Cross-Turbine Training of Convolutional Neural Networks for SCADA-Based Fault Detection in Wind Turbines

Markus Ulmer[1], Eskil Jarlskog[2], Gianmarco Pizza[3], and Lilach Goren Huber[4]

[1,4] *Zurich University of Applied Sciences, Technikumstrasse 9 Winterthur 8400, Switzerland*
markus.ulmer@zhaw.ch
lilach.gorenhuber@zhaw.ch

[2,3] *Nispera AG, Hornbachstrasse 50 Zurich 8008, Switzerland*
eskil.jarlskog@nispera.com
gianmarco.pizza@nispera.com

ABSTRACT

Machine learning algorithms for early fault detection of wind turbines using 10-minute SCADA data are attracting attention in the wind energy community due to their cost-effectiveness. It has been recently shown that convolutional neural networks (CNNs) can significantly improve the performance of such algorithms. One practical aspect in the deployment of these algorithms is that they require a large amount of historical SCADA data for training. These are not always available, for example in the case of newly installed turbines. Here we suggest a cross-turbine training scheme for CNNs: we train a CNN model on a turbine with abundant data and use the trained network to detect faults in a different wind turbine for which only little data are available. We show that this scheme is able to considerably improve the fault detection performance compared to the scarce data training. Moreover, it is shown to detect faults with an accuracy and robustness which are very similar to the single-turbine scheme, in which training and detection are both done on the same turbine with a large and representative training set. We demonstrate this for two different fault types: abrupt and slowly evolving faults and perform a sensitivity analysis in order to compare the performance of the two training schemes. We show that the cross-turbine scheme works successfully also when training on turbines from another farm and with different measured variables than the target turbine.

1. INTRODUCTION

A central challenge in training algorithms to detect and diagnose faults in technical systems lies in the fact that critical faults are very rare and are often very specific in character. This is true also in the case of fault detection on wind turbines. A standard approach is therefore to train fault detection algorithms that do not rely on the exact nature of faults, but rather extract information from turbines under normal non-faulty conditions, also known as normal state modeling (Stetco et al., 2019; Schlechtingen & Santos, 2014). After training with normal data only, the algorithm is used for online detection, in which deviations from normality manifest in high prediction errors. In order to detect incipient faults such deviations from normal behavior should be detected as early as possible.

The conventional approach for fault detection of wind turbines is a single machine approach (Tautz-Weinert & Watson, 2016; Leahy et al., 2016; Fu et al., 2019; Kong et al., 2020). A machine learning model is trained using data from a specific turbine, measured during normal behavior (healthy data). During training, the model is trained to recognize the normal behavior of a selected target variable. Provided with enough representative data, the model learns to predict accurately the target variable of unseen test data, assuming it originates from the same turbine in its healthy functioning state. However, when the turbine state is degraded, we expect the prediction of the trained network to deviate from the measured value. The deviations, or prediction errors can therefore be used as a "health index (HI)" for an early detection of incipient faults.

This approach is only applicable for turbines which are in operation for long enough to accumulate sufficient representative data. It cannot be applied to newly installed turbines, or turbines for which data are missing due to a technical reason. A scalable practical deployment of fault detection algorithms for wind turbines requires a solution to this problem.

The problem of little or no training data and the need to use fleet information for fault detection has been intensely dis-

cussed in the fault diagnosis community, as described in recent review articles (Zhao et al., 2019; Lei et al., 2020). Methods for transfer learning (Wang et al., 2019; Shao et al., 2018; L. Guo et al., 2018) and domain adaptation (Zheng et al., 2019; Li et al., 2018) have been developed for fault diagnosis applications, suggesting how to adapt detection algorithms to predict faults in a certain machine after being trained on another machine. Most of these methods rely on using high resolution (e.g vibration) data. For wind turbine fault detection, however, there have been only very few attempts to go beyond the single-turbine approach and transfer data-driven information between different turbines. A recent work (Lebranchu et al., 2019) suggested to exploit farm statistics in conjunction with single-turbine data to boost the performance of fault detection algorithms based on 10-minute SCADA data. Another paper uses transfer learning methods for the purpose of wind power prediction (Qureshi et al., 2017). Transfer learning neural networks have been used for fault detection based on wind turbine vibration data (J. Guo et al., 2020), and for the purpose of ice detection on wind turbines using SCADA data (Yun et al., 2019), but to the best of our knowledge not for generic fault detection in various turbine components based on the already available 10-minute SCADA data.

The possibility to apply trained fault detection algorithms across various operating conditions has an additional relevant aspect. Training machine learning algorithms on large fleet of machines can be costly and can serve as the decisive factor for operational deployment. The potential of training on only few wind turbines and using the trained models to predict on the entire fleet is an additional incentive for transfer learning for wind turbine fault detection.

In this paper we suggest to apply a simple approach of transfer learning for fault detection of wind turbines based on 10-minute SCADA data only. The fault detection is not limited to one fault type and has been tested on multiple components in the turbine. In particular, we show that a CNN model that was developed for a single-turbine fault detection purpose (Ulmer et al., 2020), demonstrates surprisingly high abilities to detect faults when used in a cross-turbine scheme. This means that after training the CNN model on a certain turbine, for which historical data are abundant, we can use the trained network to predict on another turbine in the same wind farm or in another wind farm, and still detect its faults early enough and with a high precision. This cross-turbine scheme includes a rescaling step of the prediction outcomes in order to show a comparable performance to the standard single-turbine scheme. The advantage of the proposed method is twofold: first is its simplicity, not requiring additional complex network architectures for domain adaptation which are then costly in terms of training times. The second advantage lies in the potential for general application, which goes beyond a specific fault type or fault location.

