Wind turbine site-specific load estimation using artificial neural networks calibrated by means of high-fidelity load simulations

Schröder, Laura; Dimitrov, Nikolay Krasimirov; Verelst, David Robert; Sørensen, John Aasted

Published in:
Journal of Physics: Conference Series

Link to article, DOI:
10.1088/1742-6596/1037/6/062027

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

Wind turbine site-specific load estimation using artificial neural networks calibrated by means of high-fidelity load simulations

IOP Conf. Series: Journal of Physics: Conf. Series **1037** (2018) 062027 doi:10.1088/1742-6596/1037/6/062027

Wind turbine site-specific load estimation using artificial neural networks calibrated by means of high-fidelity load simulations

Laura Schröder[1], Nikolay Krasimirov Dimitrov[1], David Robert Verelst[1], John Aasted Sørensen[2]

[1]DTU Wind Energy, DTU, Risø Campus, Frederiksborgvej 399, 4000 Roskilde, DK
[2]EIT, DTU Diplom, Lautrupvang 15, 2750 Ballerup, DK

E-mail: lausc@dtu.dk

Abstract. Previous studies have suggested the use of reduced-order models calibrated by means of high-fidelity load simulations as means for computationally inexpensive wind turbine load assessments; the so far best performing surrogate modelling approach in terms of balance between accuracy and computational cost has been the polynomial chaos expansion (PCE). Regarding the growing interest in advanced machine learning applications, the potential of using Artificial Neural-Network (ANN) based surrogate models for improved simplified load assessment is investigated in this study. Different ANN model architectures have been evaluated and compared to other types of surrogate models (PCE and quadratic response surface). The results show that a feedforward neural network with two hidden layers and 11 neurons per layer, trained with the Levenberg Marquardt backpropagation algorithm is able to estimate blade root flapwise damage-equivalent loads (DEL) more accurately and faster than a PCE trained on the same data set. Further research will focus on further model improvements by applying different training techniques, as well as expanding the work with more load components.

1. Introduction

Typically wind turbines are designed for specific wind conditions which are specified in site classes by the IEC standards. When a turbine is placed at locations where a site-specific parameter exceeds these design conditions, site-specific load assessments including simulations over the whole design load base have to be carried out. As this procedure can become computationally expensive, several methods and procedures have been developed for simplifying load assessments based on statistical moments, multivariate regression models [1] and expansions using orthogonal polynomial basis [2]. Previous investigations comparing different surrogate models such as polynomial chaos expansion (PCE), universal kriging with polynomial chaos basis function and quadratic response surface, have shown that the PCE results in the best overall performance for the load estimation in terms of robustness, accuracy and computing time [3].

Regarding the growing interest in advanced machine learning applications, the purpose of this study is to evaluate the potential of using models based on Artificial Neural Networks (ANNs) as a flexible and potentially better-performance alternative to the previously mentioned surrogate models that have been developed already. Therefore, different ANN models are trained for

estimating the loads of a wind turbine using a database with high-fidelity load simulations and compared to the load estimations obtained from the aforementioned surrogate model approaches. The previously developed surrogate models serving as a reference in this study are the PCE and a rather simple quadratic response surface approach.

2. Data description

For training the surrogate models a database consisting of high-fidelity load simulations is used based on the setup described in [3]. Six probabilistic parameters characterizing the wind field (wind speed, wind field variance, vertical wind shear exponent, wind veer and turbulence length scale and anisotropy factor defined by the Mann model [4]) are used as inputs for the models. A 10000-point pseudo Monte Carlo sample is drawn from the joint distribution and pre-defined ranges of the input variables (see Figure 1) [3]. All parameters, except the wind speed u, are uniformly distributed. Since most parameters have wide ranges of variation at small u, the wind speed is Beta-distributed in order to improve point spacing in the set used for training the ANN model. However, this would not influence a load estimation using the resulting model, as in this case the sampling can be done according to any target distribution of environmental conditions, for example a site-specific Weibull-distributed wind speed.

