SCADA-based neural network thrust load model for fatigue assessment: cross validation with *in-situ* measurements

SCADA-based neural network thrust load model for fatigue assessment: cross validation with *in-situ* measurements

Francisco d N Santos, Nymfa Noppe, Wout Weijtjens and Christof Devriendt

OWI-lab, Acoustics and Vibrations Research Group (AVRG), Vrije Universiteit Brussel, Pleinlaan 2, B1050 Brussels, Belgium

E-mail: francisco.de.nolasco.santos@vub.be

Abstract. In this contribution SCADA data and thrust attained through strain measurements are used to train a neural network model which predicts the thrust load of an offshore wind turbine. The model is subsequently cross-validated for different turbines with SCADA data outside of the training period as input and the thrust load from strain measurements as the expected output, and the impact of wind speed and different operating conditions studied. The results for the model, such as MAE, are kept generally under 2 %. The estimated thrust load signal is then converted into the damage equivalent stress caused by the quasi-static load, allowing to quantify the damage induced by the thrust load. The model performed, in general, well, but some over-/underpredictions are severely amplified when converting the loads into the damage equivalent stress.

1. Introduction

Topics such as wind turbine fatigue, their remaining useful lifetime and possible lifetime extensions have become increasingly prevalent as older wind farms are reaching the end of their design lifetime. Thus, the reduction of the cost of energy remains a key concern of the industry. An informed decision to extend the lifetime of a wind turbine ought to take into account the fatigue assessment of said turbine, which can be based on measurements taken in the turbine. Furthermore, current offshore wind turbine design is often driven by fatigue, wherein improvements in fatigue assessment of built wind turbines can induce further optimization of future designs. In order to properly assist wind farm operators on such decisions, an accurate fatigue assessment is required for each and every wind turbine of the farm. When considering the current paradigm, the installation of strain gauges, from which strain measurements are attained, is rendered prohibitive due to its costly and time consuming nature. An underutilized alternative to such measurements is the use of high frequency supervisory control and data acquisition (SCADA) data, which exists in every new wind turbine by default. Using this data, an artificial neural network (ANN) model can be trained to estimate quasi-static loads (thrust load) and their contribution to the fatigue for all turbines within a given farm without the need for the installation of new sensors [1]. The ubiquity of data acquisition systems in modern wind turbines and the large amount of data they produce has lead to an increased use of ANNs, exhaustively documented by [2] and where several precedents for the use of ANNs and

wind turbine SCADA data have been presented (particularly in fault detection and diagnosis, such as with [3]). This data-driven approach might then present an alluring alternative to more pessimistic physics-based models, clearly being the next logical step in operational fatigue lifetime estimation [4]. One additional advantage of an ANN model over a physics-based one is that the latter requires thrust coefficient tables, which aren't always publicly available [5]. Another fundamental bibliographical source to the present contribution has been [6], in which a feed-forward neural network has been developed to predict blade fatigue loads for different flow conditions.

1.1. Objectives

The primary objective of this contribution is to cross-validate a neural network model trained on a real-world instrumented wind turbine that estimates the thrust load using SCADA data. This model is then validated on different wind turbines within the same offshore wind farm, with the final intent of a farm-wide application. A second objective is to translate the estimated thrust load signal into a fatigue load or damage, solely induced by the quasi-static force. This is relevant information for wind farm operators, as it allows to identify the wind turbines which present a higher degree of fatigue due to thrust loading and controller actions.

1.2. Methodology

The methodology employed consisted firstly in the development of a neural network model whose training was performed with the input of 1s SCADA data and labels (intended output) the thrust load from strain measurements from an offshore real-world wind turbine located in the Belgian North Sea (see blue dashed region in Figure 1). This methodology is based on the one described by [7], and the current contribution is a continuation of said work.

The resulting model is subsequently validated for one year for the same turbine, and then for one year on a different turbine within the same farm of the same make. Measurements from both turbines (both SCADA and strain) are used to validate the predictions, wherein the measurements of one turbine are used for its training and validation, but for the second one solely for the validation of predictions. This is closely related with the expressed objective of a farm-wide employment.

