



# Overview of normal behavior modeling approaches for SCADA-based wind turbine condition monitoring demonstrated on data from operational wind farms

Xavier Chesterman<sup>1</sup>, Timothy Verstraeten<sup>1</sup>, Pieter-Jan Daems<sup>2</sup>, Ann Nowé<sup>1</sup>, and Jan Helsen<sup>2</sup>

<sup>1</sup>Artificial Intelligence Lab, Vrije Universiteit Brussel, Pleinlaan 9, 3rd floor, 1050 Brussels, Belgium

<sup>2</sup>AVRG, Vrije Universiteit Brussel, Pleinlaan 3, 1050 Brussels, Belgium

**Correspondence:** Xavier Chesterman (xavier.chesterman@vub.be)

Received: 22 December 2022 – Discussion started: 21 February 2023

Revised: 21 April 2023 – Accepted: 4 May 2023 – Published: 5 June 2023

**Abstract.** Condition monitoring and failure prediction for wind turbines currently comprise a hot research topic. This follows from the fact that investments in the wind energy sector have increased dramatically due to the transition to renewable energy production. This paper reviews and implements several techniques from state-of-the-art research on condition monitoring for wind turbines using SCADA data and the normal behavior modeling framework. The first part of the paper consists of an in-depth overview of the current state of the art. In the second part, several techniques from the overview are implemented and compared using data (SCADA and failure data) from five operational wind farms. To this end, six demonstration experiments are designed. The first five experiments test different techniques for the modeling of normal behavior. The sixth experiment compares several techniques that can be used for identifying anomalous patterns in the prediction error. The selection of the tested techniques is driven by requirements from industrial partners, e.g., a limited number of training data and low training and maintenance costs of the models. The paper concludes with several directions for future work.

## 1 Introduction

In recent years, investments in renewable energy sources like wind and solar energy have increased significantly. This is the result of goals set in climate change agreements and changes in the geopolitical situation. According to the *Global Wind Report 2022*, an additional 93.6 GW of wind energy production capacity was installed in 2021. This brings the total to 837 GW, which corresponds to a 12 % increase compared to the previous year (Lee and Zhao, 2022). To keep the transition on track, the profitability of the investments needs to be guaranteed. Furthermore, to keep the European economy competitive in the globalized market, the price of energy production using wind turbines needs to be kept as low as possible. Both depend to a large extent on the maintenance costs.

According to Pfaffel et al. (2017), recent studies have shown that the operation and maintenance of wind turbines

make up 25 %–40 % of the levelized cost of energy. A more detailed analysis shows that premature failures due to excessive wear play a considerable role. These are caused by, among other things, high loads due to environmental conditions and aggressive control actions (Verstraeten et al., 2019; Tazi et al., 2017; Greco et al., 2013). If it were possible to identify these types of failures well in advance, it would create the opportunity to avoid unexpected downtime and organize the maintenance of turbines more optimally. This in turn would result in increased production and a further reduction in maintenance costs, which would improve the profitability of the investments and reduce wind energy prices.

This paper gives an overview of the current state of the art on condition monitoring for wind farms using SCADA data and the normal behavior modeling (NBM) framework. The focus on SCADA data is motivated by the fact that they are an inexpensive source of information that are readily available. This is valuable in an industrial context where adding new

sensors is expensive and not straightforward. The focus on the NBM methodology can be justified by the fact that it has shown its merits and that properly trained NBM models can result in interesting engineering insights. Several techniques used in the state-of-the-art research are also implemented and compared. For this, 10 min SCADA data from five different wind farms are used. Furthermore, failure information is also available for these wind farms. More specifically, there is information on generator bearing, generator fan, and rotor brush high-temperature failures. The different techniques are implemented and compared using six demonstration experiments. Five experiments focus on NBM, and one focuses on the analysis of prediction error. By conducting a comparative analysis and discussion of the results on real data, a better understanding can be achieved of the performance of different techniques applied to real data. Because an exhaustive overview is unfeasible, only a limited selection can be discussed. This selection is based on several assumptions, e.g., that there are only a relatively limited number of training data and a limited amount of time and that due to maintenance constraints the complexity of the methodology needs to be kept as low as possible. These constraints are based on feedback received from several industrial partners.

The paper is built up as follows. The first section is the Introduction. The second section discusses the current state of the art. In the third section, an experimental methodology is designed that combines, compares, and demonstrates the performance of several techniques mentioned in the state-of-the-art overview. The fourth part is the comparative analysis of several techniques from the state of the art. The fifth and last part is the conclusion, which also includes a discussion of possible future directions for research.

## 2 Overview of the state of the art

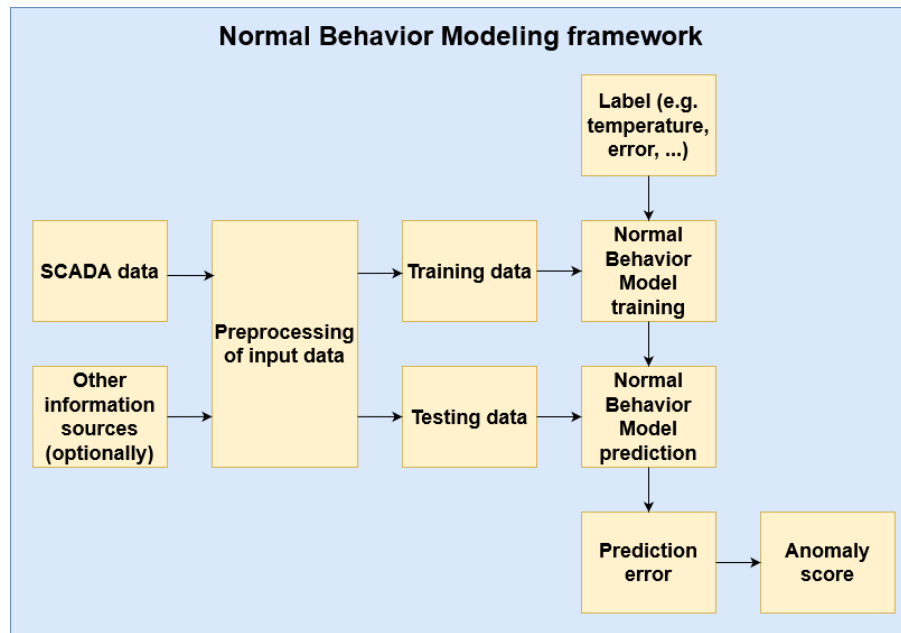
Failure prediction on wind turbines using SCADA data is a hot research topic. This is due to the fact that over time more sensor data have become available (Helsen, 2021). There are several different families of methodologies that compete in this domain. According to Helbing and Ritter (2018), the methodologies can be divided into model-based signal processing and data-driven methods. An alternative classification can be found in Black et al. (2021), where a distinction is made between (1) trending, (2) clustering, (3) NBM, (4) damage modeling, (5) alarm assessment, and (6) performance monitoring. In Tautz-Weinert and Watson (2017) five categories are identified: (1) trending, (2) clustering, (3) NBM, (4) damage modeling, and (5) assessment of alarms and expert systems.

The NBM methodological family is very diverse. Many different algorithms can be used to model the “normal” or “healthy” behavior of a wind turbine signal. However, in all this diversity, there are several commonalities. Figure 1 gives an overview of a standard NBM flow. The SCADA

data are ingested by the pipeline. The first step is splitting the data into a training and testing dataset. This is done prior to the preprocessing and modeling steps to avoid “information leakage”. The testing dataset is a random subsample that is set aside for the final validation of the methodology and should not be used during training. How this split is made depends on the type and the number of data. Often 80 %–20 % or 70 %–30 % random splits are made. However, other options are possible. If the data are time series, which is the case when using SCADA data, and the models used as NBM use lagged predictors, then the train–test split should be done more carefully so that the relation between the target and the lagged predictors is not broken. A possibility is assigning the first 70 % of the observations (based on their timestamp) to the training and the rest to the testing dataset. In the next step, the data are preprocessed. This is done to clean the signals (e.g., removing measurement errors, filling in missing values) and in some cases to reduce the noisiness of the signals (e.g., binning). Filtering is sometimes used if it is expected that the relation between the signals is influenced by certain other factors (e.g., wind turbine states). The preprocessing is done on the training and testing dataset separately. However, the same techniques are used on both datasets.

In the next step, the training dataset is used to learn or train the normal behavior model. For this, a health label of some kind is required. In the case of SCADA-based anomaly detection for wind turbines, this is in general a temperature signal that is related to the failure that needs to be detected. For example, if the research is focused on predicting generator bearing failures, then the label might be the temperature of a generator bearing. This means that it is a supervised regression problem. Many different algorithms are suitable as NBM models, e.g., ordinary least squares (OLS), random forest (RF), support vector machine (SVM), neural network (NN), and long short-term memory (LSTM). The NBM model is trained on healthy data (meaning not polluted with anomalies that can be associated with a failure). Once the NBM model has been trained, it can be used for predicting the expected normal behavior using the test dataset. In the next step, the difference between the predicted and the observed behavior is analyzed. If there is a large deviation between the two, this can be considered evidence of a problem. The deviation is in general transformed into an anomaly score that says something meaningful about for example the probability of failure or the remaining useful life (RUL).

In what follows an overview is given of different techniques that are used in the state-of-the-art literature for each step of the pipeline. The papers that will be discussed in this section have the following properties: firstly, they are based on wind turbine SCADA data; secondly, they perform condition monitoring and anomaly detection on temperature signals of the turbine (this excludes for example research that focuses on the power curve); and thirdly, they follow the NBM methodology. By limiting the scope of the overview,



**Figure 1.** Schematic overview of NBM framework.

it can be more exhaustive and give the reader a better insight into what has been tried in the literature.

## 2.1 Preprocessing techniques

Preprocessing is an important, although often somewhat underexposed, part of the NBM pipeline. Decisions taken during this step can influence the training and performance of the NBM models later on. Different preprocessing techniques exist and have been used in recent research. The choice of a technique is to a certain extent guided by the properties of the input data; e.g., for time series data, the order of the data points, as well as the relation between them, is relevant. This means that only preprocessing techniques that retain this property of the data should be used. But even then multiple preprocessing techniques are usable. Why a certain technique is chosen over a different one is often not thoroughly explained in papers. This subsection attempts to give an overview of which techniques are used in current state-of-the-art research. An analysis of the literature shows that the preprocessing of the data is used for among other things the handling of missing values, outliers, noise reduction, filtering, and transforming the data.

Missing values can be problematic for certain statistical and machine learning models. For this reason, they need to be treated or filled in properly. Several techniques are used in the literature. A first technique is removing the observations with missing data (see Maron et al., 2022; Miele et al., 2022; Cui et al., 2018; and Bangalore et al., 2017). This can be difficult when time series modeling is used. Furthermore, the question needs to be asked why the data are missing.

If they are not “missing completely at random” (MCAR), this can result in bias (Emmanuel et al., 2021). A different solution is single imputation. A first example of this is carry forward and/or backward. In this technique, the missing value is replaced by the last known value preceding the missing value (carry forward) or the first known value following on from the missing value (carry backward). This is used in Bermúdez et al. (2022), Campoverde et al. (2022), Chesterman et al. (2022, 2021), and Mazidi et al. (2017). An alternative is interpolation. This can be done in several ways, e.g., Hermite interpolation (see Bermúdez et al., 2022, and Campoverde et al., 2022) or linear interpolation (see Chesterman et al., 2022, 2021, and Miele et al., 2022). There are several elements that need to be taken into account when using these techniques. First of all, extra care needs to be taken when using them. Large gaps in time series are a problem, since the imputation can become meaningless. This can result in pollution of the relation between multiple signals. High dimensionality of the data can also be a problem (Emmanuel et al., 2021).

If the number of missing data is fairly limited (and there are no long stretches of missing data), an aggregate like the mean or median can be calculated and used as a proxy. The resulting time series has of course a lower resolution, but the missing values are gone. This method is used in Verma et al. (2022). The success of this methodology of course depends on how many data are missing and on how many data each aggregated value is based on. Another solution is what is called machine-learning-based imputation. This technique uses certain machine learning models to impute the missing values (Emmanuel et al., 2021). There are several ways to do

this. For example, clustering algorithms like  $k$ -nearest neighbors can be used to find similar complete observations. This is used in Black et al. (2022).

The above overview gives several techniques that have been used in research that focuses on anomaly detection in temperature signals from wind turbines. However, there exist several other techniques that can be used for imputing missing values, e.g., hot-deck imputation, regression imputation, expectation maximization, and multiple imputation (Emmanuel et al., 2021). To the best of the authors' knowledge, no (or a very limited number of) papers in this specific research domain have been written that use these techniques. The lack of research that uses, for example, multiple imputation (like multivariate imputation by chained equations or MICE; van Buren and Groothuis-Oudshoorn, 2011) can be considered a blind spot.

A second problem that might influence the NBM training is outliers. If these occur in sufficient quantity, they can impact the modeling severely. Oftentimes the decision is made in the literature to simply remove them. This of course implies that the outliers can be detected in the first place. This can be done for example by using the interquartile range (see Campoverde et al., 2022), a custom dynamic or user-defined threshold (see Chesterman et al., 2022, 2021, and Castellani et al., 2021), the  $5\sigma$  rule (see Miele et al., 2022) or clustering (see Cui et al., 2018, and Bangalore et al., 2017). Removing outliers needs to be done carefully to avoid abnormal values associated with the failure of interest also being removed. This requires a good understanding of the data. In some cases the outliers will be clearly visible. This can for example be the case when measurement errors result in values that are multiple times larger or smaller than what is physically possible. In these cases removing the outliers is straightforward. However, in other cases, the difference between outliers or anomalies caused by the failing component and outliers caused by another reason is much less clear. Removing these types of outliers should only be done after careful consideration.

Some papers also use noise reduction techniques. Noise in signals can make it more difficult for the NBM algorithm to model the relation between them. If it is possible to clean the signal, this should be considered, since it will improve the performance of the NBM model. This can be done for example by aggregating the data to a lower resolution (see Chesterman et al., 2022, 2021, and Turnbull et al., 2021) or cleaning or filtering the data using expert knowledge (see Peter et al., 2022; Takanashi et al., 2022; Verma et al., 2022; Turnbull et al., 2021; Udo and Yar, 2021; Beretta et al., 2020; and Kusiak and Li, 2011).

In some cases, it might be useful to transform the data. This can result in features with more favorable properties. For example, principal component analysis (PCA) transformation (see Campoverde et al., 2022, and Castellani et al., 2021) and zero-phase component transformation (see Renström et al., 2020) result in uncorrelated features which are

linear combinations of the original signals. This can be beneficial for the training of the NBM. Another transformation that might be done is rebalancing the dataset. This can be necessary when certain operational states of the turbine are underrepresented in the training data. Oversampling of the minority class or undersampling of the majority class is an option. A more sophisticated technique is the synthetic minority oversampling technique (SMOTE) used in for example Verma et al. (2022). Lastly, in some papers, new features are created by clustering the original signals of the SCADA data into several groups using clustering algorithms. In the next step, the new features are used as input to the NBM model (see Liu et al., 2020).

This overview shows that many different preprocessing techniques are available and have been tried. However, the technique choice is often not well motivated in the papers. Also, the impact of a certain technique on the results is in general not extensively discussed, even though it is known from statistical research that this can be significant.

## 2.2 The data and signals

SCADA data can come in different resolutions. The most available resolution is 10 min, since it reduces the number of data that need to be transmitted (Yang et al., 2014). This means that, for each 10 min window, the dataset contains the average signal value. In general, the SCADA data also contain information on the minimum, maximum, and standard deviation of the signal during the 10 min window. Less common resolutions are for example 1 min and 1 s. In the state-of-the-art literature, the following resolutions can be found:

- 10 min – Bermúdez et al. (2022), Black et al. (2022), Campoverde et al. (2022), Chesterman et al. (2022, 2021), Maron et al. (2022), Mazidi et al. (2017), Miele et al. (2022), Peter et al. (2022), Takanashi et al. (2022), Beretta et al. (2021, 2020), Castellani et al. (2021), Chen et al. (2021), Meyer (2021), Turnbull et al. (2021), Udo and Yar (2021), Liu et al. (2020), McKinnon et al. (2020), Renström et al. (2020), Zhao et al. (2018), Bangalore et al. (2017), Dienst and Beseler (2016), Bangalore and Tjernberg (2015, 2014), Schlechtingen and Santos (2014, 2012), Zaher et al. (2009), Garlick and Watson (2009);
- 5 min – Kusiak and Li (2011);
- 10 s – Kusiak and Verma (2012);
- 1 s – Sun et al. (2016), Li et al. (2014);
- 100 Hz – Verma et al. (2022), Kim et al. (2011).

The 100 Hz data used in for example Verma et al. (2022) come from the Controls Advanced Research Turbine (CART) of the National Renewable Energy Laboratory (NREL). The fact that it is a research turbine makes it possible to sample at much higher rates than what is normally



possible. Some papers combine the SCADA data with other information sources like event logs that contain wind turbine alarms (see Miele et al., 2022; Beretta et al., 2020; Renström et al., 2020; and Kusiak and Li, 2011) or vibration data (see Turnbull et al., 2021).

The SCADA data contain information on many different parts of the turbines. This implies that the datasets consist in general of dozens or even hundreds of signals (depending on the turbine type). However, not all of them are relevant to the case that is being solved. Some papers focus on a small subset of expert-selected signals (see for example Peter et al., 2022; Bermúdez et al., 2022; and Chesterman et al., 2022). Other papers use a large subset of signals and reduce the dimensionality of the problem during the preprocessing step or during a model-based automatic feature selection step that extracts the relevant information (see for example Lima et al., 2020; Renström et al., 2020; and Dienst and Beseler, 2016). Some papers select signals based on the internal structure of the wind turbine (for example on the subsystem level; Marti-Puig et al., 2021). The advantage of the first method is that the number of signals used for training is limited, which reduces the computational burden of the training process. The disadvantage is however that for cases in which the expert knowledge is not complete, important signals might be missed. This is less likely when the second method is used. The disadvantage of this method is however that the computational cost is significantly higher and that the selected subset can change over different runs. The third method uses the wind turbine ontology or taxonomy as a guideline. Its performance depends of course on the quality of the ontology or taxonomy.

The signals that are often used for condition monitoring of wind turbines can be more or less divided into three groups: (1) environmental data like wind speed or outside temperature (used in for example Bermúdez et al., 2022; Black et al., 2022; Campoverde et al., 2022; Mazidi et al., 2017; and Miele et al., 2022); (2) operational data from the wind turbine like active power or rotor speed (used in for example Bermúdez et al., 2022; Black et al., 2022; Chesterman et al., 2022; Miele et al., 2022; and Peter et al., 2022); and (3) wind turbine temperature signals like the temperatures of the generator bearings, the temperature of the main shaft bearing, or the temperature of the generator stator (used for example in Bermúdez et al., 2022; Black et al., 2022; Campoverde et al., 2022; Chesterman et al., 2022; and Mazidi et al., 2017). The first two groups are often used by default, independent of the target signal that needs to be modeled or the failure that needs to be detected. This is because they contain information on the wind turbine context (e.g., it is a stormy day, a very hot day, the turbine is derated). The third group of signals is much more tied to the case at hand. For example, generator temperature signals are used if the focus lies on generator failures and gearbox temperature signals are used if gearbox failures need to be detected.

