Uncertainty propagation and sensitivity analysis of an artificial neural network used as wind turbine load surrogate model

Schröder, Laura; Dimitrov, Nikolay Krasimirov; Sørensen, John Aasted

Published in:
Journal of Physics: Conference Series

Link to article, DOI:
10.1088/1742-6596/1618/4/042040

Publication date:
2020

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

Link back to DTU Orbit

Citation (APA):
Uncertainty propagation and sensitivity analysis of an artificial neural network used as wind turbine load surrogate model

To cite this article: Laura Schröder et al 2020 J. Phys.: Conf. Ser. 1618 042040

View the article online for updates and enhancements.
Uncertainty propagation and sensitivity analysis of an artificial neural network used as wind turbine load surrogate model

Laura Schröder¹, Nikolay Krasimirov Dimitrov¹, John Aasted Sørensen²

¹DTU Wind Energy, DTU, Risø Campus, Frederiksborgvej 399, 4000 Roskilde, DK
²EIT, DTU Diplom, Lautrupvang 15, 2750 Ballerup, DK
E-mail: lausc@dtu.dk

Abstract. Recent studies have shown the advantage of replacing aeroelastic simulations with regression models based on Artificial Neural Networks (ANNs), which can be used as surrogate models for fast and efficient wind turbine load assessments. Once trained on a high-fidelity load simulation database covering a broad range of conditions, the surrogate model can be applied to predict loads for any site with wind climate falling within the range covered by the database. The aim of this study is to quantify the uncertainty propagation through such an ANN and to analyse how much the selected input variables influence the variance of the fatigue blade load estimations by means of a global sensitivity analysis. Results confirm that the selected ANN architecture seems suitable for this task resulting in small output uncertainties. Furthermore, the sensitivity analysis shows that the turbulence is mainly responsible for the blade load estimation, followed by the wind shear and the wind speed. The contributions of the turbulence length scale, turbulence anisotropy factor and wind veer angle are comparatively low. Comparing three different methods for sensitivity analysis shows that the partial derivative algorithm, Sobol variance decomposition and Shapley effect result in similar sensitivity measures.

1. Introduction

Wind turbines are typically designed based on reference wind conditions that are provided by the design standard IEC 61400-1 [1]. In order to ensure a turbine’s structural capacity at a specific location, site-specific load assessments need to be carried out using aeroelastic simulations which is a time consuming process and involves complex modeling. Recent studies have shown the advantage of replacing these aeroelastic simulations with regression models, which can be used as surrogate models for fast and efficient load assessments. Once trained on a high-fidelity load simulation database covering a broad range of wind field conditions, the surrogate model can be applied to predict the loads of a specific turbine type for any site with wind climate falling within the range covered by the database. However, since real world data comes with uncertainties it is important to assess how these uncertainties propagate through the surrogate model to the output, as well as to understand which variables have highest influence on the load estimations. In [2] such a sensitivity analysis is carried out for various surrogate models, such as Gaussian process modelling and polynomial chaos expansion amongst others. In [3] it was shown that Artificial Neural Networks (ANNs) can result in better prediction performances compared to the
previously investigated regression techniques. Hence, it would be good to carry out a sensitivity analysis on a ANN based surrogate model as well.

The aim of this study is therefore to quantify the parameter uncertainty propagation through a ANN based surrogate model and to analyse how much the selected input variables influence the variance of the load estimations by means of a global sensitivity analysis. This study can lead to a better understanding how suitable the selected ANN architecture is for the specific problem, as well as how the choice of different sensitivity methods can effect the analysis results.

2. Methodology

Data and model

The detailed framework for setting up the surrogate model and selecting the most suitable ANN architecture is presented in [3]. The six probabilistic input variables characterizing the wind field are wind speed, turbulence, vertical wind shear exponent, turbulence length scale, turbulence anisotropy factor and wind veer. The input samples are generated in a uniform space and transformed to physical space by assuming a certain joint probability distribution. The corresponding damage-equivalent loads (DEL) which serve as target for training the ANN are obtained with a 10,000-point Monte Carlo simulation using the aeroelastic tool HAWC2 [4] with the DTU 10MW reference turbine [5]. This sampling approach ensures a wide range of combinations of wind speed, turbulence, wind shear, turbulence length scale, and anisotropy factor. Hence, various atmospheric stability ranges [6] and load scenarios under normal turbine operation are implicitly considered in this set up. This study focuses on estimating the lifetime DEL for the blade root flapwise bending moment $M_x$. A feed-forward ANN is trained using the same network architecture as [3], however with the difference of using the input variables in the uniform space instead of using input variables in real space. This is done in order to ensure having independent input variables which is necessary to carry out the sensitivity analysis.

