



## Review

## Machine learning methods for wind turbine condition monitoring: A review

Adrian Stetco<sup>a,\*</sup>, Fateme Dinmohammadi<sup>b</sup>, Xingyu Zhao<sup>b</sup>, Valentin Robu<sup>b</sup>,  
David Flynn<sup>b</sup>, Mike Barnes<sup>c</sup>, John Keane<sup>a</sup>, Goran Nenadic<sup>a</sup>

<sup>a</sup> School of Computer Science, University of Manchester, UK

<sup>b</sup> School of Engineering and Physical Sciences, Heriot-Watt University, UK

<sup>c</sup> School of Electrical and Electronic Engineering, University of Manchester, UK

## ARTICLE INFO

## Article history:

Received 6 July 2018

Received in revised form

7 October 2018

Accepted 8 October 2018

Available online 9 October 2018

## Keywords:

Renewable energy

Wind farms

Condition monitoring

Machine learning

Prognostic maintenance

## ABSTRACT

This paper reviews the recent literature on machine learning (ML) models that have been used for condition monitoring in wind turbines (e.g. blade fault detection or generator temperature monitoring). We classify these models by typical ML steps, including data sources, feature selection and extraction, model selection (classification, regression), validation and decision-making. Our findings show that most models use SCADA or simulated data, with almost two-thirds of methods using classification and the rest relying on regression. Neural networks, support vector machines and decision trees are most commonly used. We conclude with a discussion of the main areas for future work in this domain.

© 2018 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## Contents

1. Introduction .....	620
2. Condition monitoring of wind turbines .....	621
3. Machine learning (ML) overview .....	621
4. Machine learning for condition monitoring .....	624
4.1. Data .....	624
4.2. Feature selection and extraction .....	624
4.3. Models: regression-based .....	626
4.4. Models: Classification-based .....	628
4.5. Models: validation .....	630
4.6. Using ML models in CM decision support systems .....	631
5. Conclusions .....	631
Acknowledgements .....	632
References .....	633

## 1. Introduction

Prompted in part by public investments [1] and climate change awareness, rapid advances in the technology used for renewable energy collection have resulted in an increasing proportion of such sources relative to conventional ones (e.g. fossil fuels). Specifically,

\* Corresponding author.

E-mail address: [mihai.stetco@manchester.ac.uk](mailto:mihai.stetco@manchester.ac.uk) (A. Stetco).

wind energy is captured via the use of turbines that can be situated onshore (on land) or offshore (at sea). Wind farms are increasingly being built offshore for several reasons, including wind condition being stronger and more stable at sea, larger units being more easily transported and deployed, less visual disturbance and potential conflicts of interests being minimized etc. [2]. However, the cost of maintaining wind turbines in offshore locations is significant: ensuring that wind turbines perform at their optimal level over their lifetime (usually 20–25 years) costs around 25% of the offshore installation [3].

Condition monitoring (CM) involves observing the components of a wind turbine to identify changes in operation that can be indicative of a developing fault. It is clear that predicting faults before they occur, through robust CM, should lead to significant reduction in Operation and Maintenance (O&M) costs [4]. CM approaches have relied on analyses of specific measurements and aspects of the operation (e.g. vibration analysis, strain measurement, thermography and acoustic emissions). Recent developments in sensors and signal processing systems, big data management, machine learning (ML) and improvements in computational capabilities have opened-up opportunities for integrated and in-depth CM analytics, where different types of data can facilitate informed, reliable, cost-effective and robust decision-making in CM.

This paper reviews recent ML-based approaches (2011 onwards) to CM of wind turbines. To conduct the review, articles were retrieved from Google Scholar using the search terms “wind turbine condition monitoring regression” and “wind turbine condition monitoring classification” and filtered by year (>2011), access, citations and relevancy; selected papers from pre-2011 are introduced for their historical importance.

We screened 144 papers for task relevance (fault diagnosis/prognosis) and related data, ML pipeline (feature selection and extraction, model) and decision-making. For each category of model identified, we discuss related challenges, potentials and drawbacks. Appendix A presents an overview of feature extraction and selection methods; Appendix B summarizes the main trends identified relating to datasets, tasks, methods and evaluation.

The paper is structured as follows: Section 2 overviews CM of wind turbine; Section 3 introduces typical steps in ML; Section 4 presents specific CM approaches; Section 5 concludes and discusses future work.

## 2. Condition monitoring of wind turbines

CM of wind turbine is an integral part of O&M, where operations include the management, monitoring and high-level onshore control of the wind farm site, while maintenance covers interventions required to upkeep the installation. Maintenance can be reactive, preventive or predictive [5]: *reactive* (or corrective, run-to-failure) is the most expensive type and does not utilize CM with components being replaced when defects occur or accumulate; under *preventive* (scheduled) maintenance, components are replaced at the next intervention, hopefully before a related fault occurs; a *predictive* maintenance strategy based on CM can inform maintenance about components that are likely to fail and have them replaced in due time.

CM can be viewed along several aspects. Firstly, CM can be applied at different levels of granularity: at the most granular level, we can monitor the condition of wind turbine sub-components (e.g. drivetrain); at the most coarse-grained uppermost level we can consider the whole wind farm. The signals provided by different models can be combined to provide higher-level warnings for the whole turbine.

Secondly, the ways in which monitoring is performed can have a

physical impact on the component being monitored. The literature identifies two main types of monitoring:

- **Intrusive Monitoring:** involves Vibration Analysis [6], oil debris monitoring, shock pulse methods etc. Such methods impose a penalty (wear) on the component being monitored.
- **Non-intrusive Monitoring:** involves ultrasonic testing techniques, visual inspection, acoustic emission, thermography, performance monitoring using power signal analysis etc. [7].

Thirdly, CM can be used for fault detection in real-time or in the future, so we distinguish between:

- **CM for diagnosis (fault detection),** where we identify a fault when it happens. Ensuring that the CM can identify the presence of failure should be a prerequisite for building a ML model for prognosis.
- **CM for prognosis (fault prediction),** where the underlying model finds patterns in the signal data that are predictive of failures in the future.

When deciding which components to monitor, it is important to consider the failure rates and downtimes per failure of different sub-components. Prioritized considerations are given to components that are more likely to fail or can lead to long downtimes, as they may incur the greatest potential impact. Pfaffel et al. [8] aggregated data from seven surveys to identify annual failure rates of different sub-systems of a wind turbine together with mean downtime per day. They concluded that some components (such as the rotor (especially pitch system), transmission and power system) tend to have a higher failure rate than others. Carroll et al. [9] analyzed over 300 offshore wind turbines and found that failure rate per offshore turbine per year is about 10, with around 80% requiring minor repairs (<1 k Euro), 17.5% major repairs (1–10 k Euro) and 2.5% major replacements (>10 k Euro). They identified the pitch/hydraulic, generator and other subsystems (door/hatch issues, covers, bolts, lightning) as contributing the most to failure rates. Generators and converters tend to have a higher level of failure rates in offshore wind turbines than onshore ones. Typical failures in gearboxes include slip ring, grease pipe, rotor issues [9] failures in planetary gears and bearings, intermediate and high-speed shaft bearings and lubrication system malfunction [10].

Crabtree et al. [11] found that although there is a wide variety of commercial CM system in use, there is no consensus regarding the future direction of research. They reported that current commercial wind turbine CM relies heavily on established methodologies borrowed from conventional rotating machine industries. Common ways of performing CM include acoustic measurement-based methods, electrical effects monitoring, power quality and temperature monitoring, oil debris monitoring, vibration analysis [12] [13], physics based data analytics [14] etc.

## 3. Machine learning (ML) overview

ML is the process of building an inductive model that learns from a limited amount of data without specialist intervention. This learning implies finding an underlying set of structures (or patterns) that are useful to understand relationships in data that might not be exactly similar to that on which learning occurred. In the taxonomy of ML models (Fig. 1), *supervised learning* predicts an output variable using labeled input data, while *unsupervised learning* draws inferences from data without labeled inputs (such as done by clustering algorithms, recommender systems etc.). For supervised learning we distinguish between models that predict a numeric variable (regression) or a categorical variable (classifiers).

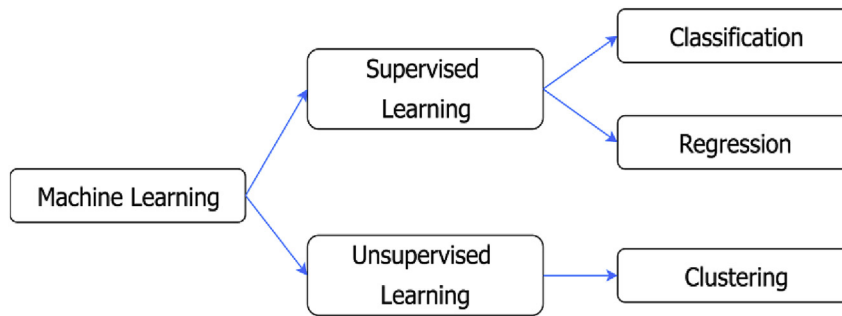


Fig. 1. Taxonomy of ML models.

Learning in models translates into fitting a model's parameters to a specific dataset, iteratively updating them with several passes through the data until a specific predefined function is minimized.

The ML process can be represented as a series of steps:

- **Data acquisition and preprocessing:** where possibly different data sets and modalities are integrated, cleaned of outliers, etc.
- **Feature selection and extraction:** important signals and characteristics are identified and extracted from the data.
- **Model selection:** an appropriate model is chosen, taking into consideration the task to be solved.
- **Validation:** a performance measure is used that is specific to the task, including accuracy (classification) and mean absolute error (regression), evaluated on a validation set of data.

Fig. 2 illustrates a typical detailed workflow for two common ML tasks:

- **Classification/prediction:** Important steps include data pre-processing (dealing with missing data, outliers, etc.), classes equalization (ensuring the classes to be predicted have equal distribution so that the model is not biased [15]), filter/wrapper feature selection and extraction (for keeping only relevant features), classification model fitting (where the model's parameters are estimated), cross-validation (where the model's generalizability is tested).

The solution can be cast as a prediction problem if the features are fed into the model with labels at future times (e.g.  $t+1$ ), which extends the solution from diagnostic to prognostic.

- **Regression-based anomaly detection:** Here the task is to identify how signals and features are related to outputs in different components. This relationship is captured by fitting regression models when the system is in a healthy state. When new data comes in, it is compared to what the model predicts for a healthy state and if a deviation is found for several consecutive time intervals, an alarm is raised. Behaviour of a component (from low to high granularities) can be captured through regressions of different complexities (from a simple linear model to a complex non-linear one).

The ML model selection step is particularly significant as it is the core functionality that learns from past data and generalizes into the future. Such models have been used for different tasks, including classification, regression, anomaly detection, synthesis and sampling, imputation of missing values, denoising, density estimation and many others [16].