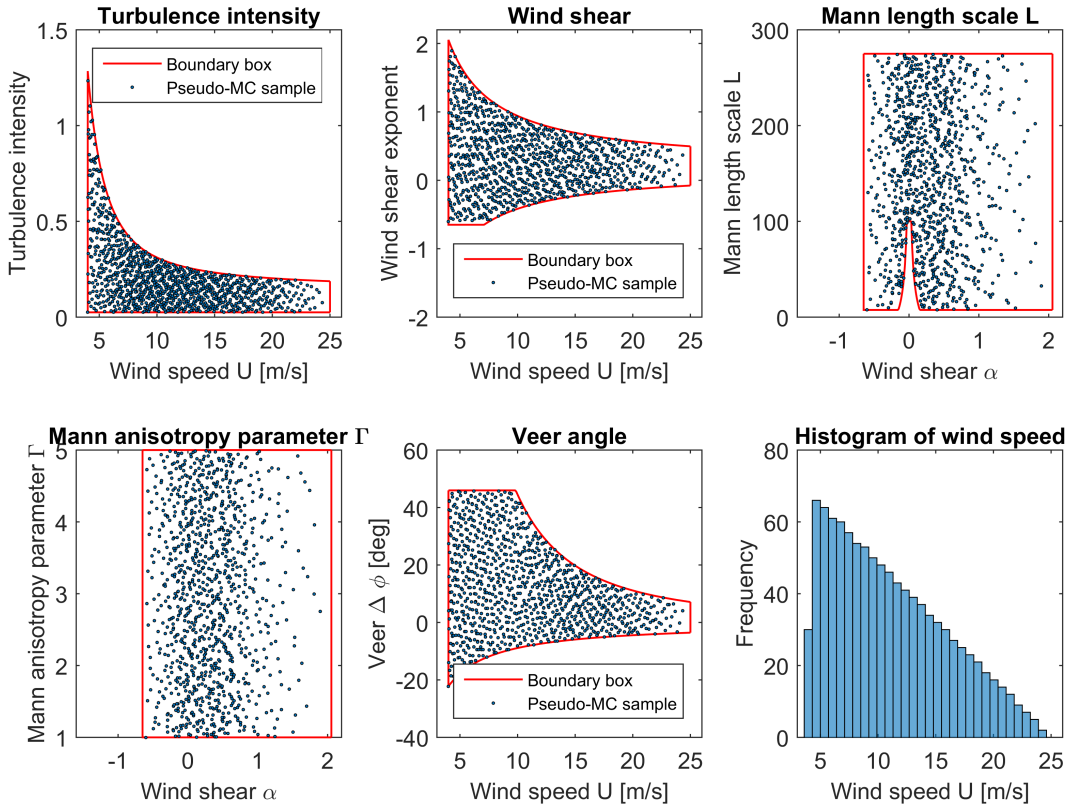

Figure 1. Distributions of input variables: The wind speed is considered Beta-distributed while the remaining input variables are uniformly distributed. The figure is reproduced from [3] with the author's permission.

Aeroelastic load simulations are carried out using the HAWC2 simulation tool with the reference turbine DTU 10MW RWT to obtain the damage-equivalent loads (DEL) which serve

as target for training the surrogate models (see Figure 2). This study focuses on estimating the lifetime DEL exemplarily for the blade root flapwise bending moment.

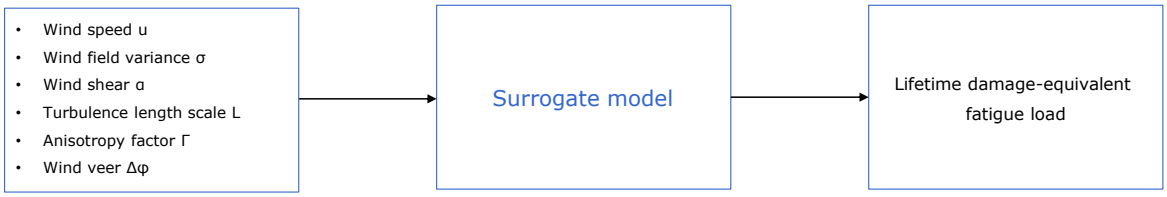

Figure 2. Schematic illustration of site-specific load estimation using surrogate model.

