Early Fault Detection Based on Wind Turbine SCADA Data Using Convolutional Neural Networks

Markus Ulmer[1], Eskil Jarlskog[2],Gianmarco Pizza[3],Jaakko Manninen[4], and Lilach Goren Huber[5]

[1,5] *Zurich University of Applied Sciences, Technikumstrasse 9, Winterthur, 8400 Switzerland*
markus.ulmer@zhaw.ch
lilach.gorenhuber@zhaw.ch

[2,3,4] *Nispera AG, Hornbachstrasse 50, CH-8008 Zurich, Switzerland*
eskil.jarlskog@nispera.com
gianmarco.pizza@nispera.com
jaakko.manninen@nispera.com

ABSTRACT

Early fault detection in wind turbines using the widely available SCADA data has been receiving growing interest due to its cost-effectiveness. As opposed to the large variety of fault detection methods based on high resolution vibration data, the use of 10-minute SCADA data alone does not require any additional hardware or data storage solutions and would be immediately implementable in most wind farms. However, the strong variability of these data is challenging and requires significant improvements of existing methods to ensure early and reliable fault detection and isolation. Here we suggest to use Convolutional Neural Networks (CNNs) to enhance the detection accuracy and robustness. We demonstrate the superiority of the CNN model over standard fully connected neural networks (FCNN) using examples for faults with very different time dependent characteristics: an abruptly evolving and a slowly degrading fault. We show that the CNN is able to detect the faults earlier and with a higher accuracy and robustness of prediction than the FCNN model. We then extend the CNN model to a multi-output CNN (CNNm) which provides early fault detection based on a multitude of output variables simultaneously. We show that with the same training time and a similar detection quality as the single output CNN, the CNNm model is an ideal candidate for a practical and scalable fault detection algorithm based on already available 10-minute SCADA data for wind turbines.

1. INTRODUCTION

The importance of wind turbine condition monitoring (CM) for the purpose of early fault detection and isolation (FDI) is widely acknowledged by all stakeholders in the wind energy industry. Existing CM systems can provide high frequency (e.g vibration) data that is then analysed in a multitude of advanced methods (see Stetco et al., 2019 for a recent review). However, such systems require additional sensor installation and are usually a too costly solution to be implemented. On the other hand, Supervisory Control and Data Acquisition (SCADA) systems are already integrated in all wind farms since their first installation, generating large amounts of CM data which is then averaged over 10-minute time windows in order to be cheaply stored for production supervision purposes. Thus, methods for automatic early fault detection and diagnostics which rely on 10-minute SCADA data and yet provide reliable and accurate information, are currently gaining interest in the wind energy community (Lebranchu et al., 2019; Tautz-Weinert & Watson, 2016). The work presented here focuses on using 10-minute SCADA data exclusively.

Compared to the high rate CM data, the 10-minute averages suffer from the averaging out of the sub-10-minute time dependencies. This poses major challenges for FDI algorithms based on this data. Various data-driven methods have been applied for this purpose in the past (Schlechtingen & Santos, 2014; Leahy et al., 2016). Many of these approaches are based on modelling the normal behaviour of the turbines and analysing deviations from normality in real time to detect incipient faults. In particular, several neural network (NN) models have been applied (Garcia et al., 2006; Zaher et al., 2009; Li et al., 2014; Z. Zhang, 2014) to detect faults in one component of the turbine drivetrain or more. Up to now, all NN models using 10-minute data are of the fully connected type (FCNN), also known as Multi-Layer-Perceptrons (MLP). These models are typically local in time and do not make use of time-dependent structures or correlations that are obviously abundant in the time-series SCADA data. Exploit-

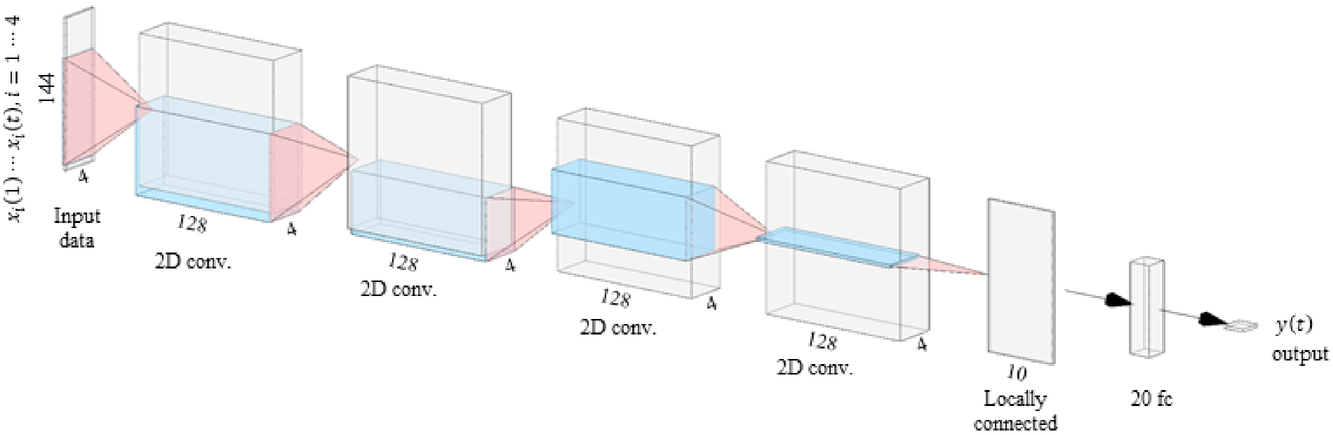

Figure 1. CNN architecture.