Several different models have been suggested for learning from data. Support vector machines (SVMs) and neural networks (NNs)

are two common models that have been used in ML for diagnostics and prognostics. Connectionist models, such as NNs, consist of simple replicated computing units called neurons which can communicate with and pass information to each other through links between the synapses. In the beginning, these models were basic and could only solve linear classification problems (e.g. see the perceptron [17]). Solving non-linearly separable cases requires more complex architectures, typically made of several layers of neurons (see Fig. 3). Such architectures are able to approximate any classification function and can be understood as universal approximators [18]. Feed-forward multi-layered is a type of NN architecture that has no cycles between neurons and where information propagates in one direction, making it simple to model and implement (as opposed to recurrent neural networks (RNNs)). While the main principles behind NNs have been around for some time (e.g. the backpropagation algorithm for error contribution of each neuron [19]), availability of larger data sets, better initialization algorithms, larger sets of neuron activation functions and more powerful machines have made it possible to train NNs composed of hundreds of stacked hidden layers. This approach, termed “deep learning”, has shown disruptive capabilities in many domains from image recognition to speech translation and has started to penetrate the wind energy industry (e.g. general rotary machineries [20], [21]). Although the training time of an NN can be potentially long, when it comes to actual classification or regression, the application of models is comparatively very fast. However, results obtained using NNs are highly dependent on the choice of architecture used, weight initialization, activation function, optimization procedure etc., and the process can require much effort and expertise. Moreover, if transparency, explanation or audit of a model is important, NNs are not well suited.

NNs have been used widely in the wind energy sector for forecasting (e.g. wind speed forecasting), control (e.g. wind turbine power control), identification and evaluation (e.g. fault diagnosis) [22,23].

SVMs are often used in fault detection and CM [22], [23], and for complex data sets in general [25]. They perform linear/non-linear classification or regression by finding decision boundary hyperplanes that best separate classes of instances, i.e. by leaving the widest possible margin to the instances closest to the margin (see Fig. 4). It is not always possible to clearly separate instances (e.g. in the presence of outliers) and a parameter  $C$  (slack variable) allows control of margin violations. For non-linear problems, adding polynomial features created from existing ones can make the problem linearly separable in a higher dimensional space. Implementations of SVMs (e.g. SCIKIT-Learn [26]) have several ways to transform a problem into a linearly separable one with the use of kernels (polynomial, RBF, etc.).

As with other classes of learning algorithms, NNs and SVMs can be used as classifiers (where a nominal variable is predicted) or

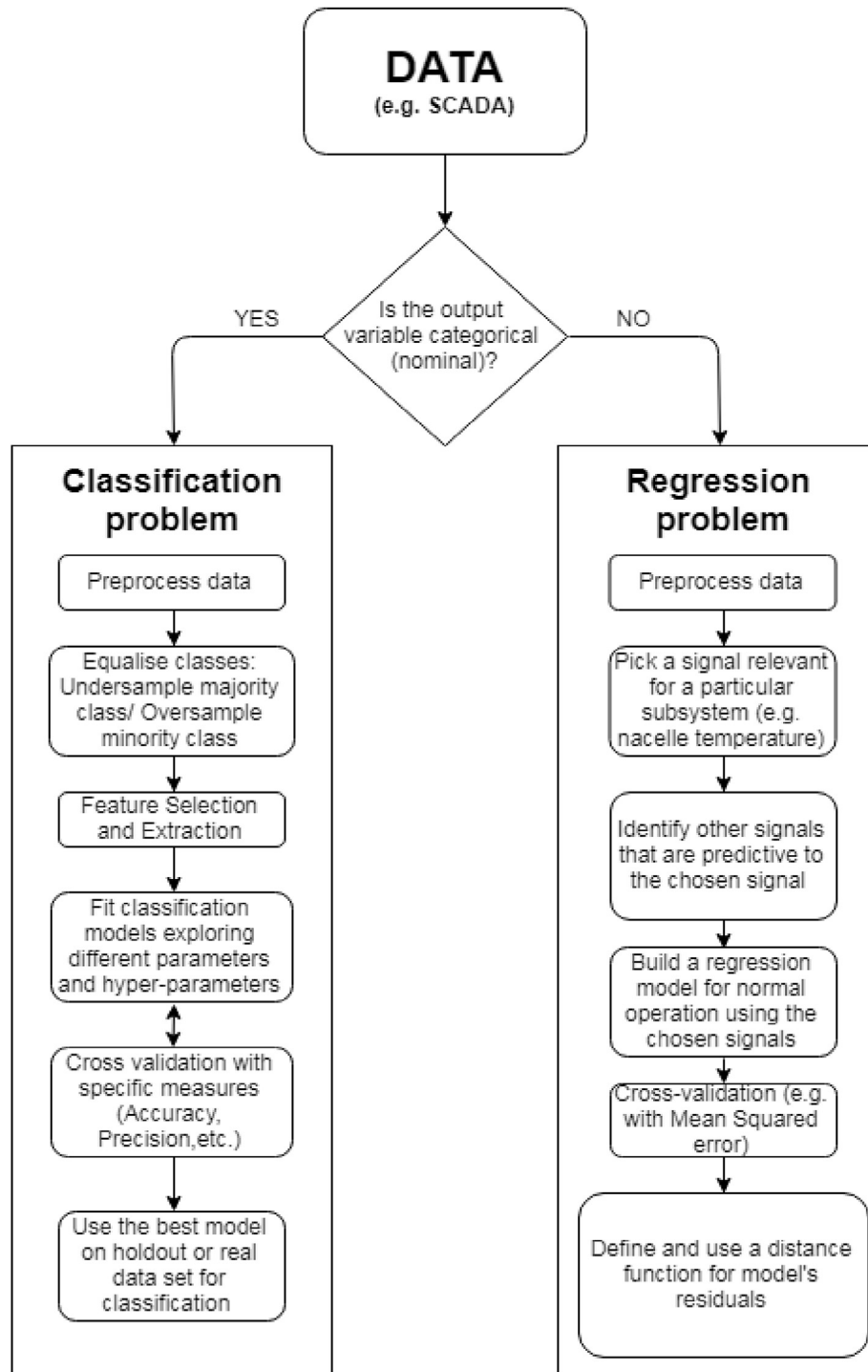


Fig. 2. Two main approaches to learning from data.

regressors (where a numeric variable is predicted). Compared to NNs, not only do SVMs always find the global minimum when performing optimization on the training set, but they also have an intuitive graphical interpretation [27]. SVMs can be slow and training on large dataset remains a challenge [27]; time complexity is usually between  $O(m^2n)$  and  $O(m^3n)$ , where  $m$  is the number of instances and  $n$  is the number of features [25]. Having a multitude of kernels to choose from is another complication and without any assumptions about the underlying data distribution, the search for

optimum kernel (and kernel parameter) can introduce a significant time burden.

Validity of ML models can be estimated through several specific measures in combination with an out-of-sample technique such as  $n$ -fold cross validation which assess how well the results of the model will generalize beyond the training data. We discuss several specific measures such as MAE (mean absolute error), MAPE (absolute percentage error) and its variants for regression-based models. Classification models are typically evaluated using

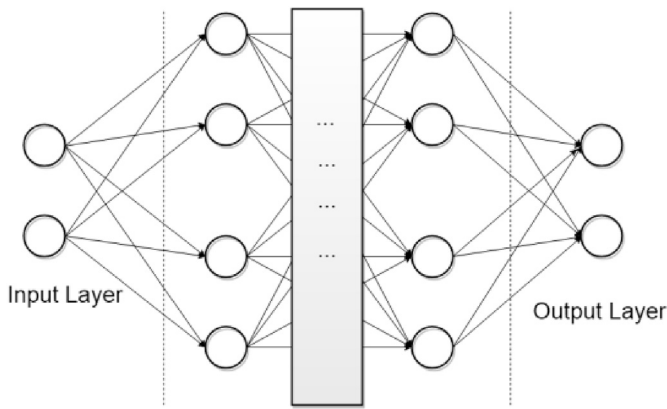


Fig. 3. Feed-forward multi-layered NN ([24]).

accuracy, sensitivity, specificity and F1-measure. These metrics are discussed in more detail in Section 4.5.

#### 4. Machine learning for condition monitoring

This Section investigates recent regression and classification-based models proposed for CM of different components in a wind turbine. As indicated above, 144 papers that used ML for wind turbine CM were screened and an aggregated summary of the methods was developed. A variety of tasks have been considered, including identification of blade faults, generator brush failure prediction, transmission system fault diagnosis, lubricant pressure monitoring, etc. In presenting the findings, we follow the typical ML steps as presented in Section 3.

##### 4.1. Data

A wind farm CM system may rely on several types of datasets. Most CM models discussed in the literature are based on operational and event datasets, such as the ones provided by SCADA (Supervisory Control And Data Acquisition). SCADA systems have been built into turbines to control electricity generation [29], by providing time-series signals in regular intervals. This type of system collects basic information with the use of sensors placed on key wind turbine components (e.g. bearing vibration, temperature, phase currents, wind speed, etc.) [30] [31]. There is no common set of available SCADA signals nor is there a generally accepted taxonomy of signals, with different systems having different names [32]. As well as SCADA time series signals, different other types of

data might be collected such as drone images or event data in free-text form.

Data typically available in a wind turbine pose the “big data” challenges:

1. *Volume*: a typical wind farm with 20–30 sensors for each wind turbine would generate between 60 and 100 SCADA signals which, when sampled every second, would produce about 0.2 GB of raw data per turbine [33].
2. *Velocity*: the frequency at which data is produced and transmitted, with new wireless and acoustic sensors [34].
3. *Variety*: CM systems have to integrate sensor data with images, video (e.g. captured by drones), and free-text action reports [35], etc.
4. *Veracity*: ideally, data should be free of missing or impossible values and inconsistencies; if not, automatic or semi-automatic data cleaning (scrubbing) procedures are typically needed. This need increases with the number of data sources of data, especially if heterogeneous [36].

Given such big data, CM needs to rely on efficient, scalable and fault tolerant data management systems. For example, Canizo et al. [37] use a technology stack consisting of HDFS (Hadoop Data File System [38], a distributed, fault tolerant file system), Apache Kafka [39] (a stream processing framework), Spark [40], [41] (a cluster computing framework suited for big data ML) and Apache Mesos [42] and Zookeeper (for cluster management) that can be deployed to cloud computing environments. The Map/Reduce processing framework is often used in combination with Hadoop infrastructure for parallel data processing. Batch distributed parallel applications with big data requires movement of data in and out of disk [43]. ML algorithms, which typically make several passes over the data in estimating parameters and decreasing a predefined cost function, can be optimized by using Spark which holds the dataset in memory. Hence, for wind farm CM that needs to perform in real-time and update/tune the models at regular intervals, Spark is quite appropriate.

##### 4.2. Feature selection and extraction

One of the initial tasks before building an ML model is outlier identification (i.e. extreme or likely impossible values, or measurement errors). Careful consideration should be given to the relationships between variables when implementing outlier filtering methods; for example, generator temperature spikes could arise due to ambient temperature spikes and not due to an internal malfunction of the generator, and a simple outlier selection might

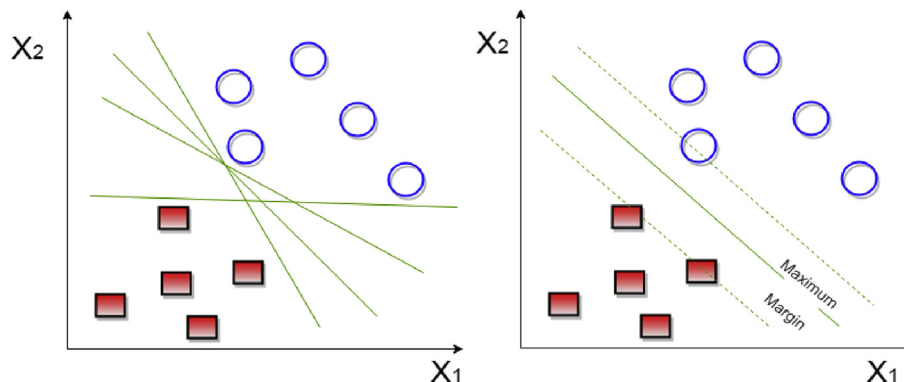


Fig. 4. A multitude of linear decision boundaries separate the two classes (left). SVMs find support vectors which maximize the distance between decision hyperplanes and closest data points (right) [28].



remove these data points. Marti-Puig et al. [44] investigated the effects of removing outliers in fault diagnostics of wind turbine using common filtering methods such as quantile filter, extreme studentized deviate test and the Hampel identifier. They found that the outlier filtering methods can decrease the error on the training data set but increase the error in the test data set because many of the outliers discovered automatically were actual failure states of the turbine. Therefore, they recommend expert input to predefine variable's absolute and relative vibration ranges. Once the data has been cleaned, feature selection and extraction are the next critical steps in building an ML model.