Finally the tower fatigue caused by the quasi-static loads is calculated (see orange dashed region in Figure 1). For this, the estimated thrust is converted into a stress signal. This signal is then reduced to a fatigue spectrum through a rainflow counting algorithm. Then, through the employment of Miner's rule and a S-N curve, these fatigue cycles are translated into the damage caused by the thrust load. Although not explored in the present contribution, it is acknowledge that higher frequency loads must also be taken into consideration for a full fatigue assessment, as these are also contributors to fatigue.

2. Neural network

In order to estimate the damage originating from the thrust load, an artificial neural network model was developed through the employment of the tensor-manipulation framework TensorFlow, in particular its high-level API machine learning library, *keras* [8], implemented on the programming language Python.

2.1. Topology

The implemented model consisted of a feed-forward neural network [9] composed by 4 hidden dense layers with a varying number of neurons (from 64 to 300). Several topologies were tested by trial and error, wherein the final topology was achieved through the best combination achieved through hyperparameter tuning (non-directly learned parameters, such as number of neurons

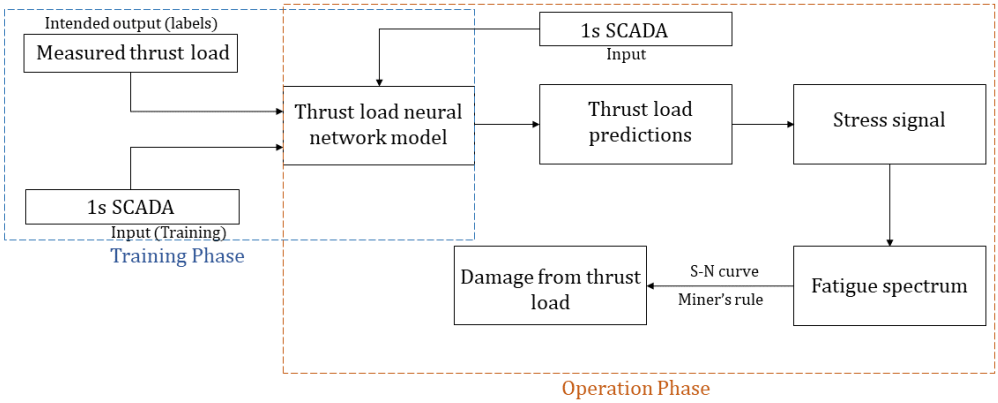


Figure 1: Schematics of the thrust-related damage calculation, with the training phase at blue, and the operation phase in orange. *N.b.* the thrust load predictions are validated on a one year period for different turbines.

and depth of model, *i.e.*, hidden layers). The layers were selected as densely connected, as the input tensors were 2-dimensional vector data and had rectified linear activation functions to ensure non-linearity. The model was then compiled using the `Adamax` optimizer, described in [10], which was found to be the most adequate for this particular application, and had as loss function mean squared error, selected as to increase the sensitivity to outliers. Finally, the model performance was evaluated through two metrics: the mean absolute error (Eq. (1)) and root mean squared error (Eq. (2)), where n represents the number of observations, Y_i the predictions and $\hat{Y}_i$ the real value.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i) \qquad (1)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \qquad (2)$$

In order to avoid overfitting of the model, an early stop call-back mechanism with a patience of 10 epochs, which monitors the validation loss, is implemented and the epoch-threshold is fixed at 100 epochs.

2.2. Input data

The 1 second SCADA data fed into the model consists of wind speed (m/s), rotor speed (rpm), mean pitch (deg), nacelle orientation (deg) and actual active power (kW) attained for a period of 12 carefully selected (taken as being representative of the operating conditions), non-consecutive, days from a real-world offshore wind turbine. This data is uploaded into PYTHON, pre-processed through normalization (in relation to the maximum) and the insertion of lags (of 1, 2 and 3 seconds; as to account for the lagged reaction of the thrust to SCADA parameters) to rotor and wind speed and power, before serving as input to the neural network. Strain measurements from the same time period are uploaded and undergo a temperature compensation before calculating the resulting bending moments and filtering this last signal. The filtered bending moment signal is subsequently translated into thrust load.

