Training data requirements for SCADA based condition monitoring using artificial neural networks

S Letzgus[a]

[a]Institute of Software Engineering and Theoretical Computer Science, Machine Learning Group, Technical University Berlin, Germany

E-mail: simon.letzgus@tu-berlin.de

Abstract

SCADA data analysis has attracted considerable research interest for monitoring wind turbine condition without additional equipment. Above all, normal behaviour models using artificial neural networks have shown promising results. However, the crucial question of how much training data is actually required to train robust and reliable models has not been addressed in literature so far. In fact, contradictory statements ranging from a few months up to more than a whole year of training data can be found. This paper therefore empirically investigates the relationship between available training data and model performance. A small feed-forward network as well as a larger recurrent network architecture are trained with variable training length and evaluated on a healthy as well as on a turbine with gearbox-problems. It is shown that longer training periods minimize the risk of poor model performance and larger model architectures can be beneficial. Based on these findings at least one full year of data is recommended for model training.

Keywords: SCADA Data Analysis, Artificial Neural Networks (ANN), ANN Training Data, Normal Behaviour Models

1 Introduction

Wind power has seen rapid growth around the globe and the technology has seen significant cost reduction within the past decades [1]. In order to further increase the technologies competitiveness operators have the option to switch from a scheduled maintenance scheme to a so-called condition-based maintenance strategy. Here, maintenance decisions are based on information about the turbine's actual condition rather than on periodical inspections. In this context analysis of data from the turbine's supervisory control and data acquisition (SCADA) system has attracted considerable research interest (compare [2]). Above all, normal behaviour models (NBM) using artificial neural networks (ANNs) have shown promising result. In this approach an ANN is trained to predict SCADA parameters under healthy turbine conditions. During the following application phase, ANN predictions are compared to the values measured by the SCADA sensors. Deviations that exceed a certain threshold are indicative of a component malfunction (compare Figure 1).

A particularly crucial requirement for successful NBM using ANNs is the availability of sufficient healthy training data. This is due to the fact that ANNs are good in sample weak out-of-sample predictors. Therefore, the range of training parameters should be as varied as possible while still representing normal turbine behaviour. [3], who were among the first to apply ANN based NBM within the wind domain, therefore selected behaviour patterns from the SCADA data sets manually. This helped to reduce the required training data to an equivalent an equivalent of 3 months of operational data while still covering the full range of each input parameter. After this tedious procedure a relatively small feed-forward network was trained to predict gearbox related parameters. [4] picked up the promising early findings and conducted a comparison between linear and non-linear methods for modelling SCADA temperature measurements. With respect to the amount of required training data the authors refer to [3]. However,

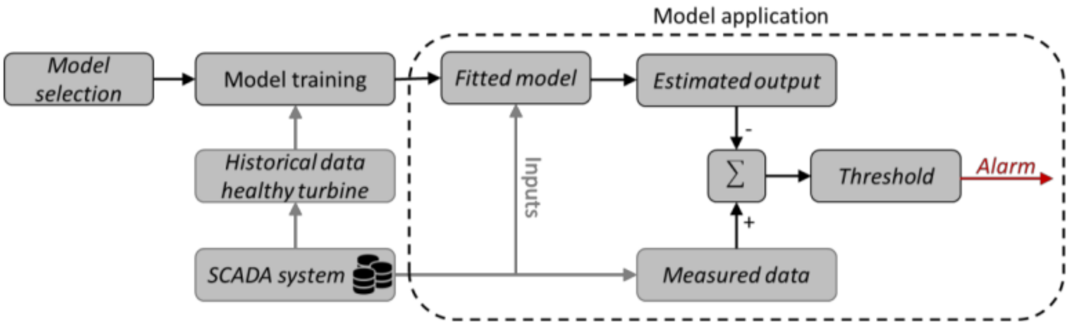

Figure 1: Scheme of normal behavior model-based anomaly detection

data from three continuous months of operation are used and described as sufficient. [5] use we well three months of continuous operational data for a comparison of different types of models to predict SCADA temperatures. Additionally it is reported, that depending on the turbine and its operational behaviour as little as one month of training data could be sufficient. The authors specifically highlight the need for more research regarding training data requirements.