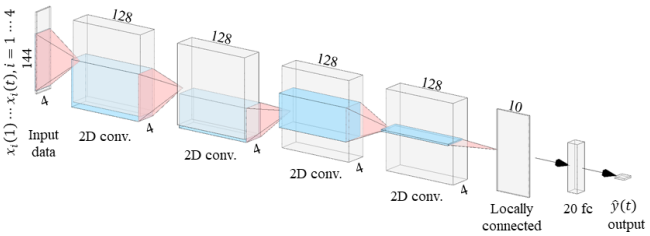

Figure 1. CNN architecture.

The focus of this paper is not on showing the superiority of the CNN architecture over other algorithms. This was argued in a previous publication (Ulmer et al., 2020). We also note that the examples we selected for the sake of demonstration of our method were chosen due to a limited availability of clearly labeled data. However, extensive testing of the algorithm is at the center of our on-going research.

In the next section we describe the architecture, input and output of the CNN model which is used in both the single- and the cross-turbine schemes.

2. ALGORITHM DESCRIPTION

2.1. Network Architecture

In this work we apply a CNN model for time series prediction using a multivariate input. The details of the neural network can be found in our previous publication (Ulmer et al., 2020). The target variable y is typically a temperature of a certain component of the turbine, e.g a generator bearing, the gearbox oil or the hub temperature. In order to remain generic, we select a small number of measured variables from the SCADA system and use them as inputs to the CNN model: output power, ambient temperature, wind speed and rotor rpm, denoted by x_1, x_2, x_3, x_4. Selecting a small yet representative set of inputs helps to keep our algorithm generic and address fault types in as many components as possible.

The CNN receives multivariate input sequences of dimension 4x144 corresponding to the 4 input variables over a period of one day: $x_i(1)\dots x_i(t)$ with $i = 1\dots 4$. We generate the input sequences with a sliding window with a 10 minute overlap. The network has 4 convolutional layers, with 128 2D filters each. The time dimensions of the filters are 32,18,8,8. Their width is 4, covering the four inputs, see Figure 1.

The last convolution layer is locally connected. The representation is then flattened to a fully connected layer of 20 neurons which then connect to a single output. For more details about the network architecture we refer the reader to our previous paper (Ulmer et al., 2020). The output $\hat{y}(t)$ is the prediction target variable at the end of the 1 day period of the input sequence. The loss is the squared error between $\hat{y}(t)$ and $y(t)$.

The CNN model yields an output $\hat{y}(t)$ every 10 minutes. The prediction error (residuals) time series $\delta(t)$ are calculated by:

$$\delta(t) = y(t) - \hat{y}(t) \tag{1}$$

2.2. Training Schemes

We distinguish between two training schemes for fault detection on a target turbine T:

1. Single turbine scheme: a training and validation set from turbine T are used to train the CNN. The trained CNN is then used to predict the output variable of turbine T at all times.
2. Cross-turbine Scheme: training and validation sets from turbine S (different from T) are used to train the CNN. The trained CNN is then used to predict the output variable of turbine T at all times. The cross turbine scheme includes an additional step of rescaling of the prediction errors, see Algorithm 1. We expect discrepancies between turbines in the typical normal values of certain input and output variables. We correct for these discrepancies by means of a linear regression of $y(t)$ on the prediction $\hat{y}(t)$ using data from a short period of three months (the "reference set" denoted by $\mathcal{R}$). We assume that such data are available also for a newly installed turbine after a time period of several months.

The performance of the proposed CNN model in the standard single turbine straining scheme has been analyzed in our previous publication (Ulmer et al., 2020).

Algorithm 1: Algorithm for Cross-Turbine Health Index Calculation

Result: Health Index $h(t_w)$
Train CNN on turbine S;
if *Cross-Turbine Training* **then**
  Use trained CNN to predict $y_T(t)$ on turbine T;
  Calculate residuals $\delta_T(t) = y_T(t) - \hat{y}_T(t)$;
  Linear regression $y_T(t) \sim \hat{y}_T(t) + X_T(t)$ for $t \in \mathcal{R}$ (a small healthy subset);
  Rescale residuals $\delta_T(t) \rightarrow \delta(t)$ using regression coefficients;
else
  Use trained CNN to predict $y(t)$ on turbine S;
  Calculate residuals $\delta(t) = y(t) - \hat{y}(t)$;
end
Sliding window mean of $\delta(t)$ for $t_w - 1 < t < t_w$.

2.3. Post-processing and Threshold Setting

The CNN is trained on healthy data, and is therefore expected to predict accurately as long as the turbine state is normal. We thus expect large prediction errors $|\delta(t)| \gg 0$ only when the turbine condition deviates from normality, which we aim to detect as early and accurately as possible. Here we focus on critical faults which lead to an increased component temperature, and we therefore aim at detecting the onset time of faults with a large positive $\delta(t)$. To this end we post process the prediction error time series by applying a high power filter followed by a sliding window aggregation. As a consequence, we obtain the time series of HI, $h(t_w)$, with one hour time resolution.