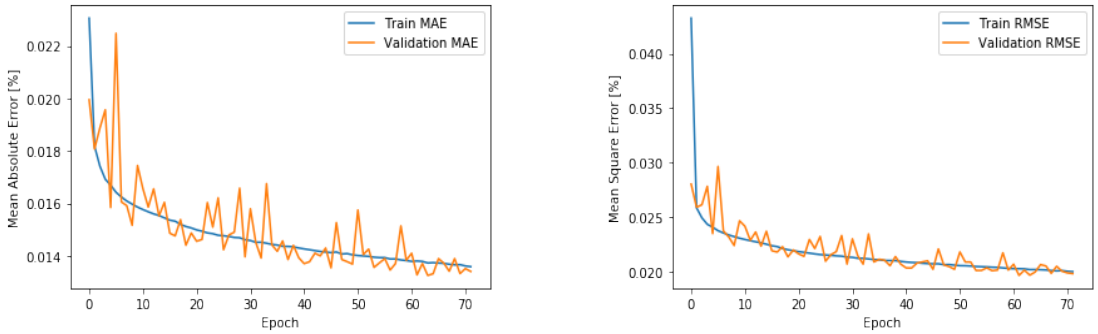
The local air density is also accounted for, through the linear relation: $F_{T,corr} = \rho \cdot F_T$, where the air density-corrected thrust load, $F_{T,corr}$, takes into consideration the influence of the air density, ρ. The filtering process, as well as the density correction are described in [7]. The thrust load, attained from the strain measurements, will be employed in the training of the neural network as the labels (the intended output).

It is then that all available pre-processed data (including the thrust load measurements) is sampled, being divided into a 80 % part, in-fed to the neural network training phase, and 20 % for testing. The testing data will not be fed into the neural network during the training phase, but instead will serve to evaluate the network performance (through mean absolute error and root mean squared error) on data it has not dealt with before. The data that runs through the network's training is itself sub-divided by a validation split of 20 % (necessary to evaluate the network's training progress and avoid overfitting).

2.3. Training phase results

The neural network model was trained for a maximum number of epochs of 100 but, due to the early-stopping call-back (which monitored the improvement of the validation loss, being triggered if no significant improvement was achieved after 10 epochs), the number of epochs the model required for training was inferior.

The metrics used to evaluate the performance of the training stage were, as previously mentioned, the mean absolute error (MAE) and the root mean squared error (RMSE), for both the training data, as well as the validation data, expressed in a percentage of the maximal thrust value. The evolution of these metrics according to the number of epochs can be seen in Figure 2a and 2b.



(a) MAE evolution.

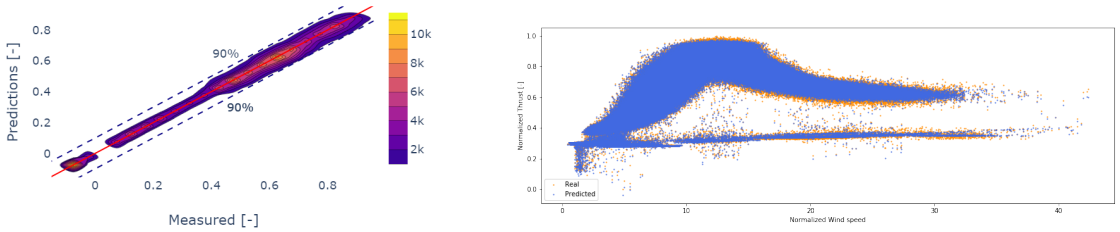
(b) RMSE evolution.

Figure 2: Evolution of training and validation MAE and RMSE according to the number of epochs.

As one can observe from both figures, after a relatively short number of epochs (≈ 10), the values quickly plateau, and converge. Additionally, we can verify that the validation MAE and RMSE curves follow the training curves, converging, which is an indicator that the ANN model hasn't overfitted (in which the validation curve would present higher values than the training).

To further verify the model's robustness, the previously sampled test data was fed into the trained ANN model. The model's performance for such an input was evaluated, again through the MAE and RMSE, which were, respectively, 0.013% and 0.020%. These values are in line with the training and validation MAE and RMSE of Figure 2a and 2b. Another form to visualize the model's accuracy is to plot the predictions and the true values of the thrust load and examine the deviation from the line $y = x$, which represents predictions equal to the measured values, as shown by the contour plot of Figure 3a.