Overall it can be stated that most state-of-the-art research is focused on 10 min SCADA data. This means that there are research opportunities on data with a higher resolution like 1 min or 1 s. Furthermore, several signal or feature selection techniques are used in the literature. However, a thorough examination of their performance (expert-knowledge-based vs. automatic or model-driven vs. ontology- or taxonomy-guided feature selection), advantages, and disadvantages has, according to the best of our knowledge, not been done yet.

### 2.3 Normal behavior modeling algorithms

The next part of the NBM framework is the algorithm that is used for modeling the normal behavior. In general, a considerable amount of attention is paid to this in the literature. This part models the normal (or healthy) behavior of the signal of interest. For this, the current state-of-the-art literature uses techniques from the statistics and machine learning domains. These domains contain a large variety of algorithms that are suitable for the task. More or less three categories can be distinguished: (1) statistical models, (2) shallow (or traditional) machine learning models, and (3) deep learning models. Although this classification gives the impression that the papers can be assigned to a single category, it is quite often the case that research uses or combines models from multiple categories.

Even though there are examples of recent papers in which statistical techniques are used for the modeling of the normal behavior, they are a minority. Models that are used are for example OLS and autoregressive integrated moving average (ARIMA) (see Chesterman et al., 2022, 2021), where they are used to remove autocorrelation and the correlation with other signals from the target signal. A different kind of statistical algorithm that is occasionally used is the PCA (see Campoverde et al., 2022) and its non-linear variant (see Kim et al., 2011). These are, contrary to the previous algorithms, unsupervised, but they can be used to learn the normal or healthy relations between the signals. In Garlick and Watson (2009) an OLS and an autoregressive exogenous (ARX) model are combined to model the normal behavior, which makes it possible to take time dependencies into account. The advantage of statistical models is that they are relatively simple, computationally lightweight, and data-efficient. They are well studied, and their behavior is well understood. This makes them often suitable as a first-analysis tool or in domains where there are constraints on the computational burden or the number of data that are available. The downside is however that they are relatively simple, which makes them in general unsuitable to model highly complex non-linear dynamics. Whether this is a problem depends of course on the case that needs to be solved.

Techniques from the traditional (shallow) machine learning domain are used more often. By traditional machine learning is meant models like decision trees, random forests, gradient boosting, and support vector machines. These mod-

els are more complex than traditional statistical models and are better able to model non-linear dynamics. However, they require in general more training data and time. Examples of algorithms that are used in the current state of the art are tree-based models like random forest (see Chesterman et al., 2022; Turnbull et al., 2021; and Kusiak and Li, 2011) and gradient boosting (see Chesterman et al., 2022; Maron et al., 2022; Beretta et al., 2021, 2020; Udo and Yar, 2021; and Kusiak and Li, 2011) and models like support vector machine and regression (see Chesterman et al., 2022; Castellani et al., 2021; McKinnon et al., 2020; and Kusiak and Li, 2011). Another type of model that is occasionally used is derived from the linear model (OLS) but includes some form of regularization to be better able to cope with high dimensional data and highly correlated features, i.e., least absolute shrinkage and selection operator (LASSO) (see Dienst and Beseler, 2016). The latter model can be situated somewhere between the statistical and traditional machine learning categories.

In recent years deep learning models, e.g., neural networks, have become popular. Deep learning models are more complex than traditional machine learning models. Their advantage is that they are even better at modeling non-linear dynamics. Their disadvantages are that they require even more data, they are computer-intensive to train, and the results are more difficult to interpret. This however has not diminished their popularity, and at the moment they are the most popular model category in the state-of-the-art research. Just like the traditional machine learning domain, the deep learning domain is very diverse. Over the years many different types of models have been developed. These are either completely new models or combinations of already existing deep learning models. Both can be found in the state-of-the-art literature. Deep neural networks are used in Black et al. (2022), Jamil et al. (2022) (transfer learning), Mazidi et al. (2017), Verma et al. (2022), Meyer (2021) (multi-target neural network), Turnbull et al. (2021), Cui et al. (2018), Sun et al. (2016), Bangalore and Tjernberg (2015) (nonlinear autoregressive exogenous model, NARX), Bangalore and Tjernberg (2014), Li et al. (2014), Kusiak and Verma (2012), Kusiak and Li (2011), Zaher et al. (2009). Another popular type of model is the autoencoder (AE). Just like a PCA, this model learns normal or healthy behavior through dimension reduction which makes it ignore noise and anomalies. However, compared to the PCA it is better at learning nonlinearities. This model type is used for example in Miele et al. (2022), Chen et al. (2021), Beretta et al. (2020), Renström et al. (2020), and Zhao et al. (2018). Convolutional neural networks (CNNs), originally designed and used for the analysis of images, can also be used for the detection of anomalies and failures. Examples can be found in Bermúdez et al. (2022) (combination of CNN and LSTM), Xiang et al. (2022), Zgraggen et al. (2021), and Liu et al. (2020). Another model that is used is the LSTM. This model is particularly suitable for time series, since it is able to model the time dependencies. This model is used in Bermúdez et al.

(2022) and Udo and Yar (2021). Two other models that have also been used but to a much lesser extent are the extreme learning machines (ELMs) (see Marti-Puig et al., 2021) and the generative adversarial network (GAN) (see Peng et al., 2021).

Another type of algorithm that is occasionally used in the literature is based on fuzzy logic. An example of such a model is the adaptive neuro-fuzzy inference system (ANFIS). Papers that use this model are Schlechtingen and Santos (2014, 2012) and Schlechtingen et al. (2013). In Tautz-Weinert and Watson (2016), experiments are performed with, among others, ANFIS and Gaussian process regression. And finally, there is also research that uses copula-based modeling. An example is the research presented in Zhongshan et al. (2018).

Overall it can be stated that the NBM ecosystem is diverse. In recent years, deep learning has become the most popular methodology. The merits of these models are clear from the results. Often they outperform the statistical and traditional machine learning models. However, the question is whether they are always the most suitable methodology for implementation in the field. The data requirements mean that deploying the system quickly on a new wind farm is not possible (transfer learning alleviates this issue to a certain extent). Also, the high computational requirements result in more costly retraining and higher maintenance costs. The question is whether these disadvantages are outweighed by the improved performance once deployed in the field. Not much attention is paid in the literature to this question.

#### 2.4 Algorithms for the analysis of the NBM prediction error

The last step of the NBM methodology is the analysis of the prediction error made by the NBM model. This model predicts the expected normal behavior of a signal. If the true or observed signal deviates abnormally much from this prediction or the deviation shows certain trends, then this might indicate that something is going wrong with the related component and that a failure is imminent. The main goal of the last step is to search for these patterns in the prediction error. There are many different techniques (and combinations of techniques) that can be used for this. There are different ways to classify them. They can be divided by domain. Firstly there are statistics-based methods that use the distribution of the prediction error under healthy conditions to determine a threshold that can be used to classify the prediction errors as normal or anomalous. Secondly, there are methods that are based on models from the statistical process control (SPC) domain. Thirdly there are methods that are based on models from the machine learning domain. A different classification focuses on the number of signals they analyze in a single pass. There are univariate methods, which only take a single signal at a time into consideration. There are also multivariate methods, which look at multiple signals. In gen-

eral, machine-learning-based methods are multivariate. The SPC-based method can be both, since the univariate control chart algorithms like Shewhart, cumulative sum (CUSUM), and exponential weighted moving average (EWMA) have their multivariate counterparts. However, in the state-of-the-art literature often only the univariate versions are used. The statistics-based methods that use the distribution of the prediction error are in general univariate. An exception to this is the Mahalanobis distance, which is mostly used in a multivariate setting.

The overview of the state of the art given in this paper will use the first classification as a guideline. The technique that uses the distribution of the prediction error to find a suitable threshold to identify anomalies is for example used in Meyer (2021), Zhao et al. (2018), and Kusiak and Verma (2012). The Mahalanobis distance combined with an anomaly threshold is used in Miele et al. (2022), Renström et al. (2020), Cui et al. (2018), and Bangalore et al. (2017). SPC techniques are used in Udo and Yar (2021) (Shewhart control chart), Chesterman et al. (2022) (CUSUM), Chesterman et al. (2021) (CUSUM), Bermúdez et al. (2022) (EWMA), Campoverde et al. (2022) (EWMA), Xiang et al. (2022) (EWMA), and Renström et al. (2020) (EWMA). The machine-learning-based methods for the analysis of the prediction error are in general modifications of traditional machine learning algorithms. For example, the isolation forest, which is used in Beretta et al. (2021, 2020) and McKinnon et al. (2020), is similar to the random forest algorithm, while the one-class SVM, used in Turnbull et al. (2021), Beretta et al. (2020), and McKinnon et al. (2020), is similar to the SVM.

Overall it can be stated that in the current state of the art, multiple techniques are used for the analysis of the prediction error, without a single category clearly having the upper hand. Furthermore, both univariate and multivariate techniques are still used. The multivariate techniques can analyze the prediction errors in multiple signals, which gives them an advantage compared to the univariate techniques. However, their disadvantage is that when analyzing several signals at the same time, a deviation in a single signal might be masked. This is shown in Renström et al. (2020), where the authors observe that when the Mahalanobis distance is calculated on several prediction errors at the same time (multivariate setting), it does not always clearly increase when only a single prediction error deviates. For this reason, they point out that it would be interesting to combine multivariate and univariate techniques.

### 3 Design of demonstration experiments for evaluating the performance of techniques used in the state-of-the-art literature

In this section, the methodology of the experiments is explained, which will be used to demonstrate certain techniques

found in the state-of-the-art literature. For this demonstration, an NBM pipeline is designed, which consists of the following steps: data preprocessing, NBM, anomaly detection, and health score calculation. The pipeline is validated on data from five operational wind farms. The data contain information on three types of failures, i.e., generator bearing replacements, generator fan replacements, and rotor brush high-temperature failures. In each step of the pipeline, multiple techniques and configurations are tested and compared. To this end, six experiments are designed. Care is taken to create a lab environment as much as possible. This means that the parts of the pipelines that are not relevant to the experiment are kept constant.

#### 3.1 The input data

The experiments are based on two data sources (for confidentiality reasons not all the details of the input data can be shared). The first one is 10 min SCADA data originating from five different onshore wind farms (wind farms 1–5). The geographic location of each wind farm is different. The wind turbines in these wind farms are all of the same type with a rated power of 2 MW. The wind farms are relatively small, containing only four to six wind turbines. Some datasets contain a substantial number of missing values. The number of (obvious) measurement errors is low. The SCADA data contain over 100 signals. Only the 10 min averages of the signals related to the drive train or the operational condition of the wind turbine are used in this research. The signals that are selected from the SCADA data are based on the state-of-the-art literature. It is a relatively large subset, larger than what would be used if the selection were based on only expert knowledge. Table 1 gives an overview.

The second source of information comprises the maintenance logs, which contain the replacements and failures. Information is available on three types of events: generator bearing replacements, generator fan replacements, and rotor brush high-temperature failures. For each replacement or failure event, information is available on the turbine ID, the date, and the event type. This is valuable information for validating the methodologies. However using these logs also involves a couple of challenges, e.g., imprecise event dates or missing events. The logs give only an approximate indication of when something went wrong. Furthermore, it needs to be pointed out that a replacement of a component does not necessarily mean that the component failed. Some components will have failed, whereas others will have been replaced as a preventive measure.

#### 3.2 Preprocessing

The first step of the NBM framework or pipeline is the preprocessing of the data (see Fig. 1). Because the main focus of this paper lies on the NBM and the anomaly detection techniques, no extensive comparative analysis of different pre-

**Table 1.** Overview of the SCADA signals used as input for the NBM model.

Signal name	Symbol	Unit	Information
TempGenBearing_1 (avg)	$T_{\text{gen\_bear\_1}}$	$^{\circ}\text{C}$	Temperature of the first generator bearing
TempGenBearing_2 (avg)	$T_{\text{gen\_bear\_2}}$	$^{\circ}\text{C}$	Temperature of the second generator bearing
TempStatorWind (avg)	$T_{\text{stator}}$	$^{\circ}\text{C}$	Temperature of the generator stator
GeneratorSpeed (avg)	$V_{\text{gen}}$	rpm	Rotational speed of the generator
Generator torque (avg)	$\tau_{\text{generator}}$	N m	Torque at the generator
TempConvInlet (avg)	$T_{\text{conv\_inlet}}$	$^{\circ}\text{C}$	Temperature of the converter inlet
TempGearbBear_1 (avg)	$T_{\text{gear\_bear\_1}}$	$^{\circ}\text{C}$	Temperature of the gearbox bearing 1
TempGearbBear_2 (avg)	$T_{\text{gear\_bear\_2}}$	$^{\circ}\text{C}$	Temperature of the gearbox bearing 2
TempGearbInlet (avg)	$T_{\text{gear\_inlet}}$	$^{\circ}\text{C}$	Temperature of the gearbox inlet
GearboxSpeed (avg)	$V_{\text{gearbox}}$	rpm	Rotational speed of the gearbox
TempRotorBearing (avg)	$T_{\text{rotor\_bear}}$	$^{\circ}\text{C}$	Temperature of the rotor bearing
Rotor speed (avg)	$V_{\text{rotor}}$	rpm	Rotational speed of the rotor of the wind turbine
Active power (avg)	$P_{\text{active}}$	kW	Amount of power that is being produced by the wind turbine
Nacelle temperature (avg)	$T_{\text{nacelle}}$	$^{\circ}\text{C}$	Temperature measured inside the nacelle of the wind turbine
Outside temperature (avg)	$T_{\text{ambient}}$	$^{\circ}\text{C}$	Ambient temperature
Wind speed	$V_{\text{wind}}$	$\text{m s}^{-1}$	Wind speed measured at each turbine

processing techniques is done. However, the NBM pipeline makes use of several preprocessing techniques, which makes it necessary to at least discuss or mention them. Some techniques are trivial but necessary. They will just be mentioned without going into more detail. The more interesting ones, like for example the healthy data selection or the fleet median normalization, will be discussed more thoroughly.

The NBM framework makes use of the following preprocessing steps:

- *Data cleaning.* This involves selection of relevant variables and turbines, renaming of the variables, matching of the SCADA data with the replacement information, and linear interpolation or carry forward/backward of the missing values (similar to what was done in Bermúdez et al., 2022; Campoverde et al., 2022, Mazidi et al., 2017; and Renström et al., 2020, but with linear interpolation). This step will not be discussed in more detail, since it is a trivial transformation.
- *Selecting healthy training data.* This rule-based method will be discussed in depth.
- *Determining the operating condition of the turbines.* This is done using the IEC 61400-1-12 standard (Commission, 2022) as a guideline.
- *Signal filtering using the wind farm median.* This is an important transformation with a significant impact on the results. For this reason, this step is discussed in depth.
- *Removal of sensor measurement errors.* The removal of sensor measurement errors is done in a fully automated way. A short discussion of this step is given.

- *Aggregating to the hour level.* The purpose of this step is to reduce the amount of noise and the number of missing values in the data. Similar actions were taken for example in Turnbull et al. (2021) and Verma et al. (2022). This step is not always appropriate. This is for example the case for failures that form very fast over time or for signals that exhibit damage patterns that are very short-lived like in vibration analysis. It is up to the data analyst to determine whether the advantages outweigh the disadvantages. This preprocessing step will not be discussed in more detail, since it is a trivial technique.

### 3.2.1 Signal filtering using the fleet (wind farm) median

The wind turbine signals in the SCADA data are quite complex, meaning there are a lot of factors that influence them. This complexity makes it more difficult to model the normal behavior. This means that if a part of this complexity could be removed, it would simplify the problem, which normally should result in an improved modeling performance and an overall more data-efficient model. This can be accomplished by calculating the fleet median of a signal (e.g., temperature of generator bearing 1 at time  $t$  for turbines 1, 2, 3, 4, 5) and subtracting it from the wind turbine signals (e.g., temperature of generator bearing 1 at time  $t$  of turbine 1). This technique is also used in Chesterman et al. (2022, 2021). The fleet median can be seen as an implicit normal behavior model. It is implicit because it does not require the selection of predictors or the training of a model. It models the normal behavior as long as 50% + 1 turbines are acting normally at any given time. By taking the median over a whole farm, it captures farm-wide effects, which are common to all the turbines, which is why it will be called the “common component”. Elements modeled by this component are for example



the wind speed, the wind direction, and the outside temperature. By subtracting the median from the wind turbine signal, these farm-wide effects are removed. What is left are turbine-specific effects, which will be called the “idiosyncratic component”. Turbine-specific anomalies should only be visible in there.

In practice, this preprocessing step means that from each signal in Table 1 the fleet median is subtracted (e.g.,  $T_{\text{rotor}} - \text{fleet median } T_{\text{rotor}}$ ,  $T_{\text{rear}} - \text{fleet median } T_{\text{rear}}$ ). Figure 2 shows the results of the decomposition for the five turbines of wind farm 2. The top subplot shows the original generator bearing 1 temperature signal. The middle subplot shows the common components. From the plot, it is obvious that the common component captures the general (fleet or farm-wide) trend, while turbine-specific evolutions are ignored. The bottom subplot shows the idiosyncratic components. The power down of turbine 4 is (more) clearly visible in the idiosyncratic component than in the original signal. This indicates that the decomposition is successful. Figure 3 shows that the fleet median is useful for filtering out macro-level or fleet-wide effects like seasonal fluctuations from the data. The common component captures clearly the seasonal fluctuations in the nacelle temperature, which results in an idiosyncratic component that is free of them. This is beneficial because it means that seasonal fluctuations will not influence the false positive rate when less than 1 year of training data is used. Furthermore, the common component also captures the transient behavior like cool-downs that are common to all turbines in the fleet. This means that the idiosyncratic component is free from most transient behavior. What remains of transient behavior are cool-downs that are unique to the turbine. This means that they are caused for example by a turbine that is turned off for maintenance. These events are relatively rare, which means that subtracting the fleet median from the signals has reduced the modeling complexity substantially.

How well the fleet median removes the macro-level effects depends of course on the quality of the fleet median. An issue that may arise, especially in small wind farms, is that a substantial number of turbines are offline for maintenance. For example, it can be that two turbines are offline for maintenance in a wind farm with four turbines. This will of course have an impact on how representative the median is of the normal behavior. For this reason, a rule-based safeguard is added that under certain conditions will convert the fleet median to NaN. The rules are the following:

- If fleet size  $< 5$ , no missing values at time  $t$  are allowed.
- If  $5 \leq$  fleet size  $< 10$ , at most 20 % missing values at time  $t$  are allowed.
- If fleet size  $\geq 10$ , at most 40 % missing values at time  $t$  are allowed.

