# EDP Wind Turbine Failure Detection Submission

Marco A. Ferra

## Acknowledgements and Background

I would like to thank EDP and especially the EDP Open Data Team for this challenge. Your support has been top–notch and I'm very pleased to see excellent teams thriving for excellent results. I have discovered the challenge by chance (I believe I saw it on LinkedIn) and started to develop a solution to win the prize, but the more I got involved, the more I learned about wind turbines, machine learning and algorithms. The challenge gradually shifted from a means to an end to the other way around, being the end learning more about data science and less about the prize (but the prize is also great!). I have a background in Electrical and Computer Engineering, not informatics or mathematics, and so my approach has been, from the beginning to the end, to understand the electrical, mechanical and physical aspects of wind turbines, and use the data to understand how to correctly model the physics behind the numbers. In my mind and in my approach the numbers should reflect the reality. If they don't fit, then I don't understand them well enough and should investigate the problem more deeply.

Also, I didn't use any start–of–the–art commercial software tools like MATLAB, simply because I don't have access to such tools. My setup was a 6–year old laptop with Python, NumPy, SciPy, sklearn and TensorFlow. I believe that my programming skills are decent, but my knowledge about machine learning before the challenge was close to nil. This challenge has been pivotal to my understanding of these algorithms.

My approach is supported by many intermediate results, algorithms, code and assumptions. This was also an approach based on trial–and–error. These intermediate results are not presented here. This document is not a single slide presentation, but it's also not an extensive article. I have tried to make this document as succinct as possible, and as a by-product of that, it's also a bit informal. Please feel free to contact me for further information, code and graphs.

## The big picture – Holistic View

I have tried to understand the logs file, and after learning about wind turbines and power curves, I concluded that the wind turbine must be a V90/2.0 MW 50 Hz VCS from Vestas. If it is not, hopefully, their technical characteristics are close enough for my analysis. Very (very) shortly, the wind makes the rotor spin through the blades, the gearbox (GEARBOX) transmits rotational torque from the rotor to the generator (GENERATOR and GENERATOR_BEARING) and the generator is a three-phase asynchronous generator that is connected through a slip ring system that establishes a star and delta configuration automatically. The power generated by the generator is fed to a high–voltage transformer (TRANSFORMER) that connects to the grid. I had difficulties modelling and understanding the hydraulic system, not only because it functions as a cooling mechanism for the various turbine components, but because hydraulics is also used to control the blade pitch system – I'm not sure if the hydraulic fluid is shared between these systems, and I'm not sure about its temperature behaviour among the different systems.

## Data pre-processing

I have tried to merge all the datasets in a structured manner, but the result was not satisfactory. The signals, metmast and failures datasets could be merged into one, but the logs dataset didn't fit – there are multiple equal timestamps with different information about different turbine conditions that didn't fit nicely in an equally interval spaced timestamp in single merged dataset. Also, I didn't find the metmast information useful mainly for two reasons: first, all the turbines are (or perhaps, should) already be equipped with sensors that measure key environmental factors, and second, the malfunction of the turbine sensors only lasted a few hours, so they didn't pose any problems about the truthfulness of data (I hope!).

I have also crossed the logs dataset with the Vestas information manual: the information in the logs is unknown to me, and it's important to know if the information regards proper operating function, a warning or an alarm. My success in this approach has been only average because not all information is available in the Vestas manual and I'm sure I may be missing some crucial information.

## Exploratory Data Analysis

I have tried to visually explore all the signals in the domains of time, through a time series analysis, and occurrence, through a Kernel Density Estimation. Something unexpected has popped up on the temperature profile (HVTrafo_Phase{1,2,3}_Temp_Avg) of the turbine transformers: T06 and T11 seem to belong to a group, T01, T07 and T09 to another. I'm not sure why this happens: perhaps a fan replacement in the cooling system of the transformers that are of a different type?

