

# UNCERTAINTY QUANTIFICATION WITH DEPENDENT INPUTS: WIND AND WAVES

ANNE W. EGGELS<sup>1</sup>, DAAN T. CROMMELIN<sup>1,2</sup>

<sup>1</sup> Centrum Wiskunde & Informatica  
P.O. Box 94079, 1090 GB Amsterdam, the Netherlands  
a.w.eggels@cwi.nl

<sup>2</sup> Korteweg-de Vries Institute for Mathematics, University of Amsterdam,  
P.O. Box 94248, 1090 GB Amsterdam, the Netherlands  
d.t.crommelin@cwi.nl

**Key words:** UQ, dependency analysis, sensitivity analysis, Gaussian processes

**Abstract.** A framework for performing uncertainty quantification is presented which is well-suited for systems with dependent inputs with unknown distributions. The multivariate input is given as a dataset whose variables can have strong, nonlinear dependencies. For each of the elements in the framework (dependency analysis, sample selection and sensitivity analysis), we recently developed new methods, which are here combined for the first time. The framework is tested on an example involving a wind farm simulation with offshore weather conditions as input.

## 1 INTRODUCTION

Uncertainty in wind and wave conditions during the turbine life is one of the reasons for the high costs of offshore wind farms. Although these uncertainties cannot be eliminated, it is crucial to know their effects on both the loads of the turbines and the power output. Since the power output is proportional to the third power of the wind speed, small differences in wind speed statistics can lead to substantial differences in the mean and variance of the power output, potentially making the difference between a profitable or unprofitable wind farm. Numerical simulation of a wind farm under different wind/sea conditions can be a valuable tool for decision making. However, because of the complexity of the system consisting of weather conditions and turbines, it is not feasible to perform simulations for all possible or observed input conditions.

It is therefore important to select in a careful way the input conditions for which the wind farm simulation model is going to be evaluated (we refer to these selected input conditions as training samples hereafter). Possible dependencies in the input conditions need to be taken into account for this selection. Predictions for the output for input conditions not included in the training samples can be made with the use of an emulator. An emulator gives an approximation of the simulation model output, but is computation-

ally cheaper to evaluate than the simulation model itself. These predictions can aid in performing sensitivity analysis.

The aim of this paper is to present a framework in which the analysis of the input data and the model output is combined, building on our earlier work in [1, 2]. We test this framework on a wind farm simulation based on the layout of Horns Rev, with wind/wave data obtained at the Meteomast IJmuiden serving as input. The simulation model that we use is not very advanced, however our goal is to demonstrate the proposed framework rather than to obtain highly realistic results, hence the limitations of the model are less important for the present study.

Section 2 describes the weather data, the wind farm and the simulation to obtain the power output and efficiency of the wind farm under the given weather conditions. Section 3 describe the methods in the framework, while Section 4 shows the results. The conclusion follows in Section 5.

## 2 SETUP

We briefly describe the data in Section 2.1, the wind farm that we simulate in Section 2.2 and the simulator (computational model) itself in 2.3.

### 2.1 Meteomast IJmuiden

Meteomast IJmuiden (MMIJ) is a meteorological mast in the North Sea, located approximately 75km west of the coast of IJmuiden (the Netherlands) [3]. We use data from a measurement campaign that has run from November 1st, 2011 up to and including March 9th, 2016. Atmospheric conditions (e.g., wind speed and direction, air pressure, air temperature) have been measured at several heights in and above the mast. The data has been post-processed and is publicly available online, see [3]. The list of measurement variables used in our analysis can be found in Appendix A.

### 2.2 Horns Rev I

Horns Rev I is an offshore wind farm in the Danish North Sea consisting of 80 Vestas V80 2MW turbines [4] in a oblique rectangular layout with a spacing of 560 meter [5]. The rotor diameter is 80 meters and the hub height is 70m. It is very well known in the wind energy community [6, 7, 8, 9], especially for its wind turbine wakes affecting nearby turbines. We will use the layout of the Horns Rev I wind farm for our computational model, described in the next section.