More recent work applies larger and more complex models than the above mentioned literature. [6] train a recurrent neural network structure with 20 neurons in the hidden layer using 12 months of operational data. [7] train a recurrent neural network with 256 units in the hidden layer using 14 months of continuous data. It is particularly argued for a training period covering all seasonal effects. [8] apply a deep neural network with 3 hidden layers and 100 neurons each. In order to increase the training set, data from all turbines in a park is combined. This strategy was followed earlier (see for example [9] and [10]) but no clear advantages have been reported compared with turbine specific sampling.

It becomes clear, that even though the question of how much training data is required to train robust and reliable normal behaviour models is essential to the method's success, it has not been systematically addressed in literature so far. This is why this contribution aims to answer this question more systematic way. The result are meant to serve as a guideline for fellow researchers or practitioners from the industry.

2 Method

In the present contribution ANNs are used to solve a regression task in a supervised learning setting. This section aims to gives a compact background on artificial neural networks based on [11] and [12] to which the author refers for more details.

2.1 Artificial Neural Networks

Artificial neural networks are a class of flexible non-linear function approximator consisting of interconnected layers of neurons. Each neuron computes an output value by evaluating a linear-combination of its inputs with a non-linear activation function f. The coefficients or weights of the linear combinations represent adaptive parameters that are adjusted during model training (W_x and W_h). In feed-forward neural networks layers of neurons are connected in a way so that there is no closed directed circles in the graph. Recurrent neural networks on the other hand allow this kind of cycles. This allows the present value of a variable to influence its own value in the future. This has shown to be particularly beneficial in the context of sequential data. The mathematical expressions of a feed-forward layer and a recurrent layer are displayed in (1) and (2) in matrix notation. W are the respective weight matrices, b the bias terms, x_t the layer inputs at time step t. For the recurrent layer h_t is the hidden state at time t and $h_{(t-1)}$ is the hidden state of the previous layer at time $t-1$ or the initial hidden state.

$$h_t = f(W_x x_t + b) \tag{1}$$

$$h_t = h(W_x \; x_t + b_x + W_h \; h_{(t-1)} + b_h) \tag{2}$$

2.2 ANN training

During ANN supervised training the network's adaptive parameters are adjusted in order to minimize a cost function cost-function which is indicative of the model's ability to predict the selected target variable. This can be the mean squared error for example. Numerically this represents a non-convex optimization problem over the model's weight space. Its approximate solution is mostly found by applying gradient-based methods in combination with error backpropagation. Error backpropagation starts with a randomly initialized set of weights $w_{t=0}$ and consequently adjusts them in a sequence of steps. Each step involves the derivation of the error function $E(w(t))$ with respect to the model weights and calculating the consequent adjustment to be made to each weight (compare (2)). The parameter η is known as learning rate and scales the step-width in a certain direction.

$$w(t+1) = w(t) - \eta \Delta E(w(t)) \tag{3}$$

In practice the full training data set is often divided into multiple batches which contain a certain amount of training examples based on which a weight-update is conducted. This allows to handle redundancy in the data more efficiently and helps to escape local minimums. The weight update process is stopped upon a specific stopping criterion. This can be the number of times the complete training data set was presented to the network (called epochs), training error improvement saturation or early stopping criteria to prevent overfitting the training data set.

3 Design of experiments

Training data requirements for machine learning problems depend on the properties of the specific learning problem itself and are therefore hard to generalize. This is why an empirical approach is chosen. Certainly, the complexity of the learning problem and the chosen model class will have a major impact. The complexity of the learning problem is implicitly accounted for whereas the impact of model complexity will be analysed by comparing two inherently different models. All investigations were conducted using the python package Pytorch [13].