In this figure the binned density of the points is represented by a gradient of colors, labeled by the bar on the right. As this figure shows, although there is some spread, the predictions do not deviate excessively from the red line in a given direction, and this spread is kept between the 90% lines, which can further reassure our confidence in the model's predictions. We can also verify that the higher density of points is firmly aligned with the $y = x$ line. Furthermore, we can



(a) Normalized predictions *vs.* measured thrust. Red line represents 100 % accuracy.

(b) Predicted and measured thrust plotted against wind speed (for the full training period of 12 days).

Figure 3: Comparison between predictions and measured thrust values.

notice that the plotted values intersect the lower 90% dashed line for higher normalized values, but not the upper one, leading us to conclude that, for this region, the model is underpredicting.

Moreover, we can attest the model's adequacy through Figure 3b, which has the predicted and measured thrust load plotted against the wind speed. As we can observe, the predictions encapsulate most of the measured thrust curve's behaviour.

The analysis underwent in this section allows us to affirm that the ANN model has been adequately trained and that the predictions from said model are sufficiently trustworthy for us to advance to the model's validation.

3. Model validation and cross-validation

3.1. Model validation

The neural network model presented in Section 2, was then validated over the course of one year on the same turbine of the training phase. In order to evaluate the model performance, the MAE and the RMSE were taken for every 10 minute interval. In Figure 4 the measured and predicted thrust plotted against the wind speed are plotted for a 3-month period, outside of the training period. As we can observe in this figure, apart from some areas of the measured thrust curve (which mightn't have been on the training period and aren't statistically as relevant), the predictions of the neural network model are generally in-line with said measured values, capturing most of the behaviour associated with the functioning of the turbine.

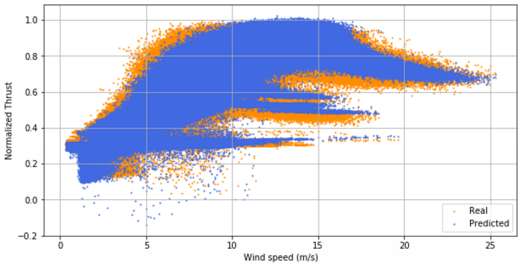
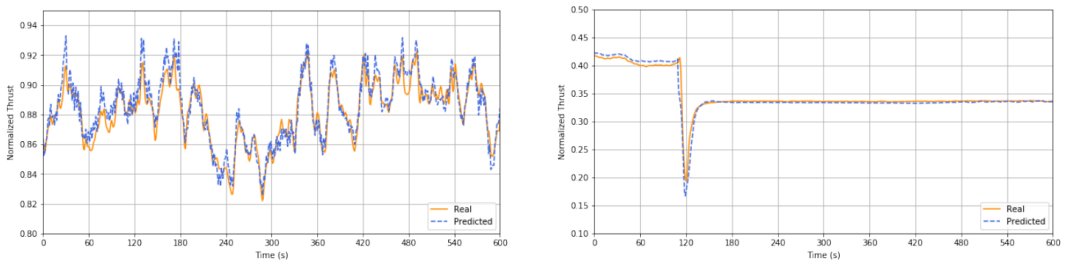


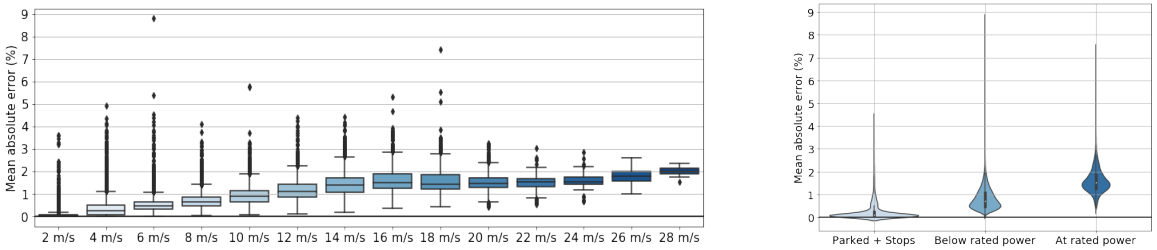
Figure 4: Predicted and measured thrust for 3 months plotted against wind speed.

