Dear reviewers Fausto Pedro García Márquez and Martin Felder,

Thank you very much for taking the time to review as well as your valuable comments and suggestions for the revision of our article. We have taken all of them into consideration to improve our work and prepared a revised version of the manuscript accordingly. Please find specific replies and pointers to your comments down below.

Many greetings,

Tuhfe Göçmen, Albert Meseguer Urbán, Jaime Liew, and Alan Wai Hou Lio

---

Fausto Pedro García Márquez

Major:

1. The paper presents some plagiarism issue, not for rejecting it, but I suggest to correct it because it looks like copy and paste with minor changes: Lines 16-26; 45-60; 74-78; 84-96; 178-179; 191-195

We believe what is referred here is not plagiarism (from other works) but a repetition of certain parts of the text within the manuscript. They are often included to emphasise a message further. Regardless, the indicated paragraphs are slightly rephrased to avoid 1-1 recurrence within the article.

2. The abstract shows what the paper describes, it is good, ´ but it does not show the novelty of the paper. The main novelties should be also ´ shown clearly in the introduction, I have not seen the keywords in the paper

The novelty of the manuscript originates from its multi-disciplinary character. The fact that it combines wind turbine control considerations and state-of-the-art forecasting tools to develop a model-free approach with online learning capabilities is novel. Additionally, it evaluates and validates the tool from power systems perspective with quantified grid code compliance, which is often overlooked. To reflect that, a few sentences have been added to the abstract, which are also included in the Introduction section.

The keywords section could not be located in the manuscript template, but could be added editorially if needed.

3. To describe all the variables mentioned in the equations employed

They are all now described, including the newly added set of equations for LSTM formulation.

4. To describe ´ and justify the hiper-parameters of the NN employed (“the final network has 3 hidden layers with 100, 100 and 40 LSTM neurons, as indicated in Figure 5.”) It is not clear enough.

A selective grid search is presented in Table 1 in the revised manuscript and corresponding discussions are now included in the paragraph before/after Table 1.

5. To include Appendix A to the text of the paper. The curves shapes are ´ a bit strange from my experience, are they correct? I do not say that they are wrong.

It is now Figure 5 in the text. The behaviour of the training history is now plotted for further epochs, which we hope makes trends a bit clearer. As included in the response to Reviewer #2 Martin Felder, the low validation loss at the very first epoch is still interesting – further details for that can be read further down in the corresponding response #4 to Reviewer #2.

6. Please detail clearly how the accuracy is calculated, and where is obtained the ´ accuracy of 1% mentioned in the abstract

For the First, Second, Transfer and Further Transfer LSTM networks the mean of the 1-sec and 1-min error distribution is less than 1% in almost all the test cases, except for 1 (with bias = 1.28%) - see Figures 6c, 11d, 14b, 16b and 17b for the presented test cases with varying inflow wind speed and TI levels.

Minor:

7. Do not use acronyms in the abstract. Use the acronyms origin before to use them, of to use the acronym when ´ they are mentioned before, and do not repeat, e.g. turbulence intensity. To revise all the paper carefully

When it comes to LSTM, it simply is a convention to use the abbreviation rather than 'long-short-term memory' within literature. Accordingly, LSTM is simply better known (and more searched in scientific literature databases) than its longer name and always referred accordingly in the other abstracts in relevant fields.

8. What does DTU mean? Technical University of Denmark?

Yes ☺ it is added to author affiliations as well. DTU 10MW follows the same convention with other reference wind turbines such as NREL 5MW.

9. Terms as "heavily" are not correct for a scientific paper

Substituted with highly.

10. Do not include a large ´ number of references together. They should be reduced or describe their contribution in the paper individually. See e.g. "e.g. Ministere des Affaires Economiques (2019); Energinet.dk (2017); 50Hertz, Amprion, Tennet, TransnetBW (2016); 20 Elia, Belgium (2015); EirGrid (2015); National Grid (2014)).": 1-2 is good, could be until 3 in certain cases, no more

The mentioned reference on page 1 is omitted.

11. "or as delta control" not correct

'either balance control, …, or delta control, …' now changed to: 'both balance control, …, and delta control, …'

12. “The wind turbine operator ´ can then announce its participation in the reserve market online or ahead of time with the intention of performing delta or balance control.” To rewrite it, it is confuse

Elaborated slightly as: ‘… Accordingly, the wind turbine operator can announce its participation in the reserve market online or ahead of time with the intention of performing delta or balance control for curtailment’

13. What is U in equation 1?

(effective) wind speed, now added below equation 1.

14. Finally, just curious, how have you plotted Fig. 6b-d?

The figures are generated by combining distplot and boxplot functions from Seaborn library in Python – see https://seaborn.pydata.org/generated/seaborn.distplot.html, and https://seaborn.pydata.org/generated/seaborn.boxplot.html.

Martin Felder

1. 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 ∼10 on power curve modelling.

Agreed – now the citations are updated to reflect more recent studies, also added articles that are published after the original submission of the manuscript.

2. 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.

The section on LSTM description is elaborated in the revised version. Mathematical procedure to update the cell state and define the final output of an LSTM neuron is presented.

3. There is no mention of the loss function or the training algorithm used

Adam optimisation algorithm and mean absolute error loss function are implemented for the training process - Information is now added to newly Figure 5 (previously in Appendix) training history.

4. 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?

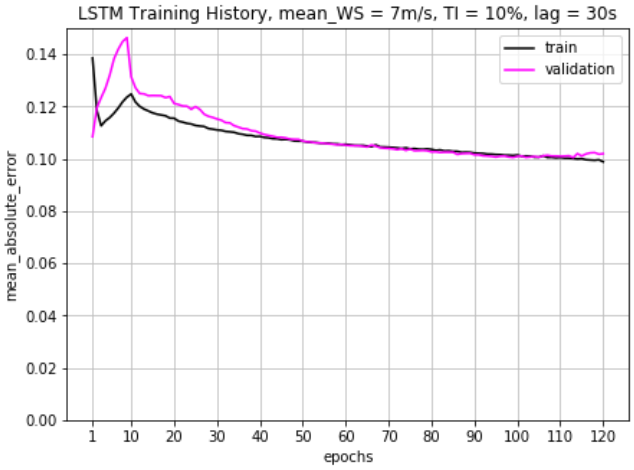
Appendix A is now Figure 5 – the early stop is not implemented any longer and the model training is shown until epoch = 120. Accordingly, it is clarified now that the validation loss is lower in the later epochs than the very first epoch. This low validation loss at the very first epoch is still interesting and found to be consistent for most of the architectures trained in Table

1. On the other hand, the trend disappears for other loss functions than MAE, *e.g.* MAPE which is not used in this study since it is based on the scaled parameters and the resulting normalised losses do not translate/agree well with the rest of the percentage error evaluations in the paper. Therefore, since it clearly does not indicate an overall higher accuracy, the network is trained further to its fuller potential using MAE as the loss function.

5. 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.

Both Table 1 (LSTM vs FFNN) and Table 3 (4s lag vs. 29s lag) in the revised version point to added benefits of using neurons with memory. Although 29s might not seem as much, it is still 29-steps lag for our 1Hz predictive network and covers about half the evaluation period as set by the grid codes.

Regarding the comment about the scale of the plot - since the difference between training and validation losses are small, it becomes hard to see if the training history plot y-axis ordinates at zero. For your convenience, we have put the example here. However, we keep the 'zoomed-in' version in the paper for enhanced clarity.



6. 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.

It is now formatted as Table 2. Similarly for transfer learning LSTM architecture on Table 5.

7. 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 quasistandard here, to avoid further confusion.

Agreed – the nomenclature now follows the ‘quasi-standard’.

8. 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.

It is now moved to the second paragraph of Section 3.1

9. 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.

These are valid and fair points – thanks again for detailed input here. Especially the reduction of the effective number of samples used for evaluation is the attribute that would be the most concerning for the structure and the concluding remarks of the paper. As listed in the next response, a broader grid search (which is now presented in Table 1) as well as further training of the model for large number of epochs have indeed provided a better performing network that the output smoothing with Gaussian convolution is no longer needed. Corresponding discussions have been omitted from the manuscript.

10. 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.

As you mentioned earlier, the journal not being an ML based journal we have avoided to include the grid search performed. It is now selectively included in Table 1 for both LSTM and feed-forward networks for a few combinations. The neurons have 50 step increase per layer and accordingly the final selection of neurons for the last layer have been increased from 40 to 50 neurons. This update in 3-layer LSTM with 100-100-50 neurons have resulted in better validation loss, and overall better performance for the test dataset of the First LSTM network. The results indeed alleviate the need of applying Gaussian smoothing function, but transfer learning is still found to be beneficial.

11. 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.

As indicated, the full training of the final LSTM network with the optimal performance is in the order of an hour, where the smaller networks take a bit less time. Accordingly, performing an automatic hyperparameter search for number of layers, number of neurons per layer and ideal lag would take up to a day for a single flow case with a given wind speed when 20+ models are trained and compared against each other, depending on the combination of the gridded hyperparameter domain. This is clearly far from desired flexibility in the final application, where online training and continuous improvement in network performance is envisaged during the operation of the turbine. We should also note that, since the transfer learning is updating the last layer of the weights using the latest normal operation data, potential degradation in turbine available power (both small and large scale, temporally) is also taken into account.

The phrase '…It indicates the need to specifically tune the hyper-parameters for each separate flow case. It is a cumbersome process with very little room for automation.' is now changed to 'It is a cumbersome process with high computational cost.' to more clearly state the issue.

12. 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.

In Figure 11, all the subfigures a), b) and c) are now described. The reference for the control strategies are now unified as suggested, *i.e.* following Const-Ω etc. The comparison between Figure 11 a) and c) is performed in Figure 12 in terms of error distributions, therefore, although it changes the scale the 'Power via Cp' approach is kept in Figure 11 c) as it is still a widely implemented approach to estimate available power.

13. 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.

A general description of the evaluation periods and frequency of error distributions is now added at the end of Section 3.1 (just before Section 3.1.1). Since the training parameters reduce significantly for the transfer learning, the training/validation split of the 3-hour dataset is changed from 80-20 to 60-40 which provides more than an hour of data to validate the Transferred and Further Transferred LSTM networks. This is already mentioned in Section 3.1.3 – now lines 305-310. The ambigious x-axes for time labels are now corrected for all the time series plots to represent the steps in seconds.

14. 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.

Although agreed, for sharper focus of the article on LSTM performance evaluation, the discussions on model-based approach is kept limited to Section 3.1.3, hence on 1-sec basis within 10min test period. In Section 3.1.3, a well as the Introduction, there exist several arguments on why the focus of the article is model-free network – an additional argument being now Figure 12, following the suggestion raised in the next point of the review.

15. 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.

Thank you for the suggestion, that motivated us to include now Figure 12 and associated discussions on lines 355-360; as well as updated Figure 10(c). We have introduced ±5% uniform uncertainty to Cp for curtailment, where full availability of the information is assumed for max Cp, *i.e.* under normal operation. The results indeed point to significant deviations in the lowest biased performance for the model-based approach – further contributing to the arguments on high Cp sensitivity. That would clearly have no affect on the neural network performance as the wind speed and therefore the available power output of the turbine (*i.e.* the input and the output features for the network) would remain the same.

05/08/2020 15:25:30

Compare Results

Old File:

Model_free_Available_Power_Estimation_WES_Journal_v1.pdf

29 pages (11.26 MB)
07/11/2019 11:34:50

versus

New File:

Revision_Model-freeEstimationofAvailablePowerusingDeepLearning.pdf

32 pages (2.05 MB)
29/07/2020 12:34:45

Total Changes

1101

Content

409 Replacements
307 Insertions
137 Deletions

Styling and Annotations

12 Styling
236 Annotations

Go to First Change (page 1)

Model-free Estimation of Available Power using Deep Learning

Tuhfe Göçmen[1], Albert Meseguer Urbán[1], Jaime Liew[1], and Alan Wai Hou Lio[1]

[1]Department of Wind Energy, DTU, Technical University of Denmark, Frederiksborgvej 399, DK-4000 Roskilde

Correspondence: Tuhfe Göçmen (tuhf@dtu.dk)

Abstract. In order to assess the level of power reserves during down-regulation, the available power of a wind turbine needs to be estimated. The current practice in available power estimation is heavily dependent on the pre-defined performance parameters of the turbine and the curtailment strategy followed. This paper proposes a single input model-free approach dynamic estimation of the available power using recurrent neural networks. Accordingly, it combines wind turbine control considerations and modern forecasting methodologies for a model-free, single input estimation of available power. It enables a robust real-time implementation of dynamic delta control, as well as higher accuracy provision of the reserves to the system operators.

The model-free approach requires only 1-Hz wind speed measurements as input and estimates 1-Hz available power as output. The neural network is trained, tested and validated using the DTU 10MW reference wind turbine HAWC2 model under realistic atmospheric conditions. The unsteady patterns in the turbulent flow are represented via Long Short-Term Memory (LSTM) neurons which are trained during a period of normal operation. The adaptability of the network to changing inflow conditions is ensured via transfer learning, where the last LSTM layer is updated using new measurements. It is seen that the sensitivity of the networks to changing wind speed is much higher than that of turbulence, and the updates are to be implemented solely based on the altering inflow velocity. The validation of the trained LSTM networks on time series with 7, 9 and 11 m/s mean wind speeds demonstrates high accuracy (less than 1% bias) and capability of transfer-learning online. Including highly turbulent inflow cases, the networks have shown to comply with the most recent grid codes, which require the quality of the available power estimations to be evaluated with high accuracy (less than 3.3% standard deviation of the error around zero-bias) at 1-min intervals.