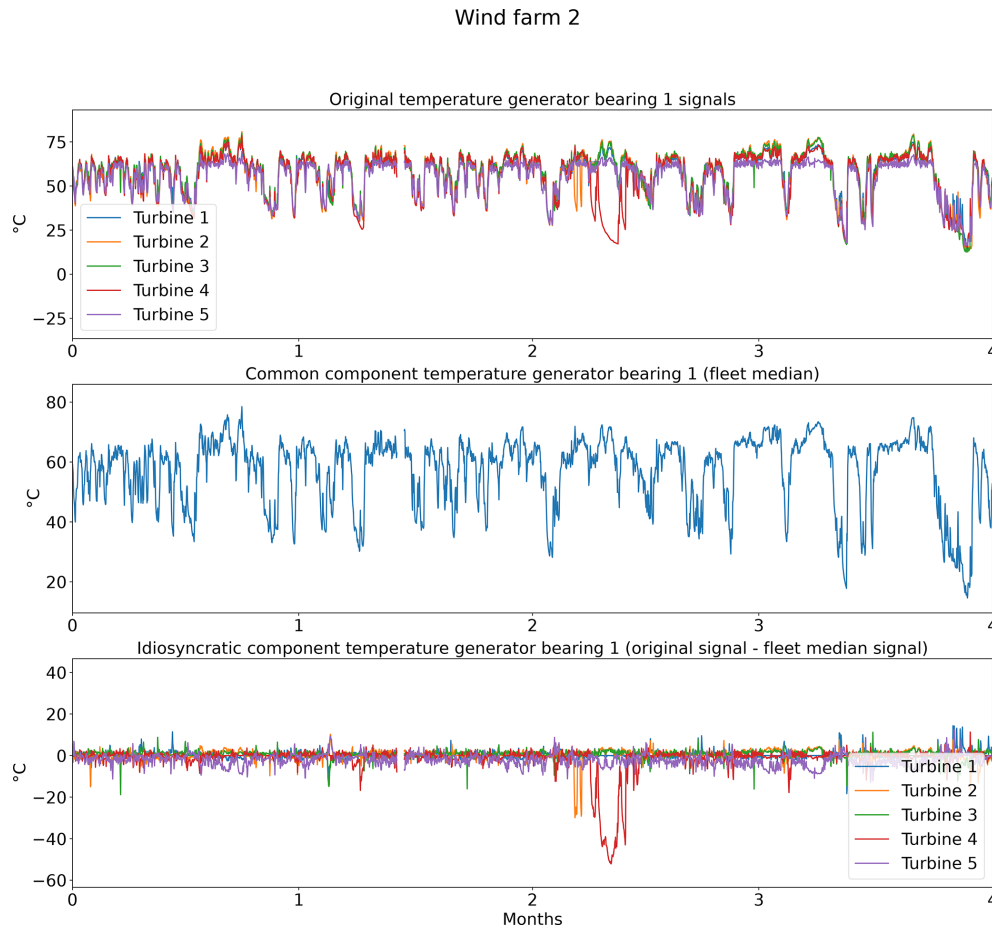
### 3.2.2 Selecting healthy training data

The selection of healthy training data is an important step in the NBM framework. If it is not done properly, it can result in the contamination of the training data with anomalous or “unhealthy” observations, which can disturb the training of the normal behavior. Data are considered healthy if they are not polluted by abnormal behavior caused by a damaged component. Unfortunately, data from real machines do not contain a label that indicates whether they are healthy or not. This means that certain assumptions need to be made about the data, i.e., a “healthy data” rule. The rule used in this paper considers data that precede a failure by less than 4 months (which is the same as what is used in Verma et al., 2022) or follow a failure by less than a month (to avoid test and upstart behavior) unhealthy.

Once the unhealthy data have been identified, healthy data can be selected. The methodology presented here selects the healthy data in a fully automated fashion. The user can determine how many training data (number of observations) per turbine are required for modeling the normal behavior. The healthy data are selected in chronological order from the time series. For most experiments, the first 4380 (which equals roughly 6 months of data) healthy samples of each turbine are selected for training. The selected data are combined into a single training dataset. This implies that only a single model per signal per farm is trained, which results in a large reduction in the training time and more efficient usage of the training data. Figures 4 and 5 give a schematic representation of how the healthy training data are selected.

### 3.2.3 Handling of measurement or sensor errors

Removing outliers and/or measurement errors is something that is done in most research. The SCADA data used for the demonstration experiments contain a small number of unrealistic values. The technique used here uses thresholds to determine which observations are measurement errors and which are not. The overview of the state of the art shows that this is a technique that is often used. The thresholds used in the demonstration experiments are determined in a fully automated fashion so that they do not need to be manually determined for each signal of each turbine type. This manipulation of the data is not without risk. Only measurement errors should be removed and not the deviations in the signals caused by the failure of the component. The threshold depends on the median value of the original signal (meaning that the fleet median has not been subtracted from the wind turbine signal). This value is multiplied with a scaling factor, which in this paper is set to 1. This threshold is used on the signal from which the fleet median has been subtracted. All absolute values in this signal that are larger than the threshold are considered measurement errors. Concretely this means that values from the wind turbine signal that deviate from the fleet median (on the positive side) by more than its me-



**Figure 2.** Example of the impact of decomposing the generator bearing 1 temperature of the turbines in wind farm 2 in a common and idiosyncratic component.

dian value are considered measurement errors. Equation (1) shows the procedure using mathematical notation. The measurement errors are replaced with NaN values. In a later step, they are replaced using linear interpolation or if necessary carry forward/backward.

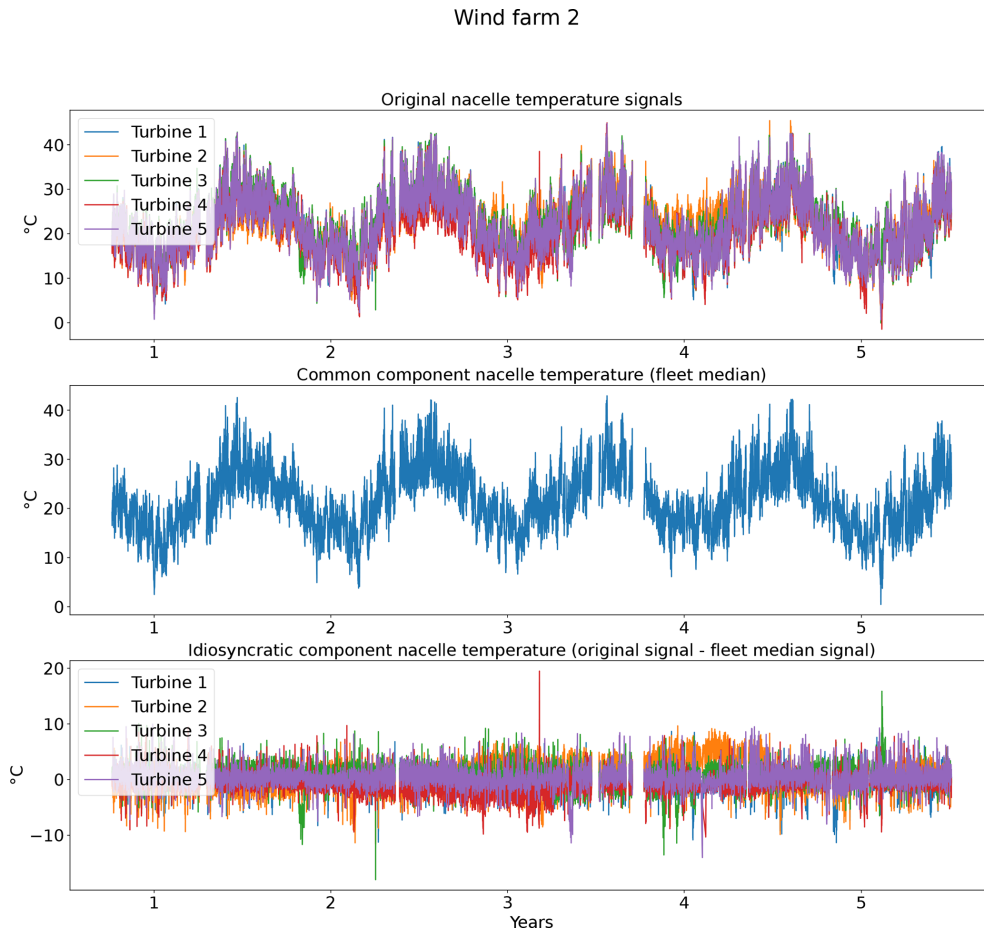
$$\text{Error}_{\text{measurement}} = \begin{cases} 1 & \text{if } |\text{Value}_{\text{fleet\_corrected}}| > \\ & \text{Value}_{\text{original}} \text{ outlier\_factor}, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where  $\text{Error}_{\text{measurement}}$  is an indicator whether the observation is a measurement error or not,  $\text{Value}_{\text{fleet\_corrected}}$  is the signal value from which the fleet median value has been subtracted,  $\text{Value}_{\text{original}}$  is the signal value, and  $\text{outlier\_factor}$  is a constant multiplier.

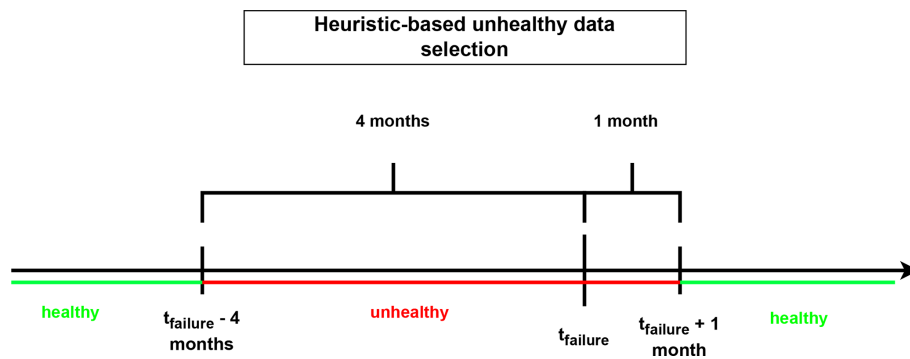
### 3.3 Normal behavior modeling

The normal behavior model is the core of the NBM framework. It models the normal or healthy relation between one

or more predictors and a target signal. Once trained it can be used to predict the expected normal behavior. The differences between the observed and predicted values are then analyzed by anomaly detection algorithms in the final step. In this section, the methodological setup of the experiments will be discussed. To assess the performance of the NBM configurations, two metrics are used. The first metric is the healthy test data RMSE. A proper NBM model should be able to model the healthy data well. However, this metric does not tell us much about how well it can distinguish unhealthy from healthy data. For this reason, a second metric is introduced, which is the difference between the median prediction error in the healthy and unhealthy data ( $\Delta\text{PE}_{\text{unh-h}}$ ) (see Eq. 2). The idea behind this metric is that good NBM models should have a small prediction error in the healthy data ( $\text{PE}_{\text{h}}$ ) (because they are trained on the healthy data) and a large prediction error in the unhealthy data ( $\text{PE}_{\text{unh}}$ ) (because something has changed compared to the healthy situation). For the type of failures studied in this paper,  $\Delta\text{PE}_{\text{unh-h}}$  should be positive because the damage to the related components should result in higher-than-normal temperatures. The



**Figure 3.** Example of the impact of decomposing the nacelle temperature of the turbines in wind farm 2 in a common and idiosyncratic component.



**Figure 4.** Unhealthy data identification.

more positive  $\Delta PE_{\text{unh-h}}$  is, the better the NBM because it indicates that the model can better distinguish healthy from unhealthy data.

$$\Delta PE_{\text{unh-h}} = \text{median}(PE_{\text{unh}}) - \text{median}(PE_{\text{h}}), \quad (2)$$

where  $\Delta PE_{\text{unh-h}}$  is the difference between the median unhealthy and healthy data prediction error,  $PE_{\text{unh}}$  is the un-

healthy data prediction error, and  $PE_{\text{h}}$  is the healthy data prediction error.

As a baseline the fleet median signal for the healthy training data is used. Since the fleet median also models the normal behavior, although without the requirement of specifying the predictors, it will be called an “implicit NBM” (vs. the “explicit NBM” models that do require a specification of

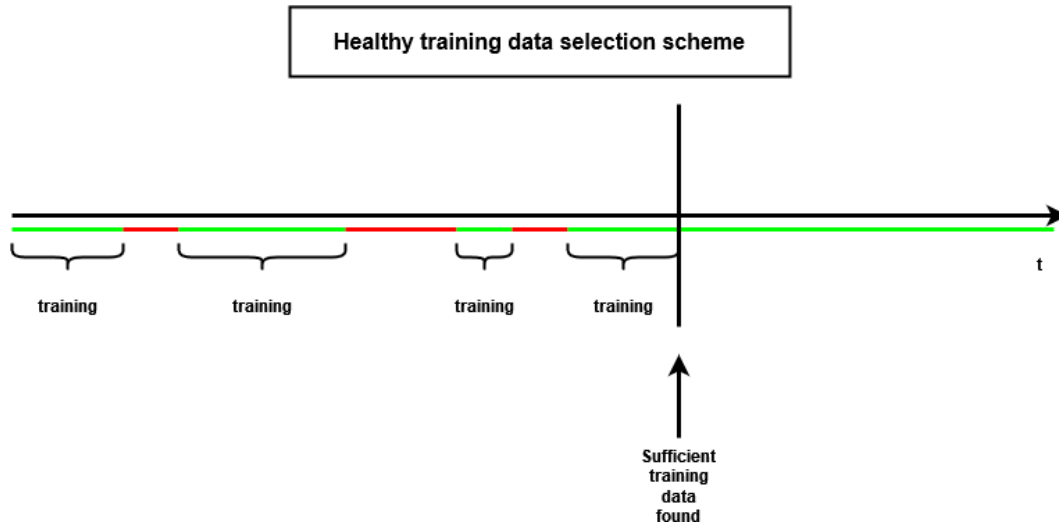


Figure 5. Heuristic-based healthy data identification.

predictors, model, parameters, etc.). The explicit NBM that will mainly be used is the elastic net. It is a simple, transparent, and robust model that can handle large numbers of (correlated) predictors. At the same time, it can work with a limited number of training data. This corresponds to requirements set by industrial partners, e.g., at most only a couple of months of 10 min training data per turbine, low maintenance cost, low training cost, and high transparency. Nevertheless, in experiments 2 and 6 the performance of the elastic net will be compared to that of more complex models from the shallow machine learning domain (i.e., light gradient-boosting machine (light GBM), support vector regression (SVR) in experiment 6) and the deep learning domain (i.e., multi-layer perceptron (MLP) in experiments 2 and 6). This should give an idea of the limits and usability of the elastic net model and whether the trade-off between computational cost and complexity on the one hand and the performance on the other hand is acceptable or not.

### 3.3.1 Elastic net regression for modeling the normal behavior

It has been shown in the literature that linear models can be good modelers of the normal behavior of wind turbines, and they are also time-efficient (see Dienst and Beseler, 2016). However, by using elastic net (which was developed in Zou and Hastie, 2005), which is basically a linear regressor with  $L_1$  and  $L_2$  regularizers added to it (see Eq. 3), there are some extra advantages. Firstly, the model is more robust when many (correlated) features are used. Secondly, it also performs an automatic feature selection. This implies that it is possible to model for example the generator bearing 1 temperature by giving the model all the signals that are connected to the whole drive train of the turbine. This reduces the configuration burden for the user. Furthermore, the algo-

rithm works in a transparent way, which avoids the “black box” problem. The number of training data it requires is also favorable compared to more complex machine learning algorithms. A disadvantage of the model is that it is relatively simple, which means that it is not good at modeling highly non-linear dynamics. Whether this is a problem will be tested in experiment 5, where the performance of the elastic net is compared with that of more complex shallow machine learning and deep learning models.

$$\hat{\beta}^{\text{elastic-net}} = \operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2 \right\}, \quad (3)$$

where  $\hat{\beta}^{\text{elastic-net}}$  denotes estimates of the coefficients or weights by the elastic net model,  $\beta_0$  and  $\beta_j$  denote coefficients or weights of the model;  $\sum_{j=1}^p |\beta_j|$  is the  $L_1$  penalty term;  $\sum_{j=1}^p \beta_j^2$  is the  $L_2$  penalty term, also called Tikhonov regularization;  $\lambda_1 \geq 0$  is the weight of  $L_1$  penalty term; and  $\lambda_2 \geq 0$  is the weight of  $L_2$  penalty term.

### 3.3.2 Training of the NBM model

The NBM models are trained on healthy data that are extracted from the SCADA data. Failing to train on more or less healthy data can result in severe degradation of the modeling performance of the NBM (this depends on the relative quantity of anomalies). To reduce the computational and maintenance burden of the pipeline, a single NBM model per signal per wind farm is trained. This means that healthy training data from several wind turbines are combined in a single



training dataset. This decision was taken in response to concerns raised by wind turbine operators that if separate models were trained per turbine, this would result in an unacceptable maintenance burden. Combining training data from multiple wind turbines is however not without risks. Structural signal differences (e.g., a turbine with a generator bearing that is always 1 or 2 °C warmer than the one of a different turbine under the same conditions) between the different wind turbines are not modeled (unless wind turbine dummies are added to the predictor list). This can result in structural deviations in the prediction errors, e.g., a prediction error that is structurally positive or negative. However, data analysis showed that the temperature or behavior differences between the different bearings are small. There are no indications in the results (see below) that the differences between the different bearings seriously hamper the analysis. Furthermore, experiments in which the model was retrained after each bearing replacement did not show any clear performance improvement.

In general, the training of the NBM models will be done by using the first 4380 healthy observations (or 6 months of data) of each turbine. This number is limited on purpose so that it answers the requirements of the industry. Fewer training data mean that new wind turbines can more easily be added to the anomaly detection system (less startup time). The subtraction of the fleet median from the wind turbine signals neutralizes seasonal fluctuations. The NBM models are trained on the training data using a full grid search or a random grid search when the number of hyperparameter combinations is large, over sensible ranges for the hyperparameters. To avoid overfitting 5-fold cross-validation (CV) is used. The trained model is tested on the test dataset to assess the performance. For each model, the healthy test data RMSE is calculated. This is used to compare the models from the different experiments. The third experiment examines the impact of further reducing the quantity of training data to 2 months per turbine.

### 3.4 The anomaly detection procedure

The trained NBM model is used to predict the expected normal behavior. The prediction error in the model indicates how anomalous the observed behavior is. As shown in the state-of-the-art overview, there are many different anomaly detection techniques that can be used to analyze it. The techniques used in the sixth experiment are based on univariate statistical techniques that are transparent, robust, and computationally light. More specifically, two different techniques are tested. The first technique is based on the prediction error distribution. The second technique is based on a technique from the SPC domain.

The first technique is most suitable for identifying point anomalies. It is based on the principle of iterative outlier detection (IOD) (also called iterative outlier removal). This means that outliers are removed over several iterations un-

til the outlier thresholds (these are the thresholds that determine which observations are outliers and which are not) have stabilized. To make these thresholds more robust against outliers, the standard deviation is approximated by the median absolute deviation (MAD) (see Eqs. 4 and 5, with  $k = 1.4826$ ). The anomaly scores are calculated using Eq. (6).

$$MAD = \text{median}(|X_i - \tilde{X}|), \tag{4}$$

where  $X_i$  is the signal observation at time  $t = i$ ,  $\tilde{X}$  is the signal median, and MAD is the median absolute deviation.

$$\hat{\sigma}_{\text{robust}} = k \text{MAD}, \tag{5}$$

where  $\hat{\sigma}_{\text{robust}}$  is a robust estimate of the standard deviation signal,  $k$  is a constant multiplier or scaler, and MAD is the median absolute deviation.

Anomaly score =

$$\begin{cases} -3 & \text{if } \text{idio\_comp} > \text{median}_{\text{idio\_comp}} - 5\hat{\sigma}_{\text{robust}}, \\ -2 & \text{if } \text{idio\_comp} > \text{median}_{\text{idio\_comp}} - 4\hat{\sigma}_{\text{robust}}, \\ -1 & \text{if } \text{idio\_comp} > \text{median}_{\text{idio\_comp}} - 3\hat{\sigma}_{\text{robust}}, \\ 1 & \text{if } \text{idio\_comp} > \text{median}_{\text{idio\_comp}} + 3\hat{\sigma}_{\text{robust}}, \\ 2 & \text{if } \text{idio\_comp} > \text{median}_{\text{idio\_comp}} + 4\hat{\sigma}_{\text{robust}}, \\ 3 & \text{if } \text{idio\_comp} > \text{median}_{\text{idio\_comp}} + 5\hat{\sigma}_{\text{robust}}, \end{cases} \tag{6}$$

where  $\text{idio\_comp}$  is the idiosyncratic component,  $\text{median}_{\text{idio\_comp}}$  is the median idiosyncratic component, and  $\hat{\sigma}_{\text{robust}}$  is a robust estimation of the standard deviation.