A fault is detected whenever $h(t_w) > h_c$, with h_c being the threshold level of the HI. We assign a p-value significance score to each $h(t_w)$ result with respect to the estimated distribution of the validation set errors $N(\mu, \sigma^2)$. The threshold can then be set using a desired significance level α that is related to the confidence of detection: all points with a p-value smaller than the fixed significance level α are highly unlikely to be drawn from the healthy error distribution and are declared as faulty (see for example Clifton et al., 2008). In the cross-turbine case, we use the reference set $\mathcal{R}$ instead of the validation set for the purpose of threshold setting.

3. RESULTS AND DISCUSSION

In this section we present an evaluation of the cross-turbine scheme. In order to do this, we detect faults on test cases of two target turbines A_0 and B_0. To emulate a situation of low data availability we intentionally use only a small part of data from the target turbines when training on other turbines in the cross-turbine scheme. The smaller data sets from the target turbines A_0 and B_0 are then only used at the rescaling step. To evaluate the cross-turbine scheme we compare the results to a single-turbine training scheme under two opposite scenarios: in the "limited data" scenario we use the standard single turbine scheme with only three months of data. In the "Baseline" scenario we emulate the ideal case, in which there is indeed enough (in this case 9 months of) training data from both A_0 and B_0, such that the cross turbine scheme is not required and one could resort to the usual single turbine scheme. In practice, however, we apply the cross-turbine scheme to wind turbines with little historical data for which the single turbine scheme does not perform well enough.

In the following we present the comparison of fault detection between the standard single-turbine training and the cross-turbine training scheme. We demonstrate the comparison on two fault types in two different wind farms. on Turbine A_0 two faults with an abrupt time evolution have been detected. This means that early signatures of an abnormal condition can develop within hours, but still lead to a turbine stoppage several weeks or months later. On Turbine B_0, on the contrary a slow condition degradation over several months was observed, and eventually led to a turbine stoppage. The goal of early fault detection in both cases is to detect the abnormality signatures or the degradation as early and accurately as possible.

It is important to point out that we apply two different evalua-

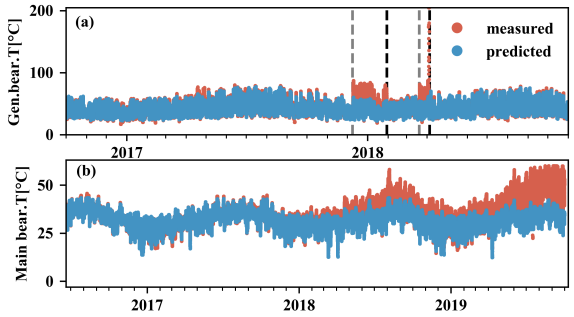

Figure 2. predicted and observed target variables in a standard single-turbine scheme with enough representative data. (a) Generator bearing temperature in turbine A. Grey dashed: true fault initiation. Black dashed: true turbine stoppage. (b) Main bearing temperature in turbine B.

tion criteria for these two cases. In the case of turbine A_0 we had access to information that allowed us to label an entire period as "faulty". In this case it is possible to use the standard evaluation metrics such as recall and precision. However, for turbine B_0 labels were not available. In this case we inferred the degradation directly from the data, and evaluate different schemes according to their ability to detect the degraded state as early as possible. We therefore compare the first detection date of the various schemes. This criterion is particularly relevant for slowly evolving faults, and less for abrupt faults such as the ones observed for turbine A_0.

3.1. Detecting Abrupt Faults

Figure 2(a) displays the measured and predicted values of the generator bearing temperature of wind turbine A_0 over a period of about 2 years. The first 9 months of data from A_0 were used for training and validation and the rest for testing (standard single-turbine training with a large data set).

Figure 3 shows the calculated health indices for the generator bearing temperature $h(t_w)$ in degrees Celsius in each available time window. The color code denotes the selected threshold, in this case parameterized by $\alpha = 0.0001$ in all panels. All red colored points in the plots indicate a detected faulty behavior. For this turbine we had access to "true labels" from the operator, indicating the onset of two faults, followed by their actual detection time by the staff on site. The first one (f_1) started showing up 9.12.2017 and lead to a complete turbine stoppage on the 30.1.2018. The second fault (f_2) started on the 20.3.2018, causing a stoppage on the 5.4.2018.

Figure 3(a) shows the results of training and predicting on the same turbine, A_0 in the "Limited Data" scenario, in which only three months of data are available for training the CNN model. This should first be compared with the ideal "Baseline" scenario of panel 3(b) in which a large data set of 9 months can be used for training. As expected, when training on a small data set, the prediction errors and thus the extracted health indices are much noisier, whereas training with 9 months allows for a much clearer distinction between healthy and faulty periods. From the color code of these two plots it is clear, that the detection with little training data is less precise (not all faulty points are above the detection threshold) and suffers from more false positives (some healthy points are detected as faulty). In order to achieve a high detection quality, while using only three months of data, we apply the cross-turbine scheme, thereby training on large data sets from other turbines and rescaling the results using the three months reference set of turbine A_0.

The results of the cross-turbine training are displayed in panels (c) to (g). Figures 3(c) and (d) show the results when training on large data sets (of 9 months) from turbines A_1 and A_2 from the same wind farm, and using the trained CNN to predict with the data of turbine A_0 and detect its faulty behavior. Panels (e) and (f) show the results when training on turbines B_0 and B_1 from a different wind farm than the target turbine A_0, but using the same variable (the generator bearing temperature) as output when training the CNN. In Panel (g) the training was done using a different output variable (the main bearing temperature) with turbine B_0 from the other wind farm.