ing such time-dependent information is commonly done for FDI in other application fields by means of Convolutional Neural Networks (CNNs) or Recurrent Neural Networks. This has been explored extensively in recent reviews of machine learning applications in prognostics and health management (Khan & Yairi, 2018; Zhao et al., 2019; Hoang & Kang, 2019; Lei et al., 2020), with references therein for an abundance of network architectures and models. In most cases, both 1D- and 2D-CNN models have been applied to high resolution vibration or acoustic spectrum data, as in several recent examples (Zhu et al., 2019; Han et al., 2018; Peng et al., 2018; W. Zhang et al., 2018; Pan et al., 2017; Yang et al., 2019), and only rarely to other data types. This is true also in the case of fault detection for wind turbines, see for example Jiang et al., 2018; Bach-Andersen et al., 2018. In addition, CNNs have recently been applied for 1-minute SCADA data of wind turbines (Fu et al., 2019; Kong et al., 2020). However, to the best of our knowledge the advantage of CNNs has not yet been demonstrated for fault detection using the readily available 10-minute data.

In this paper we show that the use of CNNs can be profitable for practical applications of fault detection, that is, with 10-minute SCADA data. We demonstrate the superiority of the CNN compared to MLP models for fault prediction of various fault types on several different turbine components, owing to its ability to account for complex time dependent multivariate correlations. The high availbility of the 10-minute SCADA data allows for an immediate training of the CNN with abundant historical data for most of the operating wind turbines nowadays, and therefore for an efficient and scalable deployment of the algorithm in practice. Our focus in this paper is on comparison amongst different NN models. Showing their superiority with respect to other machine learning algorithms for fault detection is beyond the scope of this paper.

We then extend the CNN model to a multi-output architecture which allows simultaneous fault detection on a multitude of components with a similar performance to the single output CNN but no additional training time. We stress the generic nature of the model, based on a small number of inputs with a generic framework for post-processing, yet providing detections of a wide and diverse scope of faults. This, together with the performance and scalability of this model, make it attractive for practical implementations, providing not only fault detection, but also enabling fault localization and diagnosis. Another important aspect of this model is that it uses a multi-output architecture in a prediction rather than classification setup, thereby differing from the CNN models mentioned in the references above for fault diagnosis applications. Here we focus on the drivetrain system, where many critical faults tend to develop. However, our approach is generic and we believe it can be easily extended beyond the drivetrain in the future.

2. CNN FOR FAULT DETECTION

As in many practical applications of fault detection, critical faults in wind turbines are rare and of very diverse nature. We therefore adopt the common approach of modelling the normal (or healthy) behaviour and detecting deviations from normality in real time in order to detect incipient faults. In particular, we apply a family of NN models which regress a target variable on a multivariate input. The target variable y is typically a temperature of a certain turbine component, e.g a generator bearing, the gearbox oil or the hub temperature. The input is chosen to include a minimal set of measured variables x_1, x_2, x_3, x_4, which are believed to serve as effective predictors independent of the fault type: output power, ambient temperature, wind spped and rotor rpm. We train the models using data from a specific turbine, measured during normal behaviour (healthy data). During training, the NN is trained to minimize the prediction errors, defined as the absolute difference between the predicted and the measured target variable. Provided with enough representative data, the

