Prediction and Validation of Fatigue loads using Artificial Intelligence on Real World Measurement Data

Anish Venu, Mike Lüdde, John Omole
DNV GL, Brooktorkai 18, Hamburg, Germany

Abstract. Monitoring the fatigue loads acting on the crucial structural components of a turbine plays a vital role in lifetime assessment, operational maintenance, load and power optimization of the wind farm. Generic physical models used in recent times are not highly reliable and does ascertain true decision making situations. To determine the realistic fatigue loads acting on the turbine components requires expensive instrumentation with strain gauges. This work describes using an artificial intelligent model on real measurement data to make cost-effective estimation of fatigue loads acting on the wind turbines. The model is validated with real measurement data gathered from two onshore and one offshore commercial wind turbines.

1. Introduction
The world has never been so digital as it is today. The rapid growth in digitalization has motivated the global industries to adopt these technologies into the wind energy business. Today’s wind industries are continuously looking for innovation that would help in increasing the turbine performance and making the technology a cost-effective one by reducing the Capital and Operational Expenditure (CAPEX and OPEX). In this case, knowing the mechanical loads acting on the crucial structural components of the turbine plays a significant role as the performance, and operational efficiency of the wind turbines are highly influenced by these dynamic loads. Currently, the operators/owners of the wind farm use a generic aeroelastic model to estimate acting loads on the turbine. These models come up with a lot of assumptions and uncertainties and in turn, does not contribute a lot during critical decision making situations regarding loads and power-performance optimization. Another standard way of knowing the mechanical loads is to carry out a measurement campaign on the subjected turbine and to record the load quantities on components of interest. This is done by implementing an extensive measurement set up and later analyzing the recorded data to know the acting loads due to the dynamic behavior of the turbine. This method of knowing loads is very reliable but on the other hand very expensive to install, maintain and calibrate the sensors used.

To address this challenge, DNVGL has introduced a digital framework, where the real-world measurement data are combined with artificial intelligent algorithms to enable smart and reliable monitoring of mechanical loads acting on the wind turbine. This framework would provide the operators/owners more confidence in critical decision making, in respect to load monitoring, power optimization and to strategically balance wind farm loads. Knowing the fatigue history will also help the original equipment manufacturers to determine the remaining operational lifetime of crucial turbine’s components.

This data driven approach fuses the commonly available SCADA data (Supervisory Control and Data Acquisition) along with the measured mechanical loads from the respective turbine, into a neural network model, where the rules of the model are built by the data itself. During the training phase, the neural network model captures the complex non-linear relationship between the SCADA data and the structural fatigue loads from the training data. Once the learning process is completed, the trained model
can give deep insight into the fatigue loads acting on the crucial structural components without the need for additional sensors in the future.

This paper explains the development and validation of a structural load prediction algorithm by using the mechanical loads data recorded on the commercial wind turbines by instrumenting conventional sensors. Data recorded from three commercial turbines (two onshore and one offshore turbine) are used for the validation, to check the flexibility, compatibility and reliability of the used algorithm. In all cases, a set of available SCADA and acceleration signals are used as input to the model, to predict the structural loads acting on the blade and tower of the turbine.

![Figure 1: Schematic representation of the model.](image1)

This paper is organized as follows. Section 2 describes briefly about the available commercial data used for validation of this model. Section 3 describes the data pre-processing of the available raw data and technique used to reduce the dimensionality of the used inputs. Section 4 gives a basic insight into the used neural network algorithm. Section 5 shows the results generated using the model and presents a quantification of the performance of the used model. Finally, Section 6 makes some concluding remarks and suggests useful future work.

2. Project data
Project data recorded from four reference commercial turbines was used to validate the load-prediction algorithm. 1. Prototype model of a 6-MW turbine erected in North Germany, 2. Prototype model of 2-MW turbine located in China, 3. One offshore 5-MW Senvion turbine located at Nord Sea, which are part of the Research at Alpha Ventus (Rave) offshore research project are used. All the structural loads in all reference turbines are measured in accordance with IEC TS 61400-13 and the coordinate system used for the measurement of loads is shown in Figure 2. More technical details from the reference turbines are not described in this paper due to the clause of confidentiality. Table 1 and Table 2 describes the list of available SCADA signals and structural data in the reference turbines. Table 3 describes the number of available concurrent SCADA and structural data available from each reference turbine for validation. The red stars in the table indicates that a particular signal is not measured in the respective turbine.