The cross-turbine results in Fig. 3(c)-(g) show that the two faults in turbine A_0 would be successfully detected by the CNN model, not only when trained with enough representative data from the very same wind turbine A_0 but also when trained on similar amounts of data from turbine A_1 or A_2 from the same wind farm. Moreover, the transfer learning is possible also when training on similar turbines B_0 or B_1 from another wind farm in another geographical location, and even training with another target variable yields a health index of a similarly high ability to identify faulty behavior. The advantage of the cross-turbine training scheme over the single turbine training in the case of only little data (panel (a)) is clearly demonstrated in this Figure.

3.1.1. Detection Performance Evaluation

The above results can be quantified by measuring the detection performance against true labels. Here we label the entire period between fault initiation and turbine stoppage as "faulty". This applies for both faults f_1 and f_2. The rest of the data is labeled as "healthy". High performance corresponds to detecting a maximal fraction of the faulty time windows (True Positives or TP) with a minimal false positive (FP) rate. We introduce the following performance metrics:

- Time of first detection. Earliest time window t_w for which

$$h(t_w) > h_c \tag{2}$$

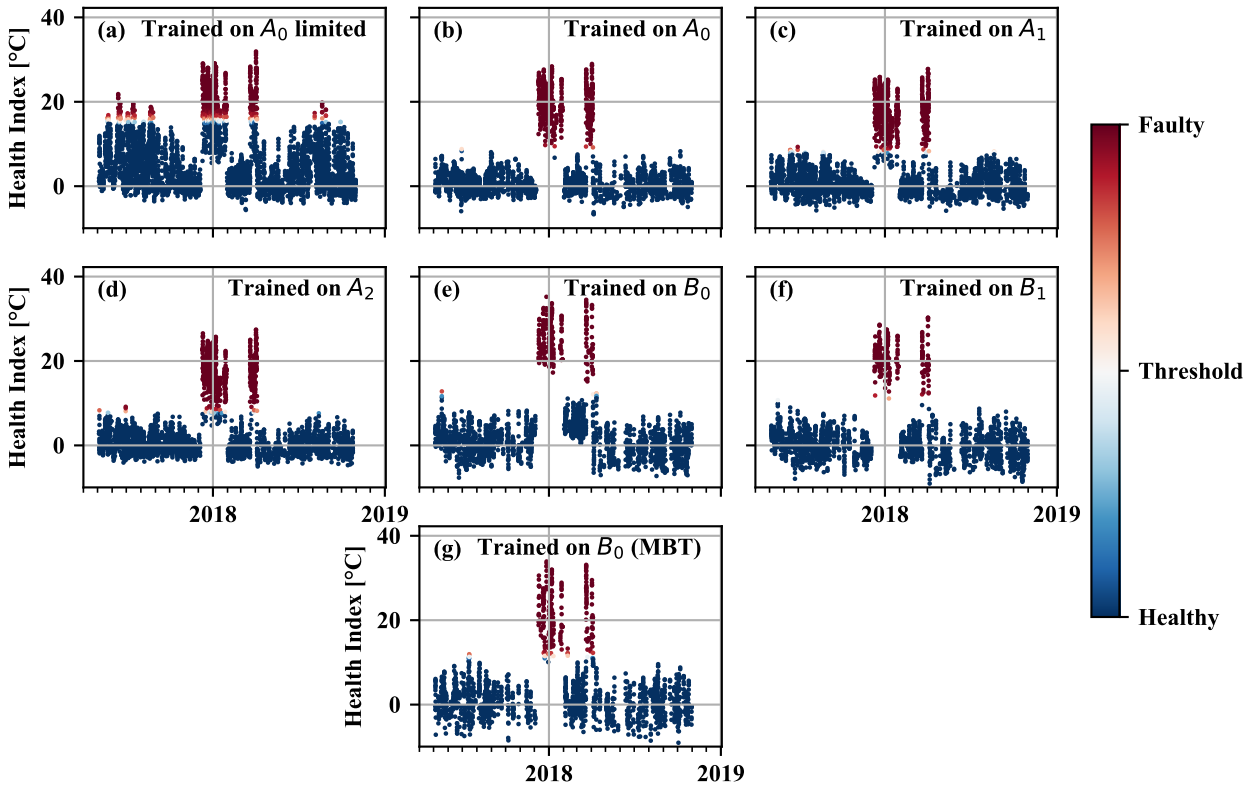

Figure 3. Health index for detection of abrupt faults in the generator bearing temperature of turbine A_0 using training data from turbine (a) A_0 in the single turbine scheme with limited training data (b) A_0 in the single turbine Baseline scheme with abundant data (c) A_1 in cross-turbine scheme (d) A_2 in cross-turbine scheme (e) B_0 in cross-farm scheme (f) B_1 in cross-farm scheme (g) B_0 in cross-farm with the Main Bearing Temperature (MBT) as target variable.