3. Methods

In the following section the different approaches are briefly described and the choice for the specific model setup is explained. Within each approach a 10-fold cross-validation is applied in order to estimate the generalization error of the corresponding load estimations.

3.1. Neural Network approach

Originally, ANNs were invented as mathematical models of the information processing by neurons [5]. There are clear differences between ANNs and biological neurons, nevertheless the basic structure exhibits some similarity. ANNs consist of a net of various connected information processing units, so-called artificial neurons. The neurons are organized in layers so that the information is processed sequentially in the network (see Figure 3).

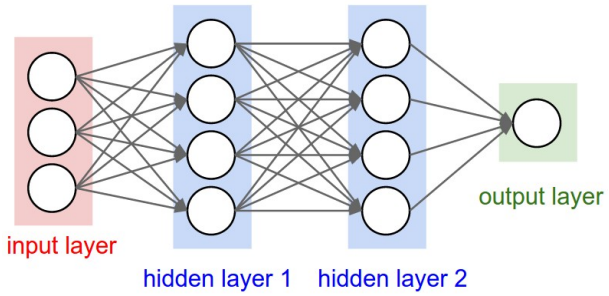

Figure 3. A 3-layer neural network with three inputs, two hidden layers consisting of four neurons each and one output layer [6].

An ANN with a linear transfer function can be written in the form of a linear regression model to predict the output [7]:

$$f(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{M} x_i w_i + w_0 \tag{1}$$

The input vector $\boldsymbol{x} = (x_1, ..., x_M)$ containing M attributes is fed into the input layer where neuron i is given an activation equal to x_i. The parameter w_i is referred to as weights and w_0 as biases. The activation is propagated through the hidden layers and finally to the output layer. The number of neurons in the output layer corresponds to the dimension of the output vector that is aimed to be predicted. This procedure is called forward pass which is used for the

simplest form of ANNs, the so-called feedforward networks.

With

$$\tilde{\boldsymbol{x}} = [1\ x_1\ x_2\ ...\ x_M]^T \tag{2}$$

we get the more condensed form

$$f(\boldsymbol{x}, \boldsymbol{w}) = \tilde{\boldsymbol{x}}^T \boldsymbol{w} \tag{3}$$

Having a nonlinear activation h and a number of H hidden neurons, the activation of the kth output neuron can simply be written as

$$f_k(\boldsymbol{x}, \boldsymbol{w}) = h^{(2)}(\sum_{j=1}^{H} W_{kj}^{(2)} h^{(1)}(\tilde{\boldsymbol{x}}^T \boldsymbol{w}_j^{(1)})) \tag{4}$$

with the weight matrix $W_{kj}^{(2)}$ containing weight terms for the jth hidden neuron and kth output neuron. Different functions can be used as the activation function h such as for instance hyperbolic tangent, logistic sigmoid or rectified linear unit.

In this work a feedforward neural network (FNN) is used. Inside the 10-fold cross-validation the data are divided into a 70 % training, a 20 % validation and a 10 % test set for each iteration. Since the sample set used in this study is considerably big, the data division is less crucial for the model performance. Another approach for splitting the data would be to follow the scaling law for the validation-set training-set size ratio as described in [8]. After dividing the data, the model is trained using back-propagation with the Levenberg-Marquardt algorithm [9] based on the training data set, and using the validation set to prevent overfitting. Finally, the performance of the FNN is tested on the test set.

For selecting the optimal network architecture an inner loop is added to the 10-fold cross-validation. Inside the loop networks with one to two hidden layers with one to twenty neurons in each layer respectively are trained and the generalization error is estimated for each model in the outer cross-validation loop. The final optimal network architecture is selected based on achieving a balance between high accuracy (achieved with a high number of neurons) and low computational time (low number of neurons) for the DEL estimation.