### 2.3 Wind farm simulation

We use the observation data from MMIJ as input for a computational model of Horns Rev I. The output of this model consists of power output and efficiency of the wind farm, given the wind speed and direction at various heights as input. The computer model is described in [10]. It makes use of a wake model from Bastankhah and Porté-Agel [11]. Furthermore, since the incoming wind speed can vary over the rotor-swept area due to wake effects, an equivalent incoming wind speed is computed as in [12].

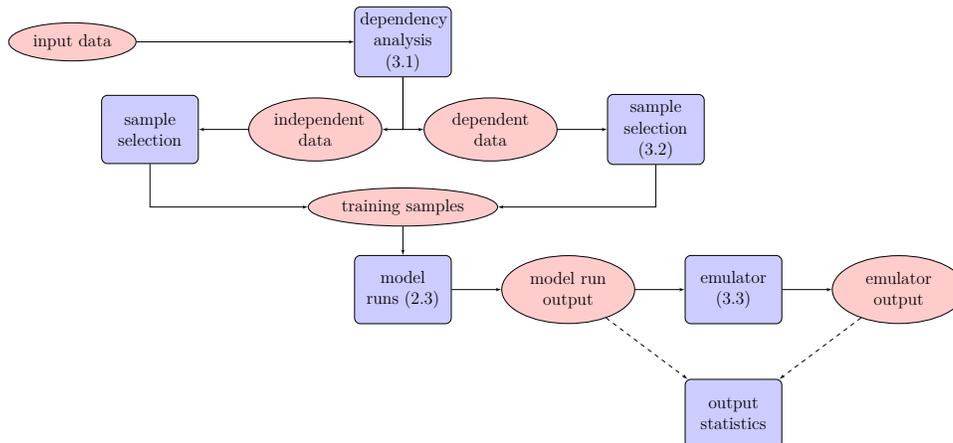


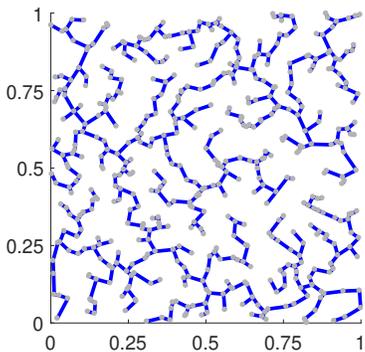
Figure 1: The proposed framework. The numbers in brackets indicate the corresponding section.

We point out that the computational model is not very advanced and has several limitations. However, it is useful in the present context as our objective is to test the coupling of several methods for uncertainty quantification and sensitivity analysis, rather than to obtain output for operational purposes. We mention two limitations here. First, the incoming wind speed in the model from [10] is constant over the height of the turbine. We made an adaptation such that the wind speed is interpolated linearly between measurements at different heights. Second, the turbulence intensity is fixed at 0.075.

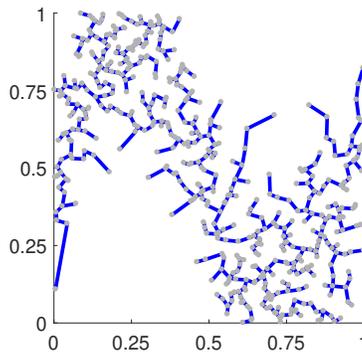
### 3 METHODS

Our objective in this study is to demonstrate how to combine two previously developed methods for uncertainty quantification (UQ) and dependency/sensitivity analysis (DA/SA) with dependent inputs [1, 2]. These methods are aimed at situations where the input distribution is unknown and where a dataset of the (multivariate, dependent) inputs is available instead. The dataset is assumed to be large (i.e., a large number of data points). Furthermore, the computational model is assumed to be expensive so that only a small number of model evaluations (or simulations) can be performed. Focal point is the presence of dependencies in the data. When no dependencies exist, i.e., all input variables are independent, a range of methods is available for performing UQ. However, in the presence of input dependencies, most of these methods do not apply. Although Monte Carlo sampling can still be used in this case, it requires a large number of model evaluations, making it often too costly in case of expensive computational models.

We combine these methods in the following manner, see also Figure 1. First, the input data is analyzed for dependencies, as described in Section 3.1. If the data is independent, then methods such as quasi Monte-Carlo [13] or stochastic collocation [14] can be used for the training sample selection. If the data is dependent, training samples must be selected in a different way, as explained in Section 3.2. These training samples serve as the input conditions for the computational model. Once the training samples are selected,



(a) MST on rank-transformed data from the bivariate uniform distribution.