1 Introduction

As the share of wind energy increases in power systems around the world, new challenges regarding the control and operations of wind power plants are encountered. In order to maintain power system stability, transmission system operators (TSOs) are developing new grid codes requiring contributions not only from conventional generators, but also from wind power plants, globally. In this context, here we focus on the active power contribution of wind turbines to provide frequency support and power reserves via down-regulation (also referred to as curtailment, de-rating or de-loading).

The curtailment of wind turbines can be implemented both as balance control, where the turbine power output is reduced to a constant value, and as delta control, where the output is reduced by a certain percentage of the available power (Attya et al., 2018; Fleming et al., 2016; Hansen et al., 2006). Additionally, the active power reduction for both strategies can be achieved by

adjusting the rotor blade pitch angles and/or operating at a sub-optimal rotor speed compared to the maximum energy capture value (Wilches-Bernal et al., 2016). Although modern turbines are capable of implementing both balance and delta control, due to the uncertainties in the estimated available power (Göçmen et al., 2019; Göçmen and Giebel, 2018; Göçmen et al., 2016; Pinson et al., 2007; Pinson, 2006), balance control is the preferred industrial application as its set-point is independent of the available power (Kristoffersen, 2005). The amount of power reserves however, which is defined as the difference between the available and the produced power under curtailed operation, does depend on the available power in the wind for both delta and balance control. This is particularly critical for compensation schemes under mandatory down-regulation, as well as (existing and expected) flexible balancing market structures, where the reserve power is traded at different time scales depending on the regional balancing market schemes (Chinmoy et al., 2019).

Generally, trading in the electricity markets is performed in advance with a given forecast horizon. Depending on the bidding structure, the available power production of an asset is to be predicted sometimes as short as 5-min ahead (*e.g.* (Rana and Koprinska, 2016)). The forecasting tools can be based on physical or statistical modelling, as well as the combination of both. Many perform post-processing via model output statistics to reduce the remaining error. Some approaches focus on the best possible estimate of the local wind speed while some directly extract the wind power generation potential. Statistical models use explanatory variables and historical/online information (measurements, log-data, etc.), generally implementing recursive techniques, such as recursive least squares or artificial neural networks (or deep learning) (Wang et al., 2016). In fact, forecasting is the field with the most deep learning (and broadly artificial intelligence) applications in wind energy, *e.g.* (Ghaderi et al., 2017; Chen et al., 2018; Mujeeb et al., 2019), for a recent review on deep learning based wind speed forecasting for several forecast horizons, see (Bali et al., 2019). However, while forecasting the available power the operational status and potential effects of control scenarios are often overlooked, especially for higher (than *e.g.* 5-min) frequencies at a single turbine level.

For the operational considerations and higher frequency system stability issues, the time scales considered in the market-based forecasting are already long-term ahead. In order for the balancing responsible parties to get compensated during mandatory down-regulation by the TSOs, wind power plants are expected to provide information regarding their power production in much shorter time scales. As stated in the recent grid requirements in Germany (50Hertz, Amprion, Tennet, TransnetBW, 2016), the available power is to be calculated for 60-seconds intervals for down-regulated wind farms. Additionally, the 1-minute standard deviation of the percentage error of the available power is required to be less than $\pm$ 3.3% (after the pilot phase). The enforced regulations are difficult to comply and are subject to penalty if not met.

The current practice in available power estimation is to assess the incoming wind speed to derive the possible power output of the turbine via optimum performance curve. One of the most common approaches to approximate the (effective) wind speed is by solving the static wind power equation, which is widely adopted in the wind turbine industry as well as the wind research communities (*e.g.* (van der Hooft and van Engelen, 2004; Göçmen et al., 2014)). More details on the approach is provided in Section 2. (Ma et al., 1995) demonstrates that directly mapping the static relation does not give satisfactory performance and concludes that the inclusion of dynamic models can significantly improve the wind speed estimate. Thus, an increasing number of studies (*e.g.* (Østergaard et al., 2007; Meng et al., 2016)) began to utilize observer theory, in particular, Kalman

filtering. For example, in (Østergaard et al., 2007), the aerodynamic torque is considered as a system disturbance state, and it is estimated by the use of an observer-based system on a simple drive-train model with pre-defined dynamics for the aerodynamic torque. Subsequently, the calculation of the wind speed is done by inversion of the static mapping between the aerodynamic torque and wind speed. One of the drawbacks of searching through the static relation, to find the wind speed estimate, is its computational cost, where a Newton-Raphson method is often employed to find the corresponding wind speed given the turbine measurement on a discrete power coefficient C_p surface. On the other hand, some methods do not require the use of iterative gradient methods, for example, by considering the wind speed directly as a state to the system and such a wind state can be estimated via an observer/Kalman filter. In (Selvam, 2007), the wind dynamics are modelled as a random walk and augmented with a linear turbine model including a simple drive-train and tower dynamic model. A linear Kalman filter is then employed to estimate the wind speed for feed-forward control purpose. Similar techniques also have been utilised in (Stol and Balas, 2003; Simley and Pao, 2016). A study by (Knudsen et al., 2011) employed a non-linear turbine model including a simple drive-train, tower and wind speed dynamics where the effective wind speed is estimated by an extended Kalman filter . Similar methods are also reported in (Henriksen et al., 2012), where dynamic inflow model is included. Besides the Kalman filter-based approaches, some studies (Ortega et al., 2011, 2013) used a more advanced state estimation technique of immersion and invariance to construct a wind speed estimation with proof of global convergence under certain assumptions. For more details and further information on wind speed estimation, see (Soltani et al., 2013) and references therein.

The state-of-the-art available power estimation is highly dependent on the considered turbine models, as well as the operation strategy for curtailment. More specifically, the majority of the methods rely on the pre-calculated power coefficient, C_p, or the certified nominal power curve to convert (rotor-effective) wind speed to (available) power. However, the varying wind speed and turbulence levels activate different dynamics within the turbine structure and cause different control responses (Murcia et al., 2018). In addition, temporally and spatially local characteristics of the flow (*e.g.* humidity, temperature, etc.) and the condition of the turbine (*e.g.* blade erosion, dust, component wear or failure, etc.) highly affect the C_p and the power curve behaviour. Therefore, these generally deterministic approaches fail to represent the detailed dynamics required to produce high frequency available power signal accurately (Jin and Tian, 2010), and they are an important source of uncertainty (Lange, 2005). In order to tackle the inadequacy of the turbine models to fully represent the dynamic power output of a turbine under turbulent inflow, a model-free approach to transfer the wind speed to power is a strong alternative.

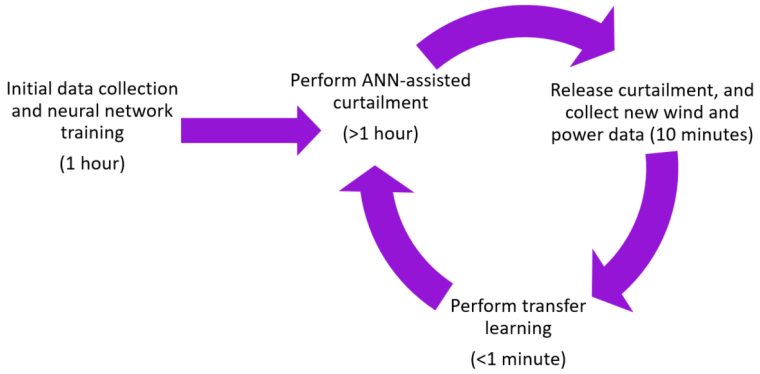


Figure 1. Flow diagram of the model free approach to estimating available power.

Therefore in this study, the aim is to bring the wind turbine generator (WTG) control considerations and modern forecasting methodologies for a model-free, single input estimation of available power. It enables a robust real-time implementation of dynamic delta control, as well as the provision of the reserves to the system level within the frame of (strictest) European grid regulations. In the model-free estimation of available power, the unsteady patterns in the turbulent flow is represented via Long Short-Term Memory (LSTM) neurons (Hochreiter and Schmidhuber, 1997), which is a special building unit for Recurrent Neural Networks (RNNs). The proposed method for integrating an LSTM network in a curtailment strategy is outlined in Figure 1. During a period of normal operation of a WTG, wind speed and power output time series data is collected for at least an hour to establish a training data set. Next, the network is trained on the collected data which, depending on the available processing power, can be performed within seconds. Accordingly, the wind turbine operator can announce its participation in the reserve market online or ahead of time with the intention of performing delta or balance control for curtailment. Down-regulation is then performed using the LSTM predictor which provides the set point based on the available power. The model-free estimation approach can be rapidly retrained with newly collected data using transfer learning, where the last LSTM layer of the network is updated using the new information.

The synthetic time series used in this study is generated using the DTU 10MW reference wind turbine (et al., 2013) with the aeroelastic code HAWC2 (Bak et al., 2012) under realistic atmospheric conditions, and the simulation results are publicly available[1]. First, the sensitivity of the state-of-the-art available power predictions to the curtailment operation strategy is briefly discussed and quantified in Section 2. To address the issue, a detailed analysis of LSTM neural networks and the potentials of transfer learning to adapt changing inflow conditions is presented throughout Section 3. This research focus is highly important for the individual turbine control and its role in the power system stability, as well as the business case of wind energy in the existing and upcoming market scenarios.

[1]The generated time series can be accessed here: https://gitlab.windenergy.dtu.dk/tuhf/deep-learning-for-available-power-estimation/tree/master/data

2 Wind speed-to-Power via Turbine Model

As stated earlier, current methods for estimating available power typically make use of pre-defined power curves or power coefficient calculations. Here in this section, we discuss the assessment of wind speed and the sensitivity of the model-dependent approaches to the implemented curtailment strategy.

Point measurements of the wind speed using, for example, cup or sonic anemometers, are often unreliable at estimating the potential power production of a wind turbine as the spatial variations in the wind field are not captured. For example, a naive approach at estimating available power is:

$$P_{avail}(U) = \frac{1}{2}\rho\pi R^2 C_p U^3 \tag{1}$$

where the air density, ρ, rotor radius, R, and power coefficient, C_p are assumed to be constant with variable (effective) wind speed, U. Equation (1) presents a number of weaknesses, namely the inability to capture the dynamic response of the wind turbine to changing wind speeds, or the spatial variations in the wind field. For this reason, the rotor effective wind speed, which is defined as the spatial average wind speed over the rotor plane, is preferred in terms of power estimation. Although there are numerous methods for estimating rotor effective wind speed, the majority of methods use operating data of the wind turbine to create the estimate (Jena and Rajendran, 2015). A simple strategy is to estimate the wind speed for a given power output using a polynomial fit (Thiringer and Petersson, 2005). This method can be extended by including the rotor speed and blade pitch angle in conjunction with a C_p look up table to infer the wind speed as shown in Bhowmik et al. (1998). For derated operation, these methods are problematic as the dependency between the wind speed and a turbines operating points vary based on the desired level of down-regulation. The use of a predefined C_p curve to estimate available power therefore becomes unreliable. However, state-space approaches where the convergence of the wind estimation error is analysed systematically can potentially respond to that problem.

There are several benefits of formulating a wind speed estimation as a system state estimation problem compared to methods that use static relation mapping the power or aerodynamic torque to wind speed. For example, a substantial body of mature and sophisticated state estimation theory can immediately be brought to bear upon the design of the wind estimator. Moreover, in an observer design where the wind speed is considered as a system state, the use of slow gradient methods for solving the static relations can be avoided, resulting in better computational speed and a smooth wind speed estimate.

Typically, to formulate a state estimation problem, a simplified model of the non-linear dynamics is required that needs to capture the key dynamics of the turbine. For brevity, a widely used non-linear turbine system model is employed, including the dynamics of rotor drive-train, tower and wind speed (See Knudsen et al. (2011); Lio et al. (2019)):

$$x_{k+1} = f(x_k, u_k) + w_{\mathrm{n},k}, \tag{2a}$$

$$y_k = h(x_k, u_k) + v_{\mathrm{n},k}, \tag{2b}$$

where $x_k = [\omega_k, \dot{x}_{\mathrm{fa,k}}, x_{\mathrm{fa,k}}, v_k]^T \in \mathbb{R}^{n_x}$ is the system state vector at the sample time $k \in \mathbb{Z}$ containing the rotor speed, fore-aft velocity and displacement of the tower-top and ambient wind speed, whilst the system input $u_k = [\tau_{g,k}, \theta_k]^T \in \mathbb{R}^{n_u}$ contains

the generator torque and pitch angle and $y_k = [\omega_k, \dot{x}_{\mathrm{fa,k}}, x_{\mathrm{fa,k}}]^T \in \mathbb{R}^{n_y}$ denotes the system output. The state transition and output functions are denoted as $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}, h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$. The Gaussian process noise $w_{\mathrm{n},k} \in \mathbb{R}^{n_x}$ represents the modelling errors whilst the Gaussian measurement noise $v_{\mathrm{n},k} \in \mathbb{R}^{n_y}$ represents the sensor noise and modelling error of the sensor dynamics.