3.2. Polynomial chaos expansion approach

The polynomial chaos expansion, introduced by Norbert Wiener [10] and further developed into computational tools for stochastic modeling such as in [11] and [12] is a method to approximate a stochastic function $S(\boldsymbol{X})$ of multiple random variables $\boldsymbol{X} = [X_1, X_2, ..., X_M]$ using an orthogonal basis ϕ_j [3]:

$$S = \sum_{j=0}^{\infty} S_j \phi_j(X_1, ..., X_M) \tag{5}$$

where ϕ_j is a Hilbertian basis of the Hilbertian space containing the response [3]. Since the input random variables in this study are non-Gaussian distributed a generalized PCE [13] using Legendre polynomials is applied. The stochastic function $S = g(\boldsymbol{X})$ can be expressed as a truncated sequence $\tilde{S}(\boldsymbol{X}) + \epsilon$ with the zero-mean residual ϵ. A least-squares regression approach is then used for determining the coefficients of $\boldsymbol{S}$ which requires setting up a design of experiments with a so-called design matrix $\boldsymbol{\Psi}$ [3]. Assuming that the residuals are approximately normally distributed the closed-form solution is

$$(\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{-1} \cdot \boldsymbol{\Psi}^T \cdot \boldsymbol{y} \tag{6}$$

with $\boldsymbol{y}$ being a vector with the realizations obtained from the design experiment [3].

Before calibrating the PCE, the data samples are divided into a 90 % training and a 10 % test set in order to use the same amount of data for training the model as in the ANN methodology. The design matrix $\boldsymbol{\Psi}$ is calculated and the PCE is calibrated using the training set. The obtained coefficients of the PCE are then used for load predictions using the input variables from the test samples.

3.3. Response surface approach

Another reduced-order model is the quadratic-polynomial response surface (RS) which has found its application also for wind turbine load predictions [1]. In this method a quadratic polynomial regression is fit to the design points from the Monte Carlo simulations. A single response surface is not able to fully represent the turbine response with respect to the mean wind speed due to the low-order of the response surface [3]. Therefore, multiple response surfaces are calibrated for wind speeds from 4 m/s to 20 m/s in 1 m/s steps. Using the closed-form solution from the least-squares regression noted in Equation 6 the polynomial coefficients of the response surface are calculated.

As for the PCE the data samples are divided into a 90 % training and 10 % test set. For each wind speed bin of the training samples the design matrix $\boldsymbol{\Psi}$ is created and the corresponding coefficients of the quadratic regression function are calculated. The RS model is then validated using these coefficients on the test set.

4. Results

4.1. Performance evaluation: Neural Network

In the following section the results of the DEL prediction of the blade root flapwise bending moment using an ANN are presented.

As mentioned before, the optimum network architecture is selected based on achieving a high accuracy at low computational time. Figure 4 to Figure 7 illustrate the coefficient of determination and the test evaluation time of the load estimations using an ANN with one hidden layer (left side) and two hidden layers (right side) as a function of the number of hidden neurons per layer. The red dashed lines refer to the finally chosen architectures.

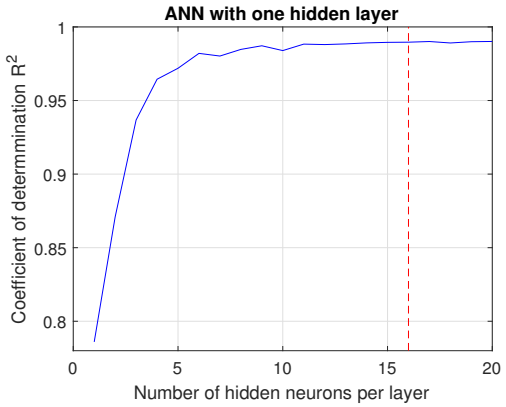

Figure 4. R^2 of load estimations using ANN with one hidden layer as a function of number of neurons per layer.