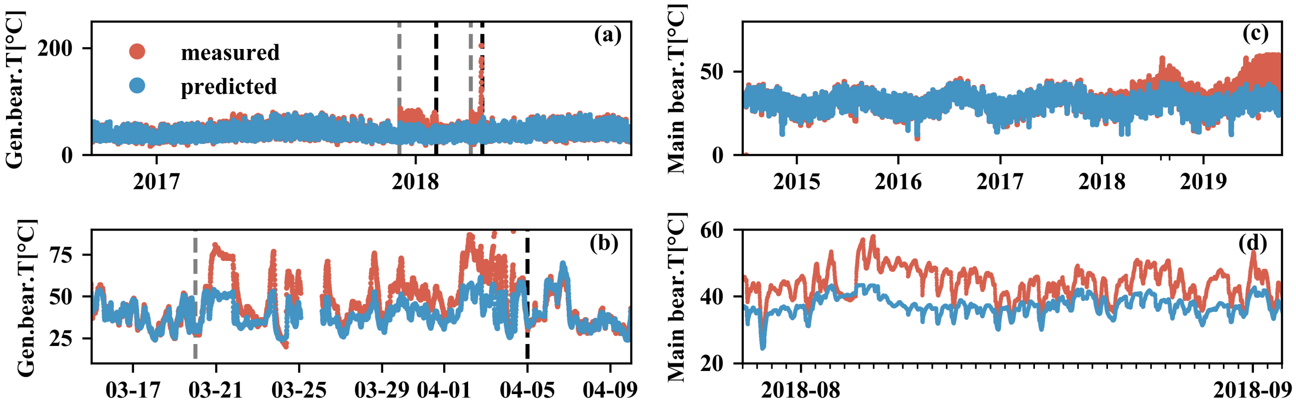

Figure 2. Predicted and observed target variables. (a) Generator bearing temperature in turbine I. Grey dashed: true fault initiation. Black dashed: true turbine stoppage. (b) Zoom-in of (a). (c) Main bearing temperature in turbine II. (d) Zoom-in of (c).

NN 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 "health indicators (HI)" for an early detection of incipient faults. Below we describe the architecture, input and output of the CNN model we use for prediction.

2.1. Network Description

We train a CNN which receives multivariate input sequences of dimension 4x144 corresponding to 4 measured variables over a time period of 1 day (consecutive 144 values of 10 minute averages): $x_i(1) \dots x_i(t)$ with $i = 1 \dots 4$. The input sequences are generated with a sliding window with a maximal overlap. The architecture we use has 4 convolutional layers, each containing 128 2D filters of lengths 32,18,8,8 in the time dimension and width 4 (over all four inputs), see Fig. 1.

Each layer is followed by batch normalization and a spatial dropout with rate 0.1. After an additional locally connected convolution layer the representation is flattened to a fully connected layer of 20 neurons from which a single output is extracted. The output $\hat{y}(t)$ is the regression target variable at the end of the 1 day period of the input sequence ("many to one" prediction configuration). The loss function is the squared error between the predicted output and the measured value $y(t)$, $L = |y(t) - \hat{y}(t)|^2$.

The choice of 2D convolutional layers was done after comparison of performance with a 1D-CNN, which resulted in superiority of the 2D filters. The advantage of the 2D convolution is that it is able to capture correlations between different input variables in a more effective way. A detailed feature analysis of the CNN model is however beyond the scope of this paper.

To evaluate the model performance we compare it to the commonly used FCNN (or MLP), which predicts the target variable $y(t)$ based on equal time inputs $x_1(t), x_2(t), x_3(t), x_4(t)$. It includes two FC hidden layers each of dimension 20. We train, validate and test each of the NN models with the same data sets, and apply learning rate adaptation, early stopping criteria and identical post-processing steps to both. Variations in the MLP hyperparameters (network depth, number of neurons, batch size, optimizer and learning rate) did not change our conclusions significantly.

2.2. Results and Discussion

As described above, the output of the CNN model is a predicted value $\hat{y}(t)$ every 10 min. We then construct the time series of prediction errors (residuals) $\delta(t)$ by subtracting $\hat{y}(t)$ from the observed value $y(t)$ of the target variable:

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

Since the model is trained on healthy data, it is expected to yield accurate predictions as long as the turbine state is normal (see for example Li et al., 2014). We thus expect small prediction errors in healthy times and large (absolute) prediction errors $|\delta(t)| > 0$ when the behaviour of the system deviates from normality. The goal is to detect deviations from normality as early and accurately as possible. We note that based on domain knowledge we do not expect critical faults to lead to temperature reduction, but rather to temperature increase. We thus aim at detecting the onset time of faults with a large positive $\delta(t)$. To achieve this, we post-process the prediction errors to obtain HIs for the target variable. The post processing is aimed at signal-to-noise enhancement and includes a high output power filter, followed by a 5 hours sliding window aggregation. As a result, we remain with (at most) one value of the HI, $h(t_w)$, every hour corresponding to the post processed prediction errors $\delta(t)$ for $t_w - 1 < t < t_w$.

The next step is to determine the threshold h_c, where we define a fault through of $h(t_w) > h_c$. There are various approaches to the problem of threshold setting. Here we aim at maximally automizing this procedure and allowing the user for transparency regarding the physical meaning of the detection threshold. A similar approach was taken for example in Clifton et al., 2008. To this end, we estimate the distribution of prediction errors of the (normal) validation set, assuming for simplicity a Gaussian distribution of the residuals with the estimated mean μ and variance σ^2. We then calculate for each time window t_w in the test set its cumulative distribution function $P(h(t_w); \mu, \sigma)$ and its p-value $1 - P(h(t_w); \mu, \sigma)$, which is the probability to obtain this HI or higher if the point corresponds to a healthy state (the null hypothesis in this case). We then set a threshold in terms of a desired significance level α and declare a point as faulty if its p-value is smaller than α, or equivalently if $P(h(t_w); \mu, \sigma) \geq 1 - \alpha$. In the next section we interprete the significance level as a confidence score, such that a given significance threshold α corresponds to a certain desired confidence of the fault detection.