(b) MST on rank-transformed data from an artificially constructed distribution.

Figure 2: An illustration of the idea behind the method for dependency analysis proposed in [2]. The dependent data in the panel on the right has a shorter MST length, indicating dependence between the two variables.

the wind farm simulations as described in Section 2.3 can be performed. The results of these simulations are analyzed in Section 4. The resulting simulation output serves as the basis for an emulator as described in Section 3.3. The goal of this emulator is to construct an approximation of the output for the input conditions that were not part of the training samples (thus, the computational model was not evaluated for these inputs). The output of this emulator is used to perform sensitivity analysis, as expressed in Section 3.4.

### 3.1 Dependency analysis

Since we anticipate that several input variables are strongly, but not linearly, dependent, we use the method of [2] for the dependency analysis. This method uses the length of a minimum spanning tree (MST) on the data to quantify dependencies. An illustration can be found in Figure 2. The data is rank-transformed, such that each marginal distribution is quasi-uniform. Each combination of two variables can be tested for dependency in the following way. When the data is dependent, the data points do not cover the complete unit square, resulting in a shorter MST length. The shorter the MST length, the stronger the dependency. In [2], the distribution of the MST length for independent data was assessed numerically. With this distribution, a statistical test can be constructed. If the length is shorter than the 0.01-quantile, the data is considered to be dependent.

The outcome of the dependency analysis is an ordering of all bivariate subsets of the input data from most strongly dependent to least dependent. This information can be visualized in a heatmap in the shape of a matrix, in which the item with row  $i$  and column  $j$  represents the dependency (quantified by the logarithm of the normalized MST length, see [2]) between input variables  $i$  and  $j$ . In this way, groups of dependent variables can be recognized easily. When all input variables turn out to be independent, more methods are available for the sample selection.

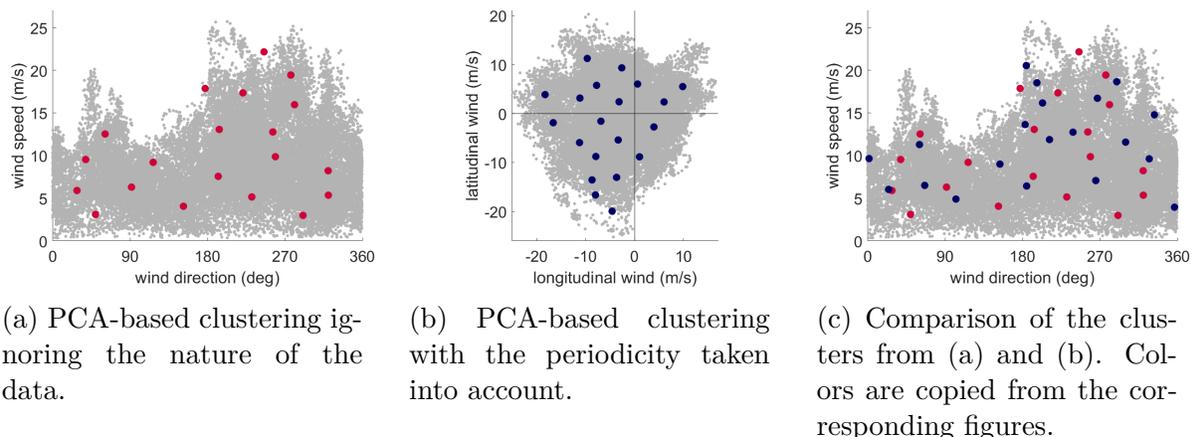


Figure 3: The effect of periodic variables. In Figure 3a, the periodicity of the wind direction is ignored, while it is taken into account in Figure 3b by converting the polar coordinates in (a) to a Cartesian grid. They are compared in Figure 3c.

### 3.2 Selecting representative cases

When the input data is dependent, the selection of training samples is more difficult. Since these training samples are the only input conditions for which the computational model is evaluated, it is important that the training samples are chosen to best represent the data. In [1], we proposed a method based on clustering analysis for this selection. Examples for 2D can be found in Figure 3. We see that the training samples are spread throughout the data, and no training samples appear in regions without data points.