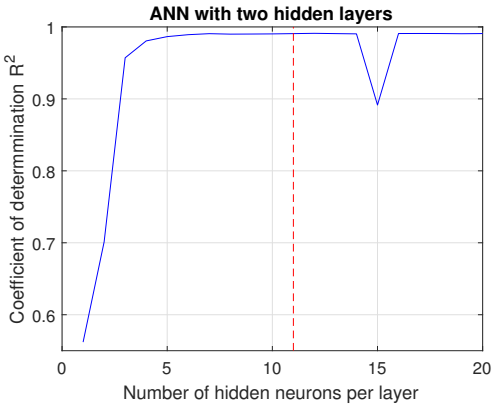

Figure 5. R^2 of load estimations using ANN with two hidden layers as a function of number of neurons per layer.

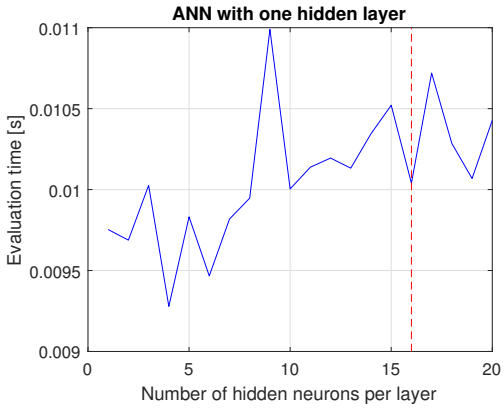

Figure 6. Test evaluation time of load estimations using ANN with one hidden layer as a function of number of neurons per layer.

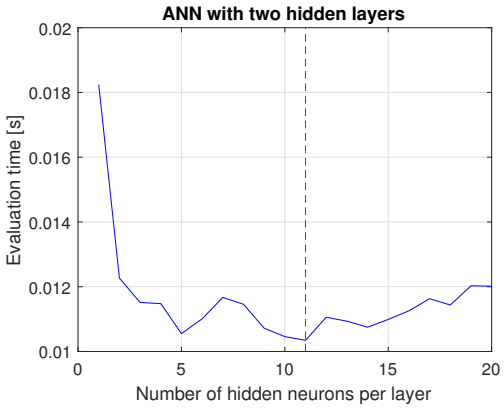

Figure 7. Test evaluation time of load estimations using ANN with two hidden layers as a function of number of neurons per layer.

A 1-layer ANN with 16 neurons per layer and a 2-layer ANN with 11 neurons per layer are selected as most suitable network architectures in this study. The performance of the two networks is presented separately for the training, validation and testing set in Table 1.

Table 1. Performance evaluation of ANN for training, validation and testing.

Criteria	ANN [16]	ANN [11 11]
Number of coefficients	129	222
Training evaluation time [s]	2.462	3.520
Testing evaluation time [s]	0.010	0.010
Training R^2	0.991	0.992
Validation R^2	0.990	0.991
Testing R^2	0.990	0.991

The test error distribution of the load estimations is illustrated for both networks in Figure 8 and Figure 9. The ANN with one layer predicts the DEL on the test set with a mean error of $\mu = 36.17\,\text{Nm}$ and a standard deviation of $\sigma = 633.21\,\text{Nm}$. The ANN with two layers predicts the DEL on the test set with a mean error of $\mu = 33.12\,\text{Nm}$ and a standard deviation of $\sigma = 609.98\,\text{Nm}$.

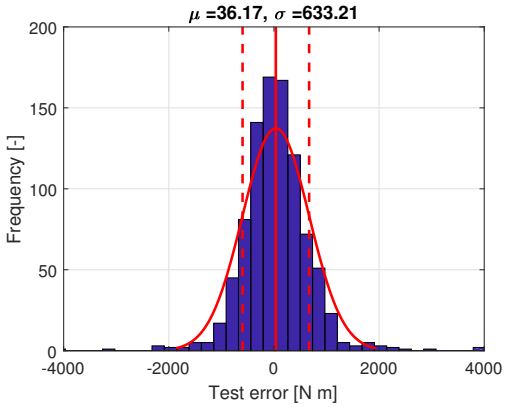

Figure 8. Error distribution of load estimation for testing using an ANN with one hidden layer with 16 hidden neurons.