If we then inspect discrete time series of interest, as with Figs. 5a and 5b we can recognise that, both for operating conditions (Figure 5b), as for idling and rotor stops (5b), the model's predictions accompany closely the measured thrust, even for more dramatic variations, as with rotor stops. The trustworthiness of the model is additionally confirmed by Figure 6a, where the MAE (binned according to the wind speed with a step of $2m/s$) is plotted against the wind



(a) (b)

Figure 5: (a) Measured (orange) and predicted (blue) thrust values for a 10 minute time instance at rated power. (b) As for (a) but for a rotor stop.



(a) (b)

Figure 6: (a) Box plot of MAE (binned) plotted against wind speed. (b) Violin plot of MAE for parked/rotor stops, below rated and at rated power.

speed. As we can observe from this figure, the mean MAE is generally under 2 %, slightly increasing with the wind speed.

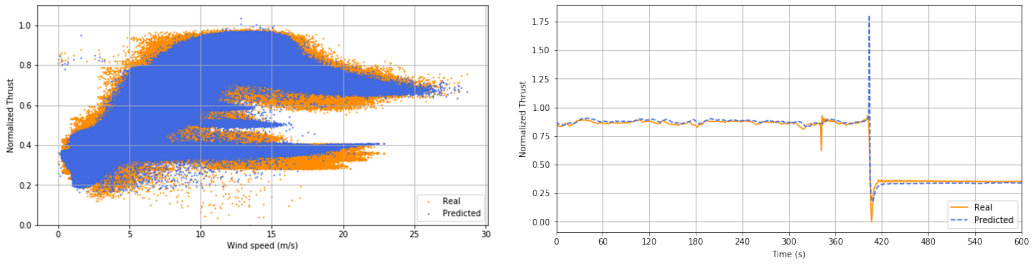
Finally, the MAE can be plotted for operation conditions of interest, namely, parked and rotor stops, below rated and at rated power (*vd.* Figure 6b). What becomes evident with this figure is the relatively higher MAE of rated power operation conditions and the highest value belonging to a below rated power condition. This can lead us into the conclusion that, although the error is systematically higher at rated power than for below rated power, one missed prediction for the latter events can have a high impact on the mean absolute error (the same applied for parked/rotor stops).

3.2. Model cross-validation

After having validated the ANN model for the same turbine of training, it was then cross-validated on a different, equally instrumented, wind turbine of the same model. As with Section 3.1, we begin by analysing the measured and predicted thrust loads plotted against the wind speed, present in Figure 7a, highly analogous to Figure 4, which is a good indicator of the robustness of the trained ANN model.

Upon closer analysis it was found that, even though for the vast majority of inspected time series the predictions match sufficiently well with the measured data, a small number of instances (3) were noted for having produced rather poor results, as evidenced by Figure 7b. The inspection of each individual instance lead to the conclusion that, for a small number of rotor stops, the model was overpredicting just before the drop in thrust. The SCADA data in and around these moments was carefully examined, without having produced a conclusion. The reason behind the overshoot remains, as of this moment, unknown.

Another point of comparison with Section 3.1 is the MAE box plot, as presented in Figure



(a) Cross validation predicted and measured thrust plotted against wind speed.

(b) (Measured (orange) and predicted (blue) thrust values for a 10 minute time instance where an overshoot before a rotor stop occurs.

Figure 7: Model cross-validation results.

8, where both the original and cross-validation turbines' MAE box plots are presented. We can observe that both curves are rather similar, with one and the other presenting a slight increase in the MAE as the wind speed increases and most values staying below 2 %. The two curves differ in the fact that, for the cross-validation, the MAE box plots present a higher number of outliers. We can understand the higher maximal value for the cross-validation (attained for $4m/s$) as being related to the overshoots pre-rotor stop, but, both the original and cross-validation turbines' MAE box plots, remain highly similar and allow us to affirm the applicability of the ANN model to offshore wind turbines, other than the one of training.