Since the turbine model is a nonlinear model (2a), an extended Kalman filter (EKF) is employed to compute estimates of the wind turbine state. A Kalman filter is a computationally efficient and recursive algorithm that provides the optimal state estimates $\hat{x}_k \in \mathbb{R}^{n_x}$ by minimising the mean square state error or the state error covariance matrix $P_k := E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$. Kalman filtering approaches have been effectively employed in many examples of wind energy (e.g. Ritter et al. (2018); Lio (2018); Annoni et al. (2018)). Typically, in EKF, the estimate of the state $\hat{x}_k$ is computed in two-step processes: prediction and measurement update. The superscripts x_k^+, x_k^- are denoted as the variable x at sample time k after the measurement update and before the measurement update, respectively. The hat notation $\hat{x}$ denotes the estimate of x.

Prediction :

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_k), \;\; P_k^- = F_k P_{k-1}^+ F_k^T + Q_k, \;\; F_k := \frac{\partial f(\hat{x}_{k-1}^+)}{\partial x}, \tag{3a}$$

Measurement update :

$$\hat{y} = h(\hat{x}_k^-, u_k), \;\; \hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k), \;\; P_k^+ = (I - L_k H_k) P_k^-, \;\; H_k := \frac{\partial h(x_k^-)}{\partial x}, \tag{3b}$$

where $L_k \in \mathbb{R}^{n_x \times n_y}$ is the filter gain and it is computed as follows:

$$L_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}, \tag{3c}$$

where $Q_k \in \mathbb{R}^{n_x \times n_x}$, $R_k \in \mathbb{R}^{n_u \times n_u}$ denote the co-variance matrices of the process and measurement noises, respectively, that can be computed as $Q_k = E[w_{\mathrm{n},k} w_{\mathrm{n},k}^T]$, $R_k = E[v_{\mathrm{n},k} v_{\mathrm{n},k}^T]^T$. The process co-variance Q_k is chosen by approximating the variance of the modelling error and the typical wind speed. In this work, there is no measurement noise, thus, the measurement co-variance R_k is chosen as a small value.

One of the weaknesses of the EKF filtering approach is being a model-based method, that requires relatively an accurate model of the turbine. Besides, the choice of the model, operating conditions and sensor locations also strongly affect the EKF-based estimator performance (Lio et al., 2019). Some studies (*e.g.* Lio et al. (2018)) showed that down-regulation can be achieved by either modifying the generator torque solely or the combinations of rotor speed and torque. The constant and maximum rotation (Const-Ω and Max-Ω) strategies perform down-regulation by setting the rotor speed to a pre-determined or maximum value, respectively, whilst the min-C_{t} methods operate the turbine at minimum thrust coefficient in down-regulation. The performances of the EKF based upon these operations are shown in 2. The simulations are based on DTU 10MW reference wind turbine HAWC2 model (Bak et al., 2012) under 9 m/s mean wind speed and 10% mean turbulence intensity over 700 seconds. The turbines are commanded to operate at 40% and 80% of the rated power. One clear message from Figure 2 is that the performance of the EKF-based wind estimator is highly subjected to the turbine operating conditions, for example,

the performances were similar for strategies operating at 80% but the Max-Ω performed the worst at 40% down-regulation. Similarly, Min-C_t shows the best agreement with the available power for 40% down-regulation whereas it performs the worst for 80% curtailment. Therefore, Figure 2 indicates no clear trend and high sensitivity of model-based methods to the control scenario.

It should be noted that, the sensitivity observed in the synthetic time series in Figure 2 is expected to grow under the field conditions. This is due to the fact that the manufacturer-calibrated power coefficients cannot account for variability influenced by local conditions (Bandi and Apt, 2016). Additionally, the resulting uncertainty of the C_p dependent approaches is likely to be amplified also due to the lack of detailed information regarding the pre-defined C_p and implemented operation strategy for curtailment caused by the limited access to the controller in practice. To avoid the dependency on operating point estimations of available power, the use of wind speed measurements is revisited with the state-of-the-art deep learning architecture in the next section. The performance of this model-free approach is then compared with the presented wind speed observer, also for the scenario with limited C_p information during down-regulation.

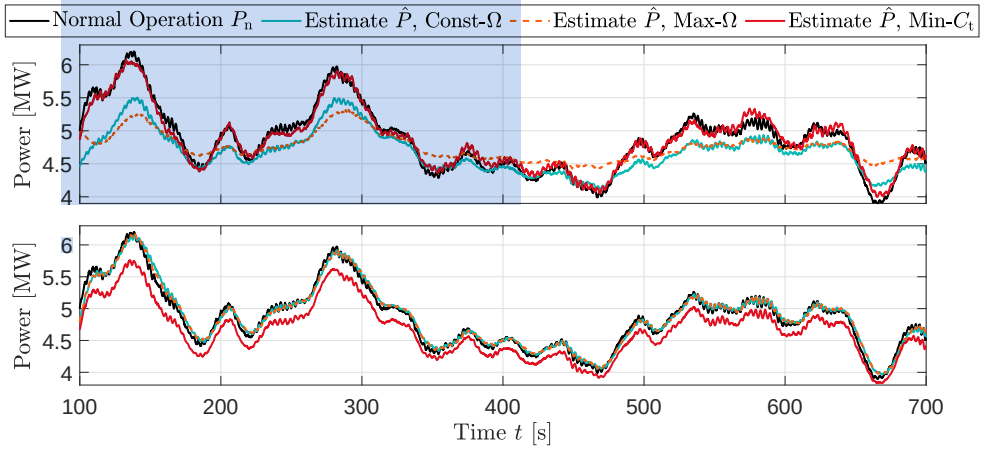


Figure 2. Time series of normal power P_n and available power estimation $\hat{P}$ based upon various down-regulation strategies. The top and bottom plots indicate the estimation based on measurements of turbines operating at 40% and 80% of the rated power, respectively.

3 Neural Networks for Available Power set-point

For a more robust operation and delta control, the bias and the uncertainties which is partly originated from the the natural variability of the flow and turbulence, and partly due to the uncertainty associated with the turbine models *i.e.* C_P surfaces, should be reduced. The former is investigated through a state-space update via Kalman filters in Section 2. Here, we implement a fully data driven approach, which is purely based on the atmospheric inputs to eliminate the dependency of the estimated available power production to the C_P surfaces and/or the control strategy.

Although the deep learning techniques have been applied to numerous engineering fields, their application in wind farm flow modelling has been rather limited. For the turbine level power estimation, recently neural networks have been implemented

to approximate the power curve mainly based on field data (for a detailed review, see *e.g.* Lydia et al. (2014)). Pelletier et al. (2016) applied feed forward neural networks (FFNNs) in a steady-state manner with 6 atmospheric inputs including shear and yaw error of the investigated turbine. Ouyang et al. (2017) approached to the problem by sectioning the regions of the power curve and developed a support vector machine algorithm for each partition, capable of capturing the dynamic response of the turbine. Manobel et al. (2018) on the other hand, underlines the importance of data filtering and normal behaviour recognition for such problems and also indicates that the architecture of the neural network needs to be re-optimised for each turbine within a wind farm, to increase accuracy.

Here in this study, we use the open-source machine learning repository, TensorFlow (Abadi et al., 2016) to implement the LSTM algorithm. LSTM architecture is a special type of RNN, which is shown to perform faster and better for highly fluctuating time series than many other RNN architectures. An LSTM neuron is illustrated in Figure 3, where there is no direct connection between the input i_t and the output o_t gates. All the information flows through the cell state c_t, which is the actual memory of the LSTM neuron and it is regulated by the forget gate f_t to avoid indefinite growth and eventual network break down (Gers et al., 2000). Through the calibrated weights, f_t decides how much of the previous cell state(s) is preserved, following equation 4.

$$f_t = \sigma\left(W_{fi}x'_t + W_{fo}h_{t-1} + b_f\right) \tag{4}$$

where σ represents the sigmoid gate, x_t is the input tensor of the current state and h_{t-1} is the output tensor of to previous state of the cell, W_{fi} and W_{fo} are the weights applied to input and output tensors of the forget gate respectively, b_f is the bias vector. Information is then transferred to the input gate i_t which is then forwarded to the cell state, c_t where they are selectively saved in the long-term memory. The mathematical procedure can be written as equations 5 and 6.

$$i_t = \sigma\left(W_{ii}x'_t + W_{io}h_{t-1} + b_i\right) \tag{5}$$

with W_{ii} and W_{io} are the weights applied to input and output tensors of the input gate respectively, b_i is the bias vector. The previous cell state, c_{t-1}, is then updated via:

$$c_t = \sigma\left(f_t c_{t-1} + i_t\right) \tag{6}$$

The updated cell state c_t then feeds regulated information to the output gate and finally the actual output of the neuron via equations 7 and 8.

$$o_t = \sigma\left(W_{oi}x'_t + W_{oo}h_{t-1} + b_o\right) \tag{7}$$

similarly W_{oi} and W_{oo} are the weights applied to input and output tensors of the output gate respectively, b_o is the corresponding bias vector. The final output of the cell is then defined using o_t and c_t via tanh function by:

$$h_t = o_t \tanh\left(c_t\right) \tag{8}$$

LSTM algorithms are heavily used in a variety of sequential/temporal predictive modelling, from language processing (*e.g.* Gers and Schmidhuber (2001)) to short-term forecasting (*e.g.* Zhang et al. (2019)). However, they require large amounts of

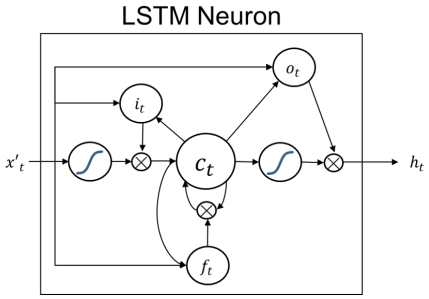


Figure 3. LSTM Neuron with cell state c_t, as well as input i_t, output o_t and forget f_t gates. x'_t and h_t indicate the inputs and outputs of the neuron, respectively. The curves represent sigmoid gates, σ.

data and computational resources to reach their full potential and achieve a generic solution without over-fitting. Therefore, although RNNs (and LSTMs in particular) have additional capabilities of modelling longer-term temporal properties, they remain highly challenging to train especially with limited training data. In recent years, the transfer learning (or knowledge transfer) approach that addresses such problems (Pan and Yang, 2010) has been increasingly popular. The basic idea of the transfer learning is that a well-trained model and its hyper-parameters that involve rich knowledge of the target task can be used to guide the training of other models.

Throughout the rest of this section, we will firstly present the details of the architecture and the hyper-parameter tuning of an LSTM model fully trained on 3-hours of HAWC2 simulations and compare the initial performance of LSTM neurons with simpler perceptrons in FFNN. We will then challenge our LSTM network to perform on another case with a different inflow condition than the original training domain. In pursuit of better performance on a different flow case, we will present the results from blind training as well as the transfer learning. We will then discuss their behaviour both in terms of the resulting error distributions and fitted parameters in between the layers. Finally, we will extend the application of the transfer learning to other flow cases, to demonstrate the flexibility and automation of the approach.

3.1 Data Pre-Processing and Training Strategy

The investigated case studies for available power estimation is generated and implemented using HAWC2 simulations with DTU 10MW reference wind turbine. For the training of the LSTM models, the high frequency (100Hz) wind speed signals from HAWC2 is down-sampled to 1-Hz, which is equivalent to the SCADA system of a wind-turbine (Göçmen and Giebel, 2018). The second input to the model is a moving (or rolling) standard deviation of the 1-Hz wind speed, with a 10-min rolling window as an indication of inflow turbulence intensity (TI). In contrast to the regular definition, this approximation of TI assures the same number of samples for both of the inputs.

The two inputs of wind speed and its moving standard deviation are first normalised between (0, 1) and then fed to the LSTM network to predict the power output during normal operation. As an LSTM neuron expects a 3-dimensional input shape in the order of samples, lag and features, the input data is shaped accordingly. For the defined architecture with 2 input features

listed above, the hindsight horizon to base the real-time estimations on, is another hyper-parameter to be tuned. The hindsight horizon, or lag, is the number of previous time steps that have been taken into account to predict the power output in the current time step. Note that longer lag would increase the initialisation period for the curtailment implementation and could be a limiting factor if the architecture is to be further adapted for online learning/training. Accordingly, for the LSTM networks the lag of 4s, 9s, 29s, 59s and 89s are investigated. Note that since the model is trained to map the atmospheric inputs to the actual production data under normal operation, the power predictions are ensured to follow the normal operation trend that is required for the available power estimation and not affected by the curtailment strategy.

For the preliminary evaluation of the training and hyper-parameter tuning of the model, a split validation dataset is generated. The final test of the model is based on an independent time series with a similar mean wind speed and turbulence intensity, but covers a shorter time period. Since the target application of the model is to estimate real-time available power for more certain delta control (or reserve provision), the main criteria of evaluation is 1-Hz error distribution for shorter test cases (10-mins), where grid code compliance is tested for longer available periods (1-hour) based on 1-min error distributions.

3.1.1 Training of the First LSTM Model : Low Wind Speed, High Turbulence Intensity

The case study to train the first LSTM model consist of 3-hours period, where hub-height wind speed and corresponding moving TI are used to estimate the power output of DTU 10MW turbine under nominal operation. The input time series are presented in Figure 4.

