Review of ‘Data-driven surrogate model for wind turbine damage equivalent load’, R. Haghi and C. Crawford

This manuscript presents a neural network-based surrogate model of Damage Equivalent Loads (DEL) from aero-servo-elastic simulations of a wind turbine. It specifically generates synthetic wind timeseries from Sobol samples of inflow conditions (mean wind speed, turbulence intensity and wind shear) and utilizes a Temporal Convolutional Network (TCN) to project these into a latent space, with a Fully-Connected Neural Network (FCNN) topology acting as the read-out. In the results, the paper begins by i. benchmarking the TCN-FCNN model against a generic FCNN; ii. addressing worse side-to-side (SS) performance by infusing tower SS accelerations into the inputs, iii. demonstrating the TCN-FCNN expressiveness by estimating DELs under wake utilizing transfer learning and iv. discussing the minimal amount of data required.

The manuscript is well-written and the methodology well-described. The authors have produced an adequate amount of analysis. Moreover, the topic of surrogate models of numeric simulations is of great relevance for the wind community, and this work is a welcomed contribution.

There are, nonetheless, some comments that ought to be addressed in order to improve the overall quality of the paper.

Overall Methodology:

- There are good reasons for directly estimating DELs instead of bending moment reconstruction, but these are never provided in the manuscript. Moreover, it is explicitly stated that not only fatigue loads, but also maximal loads are of interest. Doesn’t it make more sense then to reconstruct the bending moment signal and from it, extract the maximal loads and cycle count to get the DEL?
- In lines 65—80 there’s a decent literature review of surrogate models of wind energy systems using neural networks. However, some important bibliographic elements are missing. Please consider reading and adding:
    - i. Movsessian A, Schedat M, Faber T. Feature selection techniques for modelling tower fatigue loads of a wind turbine with neural networks. Wind Energy Science 2021; 6(2): 539–554.
    - ii. d N Santos F, D’Antuono P, Robbelein K, Noppe N, Weijtjens W, Devriendt C. Long-term fatigue estimation on offshore wind turbines interface loads through loss function physics-guided learning of neural networks. Renewable Energy 2023
    - iii. Mylonas C, Abdallah I, Chatzi E. Deep unsupervised learning for condition monitoring and prediction of high dimensional data with application on windfarm scada data. In: Springer. 2020 (pp. 189–196).
    - iv. Vera-Tudela L, Kühn M. Analysing wind turbine fatigue load prediction: The impact of wind farm flow conditions. Renewable Energy 2017; 107: 352–360.
    - v. Duthé, G., de Nolasco Santos, F., Abdallah, I., Réthore, P.-É., Weijtjens, W., Chatzi, E., and Devriendt, C. (2023b). Local flow and loads estimation on wake-affected wind turbines using graph neural networks and pywake. In Journal of Physics: Conference Series, volume 2505, page 012014. IOP Publishing.

- vi. Liew, J., Riva, R., and Göçmen, T. (2023a). Efficient mann turbulence generation for offshore wind farms with applications in fatigue load surrogate modelling. In Journal of Physics: Conference Series, volume 2626, page 012050. IOP Publishing.