The training samples are selected according to [1]. Both the  $k$ -means and the PCA-based clustering are used, with the adaptation that the data points after the PCA-based clustering method are reassigned to the nearest cluster center.

One point of interest here is the presence of periodic variables (e.g., wind directions). We give a small example with wind direction and wind speed at 27 meters height in Figure 3. Data points at  $1^\circ$  and  $359^\circ$  are far apart if the periodicity is ignored. However, when multiple variables are periodic, difficulties arise in the computation of distances if the periodicities are taken into account (since the topology is no longer equivalent to  $\mathbb{R}^p$ , with  $p$  the number of input variables). Therefore, we ignore this aspect of directional variables. (In Figure 3 the practical consequence is minor, as the clusters represent the shape of the dataset quite well, both with and without accounting for periodicity. )

### 3.3 Emulation

The training samples selected as described in the previous section are used as input for evaluations of the computational model described in Section 2.3. The output of the model evaluations is essential for performing sensitivity analysis.

However, the number of evaluations of the computational model (i.e., the number of training samples) is not large enough for detailed sensitivity analysis. Hence, we want to

be able to predict the output for other data points (outside the training sample) as well, while avoiding additional expensive simulator evaluations. This can be achieved by the use of an emulator. An emulator based on the obtained simulation results is constructed as a Gaussian process [15]. The advantage of Gaussian processes with respect to other types of emulators is the automatic inclusion of uncertainty in terms of variance.

Our implementation of Gaussian processes uses the Wendland kernel [16] optimized with respect to the Gaussian kernel. We use ordinary kriging with the mean and variance estimated from generalized least-squares and the compressed log-likelihood, respectively. The length scales are optimized within bounds from the log-likelihood with several restarts to avoid local maxima. The constructed Gaussian processes are tested on prediction suitability in Section 4.4.

### 3.4 Sensitivity analysis

The emulator constructed according to the previous section is helpful for performing sensitivity analysis (SA). In its most general form, sensitivity analysis concerns the analysis of output uncertainty by allocating it to the different uncertainties in the inputs. Seen from a different viewpoint, it concerns the dependence between input and output variables. Some input variables may have a negligible effect on the output, while others are more influential.

The sensitivity analysis is performed in a similar way as the dependency analysis described in Section 3.1. Instead of quantifying the dependency between combinations of input variables, we now quantify the dependency between each combination of an input and an output variable. The stronger the dependency between an input and output variable, the more this input variable drives the output. The dependencies found do not need to be causal as confounder effects may be contained in the data. A thorough understanding of the physics involved in the model is necessary to make a distinction between causal effects and confounder effects.

## 4 RESULTS

We will first describe the dataset of input conditions in more detail in Section 4.1. Then the dependency analysis is performed in Section 4.2 before continuing with the evaluations of the computational model. We present the results from the sample selection and the model evaluations in Section 4.3. In Section 4.4, we discuss the construction of the Gaussian process emulator. The output from the emulator is further used in Section 4.5 for sensitivity analysis.

### 4.1 Dataset

We constructed three datasets from the given MMIJ data as given in Appendix A. The first dataset contains only mast measurements (146686 instances in total) and includes quantities (e.g., air temperature) that are not used as input for the computational model. The second dataset is a subset of the first, restricted to only the input variables required for the computational model. The third dataset also contains wave variables, but is

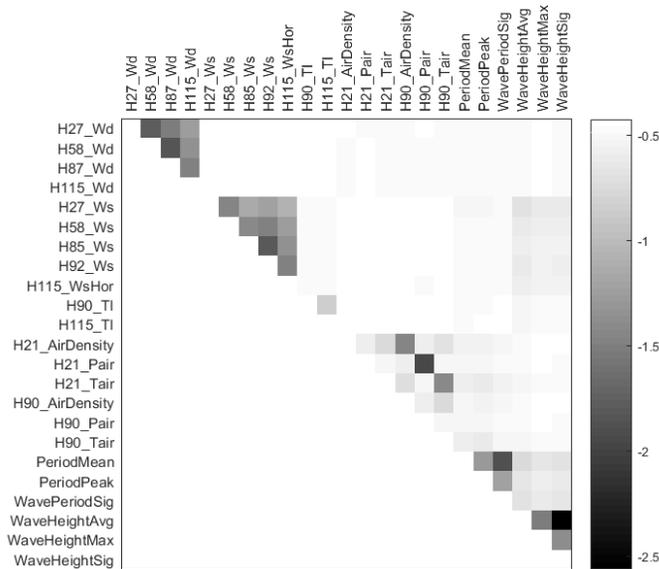


Figure 4: Computed values for the quantifier of dependence for dataset 3. Darker squares indicate higher dependence. Because of symmetry, only the upper triangle is shown.

smaller due to the reduced recording frequency (19353 instances).