![Figure 2: Coordinates of all structural loads measurements.](image2)
Table 1: List of available SCADA and acceleration signals in the reference turbines.

| Signal Name            | Reference Turbine 1 | Reference Turbine 2 | Reference Turbine 3 |
|------------------------|---------------------|---------------------|---------------------|
| Electrical power output| *                   | *                   | *                   |
| Apparent power output  | *                   | *                   | *                   |
| Reactive power output  | *                   | *                   | *                   |
| Grid status            | *                   | *                   | *                   |
| Nacelle wind direction | *                   | *                   | *                   |
| Nacelle wind speed     | *                   | *                   | *                   |
| Pitch angle            | *                   | *                   | *                   |
| Rotor speed            | *                   | *                   | *                   |
| Generator speed        | *                   | *                   | *                   |
| Acceleration           | *                   | *                   | *                   |

Table 2: List of available structural measurements in the reference turbines.

| Signal Name                      | Reference Turbine 1 | Reference Turbine 2 | Reference Turbine 3 |
|----------------------------------|---------------------|---------------------|---------------------|
| Blade flapwise moment            | *                   | *                   | *                   |
| Blade edgewise moment            | *                   | *                   | *                   |
| Tower bottom normal moment       | *                   | *                   | *                   |
| Tower bottom lateral moment      | *                   | *                   | *                   |
| Tower top normal moment          | *                   | *                   | *                   |
| Tower top lateral moment         | *                   | *                   | *                   |

Table 3: Quantity of concurrent SCADA and Structural data available for the reference turbines.

| Reference Turbine 1: 6MW | 40 days |
|--------------------------|---------|
| Reference Turbine 2: 2MW | 89 days |
| Reference Turbine 3: 5MW | 180 days |

3. Data-dimensionality reduction

Due to computational complexity, data dimensionality has become one of the major barriers for neural network models. The existence of collinearity and multilinearity between the variables increases the search space in an exponential manner and sometimes also results in invalid models. Principal Component Analysis (PCA) is one of the well-known dimensionality reductions, which is used in conjunction with the learning algorithms to get rid of irrelevant and redundant information. PCA also enable us to reconstruct the original data from the compressed data, making the data reduction a lossless process.

Usually, the process of PCA is described in 5 steps [1][2]. The first step is to normalize the raw data set by subtracting the respective means from the numbers in the respective column. This produces a standardized data set whose mean will be equal to zero.

The second step is transforming the data vectors into a covariance matrix. The covariance matrix is simply a square matrix which describes the empirical relationship between the variables. In simple words, covariance is a matrix of numbers which describe the variance of data and the covariance among the data. Let us consider a data set of two variables x and y, the respective covariant matrix is calculated as follows.
Covariance matrix, \( \mathbf{C} = \begin{pmatrix} \text{cov}(x,x) & \text{cov}(x,y) \\ \text{cov}(y,x) & \text{cov}(y,y) \end{pmatrix} \)  

where \( \text{cov}(x,x) \) is defined as 

\[
\text{cov}(x,x) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x}) (x_i - \bar{x})
\]  

The next step is to calculate the eigenvalue and eigenvector of the covariance matrix by eigen decomposition. The eigen vector traces the line of the forces applied to the input and axes of greatest variance where the eigen value describes the magnitude of the force. The eigen vectors and eigen values are the solutions of the following characteristic equations

\[
\det(\lambda I - A) = 0
\]

where, ‘det’ is the determinant of the matrix, ‘I’ is the identity matrix and ‘A’ is the obtained covariance matrix. Once, the eigen values are found, the corresponding eigen vectors are calculates as follows

\[
\det(\lambda I - A) v = 0
\]

where ‘v’ is the calculated eigen vector for each of the corresponding eigen value.