In the next step, the anomaly scores are transformed into health scores. This is done by calculating the moving average of the anomaly scores for different windows (1, 10, 30, 90, and 180 d). For these moving averages, upper and lower bounds are calculated. This is done by combining the moving averages with the same window length of the same signals from the different wind turbines and calculating Tukey’s fences. Three positive thresholds are used to determine the moving-average anomaly score. Next, the sum of the moving-average anomaly scores is taken over the different windows for each time step  $t$ . This sum is the health score and determines the health category (Eq. 8).

$MA_{\text{anomaly score win } x} =$

$$\begin{cases} 1 & \text{if } MA_{\text{win } x} > q(MA_{\text{win } x}, 0.75) \\ & + 1.5(q(MA_{\text{win } x}, 0.75) - q(MA_{\text{win } x}, 0.25)), \\ 2 & \text{if } MA_{\text{win } x} > q(MA_{\text{win } x}, 0.75) \\ & + 2.5(q(MA_{\text{win } x}, 0.75) - q(MA_{\text{win } x}, 0.25)), \\ 3 & \text{if } MA_{\text{win } x} > q(MA_{\text{win } x}, 0.75) \\ & + 3.5(q(MA_{\text{win } x}, 0.75) - q(MA_{\text{win } x}, 0.25)), \end{cases} \tag{7}$$

where  $MA_{\text{win } x}$  is the moving average with window length  $x$ ,  $q(MA, \cdot)$  is the quantile of the moving-average distribution, and  $MA_{\text{anomaly score win } x}$  is the moving average of the

anomaly score for window length  $x$ .

$$\begin{aligned} \text{Health score} = & \text{MA}_{\text{anomaly score win 1d}}, \\ & + \text{MA}_{\text{anomaly score win 10d}}, \\ & + \text{MA}_{\text{anomaly score win 30d}}, \\ & + \text{MA}_{\text{anomaly score win 90d}}, \\ & + \text{MA}_{\text{anomaly score win 180d}}, \end{aligned} \quad (8)$$

where  $\text{MA}_{\text{anomaly score win } x}$  is the moving average of the anomaly score over a window with length  $x$ .

$$\text{Health category} = \begin{cases} \text{good} & \text{if health score} \leq 5, \\ \text{mediocre} & \text{if } 5 < \text{health score} \leq 10, \\ \text{bad} & \text{if health score} > 10. \end{cases} \quad (9)$$

The second technique is based on CUSUM (Page, 1955), which comes from the statistical process control (SPC) domain. CUSUM is designed to be more sensitive to small changes in the mean than for example Shewhart charts (used for example in Udo and Yar, 2021). The algorithm is run with different subgroup sizes, e.g., 10, 30, 90, and 180 d. Instead of using the subgroup mean and the overall mean, the subgroup median and overall median are used. The standard deviation of the subgroups is replaced with the robust standard deviation estimated using the MAD (see Eq. 5). This makes the algorithm more robust against anomalous trends. For each subgroup size, anomaly thresholds are calculated using Eq. (7). For each signal, there are three thresholds. These are common for all turbines in the wind farm. The signal health scores are calculated by summing the anomaly scores for the different subgroup sizes (meaning for each time step  $t$  the sum is taken over all the subgroup sizes) (Eq. 8). The health category is calculated using Eq. (9).

Assessing the performance of the anomaly detection algorithms on real data is a non-trivial task due to data imperfections. Imprecisions in the replacement dates, problems that are not resolved after a first attempt, incomplete event lists, preventive maintenance, etc. make it hard to automate the validation process. This means that each detection or non-detection needs to be validated by a human. Also, it introduces a certain inexactness into the validation process. For this reason, a somewhat different validation procedure will be used. The performance of the anomaly detection algorithms is assessed by calculating the percentage of failures that are correctly identified. This is the case when a cluster of bad health is found around the time of the failure. The ratio of false positives is also estimated. This is done using the following methodology. Firstly, turbines are selected that have experienced no known failures. This is the case for 10 turbines in total. For those turbines, it can be assumed that the components were probably relatively healthy during the observation period. This means that the number of bad health observations will be fairly limited. Bad health observations

that are found for those turbines are probably false positives. For each signal, the percentage of “bad” health observations is calculated. The median ratio for each signal over all the selected turbines is used as an approximation of the false positive ratio of the anomaly detection model.

### 3.5 The experiments

In total six demonstration experiments will be conducted. Five experiments will focus on the NBM model, and one experiment will focus on the anomaly detection algorithms. Experiment 1 compares the performance of the base elastic net regression with that of the implicit NBM. Experiment 2 evaluates the added value of using lagged predictors. Lagged predictors have also been used in Garlick and Watson (2009). Experiment 3 analyzes the impact of reducing the quantity of training data from 6 months per turbine to 2 months. In the state of the art, different numbers of training data are used. This is mainly driven by the number of data available. Experiment 4 discusses the added value of PCA-transformed input for the elastic net. Using PCA for preprocessing of the data is also done in Campoverde et al. (2022). Experiment 5 examines the added value of using more complex machine learning models like SVR (with a radial kernel) and light GBM. The performance of these models compared to that of the elastic net will say something about the importance of non-linearities. Experiment 6 compares the performance of the IOD–MAD and the CUSUM anomaly detection algorithms. For the analysis, the prediction error in the base elastic net model is used.

## 4 Results

### 4.1 Experiment 1: the added value of using the elastic net regression model on top of the results of the implicit NBM

The pipeline configuration is as follows: implicit NBM based on fleet median, explicit NBM based on elastic net regression, heuristic-based healthy data selection, full-grid-search hyperparameter tuning (5-fold CV), and 6 months of training data per turbine.

The first experiment investigates the usefulness of adding an explicit NBM (elastic net regression) model to the pipeline. One of the downsides of the implicit NBM (fleet median) is that it is unable to model turbine-specific transient behavior. Whether this is a serious problem depends on the case. However, if it is a problem, it can be solved by adding an explicit NBM to the pipeline. If the above reasoning is correct, it can be expected that the healthy test data RMSE will decrease considerably if the elastic net regression is added to the pipeline.

Figures 6 and 7 show that the prediction error when using the elastic net is indeed smaller than when only the implicit NBM is used. The most obvious improvement is that the

large negative spikes in the prediction error in the implicit NBM, which correspond to cool-downs caused by power downs of the turbine, are much smaller in the prediction error made by the elastic net. This indicates that the elastic net is modeling the transient behavior to a certain extent. The error is however still larger during transient phases than during steady-state phases. The healthy test data RMSEs in Fig. 8 further support the findings that the elastic net is a useful addition to the pipeline. The RMSEs of the prediction errors are substantially smaller when the elastic net is used.

Based on the results of the first experiment, it can be concluded that using the elastic net has a clear added value. The healthy test data RMSE is always smaller for the pipeline with the explicit NBM. The fact that the RMSE of the elastic net model is quite small shows that relatively simple and lightweight models can be useful for the modeling of normal behavior.

#### 4.2 Experiment 2: the added value of using lagged predictors

The pipeline configuration is as follows: implicit NBM based on fleet median; explicit NBM based on elastic net regression; heuristic-based healthy data selection; full-grid-search hyperparameter tuning (5-fold CV); 6 months training data per turbine; and lags 1, 2, and 3 of each predictor are added.

In the second experiment, the input data are augmented by adding the lagged values of the input signals (excluding the target signal that is being modeled). The idea behind using the lagged terms is that it makes it possible to model the time dependencies. This can be useful when modeling the transient behavior of the turbine. In steady-state situations where factors like active power change little, the positive impact will most likely be less clear. Three lags ( $t, t-1, t-2, t-3$ ) for each input signal are added. This means that the model can look up to 3 h in the past.

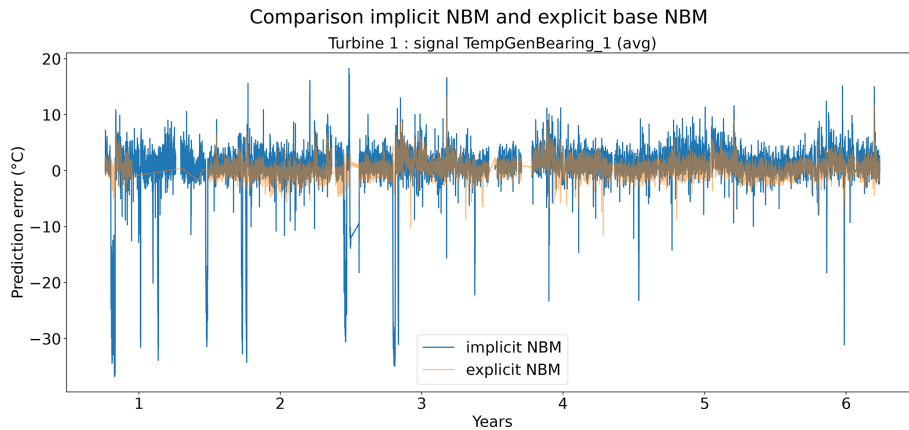
Figures 9 and 10 show that the difference between the prediction errors for the NBM with no lags and the NBM with three lags is marginal. In general, there is no clear difference visible between the two. Surprisingly, there is also no clear improvement to be found in the modeling of the transient behavior. Figure 11 gives an overview of the RMSE in the healthy data for all the target signals for the turbines in wind farms 1–5. The results show that adding the three lags to the model results for a majority of the signals in a marginal reduction in the median healthy test data RMSE.

Figure 12 shows the difference between the unhealthy and healthy median prediction errors ( $\Delta PE_{\text{unh-h}}$  calculated using Eq. 2) for the elastic net and multi-layer perceptron (MLP). The figure focuses on the results for three target signals, i.e., TempGenBearing\_1 (avg) ( $T_{\text{gen\_bear\_1}}$ ), TempGenBearing\_2 (avg) ( $T_{\text{gen\_bear\_2}}$ ), and TempStatorWind (avg) ( $T_{\text{stator}}$ ). For the three failures that are being examined, i.e., the rotor brush high-temperature failure, the generator bearing failure and the generator fan failure, it is assumed that the degra-

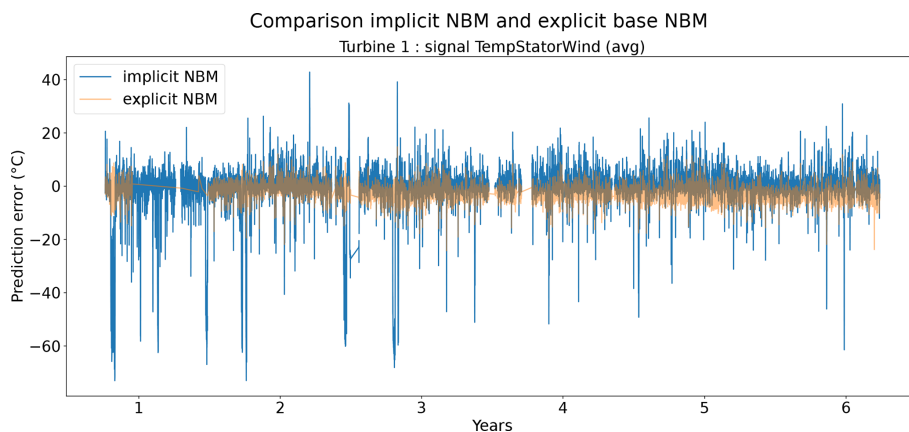
dation of the component can be observed directly or indirectly in the  $T_{\text{stator}}$ ,  $T_{\text{gen\_bear\_1}}$ , and  $T_{\text{gen\_bear\_2}}$  and the  $T_{\text{stator}}$ ,  $T_{\text{gen\_bear\_1}}$ , and  $T_{\text{gen\_bear\_2}}$  respectively. More specifically, in all three cases, an increase in the temperatures is expected when the component is damaged. This means that  $\Delta PE_{\text{unh-h}}$  should be positive. The more positive it is, the more useful the NBM is for anomaly detection.

The results in Fig. 12 show that for the elastic net the  $\Delta PE_{\text{unh-h}}$  only marginally increases when three predictor lags are used. For rotor brush high-temperature failures,  $\Delta PE_{\text{unh-h}}$  is clearly positive for the elastic net model with zero or three lagged predictors. This corresponds with the expectations. However,  $\Delta PE_{\text{unh-h}}$  is also clearly positive for  $T_{\text{gen\_bear\_1}}$ . This is more difficult to explain, since it is unlikely that the rotor brush high-temperature failure can be linked to abnormally high temperatures at the first generator bearing. For the generator bearing failure, the  $\Delta PE_{\text{unh-h}}$  is clearly positive for  $T_{\text{gen\_bear\_2}}$ . This is also in line with expectations. Since  $\Delta PE_{\text{unh-h}}$  is only positive for the second generator bearing and not for the first generator bearing, it is likely that most of the bearing failures happened at the second generator bearing. Unfortunately, the replacement information received from the wind turbine operator does not indicate which bearing has failed, so this statement can not be verified. For the generator fan failures, the  $\Delta PE_{\text{unh-h}}$  is clearly positive for the elastic net with three lags. This is not the case for the other signals, which is quite surprising, since the hypothesis is that a generator fan failure should be indirectly visible in all three generator signals. Furthermore, there is no evidence that the MLP (with zero or three lags) improves upon the elastic net model. The meaning of this will be discussed below. The results in Fig. 12 give only a first indication of whether the NBM is useful for anomaly detection or not. The lack of a clear positive  $\Delta PE_{\text{unh-h}}$  for signals where it is expected to be positive does not mean that all is lost. The anomaly detection techniques discussed in Experiment 6 are more sensitive for small deviations. This means that they can in some cases still detect anomalies even though the  $\Delta PE_{\text{unh-h}}$  is not clearly positive.

From the results of the second experiment, it can be concluded that the addition of three lags only marginally improves the model accuracy using healthy data. There is some evidence that the addition of the predictor lags results in NBM models with more anomaly detection potential. However, the improvement is in general small. Taking into account that adding the lags of all the predictors results in a strong increase in the dimensionality of the problem and the computational time, it is debatable whether the (limited) performance gains outweigh the extra cost. Reasons for the low added value of the lags can perhaps be an insufficient number of lags, a lack of information on the dynamics in the aggregated SCADA data, or the combination of transient and non-transient behavior. The first hypothesis seems to be unlikely, since limited experimentation using more lags showed no clear improvement in performance. The second hypothe-



**Figure 6.** Prediction error in explicit and implicit NBM model for the TempGenBearing\_1 (avg) signal of turbine 1.



**Figure 7.** Prediction error in explicit and implicit NBM model for the TempStatorWind (avg) signal of turbine 1.

sis is possible due to the fact that subtracting the fleet median from the SCADA data signals (see Preprocessing section) results in less autocorrelation in the data. Also, the aggregation of the data to the hour level will have an impact. This results in lagged predictor values that are less informative. The third hypothesis would imply that the dynamics of the steady state and the transient behavior of the turbine are so different they can not be learned by one elastic net model. This is a possible explanation. A solution to this problem would be to train a separate model for the transient and steady-state behavior or use a more complex model that is better able to learn the differences between the two states. The fact that no performance gains are achieved when using the MLP makes this hypothesis however somewhat less convincing. Nevertheless, to give a conclusive answer to which hypothesis is the correct one, further research is required.

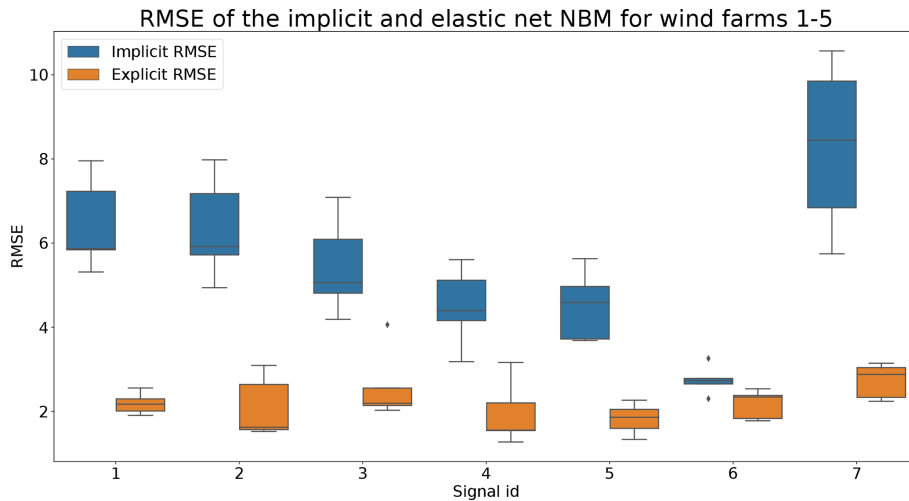
#### 4.3 Experiment 3: impact of reducing the training data to 2 months per turbine instead of 6

The pipeline configuration is as follows: implicit NBM based on fleet median, explicit NBM based on elastic net regres-

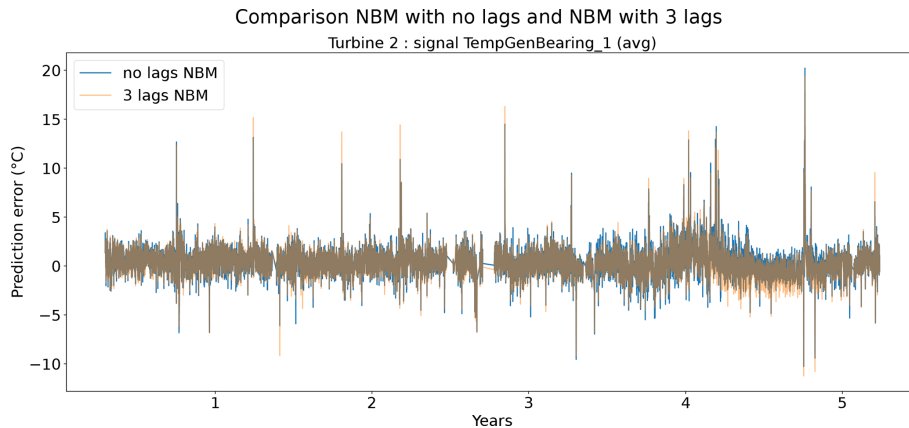
sion, heuristic-based healthy data selection, full-grid-search hyperparameter tuning (5-fold CV), and 2 months of training data per turbine.

In this experiment, the impact is analyzed of reducing the quantity of training data from 6 months per turbine to 2 months. This is relevant since fewer training data mean less computational and startup time when using the pipeline on a new wind farm. However, reducing the number of training data in general also comes at a cost. The model accuracy tends to decrease. The question is how much and whether it is outweighed by the advantages. Furthermore, since the training data are selected in chronological order (meaning the first  $X$  healthy observations), fewer training data mean that it becomes more likely that certain turbine conditions are missed (or are underrepresented in the data). This can for example be the case with long-duration power downs which cause exceptionally low temperatures for certain components. The most likely result will be that those conditions will be less well modeled, resulting in a larger prediction error. Depending on the use case of the pipeline, this might be a problem.