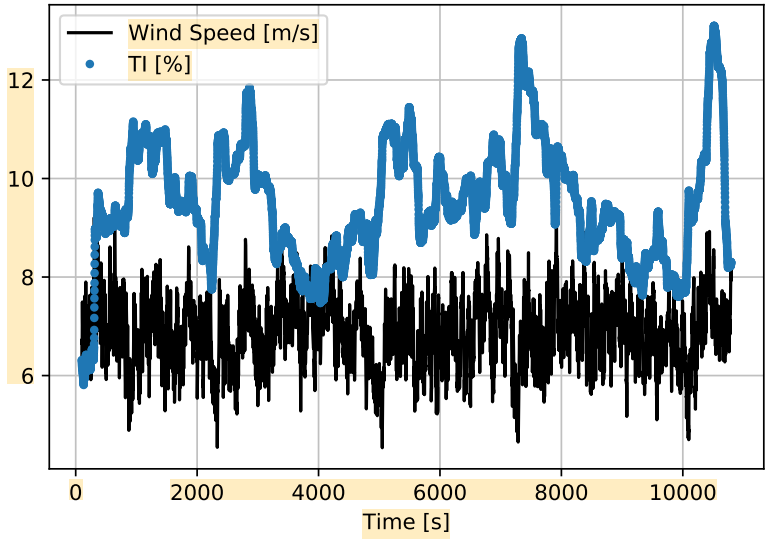


Figure 4. First LSTM model training input time series generated by HAWC2, down-sampled to 1-Hz. Mean Wind Speed = 7 m/s, Mean TI = 10%.

In order to have an adequate quantity of training samples while assuring a representative test dataset for hyper-parameter tuning, the 3-hour period of training and validation signals are split as $80\% - 20\%$, respectively. Since the target available power output is 1-Hz, the final model needs to be able to handle high frequency dynamics in the inflow and successfully map it to the produced power in normal operation, by taking the inertia into account. Given the complexity the model is required to manage, the minimum number of neurons per layer is kept at 50 where 2-3 hidden layers are evaluated as candidate architectures. Table 1 compares the performance of different network configurations on validation data, both for more traditional FFNN perceptrons and LSTM neurons with lag =29s. It shows overall higher performance for LSTM configurations, indicating added value of using neurons with memory capabilities. In fact, LSTM is shown to outperform also more modern architectures such as Extreme Learning-Machine (ELM) (Saini et al., 2020) and Gaussian mixture models (GMMs) (Zhang et al., 2019) for short-term forecasting. Given the best overall performance, the final network has 3 hidden layers with 100, 100 and 50 LSTM neurons, as detailed in Table 2. The hyperbolic tangent function, tanh, is used as the activation function in between the layers. With the listed input structure, the final architecture corresponds to approximately 7 times more data than the trainable parameters, slightly less than the general rule of thumb to avoid overfitting, hence even higher number of neurons are avoided. Nevertheless, the training history in Figure 5 and the model performance on the validation dataset do not indicate a clear overfit, increasing confidence to the training. For the mean absolute error as the loss function, the training history on the very first epoch for validation data shows 'too good' performance of the initial fit. However, since it clearly does not indicate an overall higher accuracy, the network is trained further to its fuller potential, where the validation loss is expectedly lower and convergence is achieved around 100 epochs.

Figure 5. Training history of the First LSTM network with mean wind speed = 7 m/s, mean TI = 10%, time lag = 29s. The network architecture is presented in Table 2, where batch size = 60 with adam optimisation algorithm and mean absolute error (MAE) loss function implemented. The training is stopped at epoch=100, after which the network training starts to show symptoms of overfit.

Architecture	# of neurons			μ_{err} [%]	σ_{err} [%]
	layer 1	layer 2	layer 3		
FFNN	50	50	-	3.1	13.4
LSTM				0.8	10.1
FFNN	100	100	-	1.1	13.3
LSTM				0.7	10.1
FFNN	50	50	50	3.0	13.6
LSTM				0.5	10.0
FFNN	50	100	50	2.0	13.5
LSTM				0.3	10.1
FFNN	100	100	50	1.8	13.7
LSTM				0.3	9.9

Table 1. Representative grid search for best architecture of the First network using Feed Forward Neural Networks (FFNN) and LSTM with lag = 29s and tanh activation function in between the hidden layers. Both FFNN and LSTM trained using adam optimisation algorithm and mean absolute error loss function. The listed percentage error estimation with mean μ_{err} and standard deviation σ_{err} is based on the validation dataset where $\frac{y_{\text{prediction}} - y_{\text{observation}}}{y_{\text{observation}}} \times 100$.

Layer	# of neurons	# of parameters
`lstm_1`	100	41 200
`lstm_2`	100	80 400
`lstm_3`	50	30 200
`Dense_1`	1	51
Total parameters : 151 851		
Trainable parameters : 151 851		
Non-trainable parameters : 0		

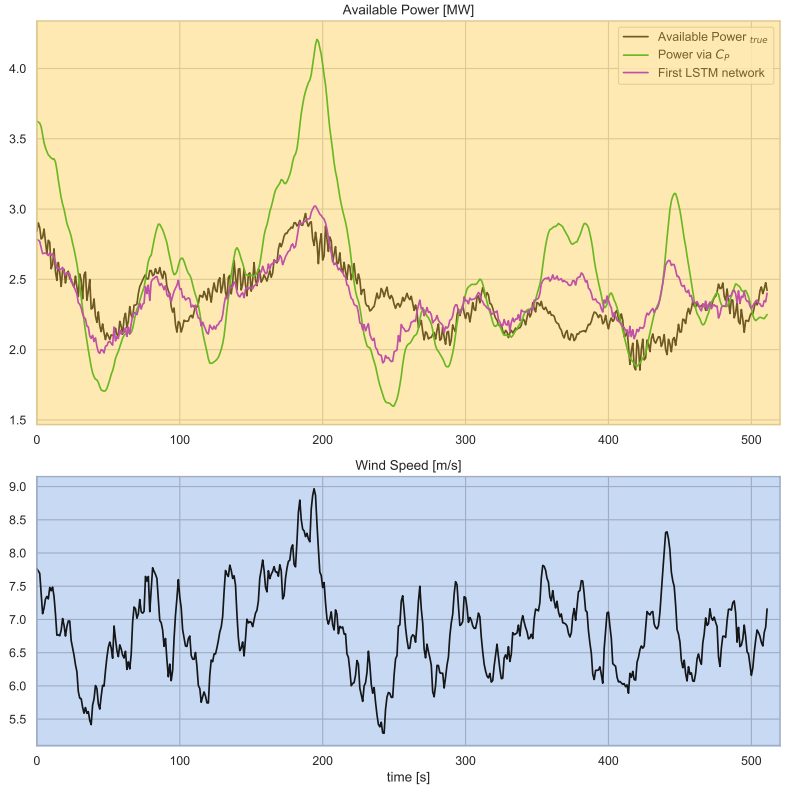
Table 2. Architecture of the First LSTM network with lag = 29s. *Hidden Layers:* `lstm_1`, `lstm_2`, `lstm_3`. *Output Layer:* `Dense_1`. tanh is used as the activation function in between the hidden layers.

The First LSTM network is tested on separate 10-min dataset and compared with the *true* Available Power (actual production of the DTU 10MW turbine under normal operation) as well as the predecessor method of pre-defined C_P look-up tables of the same turbine. The 1-Hz time series and the corresponding 1-second percentage error distribution of the direct C_P look-up table approach and the LSTM model results with lag = 29s are presented in Figure 6. The sensitivity of the mean, μ_{LSTM}, and the standard deviation, σ_{LSTM} to the hindsight horizon up to 89s is listed in Table 3. Due to highest overall performance, the results from LSTM model with lag = 29s will be discussed from now on.

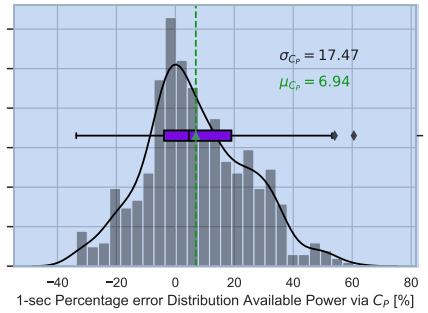
Lag	μ_{LSTM} [%]	σ_{LSTM} [%]
4s	2.52	14.10
9s	1.87	14.67
29s	0.98	8.53
59s	-1.56	12.02
89s	0.48	13.21

Table 3. Sensitivity of the First LSTM model to the hindsight horizon, evaluated based on test dataset.

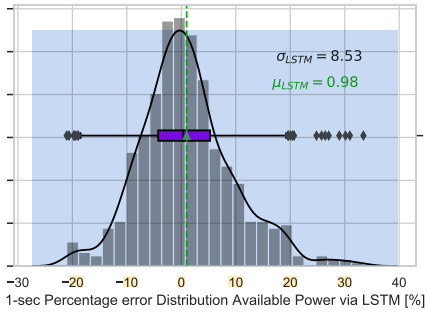
Figure 6 shows that the LSTM model significantly improves the agreement between the actual and the predicted available power production compared to the direct C_P curve interpolation. Since both the trained LSTM model and the C_P curve interpolation approach uses the same input (hub-height wind speed), it can be said that the described deep learning architecture is much more capable of reproducing the dynamic power curve of the turbine than the steady-state C_P surface, even with limited information. For the investigated 10-min period, the bias in the second-wise LSTM available power predictions is less than 1%, as opposed to nearly 7% observed with the direct C_P interpolation approach, where the percentage error is defined as $\frac{\hat{y}(t_i)-y(t_i)}{y(t_i)} \times 100$ with $y(t_i)$ being the power produced by DTU 10MW under normal operation, *i.e.* available power, and $\hat{y}(t_i)$ is the LSTM model prediction at every time step t_i. The standard deviation of the second-wise error distribution, which is regarded as indication of uncertainty in the model results for this study, is also reduced significantly to 8.5%. Note that it is expected to further decrease when the available power prediction is to be delivered at larger time scales (*e.g.* highest frequency being the 1-min minute scale as requested by the German TSOs (50Hertz, Amprion, Tennet, TransnetBW, 2016)). This will be discussed further for larger evaluation periods later in the study.



(a) 1-Hz time series of available power and wind speed for the First test case.



(b) Available power estimation error via direct C_P curve interpolation of wind speed.



(c) Error distribution of the First LSTM model.

Figure 6. Second-wise comparison of the Available Power of the 10MW turbine for mean wind speed = 7 m/s and mean TI = 10% flow case represented in Figure 4. μ and σ are the mean and the standard deviation of the 1-Hz percentage error distributions for direct C_P curve interpolation approach and the First LSTM model with lag=29s.

3.1.2 Training of the Second LSTM Model : High Wind Speed, High Turbulence Intensity

One of the most crucial challenges of purely data driven models is the fact that they are not valid for the input variables outside the training domain, also referred as generalisation problem. As seen in Figure 4 the First LSTM model is trained for mean wind speed 7 m/s, where the turbulent fluctuations occasionally reach above 8 m/s. However, for higher wind speeds *e.g.* around 9 m/s, the First LSTM model is expected to perform poorly as it has not been taught to map the relationship between wind speed, TI and Power for that inflow.

In order to reduce the effort in hyper-parameter tuning and test the universality of the network architecture for a similar problem, the same configuration as in Table 2 is implemented with the inflow time series presented in Figure 7. The final model is referred as the Second LSTM model throughout this study.

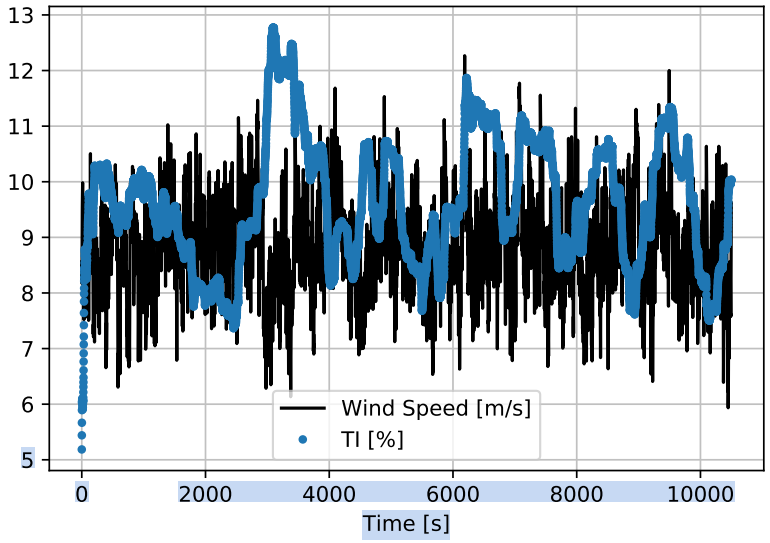
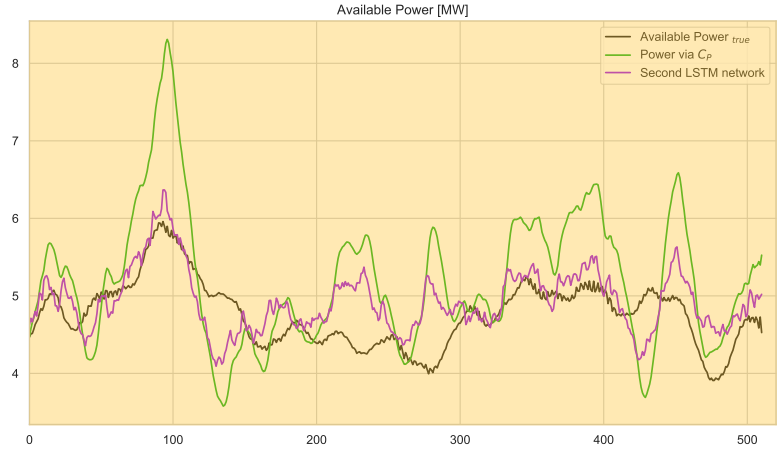


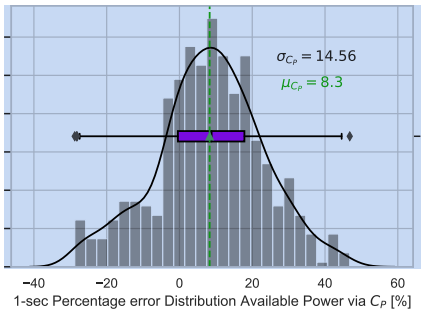
Figure 7. Second LSTM model training input time series generated by HAWC2, down-sampled to 1-Hz. Mean Wind Speed = 9 m/s, Mean TI = 10%.