**Feature selection** is the process of selecting variables, here time series signals, that relate to the outcome that we wish to study, understand or predict. This can be achieved automatically or semi-automatically under the guidance of an expert. For example, choosing to keep generator vibration sensor data and discard acoustic sensors in the tower when studying generator faults is a form of feature selection. Feature selection can be achieved automatically using:

- **Wrapper methods** for feature selection view ML algorithms as black boxes and feed them with different subsets of features. The prediction performance of a given model is then used to assess the relative usefulness of subsets of variables [45], [46]. For wrappers, the user needs to specify the algorithm to use, the strategy for selecting features and the performance criteria for the model. Exhaustive searches are NP-hard [47] and sub-optimal strategies such as forward (start small and add features as long as they increase performance) or backward selection (start with all features and remove as long as it increases performance) are often used.
- **Embedded methods** perform feature selection as part of the training of the ML model (consider the relative importance of nodes in a decision tree) and can be very successful when used in combination with filters [48].
- **Filter methods** are independent from the model itself [49]. They perform a significance test between each feature/signal and an outcome (e.g. through correlation) and then rank them. A user selects the top  $k$  features, where  $k$  can further be selected based on the performance it offers with a specific model.

**Feature extraction** is used to compress high-dimensional time series (such as sensor signals) by keeping their main characteristics intact while discarding noise and removing correlations [50,51]. This should speed up model training and produce better outcomes than when applied to the original, raw data. For time series, some of the most common techniques for feature extraction involve computing:

- **Statistics:** This class of features is the simplest to compute and sometimes the most efficient; these include mean, standard deviation, maximum, minimum, skewness, kurtosis, peak-to-peak, crest factor, wave factor, impulse factor, margin factor, root mean square, etc.; for an example see Ref. [52];
- **Parameters of fitted time series models:** Coefficients of fitted ARIMA models [53], autocorrelation coefficients, of fitted stochastic processes (e.g. Hidden Markov [54]), etc.;
- **Time-Frequency domain properties:** These techniques transform time domain signals into frequency domain; examples include the largest coefficients of the Discrete Fourier/Wavelet transform [51,55,56], Morlet Wavelet feature extraction [57] and Wigner-Ville distribution [58], SVD [59], cluster-based Wavelet feature extraction [60], Multi-Wavelet feature extraction from vibration data [61], energy tracking [62] and with S-transform [63], instantaneous power spectrum [64], etc. For a

comprehensive survey on Wavelet transform applications to fault diagnosis in rotary machines, see Ref. [65]. Empirical mode decomposition (EMD) [66,84] and its derivatives (e.g. Ensemble-EMD [67]) has received interest in fault-diagnosis research [68–71].

Time-frequency strategies have been the main feature extraction approach in wind turbine CM research. Compared to the traditional Fourier transform which decomposes a signal in the time domain to its constituent frequency components, the Wavelet transform [72,73] can provide both time (time at which the frequency changes) and frequency localization (close frequencies can be separated). The principle behind the Wavelet transform is the use of a “mother” Wavelet function (e.g. Morlet, Haar, Daubechies) to transform a signal from the time-domain to the time-frequency (scale) domain. Compact daughter functions, created using dilation and translation of the chosen mother function, are convoluted with the signal, resulting in Wavelet coefficients that express the amount of similarity between the two.

It should be noted that methods that work well on stationary, lab-generated signals may not transfer to real-world conditions. For example, frequency-domain feature extraction using the Fourier transform is unsuitable when the underlying signal is non-stationary, i.e. when the signal does not have the same mean/variation over the entire time domain space [74]. For example, non-stationarity can express itself in vibration time series produced by rotating machinery (such as the gearbox and generator of a wind turbine) and their handling might necessitate special processing [75]. The Short Time Fourier transform (STFT) segments the series into equal windows that are assumed to be stationary. Making the segments short provides good time resolution but poor frequency resolution; conversely, making segments larger provides better frequency resolution with poor time resolution. The Wavelet transform trades-off between time and frequency localization: high frequency spectra have good time/poor frequency resolution, while low frequency spectra have the opposite.

Fig. 5 presents an example of a non-stationary time series, showing hourly total wind power production in the UK for 2016. We note the heteroskedastic nature with higher variance of the signal at both ends and lower in the middle (see Fig. 6).

The computational cost of a transform for a fast implementation of Fourier transform is  $O(N \log N)$  (STFT is based on Fourier transforms) and for a Wavelet transform is  $O(N)$ . Given this ability to work on non-stationary, non-linear signals, Wavelet transforms is considered superior to Fourier and STFT transforms [75] [77]. Examples of application of Wavelet transform in CM include [78–81]. Empirical Mode Decomposition (EMD) is another important time-frequency method used for signal decomposition that finds Intrinsic Mode Functions (IMFs) of different frequencies that sum to obtain the original signal. This gives another time series that can be further preprocessed using methods specified above to extract relevant features. The EMD algorithm, although suited for non-stationary, non-linear time series such as in wind farms and other domains [82,83], lacks the theoretical basis of Fourier transform or Wavelet decomposition.

A potential problem of the algorithm is mode-mixing, where IMFs have different frequencies that coexist in the same signal or the same frequencies scattered in different IMFs. Fig. 7 shows an example of the problem with the 1st, 5th and 10th IMF of the hourly total wind series in Fig. 5. Ensemble EMD (EEMD), an extension of EMD, address this problem by adding finite noise in the decomposition process. Applications of EMD-based techniques in fault diagnosis of rotating machines include [82,85–88]. For a more detailed comparison between time-frequency analysis techniques in terms of speed, resolution, theoretical framework, etc. see

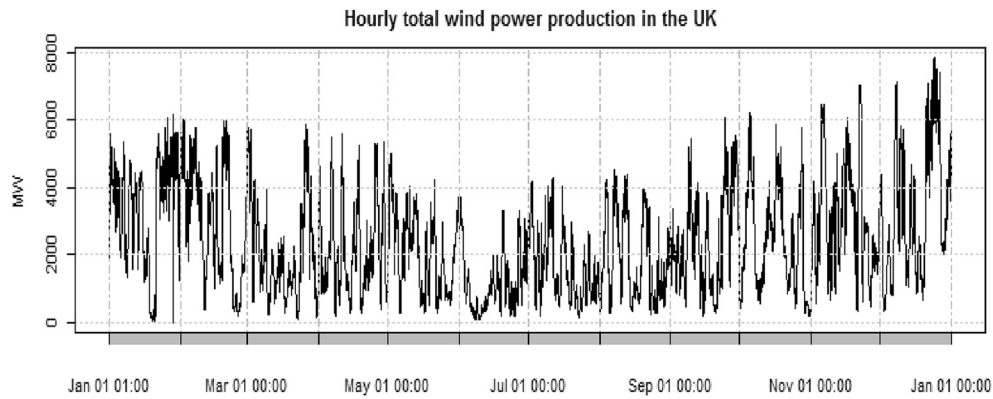


Fig. 5. Hourly total wind power production in the UK [76].

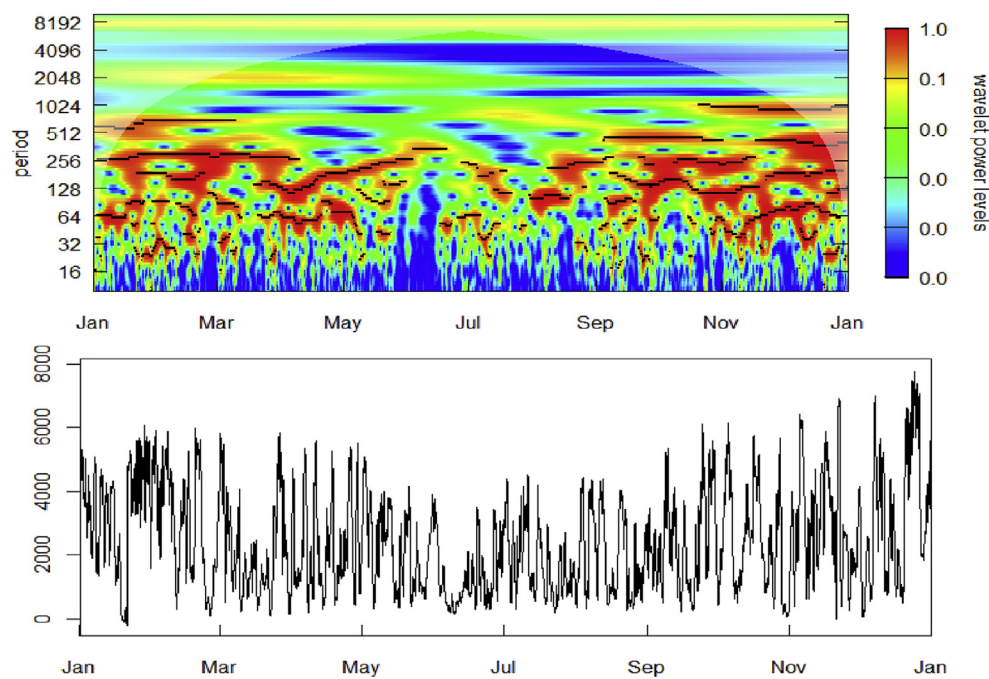


Fig. 6. Wavelet Spectrum and original signal reconstruction using significant coefficients of the Wavelet transform. The upper part shows the Wavelet transform power spectrum of the time series in Fig. 5. Given only a few coefficients that are found relevant (in red, marked with a black ridge) the original signal can be approximated (bottom figure), filtered of noise, etc. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Refs. [77,89,90].

Finally, it should be noted that feature extraction as a process may not finish using the three discussed approaches (statistics, parameters of fitted models or time-frequency properties). For a specific task we may need to further reduce the extracted features or combine them [91]. An interesting approach to drivetrains includes use of auto-encoders [92]. These are potentially deep NNs, composed of an encoder which converts the input to an internal representation (in low dimensions) and a decoder that converts internal representation to outputs [25]. It is more general than classical Principal Component Analysis (PCA), although it can perform similarly when the activation functions are linear, and the cost function is MSE (mean squared error). A comparison to many dimensionality-reducing techniques found that non-linear dimensionality techniques are often incapable of outperforming traditional linear techniques such as PCA [93].

#### 4.3. Models: regression-based

An important approach to CM in wind farms is modelling normal behaviour of different components or subcomponents, when they are assumed to be in a healthy state (also called 'steady state modelling'). Based on the inputs (independent variables such as wind speed), regression models are built which predict the numeric output (dependent variables such as power) when the component to be modeled is assumed to be performing at its optimum. What constitutes a healthy (normal, optimum) component in many cases is a function of time. The failure rate of electrical and mechanical components is distributed as shown in Fig. 8 with failing failure rates in early life, followed by a long period of healthy life and then a wear-out phase with rising failure rates as damage accumulates with operational age [94]. Ideally, normal behaviour data about the component should be recorded during the period when likelihood of failures is low.