![Figure 1. Schematic illustration of a feed-forward neural network with three hidden layers. Image taken from [7] with the author’s permission.](image)

A feed-forward ANN consists of a set of neurons which are organized in layers (see Figure 1). The input vector $x$ with $d$ elements is passed forward through the hidden layers of neurons in order to compute the output vector $y$ with $k$ elements. Each hidden layer consists of a set of hidden neurons at which the incoming information is processed in two steps. Firstly, at each hidden neuron $j$ from the first hidden layer the output $o_j^{(1)}$ is calculated by linearly scaling the
input as follows
\[
o_j^{(1)} = \sum_{i=1}^{d} w_{ji}^{(1)} x_i + b_j^{(1)}
\]  
(1)

where \(x_i\) is the \(i^{th}\) element of the input vector, \(w_{ji}^{(1)}\) the weight connection between \(x_i\) and the \(j^{th}\) hidden neuron from the first layer, and \(b_j^{(1)}\) the bias offset for the \(j^{th}\) neuron. Secondly, the result is passed through an activation function \(h\) (e.g. \(\tanh\)):
\[
y_j^{(1)} = h(o_j^{(1)})
\]  
(2)

The activated output then serves as input to the next layer, such that with a total number of hidden layers \(l\) and a total number of neurons \(m^{(l)}\) and \(m^{(l-1)}\) in \(l^{th}\) and \((l-1)^{th}\) hidden layer, respectively, we get following neuron outputs at each network layer:
\[
\begin{align*}
y_j^{(1)} &= h(o_j^{(1)}), \quad o_j^{(1)} = \sum_{i=1}^{d} w_{ji}^{(1)} x_i + b_j^{(1)} \\
... \\
y_j^{(l)} &= h(o_j^{(l)}), \quad o_j^{(l)} = \sum_{i=1}^{m^{(l-1)}} w_{ji}^{(l)} y_i^{(l-1)} + b_j^{(l)} \\
y_k &= h(o_k), \quad o_k = \sum_{i=1}^{m^{(l)}} w_{ji}^{(l)} y_i^{(l)} + b_j^{(k)}
\end{align*}
\]  
(3)

where \(w_{ji}^{(l)}\) denotes the weight connection between the \(i^{th}\) neuron from layer \((l-1)\) and \(j^{th}\) neuron from layer \(l\).

Training the neural network model consists of tuning the model parameters \(W\) so that the model attains optimal predictive performance, which is evaluated in terms of a cost function. In the present study, the ANN is trained on 70% of the generated data using back-propagation with the Levenberg-Marquardt algorithm and validated on 20% of the data to prevent overfitting. The finally selected ANN consists of two hidden layers and 11 neurons per layer resulting in 221 model parameters (see Table 1) and its performance is tested on the remaining 10% of the data.

Figure 2 shows the estimated DEL of the blade-root flapwise bending moment of the ANN with respect to the simulated values for the test data. The residuals between the simulated targets \(y(x)\) and the model outputs \(f(x,W)\) are calculated for the test set:
\[
\epsilon = f(x,w) - y(x)
\]  
(4)

The ANN predicts the DEL on the test set with a mean error of \(\mu = 10.12\) Nm and a standard deviation of \(\sigma = 667.44\) Nm assuming normally distributed residuals (see Figure 3).
**Figure 2.** Scatter plot of blade-root flapwise bending moment estimated by ANN against simulated database.

**Figure 3.** Error distribution of load estimation on test set using an ANN with two hidden layers with 11 neurons per layer.

**Table 1.** Number of weight and bias parameter per network layer

<table>
<thead>
<tr>
<th>Layer 1</th>
<th>Layer 2</th>
<th>Output layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of weight parameter $W$</td>
<td>66</td>
<td>121</td>
</tr>
<tr>
<td>No. of bias parameter $b$</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

**Linear error propagation for non-linear models**

Each parameter estimator $\hat{W}$ has an uncertainty that is propagated to the model output. This uncertainty can arise from uncertainties in the input variables, selection of network architecture or also in the optimization algorithm used for finding the optimal parameters. To calculate the parameter estimator uncertainty a linear error propagation for non-linear models is used [8][9].