Below we present two examples for early fault detection using the described algorithm. The two examples represent two very different fault evolutions which are both commonly observed on wind turbines: abrupt faults and slowly evolving faults. In both cases we compare the results of the CNN model to the ones of the fully connected MLP network described in Section 2.1 above. For the comparison we use both models to predict the same target variable and perform an identical post-processing of the prediction errors.

2.2.1. Detecting Abrupt Faults

Figure 2(a) shows the raw measured temperature and the predicted values of the generator bearing of wind turbine I over a period of about 2 years, with a zoom-in on a period of about one month in panel (b) for the sake of clarity. The first 9 months were used for training and validation and the rest for testing. Figure 3 shows the resulting generator bearing temperature HI $h(t_w)$ in degrees Celsius for each window (some time windows were dropped on post-processing). Figures 3(a) and (b) show the results for the MLP and the CNN networks respectively. The threshold for detection was set to $\alpha = 0.01$. All red coloured points in the plots indicate a detected faulty behaviour. For this turbine we could obtain "true labels" from the operator, indicating the onset and actual detection time of two faults by the staff on site. The first one (f_1) started showing up 9.12.2017 and caused a turbine stoppage on the 30.1.2018. The second fault (f_2) started 20.3.2018 and lead to a stoppage on the 5.4.2018. The figure shows that the two faults would be successfully detected by both models, the MLP and the CNN, several weeks prior to the stoppage. However, it can also be seen that the variance of the HI of the MLP is considerably higher than with the CNN. As a result, using the CNN, the faults are more clearly distinguishable from the healthy periods. This advantage of the CNN 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 labelled as "healthy". Optimality corresponds to detecting a maximal fraction of the faulty time windows with minimum false positives. We introduce the following performance measures:

1. Time of first detection. Earliest time window t_w for which
$$h(t_w) > h_c \tag{2}$$
2. Recall: what fraction of "faulty" time windows are identified as such (True Positives)?
$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \tag{3}$$
3. Precision: reflects the fraction of False Positives (detected as "faulty" but are actually "healthy").
$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} \tag{4}$$
4. F_1 score:
$$F_1 = 2 \cdot \frac{\mathbf{Recall} \cdot \mathbf{Precision}}{\mathbf{Recall} + \mathbf{Precision}} \tag{5}$$
5. The MSE ratio between faulty and healthy times, defined as:
$$\mathbf{MSE\ ratio} = \frac{\mathbf{MSE}\{\mathcal{F}\}}{\mathbf{MSE}\{\mathcal{H}\}} \tag{6}$$
Where $\mathcal{H}$ and $\mathcal{F}$ are faulty and healthy validation sets of the same size. Note that the MSE is calculated with the bare prediction errors before post-processing. This score measures the distinguishability of faulty from healthy behaviour.

Table 1 summarizes the model comparison based on the above measures with a detection threshold of $\alpha = 0.01$. Since the fault of this type seemed to develop rather abruptly, the time of first detection is similar for the two models (below we show that this is very different for a slowly evolving fault): the CNN detects the fault only slightly earlier. The MSE ratio of the CNN is significantly higher and as a result its detection is much clearer. This is reflected in higher recall, precision and F_1 scores for the CNN. All together we conclude that the CNN could have detected the two faults f_1 and f_2 more precisely and reliably than a simple MLP.

The performance scores are naturally dependent on the thresh-

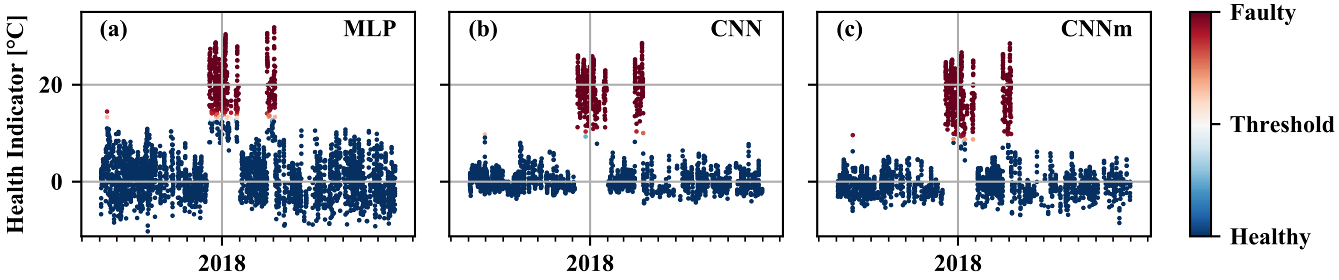

Figure 3. Health indicators for detection of abrupt faults in turbine I using (a) MLP (b) CNN and (c) CNNm