The performance of the Second model is evaluated based on an independent 10min series with similar mean wind speed and TI and compared with the direct C_P interpolation approach. Similar to the First LSTM model, the test results are presented for lag = 29 s in Figure 8.

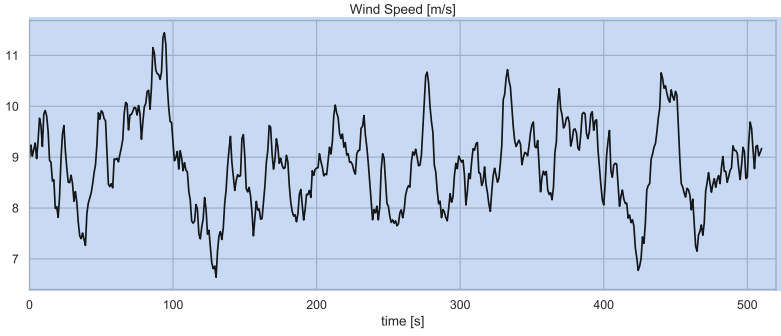
Lag	μ_{LSTM} [%]	σ_{LSTM} [%]
4s	3.76	13.45
9s	3.85	12.83
29s	3.83	8.3
59s	3.15	11.09
89s	3.51	12.14

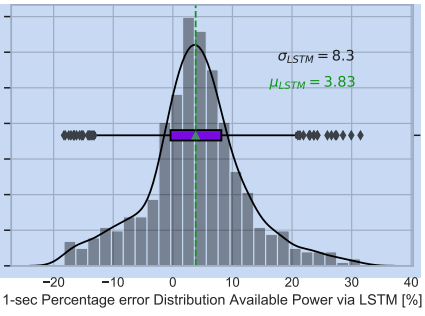
Table 4. Sensitivity of the Second LSTM model to the hindsight horizon.





(b) Available power estimation error via direct C_P curve interpolation of wind speed.





(a) 1-Hz time series of available power and wind speed for the Second test case.

(c) Error distribution of the Second LSTM model.

Figure 8. 1-sec comparison of the Available Power of the 10MW turbine for mean wind speed = 9 m/s and mean TI = 10% flow case represented in Figure 7. μ and σ are the mean and the standard deviation of the 1-Hz percentage error distributions for direct C_P curve interpolation approach and the Second LSTM model with lag=29s.

Despite the significant performance improvement achieved for 7 m/s case with the First LSTM observed in Figure 6, the Second LSTM network developed using the same procedure for 9 m/s inflow has a considerable bias of more than 3% as seen in the 1-Hz percentage error distribution in Figure 8. The mean of the test error seems to be hardly affected by the changing hindsight horizon listed in Table 4, where the standard deviation is the least at lag = 29s. This clearly implies that the architecture and the hyper-parameters optimised for the lower wind speed are not necessarily the best configuration for slightly higher wind speed cases. That trend makes it challenging to develop a generic network architecture that would successfully reproduce the high frequency available power for all the possible input realisations. It indicates the need to specifically tune the hyper-parameters for each separate flow case. It is a cumbersome process with high computational cost. Here in this study, the focus is to make the best out of the available dataset, as indicated earlier, as the generation (or collection) of a comprehensive database is a very demanding task for high frequency problems. Additionally, the observed reduction in performance of the

same hyper-parameter space for a different flow case indicates the risk of the approach where a singular 'generic' model is fit to estimate the high frequency available power for a variety of inflow cases. In other words, a single model to cover the entire domain might introduce compromises in the model performance at certain inflow cases, where the dynamic accuracy is of utmost importance, as framed by the grid codes.

3.1.3 Transfer Learning from the First Model : High Wind Speed, High Turbulence Intensity

Having trained a well-performing model for first inflow case with 7 m/s mean wind speed and 10% mean TI, the question arises: Can some of the characteristics of the First Model be conveyed to a different flow case to achieve similarly good results? Transfer learning can provide a valuable platform for such model extensions, as it is used to improve a learner from one domain by transferring information from a related domain (Weiss et al., 2016). This enables a systematic model update when new data is available from outside the training domain. Accordingly, part of the First Model with 7 m/s mean wind speed would be transferred to update some of the parameters for higher wind speed. The procedure could be repeated for all the changing wind speed and TI cases, both in HAWC2 platform and field applications.

To assess the transferability of the parameters, the trends of the weights trained for the First (Network_1) and the Second (Network_2) LSTM network is compared in Figure 9. The actual probability seen in the most recent histograms (the lightest shade in the series of distributions) are different for all three LSTM layers, with larger tails on Network_1 distributions. However, the range of values for the output weights of the first layer `lstm_1` and the second layer `lstm_2` are very similar, with interquartile range -0.05 < IQR < 0.05 for both. On the other hand, the third and shallower layer `lstm_3` seem to optimise for significantly different weights for different inflow velocities. Therefore, it is concluded that the first 2 LSTM layers are transferable from the First LSTM Network, where the last LSTM layer as well as the output layer need to be re-tuned for changing inflow case(s). The resulting architecture is presented in Table 5 where the number of trainable parameters are significantly reduced. Accordingly, the transferred architecture is a much lighter network that ensures fast training, while enclosing a profound amount of information from previous learning(s). Less number of parameters also enables a robust training with shorter time series. Hence, for the training of the transfer learning LSTM architecture, 60% of the dataset (Second inflow case, presented in Figure 7) is fed to the network, where 40% is left for validation to ensure a more definitive assessment of the training.

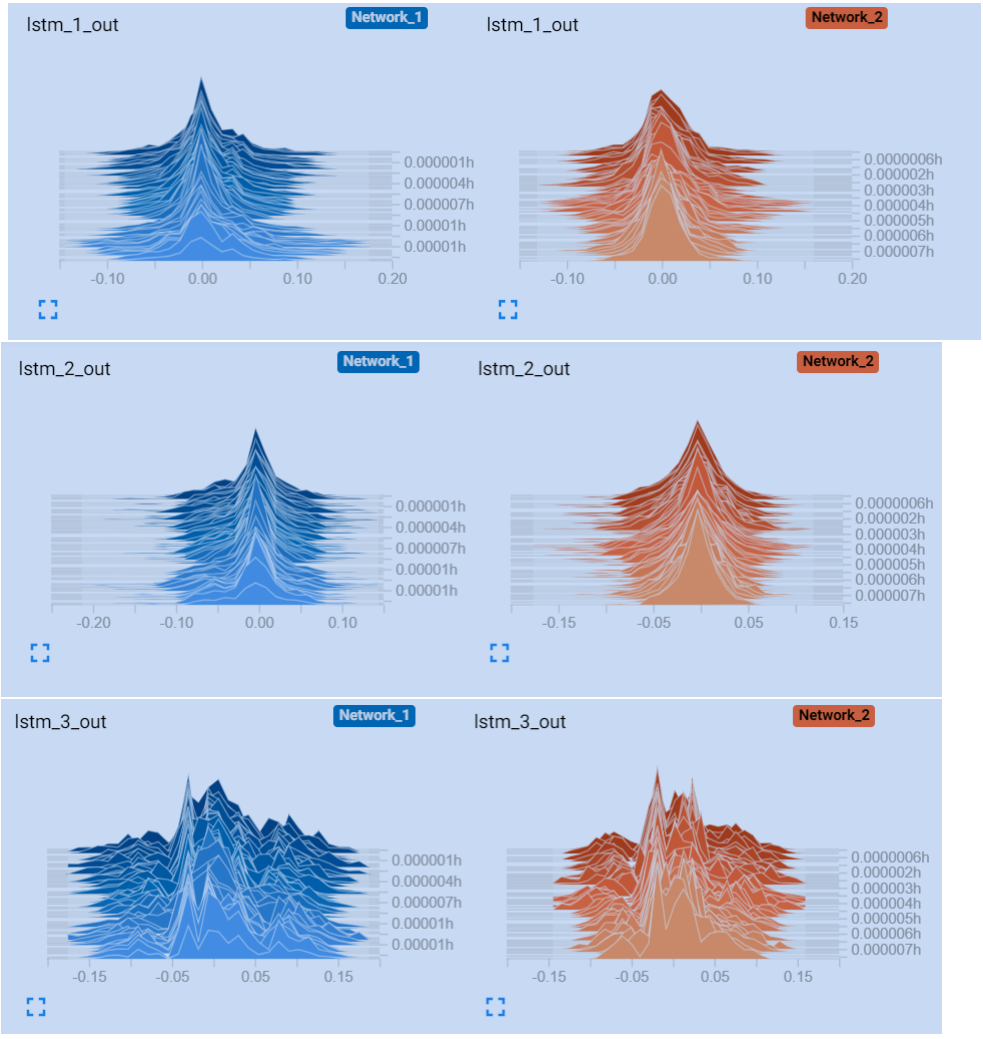


Figure 9. Distribution of the output tensors of each hidden LSTM layers for the First Network_1 and the Second Network_2 LSTM networks, visualised via TensorBoard. Each slice displays a single histogram updated at each iteration. The 'oldest' iterations are further back and darker, while the 'newer' ones are lighter and closer to the front. The y-axis indicates the relative time of each update.

Apart from update of the weights in the last LSTM and the output layers (`lstm_4` and `Dense_2` in Table 5, respectively), none of the other hyper-parameters were changed in the training process of the Transferred LSTM network. This provides a certain repeatability to the training process, where the last two layers can be updated when a new flow case is encountered by the turbine. It is particularly an important feature for the control implementation as it enables fast online learning and continuous improvement of the model.

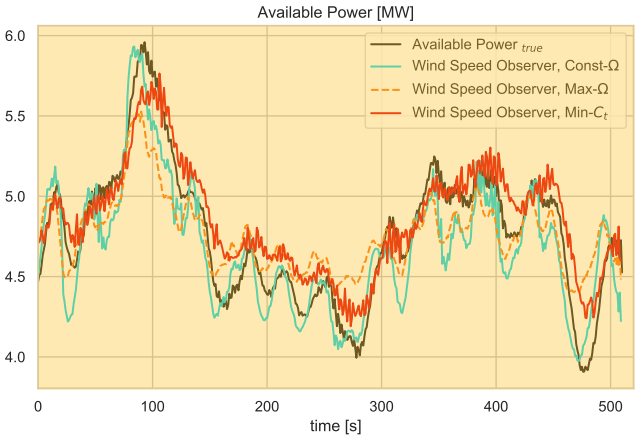
To put the performance of the Transferred LSTM model to test, the same test case as in the Second LSTM model in Figure 8 is considered. This time, the estimations from the operation dependent Wind Observer (WSO) approach (described in Section 2) are also compared with the Transferred LSTM model. The time series in Figure 10(a) illustrates the sensitivity of the WSO

Layer	# of neurons	# of parameters
`lstm_1`	100	41 200
`lstm_2`	100	80 400
`lstm_4`	50	30 200
`Dense_2`	1	51
Total parameters : 151 851		
Trainable parameters : 30 251		
Non-trainable parameters : 121 600		

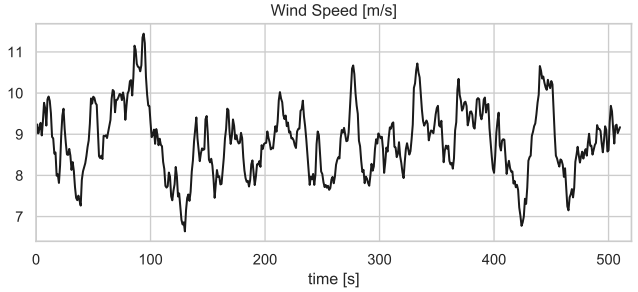
Table 5. Architecture of the Transferred LSTM network for lag = 29s. *Hidden Layers:* `LSTM_1`, `LSTM_2`, `LSTM_4`. *Output Layer:* `Dense_2`. *Frozen Layers:* `LSTM_1`, `LSTM_2` (same layers as in the First LSTM model in Table 2). *Trainable Layers:* `LSTM_4`, `Dense_2`. `tanh` is used as the activation function in both of the activation layers. Training is performed with batch size = 60, adam optimisation algorithm and mean absolute error loss function over epoch = 70.

estimations to the operation strategy under 40% down-regulation with constant rotational speed, Const-Ω, maximum rotational speed, Max-Ω and following the minimum thrust coefficient, Min-C_t. In Figure 11(a), (b) and (c), the error distributions of the WSO estimations under those three operational strategies are presented. While the overall performance of all WSO estimations are highly compelling, the results also indicate up to 4% variation in the mean bias of the WSO model. With the minimum mean error of 0.26%, WSO estimations with maximum rotational speed, Max-Ω, are also compared with the Transferred LSTM model in Figure 10(c). For the investigated setup with DTU 10MW reference turbine model fully recognised, the WSO results generally suggest a better agreement with the *true* available power, quantified in Figure 11(d). However, Figure 10(c) and Figure 12 points out that for a potential mismatch of 5% in the pre-defined and operational C_p surfaces due to several uncertainties listed earlier, model-based WSO results show bias up to more than 6% where the model-free LSTM performance remains unaffected. Note that in these WSO runs, we assume 'perfect knowledge' for normal operation, *i.e.* for maximum C_p, and 5% uncertainty for the rest of the C_p domain. This simulates the field operation where the nominal power curve is corrected for the site conditions (hence 'perfect knowledge') but the information of the rest of the operational C_p remains limited.