Regression models of normal behaviour can be constructed at

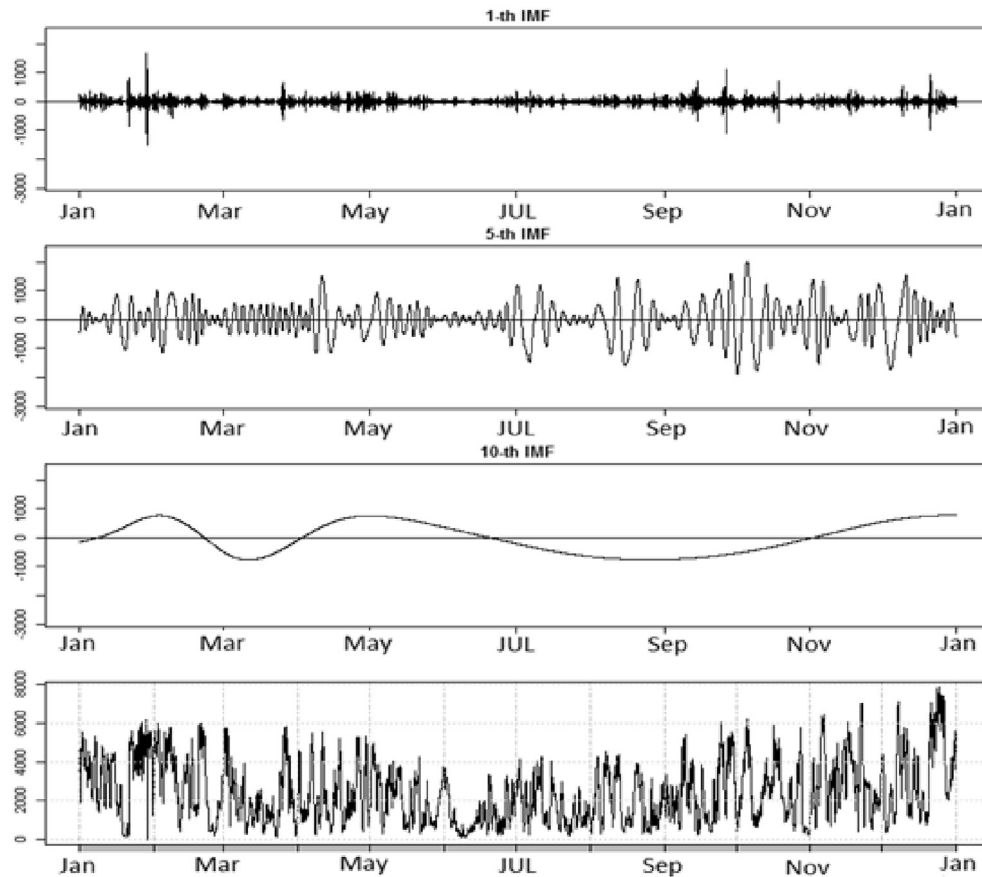


Fig. 7. IMF 1, 5, 10 of the Empirical Mode Decomposition (EMD) and original signal reconstruction (bottom).

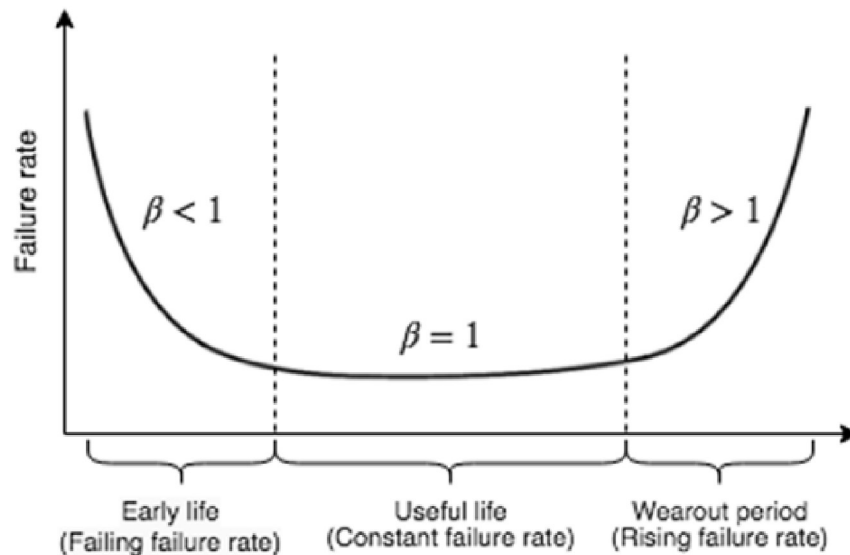


Fig. 8. Failure rate as a function of time [94].  $\beta$  represents the shape (Weibull) parameter.

various levels of complexity. For example, modelling the power curve, which specifies the relationship between wind speed and the power generated by the wind turbine, is the highest conceptual level (the entire wind turbine is assumed to be a black box). Power curves made available by manufacturers are specific to the location where the turbines were tested, which means they were subject to

particular meteorological conditions which may be different to where they will be installed. By independently modelling these power curves empirically, we can better assess and predict the wind energy potential at specific site, make better wind turbine choices, construct monitoring and troubleshooting, and predictive control and optimization applications. Empirically, points will appear



scattered around this curve, with points on the left representing instances where the wind turbine over-performed (generated more power for a given speed) and points on the right representing instances where the turbine under-performed (generated less power) [95]. There are many reasons for outliers around the power curve, such as failure of the control mechanism to adapt to the malfunction of the sensors, pitch control malfunctions, blade pitch angle errors, blade damage, control program problems, incorrect controller settings, blades affected by dirt, bugs, and ice, etc. [96]. Power curves have been modeled in two ways [97]:

1. **Parametric modelling techniques:** linearized segmented model, polynomial curve, maximum principle method, dynamical power curve, probabilistic model, ideal power curve, 4-parameter logistic function, etc. [96,98–100].
2. **Non-parametric modelling techniques:** copula power curve, cubic spline interpolation, NNs, fuzzy methods, ML algorithms [95,96,98,101–103].

Parametric models can be described by a finite set of parameters in a parametric vector. In contrast, non-parametric models are defined by parametric vectors that are unbounded in length [98]. For example, as NNs are non-parametric models with many neurons, increasing the number of connections will typically translate into an increasing number of parameters. We note the distinction between parameters (which change during the training process) and hyperparameters which are fixed before the training commences (e.g. the number of hidden layers which are specified in advance).

When modelling power curves, wind speed may not be the only dependent variable used. For example, Schlechtingen et al. [104] compared two classes of models: one using only wind speed as the dependent variable and one also using wind direction and ambient temperature. For each class, they trained and compared four types of models: CCFL (cluster center fuzzy logic), NNs, k-NN (k-nearest neighbor) and ANFIS (Adaptive Neuro-Fuzzy Inference System). ANFIS is a type of ML model which combines NNs with fuzzy theory. An extension of the theory of sets, fuzzy functions define memberships of objects to sets. Schlechtingen et al. have shown that by adding wind direction and ambient temperature, the models fit the data better (with the errors having a lower variance), with the ANFIS model achieving the best performance (detecting anomalies five days in advance). Once a suitable regression model is chosen, the distribution of the errors can be modeled for defining what constitutes outliers. A power curve model alone cannot pinpoint why an outlier occurred, but can focus attention on a possible faulty turbine. Deeper, more granular regression models are needed for subcomponents if specific faults are to be discovered. By using maximum likelihood, error distribution parameters can be estimated in order to define what constitutes outliers [105]. For example, if the errors are normally distributed, points two or more standard deviations away from the mean can be considered outliers, with the choice of threshold being flexibly defined. This may not be the case in general and other outlier detection techniques can be investigated [106].

The use of unsupervised methods for wind turbine CM has been relatively less explored. For example, Lapira et al. [107] compared a regression model based on feed forward NNs (FFNNs) with two unsupervised methods (Self Organizing Maps (SOMs) and Gaussian Mixture Models (GMMs)) trained on an operational large-scale on-shore wind turbine. They modeled the power curve and analyzed the system health by using a new approach (confidence health value) based on residuals being greater than a given threshold during a given time segment. They found the GMM model presents a more gradual health change being more suitable in performance

prediction.

Power curve modelling is the highest conceptual level that can be modeled as a regression – though it is possible to model the behaviour of different subcomponents by applying the same principles. The choice of regression model will be informed by a combination of expert knowledge and optimized parameter searches (e.g. grid-based cross validation). Other aspects that may influence the choice are performance of the model (both in training and real-time), transparency of the model, etc.

SVMs with a radial basis kernel have been used to detect faults in blade pitch positions, generator and rotor speeds [108]. The model is learned from the data to detect all sensors, actuators and system faults. Schlechtingen et al. [109] investigated three models (regression, NNs and autoregressive NNs) that learn to approximate the normal bearing temperature using SCADA input signals such as power output, nacelle temperature, generator speed, generator stator temperature, etc. For identifying the lag and the signals which are related to the output signal (bearing temperature), cross-correlation was used. They found that the nonlinear NN approaches outperform the regression models.

Guo et al. [110] built a regression model for generator faults based on the non-linear state estimate technique (NSET), introduced by Singer in Ref. [111]. For each input vector at any time step, the output of the model is a linear combination of historical observations held in a special matrix called the memory matrix. The model for predicting generator temperature is built using five variables (out of 47 SCADA signals): power, ambient temperature, nacelle temperature, generator cooling air and stator winding temperature. The residuals obtained are analyzed using a moving average window. Abnormality is identified either when the residual mean value remains zero or standard deviation increases dramatically, residual mean deviates from zero and standard deviation remains low or a combination.

Wang et al. [112] built regression-based deep NN models of lubricant pressure based on SCADA data. They used MAPE and SDAPE (standard deviation of the absolute error) as an accuracy measure and showed performance to be superior to other models including Lasso, Ridge, k-NN, SVMs and NNs. They used Exponentially Weighted Moving Average (EWMA) charts to identify shifts in absolute percentage errors signifying failures and to prevent overfitting they use dropout layers in the network [112]; six wind farms were considered and the trained deep NNs achieved a MAPE between 2.9 and 14.25.

Orozco et al. [113] built regression models for normal temperature behaviour on a large set of SCADA data (614 turbines from 7 plants). As the data set used is large (948 GB) Orozco et al. make use of novel distributed processing frameworks such as Hadoop [38] and Spark [40]. The models learn the causal relationship between independent variables such as ambient temperature and power output and dependent ones representing component temperature. Multiple types of regression models were built including linear and polynomial regression, random forests and NNs. Root mean squared error (RMSE) was used as a measure to find the best model fit and the fitted values were used to adjust the entire temperature data. This adjustment has the purpose of removing the false positives, for example temperature spikes that can be explained by a high ambient temperature or turbine producing more energy. The residuals above the 99th quantile were flagged and those followed by turbine shutdown (power drops to zero) were labeled as turbine failures.

#### 4.4. Models: Classification-based

Classification models find a relationship between independent variables typically grouped in vectors and one of several predefined

categories identified by labels. During the training phase, we feed in each input vector together with the corresponding system state indicated by a label. The input vector can consist of features extracted from preprocessed time series of signals relevant to the modeled component. For generator CM we might, for example, have categories such as “healthy”, “winding failure”, “brush failure”, etc.