The final step is to order the eigen values from largest to smallest so that it gives us the components in order of significance. Suppose if we have a data set of n attributes, we get n eigen values and eigen vectors. It turns out that the eigen vectors corresponding to the highest eigen values are the principal components of the data set. Usually the first p eigen values which corresponds to 95% (This value is usually decided by the user depending upon the application) of the data variance is selected for further proceedings. By this way, the dimensionality of the given data set is reduced without loss of valuable information. In addition, only the principal component with very small eigen value (lowest variance) is ignored as they do not contribute much to our model.

4. Deep Neural Networks [3]

Deep neural network attempts to simply and mimic the behavior of the human brain. In other words, ANN tries to replicate the highly-interconnected network of neurons where the output of any given neuron may be the input to thousands of other neurons. Let us consider a single input neuron as shown in Figure 3. In this case, a scalar input \( p \) is multiplied with a scalar weight \( w \) to form \( wp \). This is one of the inputs sent to the summer, where it added to a bias \( b \). The net output \( n \) from the summer goes into a transfer function \( f \), which produces a scalar neuron output \( a \).

Figure 3: Artificial neuron.

A log-sigmoid transfer function is used in this paper. These functions take the input values which might vary between plus and minus infinity and produces the output in the range 0 to 1, according to the below expression.

\[
f(z) = \frac{1}{1 + \exp(-z)}
\]
The log-sigmoid transfer function is commonly used in multiplayer networks that are trained using back-propagation algorithms. The sigmoid function itself isn’t a step function. However, the edge is soft, and the output doesn’t change instantaneously. This means there is a derivative of the function and this is important for training the network.

**Training of Deep Neural Networks [4]**

Many layers of interconnected neurons form a multi-layer deep neural network. Figure 4 shows an example of a three-layer neural network. In general, the training process of a neural network takes place in three steps, as follows.

The first step is to calculate the output of a given network with randomized or initialized weights and bias. In machine learning concepts, this step is commonly known as feed-forward pass and the algorithm used to perform this is known as feed-forward propagation.

\[
h_{w,b}(x) = h_1^{(3)} = f \left( w_{11}^{(2)} h_1^{(2)} + w_{12}^{(2)} h_2^{(2)} + w_{13}^{(2)} h_3^{(2)} + b_1^2 \right)
\]

In equation (6), \( f() \) refers to the node activation or transfer function (log-sigmoid function). The final output \( h_{w,b}(x) \) is nothing but the ultimate output of the network. Instead of taking the weighted input variables \((x_1, x_2, x_3)\), the final node takes as input the weighted output of the nodes of the second layer plus the weighed bias. In this way, the final output of the network is calculated using feed propagation pass.

![Figure 4: A three-layer neural network.](image)

The second step is to calculate the error between the network output and the real output and back-propagating the errors due to each neuron in the hidden layers. By this, the contribution of each neuron for a given error can be found. This is carried out using an algorithm known as back-propagation. So, the aim is to define an error function for the nodes in the output layer to find the effect of a change in particular weight on the final error. The generalized error function is derived as follows

\[
\delta_i^{(l)} = - (y_i - h_i^{(l)}) \cdot f'(z_i^l)
\]

where \( i \) is the node number and \( l \) is the layer number. \( Y \) represents the real output from the training data and \( h \) is the output from the node (calculated using feed-forward propagation). \( z \) here is used to represent the final output of the network in a simplified way.