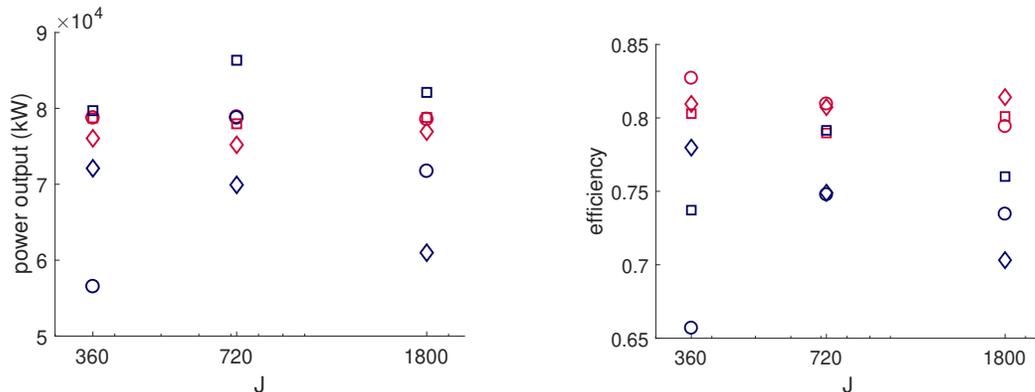
## 4.2 Dependency analysis

We quantify the dependency strength between observation variables from dataset 3, as discussed in Section 3.1. The estimated reference levels for the 0.05- and 0.01-quantile correspond to the logarithm of the normalized MST lengths, being  $-0.4237$  and  $-0.4258$ , respectively. The results are shown in Figure 4, where white indicates independence or minor dependence.

As expected, strong dependencies exist between similar variables at different heights, such as wind speed, direction, air density, pressure and temperature. For wind speed and wind direction, these are the darker triangles in the top-left corner of the figure. The two included turbulence intensities (TIs) are strongly dependent as well. Furthermore, strong dependencies can be seen to exist between air density, air pressure and temperature (lower-right near center, above diagonal). Two other groups of strongly dependent variables are the wave period and height variables (lower-right corner). Dependencies between them exist too. The last identifiable group of dependencies is between wind speeds and wave heights. Some dependencies are seen to exist between the air variables and the wave period, although we have no clear physical explanation for these.

## 4.3 Simulation of the wind farm

Because of the dependencies in the input data, we select samples using the methods described in Section 3.2. For each of the three datasets, we construct three sets of training samples with a varying number (denoted  $J$ ) of training samples: 360, 720 and 1800. The wind farm simulation ran for all the constructed training samples.



(a) Computed values for the power output. For  $J = 720$ , the circles overlap.

(b) Computed values for the efficiency.

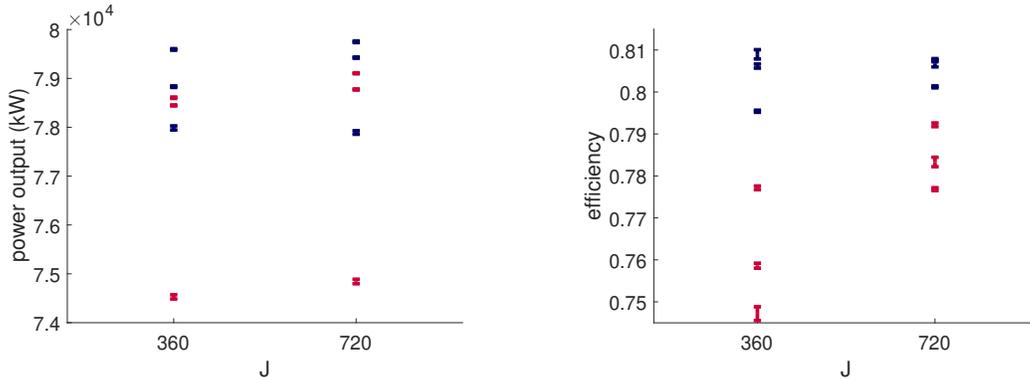
Figure 5: Results of the simulation. Red indicates PCA-based clustering, blue indicates  $k$ -means. Dataset 1 to 3 are indicated by circles, squares and diamonds, respectively.