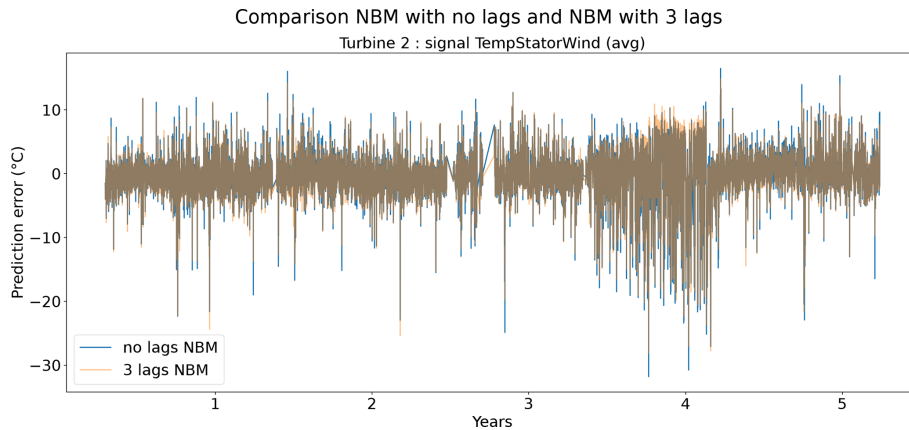
**Figure 8.** Comparison of RMSE of implicit and explicit NBM for wind farms 1–5. Signal ID 1: TempGearbBear\_1 (avg); signal ID 2: TempGearbBear\_2 (avg); signal ID 3: TempGearbInlet (avg); signal ID 4: TempGenBearing\_1 (avg); signal ID 5: TempGenBearing\_2 (avg); signal ID 6: TempRotorBearing (avg); signal ID 7: TempStatorWind (avg).



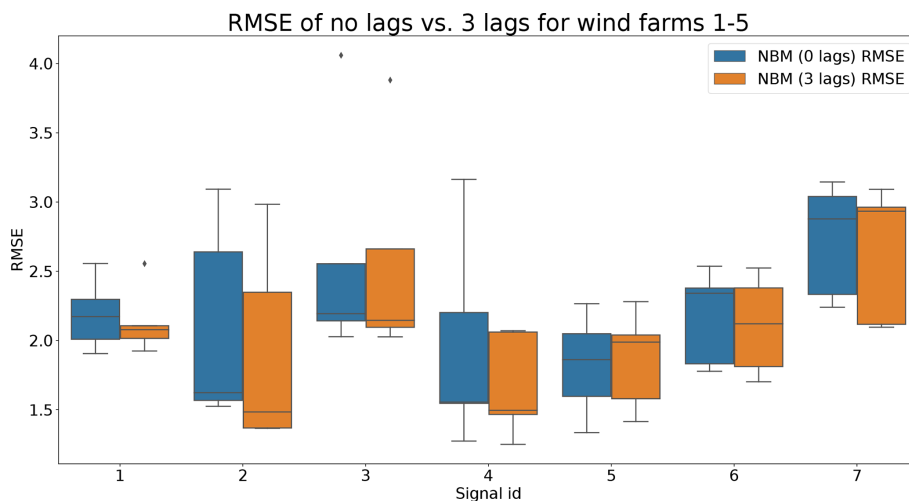
**Figure 9.** Prediction error in explicit and implicit NBM model for the TempGenBearing\_1 (avg) signal of turbine 2.

Figures 13 and 14 show indeed that there is some loss of prediction accuracy when the number of training data are reduced from 6 to 2 months. This shows itself as an increase in the prediction error. During steady-state behavior, this loss is not really visible, but during transient behavior, the loss of fit can be substantial (see for example Fig. 13). This is most likely caused by the fact that the training data do not (sufficiently) contain similar transient behavior examples. Figure 15 gives a more general overview of the RMSE results. It shows that the reduction in training data in general leads to an increase in the healthy test data RMSE. This increase is not massive but often also not negligible. For some signals, the median RMSE is slightly smaller. This reduction should not be considered evidence for a superior model but more an indication that the influence of the sample on the results is considerable. This is something that should be taken into account when analyzing the results.

Figure 16 shows the differences between the median prediction errors in the unhealthy and healthy data ( $\Delta PE_{\text{unh-h}}$ ). The results indicate that for detecting rotor brush high-temperature failures, 6 months of training data is better than 2. This is clear from the fact that  $\Delta PE_{\text{unh-h}}$  for  $T_{\text{stator}}$  is much larger when using 6 months of training data. The generator bearing failures are detected clearly in  $T_{\text{gen\_bear\_2}}$ . Again it can be observed that  $\Delta PE_{\text{unh-h}}$  is somewhat smaller when using only 2 months of training data. The analysis for the generator fan failures is somewhat less clear. On the one hand, there is, surprisingly, a larger  $\Delta PE_{\text{unh-h}}$  for the model trained on only 2 months of data when using  $T_{\text{gen\_bear\_2}}$  as the target signal. On the other hand, the model trained on 2 months of training data does not result in a positive  $\Delta PE_{\text{unh-h}}$  for the  $T_{\text{stator}}$  signal, contrary to the model with 6 months of training data. Overall it can be stated that fewer training data result in



**Figure 10.** Prediction error explicit and implicit NBM model for the TempStatorWind (avg) signal of turbine 2.



**Figure 11.** Comparison of RMSE of the no lags and the three lags NBM for wind farms 1–5. Signal ID 1: TempGearbBear\_1 (avg); signal ID 2: TempGearbBear\_2 (avg); signal ID 3: TempGearbInlet (avg); signal ID 4: TempGenBearing\_1 (avg); signal ID 5: TempGenBearing\_2 (avg); signal ID 6: TempRotorBearing (avg); signal ID 7: TempStatorWind (avg).

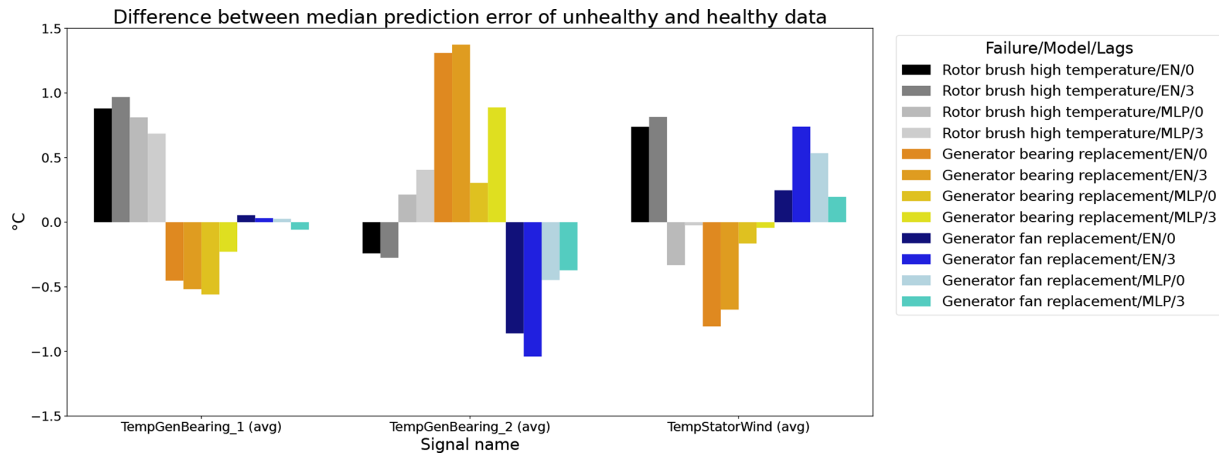
general in NBM models with less potential to be useful for anomaly detection.

Experiment 3 has shown that reducing the training data to 2 months results in reduced performance of the model. This is not so much a problem for the behavior states that are frequently shown by the turbine (e.g., steady-state behavior). It does have however a large influence on the states that are rare (e.g., long-term cool-downs). Furthermore, the results also show that the anomaly detection potential of the NBM decreases if the number of training data is reduced to 2 months. Whether the reduced performance of the NBM is a problem really depends on the use case. For some use cases, the larger prediction error in certain rare states is not an issue. For those cases, it might be useful to reduce the number of training data because this will reduce the computational burden of the pipeline. However, if rare behavior is important, the advantages may not outweigh the disadvantages.

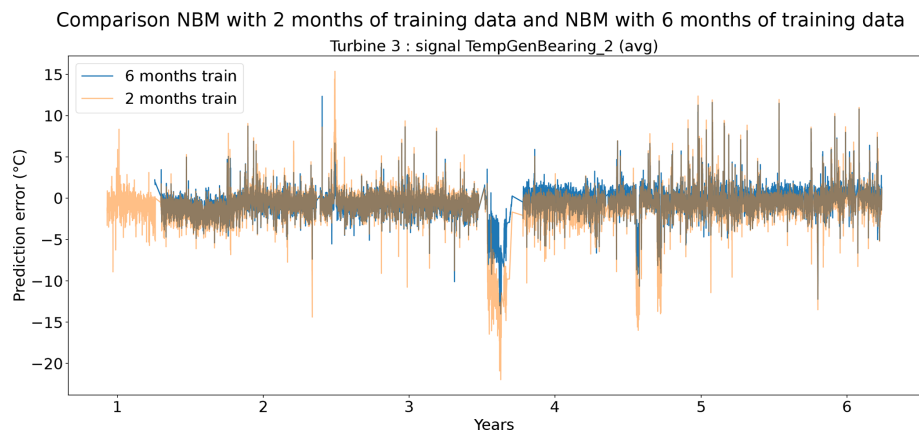
#### 4.4 Experiment 4: added value of a PCA transformation step before the explicit NBM

The pipeline configuration is as follows: implicit NBM based on the fleet median, PCA transformation, explicit NBM based on elastic net regression, heuristic-based healthy data selection, full-grid-search hyperparameter tuning (5-fold CV), and 6 months of training data per turbine.

In the fourth experiment, the impact of PCA-transforming (only a transformation, no dimensionality reduction) the data prior to the elastic net modeling is analyzed. Normally the elastic net algorithm should be able to handle high dimensional data with strong correlations between some of the predictors. However, in practice, there might still be some benefit of first PCA-transforming the data. Figure 19 shows that the prediction accuracy in general does not (or only marginally) improve when a PCA transformation step is added to the pipeline. For several signals, the opposite hap-



**Figure 12.** Difference between the median prediction error in the unhealthy and healthy data for elastic net and MLP models with no or three lags of the predictor variables. EN: elastic net; MLP: multi-layer perceptron; 0: no predictor lags; 3: three predictor lags ( $x_{t-1}$ ,  $x_{t-2}$ ,  $x_{t-3}$ ).



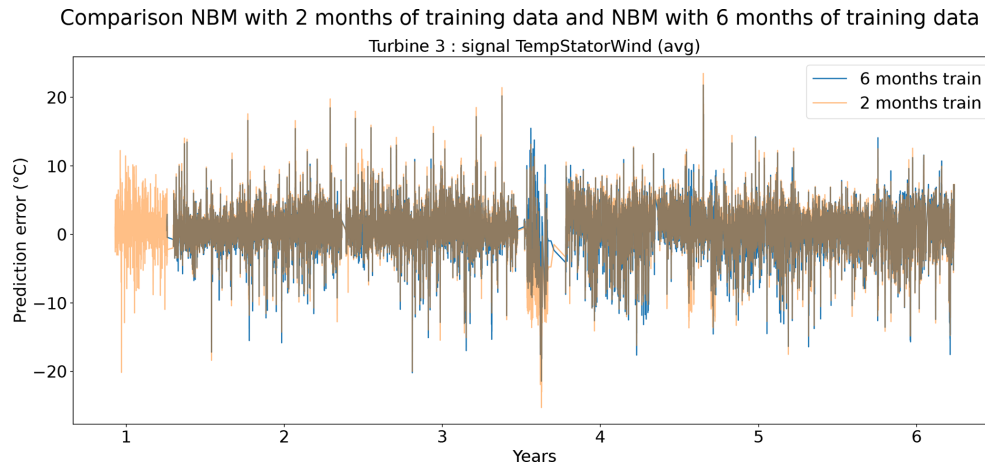
**Figure 13.** Prediction error in NBM models with 6 months and 2 months of training data for the TempGenBearing\_2 (avg) signal of turbine 3 of wind farm 5.

pens. However, these results hide certain interesting side effects. Figure 17 shows that the pipeline with the PCA tends to be a better modeler of the cool-downs than the model without the PCA. Figure 18 also shows that the pipeline without the PCA in some rare cases generates (unrealistically) large prediction errors, while this is not the case when the PCA is used. Furthermore, the training time (and hyperparameter tuning) is considerably shorter for the pipeline with the PCA, even though the number of combinations being tested during tuning is much larger (2728 for the pipeline with PCA, 341 for the pipeline without PCA). This might have something to do with the fact that the new features generated by the PCA transformation are uncorrelated. This can improve the training of the elastic net. For example, for wind farm 5 the hyperparameter tuning without PCA took 4329 s, while with the PCA it took only 3859 s.

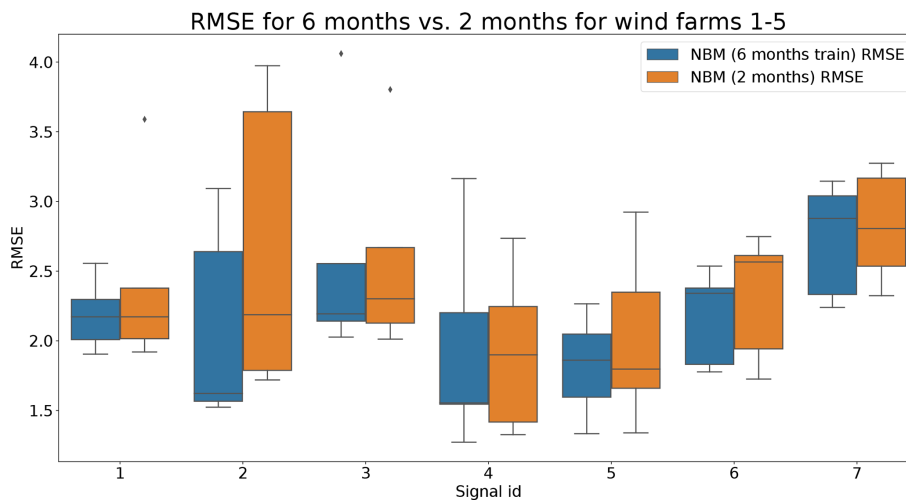
Figure 20 shows the differences between the unhealthy and the healthy prediction error ( $\Delta\text{PE}_{\text{unh-h}}$ ). The results show

that using the PCA as a preprocessing step has in general a small positive impact on  $\Delta\text{PE}_{\text{unh-h}}$ . For the rotor brush high-temperature failures, there is no clear change in the positive difference for the  $T_{\text{stator}}$  signal for the model with and without PCA preprocessing. For the generator bearing failures, the positive difference for the  $T_{\text{gen\_bear\_2}}$  signal is slightly larger when using the model with PCA preprocessing. The same is true for the  $T_{\text{stator}}$  signal for the generator fan failures.

Based on the results of the fourth experiment, it can be concluded that PCA-transforming the data prior to the elastic net modeling can be beneficial in some cases. Although the healthy test data RMSE does not decrease, it does avoid certain large (erroneous) prediction errors and it seems to improve the modeling of the transient behavior. It also seems to improve somewhat the distinction the NBM can make between healthy and unhealthy data. Furthermore, it also reduces the training time considerably. This means that PCA-transforming the data prior to the elastic net modeling has



**Figure 14.** Prediction error in NBM models with 6 months and 2 months of training data for the TempStatorWind (avg) signal of turbine 3 of wind farm 5.



**Figure 15.** Comparison of RMSE when using 6 months or 2 months of training data for the NBM model for wind farms 1–5. Signal ID 1: TempGearBear\_1 (avg); signal ID 2: TempGearBear\_2 (avg); signal ID 3: TempGearInlet (avg); signal ID 4: TempGenBearing\_1 (avg); signal ID 5: TempGenBearing\_2 (avg); signal ID 6: TempRotorBearing (avg); signal ID 7: TempStatorWind (avg).

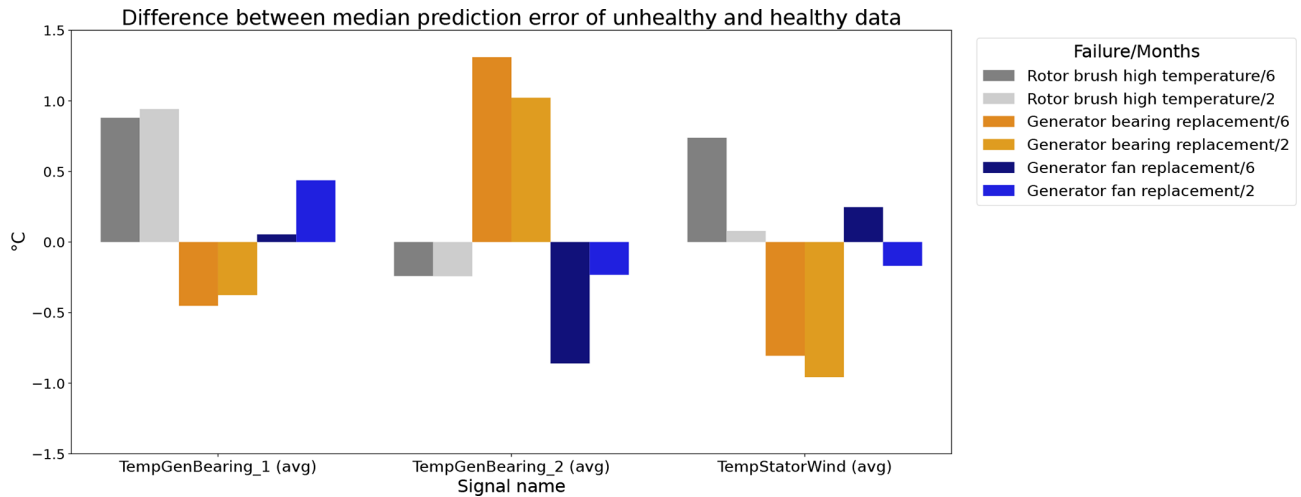
some benefits. Whether these outweigh the added complexity and the loss of the original features (the principal components are linear combinations of all the original features) depends of course on the case.

#### 4.5 Experiment 5: added value of more complex NBM models

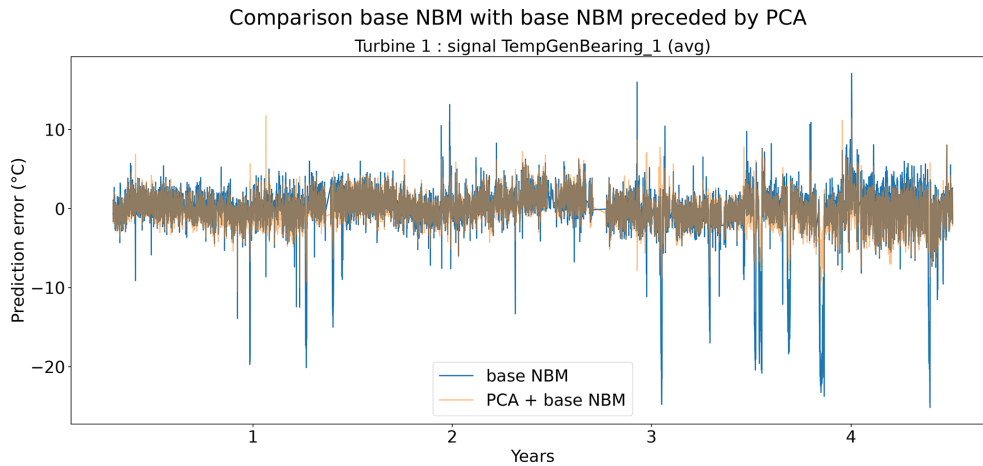
The pipeline configuration is as follows: implicit NBM based on the fleet median; explicit NBM based on elastic net regression, SVR, light GBM, or MLP; heuristic-based healthy data selection; full-grid-search hyperparameter tuning for elastic-net; SVR and light GBM and randomized grid search tuning for MLP (5-fold CV); and 6 months of training data per turbine.