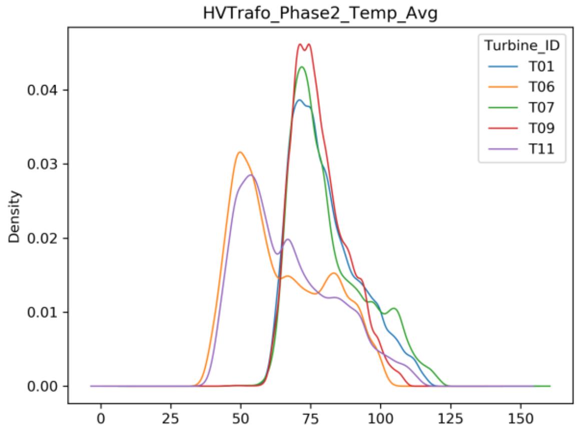

*Figure 1 – KDE of HVTrafo_Phase2_Temp_Avg for all turbines. The other two phases temperatures exhibit similar behaviour.*

Also, the hydraulic oil temperature profile is different between turbines: T07 and T09 show a similar profile but differs from T01, T06 and T11. I still don't know why this happens, and because of that, I have tried not to rely on the Hyd_Oil_Temp_Avg signal – I don't understand their behaviours well enough to be a reliable source of information. All other signals seem to share the same distribution among the turbines.

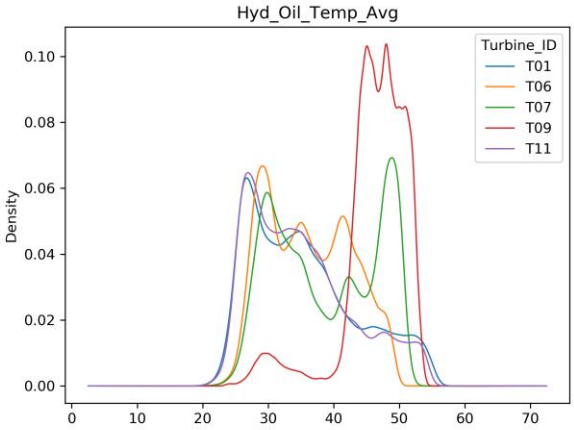

*Figure 2 – KDE of Hyd_Oil_Temp_Avg for all turbines. Turbines T07 and T09 seem to belong to a different group.*

In the time domain, there also seems to be conflicting or missing information. The Amb_WindDir_Relative_Avg don't have useful information after the end of 08–2017, perhaps a sensor error? I didn't rely on this signal to predict any failure.

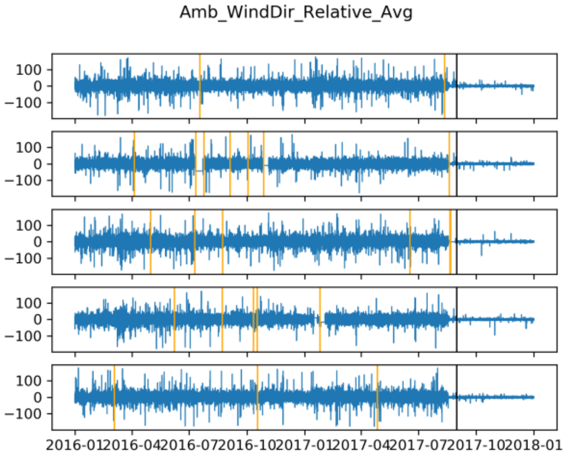

*Figure 3 – Time series of Amb_WindDir_Relative_Avg. Data points with a blue line, failures marked with a vertical orange line, separation between training and testing periods marked with a black vertical line. Apparently, there is missing data after 2017–09–01.*

The Hyd_Oil_Temp_Avg signal also shows a very high–temperature regime in the spring/summer months on turbines T06 and T09 (matches the information from the KDE) but not in the other turbines. Perhaps is a different oil type?