Table 1. Model comparison for abrupt faults

Model	First detection	MSE ratio	Recall	Prec.	F_1
MLP	9.12.17 23:20	11.9	0.94	0.88	0.91
CNN	9.12.17 22:20	32.0	0.96	0.98	0.97
CNNm	9.12.17 22:20	18.1	0.96	0.98	0.97

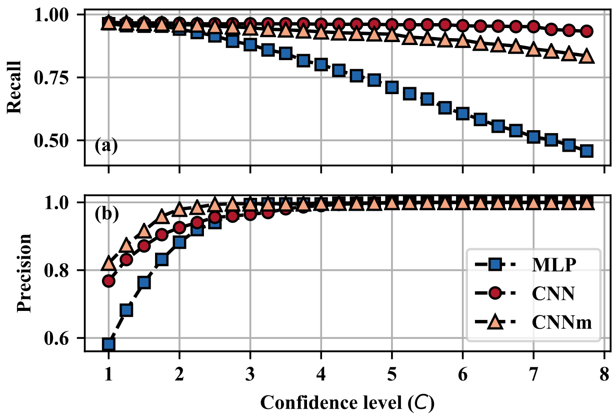

Figure 4. Sensitivity analysis for abrupt fault detection. Dependence of the (a) Recall and (b) Precision scores on the desired confidence level $C = -\log_{10}\alpha$ for the 3 models MLP, CNN and CNNm.

old setting. A higher detection threshold leads to a higher precision and a lower recall rate. This tradeoff is demonstrated in Fig. 4. The figure analyses the sensitivity of the scores to the confidence level C, defined in terms of the significance threshold α as

$$C = -\log_{10}\alpha \qquad (7)$$

A smaller α corresponds therefore to an exponentially higher confidence of fault detection. The user can select a desired confidence level C and detect a fault with a significance level of $\alpha = 10^{-C}$. Here it is clearly seen that for any choice of α, the CNN (red circles) has considerably higher recall and precision rates than the MLP (blue squares). Moreover, the CNN model is significantly less sensitive to the choice of α (or the desired confidence) than the MLP, thus providing more reliable fault detection results.

2.2.2. Detecting Slow Degradation

In order to demonstrate the generic detection abilities of the CNN architecture we discuss a second example of turbine II in a different wind park. Here the time evolution of the early fault indicators 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 park operator in retrospect. Below we demonstrate the advantage of using a CNN instead of a standard MLP for an early and reliable fault detection.

The NNs were trained with a period of 10 months and tested on over 4 years of data, with the generic set of input variables described above in 2. The measured values of the target variable, in this case the main bearing temperature, are shown in Fig. 2(c), together with the CNN model predictions. Panel (d) of this figure displays a zoom-in on a period of one month for the sake of clarity. The post processing steps described above in Section 2.2 are identical for all turbines and all models, and yield a HI for the target variable. Figure 5(a) and (b) display the HI $h(t_w)$ of the MLP and CNN models respectively. The signal to noise level of $h(t_w)$ is clearly higher for the CNN. In the absence of true labels for the period before the stoppage, we measure the model performance in terms of the time of first detection of faulty condition, with a detection threshold of $\alpha = 0.0001 (C = 4)$, see Table 2. With this threshold the CNN detects the fault two months earlier than the MLP.

Table 2. Model comparison for slowly evolving faults

Model	First detection ($\alpha = 0.0001$)	First detection high confidence
MLP	16.10.17 2:00	25.4.18 00:00
CNN	15.8.17 17:00	10.9.17 19:00
CNNm	15.8.17 17:00	9.9.17 14:00

A detailed sensitivity analysis of the first detection date with respect to the confidence level C (or equivalently the thresh-

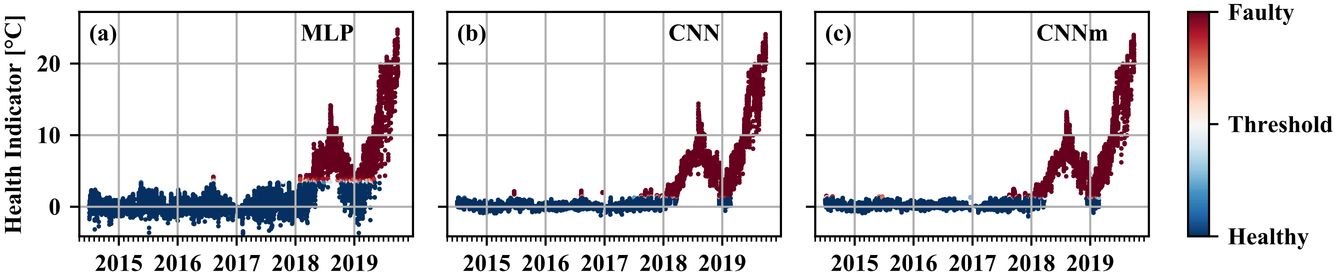

Figure 5. Health indicators for detection of slowly evolving faults in turbine II Using (a) MLP (b) CNN and (c) CNNm.