The first-order derivative of the model with respect to the parameter estimators (Jacobian) is formulated as followed:

$$J_W = \frac{\delta f(W)}{\delta W}$$

(5)

With a total number of simulations $n = 998$ and number of estimated parameters $p = 221$, the unbiased variance of the residulas of the parameter estimation is estimated:

$$s^2 = \frac{\epsilon^T \cdot \epsilon}{n-p}$$

(6)

Subsequently, the covariance matrix of the parameter estimators can be calculated using the Jacobian and the estimated variance:

$$\text{Cov}(W) = s^2(J_W^T J_W)^{-1}$$

(7)

Using linear error propagation the errors from the parameters can be propagated to the model outputs by estimating the covariance of outputs:

$$\text{Cov}(y) = J_W \cdot \text{Cov}(\hat{W}) \cdot J_W^T$$

(8)
Partial derivative algorithm

Amongst other techniques that have been suggested in the last 30 years for analysing the sensitivity of an ANN, the partial derivative algorithm \[10\] is one of the most commonly used \[11\]. This method is based on the Jacobian matrix \(J\) representing the partial derivatives of the output \(y_k\) with respect to the inputs \(x_i\). The Jacobian matrix includes the sensitivity of the ANN output to small input variations and can be calculated using the chain rule \[12\]:

\[
\frac{\delta y_k}{\delta x_i} = \frac{\delta y_k}{\delta o_k} \frac{\delta o_k}{\delta y^{(l)}} \ldots \frac{\delta y^{(l)}}{\delta o^{(l)}} \frac{\delta o^{(l)}}{\delta x_i}
\]

where \(h'\) denotes the derivative of activation function \(h\). Using the first-order derivatives a sensitivity index \(c_i\) can be calculated for \(n\) number of samples of each input \(x_i\) on the output \(y\) \[10\]:

\[
c_i = \frac{1}{n} \sum_{n} \left| \left( \frac{\delta y}{\delta x_i} \right)_{n} \right|
\]

Sobol variance decomposition

Another method for analysing the sensitivity of the model output is by means of Sobol indices that are obtained from a Sobol’s variance decomposition using sampling with binning. The variance decomposition method decomposes the variance of the output into contributions from each of the \(d\) individual inputs \[13\]:

\[
V(y) = \sum_{i=1}^{d} V_i + \sum_{i=1}^{d} \sum_{j>i}^{d} V_{ij} + \sum_{i=1}^{d} \sum_{j>i}^{d} \sum_{k>j}^{d} V_{ijk} + V_{1,...,d}
\]

with \(V_i = V(E_{x_i \neq x_i}(f(x|x_i)))\) being the variance due the main effect of a single variable and \(V_{ij} = V(E_{x_i \neq x_i,x_j}(f(x|x_i,x_j)))\) being the output variance due to the interactions between the \(i\)-th and \(j\)-th variables. In the global sensitivity analysis the variances are normalized by the total output variance:

\[
S_i = \frac{V_i}{V(y)} \quad S_{ij} = \frac{V_{ij}}{V(y)}
\]

The main effect Sobol index \(S_i\) is the output variance that can be explained by the main effect of the \(i\)-th variable. Following the sampling with binning method, the variance \(V_i\) and subsequently the Sobol index \(S_i\) can be calculated separately for each input variable. The total sum of the global Sobol indexes can be greater than one as the effect of variable correlation is not taken into account.

Shapley effects

In the following, the Shapley value is presented as an alternative to the partial derivative algorithm and the Sobol’ index for determining the individual input variables contribution to the total output variance. The Shapley value of input variable \(i\), denoted \(\phi_i\), is constructed such that the sum of all input Shapley values results in one when using independent input variables.
Following [14], here are presented some of the key elements in the definition of a Shapley value of the above multilayer ANN.

A constructive definition of the Shapley value can be found in [15], page 991: "If we rearrange the $d$ variables into all $d!$ orders and find the improvement in $R^2$ that comes at the moment the $i^{th}$ variable is added to the regression, then $\phi_i$ is the average of all those improvements." The Shapley value of input no. $i$ is then determined by

$$\phi_i = \frac{1}{d} \sum_{u \subseteq -\{i\}} \left( \binom{d-1}{|u|} \right)^{-1} (\tau^2_u + i - \tau^2_u)$$

where

- $-i := [d] \setminus i$ is the set of input variables with input $i$ removed.
- $u \subseteq -\{i\}$ is all the subsets of the input variables with input $i$ removed $\cup$ with the set of input variables, with $i$ removed.
- $\binom{d-1}{|u|}$ the number of ways a subset of cardinality $|u|$ can be selected from a set of cardinality $d-1$.
- $\tau^2_u = \sum_{v \subseteq u} \sigma^2_v$, where $\sigma^2_v$ is the variance of the set $v$.