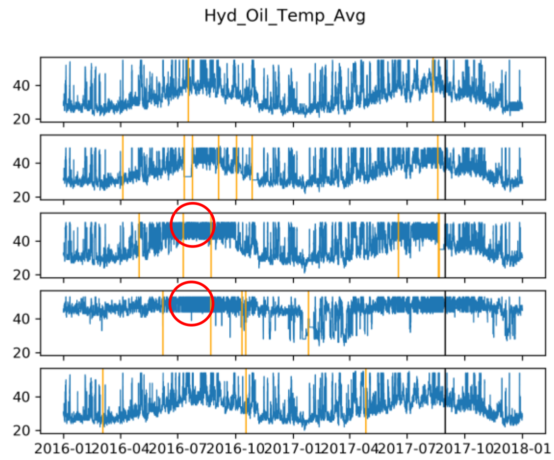

*Figure 4 – Time series of Hyd_Oil_Temp_Avg. Data points with a blue line, failures marked with a vertical orange line, separation between training and testing periods marked with a black vertical line. There are more failures in the summer months, and turbines T06 and T09 exhibit unusual behaviour.*

## First approach – Reliability and Survival Analysis

One of my initial approaches was to use Survival Analysis and I tried to estimate a Weibull distribution to predict when the turbine would fail, but two problems arose: 1) I wasn't able to make a single binary output (failure, no–failure) with the signals data, mainly because there are different components that could fail at different times and 2) the failures list was too short (23 failures) to use this tool. I also tried to use a specific tool from Six Sigma called Root–Cause–Analysis to understand the true nature of the failures, and then use the available signals data to validate the assumption, but many different things could go wrong with each component itself, so this approach didn't seem to get any good results either.

Nonetheless, the weeks that I was thinking about these approaches led me to conclude that there should be two time periods that should be considered: a time when the turbine is working properly, whose signals have values that are within the tolerances defined by the turbine R&D team – I call this the healthy period – and a time period when the turbine is going to fail because a defective system is already operating for some time – I call this unhealthy period. For this, I have seen the KDE and time domain data of the signals and assumed that the turbines should exhibit a healthy condition at least 30 days before a major failure and resume the healthy condition profile 5 days after the component replacement. Because this is a challenge, it's also unknown if the turbine is going to fail in the very first day of the testing period (2017–09–01), so I also subtracted 30 days from this date. In fact, 30 days and 5 days are somewhat arbitrary: after a proper maintenance operation, 5 days seems good enough to start considering the healthy period, but the 30 days was chosen to maximize the data available and minimize considering the unhealthy turbine regime. More would be better (40, 60 days for example), but 30 days seem to fit the available information.

## The second approach – Unhealthy and healthy regimes

If the turbine, in a healthy working period, exhibits a certain profile, then there should be a difference between the true and predict signals just before failure. The next step was to model the turbine behaviour in the healthy period, predict the signals considering that the turbines were always healthy and see a difference between the measured signals – the more the difference, the more likely that the turbine is not working properly and it's going to fail soon.

## Feature Selection and Prediction model

For many weeks I used a (recurrent) neural–network (NN), using TensorFlow and then sklearn. I considered this problem to be a supervised one. I have tried to create a model from a regression point–of–view, and from a classification standpoint. In both cases, the accuracy didn't go above 60%.

The regression approach was considering that the label was the temperature signal directly (a continuous variable); the classification approach was built by transforming the continuous variable into a categorical one (for example, creating temperature ranges such as 40 ºC, 60 ºC, 80 ºC). Unfortunately, 60% or 70% of accuracy for a neural–network model is almost as useless as having a coin tossed in the air: perhaps the failure could be predicted, perhaps not. I also didn't get to tune the hyper-paraments of the NN-model, the base model was simply too inaccurate.

There was also the question of selecting which features should be used to properly model the target. First, I chose them with my engineering background, and then I tried to validate my assumptions by estimate a model with a mathematical tool like Random Forests (RF). However, and it may seem counter-intuitive, although the features selected by the RF are ranked from the variation that they explain in the output, if the Hyd_Oil_Temp_Avg signal showed up I would discard it – my approach from the start was to have reliable data and not to blindly trust the results of the algorithms. The behaviour of the Hyd_Oil_Temp_Avg signal seems too erratic (or so I believe) to use.