- Recall:

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \tag{3}$$

- Precision:

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} \tag{4}$$

- F_1 score:

$$F_1 = 2 \cdot \frac{\mathbf{Recall} \cdot \mathbf{Precision}}{\mathbf{Recall} + \mathbf{Precision}} \tag{5}$$

Table 1 summarizes the comparison between the different training schemes based on the above measures with a detection threshold of $\alpha = 0.0001$. The first line in the table displays the performance scores for a "Limited data" scenario, that is, assuming that only three months of data of turbine A_0 (the reference set) are available for training. As expected, we observe considerably lower recall, precision and F_1 scores than if we use 9 months of training data as in the Baseline scheme. This is a direct consequence of the lack of representative training data when it is based only on one season of the entire year. The Baseline single-turbine scheme in turn yields the highest scores of all. However, the performance scores when training on turbine A_2 with an appropriate rescaling, are remarkably close to the baseline scores. Training on turbine A_1 yields slightly lower recall and precision than the baseline, hinting at some turbine specific properties that are not captured by the CNN model and cannot be corrected for by the rescaling algorithm suggested above. In this case, a more elaborate scheme of transfer learning may improve the performance. Interestingly enough, the performance scores are not necessarily lower when the source turbine on which we train is in another wind farm, and even when we use the main bearing temperature (MBT) of B_0 during training and detect faults in the generator bearing temperature of turbine A_0. The main point here is that with limited data from the target turbine A_0, all cross-turbine training sets perform considerably better than a single-turbine training which relies on this very same data.

Table 1. Scheme comparison for abrupt fault detection in turbine A_0

Scheme	Train on	Recall	Precision	F_1
Limited data	A_0	0.62	0.83	0.71
Baseline	A_0	0.96	0.98	0.97
Cross-turbine	A_1	0.95	0.92	0.93
Cross-turbine	A_2	0.96	0.97	0.96
Cross-farm*	B_0	0.94	0.95	0.94
Cross-farm*	B_1	0.92	0.98	0.95
Cross-farm*	B_0 (MBT**)	0.96	0.93	0.95

*Cross-turbine scheme trained on a turbine from a different farm.
**Main Bearing Temperature as target variable.

3.1.2. Sensitivity Analysis

In order to systematically investigate the robustness and sensitivity of fault detection with different training schemes we perform a detailed sensitivity analysis. The performance scores are typically dependent on the choice of threshold. A higher detection threshold leads to a higher precision and a lower recall rate. This is demonstrated in Figure 4. The figure illustrates the sensitivity of the scores towards the desired confidence level C, defined in terms of the threshold α as

$$C = -\log_{10} \alpha \tag{6}$$

Setting a a smaller threshold α is equivalent to requiring an exponentially higher confidence C of the fault detection. Figure 4(a) and (b) show clearly that standard single turbine training with limited data (solid grey) suffers from poor performance scores throughout the entire range of confidence levels. Moreover, the scores are highly sensitive to the threshold choice. This stands in natural contrast to the case of single turbine Baseline training, where enough training data is used. Here the detection performance is also high compared to most of the cross-turbine training sets. Figure 4(a) shows that the recall of the Baseline (single-turbine) training (solid black) is somewhat higher than the one achieved when training on another turbine, whether inside or outside the farm. Here we see that cross-turbine training on another target variable than the predicted one (dark red thin diamonds) is more prone to missing detections at high confidence. Figure 4(b) displays the detection precision as function of the confidence score. Here as well, there are only minor differences between single-turbine training with large data sets and cross-turbine training for the case of limited data. The precision (reflecting the False Positive rate) is sensitive to the cross-turbine training mainly in the low-confidence regime, where we allow for some false positives by lowering the threshold for detection. In this case the single-turbine scheme suffers less from false positives than the cross-turbine scheme. More importantly, the performance scores of the baseline scheme are slightly more stable against changing the detection threshold (or the confidence level) than the scores of the cross-turbine

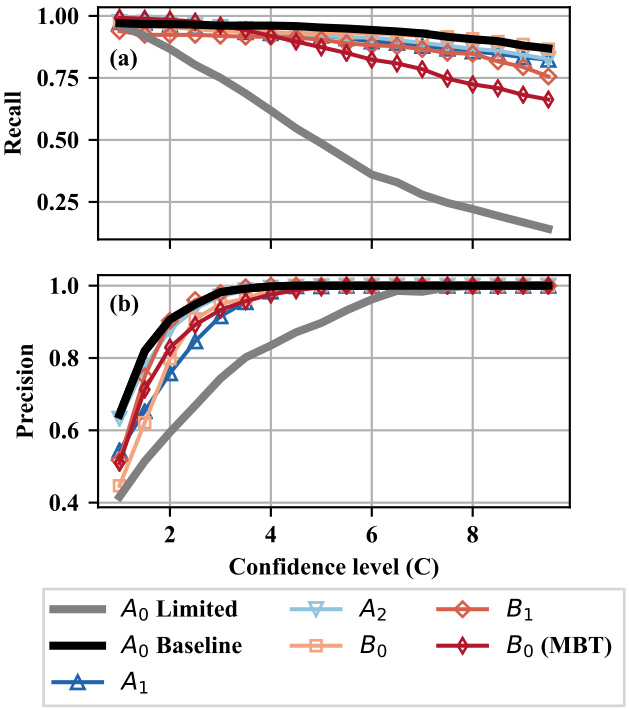

Figure 4. Sensitivity analysis for abrupt fault detection in the generator bearing temperature of Turbine A_0. Dependence of the (a) Recall and (b) Precision scores on the desired confidence level $C = -\log_{10} \alpha$ for seven training configurations: Single-turbine training on Turbine A_0 itself, with limited data or in the Baseline scheme (abundant data); cross-turbine training on Turbine A_1 or A_2 from the same farm; cross-turbine training on turbine B_0 or B_1 from a different farm, cross-turbine training on B_0 with a different output variable Main Bearing Temperature (MBT). The training sets are denoted in the legend with the corresponding source turbine.

scheme in these examples. However, when comparing all cross-turbine results with the single turbine case with limited data, the advantage of using the cross-turbine scheme is clear: the performance and robustness of all cross-turbine training sets is closer to the ideal single turbine Baseline (solid black) than to the limited data set (solid grey).