old value $\alpha = 10^{-C}$ is shown in Fig. 6(a). Higher confidence levels naturally lead to a later (but more certain) detection. This is true for all models. However, it is seen that the CNN detects the fault considerably earlier for any choice of confidence level (threshold value). Moreover, the detection date of the CNN is much more stable with respect to changes in the threshold: it is postponed by less than a month if we require a confidence of $C = 10$ instead of $C = 4$, whereas the MLP detection is delayed by 6 months (see right column in Table 2). This stability turns the CNN into a more practical solution for robust and generic fault detection.

In this section we showed that our CNN architecture performs considerably better than a standard MLP network for detection of faults of two different natures; abrupt and slowly degrading. In the next section we extend the CNN architecture to provide simultaneous fault detection on multiple sensor outputs and show that the performance of the multi-output CNN is similar to the single output one.

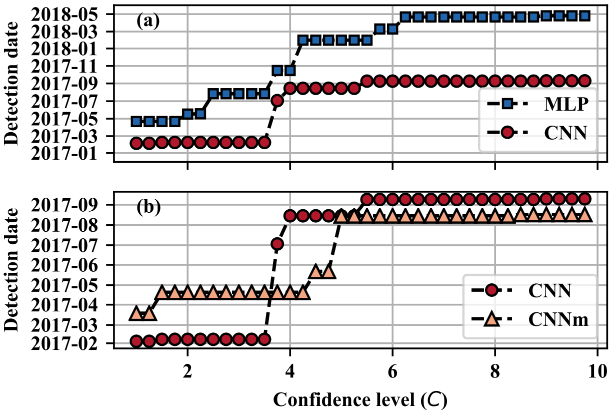

Figure 6. Sensitivity analysis for detection of slowly evolving faults. Dependence of the first time of fault detection on the desired confidence level $C = -\log_{10}\alpha$ for the 3 models: (a) MLP vs. CNN (b) CNN vs. CNNm.

3. MULTI-OUTPUT CNNS

The CNN model described in Section 2 above produces a HI from a single target variable. As such, it can detect fault types that manifest themselves as deviating values of this variable. In practice, the drivetrain of wind turbines is prone to a multitude of very diverse faults, which may be observed as temperature abnormalities of various sensors. Since each fault type is extremely unique and rare, the possibility of training a classifier that can distinguish between them is not realistic. Instead, a practical approach is to first detect deviations from normal behaviour and then characterize them in order to diagnose the incipient fault type. This can be done by repeating the prediction with a multitude of CNN models, each with its own target variable. Then, combining the resulting HIs can help in fault detection, localization and diagnostics.

Here we suggest an alternative to this approach, which has superior scalability properties, but a similar quality of fault detection and isolation. To this end we train a single CNN with N_{out} outputs, each trained to predict a different target variable. We denote this architecture as CNNm for multi-output. In the following we describe the difference of the CNNm compared to the CNN model of Section 2 and compare their performance for the use-cases of the two turbines as before.

3.1. Network Description

The architecture of the CNNm model is very similar to the one of the CNN, depicted in Fig. 1 above. The main difference is the number of neurons in the output layer which is 1 for the CNN and N_{out} for the CNNm. We train the CNNm with the same input sequences, and validate and test with the same data sets as for the CNN. We then process the prediction errors for each output variable independently in an identical way to the one described above for the CNN. It is important to note that the training time of the CNNm with any number of outputs is only slightly longer than the one of the CNN with a single output. We assign this fact to the structural similarity and the strong correlations between the outputs, that allow the network to easily adjust its weights for simultaneous predictions of all outputs. These similarities might even

