Denmark, 1993.

[15] F.-G. Yuan, S. A. Zargar, Q. Chen, S. Wang, Machine learning for structural health monitoring: challenges and opportunities, in: H. Huang (Ed.), Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020, volume 11379, International Society for Optics and Photonics, SPIE, 2020.

[16] C. R. Farrar, K. Worden, Structural Health Monitoring: a Machine Learning Perspective, Wiley, 2012.

[17] Y. Zhang, Q. Yang, A survey on multi-task learning, IEEE Transactions on Knowledge and Data Engineering 34 (2022) 5586–5609.

[18] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proceedings of the IEEE 109 (2021) 43–76.

[19] Z. Sun, A. Barp, F.-X. Briol, Vector-valued control variates, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), Proceedings of the 40th International Conference on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, PMLR, 2023, pp. 32819–32846.

[20] L. A. Bull, D. Di Francesco, M. Dhada, O. Steinert, T. Lindgren, A. K. Parlikad, A. B. Duncan, M. Girolami, Hierarchical Bayesian modelling for knowledge transfer across engineering fleets via multitask learning, Computer-Aided Civil and Infrastructure Engineering (2022).

[21] T. A. Dardeno, K. Worden, N. Dervilis, R. S. Mills, L. A. Bull, On the hierarchical Bayesian modelling of frequency response functions, Mechanical Systems and Signal Processing 208 (2024) 111072.

[22] S. M. Brealy, A. J. Hughes, T. A. Dardeno, L. A. Bull, R. S. Mills, N. Dervilis, K. Worden, Multitask learning for improved scour detection: A dynamic wave tank study, 2024. ArXiv:2408.16527 [cs].

[23] M. F. Howland, S. K. Lele, J. O. Dabiri, Wind farm power optimization through wake steering, Proceedings of the National Academy of Sciences 116 (2019) 14495–14500.

[24] K. Worden, G. Manson, N. Fieller, Damage Detection Using Outlier Analysis, Journal of Sound and Vibration 229 (2000) 647–667.

[25] ECMWF, 2024. URL: `https://www.ecmwf.int/`.

[26] M. Capelletti, D. M. Raimondo, G. De Nicolao, Wind power curve modeling: A probabilistic Beta regression approach, Renewable Energy 223 (2024) 119970.

[27] P. Verhulst, Mathematical Research on the Law of Population Growth, volume 1, Hayez, 1845.

[28] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, Springer, 2013.

[29] C. de Boor, A Practical Guide to Splines, volume 27, 1978.

[30] L. Piegl, W. Tiller, The NURBS Book, Monographs in Visual Communication, 2 ed., Springer Berlin, 1996.

[31] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, D. Rubin, Bayesian Data Analysis, Third Edition, Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis, 2013.

[32] M. D. Hoffman, A. Gelman, The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo, Journal of Machine Learning Research 15 (2014) 1593–1623.

[33] K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT press, 2012.

[34] Stan Development Team, Stan Modeling Language Users Guide and Reference Manual, 2024. URL: `https://mc-stan.org`.

[35] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, P.-C. Bürkner, Rank-normalization, folding, and localization: An improved R^ for assessing convergence of MCMC (with discussion), Bayesian Analysis 16 (2021) 667–718.

[36] J. Nielson, K. Bhaganagar, R. Meka, A. Alaeddini, Using atmospheric inputs for artificial neural networks to improve wind turbine power prediction, Energy 190 (2020) 116273.

[37] R. J. Barthelmie, L. E. Jensen, Evaluation of wind farm efficiency and wind turbine wakes at the Nysted offshore wind farm, Wind Energy 13 (2010) 573–586.

[38] T. Göçmen, G. Giebel, Estimation of turbulence intensity using rotor effective wind speed in lillgrund and horns rev-i offshore wind farms, Renewable Energy 99 (2016) 524–532.

[39] M. M. Pedersen, A. M. Forsting, P. van der Laan, R. Riva, L. A. A. Romàn, J. C. Risco, M. Friis-Møller, J. Quick, J. P. S. Christiansen,

R. V. Rodrigues, B. T. Olsen, P.-E. Réthoré, Pywake 2.5.0: An open-source wind farm simulation tool, 2023. URL: https://gitlab.windenergy.dtu.dk/TOPFARM/PyWake.