3.1 Data sets

SCADA data sets from two turbines are analysed in this study. Each contains 24 continuous months of operational data in the typical 10 minutes resolution. The turbines are from the same manufacturer and sited in Europe. In addition to the SCADA data information about major maintenance activity and replacements is present. There were no major maintenance actions in Turbine A, which will therefore serve as a validation turbine to evaluate the temperature modelling performance and model robustness against false alarms. Turbine B showed rising temperatures in the gearbox during the second year of operational records and will be used to evaluate anomaly detection capabilities. The data sets from both turbines are split into a training set and a test set of 12 months each.

3.2 ANN architectures

The most common ANN architecture for NBM using SCADA data is a feed-forward network with basic SCADA measurements as inputs, one hidden layer containing a few hidden units and targets from the gearbox or generator (compare [2]). To represent this common configuration, a feed-forward neural net (FFNN) as specified in Table 1 is applied. To account for the recent trend towards larger ANN architectures, a recurrent neural network (RNN) as specified in Table 1 is investigated as well.

3.3 Experiments

Firstly, two learning rates (0.01 and 0.001) and two batch sizes (200 and 2000) are tested for both model architectures and the five different training periods. With these optimal training hyper-parameters the effect of training data length on model performance in terms of mean absolute error is analysed. The models are trained with 1, 2, 3, 6 and 12 months of continuous operational data from healthy Turbine A. The data is pre-processed by excluding non operational periods and parameter scaling. Training is stopped and the best model selected if the error on the test set did not reach a new minimum in the last 250 epochs. For post-processing absolute model errors larger than 15°C were removed, since they were found to be caused exclusively by discontinuities in the data. A test set of one year continuous operation is used for validation. The models are trained repetitively for each training length to cover the whole year of training data in a rolling manner (compare Figure 2). The systematic split of the training year results in a varying number of models being trained for each training length as well as a varying overlap of training times. There are for example 12 models trained with a single month of training data, without any overlap between the training periods, whereas there are 7 models using 6 months of training data with an overlap of 0 to 5 months. This allows to account for the range of expected outcomes when randomly picking one of the training periods from one year of operational data.

Finally, both models are analysed regarding their robustness against false alarms and their ability to detect problems using Turbine A and B respectively. In order to increase robustness it is common to evaluate an averaged model errors for anomaly detection. Moreover, the threshold for alarm generation should be selected based on model performance during training ([6]). Therefore, all error measures are calculated as a rolling mean over 144 data points, which is an equivalent of 1 day, and the averaged test errors are divided by the standard deviation of the averaged training errors.

Table 1: Specification of selected reference ANN architectures

	Feed-forward neural net (FFNN)	Recurrent neural net (RNN)
Input parameters	Ambient temperature, Nacelle temperature, Power, Rotor speed	
Target parameters	Gearbox bearing temperature	
Input time lag	t	t, t-1, t-2...t-36
Input - Hidden - Output	4-5-1	144 - 100 - 1
Total model parameters	31	1087
Activation function	sigmoid	rectified linear unit (relu)

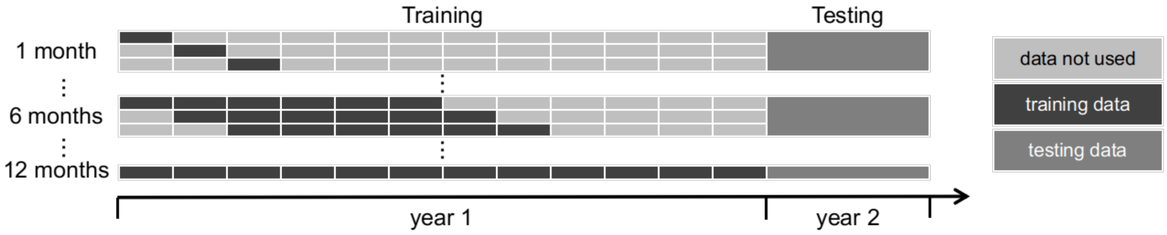

Figure 2: Split of two years of operational data into training and testing for the different training periods