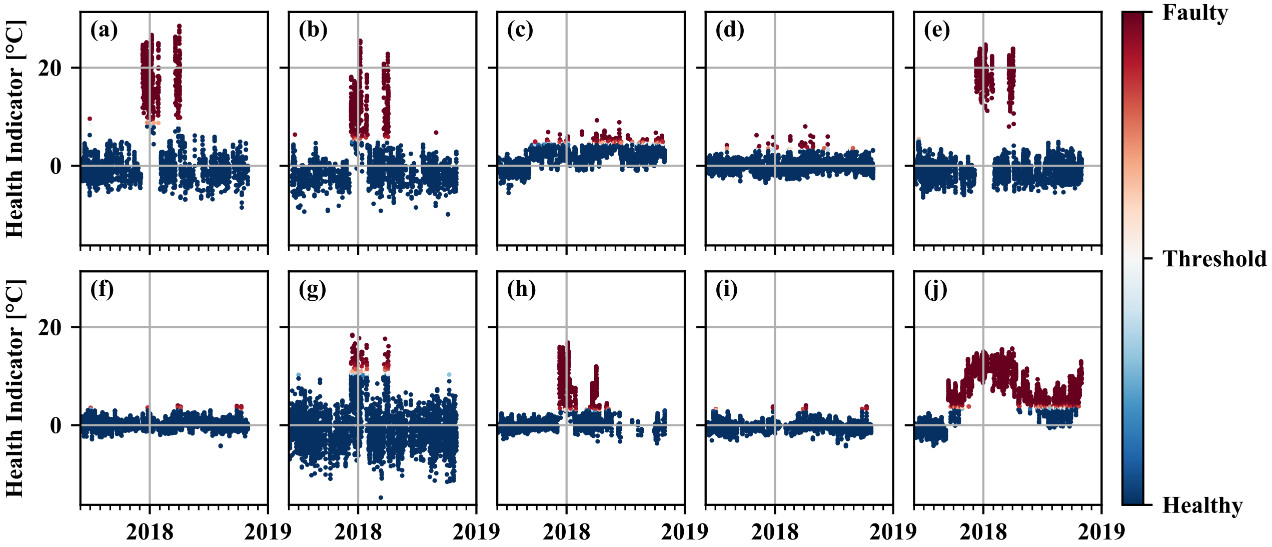

Figure 7. Health indicators for 10 component temperatures predicted simultaneously by the CNNm model for turbine I. (a) Generator bearing (b) stator phase (c) gearbox bearing (d) gearbox oil (e) slip ring (f) rotor spinner (g) grid transformer (h) controller top (i) controller hub (j) hydraulic oil.

have a stabilizing effect on the learning process of the CNNm compared to the CNN. Below we show that the fault detection performance of these two models is very similar, both for abrupt faults and for slowly evolving ones.

3.2. Results and Discussion

3.2.1. Detecting Abrupt Faults

Figure 7 shows the HIs calculated from 10 different output variables (see figure caption for variable names) of turbine I. Each HI was generated by repeating the same post-processing steps described in section 2.2. The threshold level was selected to be $\alpha = 0.01$. Interestingly, several variables from different sub-systems show similar faulty behaviour whereas others show no fault or different fault onset and nature. A detailed analysis of the multi-output HIs can lead not only to an early detection but also to an accurate fault isolation and diagnostics. Note that despite the generic choice of a confidence level for all outputs, each threshold is set based on the validation prediction errors of the individual variable. Alternatively, one could adjust the confidence level individually for each output according to the desired detection sensitivity.

In order to compare the performance of the CNNm with the single output CNN model we select the generator bearing temperature as a target variable. Figure 3(c) shows the generator bearing temperature HI $h(t_w)$ for the CNNm model. This can be compared to the previously discussed outcomes of the MLP (panel (a)) and CNN (panel (b)). The signal to noise level of the CNNm model is clearly close to the one of the CNN model, although the CNNm provides HIs for 9 additional output variables simultaneously with almost no change in training time. This observation is supported by the performance scores of the CNNm in Table 1, which are very similar to the ones of the CNN. This similarity can be explained if we assume that the CNNm exploits structural similarities and correlations between the various outputs to efficiently adjust the weights in the convolutional layers. A further investigation of the feature maps to support this hypothesis will be pursued in a separate research.

A further model comparison is displayed in Fig. 4, showing the sensitivity of the recall and precision scores towards changing the confidence level C. Here as well, we observe only a slight inferiority in performance of the CNNm (yellow triangles) compared to the single output CNN (red circles) for any given confidence level. Both models are clearly better performing and more robust than the MLP model (blue squares). In particular, it is easier to find a confidence level for which both recall and precision are high and stable.

Our results imply that instead of training a multitude of single output models (one for each target variable), it is clearly advantageous to train a single multi-output CNN, predicting all variables at once: with no additional training time we achieve similarly high performance of early fault detection, and at the same time allow for better understanding of the root cause (fault isolation and diagnostics).

3.2.2. Detecting Slow Degradation

We test the performance of the multi-output CNNm model also for an example of a slowly evolving fault as in the case of the main bearing temperature of turbine II. Here as well, we calculate the HIs of 8 output variables simultaneously and select the main bearing temperature to compare with the CNN and the MLP models. The HIs are displayed in Fig. 5(c). The variance of the prediction errors (indicative of the signal to noise level) is close to the one of the CNN 5(b) and much higher than that of the MLP model 5(a). As in Section 2.2.2, in the absence of true labels we compare the time of first fault detection of the three models. This can be seen in Table 2 for a confidence level of $C = 4(\alpha = 0.0001)$.

A more detailed model comparison is again in terms of the sensitivity towards the detection threshold (or confidence level), seen in Figure 6(b). Both the CNN (red circles) and the CNNm (yellow triangles) models are very stable against changing the threshold and allow for a similarly early detection for high confidence levels of $C \leq 4$. In particular, at an asymptotically high confidence ($C > 6$), both CNN networks would detect the fault at least 8 months before the MLP (blue squares). On the other hand, a lower confidence detection ($C < 4$) is enabled earlier with the single output CNN than with the multi-output model.