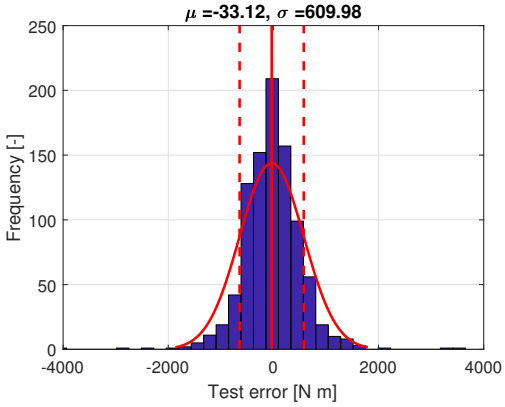

Figure 9. Error distribution of load estimation for testing using an ANN with two hidden layers with 11 neurons per layer.

4.2. Comparison with PCE and Response surface

The performance of the load estimations using neural networks is compared to the load estimations obtained from a PCE and a truncated PCE which retains 99.5 % of the model variance and a quadratic RS in Table 2.

Table 2. Performance comparison of load estimation using RS, PCE and ANN.

Criteria	PCE(1*)	PCE(0.995*)	RS	ANN [16]	ANN [11 11]
Number of coefficients	924	54	336**	129	222
Training evaluation time [s]	179.365	9.765	0.021	2.462	3.520
Testing evaluation time [s]	17.778	1.051	0.008	0.010	0.010
Training R^2	0.993	0.988	0.974	0.991	0.992
Validation R^2	-	-	-	0.990	0.991
Testing R^2	0.989	0.988	0.972	0.990	0.991

* Proportion of model variance retained
** 21 coefficients per wind speed bin for 4 m/s to 20 m/s

The scatter plots of the estimated blade root bending moment against the simulated loads from the high-fidelity database are shown in Figure 10 to Figure 12 for the truncated PCE, the quadratic response surface and the ANN with two hidden layers.

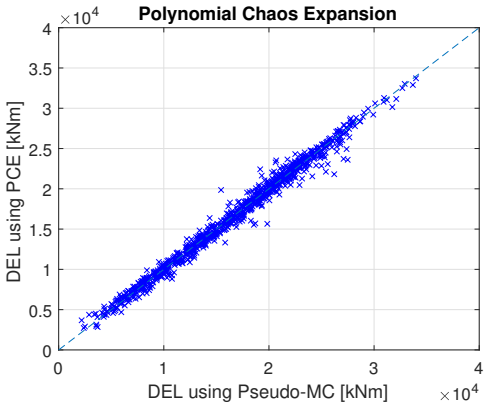

Figure 10. Scatter plot of blade root flapwise bending moment estimated by PCE against database.

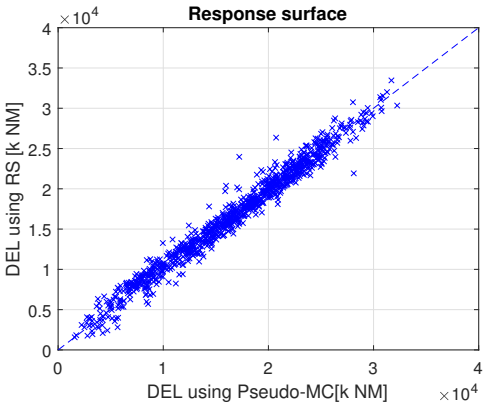

Figure 11. Scatter plot of blade root flapwise bending moment estimated by RS against database.