The analysis of the above example of detection of an abruptly evolving fault demonstrates a very successful transfer of the learning ability of the CNN model between different turbines. Moreover, the learning is transferred also from turbines in a different wind farm and even when training the network to predict another target variable such as the main bearing temperature while detecting faults of the generator bearing temperature.

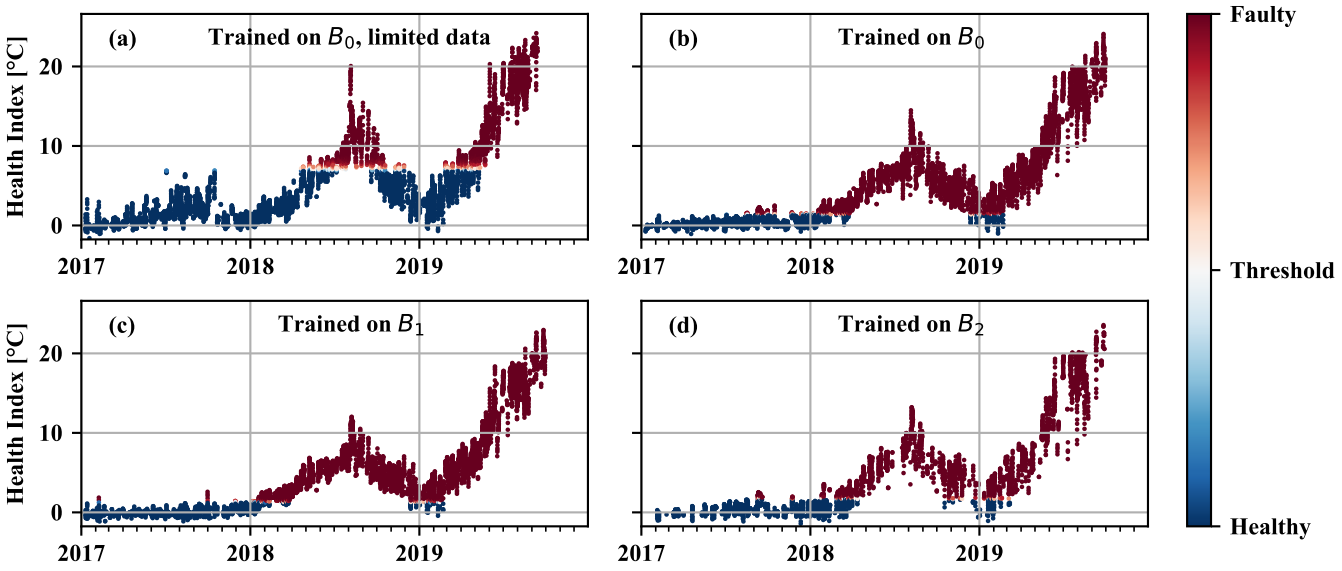

Figure 5. Health index for detection of slowly degrading faults in the main bearing temperature of turbine B_0 using training data from turbine (a) B_0 in the single turbine scheme with limited training data (b) B_0 in Baseline or single turbine scheme with abundant training data (c) B_1 in cross-turbine scheme (d) B_2 in cross-turbine scheme.

3.2. Detecting Slow Degradation

In order to test the applicability of the cross-turbine training scheme with our CNN model, we discuss a second example of turbine B_0 in a different wind farm. Here the time evolution of the faulty condition is slow, over months instead of hours as in the previous example. A fault was detected by our algorithm in the main bearing temperature and verified with the farm operator in retrospect. Below we demonstrate the possibility to train the CNN with data from a wind turbine from the same farm, Turbine B_1 or B_2, in order to detect the slowly degrading condition of Turbine B_0 as early as possible.

Figure 2(b) shows the raw measured temperature and the predicted values of the generator bearing of wind turbine B_0 over a period of about 3 years. The first 9 months of data from turbine B_0 were used for training and validation and the rest for testing ("Baseline" single-turbine scheme for large representative training data).

Figure 5 shows the resulting health index $h(t_w)$ in degrees Celsius for each time window. Figure 5(a) shows the results of training and predicting on the same turbine B_0 with only limited training data of 3 months. The results can be contrasted with the one of panel (b) of this figure, which displays the Baseline results of training and predicting on B_0 using a large data set of 9 months. Clearly, the prediction errors and thus the extracted health indices in the limited data case suffer from a lower signal to noise ratio, leading to a later detection time. The cross-turbine scheme is examined in Figures 5(c) and (d). In these cases training was performed on large data sets (similar to the Baseline) from turbines B_1 and B_2 respectively, and using the trained CNN to predict with the data of turbine B_0 in order to detect its faulty behavior. It is seen that cross-turbine training on either turbine B_1 (panel (c)) or B_2 (panel (d)) yields comparable results to the ones on panel (b), where both training and detection are on the same turbine B_0 with enough representative training data. These allow for a considerably clearer and earlier fault detection than in the limited data case of panel (a).