As a supervised ML methodology, classification needs labels to be assigned specifying the category to which training instances belong. This is time consuming, error prone and likely to result in a set of labeled vectors with an unbalanced number of classes. This is a common issue in practice [114]. There are several ways that the problem of unbalanced classes has been addressed [115], including under-sampling (remove instances belonging to the majority class), oversampling (sample more instances from minority class), SMOTE [116], Tomek-links (which removes points in the majority class that are considered borderline, noise or redundant) [117], etc.

Several CM tasks have been explored using classification. Verma et al. [15], for example, developed generator brush failure classification models based on SCADA data sampled every 10 min. For the relevant signal selection step, they used chi-square (filter technique), boosting tree (embedded method) and a wrapper method with genetic search and found 10 signals to be predictive of generator brush failure (nacelle revolution, drive train acceleration, etc.). They solved the imbalance problem using a combination of Tomek-links and Random-Forest based data sampling (which use ensembles of classification trees trained on bootstrap samples). By using these class equalization techniques, they observe an accuracy increase between 82.1% and 97.1% with a boosting ensemble model. Their results show that brush failures can be predicted with reasonable accuracy 12 h before they occur.

Leahy et al. [118] considered three generator fault classification scenarios: fault detection (two cases: fault and other), fault diagnosis (five classes including generator heating, power feeder cable, generator excitation, air cooling malfunction faults and other) and fault prognosis with the aim to predict faults at time intervals before they occurred. The data used came from a 3 MW wind turbine situated in Ireland; Leahy et al. selected 29 features from the SCADA system to be used in classification. Given the unbalanced class data, different undersampling and oversampling procedures were used when training SVM classifiers. For fault detection, recall was high (78%–95%) but precision was low (2%–4%) suggesting a high number of false positives. High recall and low precision were also found for the diagnostic and prognostic cases.

Ibrahim et al. [23] developed a general model of variable speed wind turbine in order to investigate mechanical fault detection (such as the ones related to rotor eccentricity) through the use of NNs applied to generator current signal. Several NNs are trained on the simulated healthy/faulty current signals coming at different rotational speed ranges. They report median classification accuracy between 93.5% and 98% for different NNs on simulated transient fault data; however, the models achieved lower accuracy when predicting linearly increasing faults. Besides optimizing the NNs used, future work is needed to validate this method on real data.

Kusiak et al. [22] used SCADA data to build models that could identify/predict faults at different granularity levels (fault and no-fault prediction; fault category (severity); and specific fault prediction). They reported that NN ensembles outperformed NNs, Boosting Tree Algorithms (BTAs) and SVMs when building level 1 models (that discriminate at higher granularities: failure/status). For level 2 models (that identify the category of status and failures), CART (standard Classification and Regression Tree) was identified as being the most accurate followed by SVMs, NNs and BTAs. Level 3 models identify specific types of statuses and faults (such as Malfunction of Diverter) and at this level of granularity BTAs were

identified at being the best.

Tang et al. [119] built a classifier to identify gear, bearings, shaft and general transmission failures. They performed non-linear dimensionality reduction of vibration signals using Orthogonal Neighborhood Preserving Embedding (ONPE) [120] with Shannon Wavelet Support Vector Machines (SWSVM). Compared to Locality Preserving Projection (LPP) obtaining 62% and Locally Linear Embedding (LLE) obtaining 52%, ONPE resulted in an accuracy of 92% when used for predicting inner race crack in bearings. A standard SVM with a radial basis function kernel obtained an accuracy of 76%. Jiang et al. [121] consider eight classes of health condition (normal, gear broken tooth, imbalanced shaft, etc.). They used a set of robust and abstract features extracted from vibration signals through the use of a Multiscale Convolutional Neural Network (MSCNN). The original vibration signals are down-sampled by constructing consecutive coarse-grained signals and averaging original data points with the use of non-overlapping windows before being fed to a CNN made of several convolutional and pooling layers. Jiang et al. [121] suggest the learned multiscale representations may contain complementary and rich faults that are not immediately observable in the raw vibration signal on only one single scale. By considering multiscale learning together with NNs, the algorithm can be fed raw signals and can output health condition labels without other intermediary steps such as pre-processing (an approach called end-to-end learning). The proposed method achieved the best overall performance with 98.53% F1 measure (discussed in Section 4.5).

Blades are exposed to gravitational and aerodynamic loads under possibly harsh environment conditions with vibration forces that lead to damages such as cracks, erosion, pitch angle twists, bends and loose connections with the hubs. Traditionally turbine blades are visually inspected using a time-based maintenance strategy which are infrequent, expensive and pose physical risks to the inspector [122]. Vibration signals from a variable speed wind turbine were used to construct classifiers that can distinguish between various conditions [123]. A C4.5 decision tree model [124] was used to select four features (sum, range, standard deviation, kurtosis) of 12 statistical descriptors computed from time-domain accelerometer data. Using these features, a best-first decision tree achieving 85.33% in accuracy was compared with a functional tree that achieved 91.67%. It should be noted that the confusion matrices showed a high amount of errors due to misclassification between good and loose connections between hub and blades. A possible reason was that at high speeds the blade may stick to the hub and behave as normal despite the loose bolts.

Santos et al. [125] proposed an SVM classification-based method to detect several types of faults related to rotor blade imbalance and misalignment (or a combination of both) on simulated data. They compare different SVM kernels to NNs and find that the best accuracy is obtained using a linear kernel SVM (suggesting that the data is linearly separable). As opposed to other kernels (such as Gaussian), a linear kernel has only one parameter, which reduces training and tuning time.

Godwin et al. [126] investigated rule-based classifiers for blade pitch faults (e.g. deviations of blade pitch angle from a predefined optimum) using SCADA data. They note that pitch faults are over-represented in SCADA systems, with one third of the errors attributed to this. Ripper models (a type of inductive rule-learner) were used for rule extraction [127] to classify between three types of failure (“no pitch fault”, “potential pitch fault”, “pitch fault established”); 10 independent variables (including average and maximum wind speed, pitch motor torque, etc.) were used to obtain accuracies between 85.73% (for 4 months data) and 87.41% (full data). Typically, larger datasets result in higher accuracies at the cost of larger rule-sets. Godwin et al. [126] selected a model

with a lower accuracy of 85.50% but with a smaller rule-set (14 rules) validated by a domain expert. Deployed as an expert system (with human readable rules), it reduced the number of alarms and thus the quantity of information that the operator must manage. The problem with interpretability of the rules becomes non-trivial in cases where the list of rules becomes large or/and they are derived on top of over-engineered features.

Wang et al. [128] used Unmanned Aerial Vehicles to take images used in classifiers to detect surface cracks in blades. Remotely controlled and equipped with high resolution cameras, these vehicles can be deployed far offshore to aid in visual inspection and automatic investigation of structural health of the turbine. The authors used the Viola-Jones framework for deriving Haar-features from images that were latter used in cascading classifiers (sets of classifiers applied sequentially). Although the classifier achieved high accuracy (98%), it would be interesting to explore how use of deep convolutional NNs [129] would perform on this task.

In addition to images, acoustic-based health monitoring of the blades was explored. Acoustic-based fault detection with ML algorithms is a novel and challenging area for blade CM. For example, Regan et al. [122] used hollow blade cavities of an experimental turbine, which were fitted with wireless speakers and a microphone attached to the tower. Seven types of acoustic excitation were performed to assess how well specific frequency tones or ranges can discriminate between 28 use cases. The use cases involved holes and split edges of different sizes and different locations along the blade while the turbine was stationary (24 cases) or rotating (4 cases). Twelve features representing statistical measures such as mean, median, RMS, kurtosis, etc. as well as peaks of the Fourier transform were used, which were further filtered using Fisher Ratio and a distinguishability measure. The best accuracy (98%) was achieved by using an SVM with a cubic kernel and a multi-mid acoustic excitation on stationary blade tests. It was suggested that hole-type damages are best discovered using multi-high excitations and edge splits using white noise. In contrast to the experimental turbine, operational off-shore wind turbines will be subjected to various sources of noise that will need to be filtered out; a possible way of achieving the filtering is through blind signal separation [130].

Motivated by ease of interpretation and implementation, Abdallah et al. [131] have used decision trees to identify faults, damage and abnormal operations in a windfarm data obtained from 48 wind turbines over a 12 month period (sampled every 10 min across 64 channels). For training the decision tree classifiers, Abdallah et al. [131] preferred to manually select features of interest and did not use a dimensionality reduction algorithm (such as PCA). A bagged decision tree ensemble was used where multiple CART trees were trained using bootstrap sampling, i.e., generating multiple predictors based on random subsets of the original data. Using the trained trees, a sequential trace of events leading to the fault can be expressed as a set of rules that are easier to interpret.

#### 4.5. Models: validation

Many models reported in the literature lack proper validation procedures. A standard approach in ML is to use cross validation techniques accompanied by hold-out sets. To evaluate a model, the original dataset is split into training (typically 66% of the data) and testing (33%). Using the training data, k-fold cross validation (typically  $k = 10$ ) is performed with different hyperparameters for each model tested noting the results in terms of validation measures. Relatively few papers reporting classification for CM use this approach, now viewed as standard in other domains. This may make the derived CM models more brittle by inflating accuracy results and with unknown generalization performance.

Furthermore, wider access to code, data and the use of common evaluation frameworks based on k-fold cross validation may be valuable.

Validity of regression-based normal behaviour models (such as the one modelling a power curve) can be expressed through several measures based on  $\hat{Y}(w)$  (predicted output) and actual output  $Y(w)$  as function of wind speed  $w$  [97]. Some of the most common measures include:

$$\text{MAE} = \frac{1}{N} \sum_{w=1}^N |\hat{Y}(w) - Y(w)| \quad (1)$$

$$\text{MAPE} = \frac{1}{N} \sum_{w=1}^N \frac{|\hat{Y}(w) - Y(w)|}{\hat{Y}(w)} * 100 \quad (2)$$

$$\text{sMAPE} = \frac{1}{N} \sum_{w=1}^N \frac{|\hat{Y}(w) - Y(w)|}{\left( \frac{|\hat{Y}(w) - Y(w)|}{2} \right)} * 100 \quad (3)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{w=1}^N (\hat{Y}(w) - Y(w))^2} \quad (4)$$

$$R^2 = 1 - \frac{\sum_{w=1}^N (\hat{Y}(w) - Y(w))^2}{\sum_{w=1}^N (Y(w) - \bar{Y})^2} \quad (5)$$

MAE is mean absolute error, MAPE is absolute percentage error, sMAPE is symmetric mean absolute percentage error, RMSE is root mean squared error and  $R^2$  is the coefficient of determination.

These measures are commonly used not only for training of power curve models but for any normal-behavior, regression-based models. We view these measures as distances between vectors: the predicted  $\hat{Y}$  and the actual model output values  $Y$ . We note that MAE and RMSE, the most commonly used measures in the literature, correspond to the Manhattan ( $l_1$  norm) and Euclidean distances ( $l_2$  norm), which are specific cases of Minkowski distance ( $l_k$  norm):

$$l_k = \left( \sum_{w=1}^N (\hat{Y}(w) - Y(w))^k \right)^{\frac{1}{k}} \quad (6)$$

The choice of the  $l_k$  norm can have an important effect on the results. Choosing lower norms  $k$  typically results in measures being less sensitive to outliers with some papers arguing for the use of MAE over RMSE [132]. In many cases, the choice is relative to the problem at hand and if we care that the model makes occasional large errors then we use RMSE to compare between several options [133]. MAPE, a less used accuracy measure, requires the denominator to not be zero and outputs accuracy as a percentage [134]. To overcome the symmetry problem in MAPE (where changing  $\hat{Y}(w)$  with  $Y(w)$  results in different values), sMAPE (symmetric MAPE) was introduced.  $R^2$  represents the proportion of variance in the dependent variable that can be described by the independent variable. Typically used for linear regression,  $R^2$  has values in the range [0,1] with higher values representing better models. Drawbacks of using  $R^2$  are discussed in Ref. [135]. Options for error measures for regression forecasting are discussed in Refs. [136–138].