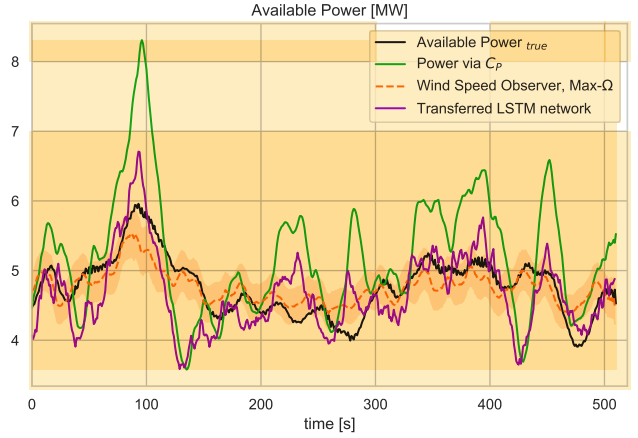
It is also seen that, the Transferred LSTM model outperforms the Second LSTM model where more than 3% model bias is eliminated compared to Figure 8(d). This improvement is very promising for the implementation of the transfer learning for modelling high frequency time series with LSTM networks. Furthermore, the results also show the potential of such a deep learning approach for avoiding the operational dependencies of dynamic delta control with relatively low uncertainties. The adaptation capabilities of transfer learning is to be tested with additional flow cases in the next sections.



(a) Wind Speed Observer (WSO) Available Power Estimation for 40% down-regulation case under 3 different curtailment strategies.

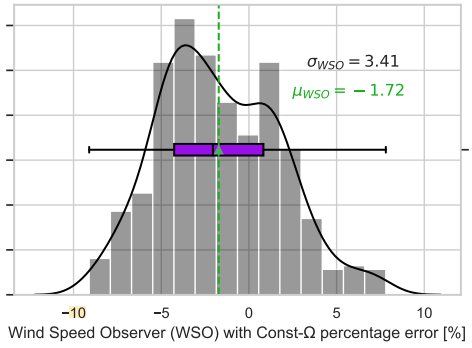


(b) 1-sec wind speed time series for mean wind speed = 9 m/s and mean TI = 10% test case.

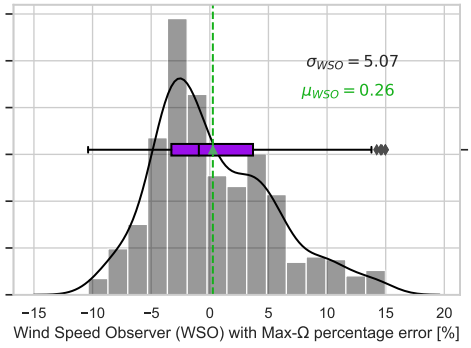


(c) Comparison of WSO Max-Ω and LSTM model results with transfer learning Available Power Estimation. Shaded area corresponds to the effects of $\pm 5\%$ over and under estimation of C_P during curtailment for WSO results. LSTM network has no dependency on level of curtailment or estimation of operational C_P.

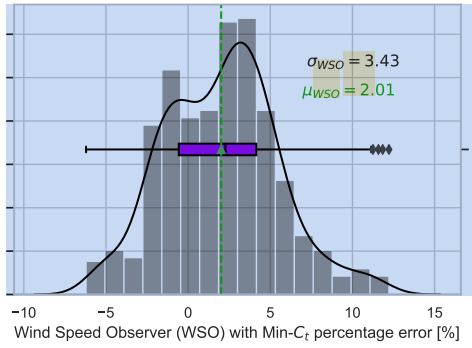
Figure 10. 1-Hz Time Series Comparison of Available Power of DTU 10MW turbine, estimated by (a) the Wind Speed Observer, see Section 2, using 3 different curtailment strategies. (b) Second-wise wind speed time series of the considered test case with mean wind speed = 9M/s and mean TI = 10%. (c) The comparison of the available power estimated via Wind Speed Observer following maximum rotational speed control strategy and LSTM network with transfer learning from lower wind speed to higher wind speed case and direct C_P approach.



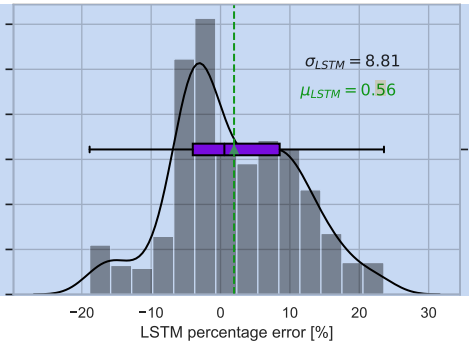
(a) WSO following constant rotational speed strategy



(b) WSO following maximum rotational speed strategy

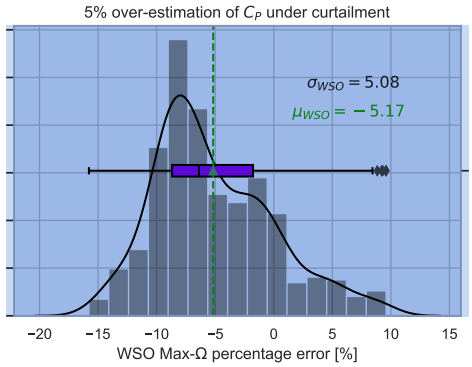


(c) WSO following minimum thrust coefficient strategy

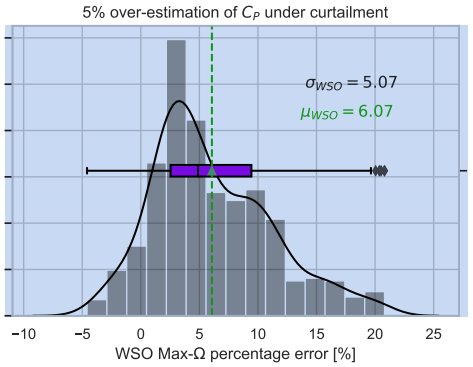


(d) LSTM network with transfer learning

Figure 11. 1-Hz percentage error distribution of the Available Power Estimation of 10MW turbine for mean wind speed = 9 m/s and mean TI = 10%, flow case represented in Figure 7. The presented performances belong to Wind Speed Observer approach in Section 2 different operational strategies. (a) Constant rotational speed, Const.-Ω, (b) Maximum rotational speed, Max-Ω, (c) Minimum thrust coefficient, Min-C_t as 40% curtailment strategy; (d) LSTM model with transfer learning from lower wind speed to higher wind speed case (no dependency on the curtailment strategy).



(a) 1-Hz percentage error of WSO with Max-Ω: 5& under-estimation of C_P during curtailment



(b) 1-Hz percentage error of WSO with Max-Ω: 5& over-estimation of C_P during curtailment

Figure 12. Sensitivity of Wind Speed Observer to correct assessment of C_P under curtailment. The simulations assumes 'perfect knowledge' of C_P for the normal operation, and 5% uniform uncertainty for the rest of the operational range.

3.1.4 Further Transfer Learning to Higher Wind Speed Flows

With the comparable results of the model-free transfer learning LSTM networks to the model-dependent WSO approach, even with potentially lower uncertainties in the simulation environment compared to the field implementation, here we test the approach for even higher wind speed flows. The First LSTM predictions were built and tested on 7m/s mean wind speed, where its information from the first two layers are then transferred to estimate the available power for 9m/s mean wind speed case. Here we further update the LSTM network to extend the training (and validity) domain to 11 m/s mean wind speed range, using the generated time series in Figure 13. Note that for all three steps of the learning, the mean TI remains 10% to isolate the effect wind speed on the network performance.

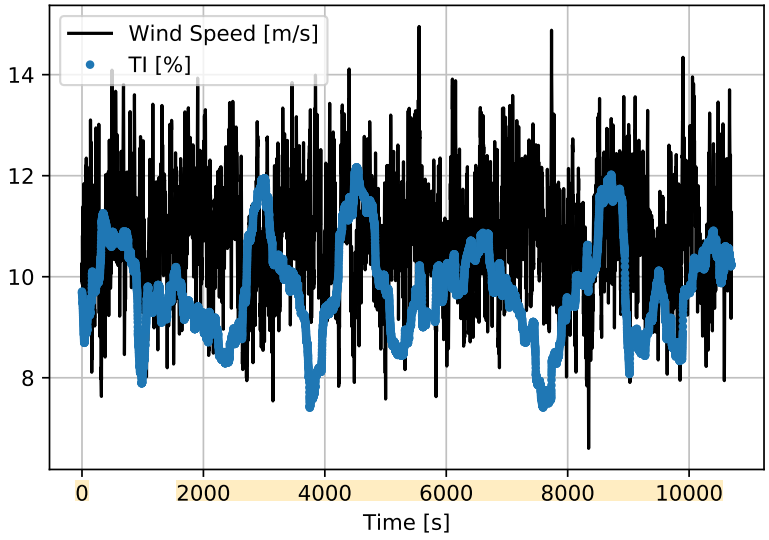
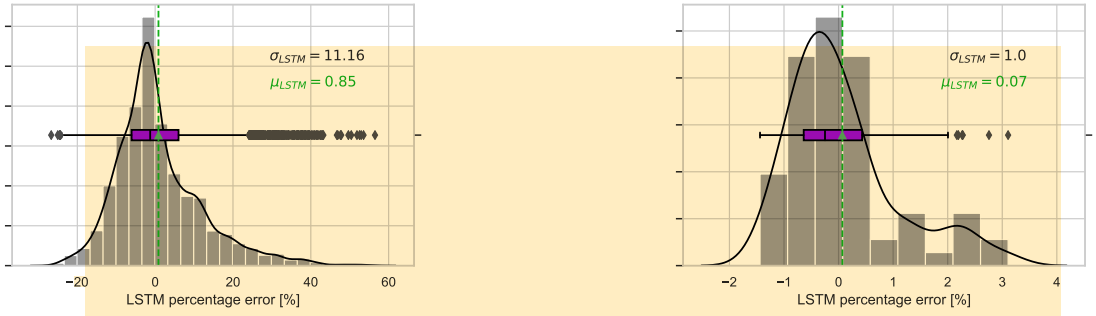


Figure 13. LSTM network training with further transfer learning input time series generated by HAWC2, down-sampled to 1-Hz. Mean Wind Speed = 11 m/s, Mean TI = 10%.



(a) 1-Hz percentage error **(b)** 1-min average percentage error

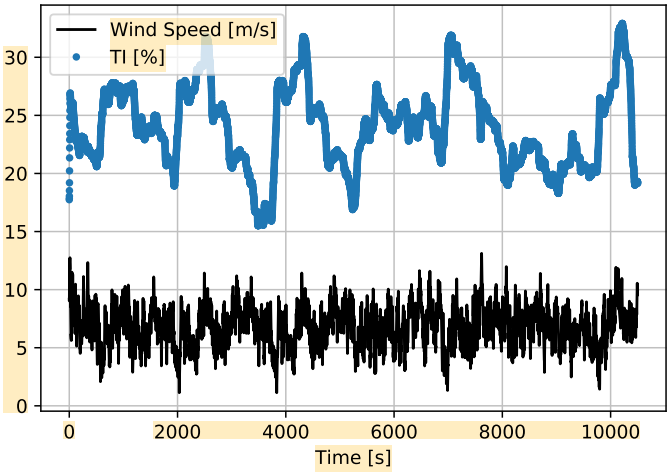
Figure 14. LSTM network prediction error for further transfer learning flow case. Mean Wind Speed = 11m/s and TI = 10% over 60-min validation case. (a) 1-Hz prediction error, (b) 1min average prediction error.

The resulting network with further transfer learning for 11 m/s mean wind speed (in Figure 14(a)) performs similar as in 9 m/s (in Figure 11(d)) and 7m/s (in Figure 6(d)) mean wind speed cases with less than 1% second-wise percentage error on average. Distinctly from the previous inflow cases, the tail towards the positive percentage error is longer in the final 1-Hz error distribution in Figure 14(a). This is mainly due to the fact that DTU 10MW reference turbine (with rated wind speed 11.4 m/s) occasionally enters the rated region according to the turbulent fluctuations around 11 m/s mean wind speed. Nevertheless, the variability of the model prediction error is significantly reduced when averaged for 1-min scale in Figure 14(b). The results

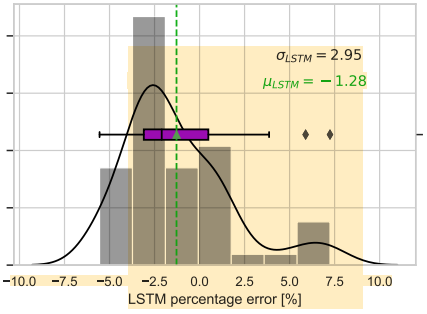
show that the model-free tranfser learning approach easily comply with the strictest TSO requirements in provision of the available power signal; *i.e.* the standard deviation of the 1-min percentage error of the available power is required to be less than $\pm$ 3.3% as stated in (50Hertz, Amprion, Tennet, TransnetBW, 2016).