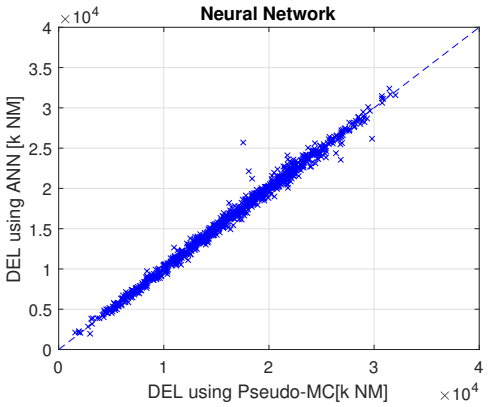

Figure 12. Scatter plot of blade root flapwise bending moment estimated by ANN against database.

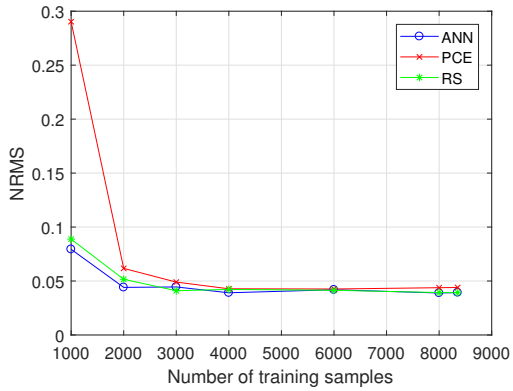

Figure 13. Convergence of ANN, PCE and response surface with respect to the number of samples used for training the model.

In order to assess the convergence of the surrogate models the normalized root-mean-square (NRMS) error between the simulated loads from the high-fidelity database and the surrogate model predictions is plotted as a function of the number of samples used for training the surrogate model (see Figure 13).

The NRMS is calculated using Equation 7 with the set of observations (i.e. DEL from high-fidelity simulation base) $\boldsymbol{y} = g(\boldsymbol{X}^{(i)}), i = 1...N$ and the prediction set from the surrogate models $\tilde{y_i} = \tilde{S}(\boldsymbol{X}^{(i)}), i = 1...N$, over the same set of N sample points $\boldsymbol{X}^{(i)}$:

$$\varepsilon_{NRMS} = \frac{1}{E[\boldsymbol{y}]} \sqrt{\frac{\sum_{i=1}^{N} (\tilde{y_i} - y_i)^2}{N}} \tag{7}$$

with the expected value of the observation variable $E[\boldsymbol{y}]$.

5. Discussion

For selecting the optimal network architecture the model performance of a 1-layer and 2-layer ANN is analysed in dependency of the number of hidden neurons in Section 4.1. While the accuracy of the load predictions increases with the number of hidden neurons (see Figure 4 and Figure 5), it was expected that the test evaluation time would increase as well. However, Figure 7 shows an unexpected high value in the test evaluation time for a 2-layer ANN with one hidden neuron per layer. This might be caused by an overhead effect during the first iteration of the program.

The load prediction performances of the ANN with one hidden layer and 16 hidden neurons and the ANN with two hidden layers and 11 neurons per layer are compared in Section 4.1. Both networks are able to give fast and accurate load estimations. As it can be seen in Figure 8 and Figure 9, the ANN with two layers predicts the DELs with a smaller mean and standard deviation of the test error while performing with the same test evaluation time. This is also represented in the higher coefficient of determination of $R^2 = 0.991$.

The time needed for training the 2-layer ANN is about 1.06 s longer. However, the surrogate model only has to be trained once using the high-fidelity data base from the pseudo-Monte Carlo simulations and can then be applied to specific sites using the site-specific wind measurements. Therefore, the test evaluation time is a more significant criteria of the model performance.

The comparison between the three surrogate models in Section 4.2 shows that the fastest site-specific load assessment can be achieved using a quadratic RS, however at the cost of the model accuracy. The so far best performing PCE results in a smaller prediction error with a coefficient of determination of $R^2 = 0.988$ and a test evaluation time of 1.051 s. Comparing these results with the 2-layer ANN shows that the ANN performs approximately 100 times faster than the PCE with an even slightly smaller prediction error.