For classification-based models, which can be used for both

diagnostic and prognostic purposes, several measures have been used which derive from the following observations: the number of True Positives (TPs, when the component was predicted as healthy and it was in fact healthy), False Negatives (FN, when the model predicted unhealthy and it was in fact healthy), True Negative (TN, predicted as unhealthy and actually unhealthy) and False Positive (FP, when the model predicted healthy and the instance was unhealthy).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (9)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (10)$$

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

While accuracy of the model is important, other measures can be relevant (8–11). For example, for a classifier to be accurate in predicting the negative (unhealthy class), we look at the specificity measure.

It is important to note that errors made by different models can be understood as a trade-off between variance and bias. Here, an error can be decomposed into a sum of bias, variance and unexplained errors [139]. Typically, the more assumptions made about the function that we want to model (a power curve for example) the higher the bias and lower the variance. If the normal-behaviour data on which we train the model is truly representative of a healthy turbine, a low bias model such as an NN might be more appropriate. In this case we want to make sure a priori that no outliers are present in the data before training as they may make the model overfit.

#### 4.6. Using ML models in CM decision support systems

Ultimately, ML models should be integrated into a decision support system that provides a stream of high-level CM information in real-time to human operators. The usefulness of ML CM models in decision making stems not only from their accuracy in identifying or predicting failures but also in their potential to explain how conclusions are reached. Thus, they can be categorized on a spectrum based on the interpretability of their underlying model. On one side we have “white box” models such as decision trees (which are hierarchical structures that can be interpreted as a set of rules). Provided they are not too large in depth, an operator can follow how the system reached a certain conclusion. The user can opt to discard or retrain the model if it does not make sense from an engineering or physical perspective, or may learn new aspects about the monitored component not previously considered. As decision trees grow in depth or if multiple trees are used (such as in ensembles) their interpretability diminishes. In contrast, we have “black box” models that provide little or no interpretation. Despite the successes of NNs on information-rich data, there remains a need to understand how they operate (e.g. combining existing techniques such as feature visualization with semantic dictionaries (explaining what the network sees) and attribution models (explaining relationships between neurons) [140]).

While standard classification models that return a specific class

label may be useful, adding a probability (or confidence) for predicted labels can be used as part of the decision strategy. Existing classifiers such as logistic regression or Naïve Bayes are by default probabilistic, while others such as SVMs can be extended [141]. Calibration plots, also known as reliability diagrams, have been traditionally used to investigate how well the probabilities the model is outputting are calibrated.

In the case of regression models, the difference between a model's predicted value in normal conditions and the actual registered value gives an indication whether there is an incipient failure in the subcomponent and its degree of magnitude. It is important to note that thresholds need to be defined in order to decide when an error magnitude is sufficient to be of concern; multiple such errors need to occur in order to trigger an alarm, otherwise the error might just be an outlier (a false alarm) [142]. The question of when to raise an alarm with regression-based methods has received much attention with many methodologies being proposed, such as absolute thresholds, confidence bands [119], experience [103], Mahalanobis distance [123], exponentially weighted moving average, etc.

Monitoring code that runs periodically and checks the deployed models is essential as performance degradation might occur with time and faults might develop in the data processing pipelines [25]. Performance might degrade because of failing hardware (e.g. sensors) that supply models with data. Monitoring these inputs on a regular basis is an important part of any online learning procedure and retraining models may be triggered by monitoring code. Alternatively, some classification models can be tuned to learn online as new data comes in, provided the signals are clean (so that they do not degrade the performance in the learning process). Online learning for regression-based models trained on a healthy state may be trickier to implement as newer data may not follow the data with which the model has been trained.

## 5. Conclusions

Recent developments in computational capabilities have opened opportunities for integrated and more in-depth CM analytics, where different types of data can be used to facilitate informed, reliable, cost-effective and robust decision-making relying on actionable information about developing hazards. Better monitoring practices, through use of ML techniques, can inform planning, resulting in fewer maintenance interventions to offshore wind farms.

This paper has reviewed ML-driven CM. The reviewed studies focus on various tasks, including blade fault detection, generator temperature monitoring, power curve monitoring, etc. We found that most models in the literature use SCADA, simulated or, rarely, experimental data; few approaches also utilised images and audio signals. Work uses classification methods (in two-thirds of cases) and regression (the remainder). Specifically, NNs, SVMs and decision trees are most commonly used.

A major hindrance to progress is a lack of large public datasets where new models could be developed, evaluated and compared. Relying on synthetically generated data produced by test rigs or mathematical models may not generalize well to actual real-world conditions. Further work is also needed for identification of relevant signals, given the potential volume of generated CM datasets.

While dimensionality reduction methods may work in specific cases, careful consideration should be taken when making assumptions about characteristics of high dimensional spaces (e.g. linearity/non-linearity).

The choice of ML model depends on the problems to be solved, as it is unlikely that a single model would outperform others over all datasets and for all tasks. However, there are indications that



deep NNs, capable of learning complex non-linear functions, may achieve better performance (in terms of accuracy) than more traditional models as data volume grows [143]. Although shallow NNs have been used, less attention has been given to deeper models and this may represent a significant avenue for achieving higher performances (caveated by their interpretability).

Although the data types might narrow the search for models and hyper-parameters, the space remains large and there are several strategies to investigate in the search for the best model for individual sub-problems. Examples of model hyper-parameter optimization approaches include grid (exhaustive), Bayesian and Randomized search facilitated by powerful hardware. Given that searching for the best models is a demanding, high-latency task, big data processing platforms such as Apache Spark, are needed to facilitate rapid model training via parallelization. Such systems also require new types of distributed, fault-tolerant storage platforms,

such as Apache HDFS, that allow for a wider variety and granularity of data types (e.g. low granularity signals from bearings; higher granularity from the whole generator) as well as flexible data refresh intervals (e.g. different sensing rates or obtaining additional CM data on demand for missing values).

## Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) through grants EP/P009743/1 and EP/L021463/1. The authors are thankful to all colleagues and partners on the HOME Offshore project (<http://homeoffshore.org/>).

## Appendix A. Feature processing <sup>1</sup>

Paper	Task	Method	Description
[44]	Outlier detection	Univariate outlier detection	Detecting outliers in time signals.
[45]	Feature Selection	Wrapper	Prediction performance of a given model is used to assess the usefulness of a subset of features.
[48]	Feature Selection	Embedded methods	Perform feature selection as part of the training of the ML model.
[49]	Feature Selection	Filter methods	Independent from the model, these methods perform significance tests between each feature and a dependent variable; can rank the features based on importance.
[52]	Feature Extraction	Statistics	Statistical features such as max/min, standard deviation, mean, median are extracted from signal and used as features.
[53], [54]	Feature Extraction	Parameters of fitted time series models	Models are fitted (e.g. Auto-regressive moving average) and their coefficients are used as features.
[51]	Feature Extraction	Time-Frequency: Fourier transform	Decomposes a signal in time domain to its constituent frequency components; unsuitable for non-stationary signals.
[55–61,65]	Feature Extraction	Time-Frequency: Wavelet transform	Dilatation and translation of a “mother” function are convoluted with the signal resulting in a set of Wavelet coefficients that express the amount of similarity between the two.
[66–69,71,85–89]	Feature Extraction	Time-Frequency: Empirical mode decomposition	Signal decomposition method that finds a set of intrinsic mode functions at different frequencies that sum to obtain the original signal; suited for non-stationary, non-linear time series.
[92], [25]	Feature Extraction	Autoencoders	Potentially deep NNs composed of an encoder which converts the input to an internal representation (extracted features) and a decoder which converts internal representation into outputs.

<sup>1</sup> Note: Some of the cited papers use methods that are applicable for machine learning, although not specifically used for wind turbine condition monitoring.



## Appendix B. Regression/Classification models<sup>2</sup>

Paper	Year	Data	Task	Method	Evaluation
[107]	2012	SCADA	Turbine Performance assessment	Regression vs Unsupervised methods	GMM models exhibited more gradual health change, being more suitable than SOMs and NNs.
[109]	2011	SCADA	Generator bearing failure prediction	Regression/Normal Behaviour Model	Nonlinear neural network approaches outperform the regression models.
[104]	2013	SCADA	Power curve monitoring	Regression/Normal Behaviour Model	MAPE: 8.25 for ANFIS
[112]	2017	SCADA	Lubricant pressure monitoring	Regression/Normal Behaviour Model	MAPE between 2.9 and 14.25
[144]	2012	SCADA	Generator temperature monitoring	Regression/Normal Behaviour Model	NSET achieves better modelling accuracy than a standard NN.
[113]	2018	SCADA	Gearbox	Regression/Normal Behaviour Model	7.66 in RMSE for Multivariate Polynomial Regression and 8.58 for Linear Regression.
[22]	2011	SCADA	Predict failures at different granularities	Classification	Status/fault: NN Ensem.:74.5% Category of status/fault: CART: 96.08% Specific fault: BTA:69.8%. SVM: 98%
[125]	2015	Simulated/Experimental data	Detect several types of faults related to rotor blade imbalance and misalignment (or both)	Classification	NN: 93%–98%
[23]	2016	Simulated/Experimental data	Mechanical fault detection	Classification	Best-first tree: 85.33% Functional trees:91.67
[123]	2017	Simulated data (small scale wind turbine: FP50W-12V)	Blade faults detection	Classification	RIPPER (rule-based): 85.5% Cascading-classifiers based on Haar-features: 98.6%.
[126]	2013	SCADA	Blade faults detection	Classification	SVM: 98.3%
[128]	2017	UAV images	Blade faults detection	Classification	Boosting tree: 82.7%–97.1% for all time stamps and 59.2%–100% for specific fault cases
[122]	2017	Simulated/Experimental data	Blade fault detection	Classification	SWSVM:92% SVM: 76%
[15]	2012	SCADA	Generator Brush Failure prediction	Classification	MSCNN obtained 98.53% F1.
[119]	2014	SCADA	Transmission system fault diagnosis	Classification	SVM obtained high recall but low precision on a number of scenarios.
[121]	2018	Simulated/Experimental data	wind turbine gearbox fault detection	Classification	Ensemble decision trees.
[118]	2018	SCADA	Fault detection, diagnosis, and prediction of generator faults	Classification	
[131]	2018	SCADA	Structural vibration errors; root cause analysis	Classification	