Examples on further references to computation and applications of the Shapley values can be found in [16], [17].

3. Results

Uncertainty analysis on model parameter estimators

To obtain the Jacobian $J_W$, the first-order derivatives of the model output with respect to the parameters estimators are calculated using the chain rule. The covariance matrix of the parameter estimators is calculated as described in Section 2.

The standard deviation $\sigma$ and the corresponding confidence intervals are calculated for all 221 parameter estimators assuming the confidence intervals follow a student t-distribution. The standard deviation of the parameter estimators varies between $\sigma_{min} = 0.002e-4$ and $\sigma_{max} = 6.254e-4$ with a mean standard deviation of $\sigma_{mean} = 0.206e-4$. Figure 4 shows the weight (blue) and bias (red) estimates including their estimated 95% confidence intervals.

![Figure 4. 221 model parameter estimates including 95% confidence intervals](image-url)
In order to analyse which set of parameters can be used uniquely, i.e. if it can be estimated independently and with sufficient accuracy, the correlation of each parameter pair is calculated as followed:

\[
Corr(\hat{W}) = \frac{Cov(\hat{W})}{\sigma \tilde{\sigma}}
\]  

(13)

98.74% of the parameter pairs meet the condition of having a correlation below 0.5, and 1.26% have a correlation above 0.5 (Figure 5). Another criteria for identifying if the set of parameters can be used uniquely is by calculating the relative error of the parameter estimate \(\sigma/W\), which should be lower than e.g. 25% [18]. All 221 parameter estimations have a relative standard error lower than 25% (Figure 6).

**Figure 5.** Correlation between 221 parameters including threshold at 0.5

**Figure 6.** Relative standard error of the parameter estimation

To further investigate what effect the uncertainty of the parameter estimation has on the model output, the covariance matrix \(Cov(\hat{y})\) and confidence intervals on the model output are estimated and illustrated in Figure 7.

**Figure 7.** Model output of ANN including 95% confidence intervals due to parameters uncertainty. The values are ordered by the model output.
Partial derivative algorithm
After obtaining the first order derivatives for all input variables $J_x$ using the chain rule as described in Section 2, the corresponding sensitivity index $c_i$ is calculated (see Table 2).

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Sensitivity Index $c_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind speed $u$ Turb. $\sigma_u$ Wind shear $\alpha$ Mann length $L$ Anisotropy $\Gamma$ Veer angle $\Delta \phi$</td>
<td>0.330 0.496 0.348 0.098 0.037 0.013</td>
</tr>
</tbody>
</table>

Sobol variance decomposition
Since the Sobol indices become challenging to interpret for dependent input variables the Sobol variance decomposition is applied to the input samples in the uniform space as well. For each input variable the model outputs are sorted with respect to the input and binned into 25 bins comprising of around 40 samples each. Figure 8 shows the binning of the model output with respect to the six input variables. With the mean output of each bin, the variance $V_i$ and subsequently the Sobol index $S_i$ is calculated separately for each input variable (see Table 3). The variation in the turbulence $\sigma_u$ accounts to 64% of the total variance in the model output. After the turbulence, the wind shear and wind speed have the highest contribution to the output variation with 22% and 17%.

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Variance $V_i$</th>
<th>Sobol Indices $S_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind speed $u$ Turb. $\sigma_u$ Wind shear $\alpha$ Mann length $L$ Anisotropy $\Gamma$ Veer angle $\Delta \phi$</td>
<td>0.029 0.092 0.032 0.004 0.005 0.004</td>
<td>0.172 0.637 0.220 0.029 0.034 0.031</td>
</tr>
</tbody>
</table>

Figure 8. Model output with respect to sorted input variables (uniform space) including mean output for each bin (red dots)
Shapley values
The Shapley values are calculated using the R function \texttt{shapleySubsetMc()} from the R language package \texttt{sensitivity} that is available on the CRAN repository [17]. The function applies an input parameter $N_i$ for the number of neighbor element, used in a nearest neighbor method in the Shapley value estimation. Figure 9 shows the Shapley values of all 6 input variables for 30 different $N_i$ values ranging from 3 to 950. The optimal number of nearest neighbours is selected based on [19] assuming a single mode density estimation applied for the neighborhood size $N_i = O(n^{4/(4+d)})$, where $d = 6$ and $n = 998$ leading to a neighbor number of $N_i = 16$ as presented in Table 4.