4 Empirical Results

4.1 Results on training length

Cross validation smaller batch sizes of 200 samples and a learning rate of 0.01 beneficial for the FFNN. For the RNN on the other hand larger batch sizes of 2000 samples in combination with a learning rate of 0.001 showed better results and less overfitting. The training length variation resulted in a large number of models of different quality with the mean average error ranging from 1.22°C to 2.64°C (compare 3). These results are comparable to others found in literature (e.g. [5]). It becomes clear that more training data leads on average to better model performance with the overall best results at a training period of one full year. At the same time longer training periods show a lower variation between training runs. When comparing the best models of each training length, however, the difference in performance is not as significant. This might be an explanation for the wide range of sufficient training data requirements reported in literature. It can be observed that when choosing a training periods shorter than a full year, the time of the year the training data was chosen from is more important than the actual length of the training period. Even though parameter ranges and training data points per month were investigated for the bad performing months, they give no clear explanation. Also, effects of model initialization were excluded since repetitive training runs yield comparable results. Further studies with multiple turbines might lead to an explanation. As a result from this study it can be noted, that maximizing the training period minimizes the risk of bad model performance significantly.

When comparing the two model architectures it can be seen that the more complex input data in combination with the more complex model architecture results in consistently superior model performance, in case enough training data is present. This advantage becomes more significant the more training data is available, which confirms the assumption, that more complex model structures require more data. The comparison of model architectures will be kept at this general level, since the RNN uses more input parameters and therefore the increase in performance can not only be attributed to the architecture itself.

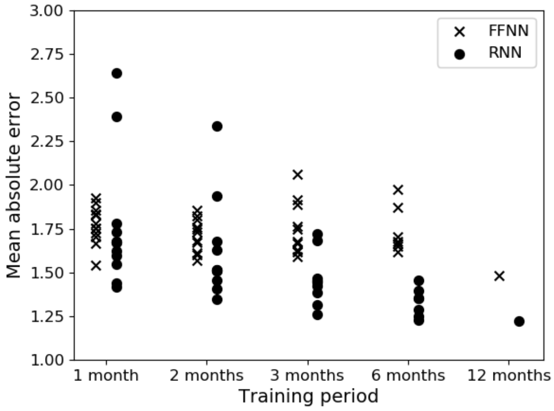

Figure 3: MAE for different training periods

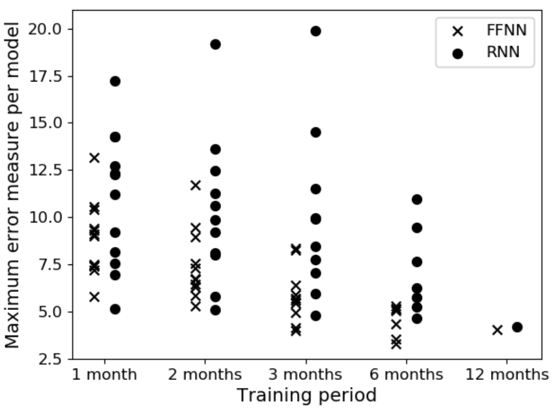

Figure 4: Test error in training standard deviations

4.2 Results on anomaly detection

Both models are first applied to healthy Turbine A for validation against false alarms. Figure 3 shows the maximum averaged and normalized error for each model. Normalization was conducted with the standard deviation of the averaged training error. In contrast to the mean absolute error comparison the the RNN structure shows a higher ratio between validation error and training error standard deviation. This can be explained by the model's ability to fit the training data better resulting in worse out of sample generalization. For anomaly detection this has severe consequences. The maximum error measure of each configuration should be selected as a the threshold for model alarms in order to minimize false positives. At the same time high thresholds translate into less sensitive failure detection capabilities. Therefore it can be concluded that RNNs should be trained with at least one year of training data to reduce the alarm threshold. This assumption is confirmed when applying the methodology to Turbine B. Model error measures were monitored for three months before the high temperature in the gearbox bearing occurred. Alarms more than 1 month before the severe problems were labeled as early alarms. Results for RNNs with less than one year of training data have shown to be highly unreliable in terms of early as well as late anomaly detection. For FFNNs this is slightly different. Figure 5 shows the exemplary results for The FFNN trained with 6 months of training data. 5 out of 7 models issue early alarms in August. Apparently, the restrictions imposed by the simple model class do not allow the model to overfit to the same degree. Nevertheless, for none of the training periods shorter than one year, all trained models are able to consistently early detect the gearbox problems. The complete results are displayed in Table 2.