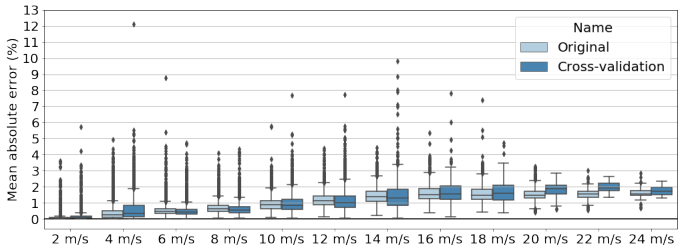
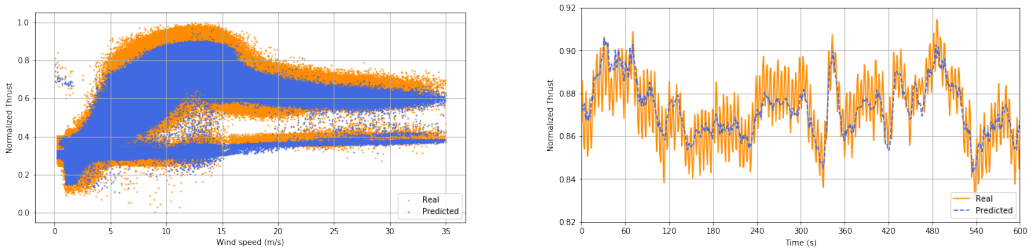


Figure 8: Box plot of MAE (binned) for the cross-validation turbine plotted against wind speed.

3.2.1. Unbalanced rotor One specificity of the cross-validation was the fact that the turbine was functioning with an unbalanced rotor for a period of the year. Thus, it was considered advisable to split the cross-validation between regular operating conditions and the unbalanced rotor period.

When dealing just with the period containing the unbalanced rotor, we can promptly identify the effect this will have on the accuracy of the predictions, as with Figure 9a, where, naturally, the predictions fall much shorter than in previous cases. The impact of the unbalanced rotor can be further evidenced by Figure 9b, wherein the unbalanced rotor effect can clearly be seen in the measured thrust (orange curve), which presents a dramatic change of values in a short amount of time.

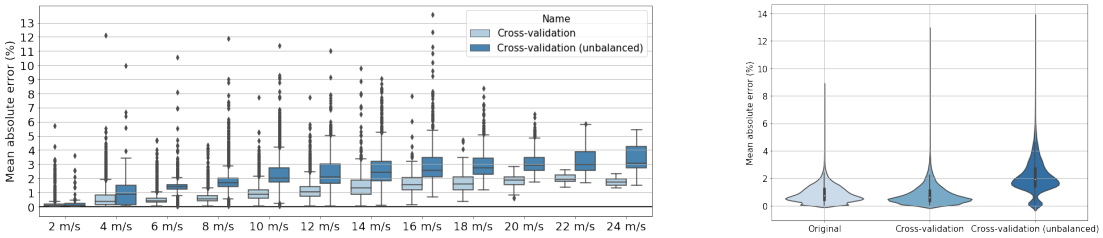
If we then observe the MAE box plots the unbalanced and regular cross-validation cases we can clearly see that, unlike with Figure 8, the evolution of the MAE according to the wind speed, although still increasing with higher values, is much different, as the mean MAE values are much higher for the unbalanced case, further confirming the disrupting influence of the unbalanced rotor.



(a) Measured (orange) and predicted (blue) thrust values plotted against wind speed for second turbine (unbalanced rotor).

(b) (Measured (orange) and predicted (blue) thrust values for a 10 minute time instance at rated power for second turbine (unbalanced rotor).

Figure 9: Unbalanced rotor predictions *vs* measured values.



(a) Box plot of MAE (binned) for the cross-validation turbine (regular and unbalanced) plotted against wind speed.

(b) Original, cross-validation and cross-validation (unbalanced) turbines' MAE violin plot.

Figure 10: Unbalanced rotor metrics evaluation.