The two simulation outputs of interest are the wind farm power output and efficiency (the latter being defined as power output relative to maximum power output in case of no wakes). For each training sample, the computational model returns one value for the power output and one for the efficiency. For each set of training samples, the weighted mean is computed for both output variables. The weights are computed as the fraction of data points associated to each training sample.

The results are presented in Figure 5. It can be seen that the estimated mean output varies with the different numbers of  $J$ , with the clustering method and with the dataset used. Note that the datasets were extracted from the same (larger) dataset and that the number of instances in datasets 1 and 2 are equal (see Section 4.1). This implies that differences in clusterings of these datasets are solely due to the different number of dimensions. It can also be seen that the  $k$ -means method gives smaller estimates than the PCA-based clustering.

For the PCA-based clustering, the average fraction of clusters for which the computational model gives nonzero (positive) power output is 0.585, while it is 0.839 for  $k$ -means. The sum of the weights for these clusters is 0.912 and 0.835, respectively. This implies that more clusters in PCA-based clustering have zero output, but the weight contained in them is lower, compared to the  $k$ -means clustering.

When comparing the results, we see that PCA-based clustering on dataset 2 gives the most consistent results for different  $J$ , has no outliers and is close to other estimates. This can be seen by the red squares which always overlap with other markers. Furthermore, these results correspond to average values around  $8 \cdot 10^4$  kW power output and around 0.8 for the efficiency.



(a) Computed values for the power output based on the emulator.

(b) Computed values for the efficiency based on the emulator.

Figure 6: Results of the simulation, indicated as errorbars. Red indicates PCA-based clustering, blue indicates  $k$ -means. Because of the similarity in the results, we do not indicate the method. For some results, the errorbars appear as rectangles because of the small standard deviation.

#### 4.4 Emulation of the wind farm simulation

The emulator as described in Section 3.3 is constructed for  $J = 360$  and  $J = 720$ , for both PCA-based clustering and  $k$ -means, and for all three datasets. Altogether, there are 12 different sets of training samples, and hence, 12 different emulators. With each emulator, we compute predictions (means and variances) of the power output and efficiency for all points (input conditions) in the dataset associated with the emulator. We note that there are 4 emulators associated with each dataset, constructed with different combinations of  $J$  (360, 720) and of the clustering method (PCA,  $k$ -means).

Because of the Gaussian properties of each simulator, it is straightforward to compute the distribution of the overall mean output  $M = \frac{1}{N} \sum_{i=1}^N m(\mathbf{x}_i)$ , in which  $N$  is the number of data points,  $\mathbf{x}_i$  the  $i$ th data point and  $m(\cdot)$  the emulator mean.

The results are shown in Figure 6. Errorbars are given for the estimated mean plus and minus one estimated standard deviation. They do not need to overlap because they are means and standard deviations for different emulators (although the emulators model the same system). The results for the power output are consistent with the results in Figure 5 and have both a small spread and a small standard deviation. For the efficiency, the PCA-based clustering has values smaller than the  $k$ -means clustering, and the values are also smaller than the previous value found (which was around 0.8). We do not have a clear explanation for this at the moment.

We note that because of their Gaussian nature, the emulators may occasionally predict a negative mean value of the power output. This is the case for 2.3% of the data points, averaged over the different emulators. As this is only a small fraction, we do not attempt to fix it here but leave it to future study to resolve this.

Finally, we examine the quality of the Gaussian process. The quality depends on

robustness, accuracy and prediction error. Robustness concerns the stability of hyper parameters, in our case being the length scales. Accuracy concerns the distance between predicted and observed values, which can be addressed by cross-validation. We focus here on prediction error by assessing the average kriging error ( $AKE$ ). A Gaussian process is considered to be a good fit if the  $AKE < 0.05$ . In our case, for the power output, most values for the  $AKE$  are between 0.10 and 0.21, with two out of twelve around 0.27. For the efficiency, the  $AKE$  varies between 0.23 and 0.64. These values may be improved if the nonimportant input variables are left out, or if  $J$  (the number of training samples) is increased for the construction of the emulator.