- Lines 84-86: 'The available literature and research indicate a lack of sufficient exploration and demonstration of a SM capable of mapping high-resolution environmental time series, specifically wind and/or wave for both on- and off-shore wind turbines, to the fatigue and extreme loads on wind turbine components.' I agree with the sentiment, our field requires more work on load surrogating of aeroelastic codes. However, I wouldn't say that there's a lack of exploration of SMs. There are many SM papers showing this can work (as pointed to in the previous comment), but we still need surrogates which are flexible enough (so it's positive that in the authors approach only u,ti and alpha are truly used as inputs) and access sufficient high quality data, which brings me to the assertion that you are using high-resolution environmental time series. Aerodynamically speaking, something like LES is more accurate than FAST. However, the major issue with any numeric simulator is two-fold: i. wave dynamics (for offshore) are really difficult to capture, ii. the controller is almost more of a black-box than ML. This nuance needs to be added into the discussion and limitations of any simulation-driven approach. I would also focus more on the innovativeness of using temporal convolutional networks.
- Concern with the train-test data split: a convolutional network which takes into account time-dependency (TCN) is being used. Even though DEL is estimated on a 10-min basis, given the time dependency I would argue you cannot simply randomly pick 90% of the data, it needs to be sequential (so, the 9 first months of a 10 month period).
- Table two as a reference to transfer learning (TL-FCNN). However, this isn't defined previously before showing up in this table.
- Section 3.1,.2,.3 and .4 aren't part of the results. I would suggest a new section dedicated to data generation including now section 3.1, 3.2 and 3.3. Section 3.4. is also more part of the methodology than the actual results.
- In figures 7b and 7c flap-wise and edge-wise aren't correct. It should be the other way around. Check figures of, e.g. Tartibu, L. K., M. Kilfoil, and A. J. Van Der Merwe. "Vibration analysis of a variable length blade wind turbine." (2012).
- Line 434. I don't understand the presence of SGD here. Above you mention you're using Adam (when you present the FCNN's topology).
- In Section 3.5. you present your results in terms of errors and predictions vs. true values. However, it would be important to actually plot the load signals (timeseries, vs wind speed, …) to understand the relative importance of each predicted quantity and their physical behaviour.
- Lines 466-467: 'One may argue that including the wind time series in the *y* direction in the input would improve the tower bottom side-side moment *R*2 value. We tested this hypothesis, but it did not improve the accuracy of the TCN-FCNN model'. This is appreciated, but any claim you make needs to be substantiated. I.e., if you say you tested this hypothesis, then you need to show the results.
- You say in line 495 and around 'The TCN-FCNN approach offers a significant benefit by examining the wind's time series rather than solely its statistical properties. The DEL results from wind and/or wave time series oscillations. If we were to reduce these oscillations solely to wind or wave input statistics, this would undermine the accuracy of the DEL prediction.' You mentioned that the TCN can handle the complexities of wind timeseries over statistics and that this undermines accuracy, but the results don't show this. FCNN and TCN-FCNN results are similar. If you say that just wave and wind statistics are worse w.r.t. to timeseries

(intuitively, makes sense), then you need to prove this. Either you point to a reference showing this or you present results. However, if you see some of the suggested literature there are some fine examples that make it work.

- Lines 506-507: ‘We tested the TCN-FCNN architecture to assess its ability to handle ultimate loads. During our analysis, we found that the SMs could accurately predict ultimate loads with a comparable level of precision as DEL.’ You are again affirming something without proving it. You need to show results to make such an assertion.
- Lines 513-515: ‘In contrast, the TCN-FCNN approach, which relies solely on the flow information at the turbine location, demonstrates the capability to address wake challenges without necessitating additional inputs, provided that the flow characteristics over the turbine are well-defined.’ I wouldn't make this assertion. Wake, specially over an entire wind farm, isn't easily captured in terms of its loads by models based on a single turbine. There are very complex phenomena like boundary layer recuperation and wake effect accumulation which make it strongly non-linear.
- Section 3.10. There is a big problem with synthetically reducing your timeseries by cutting them: as you correctly point out, you'll miss a lot of cycles. Because you're applying an exponent (m), any missed long cycles (which are the ones you miss when you shorten the signal) will make your error explode because long cycles contribute very heavily for fatigue (even more so if your m is greater, 5, e.g.). Additionally, if you want to compare to industry, then the 10-minute window is pretty much standard.
- Conclusions: what is, in your opinion, the advantages of your timeseries TCN-FCNN approach over a statistics FCNN approach? How are these related to the operation of the turbine (e.g., rotor stops) and how to model these? What is in your opinion more important: a better model like TCN vs FCNN or including other data sources (acceleration)?

Clarifications:

- In lines 17-18 ‘The time-marching simulations are necessary for our work and research as they enable us to consider the inherent and necessary non-linearity in the wind turbine models’, can you additionally explain how aeroelastic simulations encode non-linearities?
- Lines 50--. A quite thorough review of DTs for wind has been done but, after reading this section, there is no explicit mention to what in fact is a DT. Digital Twins are an often convoluted and overused concept, so it would be important for the authors to clearly state what they understand by digital twin and why it is different from a SM.
- From Section 2 it is not clear if the FCNN and TCN-FCNN training and testing samples are the same. They should be.
- How did you arrive to this topology present in Table 1? Was any hiperparameter tuning performed, any ablation studies? It is more correct to use search the variable space (randoms earch, Bayesian, etc.) in an automated manner, e.g. using Optuna or keras tuner.
- In lines 205--: ‘CNNs have been used and are well known for classification proposes (Long et al., 2015). CNNs basics are well studied in the literature, and the interested reader is referred to Goodfellow et al. (2016); Long et al. (2015). Research has shown that TCN is better than Recurrent Neural Network (RNN) and LSTM in terms of performance, implementation, flexibility and versatility (Fawaz et al., 2019; Bai et al., 2018).’ – When you refer that TCN are better than LSTMs and RNNs, is this also in the context of classification problems? It should be clear for the reader that you are using convolutional networks for regression.