The fifth experiment focuses on the complexity of the relations between the inputs (predictors) and the outputs (targets) of the NBM problem. The elastic net regression is a relatively simple model that works well when the problem is linear (meaning linear in the parameters). However, if the problem is highly non-linear, this type of model is not really suitable. In the latter case, more complex models like tree-based algorithms or neural networks are more appropriate. The trade-off is however that these models are more “black box” and require in general many more training data. Nevertheless, it is still interesting to analyze the performance of these models. If they clearly outperform the elastic net model, then that is evidence for the existence of non-linear relations. The performance of the elastic net will be compared with that of an SVR with a radial kernel (also used in Castellani et al., 2021), a light GBM (which is similar to the gradient-boosting algo-





**Figure 16.** Difference between the median prediction error in the unhealthy and healthy data ( $\Delta PE_{unh-h}$ ) for elastic net models with 6 months and 2 months of training data; 6: 6 months of training data; 2: 2 months of training data. The results are shown for TempGenBearing\_1 (avg) ( $T_{gen\_bear\_1}$ ), TempGenBearing\_2 (avg) ( $T_{gen\_bear\_2}$ ), and TempStatorWind (avg) ( $T_{stator}$ ).



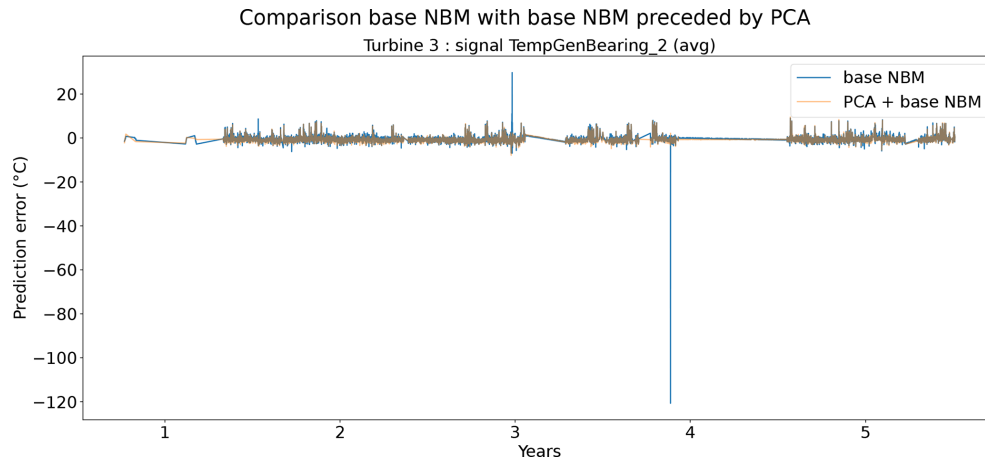
**Figure 17.** Prediction error in base NBM and PCA + NBM for the TempGenBearing\_1 (avg) signal of turbine 1 of wind farm 1.

gorithms used in Udo and Yar, 2021; Maron et al., 2022; and Beretta et al., 2021), and an MLP.

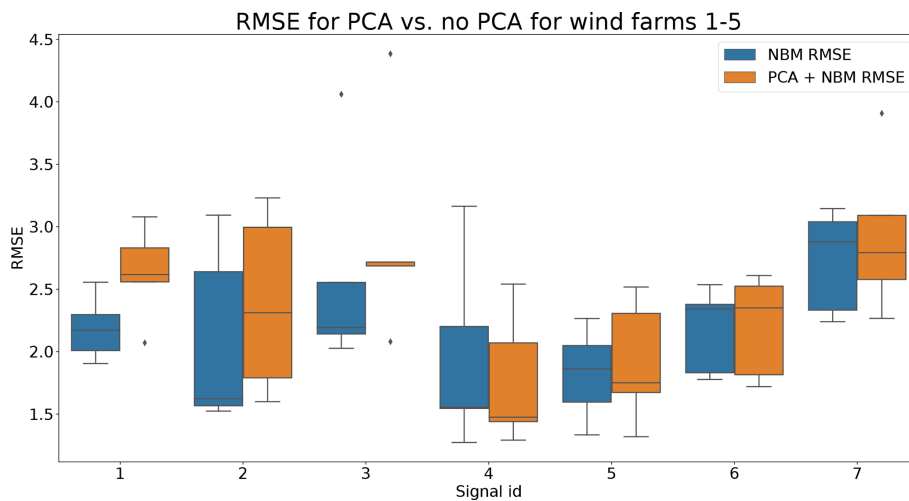
Figure 21 shows that the prediction error in the four models is roughly similar in size, with some large upward or downward spikes at certain points in time for the SVR. Figure 22 shows however that under certain conditions the SVR and light GBM performance severely degrades. This is the case during large long-term cool-downs. This problem also impacts the elastic net and the MLP but to a lesser extent. With limited examples in the training dataset, the SVR and light GBM have difficulties estimating the normal behavior in those cases. The elastic net, which is a much simpler model, might use relatively accurate extrapolations for estimating those cases. The MLP performs during cool-downs clearly better than the SVR and light GBM but does not completely reach the performance of the elastic net. A pos-

sible explanation for this is that the MLP is better at learning the non-linearities than the SVR and light GBM, but the problems it has with extrapolation mean that it does not perform as well as the elastic net. Figure 23 gives an overview of the healthy test data RMSEs. The results show that the SVR never performs best, and it often performs significantly worse than the three other models. The light GBM and MLP perform marginally better than the elastic net for some signals; however, the improvement is small. For some signals, they perform worse than the elastic net. This means that there is some evidence for non-linearities when modeling some of the signals. The evidence is however weak.

Figure 24 shows the differences between the unhealthy and the healthy prediction error ( $\Delta PE_{unh-h}$ ). The purpose of this plot is to see whether using more complex machine learning models results in NBM models that are better at distinguish-



**Figure 18.** Prediction error in base NBM and PCA + NBM for the TempGenBearing\_2 (avg) signal of turbine 3 of wind farm 3.

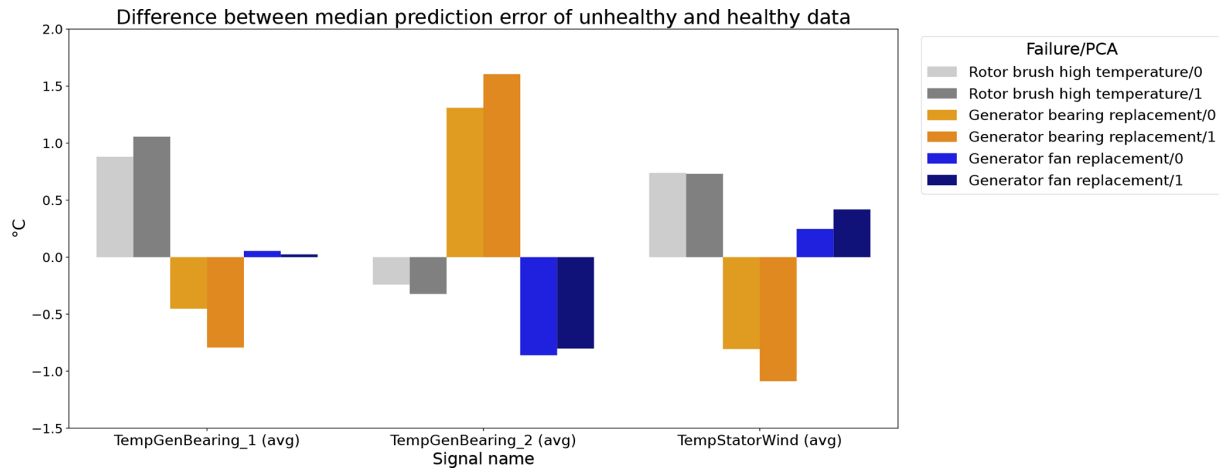


**Figure 19.** Comparison of the RMSE when the data are PCA-transformed to when the data are not PCA-transformed for wind farms 1–5. Signal ID 1: TempGearbBear\_1 (avg); signal ID 2: TempGearbBear\_2 (avg); signal ID 3: TempGearbInlet (avg); signal ID 4: TempGenBearing\_1 (avg); signal ID 5: TempGenBearing\_2 (avg); signal ID 6: TempRotorBearing (avg); signal ID 7: TempStatorWind (avg).

ing healthy from unhealthy data. For the rotor brush high-temperature failures, the elastic net model outperforms the other models. This is clear from the fact that the  $\Delta PE_{\text{unh-h}}$  for the  $T_{\text{stator}}$  signal is the largest for the elastic net. For the generator bearing failure, the story is the same. The elastic net outperforms the other models. The prediction error difference for the  $T_{\text{gen\_bear\_2}}$  signal is the largest for the elastic net. For the generator fan failures, the results indicate that the more complex models somewhat outperform the elastic net model. The light GBM has a larger  $\Delta PE_{\text{unh-h}}$  for the  $T_{\text{gen\_bear\_1}}$  signal, and the MLP and SVR have a larger  $\Delta PE_{\text{unh-h}}$  for the  $T_{\text{stator}}$  signal. These results indicate that using a combination of several of the more complex models might result in a better anomaly detection performance.

The results of experiment 5 show some (weak) evidence that the relation between the predictors and some of the tar-

gets is non-linear. This means that some non-linear models (e.g., from the traditional machine learning or the deep learning domain) might improve the modeling of the healthy data. The improvements obtained by the light GBM and the MLP are however small. Furthermore, if the  $\Delta PE_{\text{unh-h}}$  is analyzed, then there is no evidence that the more complex models outperform the elastic net for rotor brush high-temperature and generator bearing failures. However, for generator fan failures, it might be beneficial to use the more complex models or a combination of the models. Furthermore, the results also show that the more complex models are more susceptible to the underrepresentation of certain states in the training dataset. This can lead to severe performance degradation.



**Figure 20.** Difference between the median prediction error in the unhealthy and healthy data for elastic net models with and without PCA preprocessing; 0: no PCA preprocessing; 1: PCA preprocessing. The results are shown for TempGenBearing\_1 (avg) ( $T_{\text{gen\_bear\_1}}$ ), TempGenBearing\_2 (avg) ( $T_{\text{gen\_bear\_2}}$ ), and TempStatorWind (avg) ( $T_{\text{stator}}$ ).

#### 4.6 Experiment 6: identifying anomalies in the prediction error using iterative outlier detection and CUSUM

The sixth and last experiment focuses on detecting anomalies in the prediction error made by the NBM. The previous experiments all focused on the NBM itself because it is the basis of the anomaly detection pipeline. In this section, the focus shifts to the anomaly detection algorithms that can be used to find abnormal prediction error patterns. As shown in the “Overview of the state of the art” section, there are multiple ways this can be achieved. Testing and comparing all the methods that have been developed is obviously unfeasible. For this reason, a selection will be made that takes into account the requirements of the industry, namely maintainability, transparency, computational efficiency, and robustness. Preference is given to univariate statistical techniques that have been thoroughly studied, i.e., IOD–MAD and CUSUM. The accuracy of the techniques will be, as described in the “Design of demonstration experiments for evaluating the performance of techniques used in the state-of-the-art literature” section, assessed in both a quantitative and a qualitative fashion. For practical reasons, the figures shown in this section are only a subset of all figures that can be generated from the results.

##### 4.6.1 Generator bearing replacement

Generator bearing failures can normally be detected in the temperatures of the bearings. When damage or wear is forming, the temperatures start to increase due to increased friction. The information from the wind turbine operator, unfortunately, does not mention which bearing was replaced, e.g., bearing 1 or 2. However, in general, this can be deduced from the results because the health degradation is much more pronounced for one of the two bearings. Generator bearing

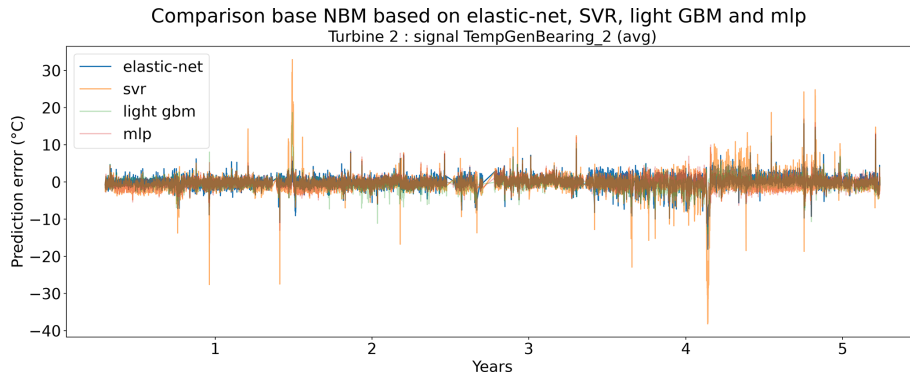
failures normally form slowly over time. This means that health degradation shows itself mostly during a prolonged period of time. The strength of the degradation depends however also on how fast the bearing was replaced. If it was replaced as part of preventive maintenance, then the degradation signal will most likely be less strong compared to situations in which the bearing truly failed. In total, information on four generator bearing replacements is available.

The generator bearing replacement in Figs. 25 and 26 is detected by the IOD–MAD and the CUSUM algorithms, although the latter detects it only lightly. IOD–MAD tends to generate more anomalies that can not be associated with this failure compared to CUSUM. The generator bearing replacement in Figs. 27 and 28 is also clearly detected by both the IOD–MAD and the CUSUM algorithm. The fact that both algorithms detect the replacement strongly indicates that the degradation of the bearing most likely was severe. Both algorithms also raise some anomalies around year 5. It is unclear at the moment what the reason for this is.

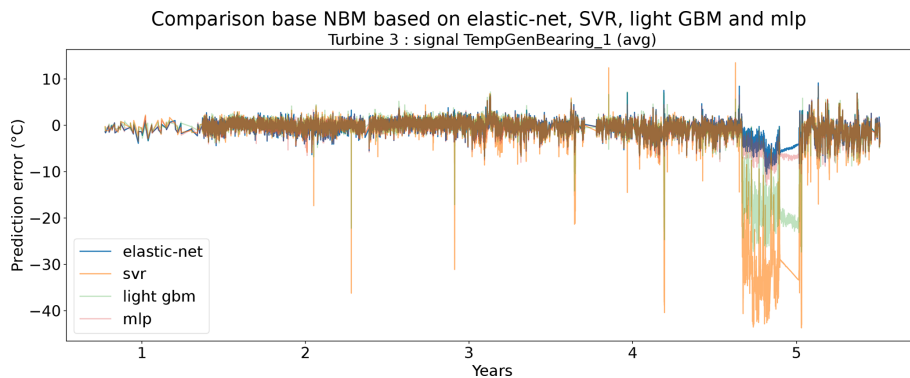
The overall results show that the IOD–MAD algorithm is able to identify three out of four generator bearing replacements, while CUSUM found only two out of four. IOD–MAD has however the tendency to generate more bad health flags that can not be associated with the failures. This probably means that the false positive rate for this algorithm is higher than for CUSUM. CUSUM on the other hand seems to be much less sensitive (probably not sensitive enough). It is likely that changing the hyperparameters would improve the performance. However, parameter tuning with such a small number of examples is difficult.

##### 4.6.2 Generator fan replacement

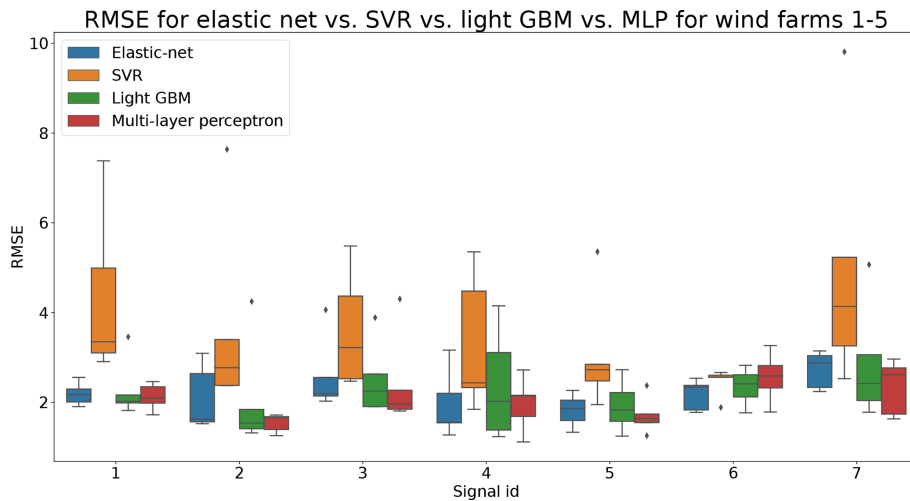
Generator fan problems can only be observed indirectly for the turbines of wind farms 1–5. This is due to the fact that



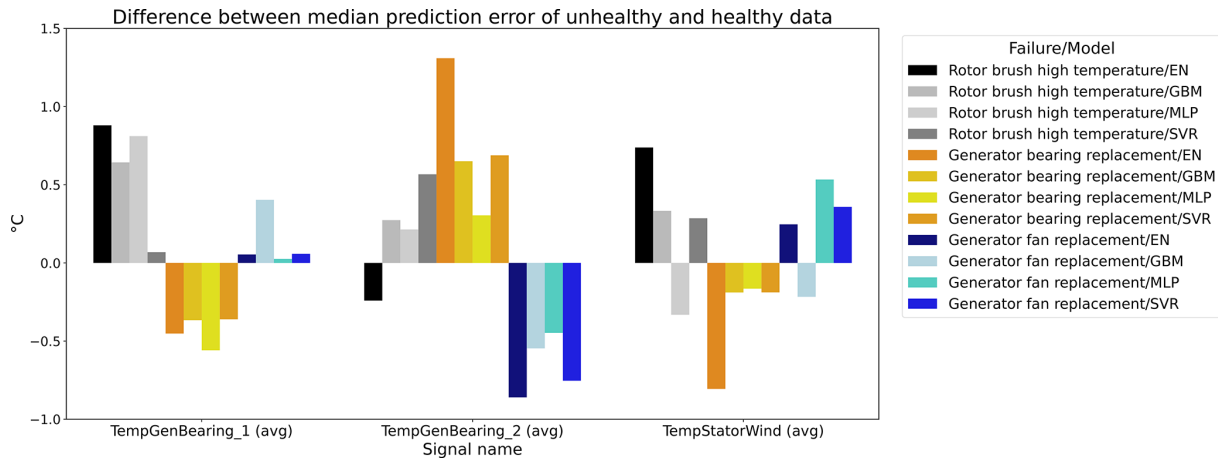
**Figure 21.** Prediction error in elastic net, SVR, light GBM, and MLP NBM models for TempGenBearing\_2 (avg) signal of turbine 2 of wind farm 5.