#### 4.5 Sensitivity analysis

For dataset 3, the emulator output is used to compute dependencies between the input variables and the output variables. We choose to keep all the variables in the sensitivity analysis, including the nonimportant input variables. We note that the inclusion of these nonimportant inputs might be the cause of the large value of the  $AKE$ .

The results are given in Figure 7. The most influential input variables for both power output and efficiency are wind speeds around hub height, followed by the other wind speeds and the wave height variables. This is expected, since the largest part of the rotor area is around hub height and the wind speeds at different heights are strongly dependent. Furthermore, the wind speeds act as confounders for the wave height variables. Finally, we see that the power output and the efficiency are mutually dependent as well.

Altogether, the results from the different emulators are consistent in their assessment of the dependencies and their strengths.

## 5 CONCLUSION

A framework for the combination of dependency analysis, sample selection and sensitivity analysis has been described which can be used for systems that have dependent input variables. It is aimed at situations where the input distribution is unknown and only a dataset of the input variables is available. The framework takes the dependencies into account in a natural way.

The framework has been tested on the combination of a dataset containing weather data and a computational model of power output for the Horns Rev wind farm. The framework correctly identifies dependencies in the data. We demonstrated the construction of various emulators, and we performed sensitivity analysis (SA) using these emulators. The SA results were consistent over the different emulators considered.

## ACKNOWLEDGEMENTS

This research is part of the EUROS programme, which is supported by NWO domain Applied and Engineering Sciences and partly funded by the Ministry of Economic Affairs. We thank the Wind Energy Section of TU Delft for providing us the code from [10].

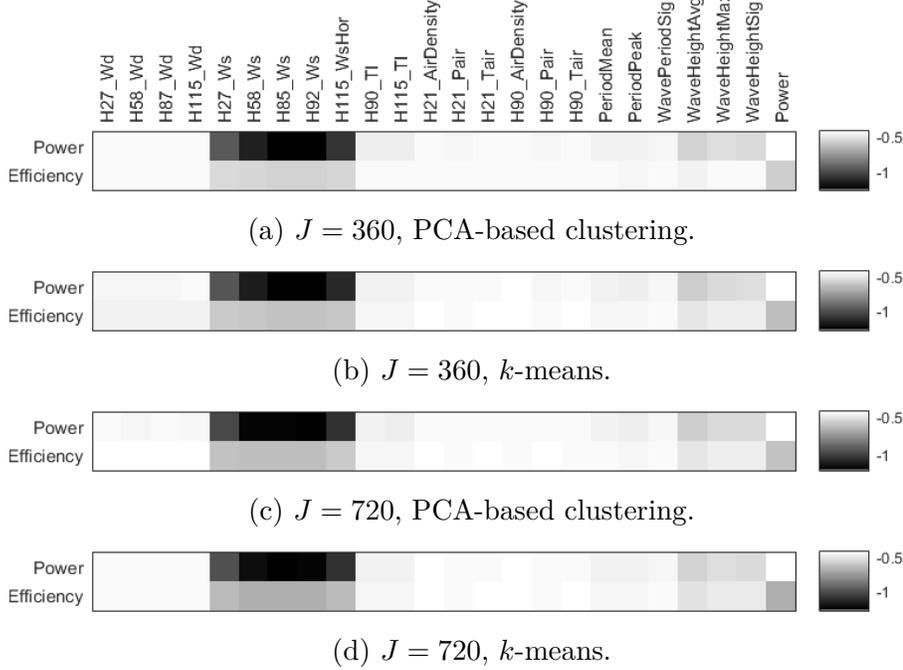


Figure 7: Results of the sensitivity analysis. The dependencies between combinations of input and output variables are quantified, as discussed in Section 3. Darker colors indicate stronger dependence (see also Figure 4).

## A VARIABLES

The variables of the three datasets are given in Table 1. The names contain (if applicable) the height at which the variables are recorded (H), the quantity measured (Wd for wind direction, Ws for wind speed, etc.), and in some cases additional information (e.g. signal quality). See [3] for more details.

