Wind Energ. Sci. Discuss.,
https://doi.org/10.5194/wes-2019-80-RC2, 2020

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

Interactive comment on "Model-free Estimation of Available Power using Deep Learning" *by* Tuhfe Göçmen et al.

Martin Felder (Referee)

martin.felder@zsw-bw.de

Received and published: 25 June 2020

This paper adresses the problem of assessing the instantaneous power available at a wind turbine from the inflow average wind speed and turbulence intensity. The goal is to provide accurate 1 min estimates of available power, to compare against actual power delivered in cases of curtailment. Classical, model-based algorithms are compared against a novel model free method based on LSTM type neural networks. The whole study is based purely on simulated data, downsampled to 1 Hz resolution.

Overall, while the method described does seem to solve the problem, some basic rules of neural network training were followed only haphazardly, if at all. The motivations for some of the decsions made in the process (e.g. early stopping, network size, post-

Printer-friendly version

Discussion paper

processing) are not clear. I am aware that this is not a machine learning journal, but throwing a bunch of data into a ML toolbox and applying some new buzzword method to them is not good science. Before moving on to advanced methods like transfer learning, the authors would do well to better understand the basic model first, and find out what the limitations of a well trained single neural network really are. Because from a practical standpoint, training a model once and applying it in an unchanged fashion operationally is much easier than implementing a fault tolerant re-training strategy of any kind. At least it would be helpful to know how a properly tuned standard machine learning method, like a single network trained on data from all wind regimes, performes against the its transfer-learned competitors. So, in my opinion this paper needs some major rework in order to be considered for publication.

Detailed comments follow:

The citations in the Introduction should be balanced an little more. Despite the paper's focus on ML, there are only two relatively old (considering the dynamics of the field) references to ML use in wind power forecasting (lines 40 and 42), while we find about 15 citations related to grid codes and curtailment, and $\sim$10 on power curve modelling.

On a similar note, it seems questionable to discuss EKF application to the turbine model on three pages, including mathematics, while only showing one sketch of an LSTM neuron, with no explanation on how these models are actually trained. There is no mention of the loss function or the training algorithm used. At this point Appendix A definitely needs to be included and discussed. For example, why is the perfomance on the test dataset at the start of training about the same as after training? Why is the training stopped at epoch 37, while the test error is obviously still decreasing after a small bump? Why do you need LSTM at all, since your best lag comes out as 29 - hardly requiring a "long" memory. Also, in Figure A1 please start the ordinate at zero, otherwise the plot scale is misleading.

Figure 5 should be reduced to a well formatted table: The TensorFlow model dump

contains misleading information (e.g. what means "None" for the time step dimension? Why does "activation" have a shape but no parameters? ...). The block diagram is trivial and therefore redundant. [FS] das ist ein Screenshot von Keras model.summary() und der Plot der raus kommt wenn man in Keras plot_model() macht. Auf der einen Seite kann das jeder lesen und weiß was das None Zeug bedeuten soll, auf der anderen Seite ist es etwas lazy ;) Da es die Keras Leute aber alle so kennen kann man das imho schon so durchgehen lassen.

In line 226, "test dataset" is used to refer to the part of the data which is utilized for determining the early stopping point during gradient descent. Since several years, it has been common practice in the ML literature to call this the "validation dataset". The "test dataset" would then be the shorter time series mentioned in line 227. While I'm personally not happy with this nomenclature, I strongly suggest adhering to the quasi-standard here, to avoid further confusion.

The definition of "lag" needs to moved from line 248ff to the start of Sec. 3.1, otherwise the description of preprocessing is hard to understand.

Line 262ff: It seems very suspicious to me that the simple application of a Gaussian smoother improves your prediction. What information are you adding here that the LSTM model does not have? And why does the LSTM not have it? Is there so much random noise in the network output? Is the loss function for the training exactly the same as the one you use for estimating the test set errors? What is the width of the Gaussian smoother, and how was it determined? By introducting a smoothing function, you introduce correlations between the errors of many neighboring 1 Hz samples. Hence it could be argued that the statistics presented are no longer comparable, i.e. you should smoothe the results from the model based methods by the same Gaussian. Furthermore, the effective number of samples you compare against is reduced by a factor corresponding to the width of the filter. This raises the question of whether or not the interval used for evaluation is long enough to draw some of the conclusions in the paper.

Line 280ff: The only hyperparameter "optimized" (if you can call a 5 point 1-dim grid search "optimizing") for the low wind speed case is the lag, since the rest of the architecture was apparently derived from some rule of thumb, with no further explanation given. It is good practice to at least check a few combinations of network depth and width, including extremely small networks. These may obviate the need for transfer learning and/or output smoothing altogether.

Line 284: Why is it not possible to perform automatic hyperparameter search when the training time for the full model is only in the order of an hour? This kind of optimization is standard procedure in many ML applications.

In Figure 11, a) and b) are described, but not c). Designating the operation modes introduced in Fig. 2 as Max-Omega, Const-Omega and Min-Cp now as Op#1 to 3 is confusing. Please stick to one nomenclature. The comparison between a) and c) would greatly benefit from not repeating the "Power via Cp" curve from Fig. 8, but instead using the same vertical scale in both graphs.

Fig. 13/14: Now you are apparently using a 60 min timeseries for evaluation, while before it was 10 min? Since Fig. 13 and others of the same kind have unclear abscissa labels (not [s], but days and hours?), it is not clear anymore which data are used for what. Please clarify.

Since the focus of the investigation is the adherence to the 1 min/3.3% error grid code, and the errors against the 1 min timeseries are much smaller than the ones against the 1 Hz timeseries, it would be helpful to see the performance of the model based algorithms against the 1 min resolution timeseries as well, because this is what we are eventually interested in.

Another point that needs to be discussed before talking about practical applications is the validation of the algorithm on real observations, as compared to a pure simulation. Training neural networks on essentially noise-free input data is of course much less problematic than dealing with real-world data issues. In fact, the authors claim that

their method is superior to model-based approaches because of not explicitly relying on manufacturer data, but there is no proof to that claim. If no real-world data are available for testing, the claim could be corroborated for instance by simulating changes to Cp and observing the result on neural network performance.

Interactive comment on Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-80, 2019.

Printer-friendly version

Discussion paper