Evaluating the convergence of the three approaches in Figure 13 has shown that, all three models converge for approximately the same number of training samples (4000). However, it can be seen that the NRMS error of the load predictions using the PCE is significantly higher when using a small training sample set.

It should be noted that the approach presented in this study is only valid for site-specific load assessments for single turbines. Further research is needed to extend the surrogate model for wind farms where the wake effect plays an important role for the load predictions. A method for representing wake-induced loads with surrogate models is currently being considered in [14].

6. Conclusions and future work

This study has presented a neural network approach for predicting the blade root bending damage-equivalent moments for site-specific load assessments. The optimal network architecture achieving low prediction errors with a small test evaluation time is an ANN with two hidden layers and 11 neurons per layer. The comparison with the PCE model shows that the ANN performs better with a slightly lower performance error and faster evaluation time. Compared to the fast but less accurate quadratic RS it performs with significantly smaller prediction error at a slightly higher test evaluation time.

All in all, the results of this study confirm the hypothesis that ANNs are a flexible and better-performance alternative to the other types of surrogate models tested in [3]. The investigations show that the ANN model is robust and sufficiently accurate in predicting the DEL of the blade root flapwise bending moment. The sensitivity study of the NRMS error of the load predictions has shown that the ANN performs better than the PCE even when only a small sampling set is available for training the surrogate model.

Further research should focus on assessing the prediction performance of the ANN using small sample sets. Additionally, in order to generalize the findings of this study future work should focus on evaluating the model performance of load estimations on other turbine

types/ratings and components as well as extending the surrogate model to enable wake-induced load predictions in a wind farm.

References

[1] Toft H S, Svenningsen L, Moser W, Sørensen J D and Thøgersen M L 2016 Assessment of wind turbine structural integrity using response surface methodology *Eng. Struct.* **106** 471–83

[2] Murcia J P, Réthoré P E, Dimitrov N, Natarajan A Sørensen J D, Graf P and Kim, T 2018 Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates *Ren. Ene.* **119** 910–22

[3] Dimitrov N, Kelly M, Vignaroli A and Berg J 2018 From wind to loads: wind turbine site-specific load estimation using databases with high-fidelity load simulations *Wind Energ. Sci. Discuss.* **2018** 1–39 URL `https://www.wind-energ-sci-discuss.net/wes-2018-18/`

[4] Mann J 1994 The spatial structure of neutral atmospheric surface-layer turbulence *J. Fluid. Mech.* **273** 141–68

[5] McCulloch W S and Pitts W 1943 A logical calculus of the ideas immanent in nervous activity *Bull. Math. Biophys.* **5** 115–33

[6] CS231n: Convolutional Neural Networks for Visual Recognition `http://cs231n.github.io/neural-networks-1/` accessed: 2018-10-03

[7] Friedman J, Hastie T, and Tibshirani, R 2001 *The elements of statistical learning* vol 1 (Springer series in statistics New York)

[8] Guyon I 1997 A Scaling Law for the Validation-Set Training-Set Size Ratio

[9] Yu H and Wilamowski B 2011 Levenberg-Marquardt Training *Industrial Electronics Handbook* vol 5 (CRC Press) pp 12-1 to 12-15 2nd ed

[10] Wiener N, 1938 The homogeneous chaos *Amer. J. Math.* **60** 897–936

[11] Ghanem R G and Spanos P D 1991 Stochastic Finite Element Method: Response Statistics *Stochastic finite elements: a spectral approach* (Springer) pp 101-19

[12] Karniadakis G E, Su C-H, Xiu D, Lucor D, Schwab C, and Todor R A 2005 Generalized polynomial chaos solution for differential equations with random inputs (Eidgenössische Technische Hochschule [ETH] Zürich. Seminar für Angewandte Mathematik)

[13] Xiu D, and Karniadakis, G E 2002 The Wiener–Askey polynomial chaos for stochastic differential equations *SIAM J. Sci. Comput.* **24** 619–44

[14] Dimitrov N 2018 Surrogate models for parameterized representation of wake-induced loads in wind farms *Wind Energy, under review*