Table 1: Variables of the MMIJ included in the analysis, see also [3].

Variables in sets 1,2,3	Variables in sets 1,3	Variables in set 3
MMIJ_H27_Wd_Q1_avg	MMIJ_ZPHS_H90_TI	MMIJ_BW_PeriodMean
MMIJ_H58_Wd_Q1_avg	MMIJ_ZPHS_H115_TI	MMIJ_BW_PeriodPeak
MMIJ_H87_Wd_Q1_avg	MMIJ_H21_AirDensity_Q1_avg	MMIJ_BW_WavePeriodSig
MMIJ_ZPHS_H115_Wd	MMIJ_H21_Pair_Q1_avg	MMIJ_BW_WaveHeightAvg
MMIJ_H27_Ws_Q1_avg	MMIJ_H21_Tair_Q1_avg	MMIJ_BW_WaveHeightMax
MMIJ_H58_Ws_Q1_avg	MMIJ_H90_AirDensity_Q1_avg	MMIJ_BW_WaveHeightSig
MMIJ_H85_Ws_Q1_avg	MMIJ_H90_Pair_Q1_avg	
MMIJ_H92_Ws_Q1_avg	MMIJ_H90_Tair_Q1_avg	
MMIJ_ZPHS_H115_WsHor_avg		

## REFERENCES

- [1] A. W. Eggels et al., “Clustering-based collocation for uncertainty propagation with multivariate dependent inputs,” *Int. J. Uncertainty Quantification*, vol. 8, no. 1,

- 2018.
- [2] A. W. Eggels and D. T. Crommelin, “Quantifying dependencies for sensitivity analysis with multivariate input sample data,” *arXiv preprint arXiv:1802.01841*, 2018.
  - [3] “Meteomast IJmuiden.” <http://www.meteomastijmuiden.nl/en/home/>. Accessed: 2018-02-09.
  - [4] “Vestas V80-2.0.” <https://www.en.wind-turbine-models.com/turbines/19-vestas-v80-2.0>. Accessed: 2018-02-09.
  - [5] “Horns Rev 1.” <https://powerplants.vattenfall.com/en/horns-rev>. Accessed: 2018-02-09.
  - [6] M. Méchali et al., “Wake effects at Horns Rev and their influence on energy production,” in *European wind energy conference*, vol. 1, pp. 10–20, 2006.
  - [7] R. J. Barthelmie et al., “Modelling the impact of wakes on power output at Nysted and Horns Rev,” in *European Wind Energy Conference*, 2009.
  - [8] K. S. Hansen et al., “The impact of turbulence intensity and atmospheric stability on power deficits due to wind turbine wakes at Horns Rev wind farm,” *Wind Energy*, vol. 15, no. 1, p. 183–196, 2011.
  - [9] M. Gaumond et al., “Evaluation of the wind direction uncertainty and its impact on wake modeling at the Horns Rev offshore wind farm,” *Wind Energy*, vol. 17, no. 8, p. 1169–1178, 2013.
  - [10] B. Hu, “Design of a Simple Wake Model for the Wind Farm Layout Optimization Considering the Wake Meandering Effect,” Master’s thesis, TU Delft, 2016.
  - [11] M. Bastankhah and F. Porté-Agel, “A new analytical model for wind-turbine wakes,” *Renewable Energy*, vol. 70, pp. 116–123, 2014.
  - [12] J. Choi and M. Shan, “Advancement of Jensen (Park) wake model,” in *Proceedings of the European Wind Energy Conference and Exhibition*, pp. 1–8, 2013.
  - [13] J. Dick, F. Y. Kuo, and I. H. Sloan, “High-dimensional integration: the quasi-Monte Carlo way,” *Acta Numerica*, vol. 22, pp. 133–288, 2013.
  - [14] L. Mathelin, M. Y. Hussaini, and T. A. Zang, “A stochastic collocation algorithm for uncertainty analysis,” Tech. Rep. NASA/CR-2003-212153, NASA, 2003.
  - [15] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. MIT press Cambridge, 2006.
  - [16] H. Wendland, “Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree,” *Adv. Comput. Math.*, vol. 4, no. 1, pp. 389–396, 1995.