- Line 209: ‘a) the length of the output and input is the same’. How do you ensure (a) - length(input)=length(output) - if you have a timeseries of 10-minutes, but you only have a single 10-min value for DEL?
- Figure 3a and 3b. From both figures, it appears to me that the dilation factor already serves as a sort of dropout, or am I interpreting it incorrectly?
- Figure 4. Average pooling isn’t defined elsewhere in the text.
- How did you arrive to the topology in Table 2? Also, if you're learning in the latent space, what is usually done (e.g. with an autoencoder) is to then have a read-out where the number of neurons per layer increases, e.g. 8,16,32. How were the number of neurons of the presented hidden layers selected?
- Line 293-294: ‘*neq* is the equivalent number of load cycles which is usually the length of the simulation in *s*.’ By writing that neq is usually the length of the simulation it induces the reader to believe that neq is variable. Neq is a fixed number we use (almsot invariable 10e7, or lifetime DEL, or 1Hz DEL [which the authors use]) that enables us to compare different dynamic load timeseries by introducing the concept of equivalent load. This must be a constant throughout any period you are comparing (like the Wöhler exponent, it must remain constant). I don't understand what is meant here with neq = s. Additionally, what is the resolution of the DELs? 10-mins? This becomes clearer in subsequent sections, but it should be clearly stated when you introduce DELs that you’re going to calculate them for a 10-minute time window.
- In Equation (6), why isn't the mean wind speed also dependent on time?
- Lines 373-374: ‘For training and testing purposes, we only took into account nine synthetic wind time series in *x* direction out of 225 synthetic wind time series.’ Does this mean that only 9 timeseries were used for training/testing or that only 9 points in the rotor plane were selected?
- Lines 431-432: ‘Rather than training the SMs on all the training data, the training data set is divided into batches of 256 samples.’ This sentence induces the read into a wrong idea. Batch training is 'training on all the training data'. The model still see the full dataset set for each epoch, just divided into batches.
- In lines 465-466 you notice how the accelerations improve the performance, specifically for the tower bottom. It is however interesting how the greatest improvement is at the bottom and not the top, where you have the sensor installed. Could you perhaps expand on this, why does it happen and specifically the relation (or relative lack thereof) between tower bottom’s bending moment and the structural dynamics at the rotor level.
- Line 482. You say that FCNN ‘needs input variables that may not be available all the time’. But this critique can also be made of TCN-FCNN and even more so: the probability of models based on 1Hz data failing is greater than on 10-min statistics.
- Line 496: ‘One challenge here is to free the input from the timeseries’ length, which is not within the scope of this study.’ What is meant by this?

Technical corrections (typos, grammar, etc.):

- In the abstract, the sentence ‘Doing so can calculate fatigue and extreme loads on the wind turbine’s components’ is convoluted and doesn’t seem to be clear English.
- Line 65. Remove character ‘f’.
- Line 292 – ‘Wöhler slope’. Wöhler exponent or inverse of the S-N curve slope.
- The correct SI unit for second is s, not sec.
- Line 510 ‘wind speed’. Add mean(u).
- Line 512. Missing citation in Dimitrov.

Suggestions for better readability and more complete content (implementation left at your own good judgement):

- In Figure 1, suggestion to change the order: top data generation, bottom SM. We read from top to bottom and sequentially, it makes more sense to present first data generation, followed by SMs. Also, I suggest to clearly label the FCNN as the ‘baseline’ or ‘benchmark’ and alter its color w.r.t. TCN-FCNN.
- The inputs used are mean wind speed, turbulence intensity and wind shear. However, in the real world there is usually a degree of yaw misalignment. You can maybe include this randomly into your input variable space (yaw misalignment).
- Figure 5 could be clearer. Why is fine-tune step trainable also include the TCN? Isn't it clearer to simply point the frozen weights of TCN and then say that FCNN can be re-trained?
- Perhaps you might consider publishing your simulation datasets under an open access license, e.g. in Zenodo.
- In Figure 8 you're taking snapshots, but maybe a better way to compare with the Gaussian velocity deficit is to time-average it.
- You trained 6 independent model. It would possibly reduce the performance (marginaly, one would expect), but you could think of training multi-objective NNs.