3.1.5 Network performance on Higher Turbulence Intensity

As stated earlier, for all three inflow cases where the First LSTM network is generated and extended via transfer learning, the mean turbulence intensity remained TI = 10%. Here in this section, the models are tested under higher turbulence intensity (TI = 20%) with the same corresponding mean wind speed. Note that the generated network structures, *i.e.* the First LSTM model (7 m/s mean inflow speed), transfer learning LSTM model (9 m/s mean inflow speed) and further transfer learning LSTM model (11 m/s mean inflow speed), are not updated for higher TI cases. In other words, here we aim to test the capability of the trained networks under higher TI with the same mean wind speed for the inflow, without further model update.

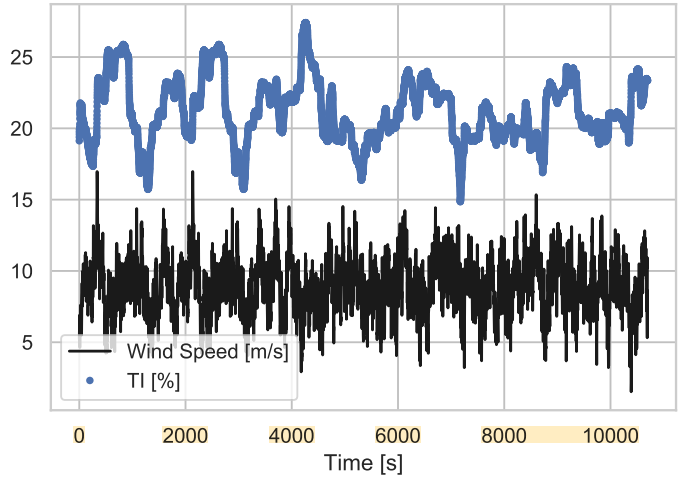


(a) Time series of the inflow dataset for higher TI test of First LSTM network, originally trained on Mean Wind Speed = 7 m/s and TI = 10%.

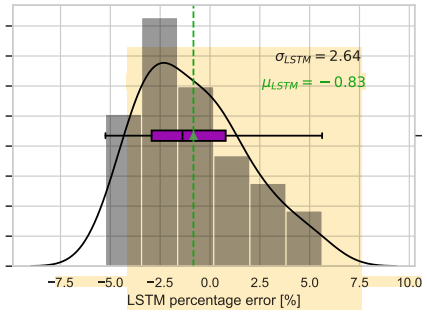


(b) 1-min average percentage error of the First LSTM network tested under higher TI

Figure 15. Input time series and 1-min prediction error of the First LSTM network, higher turbulence intensity flow case. Mean Wind Speed = 7 m/s and TI = 20% over 60-min validation case.

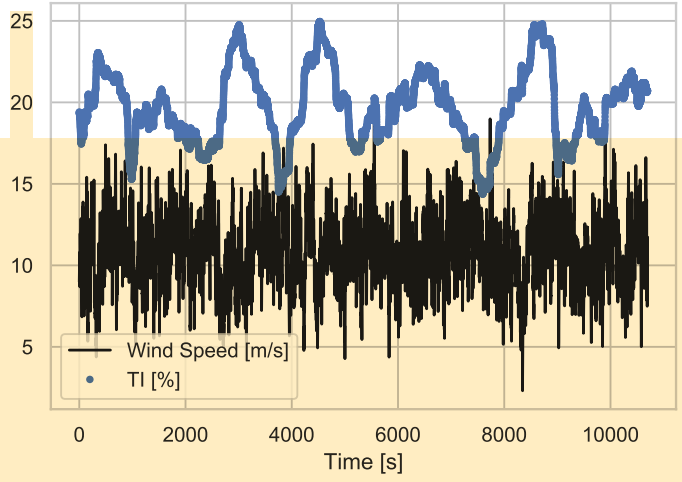


(a) Time series of the inflow dataset for higher TI test of the transfer learning LSTM network, originally trained for Mean Wind Speed = 9 m/s and TI = 10%.

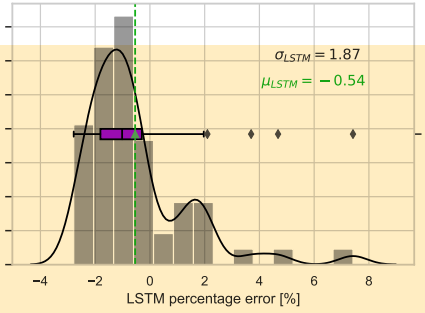


(b) 1-min average percentage error of the transfer learning LSTM model under higher TI.

Figure 16. Input time series and 1-min prediction error of the transfer learning LSTM network, higher turbulence intensity flow case. Mean Wind Speed = 9 m/s and TI = 20% over 60-min test case.



(a) Time series of the inflow dataset for higher TI test of the further transfer learning LSTM model, originally trained for Mean Wind Speed = 11 m/s and TI = 10%.



(b) 1-min average percentage error of the further transfer learning LSTM model under higher TI.

Figure 17. Input time series and 1-min prediction error of the further transfer learning LSTM model, higher turbulence intensity flow case. Mean Wind Speed = 11 m/s and TI = 20% over 60-min test case.

Figures 15, 16 and 17 focus on highly turbulent inflow cases (TI = 20%) and show the corresponding performance of the three neural networks trained and updated for increasing wind speeds via transfer learning. It is seen that the 1-min average prediction errors of the models are consistently low for highly turbulent flows as well; hence further training (or model update) is not required. The maximum standard deviation of 1-min averaged percentage error is still less than 3.3% with the largest bias of 1.3% which is slightly worse than the test results in the original domain of the networks. For the mean wind speeds of 7 and 9 m/s, the effect of higher turbulence levels is clearer as increasing fluctuations in wind speed is directly correlated to higher variance in power output. However for 11 m/s case, the fluctuations are partially dampened due to the turbine entering into the rated region. Overall, it can be said that the sensitivity of the networks to changing wind speed is much higher than the turbulence, and the updates are to be implemented solely based on the altering inflow velocity which is likely to reflect a different operational region. The available power prediction of the described LSTM architecture and the updating scheme with 2 m/s wind speed increase (7 m/s – 9 m/s – 11 m/s) is shown to comply with the strictest grid code requirements under different turbulence realisations.

4 Conclusions

The dynamic estimation of available power of a wind turbine is essential both for power system stability and marketability of the reserve power. The current estimations are highly sensitive to the down-regulation strategy and prone to turbine model uncertainties and inadequacies. Here we propose a purely data-driven, model-free methodology based on Long Short Term Memory (LSTM) neural networks. This state-of-the-art deep learning architecture is implemented to map the available power of DTU 10MW reference turbine under turbulent inflow generated in HAWC2. The trained networks are adapted to the changes in incoming mean wind speed via transfer learning, where the parameters only in the last layer are updated when the new inflow information is available.

The First LSTM network has 3 hidden layers with 100, 100 and 50 neurons respectively, which is trained using 1-Hz power output under normal operation with 7 m/s mean wind speed and 10% turbulence intensity (TI). Performed test on a separate 10-min flow case with the same mean wind speed and TI shows less than 1% bias and less than 9% standard deviation. Same architecture is used to train the Second LSTM network with increase in mean wind speed to 9 m/s and same TI level of 10%. Although the width of the distribution is similar, the bias has increased to almost 4%, indicating the need to re-tune the hyperparameters of the architecture. In fact, the comparison of the fitted parameters between the First LSTM and the Second LSTM networks for each layer shows analogous distributions of the weights. This further motivates the transferability of the learnings of the first two LSTM layers, where only the parameters of the last layer need to be updated for the changing incoming mean wind speed. With a significant reduction in the number of parameters to fit, the Transferred LSTM network has the capability of faster and more robust training, even with limited data. The performance of the Transferred LSTM network is also evaluated using a separate 10-min time series with 9 m/s mean wind speed and 10% TI. The results are very comparable with the outcome of the First LSTM model, which demonstrates the adaptability of the network to changing inflow conditions with the update of the last LSTM layer. The Transferred LSTM also outperforms the Second LSTM network with a significant

decrease in bias (around 0.5%), eliminating the need to re-tune the hyperparameters or developing a new network structure from scratch.

The Transferred LSTM network is also compared with the model and operation dependent Wind Speed Observer (WSO) approach. For the investigated setup where the DTU 10MW reference turbine model is fully transparent or known, the WSO results generally suggest a better agreement with narrower 1-Hz percentage error distributions. However, the sensitivity of the WSO approach to the curtailment strategy is also clearly seen as the results indicate up to 4% variation in the mean bias of the WSO model. The uncertainty of the approach is expected to grow further under the field conditions where there is a potential lack of detailed information regarding the operation strategy and manufacturer-calibrated power coefficients which are generally unable to account for variability influenced by local conditions. To test that hypothesis, 5% uniform uncertainty is introduced to the C_P surface under curtailment for the same evaluation period. Even for the best performing model based estimation under maximum rotational speed control strategy, the model bias significantly increased, risking both under-estimation (bias < -5%) and over-estimation (bias > 6%) of the available power for the assigned C_P uncertainty.

To ensure the applicability of the transfer learning to several inflow cases, the approach is tested for even higher wind speed flows. Further Transferred LSTM network is trained only to update the last LSTM layer with 11 m/s mean wind speed and 10% TI case, where the first two layers are coming from the First LSTM model with 7 m/s mean wind speed validity domain for the same TI. The performance of the Further Transferred network is evaluated within the framework of strict grid requirements, where the quality of the available power signal is to be assessed at 1-min intervals with required accuracy of less than 3.3% standard deviation of the error distribution. Corresponding 1-min average percentage error of the Further Transferred network indicates easy compliance with the regulations, with both bias and standard deviation less than 1%. Similar agreement is observed when all the networks (*i.e.* First LSTM with 7 m/s wind speed, Transferred LSTM with 9 m/s wind speed and Further Transferred LSTM with 11 m/s wind speed) are tested under higher TI of 20%, indicating the robustness of the developed algorithm.

Finally, it should be noted that the neural networks with transfer learning ability used in this study can easily be implemented in operating wind turbines in the field. The second-wise wind speed input to the approach can be provided either from the standard nacelle anemometers or additional sensors such as meteorological masts or remote sensors (*e.g.* lidars, radars, etc.), however, associated input uncertainties should be handled carefully. This study is a conceptual evidence that well-trained neural networks can be applied to determine the set-point for implementing delta control, or to assess the level of reserves when using balance control, even with limited information in the field conditions. The transferability of the network adds the ability for online learning which ensures the continuous improvement of the model-free available power estimation.

The networks developed in this study can be extended to forecast applications, where the input that is read throughout the hindsight horizon (*e.g.* 29-seconds for the cases presented here) is used to predict the available power in forecast horizon longer than 1-second (*e.g.* 1-minute ahead). Similarly, the approach can be implemented to several turbines within the wind farm. For this configuration, the wind direction should be defined as an additional input to take the correlations of local wake effects and power into account. Finally, the neural network algorithm can be updated given the developments within the deep learning research over time. The advancements in the sequential processing (*e.g.* convolutional LSTMs where the internal

matrix multiplications are exchanged with convolution operations, gated recurrent units (GRUs) where the 3-gated LSTMs are 'simplified' with an update and a reset gate, etc.) can easily be utilised when beneficial, keeping the approach up-to-date.

Code and data availability. Both the data and the script to train the networks (First LSTM and Transfer Learning LSTM networks) can be accessed at https://gitlab.windenergy.dtu.dk/tuhf/deep-learning-for-available-power-estimation

Author contributions. The LSTM network training as well as the transfer learning implementation is performed by Tuhfe Göçmen. The need for such an approach is underlined by Albert Meseguer Urban who also generated the time series in DTU 10MW turbine model in HAWC2. Jaime Liew also contributed in defining the research gaps in the field and developed the implementation cases for LSTM network as the set-point for curtailment in HAWC2 controller, therefore also ensured the generalizability and validity of the code. Alan Wai Hou Lio contributed with Kalman filter-based wind speed estimator and the sensitivities of the current methods for available power estimation at the single turbine level.

Competing interests. The authors declare that they have no competing interest.

Acknowledgements. The study is partially supported by the CONCERT Project (Project no. 2016-1-12396), funded by Energinet.dk under the Public Service Obligation (PSO), by TotalControl (Project no. 727-680) within the Horizon2020 framework funded by the European Union and the PowerKey Project (Project no. 64017-0045) under the EUDP Program.

References

50Hertz, Amprion, Tennet, TransnetBW: Leitfaden zur Präqualifikation von Windenergieanlagen zur Erbringung von Minutenreserveleistung im Rahmen einer Pilotphase / Guidelines for Prequalification of Wind parks to provide Minutenreserveleistung (MRL) during a Pilot Phase, Tech. rep., German Tranmission System Operators, 2016.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., and Brain, G.: TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), pp. 265–284, https://doi.org/10.1038/nn.3331, https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi, 2016.

Annoni, J., Taylor, T., Bay, C., Johnson, K., Pao, L., Fleming, P., and Dykes, K.: Sparse-Sensor Placement for Wind Farm Control, Journal of Physics: Conference Series, 1037, 032 019, https://doi.org/10.1088/1742-6596/1037/3/032019, http://stacks.iop.org/1742-6596/1037/i=3/a=032019?key=crossref.54a383a172b2c10e7c4be1e4734abc21, 2018.