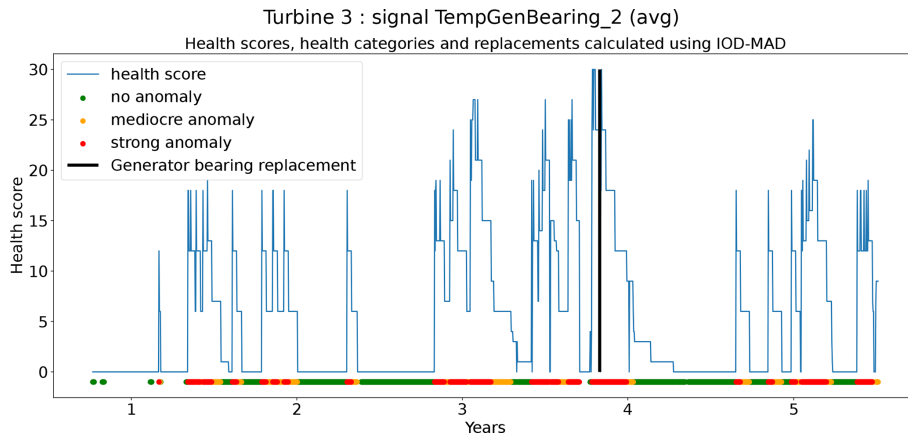
**Figure 22.** Prediction error in elastic net, SVR, light GBM, and MLP NBM models for TempGenBearing\_1 (avg) signal of turbine 3 of wind farm 2.



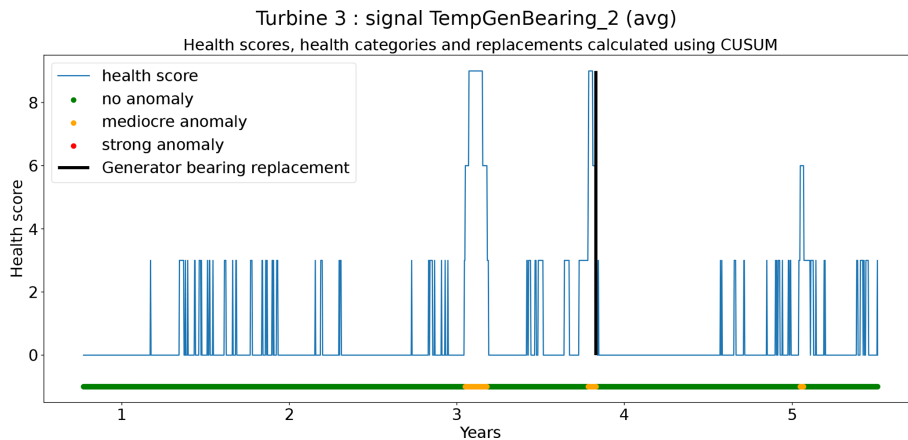
**Figure 23.** Difference between the median prediction error in the unhealthy and healthy data. Comparison of the RMSE of the elastic net, the SVR, the light GBM, and the MLP NBM models for wind farms 1–5. Signal ID 1: TempGearbBear\_1 (avg); signal ID 2: TempGearbBear\_2 (avg); signal ID 3: TempGearbInlet (avg); signal ID 4: TempGenBearing\_1 (avg); signal ID 5: TempGenBearing\_2 (avg); signal ID 6: TempRotorBearing (avg); signal ID 7: TempStatorWind (avg).



**Figure 24.** Difference between the median prediction error in the unhealthy and healthy data for elastic net, light GBM, multi-layer perceptron, and support vector regression models. EN: elastic net; GBM: light gradient-boosting machine; MLP: multi-layer perceptron; SVR: support vector regression. The results are shown for TempGenBearing\_1 (avg) ( $T_{gen\_bear\_1}$ ), TempGenBearing\_2 (avg) ( $T_{gen\_bear\_2}$ ), and TempStatorWind (avg) ( $T_{stator}$ ).



**Figure 25.** Health score and category based on the IOD-MAD algorithm for TempGenBearing\_2 (avg) of turbine 3 of wind farm 3.



**Figure 26.** Health score and category based on the CUSUM algorithm for TempGenBearing\_1 (avg) of turbine 3 of wind farm 3.



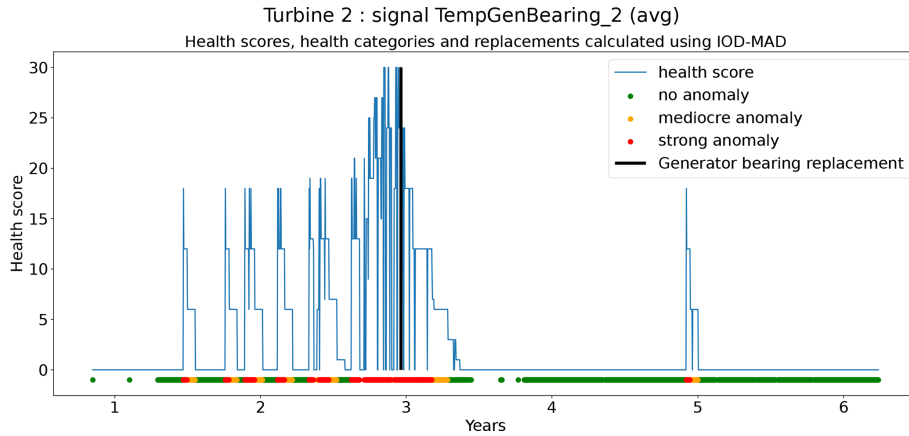


Figure 27. Health score and category based on the IOD-MAD algorithm for TempGenBearing\_1 (avg) of turbine 2 of wind farm 4.

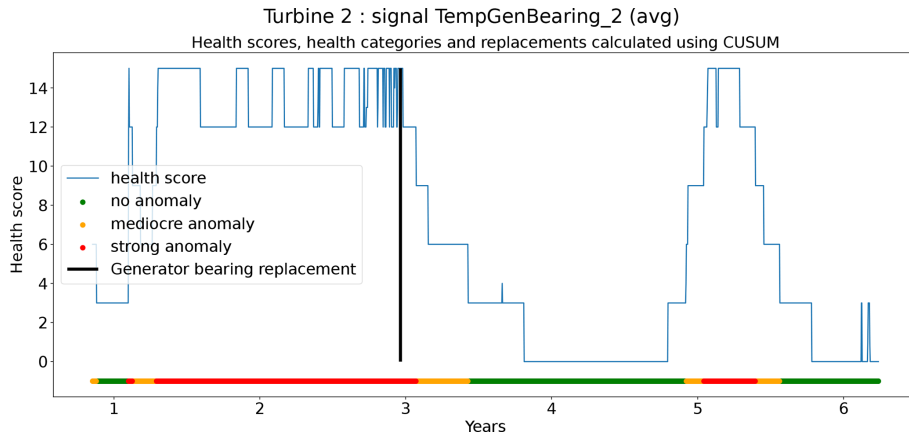


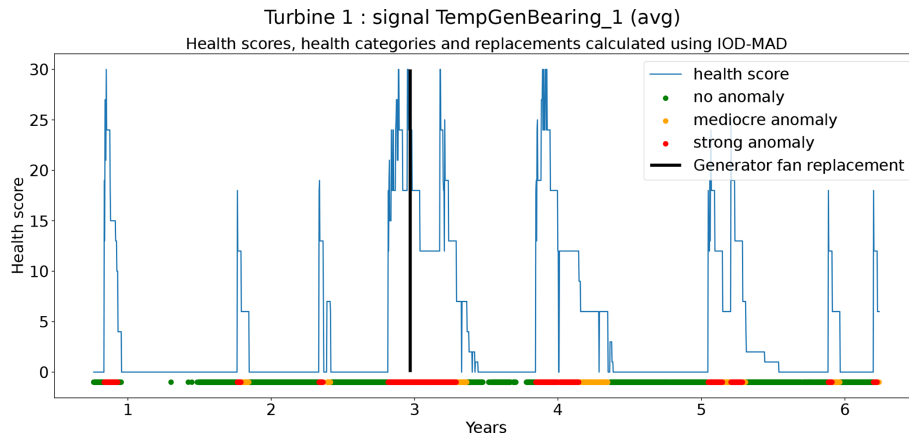
Figure 28. Health score and category based on the CUSUM algorithm for TempGenBearing\_1 (avg) of turbine 2 of wind farm 4.

there is no signal available in the SCADA data that is directly linked with the fans. However, it can be assumed that the failure of a generator can be observed indirectly by analyzing the temperatures of generator components. Even though this should be possible, indirect observations are probably less clear than direct observations, meaning that the health degradation will most likely be much less clear. Initially, it was assumed that a generator fan failure could be identified if all three generator signals, e.g., TempGenBearing\_1 (avg), TempGenBearing\_2 (avg), and TempStatorWind (avg), show health degradation. However, in practice, it appears that this is not the case and that in general the failure can only be spotted in one signal, namely the TempGenBearing\_1 (avg) signal. The result of this is that it is often difficult to make a distinction between a generator bearing failure and a generator fan failure. This means that when bad health is observed in the TempGenBearing\_1 (avg) signal, the user will receive the warning that there is an issue with the bearing or the fan.

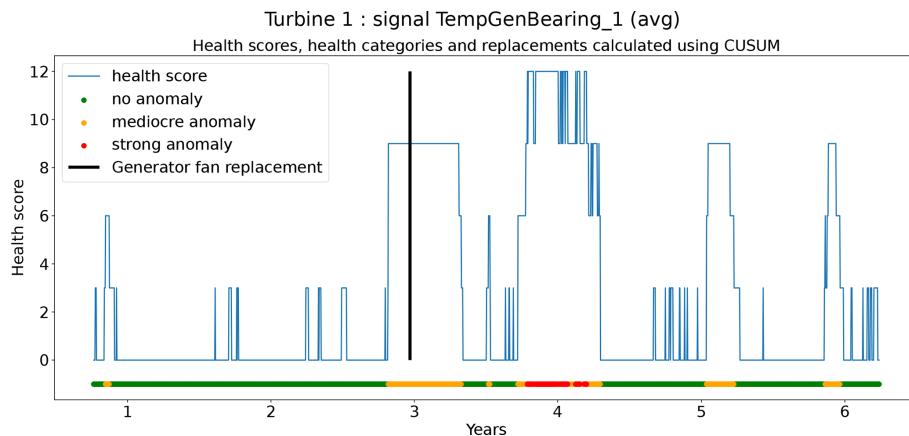
The datasets for wind farms 1–5 contain in total three usable examples. The three examples show that IOD-MAD finds all the replacements correctly. CUSUM misses one.

Figures 29 and 30 show the results for the third generator fan failure in the dataset. The IOD-MAD algorithm also raises bad health flags at some other points in time. This is less the case for CUSUM. It would be a bit premature to immediately decide that those are false positives. After all, the TempGenBearing\_1 (avg) signal can also be influenced by issues with the bearings or some other factors. In practice, it would mean that at those points in time the user would also receive a warning for a potential fan or bearing failure. However, the number of those cases is relatively small given that the observation window is nearly 6 years long.

Overall the IOD-MAD algorithm identified three out of three generator fan replacements correctly, while CUSUM found two out of three. The results show also that CUSUM is more conservative than IOD-MAD, which makes it less capable. However, it also generates fewer bad health flags that can not be linked to the generator fan replacements.



**Figure 29.** Health score and category based on the IOD–MAD algorithm for TempGenBearing\_1 (avg) of turbine 1 of wind farm 5.



**Figure 30.** Health score and category based on the CUSUM algorithm for TempGenBearing\_1 (avg) of turbine 1 of wind farm 5.

#### 4.6.3 Rotor brush high-temperature failure

The datasets for wind farms 1–5 contain five examples of rotor brush high-temperature failures that can be used for the validation of the anomaly detection models. Two failures had to be excluded due to missing SCADA data. Just like the generator fan replacements, there is no direct way to identify these failures. It is however assumed that they can be identified indirectly through the temperature of the stator windings (although there is some debate whether this is always the case). Just like for the generator fan replacement case, it means that the signal might not always be very strong.

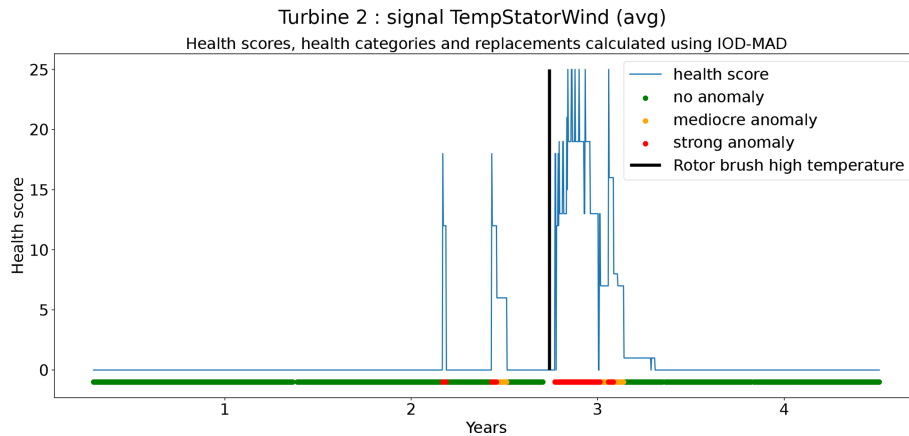
The first rotor brush high-temperature failure (Figs. 31 and 32) is detected by both algorithms. For IOD–MAD, the main detections happen after the date of the replacement, but given the uncertainty about the event dates and the fact that sometimes an issue is not solved after the first try, it can still be considered a correct detection. The detection by the CUSUM algorithm is very clear. The generator rotor brush high-temperature failure in Figs. 33 and 34 is also detected

correctly. However, the detection strength is lower for the CUSUM algorithm than for the IOD–MAD algorithm.

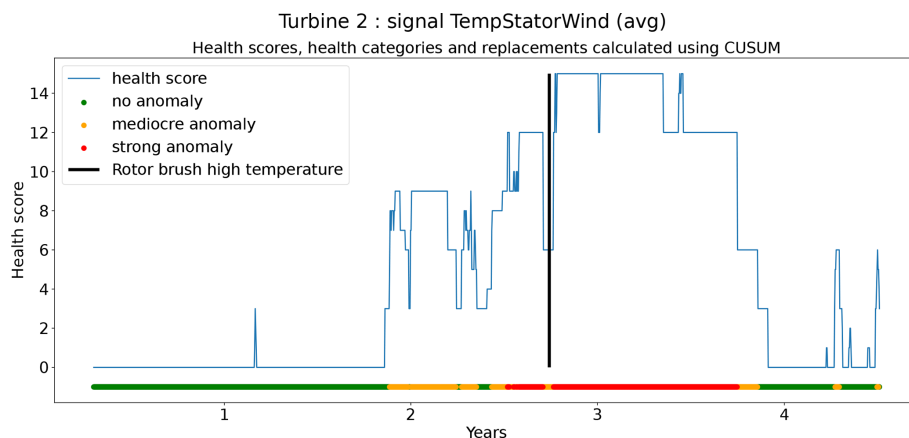
The overall results show that both algorithms can quite reliably detect the rotor brush high-temperature failures using the temperature of the stator windings. Four out of five failures were detected. However, from the results, it is also clear that the detections are not always very clear. Nevertheless, they are still sufficiently different from the rest of the data to be useful for the user.

#### 4.6.4 False positive ratio

The last part of this experiment is assessing the false positive ratio of the anomaly detection models. The analysis will be focused on the IOD–MAD algorithm because it has the highest detection accuracy. The results in Table 2 show that false positive ratios for the different signals are 0.12 for the TempGenBearing\_1 (avg) signal and 0.08 for the TempGenBearing\_2 (avg) and the TempStatorWind (avg) signals. This is a relatively small ratio given that there is always some uncertainty about the health of the data. The results show how-



**Figure 31.** Health score and category based on the IOD–MAD algorithm for TempStatorWind (avg) of turbine 2 of wind farm 1.



**Figure 32.** Health score and category based on the CUSUM algorithm for TempStatorWind (avg) of turbine 2 of wind farm 1.

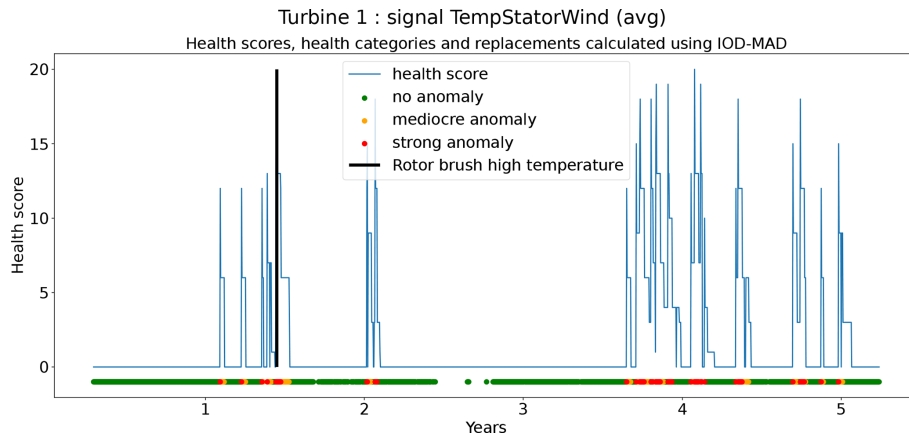
ever that for some turbines, the false positive ratio for certain signals can be high. This is for example the case for the TempGenBearing\_2 (avg) signal of turbine 1 of wind farm 2. This might indicate that there is a hidden underlying issue with the second bearing of this turbine. Nevertheless, it can be concluded that in general, the false positive ratio is quite low.

## 5 Conclusion and future research

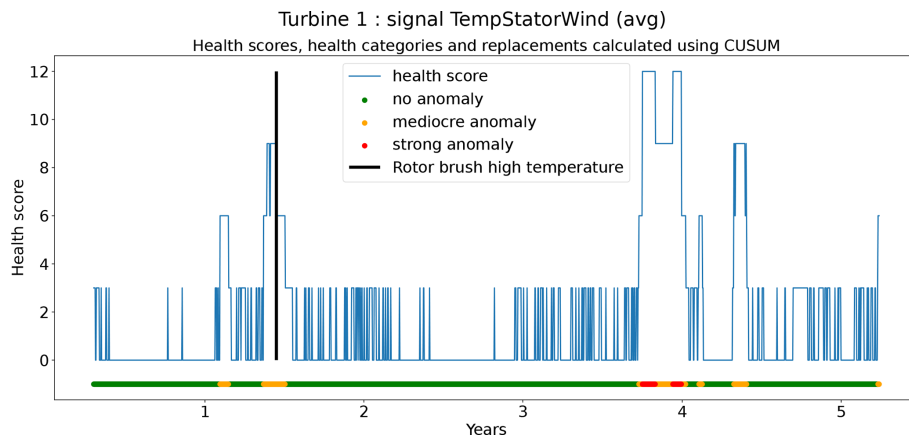
This paper gives an overview of recent research on the condition monitoring of wind turbines using SCADA data and the NBM framework. The goal is to give the reader an idea of what the current state of the art is, e.g., what has been tried and what techniques are performed using data from operational wind farms. This is done by first presenting a structured overview of the current state of the art. This gives an idea of how an NBM pipeline is normally set up and which techniques are used for the different steps. In the second part of the paper, several techniques from the state of the art are selected and applied to data from several real operational

wind farms. This is done through six demonstration experiments. The different techniques are compared, and their performance is thoroughly analyzed. Five experiments focus on the NBM model, and one experiment focuses on the analysis of the prediction error.