We can finally compare the MAE attained for these three cases (original/validation turbine, cross-validation turbine and cross-validation turbine with the unbalanced rotor) for one entire year through a violin plot, as in Figure 10b. As previously evidenced, both the original and cross-validation MAE violin plots are highly similar, and close to 0 % (mean MAE: 0.85% and 0.81%, respectively). One should note the elongation on the cross-validations' plot due to the overshoots before some rotor stops. If we focus on the unbalanced case, then we can clearly notice that it presents a much higher mean value of MAE $(2,15\%)$, as well as a radically different distribution.

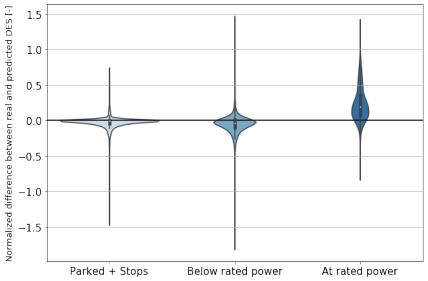
4. Damage estimate

As mentioned in Section 1.2, after proceeding with the thrust load reconstruction, the tower damage induced by said quasi-static loads is calculated. As the focus of this contribution is not the discussion of the damage calculation, this will not be debated in depth, deferring a more detailed discussion to [11]. Suffice to say that the thrust load is converted into a stress signal which, in its turn, can be reconstructed as a fatigue spectrum through the employment of a rainflow counting algorithm (reliant on a subset of the python implementation of the WAFO toolbox [12]), a well-known method widely discussed in literature, which combines the number of cycles with their stress range. A comprehensive discussion on rainflow counting can be found in [13]. It is then that, through the employment of a S-N curve and Palmgren-Miner's rule, the damage equivalent loads are calculated for the quasi-static thrust loads. For this application we present a linearization of the damage value, the damage equivalent stress (DES), as defined by [14] (*vd.* Eq. (3), wherein m is the slope of the S-N curve, n_i is the number of cycles of a given

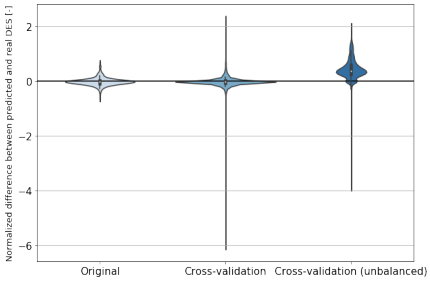
stress range, σ_i, and $N_{eq} = 10^7$, a predefined number of cycles):

$$DES = \left(\frac{\sum_i n_i \cdot \Delta\sigma_i^m}{N_{eq}} \right)^{1/m} \tag{3}$$

We can begin by analysing the impact each operating condition has on the damage. For this, the damage of the original turbine was calculated for every 10 minute interval for an entire year of operation, both for the measured, as for the predicted thrust. These values were subsequently normalized as to their corresponding mean value and the predicted damage subtracted to the measured damage. The operating conditions taken into account were: parked and rotor stops, below rated power and at rated power, representing $14,6\%$, $70,7\%$ and $14,7\%$, respectively, of the total cases.



(a) Original turbine DES difference violin plots for parked, below rated power and at rated power operating conditions.



(b) DES difference violin plots for original, cross-validation and cross-validation (unbalanced) turbines.

Figure 11: DES violin plots.

What becomes quite apparent from this figure is that the difference in the DES between predictions and measured values is on average close to zero for parked/rotor stops and below rated power (the latter having a bigger distribution spread), but with noticeable outliers. As for the rated power case, we can observe that there is a positive offset, as well as a wider distribution, which indicates us that the model is underpredicting/performing worse for these regions.

The next figure shown (Figure 11b) compares the difference in DES calculated for the three cases: the "original", training turbine, the cross-validation turbine and the cross-validation turbine during the period with the unbalanced rotor. What we can ascertain from this figure is the rather low mean DES difference for the original and cross-validation cases, both presenting a small distribution variation but, for the latter, the few overpredictions before some rotor stops induce extremely highly outliers, as noted in Figure 10b. It seems that, as the rainflow counting process is very sensitive to higher loads, the failure to accurately predict even just one of these, might have a catastrophic impact on the damage estimation. As to the unbalanced period of the cross validation turbine, again (if we take into account Figure 10b), there seems to be a clear underprediction (real values are superior to the predictions) and a bigger distribution variation, which is in line with the preceding chapters.