![Figure 9](image)

**Figure 9.** Shapley values of all 6 input variables with respect to the number of nearest neighbors $N_i$ used

<table>
<thead>
<tr>
<th>Wind speed $u$</th>
<th>Turbulence $\sigma_u$</th>
<th>Wind shear $\alpha$</th>
<th>Mann length $L$</th>
<th>Anisotropy $\Gamma$</th>
<th>Veer angle $\Delta \phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.148</td>
<td>0.587</td>
<td>0.204</td>
<td>0.037</td>
<td>0.021</td>
<td>0.003</td>
</tr>
</tbody>
</table>

4. Discussion
The linear error propagation method shows that the parameter uncertainty is relatively small with a relative standard error or the parameter estimators of $\sigma_W < 15\%$ and the resulting confidence intervals of the model outputs are acceptable when comparing to the simulated target values (see Figure 7). Furthermore, taking a look at the parameter estimates and their confidence intervals, Figure 4 shows that nearly all weights of the first hidden layer that are related to the three input variables turbulence length scale $L$, anisotropy factor $\Gamma$ and wind veer $\Delta \phi$ are close to zero with very small confidence intervals. This could already indicate that these variables
are less important for the load prediction. In general, the uncertainty analysis shows that the applied ANN architecture results in acceptably low uncertainties and is therefore suitable for this problem.

Figure 10 shows a comparison of the sensitivity analysis using the three different methods, namely partial derivative algorithm, Sobol variance decomposition and Shapley effect. It can be seen that the three methods show similar results indicating that the turbulence $\sigma_u$ inhibits the highest sensitivity, followed by the wind shear $\alpha$ and the wind speed $u$. The input variables with the lowest influence seem to be the anisotropy factor $\Gamma$ and Mann length $L$. Physically, this also makes sense as the turbulence intensity is known to be one of the main drivers of the fatigue loading of a wind turbine blade [20], while the wind shear causes cyclic load variations on the blades which also increases the fatigue. Comparing the sensitivity indices of the six input variables against a previous study that has used the same data, but a different types of regression models [2] show similar results.

The advantage of the Shapley values over the Sobol indices or the sensitivity values from the partial derivative algorithm is that it sums up to 1 and therefore gives a more exact estimation of how much variance of the output is explained by each input variable. However, Figure 10 and Figure 9 shows that the Shapley value estimates depend significantly on the number of nearest neighbours used. The higher the number of neighbours used for the estimation, the less expressive the Shapley values become. Therefore, more future work is needed in order to select the optimum number of neighbours used for the Shapley value estimation, e.g. using a cross-validation.

However, one should be careful when interpreting the results since the analysis is carried out on independent input variables in the uniform space. Further future work is necessary for considering dependent input variables. In the case of calculating Shapley values an example for this is given in [21].

**Figure 10.** Comparison of sensitivity indices for the input variables wind speed $u$, turbulence $\sigma_u$, wind shear $\alpha$, mann length $L$, anisotropy $\Gamma$, veer angle $\Delta\phi$ obtained from partial derivative algorithm, Sobol variance decomposition and Shapley effect analysis.
5. Conclusion
This study demonstrates how the uncertainty is propagated through an ANN regression model for wind turbine fatigue load estimation, and analyses how much the selected input variables contribute to the variance of the output. Results show that the parameter estimators seem to inhibit a low uncertainty with a relative standard error below 15%. The parameter estimators of the first hidden layer already indicate that the input variables turbulence length scale $L$, anisotropy factor $\Gamma$ and wind veer angle $\Delta \phi$ are less important for the prediction. The model output results in acceptably small 95% confidence intervals of the model output.

Furthermore, the global sensitivity analysis shows that the turbulence $\sigma_u$ is mainly responsible for the output variance, followed by the wind shear $\alpha$ and the wind speed $u$. The contributions of the turbulence length scale $L$, turbulence anisotropy factor $\Gamma$ and wind veer angle $\Delta \phi$ are comparatively low. Comparing three different methods for sensitivity analysis shows that the partial derivative algorithm, Sobol variance decomposition and Shapley effect result in similar sensitivity measures.

Future work is required in order to consider dependent input variables as well as finding the optimal number of neighbours used for estimating the Shapley values. Furthermore, based on the sensitivity results it could be tested if a model using only wind speed, turbulence and wind shear as inputs will result in similar prediction performance as the present model.

References

[18] Gürkan S, 2018 DTU Summer School on Uncertainty and sensitivity analysis of numerical models Lecture slides