## References

- [1] M. Bhattacharya, S.R. Paramati, I. Ozturk, S. Bhattacharya, The effect of renewable energy consumption on economic growth: evidence from top 38 countries, *Appl. Energy* 162 (Jan. 2016) 733–741.
- [2] Lena Bergström, et al., Effects of offshore wind farms on marine wildlife—a generalized impact assessment, *Environ. Res. Lett.* 9 (2014), pp. 34012–12.
- [3] H. Offshore, Offshore Windfarm Condition Monitoring and Maintenance (Position Paper), 2017.
- [4] J. Helsen, C. Devriendt, W. Weijtjens, P. Guillaume, Condition monitoring by means of SCADA analysis, in: EWEA 2015 Annu. Event, No. November 2015, 2015.
- [5] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T.A. Tameghe, G. Ekemb, Wind turbine condition monitoring: state-of-the-art review, new trends, and future challenges, *Energies* 7 (4) (2014) 2595–2630.
- [6] B.N. Madsen, Condition Monitoring of Wind Turbines by Electric Signature Analysis. A Cost Effective Alternative or a Redundant Option for Geared Wind Turbines, 2011, p. 118, no. October.
- [7] W. Qiao, D. Lu, A survey on wind turbine condition monitoring and fault Diagnosis—Part II: signals and signal processing methods, *IEEE Trans. Ind. Electron.* 62 (10) (2015) 6546–6557.
- [8] S. Pfaffel, S. Faulstich, K. Rohrig, Performance and reliability of wind turbines: a review, *Energies* 10 (11) (2017).
- [9] J. Carroll, A. McDonald, D. McMillan, Failure rate, repair time and unscheduled O&M cost analysis of offshore wind turbines, *Wind Energy* 19 (6) (Jun. 2016) 1107–1119.
- [10] Y. Feng, Y. Qiu, C. J. Crabtree, H. Long, and P. J. Tavner, Use of SCADA and CMS Signals for failure detection and diagnosis of a wind turbine gearbox..
- [11] C.J. Crabtree, D. Zappala, P.J. Tavner, Survey of Commercially Available Condition Monitoring Systems for Wind Turbines, vol. 44, 2014, pp. 0–22, no. May.
- [12] H. Dias, M. De Azevedo, A.M. Araújo, N. Bouchonneau, A review of wind turbine bearing condition monitoring: state of the art and challenges, *Renew. Sustain. Energy Rev.* 56 (2017) 368–379.
- [13] S. Huang, X. Wu, X. Liu, J. Gao, and Y. He, Overview of condition monitoring and operation control of electric power conversion systems in direct-drive wind turbines under faults..
- [14] H. Luo, Physics-based data analysis for wind turbine condition monitoring, *Clean Energy* 1 (1) (Dec. 2017) 4–22.
- [15] A. Verma, A. Kusiak, Fault monitoring of wind turbine generator brushes: a data-mining approach, *J. Sol. Energy Eng.* 134 (2) (May 2012), 021001.
- [16] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.
- [17] F. Rosenblatt, The Perceptron – a Perceiving and Recognizing Automaton, Report 85, Cornell Aeronautical Laboratory, 1957, pp. 460–461.
- [18] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Network*. 2 (5) (1989) 359–366.
- [19] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, *Nature* 323 (6088) (Oct. 1986) 533–536.
- [20] F. Jia, Y. Lei, J. Lin, X. Zhou, N. Lu, Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data, *Mech. Syst. Signal Process.* 72–73 (May 2016) 303–315.
- [21] C. Lu, Z.-Y. Wang, W.-L. Qin, J. Ma, Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification, *Signal Process.* 130 (Jan. 2017) 377–388.
- [22] A. Kusiak, W. Li, The prediction and diagnosis of wind turbine faults, *Renew. Energy* 36 (1) (2011) 16–23.
- [23] R.K. Ibrahim, J. Tautz-Weinert, S.J. Watson, *Neural Networks for Wind Turbine Fault Detection via Current Signature Analysis*, 2016.
- [24] S.A. Kalogirou, Artificial neural networks in renewable energy systems applications: a review, *Renew. Sustain. Energy Rev.* 5 (4) (Dec. 2001) 373–401.
- [25] A. Géron, *Hands-on Machine Learning with Scikit-learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2017.
- [26] F. Pedregosa, et al., Scikit-learn: machine learning in [P]ython, *J. Mach. Learn. Res.* 12 (2011) 2825–2830.
- [27] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, *Data Min. Knowl. Discov.* 2 (2) (1998) 121–167.
- [28] Open CV, Introduction to Support Vector Machines — OpenCV 2.4.13.5 Documentation .
- [29] M.L. Wymore, J.E. Van Dam, H. Ceylan, D. Qiao, A survey of health monitoring systems for wind turbines, *Renew. Sustain. Energy Rev.* 52 (1069283) (2015)

<sup>2</sup> These models were selected based on novelty, relevancy (Google Scholar) and their comprehensive approach to Machine Learning (papers which discuss for example, feature selection, validation measures, etc.).

- 976–990.
- [30] W. Yang, R. Court, J. Jiang, Wind turbine condition monitoring by the approach of SCADA data analysis, *Renew. Energy* 53 (2013) 365–376.
- [31] A. Zaher, S.D.J. McArthur, D.G. Infield, Y. Patel, Online wind turbine fault detection through automated SCADA data analysis, *Wind Energy* 12 (6) (2009) 574–593.
- [32] J. Tautz-Weinert, S.J. Watson, “Using SCADA data for wind turbine condition monitoring – a review, *IET Renew. Power Gener.* 11 (4) (2017) 382–394.
- [33] Z.J. Viharos, et al., “Big data ” initiative as an IT solution for improved operation and maintenance of wind turbines, in: *Ewea* 2013, 2013, pp. 184–188.
- [34] E. Gholamzadeh Nabati, K. Dieter Thoben, E. Nabati, and K. Thoben, “Big data analytics in the maintenance of off-shore wind turbines: a study on data characteristics 12.1 introduction,” *Lect. Notes Logist.*
- [35] J. Helsen, G. De Sitter, P.J. Jordaens, Long-Term monitoring of wind farms using big data approach, in: 2016 IEEE Second Int. Conf. Big Data Comput. Serv. Appl., 2016, pp. 265–268.
- [36] E. Rahm, H.H. Do, Data cleaning: problems and current approaches, *Bull. Tech. Commun. Data Eng.* 23 (4) (2000) 3–13.
- [37] M. Canizo, E. Onieva, A. Conde, S. Charramendieta, S. Trujillo, Real-time predictive maintenance for wind turbines using Big Data frameworks, in: 2017 IEEE Int. Conf. Progn. Heal. Manag., 2017, pp. 70–77.
- [38] Apache Hadoop Releases. [Online]. Available: <http://hadoop.apache.org/releases.html>.
- [39] Apache Kafka, Apache Kafka. [Online]. Available: <https://kafka.apache.org/>.
- [40] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: cluster computing with working sets.
- [41] Apache Spark, “Apache Spark™ - Lightning-Fast Cluster Computing.” [Online]. Available: <https://spark.apache.org/>.
- [42] Apache Mesos, “Apache Mesos.” [Online]. Available: <http://mesos.apache.org/>.
- [43] Edureka, “Apache Spark vs Hadoop: Choosing the Right Framework | Edureka Blog.” [Online]. Available: <https://www.edureka.co/blog/apache-spark-vs-hadoop-mapreduce>.
- [44] P. Martí-Puig, A. Blanco-M, J.J. Cárdenas, J. Cusidó, J. Solé-Casals, Effects of the pre-processing algorithms in fault diagnosis of wind turbines, *Environ. Model. Software* (2018) 0–1, no. February 2017.
- [45] R. Kohavi, G.H. John, Wrappers for feature subset selection, *Artif. Intell.* 97 (1997) 273–324.
- [46] I. Guyon, A. Elisseeff, A.M. De, An introduction to variable and feature selection, *J. Mach. Learn. Res.* 3 (2003) 1157–1182.
- [47] E. Amaldi, V. Kann, On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems, *Theor. Comput. Sci. Kann I Theor. Comput. Sci.* 209 (2009) (1998) 237–260.
- [48] I. Guyon, S. Gunn, A. Ben Hur, and G. Dror, “Result analysis of the NIPS 2003 feature selection challenge.
- [49] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, *Artif. Intell.* 97 (1–2) (1997) 245–271.
- [50] B. Chakraborty, Feature Selection and Classification Techniques for Multivariate Time Series, *Ieee*, 2007, pp. 2–5.
- [51] F. Mörchén, Time Series Feature Extraction for Data Mining Using DWT and DFT, 2003.
- [52] D. Kateris, D. Moshou, X.-E. Pantazi, I. Gravalos, N. Sawalhi, S. Loutridis, A machine learning approach for the condition monitoring of rotating machinery, *J. Mech. Sci. Technol.* 28 (1) (2014) 61–71.
- [53] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, *Time Series Analysis: Forecasting & Control*, 1994.
- [54] L.P. Heck, J.H. McClellan, Mechanical system monitoring using hidden Markov models, in: [Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing, vol. 3, 1991, pp. 1697–1700.
- [55] P. Li, F. Kong, Q. He, Y. Liu, Multiscale slope feature extraction for rotating machinery fault diagnosis using wavelet analysis, *Measurement* 46 (2013) 497–505.
- [56] H. Heidari Bafroui, A. Ohadi, Application of wavelet energy and Shannon entropy for feature extraction in gearbox fault detection under varying speed conditions, *Neurocomputing* 133 (2014) 437–445.
- [57] J. Lin, Feature extraction based on Morlet Wavelet and its application for mechanical fault diagnosis, *J. Sound Vib.* 234 (1) (Jun. 2000) 135–148.
- [58] T.S.B. Tang, W. Liu, Wind turbine fault diagnosis based on Morlet wavelet transformation and Wigner-Ville distribution, *Renew. Energy* 35 (12) (Dec. 2010) 2862–2866.
- [59] Y. Jiang, B. Tang, Y. Qin, W. Liu, Feature extraction method of wind turbine based on adaptive Morlet wavelet and SVD, *Renew. Energy* 36 (8) (2011) 2146–2153.
- [60] G. Yu, C. Li, S. Kamarthi, Machine Fault Diagnosis Using a Cluster-based Wavelet Feature Extraction and Probabilistic Neural Networks, 2008.
- [61] S.E. Khadem, M. Rezaee, Development of vibration signature analysis using multiwavelet systems, *J. Sound Vib.* 261 (2003) 613–633.
- [62] Y. Wenxian, P.J. Tavner, C.J. Crabtree, M. Wilkinson, Cost-effective condition monitoring for wind turbines, *Ind. Electron. IEEE Trans.* 57 (1) (2010) 263–271.
- [63] W. Yang, C. Little, R. Court, S-Transform and its contribution to wind turbine condition monitoring, *Renew. Energy* 62 (Feb. 2014) 137–146.
- [64] P. Chen, M. Taniguchi, T. Toyota, Z. He, Fault diagnosis method for machinery in unsteady operating condition by instantaneous power spectrum and genetic programming, *Mech. Syst. Signal Process.* 19 (1) (Jan. 2005) 175–194.
- [65] R. Yan, R.X. Gao, X. Chen, Wavelets for fault diagnosis of rotary machines: a review with applications, *Signal Process.* 96 (2013) 1–15.
- [66] N.E. Huang, et al., The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, *Proc. R. Soc. A Math. Phys. Eng. Sci.* 454 (1971) (Mar. 1998) 903–995.
- [67] Z. Wu, N.E. Huang, Ensemble Empirical Mode Decomposition: a Noise Assisted Data Analysis Method, 2005.
- [68] X. Zhang, Y. Liang, J. Zhou, Y. Zang, A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM, *Meas. J. Int. Meas. Confed.* 69 (2015) 164–179.
- [69] C.W.de S.L. Lei, J. Yan, Dominant feature selection for the fault diagnosis of rotary machines using modified genetic algorithm and empirical mode decomposition, *J. Sound Vib.* 344 (May 2015) 464–483.
- [70] K.W.I. Antoniadou, G. Manson, W.J. Staszewski, T. Barszcz, “A time–frequency analysis approach for condition monitoring of a wind turbine gearbox under varying load conditions, *Mech. Syst. Signal Process.* 64–65 (Dec. 2015) 188–216.
- [71] J. Ben Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello, F. Fnaiech, Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals, *Appl. Acoust.* 89 (2015) 16–27.
- [72] A. Röscher and H. Schmidbauer, WaveletComp: A guided tour through the R-package.
- [73] A. Roesch, H. Schmidbauer, M.A. Roesch, Package ‘WaveletComp’ Title Computational Wavelet Analysis, 2015.
- [74] M. Riera-Guasp, J.A. Antonino-Daviu, G.-A. Capolino, Advances in electrical machine, power electronic, and drive condition monitoring and fault detection: state of the art, *IEEE Trans. Ind. Electron.* 62 (3) (2015) 1746–1759.
- [75] Z.K. Peng, F.L. Chu, Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography, *Mech. Syst. Signal Process.* 18 (2) (2004) 199–221.
- [76] Pfbach.dk, Time Series Tree. [Online]. Available: [http://www.pfbach.dk/firma\\_pfb/time\\_series/ts.php](http://www.pfbach.dk/firma_pfb/time_series/ts.php).
- [77] Z. Feng, M. Liang, F. Chu, Recent advances in time-frequency analysis methods for machinery fault diagnosis: a review with application examples, *Mech. Syst. Signal Process.* 38 (1) (2013) 165–205.
- [78] W. Teng, X. Ding, X. Zhang, Y. Liu, Z. Ma, Multi-fault detection and failure analysis of wind turbine gearbox using complex wavelet transform, *Renew. Energy* 93 (Aug. 2016) 591–598.
- [79] C. Ng, N. Lieven, P. Morrish, C. Murray, M. Mulroy, M. Asher, Wind turbine drivetrain health assessment using discrete wavelet transforms and an artificial neural network, in: 3rd Renewable Power Generation Conference (RPG 2014), 2014, p. 8.46–8.46.
- [80] M.D. Ulriksen, D. Tcherniak, P.H. Kirkegaard, L. Damkilde, Operational modal analysis and wavelet transformation for damage identification in wind turbine blades, *Struct. Health Monit.* 15 (4) (2016) 381–388.
- [81] H. Sun, Y. Zi, Z. He, Wind turbine fault detection using multiwavelet denoising with the data-driven block threshold, *Appl. Acoust.* 77 (Mar. 2014) 122–129.
- [82] Y. Lei, J. Lin, Z. He, M.J. Zuo, A review on empirical mode decomposition in fault diagnosis of rotating machinery, *Mech. Syst. Signal Process.* 35 (1–2) (2013) 108–126.
- [83] N.O. Attoh-Okin, N. E. (Norden E. Huang, The Hilbert-Huang Transform in Engineering, Taylor & Francis, 2005.
- [84] N.E. Huang, S.S.P. Shen, Hilbert-huang Transform and its Applications, vol. 5, WORLD SCIENTIFIC, 2005.
- [85] W. Yang, R. Court, P.J. Tavner, C.J. Crabtree, Bivariate empirical mode decomposition and its contribution to wind turbine condition monitoring, *J. Sound Vib.* 330 (15) (Jul. 2011) 3766–3782.
- [86] W. Yang, J. Jiang, P. J. Tavner, and C. J. Crabtree, “Monitoring wind turbine condition by the approach of empirical mode decomposition,” *Renew. Energy*, pp. 736–740.
- [87] A. Abouhnik, A. Albarbar, Wind turbine blades condition assessment based on vibration measurements and the level of an empirically decomposed feature, *Energy Convers. Manag.* 64 (Dec. 2012) 606–613.
- [88] Y. Amirat, V. Choqueuse, M. Benbouzid, EEMD-based wind turbine bearing failure detection using the generator stator current homopolar component, *Mech. Syst. Signal Process.* 41 (1–2) (2013) 667–678.
- [89] N.E. Huang, Z. Wu, A review on Hilbert-Huang transform: method and its applications to geophysical studies, *Rev. Geophys.* 46 (2) (Jun. 2008) RG2006.
- [90] F.P. García Márquez, A.M. Tobias, J. María, P. Pérez, M. Papaelias, Condition Monitoring of Wind Turbines: Techniques and Methods, 2012.
- [91] L. Guo, N. Li, F. Jia, Y. Lei, J. Lin, A recurrent neural network based health indicator for remaining useful life prediction of bearings, *Neurocomputing* 240 (May 2017) 98–109.
- [92] G. Jiang, H. He, P. Xie, Y. Tang, in: Stacked Multilevel-denoising Autoencoders : a New Representation Learning Approach for Wind, vol. 66, 2017, pp. 1–12, no. 9.
- [93] L.J.P. Van Der Maaten, E.O. Postma, H.J. Van Den Herik, Dimensionality Reduction : a comparative review, *Rev. Lit. Arts Am.* 10 (2007) 1–35, no. February.
- [94] S. Faulstich, B. Hahn, P.J. Tavner, Wind turbine downtime and its importance