3.2.1. Detection Performance Evaluation

In order to quantify the detection performance for the slowly degrading fault we could not use true labels and we therefore compare the date of first detection among the four cases. Some results are summarized in Table 2. As an example, we set the detection threshold on $\alpha = 0.0001 (C = 4)$ and examined the different detection dates when training on B_0, B_1 and B_2. We compare these results to the "Limited data" scenario in which the CNN is trained only with the three months of the reference set of B_0. As expected, training with little data leads to lower signal-to-noise ratio of the health indices and thus to a much later fault detection compared to the Baseline training scheme with 9 month of data. Comparing the latter to the cross-turbine scheme with B_1 and B_2, we do observe a clear advantage of the Baseline scheme, where we train with abundant data and detect on the same turbine, and could detect first faulty signatures already mid-August 2017. When training on B_1 or B_2 the detection with the same confidence level is postponed by about six or three weeks respectively. This, however is still some 8 months earlier than what one

could achieve with only limited data from B_0 without using the cross-turbine scheme.

Table 2. Scheme comparison for slowly evolving fault detection in turbine B_0

Scheme	Train on	First detection ($\alpha = 0.0001$)	First detection high confidence
Limited data	B_0	20.04.18 16:00	20.07.18 16:00
Baseline	B_0	15.08.17 17:00	09.09.17 18:50
Cross-turbine	B_1	30.09.17 14:20	30.09.17 14:20
Cross-turbine	B_2	09.09.17 14:20	10.09.17 17:20

3.2.2. Sensitivity Analysis

We perform a sensitivity analysis for the second fault type, in order to compare the detection performance and robustness among the various training sets. Here we test the dependence of the earliest detection date on the desired confidence level C (or equivalently the threshold value $\alpha = 10^{-C}$). The results are displayed in Figure 6. The different curves correspond to the four training sets B_0 Limited data (black diamonds), B_0 Baseline (blue squares), B_1 (red circles) and B_2 (yellow triangles). Higher confidence levels naturally lead to a later (but more certain) detection in all four cases. Using a small data set in a standard single turbine training scheme leads to either false detections (often too early) at low confidence levels or late detections at higher confidence levels.

Cross-turbine Training with data from turbines B_1 or B_2 clearly enables early detection of the degraded state of turbine B_0 in case only little data from B_0 is available. The detection is naturally not quite as early as if trained with a large data set from the same turbine, if such data is available, but the profit in using the cross-turbine training is clearly demonstrated. The accuracy is lost especially if we lower the detection threshold and aim at a low or intermediate confidence level, thus allowing for some level of false positives in order to detect faults as early as possible (these false positives can be eliminated in a later stage of aggregated thresholding or some process control logic). For very high confidence levels, training with turbine B_2 yields as accurate, early and stable fault detection as with the original turbine B_0.

The conclusion from analyzing the results for detection of slowly degrading faults is that the CNN model with cross-turbine training in case of limited training data performs very well and rather similarly to the standard single-turbine training with a large data set. Compared to a single-turbine training with only little data, the cross-turbine scheme offers a considerable improvement concerning the earliest fault detection time. The cross-turbine scheme can thus be used as an alternative to the standard training scheme in case of limited data for the target turbine. However, the choice of the source turbine, i.e the turbine used for training, can influence to some extent the accuracy and reliability of the detection. How to select the source turbine (or perhaps turbines) in a smart way is an important question that goes beyond the scope of the present paper and will be investigated by us in a separate work.

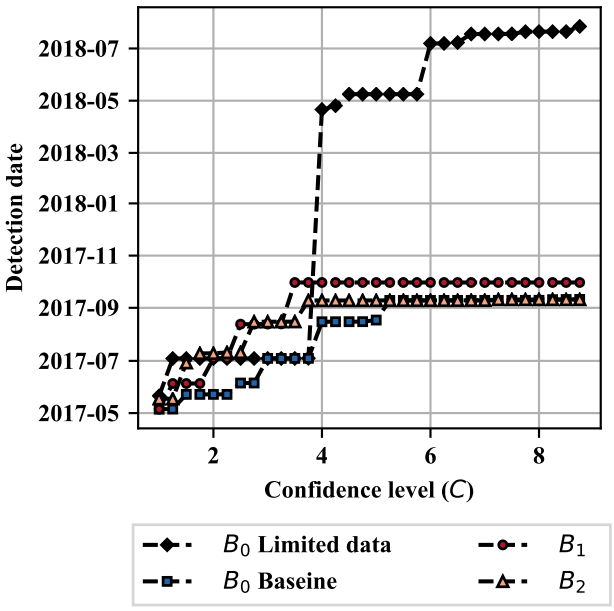

Figure 6. Sensitivity analysis for detection of slowly evolving faults. Dependence of the first detection date on the desired confidence level $C = -\log_{10}\alpha$ for the 4 schemes: Limited data (trained on Turbine B_0), Baseline (trained on Turbine B_0), cross-turbine trained on Turbine B_1, and cross-turbine trained on Turbine B_2.

4. CONCLUSION

We used a new CNN architecture that we developed previously (Ulmer et al., 2020) for early fault detection based on 10-minute SCADA data of wind turbines. The CNN was originally developed for the standard single-turbine fault detection scheme, that is, training it with historical SCADA data from a certain turbine in order to detect faults of the same turbine. Here we extend the usage of this CNN model to a cross-turbine scheme. We train the model on a certain turbine S, and use the trained network for on-line fault detection of a different turbine T from the same wind farm. The cross-turbine algorithm includes a post-processing step of rescaling of residuals in order to compensate for turbine differences. This step requires only a small reference data set from the target turbine T. We tested the performance of the cross-turbine scheme on two fault types in different wind farms: abrupt faults and slow degradation. We showed that:

- The CNN model is able to detect incipient faults reliably and accurately when used in a cross-turbine train-

ing scheme. This means that one can train the CNN on a turbine for which historical data are abundant and use the trained network to detect faults early and with high precision on another turbine in the wind farm for which there are very little data. This is particularly useful for newly installed wind turbines in already existing wind farms.