The first experiment examines the modeling performance of a relatively simple NBM model, i.e., the elastic net. The results show that the model is a capable modeler, even during the transient behavior of the turbine. The second experiment discusses the impact of using the lagged values of the predictors as input to the elastic net model. The results show only a marginal improvement of the model quality (minor reduction in the healthy test data RMSE). The modeling of the transient behavior is not noticeably better. Potential explanations for this are, firstly, data with a 1 h resolution might be insufficient; secondly, three lagged values might not be enough; and thirdly, a single model for transient and non-transient behavior might be problematic due to differences in the behavior dynamics. The third experiment examines the impact of reducing the quantity of training data from 6 to 2 months. The results show a reduction in the modeling ac-



**Figure 33.** Health score and category based on the IOD–MAD algorithm for TempStatorWind (avg) of turbine 1 of wind farm 4.



**Figure 34.** Health score and category based on the CUSUM algorithm for TempStatorWind (avg) of turbine 1 of wind farm 4.

curacy (as expected). The risk of underrepresentation of certain wind turbine states in the training data also increases. This can result in degraded model performance for these states. The fourth experiment discusses the added value of PCA-transforming the output of the implicit NBM before it is given to the elastic net. The results show that the overall model performance does not improve when the PCA transformation is used. However, some abnormally large prediction errors disappear and the run time of the pipeline is significantly reduced. A possible explanation for this is the fact that the PCA breaks the correlation between the predictors, which results in a more stable model and faster convergence. The fifth experiment investigates whether more complex machine learning models are useful for the pipeline. The results show that the SVR with a radial kernel performs overall worse than the elastic net, while the light GBM and MLP perform slightly better (for some signals). However, these more complex models can suffer from severe model degradation during the transient behavior of the turbine. A possible explanation can be found in the underrepresentation of this behavior in the training dataset. A possible solution for this

might be oversampling the minority behavior. The sixth and last experiment tests two different univariate anomaly detection techniques (IOD–MAD and CUSUM) that generate a health score for the signals. The results show that IOD–MAD is able to identify the failures more accurately, at the cost of more alerts during periods that can not immediately be linked to a failure. Most generator bearing replacements, generator fan replacements, and rotor brush high-temperature failures can be detected accurately. Furthermore, the number of false positives generated by the IOD–MAD algorithm is quite low. The end result is an NBM pipeline with relatively low computational demands, which is quite robust, has a limited number of models, and is able to detect three different failure types accurately on five different wind farms without changes to the configuration of the pipeline.

The overview of the state of the art shows that at the moment a lot of research is being done on condition monitoring and failure prediction for wind farms using SCADA data. The NBM methodology is a popular methodological choice for this. Many different configurations (preprocessing, NBM, and analysis of the prediction error) have been tried. Never-

**Table 2.** Overview of the ratio of bad health observations on the total number of observations. WF stands for wind farm and T for turbine.

WF/T	TempGenBearing_1 (avg)	TempGenBearing_2 (avg)	TempStatorWind (avg)
1/1	0.03	0.03	0.14
1/4	0.09	0.09	0.06
2/1	0.28	0.43	0.03
3/1	0.12	0.25	0.08
3/2	0.15	0.04	0.08
3/4	0.02	0.08	0.02
4/3	0.14	0.17	0.01
4/4	0.12	0.24	0.11
5/2	0.16	0.04	0.11
5/3	0.06	0.05	0.09
Median	0.12	0.08	0.08

theless, there are still some blind spots that might be interesting for future research. Firstly, a thorough structured analysis of the impact of different preprocessing techniques on the performance of the condition monitoring could be useful. This will give insights into which techniques work well and give future researchers a basis to start from. This should avoid the situation there is today where often ad hoc decisions are taken without proper explanation of the motivation as to why or a clear idea of what the impact will be on the final results. Secondly, a more thorough comparative analysis of different NBM models (e.g., statistical, traditional machine learning, deep learning) might be useful, preferably taking into account the demands/remarks from the industry. Thirdly, most research now focuses on SCADA data with a 10 min resolution. With more and more data becoming available from wind turbines and improved connectivity, research can be done on data with a higher resolution. Comparative studies of the performance of condition monitoring using 10 min SCADA data and data with a higher resolution (e.g., 1 min, 10 s, 1 s) might be interesting. It will give the industry an idea of what the added value is of collecting SCADA data with a higher resolution. Furthermore, it most likely will make it possible to detect events or failures that are short-lived. A thorough comparative analysis of the techniques used for the analysis of the prediction errors would also be useful. Fourthly, these analyses should preferably be done on data from real operational wind farms. Furthermore, it would help the research on this topic a lot if data from several operational wind farms could be made public. This would make it possible to use these datasets as standards, which would make it easier to compare the performance of different techniques.

**Data availability.** The data are covered by an NDA. This means that the data cannot be shared.

**Author contributions.** XC, TV, PJD, and JH did the conceptualization; XC developed the methodology, did the data curation and the formal analysis, and wrote the paper; XC, TV, PJD, and JH reviewed and edited the paper; JH and AN acquired funding.

**Competing interests.** The contact author has declared that none of the authors has any competing interests.

**Disclaimer.** Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Acknowledgements.** Xavier Chesterman, Timothy Verstraeten, Pieter-Jan Daems, Ann Nowé, and Jan Helsen would like to thank the Flemish AI Research Program, H2020 Energy, Fonds Wetenschappelijk Onderzoek and Agentschap Innoveren en Ondernemen for their financial support.

**Financial support.** This research has been supported by the Flemish Government (AI Research Program), H2020 Energy (grant no. 872592), the Fonds Wetenschappelijk Onderzoek (through SBO Robustify, grant no. S006119N), and the Agentschap Innoveren en Ondernemen (through ICON project Supersized 4.0, grant no. HBC.2019.0135).

**Review statement.** This paper was edited by Weifei Hu and reviewed by Yolanda Vidal and one anonymous referee.

## References

Bangalore, P. and Tjernberg, L. B.: Self evolving neural network based algorithm for fault prognosis in wind turbines: A case study, in: 2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), 7–10 July 2014,



- Durham, 1–6, <https://doi.org/10.1109/PMAPS.2014.6960603>, 2014.
- Bangalore, P. and Tjernberg, L. B.: An Artificial Neural Network Approach for Early Fault Detection of Gearbox Bearings, *IEEE T. Smart Grid*, 6, 980–987, <https://doi.org/10.1109/TSG.2014.2386305>, 2015.
- Bangalore, P., Letzgus, S., Karlsson, D., and Patriksson, M.: An artificial neural network-based condition monitoring method for wind turbines, with application to the monitoring of the gearbox, *Wind Energy*, 20, 1421–1438, 2017.
- Beretta, M., Cárdenas, J., Koch, C., and Cusidó, J.: Wind Fleet Generator Fault Detection via SCADA Alarms and Autoencoders, *Appl. Sci.-Basel*, 10, 8649, <https://doi.org/10.3390/app10238649>, 2020.
- Beretta, M., Julian, A., Sepulveda, J., Cusidó, J., and Porro, O.: An Ensemble Learning Solution for Predictive Maintenance of Wind Turbines Main Bearing, *Sensors*, 21, 1–20, 2021.
- Bermúdez, K., Ortiz-Holguin, E., Tutivén, C., Vidal, Y., and Benalcázar-Parra, C.: Wind Turbine Main Bearing Failure Prediction using a Hybrid Neural Network, *J. Phys. Conf. Ser.*, 2265, 032090, <https://doi.org/10.1088/1742-6596/2265/3/032090>, 2022.
- Black, I. M., Richmond, M., and Kolios, A.: Condition monitoring systems: a systematic literature review on machine-learning methods improving offshore-wind turbine operational management, *International Journal of Sustainable Energy*, 40, 923–946, 2021.
- Black, I. M., Cevalco, D., and Kolios, A.: Deep Neural Network Hard Parameter Multi-Task Learning for Condition Monitoring of an Offshore Wind Turbine, *J. Phys. Conf. Ser.*, 2265, 032091, <https://doi.org/10.1088/1742-6596/2265/3/032091>, 2022.
- Campoverde, L., Tutivén, C., Vidal, Y., and Benalcázar-Parra, C.: SCADA Data-Driven Wind Turbine Main Bearing Fault Prognosis Based on Principal Component Analysis, *J. Phys. Conf. Ser.*, 2265, 032107, <https://doi.org/10.1088/1742-6596/2265/3/032107>, 2022.
- Castellani, F., Astolfi, D., and Natili, F.: SCADA Data Analysis Methods for Diagnosis of Electrical Faults to Wind Turbine Generators, *Appl. Sci.-Basel*, 11, 1–14, 2021.
- Chen, H., Liu, H., Chu, X., Liu, Q., and Xue, D.: Anomaly detection and critical SCADA parameters identification for wind turbines based on LSTM-AE neural network, *Renew. Energ.*, 172, 829–840, 2021.
- Chesterman, X., Verstraeten, T., Daems, P.-J., Nowé, A., and Helsen, J.: Condition monitoring of wind turbines using machine learning based anomaly detection and statistical techniques for the extraction of “healthy data”, in: *Proceedings of the Annual Conference of the PHM Society*, 2980, <https://doi.org/10.36001/phmconf.2021.v13i1.2980>, 2021.
- Chesterman, X., Verstraeten, T., Daems, P.-J., Sanjines, F. P., Nowé, A., and Helsen, J.: The detection of generator bearing failures on wind turbines using machine learning based anomaly detection, *J. Phys. Conf. Ser.*, 2265, 032066, <https://doi.org/10.1088/1742-6596/2265/3/032066>, 2022.
- Commission, IEC: Part 12-1: Power performance measurements of electricity producing wind turbines, *Wind energy generation systems*, 317, IEC 61400-12-1:2022, 2022.
- Cui, Y., Bangalore, P., and Tjernberg, L. B.: An Anomaly Detection Approach Based on Machine Learning and SCADA Data for Condition Monitoring of Wind Turbines, 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), 24–28 June 2018, Boise, ID, USA, 1–6, 2018.
- Dienst, S. and Beseler, J.: Automatic anomaly detection in offshore wind SCADA data, *Wind Europe summit*, 27–29 September 2016, Hamburg, Germany, 1–7, 2016.
- Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., and Tabona, O.: A survey on missing data in machine learning, *Journal of Big Data*, 8, 1–37, 2021.
- Garlick, W. G., Dixon, R., and Watson, S.: A model-based approach to wind turbine condition monitoring using SCADA data, in: *20th International Conference on Systems Engineering*, edited by: Burnham, K. J. and Haas, O. C. L., 8–10 September 2009, Coventry, UK, 1–8, 2009.
- Greco, A., Sheng, S., Keller, J., and Erdemir, A.: Material wear and fatigue in wind turbine systems, *Wear*, 302, 1583–1591, 2013.
- Helbing, G. and Ritter, M.: Deep Learning for fault detection in wind turbines, *Renew. Sust. Energ. Rev.*, 98, 189–198, 2018.
- Helsen, J.: Review of Research on Condition Monitoring for Improved O&M of Offshore Wind Turbine Drivetrains, *Acoust. Aust.*, 49, 251–258, 2021.
- Jamil, F., Verstraeten, T., Nowé, A., Peeters, C., and Helsen, J.: A deep boosted transfer learning method for wind turbine gearbox fault detection, *Renew. Energ.*, 197, 331–341, 2022.
- Kim, K., Parthasarathy, G., Uluyol, O., and Patel, Y.: Use of SCADA Data for Failure Detection in Wind Turbines, *ASME 5th Int. Conf. Energy Sustain.*, 7–10 August 2011, Washington DC, USA, 1–11, 2011.
- Kusiak, A. and Li, W.: The prediction and diagnosis of wind turbine faults, *Renew. Energ.*, 36, 16–23, 2011.
- Kusiak, A. and Verma, A.: Analyzing bearing faults in wind turbines: A data-mining approach, *Renew. Energ.*, 48, 110–116, 2012.
- Lee, J. and Zhao, F.: *Global Wind Report 2022*, Global Wind Energy Council, Brussels, 2022.
- Li, J., Lei, X., Li, H., and Ran, L.: Normal Behavior Models for the Condition Assessment of Wind Turbine Generator Systems, *Electr. Pow. Compo. Sys.*, 42, 1201–1212, 2014.
- Lima, L. A. M., Blatt, A., and Fujise, J.: Wind Turbine Failure Prediction Using SCADA Data, *J. Phys. Conf. Ser.*, 1618, 022017, <https://doi.org/10.1088/1742-6596/1618/2/022017>, 2020.
- Liu, X., Lu, S., Ren, Y., and Wu, Z.: Wind Turbine Anomaly Detection Based on SCADA Data Mining, *Electronics*, 9, 1–16, 2020.
- Maron, J., Anagnostos, D., Brodbeck, B., and Meyer, A.: Artificial intelligence-based condition monitoring and predictive maintenance framework for wind turbines, *J. Phys. Conf. Ser.*, 2151, 1–9, 2022.
- Marti-Puig, P., Blanco-M., A., Serra-Serra, M., and Solé-Casals, J.: Wind Turbine Prognosis Models Based on SCADA Data and Extreme Learning Machines, *Appl. Sci.-Basel*, 11, 590, 2021.
- Mazidi, P., Bertling, L., and Sanz-Bobi, M. A.: Performance Analysis and Anomaly Detection in Wind Turbines based on Neural Networks and Principal Component Analysis, 1–9, 2017.
- McKinnon, C., Carroll, J., McDonald, A., Koukoura, S., Infield, D., and Soraghan, C.: Comparison of New Anomaly Detection Technique for Wind Turbine Condition Monitoring Using Gearbox SCADA Data, *Energies*, 13, 1–19, 2020.

- Meyer, A.: Early fault detection with multi-target neural networks, CoRR, arXiv [preprint], <https://doi.org/10.48550/arxiv.2106.08957>, 2021.
- Miele, E. S., Bonacina, F., and Corsini, A.: Deep anomaly detection in horizontal axis wind turbines using Graph Convolutional Autoencoders for Multivariate Time series, *Energy and AI*, 8, 100145, <https://doi.org/10.1016/j.egyai.2022.100145>, 2022.
- Page, E.: A test for a change in a parameter occurring at an unknown point, *Biometrika*, 42, 523–527, 1955.
- Peng, D., Liu, C., Desmet, W., and Gryllias, K.: Deep Unsupervised Transfer Learning for Health Status Prediction of a Fleet of Wind Turbines with Unbalanced Data, *Annual Conference of the PHM Society*, 29 November–2 December 2021, 13, 1–11, <https://doi.org/10.36001/phmconf.2021.v13i1.3069>, 2021.
- Peter, R., Zappalá, D., Schamboeck, V., and Watson, S. J.: Wind turbine generator prognostics using field SCADA data, *J. Phys. Conf. Ser.*, 2265, 032111, <https://doi.org/10.1088/1742-6596/2265/3/032111>, 2022.
- Pfaffel, S., Faulstich, S., and Rohrig, K.: Performance and Reliability of Wind Turbines: A review, *Energies*, 10, 1904, 2017.
- Renström, N., Bangalore, P., and Highcock, E.: System-wide anomaly detection in wind turbines using deep autoencoders, *Renew. Energ.*, 157, 647–659, 2020.
- Schlechtingen, M. and Santos, I. F.: Condition Monitoring With Wind Turbine SCADA Data Using Neuro-Fuzzy Normal Behavior Models, *Turbo Expo: Power for Land, Sea, and Air*, 6, 717–726, 2012.
- Schlechtingen, M. and Santos, I. F.: Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 2: Application examples, *Appl. Soft Comput.*, 14, 447–460, 2014.
- Schlechtingen, M., Santos, I. F., and Achiche, S.: Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description, *Appl. Soft Comput.*, 13, 259–270, 2013.
- Sun, P., Li, J., Wang, C., and Lei, X.: A generalized model for wind turbine anomaly identification based on SCADA data, *Appl. Energ.*, 168, 550–567, 2016.
- Takanashi, M., Ichi Sato, S., Indo, K., Nishihara, N., Hayashi, H., and Suzuki, T.: Anomaly Prediction for Wind Turbines Using an Autoencoder with Vibration Data Supported by Power-Curve Filtering, *IEICE T. Inf. Syst.*, E105.D, 732–735, 2022.
- Tautz-Weinert, J. and Watson, S. J.: Comparison of different modelling approaches of drive train temperature for the purposes of wind turbine failure detection, *J. Phys. Conf. Ser.*, 753, 1–11, 2016.
- Tautz-Weinert, J. and Watson, S. J.: Using SCADA data for wind turbine condition monitoring – a review, *IET Renew. Power Gen.*, 11, 382–394, 2017.
- Tazi, N., Chatelet, E., and Bouzidi, Y.: Wear Analysis of Wind Turbine Bearings, *International Journal of Renewable Energy Research*, 7, 2120–2129, 2017.
- Turnbull, A., Carroll, J., and McDonald, A.: Combining SCADA and vibration data into a single anomaly detection model to predict wind turbine component failure, *Wind Energy*, 24, 197–211, 2021.
- Udo, W. and Yar, M.: Data-Driven Predictive Maintenance of Wind Turbine Based on SCADA Data, *IEEE Access*, 9, 162370–162388, 2021.
- van Buren, S. and Groothuis-Oudshoorn, K.: mice: multivariate imputation by chained equations in R, *J. Stat. Softw.*, 45, 1–67, 2011.
- Verma, A., Zappalá, D., Sheng, S., and Watson, S. J.: Wind turbine gearbox fault prognosis using high-frequency SCADA data, *J. Phys. Conf. Ser.*, 2265, 032067, <https://doi.org/10.1088/1742-6596/2265/3/032067>, 2022.
- Verstraeten, T., Nowé, A., Keller, J., Guo, Y., Sheng, S., and Helsen, J.: Fleetwide data-enabled reliability improvement of wind turbines, *Renew. Sust. Energ. Rev.*, 109, 428–437, 2019.
- Xiang, L., Yang, X., Hu, A., Su, H., and Wang, P.: Condition monitoring and anomaly detection of wind turbine based on cascaded and bidirectional deep learning networks, *Appl. Energ.*, 305, 117925, <https://doi.org/10.1016/j.apenergy.2021.117925>, 2022.
- Yang, W., Tavner, P. J., Crabtree, C. J., Feng, Y., and Qiu, Y.: Wind turbine condition monitoring: technical and commercial challenges, *Wind Energy*, 17, 673–693, 2014.
- Zaher, A., McArthur, S., Infield, D., and Patel, Y.: Online wind turbine fault detection through automated SCADA data analysis, *Wind Energy*, 12, 574–593, 2009.
- Zraggen, J., Ulmer, M., Jarlskog, E., Pizza, G., and Huber, L. G.: Transfer Learning Approaches for Wind Turbine Fault Detection using Deep Learning, *PHM Society European Conference*, 6, 1–12, 2021.
- Zhao, H., Liu, H., Hu, W., and Yan, X.: Anomaly detection and fault analysis of wind turbine components based on deep learning network, *Renew. Energ.*, 127, 825–834, 2018.
- Zhongshan, H., Ling, T., Dong, X., Sichao, L., and Yaozhong, W.: Condition monitoring of wind turbine based on copula function and autoregressive neural network, *MATEC Web of Conferences*, 198, 1–5, 2018.
- Zou, H. and Hastie, T.: Regularization and variable selection via the elastic net, *J. Roy. Stat. Soc. B*, 67, 301–320, 2005.