The final step is to use an optimization function which would optimize our learning parameters (weights and bias) to minimize the errors calculated from the back-propagation algorithm. Gradient descent is used as an optimization algorithms together with neural networks in this paper. The gradient descent method uses the gradient to make an appropriate step change in \( w \) so that we approach towards the point in the 3D space where the error is minimum. As it sounds, gradient descent is an iterative process with multiple steps. Each time, the \( w \) is updated using the following equation

\[
w_{\text{new}} = w_{\text{old}} - (\alpha \cdot \nabla \text{error})
\]

here, \( w_{\text{new}} \) denotes the new \( w \) position, \( w_{\text{old}} \) denotes the old \( w \) position. \( \text{error} \) is the gradient of the measured error (from back-propagation) and \( \alpha \) is the fixed step size. By applying the above theories, a neural network can be trained and tuned until we have the best possible prediction.
5. Case study – Results and Discussion
The above described model is tested and validated on data recorded from three different reference turbines. In case of the structural measurements, for each 10-minute file, statistical descriptions (mean, maximum, minimum and standard deviation) are calculated from the high frequency data (50 Hz). In addition to this, Damage equivalent loads (DEL’s) are also calculated for each 10-min file. Damage equivalent load is a measure of fatigue impact for a given load quantity time history. The determination of the DEL is done by the following expression

\[ R_{eq} = \left( \sum R_i^{m} n_i / n_{eq} \right)^{1/m} \]  

where, \( R_{eq} \) is the equivalent load; \( R_i \) is the load of the \( i \)th class of the fatigue load spectrum; \( n_i \) is the number of cycles in the \( i \)th class of the fatigue load spectrum; \( n_{eq} \) is the equivalent number of cycles; \( m \) is the S-N curve slope based on the material used (In this study we used \( m=4 \)).

The SCADA data, acceleration signals and calculated DEL’s from all the three reference turbines are fused into the deep neural network model for the training phase. Only 50% of the available data from all the reference turbines are used in the training phase. Once the model is completely trained and tuned, the remaining 50% of data is used test the trained model. In this case, only SCADA and acceleration signals are given as input and DEL’s are predicted by the model itself. The predicted DEL’s are compared with the DEL’s calculated from measured data to quantify the performance of the model. It is noted that the model was trained with data irrespective of unfiltered site conditions (e.g. free-inflow sector, wake due to neighboring turbines, various shear and turbulence conditions) and all operational modes of the turbine (e.g. idling mode, normal power production, emergency start/stop conditions).

Figure 5 to Figure 10 shows DEL’s predicted by the model in comparison with the DEL’s calculated from the real measurement data in respect to the reference turbine-1. Note that the loads predicting model has access only to the SCADA and acceleration signals as it would be on a real commercial turbine. Also, note that the data shown in the plots are manipulated to protect the confidential design details.

![Figure 5: Prediction of DEL’s due to blade flapwise moment.](image-url)
Figure 6: Prediction of DEL’s due to blade edgewise moment.

Figure 7: Prediction of DEL’s due to tower bottom normal moment.

Figure 8: Prediction of DEL’s due to tower bottom lateral moment.
The Science of Making Torque from Wind (TORQUE 2020)  
Journal of Physics: Conference Series 1618 (2020) 022006  
doi:10.1088/1742-6596/1618/2/022006

In order to quantify the performance of the predicting model, coefficient of determination ($R^2$) and root mean square error (RMSE) is calculated between the predicted and real DEL’s. These factors are calculated for all the available DEL’s on all the reference turbines and the Table 4 shows the overview of all the results. The coefficient of determination and root means square error are calculated using the following formulas

$$R^2 = 1 - \frac{\text{Variance}(\text{DEL}_{\text{Estimated}} - \text{DEL}_{\text{Real}})}{\text{Variance}(\text{DEL}_{\text{Real}})}$$  \hspace{1cm} (10)$$

$$\text{RMSE} = \sqrt[2]{\sum(\text{DEL}_{\text{Estimated}} - \text{DEL}_{\text{Real}})^2}$$  \hspace{1cm} (11)$$

Table 4 that for most of the loads, the R2 factor is greater than 95 % and RMSE is less than 5 %. The predicting factors looks similar for all the turbines even though additional input signals were used for the reference turbine A. It also observed from Table 3 that the training data used for the reference turbine-3 was significantly more than the other two turbines. This extra quantity of training data does not directly reflect on the prediction accuracy of the model as results indicated in the Table 4.
Table 4: Result summary.