4. CONCLUSIONS

We developed two new CNN architectures for early fault detection using 10-minute SCADA data of wind turbines: a single output and a multi-output CNN. We developed a generic and unified scheme for post-processing of the prediction errors to extract health indicators for different sensor variables. We introduced an algorithm for the automatic threshold setting, given a desired confidence level of the detection. We then compared the performance of the CNN to the one of fully connected MLP networks, commonly used with 10-minute SCADA data. We showed that:

- Both CNN models are superior to the standard MLP model in detecting faults of different character (abrupt and slowly degrading) and of various root causes, observed through various SCADA data variables with 10-minute time resolution. To the best of our knowledge this is a first demonstration of the benefit of CNNs for fault detection based on 10-minute SCADA data. Moreover, the demonstration of a successful detection algorithm for faults of very different nature is one of the model strengths.
- Both CNN models are considerably more stable against a variation of the detection threshold, which is in our case also a confidence level of the detection. As such they have a major advantage in the practical implementation. Our sensitivity analysis provides an effective generic model comparison tool for practical implementation purposes.
- The multi-output CNN model enables simultaneous detection of faults in multiple sensor variables and therefore allows for fault isolation and diagnostics in addition to fault detection. This generic nature of fault detection is enabled owing to our modelling approach which avoids any assumptions regarding the nature or localization of the faults.
- The multi-output CNN requires almost no compromise on the detection performance and no prolongation of the training time compared to the single output CNN. It is therefore an optimal practical and scalable solution for high confidence fault detection and diagnostics for wind turbines based on already available 10-minute SCADA data.

In the paper we focussed on a comparison between a widely used fully connected NN and newly developed CNN models. The comparison with other fault detection algorithms is beyond the scope of the present work. In this work we presented fault examples out of selected wind turbines for which we had confirmed fault information. In our future research we will extend the testing of the models to a larger set of turbines out of various wind farms in order to show the scalability of our model.

ACKNOWLEDGMENT

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

REFERENCES

Bach-Andersen, M., Rømer-Odgaard, B., & Winther, O. (2018). Deep learning for automated drivetrain fault detection. *Wind Energy*, *21*(1), 29–41.

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.

Garcia, M. C., Sanz-Bobi, M. A., & Del Pico, J. (2006). Simap: Intelligent system for predictive maintenance: Application to the health condition monitoring of a windturbine gearbox. *Computers in Industry*, *57*(6), 552–568.

Han, Y., Tang, B., & Deng, L. (2018). Multi-level wavelet packet fusion in dynamic ensemble convolutional neural network for fault diagnosis. *Measurement*, *127*, 246–255.

Hoang, D.-T., & Kang, H.-J. (2019). A survey on deep learning based bearing fault diagnosis. *Neurocomputing*, *335*, 327–335.

Jiang, G., He, H., Yan, J., & Xie, P. (2018). Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox. *IEEE Transactions on Industrial Electronics*, *66*(4), 3196–3207.

Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, *107*, 241–265.

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, J., Lei, X., Li, H., & Ran, L. (2014). Normal behavior models for the condition assessment of wind turbine generator systems. *Electric Power Components and Systems*, *42*(11), 1201–1212.

Pan, J., Zi, Y., Chen, J., Zhou, Z., & Wang, B. (2017). Liftingnet: A novel deep learning network with layerwise feature learning from noisy mechanical data for fault classification. *IEEE Transactions on Industrial Electronics*, *65*(6), 4973–4982.

Peng, D., Liu, Z., Wang, H., Qin, Y., & Jia, L. (2018). A novel deeper one-dimensional cnn with residual learning for fault diagnosis of wheelset bearings in high-speed trains. *IEEE Access*, *7*, 10278–10293.

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.

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. (2016). Using scada data for wind turbine condition monitoring–a review. *IET Renewable Power Generation*, *11*(4), 382–394.

Yang, Y., Zheng, H., Li, Y., Xu, M., & Chen, Y. (2019). A fault diagnosis scheme for rotating machinery using hierarchical symbolic analysis and convolutional neural network. *ISA transactions*, *91*, 235–252.

Zaher, A., McArthur, S., Infield, D., & Patel, Y. (2009). Online wind turbine fault detection through automated scada data analysis. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, *12*(6), 574–593.

Zhang, W., Li, C., Peng, G., Chen, Y., & Zhang, Z. (2018). A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mechanical Systems and Signal Processing*, *100*, 439–453.

Zhang, Z. (2014). Comparison of data-driven and model based methodologies of wind turbine fault detection with scada data. *EWEA March*.

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.

Zhu, Z., Peng, G., Chen, Y., & Gao, H. (2019). A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis. *Neurocomputing*, *323*, 62–75.