- Training our CNN model in the cross-turbine scheme shows a considerable improvement in the detection accuracy compared to the single-turbine training with only limited data. At the same time, it shows almost no inferiority to training it with the single-turbine scheme with a large and representative data set. This is observed for diverse fault types: abrupt and slowly evolving in different components (generator or main bearing) and different wind farms. The faults were detected as early and with a similar confidence level and robustness as in the single-turbine-large-data case.
- The cross-turbine scheme works well also across wind farms and target variables: we can train the CNN on a turbine in one wind farm and detect faults with high accuracy in a turbine of a similar model in another farm, even if the CNN was trained to predict a temperature of a different component than the one we detect faults on. We believe that this is a demonstration of the robustness of our CNN model.
- In this work we have demonstrated the potential of a CNN followed by a simple transformation of the residuals to overcome the problem of training data availability for specific turbines or wind farms. In our future research we intend to test the performance of the cross-turbine algorithm on a large set of turbines from various farms and demonstrate the universality of our method, as well as deal with the practically relevant question of selecting an appropriate source turbine. One important advantage of our approach is its simplicity and its low computational load compared to standard transfer learning approaches. As a result, the algorithm is already being used in a commercial software. A detailed comparison with other approaches is the subject of our on-going research.

ACKNOWLEDGMENT

This research was funded by Innosuisse - Swiss Innovation Agency under grant No. 32513.1 IP-ICT.

REFERENCES

Clifton, D. A., Tarassenko, L., McGrogan, N., King, D., King, S., & Anuzis, P. (2008). Bayesian extreme value statistics for novelty detection in gas-turbine engines. In *2008 ieee aerospace conference* (pp. 1–11).

Fu, J., Chu, J., Guo, P., & Chen, Z. (2019). Condition monitoring of wind turbine gearbox bearing based on deep learning model. *Ieee Access*, *7*, 57078–57087.

Guo, J., Wu, J., Zhang, S., Long, J., Chen, W., Cabrera, D., & Li, C. (2020). Generative transfer learning for intelligent fault diagnosis of the wind turbine gearbox. *Sensors*, *20*(5), 1361.

Guo, L., Lei, Y., Xing, S., Yan, T., & Li, N. (2018). Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Transactions on Industrial Electronics*, *66*(9), 7316–7325.

Kong, Z., Tang, B., Deng, L., Liu, W., & Han, Y. (2020). Condition monitoring of wind turbines based on spatio-temporal fusion of scada data by convolutional neural networks and gated recurrent units. *Renewable Energy*, *146*, 760–768.

Leahy, K., Hu, R. L., Konstantakopoulos, I. C., Spanos, C. J., & Agogino, A. M. (2016). Diagnosing wind turbine faults using machine learning techniques applied to operational data. In *2016 ieee international conference on prognostics and health management (icphm)* (pp. 1–8).

Lebranchu, A., Charbonnier, S., Bérenguer, C., & Prévost, F. (2019). A combined mono-and multi-turbine approach for fault indicator synthesis and wind turbine monitoring using scada data. *ISA transactions*, *87*, 272–281.

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, *138*, 106587.

Li, X., Zhang, W., & Ding, Q. (2018). Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks. *IEEE Transactions on Industrial Electronics*, *66*(7), 5525–5534.

Qureshi, A. S., Khan, A., Zameer, A., & Usman, A. (2017). Wind power prediction using deep neural network based meta regression and transfer learning. *Applied Soft Computing*, *58*, 742–755.

Schlechtingen, M., & Santos, I. F. (2014). Wind turbine condition monitoring based on scada data using normal behavior models. part 2: Application examples. *Applied Soft Computing*, *14*, 447–460.

Shao, S., McAleer, S., Yan, R., & Baldi, P. (2018). Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, *15*(4), 2446–2455.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., ... Nenadic, G. (2019). Machine learning methods for wind turbine condition monitoring: A review. *Renewable energy*, *133*, 620–635.

Tautz-Weinert, J., & Watson, S. J. (2016). Using scada data for wind turbine condition monitoring–a review. *IET Renewable Power Generation*, *11*(4), 382–394.

Ulmer, M., Jarlskog, E., Pizza, G., Manninen, J., & Goren Huber, L. (2020). Early fault detection based on wind turbine scada data using convolutional neural networks. In *Proceedings of the european conference*

of the phm society (Vol. 5).

Wang, Q., Michau, G., & Fink, O. (2019). Domain adaptive transfer learning for fault diagnosis. In *2019 prognostics and system health management conference (phm-paris)* (pp. 279–285).

Yun, H., Zhang, C., Hou, C., & Liu, Z. (2019). An adaptive approach for ice detection in wind turbine with inductive transfer learning. *IEEE Access*, *7*, 122205–122213.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, *115*, 213–237.

Zheng, H., Wang, R., Yang, Y., Yin, J., Li, Y., Li, Y., & Xu, M. (2019). Cross-domain fault diagnosis using knowledge transfer strategy: a review. *IEEE Access*, *7*, 129260–129290.