| DEL’S Description          | Coefficient of determination $\left(R^2\right)$ (%) | Root Mean Squared error (%) |
|----------------------------|-----------------------------------------------------|-----------------------------|
| **Site A**                 |                                                     |                             |
| Blade flapwise moment      | 95.30%                                              | 4.78%                       |
| Blade edgewise moment      | 98.90%                                              | 1.05%                       |
| Tower bottom lateral moment| 93.60%                                              | 6.60%                       |
| Tower bottom normal moment | 96.58%                                              | 3.60%                       |
| Tower top lateral moment   | 95.04%                                              | 5.10%                       |
| Tower top normal moment    | 96.56%                                              | 3.50%                       |
| **Site B**                 |                                                     |                             |
| Tower top normal moment    | 94.00%                                              | 6.00%                       |
| Blade flapwise moment      | 94.45%                                              | 5.84%                       |
| **Site C**                 |                                                     |                             |
| Blade edgewise moment      | 99.20%                                              | 0.78%                       |
| Blade flapwise moment      | 95.15%                                              | 4.97%                       |

To quantify the performance of the model in respect to the amount of training data used, a sensitivity study was carried out to find the sensitivity characteristic of the model. Data from reference turbine-3 was used for this study as its database is more representative when compared to other two turbines. The database is initially divided into 20, 40, 50, 60, 80 % packages. The model is trained with each data package and remaining data points is used as test data to validate the performance of the model. In this study, DEL’s calculated from blade flapwise bending moment signal was used as the target variable. Figure 11 represents the performance sensitivity of the model with respect to the quantity of data used. It seen that with just 40% of training data i.e. 72 days of data (for reference turbine-3), the $R^2$ calculated is close to 95% and RMSE is less than 5%. It is also noted that a further increase in quantity of training data does not influence the prediction accuracy to a noticeable level. This observation gives a good insight to the designer that with just a reasonable amount of good database, an efficient relationship can be built up between the input variables and the target DEL’s.

![Figure 11: Sensitivity Analysis.](image)
6. Conclusion
Deep neural networks with SCADA and acceleration signals as input variables and structural measurement data as target variables can learn the complex non-linear dynamic behavior of the turbine in respect to the acting damage equivalent fatigue loads. Reliable SCADA and acceleration signals that are already available in all wind turbine systems are used as input to the trained neural network model to predict the fatigue loads acting on the turbine.

The results of the case study conducted on the data gathered from three reference turbines shows that the used artificial intelligence model can accurately predict the DEL’s acting on the turbine to a very good extent. Coefficient of determination gives an indication of extent to which the fatigue loads is correctly predicted and this factor was found to be more than 95% for all DEL’s signals, used from different reference turbines. The root mean square was calculated to be less than 5% for most of the cases. It is noted that, the model was trained with data irrespective of unfiltered site conditions (e.g. free-inflow sector, wake due to neighboring turbines, various shear and turbulence conditions) and all operational modes of the turbine (e.g. idling mode, normal power production, emergency start/stop conditions). A sensitivity study was carried out to find the dependency of the model with respect to the required amount of training data and it was found that with just 40% (90 days of data from reference turbine-3), the model could make accurate predictions with less than 5% of calculated error. An increase in the amount of training data did not have a direct influence in the prediction accuracy of the model.

Transferring the trained model from one reference turbine data to other turbine of the same type in the same wind farm will be future work of this study. In addition to this, determining the extra information needed for the deep neural network models (for example shear, turbulence intensity etc.) to develop an effective transfer model will be an interesting future study of this work. This will in turn will help us to adopt a fleet leader approach to predict the fatigue loads acting on all the same type wind turbines located in the same wind farm. This will potentially help all the owners/operators to monitor fatigue loads acting on the turbines and strategically optimize the wind farm performance.

7. Reference
[1] J. Melton et al., Data Mining: Concepts and Techniques. 1999.
[2] L. C. Paul, A. Al Suman, and N. Sultan, Methodological Analysis of Principal Component Analysis (PCA) Method, vol. 16, no. 2, 2013
[3] M. T. Hagan and M. H. Beale, Neural Network Design.
[4] K. Gurney, An introduction to neural networks.