Attya, A., Dominguez-Garcia, J., and Anaya-Lara, O.: A review on frequency support provision by wind power plants: Current and future challenges, Renewable and Sustainable Energy Reviews, 81, 2071 – 2087, https://doi.org/https://doi.org/10.1016/j.rser.2017.06.016, http://www.sciencedirect.com/science/article/pii/S1364032117309553, 2018.

Bak, C., Bitsche, R., Yde, A., Kim, T., Hansen, M. H., Zahle, F., Gaunaa, M., Blasques, J., Døssing, M., Heinen, J. J. W., and Behrens, T.: Light rotor: The 10-MW Reference Wind Turbine, in: European Wind Energy Conference and Exhibition 2012, EWEC 2012, 2012.

Bali, V., Kumar, A., and Gangwar, S.: Deep learning based wind speed forecasting- A review, in: Proceedings of the 9th International Conference On Cloud Computing, Data Science and Engineering, Confluence 2019, https://doi.org/10.1109/CONFLUENCE.2019.8776923, 2019.

Bandi, M. and Apt, J.: Variability of the Wind Turbine Power Curve, Applied Sciences, 6, 262, https://doi.org/10.3390/app6090262, http://www.mdpi.com/2076-3417/6/9/262, 2016.

Bhowmik, S., Spee, R., and Enslin, J. H.: Performance optimization for doubly-fed wind power generation systems, in: Conference Record of 1998 IEEE Industry Applications Conference. Thirty-Third IAS Annual Meeting (Cat. No. 98CH36242), vol. 3, pp. 2387–2394, IEEE, 1998.

Chen, J., Zeng, G. Q., Zhou, W., Du, W., and Lu, K. D.: Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization, Energy Conversion and Management, https://doi.org/10.1016/j.enconman.2018.03.098, 2018.

Chinmoy, L., Iniyan, S., and Goic, R.: Modeling wind power investments, policies and social benefits for deregulated electricity market – A review, https://doi.org/10.1016/j.apenergy.2019.03.088, 2019.

et al., B.: Description of the DTU 10 MW Reference Wind Turbine, DTU Wind Energy Report, https://doi.org/10.1017/CBO9781107415324.004, 2013.

Fleming, P. A., Aho, J., Buckspan, A., Ela, E., Zhang, Y., Gevorgian, V., Scholbrock, A., Pao, L., and Damiani, R.: Effects of power reserve control on wind turbine structural loading, Wind Energy, 19, 453–469, 2016.

Gers, F. A. and Schmidhuber, E.: LSTM recurrent networks learn simple context-free and context-sensitive languages, IEEE Transactions on Neural Networks, 12, 1333–1340, 2001.

Gers, F. A., Schmidhuber, J., and Cummins, F.: Learning to Forget: Continual Prediction with LSTM, Neural Computation, 12, 2451–2471, https://doi.org/10.1162/089976600300015015, http://www.mitpressjournals.org/doi/10.1162/089976600300015015, 2000.

Ghaderi, A., Sanandaji, B. M., and Ghaderi, F.: Deep Forecast: Deep Learning-based Spatio-Temporal Forecasting, 2017.

Göçmen, T. and Giebel, G.: Data-driven Wake Modelling for Reduced Uncertainties in short-term Possible Power Estimation, Journal of Physics: Conference Series, 1037, 072 002, https://doi.org/10.1088/1742-6596/1037/7/072002, http://stacks.iop.org/1742-6596/1037/i=7/a=072002?key=crossref.1608a74f2a2e84417d966bdc82644c2d, 2018.

Göçmen, T., Giebel, G., Poulsen, N. K., and Mirzaei, M.: Wind speed estimation and parametrization of wake models for downregulated offshore wind farms within the scope of PossPOW project, Journal of Physics: Conference Series, 524, https://doi.org/10.1088/1742-6596/524/1/012156, 2014.

Göçmen, T., Giebel, G., Réthoré, P.-E., and Murcia Leon, J. P.: Uncertainty Quantification of the Real-Time Reserves for Offshore Wind Power Plants, 15th Wind Integration Workshop, 2016.

Göçmen, T., Giebel, G., Poulsen, N. K., and Sørensen, P. E.: Possible power of down-regulated offshore wind power plants: The PossPOW algorithm, Wind Energy, https://doi.org/10.1002/we.2279, 2019.

Hansen, A. D., Sørensen, P., Iov, F., and Blaabjerg, F.: Grid support of a wind farm with active stall wind turbines and AC grid connection, Wind Energy, 9, 341–359, 2006.

Henriksen, L., Hansen, M., and Poulsen, N.: A simplified dynamic inflow model and its effect on the performance of free mean wind speed estimation, Wind Energy, 17, n/a–n/a, https://doi.org/10.1002/we.1548, http://onlinelibrary.wiley.com/doi/10.1002/we.1608/fullhttp://doi.wiley.com/10.1002/we.1548, 2012.

Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, Neural Computation, https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

Jena, D. and Rajendran, S.: A review of estimation of effective wind speed based control of wind turbines, Renewable and Sustainable Energy Reviews, 43, 1046–1062, 2015.

Jin, T. and Tian, Z.: Uncertainty analysis for wind energy production with dynamic power curves, in: 2010 IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems, PMAPS 2010, https://doi.org/10.1109/PMAPS.2010.5528405, 2010.

Knudsen, T., Bak, T., and Soltani, M.: Prediction models for wind speed at turbine locations in a wind farm, Wind Energy, 14, 877–894, https://doi.org/10.1002/we.491, 2011.

Kristoffersen, J.: The Horns Rev Wind Farm and the Operational Experience with the Wind Farm Main Controller, Proceedings of the Copenhagen Offshore Wind, 2005.

Lange, M.: On the uncertainty of wind power predictions - Analysis of the forecast accuracy and statistical distribution of errors, Journal of Solar Energy Engineering, Transactions of the ASME, https://doi.org/10.1115/1.1862266, 2005.

Lio, W. H.: Blade-Pitch Control for Wind Turbine Load Reductions, Springer Theses, Springer International Publishing, https://doi.org/10.1007/978-3-319-75532-8, http://link.springer.com/10.1007/978-3-319-75532-8, 2018.

Lio, W. H., Mirzaei, M., and Larsen, G. C.: On wind turbine down-regulation control strategies and rotor speed set-point, Journal of Physics: Conference Series, 1037, 032 040, https://doi.org/10.1088/1742-6596/1037/3/032040, http://stacks.iop.org/1742-6596/1037/i=3/a=032040?key=crossref.011cc406dd4baff3fcf3ee684120d898, 2018.

Lio, W. H., Galinos, C., and Urban, A.: Analysis and design of gain-scheduling blade-pitch controllers for wind turbine down-regulation, in: The 15th IEEE International Conference on Control and Automation (IEEE ICCA 2019), 2019.

Lydia, M., Kumar, S. S., Selvakumar, A. I., and Prem Kumar, G. E.: A comprehensive review on wind turbine power curve modeling techniques, https://doi.org/10.1016/j.rser.2013.10.030, 2014.

Ma, X., Poulsen, N. K., and Bindner, H.: Estimation of Wind Speed in Connection to a Wind Turbine, Tech. rep., Technical report, Technical University of Denmark, Technical report, Technical University of Denmark, 1995.

Manobel, B., Sehnke, F., Lazzús, J. A., Salfate, I., Felder, M., and Montecinos, S.: Wind turbine power curve modeling based on Gaussian Processes and Artificial Neural Networks, Renewable Energy, https://doi.org/10.1016/j.renene.2018.02.081, 2018.

Meng, F., Wenske, J., and Gambier, A.: Wind turbine loads reduction using feedforward feedback collective pitch control based on the estimated effective wind speed, Proceedings of the American Control Conference, 2016-July, 2289–2294, https://doi.org/10.1109/ACC.2016.7525259, 2016.

Mujeeb, S., Alghamdi, T. A., Ullah, S., Fatima, A., Javaid, N., and Saba, T.: Exploiting deep learning for wind power forecasting based on big data analytics, Applied Sciences (Switzerland), https://doi.org/10.3390/app9204417, 2019.

Murcia, J. P., Réthoré, P. E., Dimitrov, N., Natarajan, A., Sørensen, J. D., Graf, P., and Kim, T.: Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates, Renewable Energy, https://doi.org/10.1016/j.renene.2017.07.070, 2018.

Ortega, R., Mancilla-David, F., and Jaramillo, F.: A globally convergent wind speed estimator for windmill systems, Proceedings of the IEEE Conference on Decision and Control, pp. 6079–6084, https://doi.org/10.1109/CDC.2011.6160544, 2011.

Ortega, R., Mancilla-David, F., and Jaramillo, F.: A globally convergent wind speed estimator for wind turbine systems, International Journal of Adaptive Control and Signal Processing, 27, 413–425, https://doi.org/10.1002/acs.2319, http://doi.wiley.com/10.1002/acs.2319, 2013.

Østergaard, K. Z., Brath, P., and Stoustrup, J.: Estimation of effective wind speed, Journal of Physics: Conference Series, 75, 012 082, https://doi.org/10.1088/1742-6596/75/1/012082, http://stacks.iop.org/1742-6596/75/i=1/a=012082?key=crossref.fc593be72e4ad188f70f817ba825c09fhttp://stacks.iop.org/1742-6596/75/i=1/a=012082, 2007.

Ouyang, T., Kusiak, A., and He, Y.: Modeling wind-turbine power curve: A data partitioning and mining approach, Renewable Energy, https://doi.org/10.1016/j.renene.2016.10.032, 2017.

Pan, S. J. and Yang, Q.: A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering, 22, 1345–1359, https://doi.org/10.1109/TKDE.2009.191, http://ieeexplore.ieee.org/document/5288526/, 2010.

Pelletier, F., Masson, C., and Tahan, A.: Wind turbine power curve modelling using artificial neural network, Renewable Energy, https://doi.org/10.1016/j.renene.2015.11.065, 2016.

Pinson, P.: Estimation of the uncertainty in wind power forecasting, Ph.D. thesis, 2006.

Pinson, P., Chevallier, C., and Kariniotakis, G. N.: Trading wind generation from short-term probabilistic forecasts of wind power, IEEE Transactions on Power Systems, https://doi.org/10.1109/TPWRS.2007.901117, 2007.

Rana, M. and Koprinska, I.: Forecasting electricity load with advanced wavelet neural networks, Neurocomputing, 182, 118–132, https://doi.org/10.1016/j.neucom.2015.12.004, https://linkinghub.elsevier.com/retrieve/pii/S0925231215019141, 2016.

Ritter, B., Mora, E., Schlicht, T., Schild, A., and Konigorski, U.: Adaptive Sigma-Point Kalman Filtering for Wind Turbine State and Process Noise Estimation, Journal of Physics: Conference Series, 1037, https://doi.org/10.1088/1742-6596/1037/3/032003, 2018.

Saini, V. K., Kumar, R., Mathur, A., and Saxena, A.: Short term forecasting based on hourly wind speed data using deep learning algorithms, in: 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), pp. 1–6, 2020.

Selvam, K.: Individual Pitch Control for Large scale wind turbines Multivariable control approach, ECN Report, ECN-E-07-053, 2007.

Simley, E. and Pao, L. Y.: Evaluation of a wind speed estimator for effective hub-height and shear components, in: Wind Energy, vol. 19, pp. 167–184, John Wiley and Sons Ltd, https://doi.org/10.1002/we.1817, 2016.

Soltani, M. N., Knudsen, T., Svenstrup, M., Wisniewski, R., Brath, P., Ortega, R., and Johnson, K.: Estimation of rotor effective wind speed: A comparison, IEEE Transactions on Control Systems Technology, 21, 1155–1167, https://doi.org/10.1109/TCST.2013.2260751, 2013.

Stol, K. A. and Balas, M. J.: Periodic Disturbance Accommodating Control for Blade Load Mitigation in Wind Turbines Performance, Journal of solar energy engineering, 125, 379, https://doi.org/10.1115/1.1621672, http://solarenergyengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1456866, 2003.

Thiringer, T. and Petersson, A.: Control of a variable-speed pitch-regulated wind turbine, Dept. of Energy and Environ., Chalmers Univ. of Technol., Göteborg, Sweden, 2005.

van der Hooft, E. and van Engelen, T.: Estimated wind speed feed forward control for wind turbine operation optimisation, in: European Wind Energy Conference2, 2004.

Wang, N., Wright, A. D., and Johnson, K. E.: Independent blade pitch controller design for a three-bladed turbine using disturbance accommodating control, in: 2016 American Control Conference (ACC), pp. 2301–2306, IEEE, https://doi.org/10.1109/ACC.2016.7525261, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7525261, 2016.

Weiss, K., Khoshgoftaar, T. M., and Wang, D.: A survey of transfer learning, Journal of Big Data, 3, 9, https://doi.org/10.1186/s40537-016-0043-6, http://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0043-6, 2016.

Wilches-Bernal, F., Chow, J. H., and Sanchez-Gasca, J. J.: A Fundamental Study of Applying Wind Turbines for Power System Frequency Control, IEEE Transactions on Power Systems, 31, 1496–1505, https://doi.org/10.1109/TPWRS.2015.2433932, 2016.

Zhang, J., Yan, J., Infield, D., Liu, Y., and sang Lien, F.: Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and Gaussian mixture model, Applied Energy, https://doi.org/10.1016/j.apenergy.2019.03.044, 2019.