Using the RF selected features in the neural–network didn't improve the accuracy by more than 5%. However, using a linear regression did. Although I have coded the neural–network for all predictions, for the challenge I have used a linear regression to make the predictions. The accuracy is much higher, and it's faster to train.

To emphasize the differences between the prediction and the real ground data I used the Mahalanobis distance: the calculation considers the correlations found in the data and so the difference becomes more evident.

With all models created, it’s possible to see if the error between prediction and reality is getting larger, and if it is, then the turbine should fail soon. I inspected the logs files to find validation for my model. For the most part, the logs exhibit information that supports the models, but not in that very specific failure date.

## The challenge submission

Since we are interested in the best maintenance date and not the failure date (up to 60 days), I have made my prediction be around 70% or 80% of the failure date or 42 days before, and these are the results for the GENERATOR component of turbine T06 (the other failures have similar graphs):

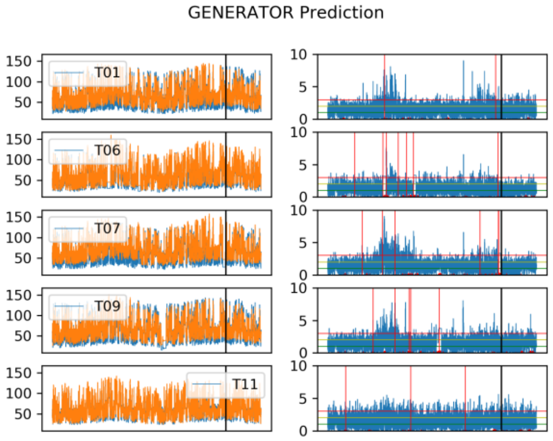

*Figure 5 – On the left side, blue lines are the ground truth and orange lines are the prediction made by the algorithm. On the right side, and in blue, the Mahalanobis distance between the real and predicted values. The red vertical lines are the failures dates, the black vertical line splits the training from the testing period. The three horizontal lines, red, yellow and green, represent threshold boundaries to the Mahalanobis distance: points above the red line have exhibited a larger error.*

Close–up view:

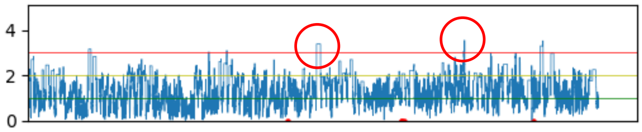

*Figure 6 – Close–up view of the prediction of the failure for T06. The error data is very noisy, but it seems that the turbine will fail at those points in time. The noise arises from the prediction model not being 100% accurate.*

## Conclusions and Final Words

The EDP Wind Turbine Failure Detection Challenge has used many 100ths of hours of my time, and in every single minute, I had a real pleasure in using them. It seems to me that trying to predict turbine failure from SCADA is not only a great challenge (since SCADA data is easily and readily available) but mainly because of the potential gains – instead of fitting an expensive specialized sensor infrastructure, one can use more intelligent engineering and mathematical tools like behaviour models and prediction estimates to learn and understand better the physical behaviour of such systems.

But, it seems to me, using neural-networks is a bit like guesswork – a work that needs further development and understanding. Linear regressions have a physical meaning (they aren’t just some values in the NN hidden layers) and I may or may not have correctly predicted the failures, but it seems that such system needs more intelligence – to be accurate (100% certain) it's needed not only mathematical expertise but also a physical understanding gained by engineering training.

Finally, it has been an absolute delight participating in this challenge and please feel free to contact me for further information, code and graphs. I would love to know where the real failures occurred. And I would like to work closely, in an informal or a formal fashion, with the EDP Open Data Team. So, if you find my work valuable in some way, please get in touch.

If by chance I have made EDP lose money (albeit somewhat virtual) in the challenge, well, I'll keep practising.

Thank you again for the challenge; I have learned so much that it's even hard to describe.

Best regards,

Marco A. Ferra