5. Discussion and outlook

The results of the damage estimation led us to verify a seasonality phenomenon, wherein a correlation between the damage and temperature was noted, leading to the need of a temperature dependency correction, as a future step (*vd.* Figure 12).

Asides from having to include temperature in further calculations, it was considered relevant to improve the training data by semi-randomly picking input variables throughout one year.

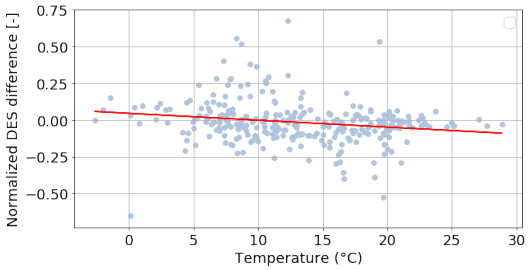


Figure 12: Correlation between damage and temperature.

Another possible investigation line would be to train a model, based on these results, but on a 10 minute scale, including other relevant data (as accelerometer or temperature) and directly predict damage. All of these approach should culminate in a farm-wide employment.

6. Conclusions

In this contribution the results of a data-driven approach to offshore wind turbines' thrust load modelling and damage estimation are presented. Firstly, an ANN was trained using data from 1s SCADA and strain gauges during a 12, non-consecutive, days period based on the methodology described by [7]. The model provided by the training was then validated on a one year period for the training turbine, for which metrics as the MAE was generally kept under 2%. The model was then validated on another turbine under regular operation with equally satisfactory results, apart from a small number of rotor stop overshoots. Finally, the model was applied to the cross-validation turbine in a period in which it was functioning with an unbalanced rotor. The model wasn't able to capture this behaviour, which is coherent, as this wasn't part of the training set. The results for the aforementioned cases were compared between themselves.

The final step was the estimation of the DES for different operating conditions and for the three cases through the use of rainflow counting and Miner's rule. The model performed well, generally, but the existence of some over-/underpredictions is severely amplified after the rainflow counting.

References

[1] Noppe N 2019 *Performance monitoring and lifetime assessment of offshore wind farms based on SCADA data* Ph.D. thesis Vrije Universiteit Brussel
[2] Marugán A P, Márquez F P G, Perez J M P and Ruiz-Hernández D 2018 *Applied energy* **228** 1822–1836
[3] Wilkinson M, Darnell B, Van Delft T and Harman K 2014 *IET Renewable Power Generation* **8** 390–397
[4] Veldkamp D 2008 *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* **11** 655–672
[5] Baudisch R 2012 *Structural health monitoring of offshore wind turbines* Master's thesis Danmarks Tekniske Universitet
[6] Vera-Tudela L and Kühn M 2017 *Renewable Energy* **107** 352–360
[7] Noppe N, Weijtjens W and Devriendt C 2018 *Wind Energy Science* **3** 139–147
[8] Chollet F 2017 *Deep Learning with Python* (Manning Publications) ISBN 9781617294433
[9] Bishop C M 2006 *Pattern recognition and machine learning* (Springer)
[10] Kingma D P and Ba J 2014 (*Preprint* `1412.6980`)
[11] Hübler C, Weijtjens W, Rolfes R and Devriendt C 2018 *Journal of Physics: Conference Series* **1037** 032035 doi: `10.1088/1742-6596/1037/3/032035`
[12] Brodtkorb P A, Johannesson P, Lindgren G, Rychlik I, Rydén J, Sjö E *et al.* 2000 *Proc. 10'th Int. Offshore and Polar Eng. Conf., ISOPE, Seattle, USA* **3** 343–350
[13] Marsh G, Wignall C, Thies P R, Barltrop N, Incecik A, Venugopal V and Johanning L 2016 *International Journal of Fatigue* **82** 757–765
[14] Hendriks H and Bulder B 1995 *Fatigue Equivalent Load Cycle Method: a General Method to Compare the Fatigue Loading of Different Load Spectrums* (Netherlands Energy Research Foundation ECN)