Table 2: Results anomaly detection in #models/total#models

	FFNN					RNN				
Training period:	1	2	3	6	12	1	2	3	6	12
Alarm threshold	13.0	11.6	8.3	5.3	4.0	17.0	19.0	19.8	10.8	4.2
Early alarms	2/12	2/11	2/10	5/7	1/1	4/12	1/11	2/10	0/7	1/1
Late alarms	12/12	10/11	10/10	7/7	1/1	9/12	4/11	3/10	7/7	1/1

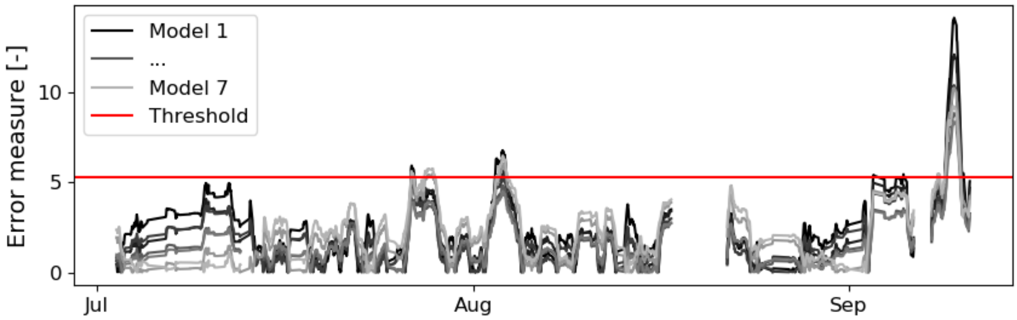

Figure 5: Exemplary anomaly detection plot FFNN 6 months of training

5 Summary and Conclusions

The empirical study has shown that training data length has a significant impact on the accuracy for parameter modelling. Moreover, the results have shown that larger and more complex ANN structures can increase prediction accuracy but tend to overfit small training data sets in an anomaly detection setting. The comparably simple FFNN does not suffer from the same problem and can detect anomalies with as little as one month of training data. Nevertheless, a larger amount of training data reduces the risk of poor model performance significantly. Based on the result the author suggests to train ANNs for modelling SCADA parameters with a full year of operational data. In order to generalize the findings of this study a larger amount of turbines has to be analysed in a similar fashion.

References

[1] IRENA 2019 Renewable power generation costs in 2018

[2] Tautz-Weinert J and Watson S J 2016 *IET Renewable Power Generation* **11** 382–394

[3] Zaher A, McArthur S, Infield D and Patel Y 2009 *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* **12** 574–593

[4] Schlechtingen M and Santos I F 2011 *Mechanical systems and signal processing* **25** 1849–1875

[5] Tautz-Weinert J and Watson S J 2017

[6] Bangalore P, Letzgus S, Karlsson D and Patriksson M 2017 *Wind Energy* **20** 1421–1438

[7] Bach-Andersen M, Rømer-Odgaard B and Winther O 2017 *Wind Energy* **20** 753–764

[8] Wang L, Zhang Z, Long H, Xu J and Liu R 2016 *IEEE Transactions on Industrial Informatics* **PP** 1–1

[9] Wang Y and Infield D 2013 *Renewable Power Generation, IET* **7** 350–358

[10] Tan M and Zhang Z 2016 *IEEE Transactions on Industrial Informatics* **12** 1–1

[11] Bishop C M 2006 *Pattern recognition and machine learning* (Springer)

[12] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* Adaptive Computation and Machine Learning series (MIT Press) ISBN 9780262035613 URL `https://books.google.de/books?id=Np9SDQAAQBAJ`

[13] Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L and Lerer A 2017 *NIPS Autodiff Workshop*