- for offshore deployment, *Wind Energy* 14 (3) (Apr. 2011) 327–337.
- [95] A. Marvuglia, A. Messineo, "Monitoring of wind farms' power curves using machine learning techniques, *Appl. Energy* 98 (2012) 574–583.
  - [96] A. Kusiak, H. Zheng, Z. Song, On-line monitoring of power curves, *Renew. Energy* 34 (6) (2009) 1487–1493.
  - [97] A comprehensive review on wind turbine power curve modeling techniques, *Renew. Sustain. Energy Rev.* 30 (Feb. 2014) 452–460.
  - [98] M. Lydia, A.I. Selvakumar, S.S. Kumar, G.E.P. Kumar, Advanced algorithms for wind turbine power curve modeling, *IEEE Trans. Sustain. Energy* 4 (3) (Jul. 2013) 827–835.
  - [99] S. Balluff, J. Bendfeld, S. Krauter, "Short term wind and energy prediction for offshore wind farms using neural networks, in: 2015 Int. Conf. Renew. Energy Res. Appl., vol. 5, 2015, pp. 379–382.
  - [100] S. Shokrzadeh, M. Jafari Jozani, E. Bibeau, Wind turbine power curve modeling using advanced parametric and nonparametric methods, *IEEE Trans. Sustain. Energy* 5 (4) (Oct. 2014) 1262–1269.
  - [101] B. Stephen, S. Gill, S. Galloway, Y. Wang, D. McMillan, D. Infield, Wind Turbine Operation Anomaly Detection Using Copula Statistics, Feb. 2013.
  - [102] T. Ustuntas, A. Sahin, Wind turbine power curve estimation based on cluster center fuzzy logic modeling ARTICLE IN PRESS, *J. Wind Eng. Ind. Aerod.* 96 (2008) 611–620.
  - [103] M.S.M. Raj, M. Alexander, M. Lydia, Modeling of wind turbine power curve, in: ISGT2011-India, 2011, pp. 144–148.
  - [104] M. Schlechtingen, I.F. Santos, S. Achiche, Using data-mining approaches for wind turbine power curve monitoring: a comparative study, *IEEE Trans. Sustain. Energy* 4 (3) (2013) 671–679.
  - [105] M. Schlechtingen, I.F. Santos, S. Achiche, Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: system description, *Appl. Soft Comput.* J. 13 (2013) 259–270.
  - [106] A.Z. Hans-Peter Kriegel, Peer Kröger, Outlier detection techniques, in: 2010 SIAM Int. Conf. Data Min., 2010.
  - [107] E. Lapira, D. Brisset, H. Davari Ardakani, D. Siegel, J. Lee, Wind turbine performance assessment using multi-regime modeling approach, *Renew. Energy* 45 (2012) 86–95.
  - [108] N. Laouti, N. Sheibat-Othman, S. Othman, Support Vector Machines for Fault Detection in Wind Turbines, vol. 44, IFAC, 2011 no. 1.
  - [109] M. Schlechtingen, I. Ferreira Santos, Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection, *Mech. Syst. Signal Process.* 25 (5) (2011) 1849–1875.
  - [110] P. Guo, D. Infield, X. Yang, Wind turbine generator condition-monitoring using temperature trend analysis, *Sustain. Energy, IEEE Trans.* 3 (1) (2012) 124–133.
  - [111] K.C. Gross, R.M. Singer, S.W. Wegerich, J.P. Herzog, R. VanAlstine, F. Bockhorst, Application of a Model-based Fault Detection System to Nuclear Plant Signals, 1997.
  - [112] L. Wang, Z. Zhang, H. Long, J. Xu, R. Liu, Wind turbine gearbox failure identification with deep neural networks, *IEEE Trans. Ind. Inf.* 13 (3) (2017) 1360–1368.
  - [113] R. Orozco, S. Sheng, C. Phillips, Diagnostic models for wind turbine gearbox components using SCADA time series data preprint, in: 2018 IEEE Int. Conf. Progn. Heal. Manag., 2018, pp. 1–9, no. July.
  - [114] E. T. Aljurf M, A.-M. F. Shoukri M, and M. Shoukri, "Classification of imbalance data using Tomek link (T-Link) combined with random under-sampling (RUS) as a data reduction method.
  - [115] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: a review.
  - [116] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, *J. Artif. Intell. Res.* (Jun. 2011).
  - [117] I. Tomek, Experiment with the Edited Nearest-neighbor Rule, vol. SMC-6, no. 6, Jun. 1976, pp. 448–452.
  - [118] K. Leahy, R.L. Hu, I.C. Konstantakopoulos, C.J. Spanos, A.M. Agogino, D.T.J. O'Sullivan, Diagnosing and predicting wind turbine faults from SCADA data using support vector machines, *Int. J. Prognostics Health Manag.* 9 (1) (2018) 1–11.
  - [119] B. Tang, T. Song, F. Li, L. Deng, Fault diagnosis for a wind turbine transmission system based on manifold learning and Shannon wavelet support vector machine, *Renew. Energy* 62 (2014) 1–9.
  - [120] X. Liu, J. Yin, Z. Feng, J. Dong, L. Wang, Orthogonal neighborhood preserving embedding for face recognition, in: Proc. - Int. Conf. Image Process. ICIP, vol. 1, 2006, pp. 133–136.
  - [121] G. Jiang, H. He, J. Yan, P. Xie, Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox, *IEEE Trans. Ind. Electron.* PP (2018) 1, no. c.
  - [122] T. Regan, C. Beale, M. Inalpolat, Wind turbine blade damage detection using supervised machine learning algorithms (draft), *J. Vib. Acoust.* 139 (2017) 1–40, no. December.
  - [123] A. Joshuva, V. Sugumaran, A Data Driven Approach for Condition Monitoring of Wind Turbine Blade Using Vibration Signals through Best-first Tree Algorithm and Functional Trees Algorithm: a Comparative Study, 2017.
  - [124] J. R. (John R. Quinlan, J. Ross, C4.5 : Programs for Machine Learning, Morgan Kaufmann Publishers, 1993.
  - [125] P. Santos, L.F. Villa, A. Re?ones, A. Bustillo, J. Maudes, An SVM-based solution for fault detection in wind turbines, *Sensors (Switzerland)* 15 (3) (2015) 5627–5648.
  - [126] J.L. Godwin, P. Matthews, Classification and detection of wind turbine pitch faults through SCADA data analysis, *Int. J. Prognostics Health Manag.* (2013) 2153–2648.
  - [127] W.W. Cohen, Fast effective rule induction, in: Proceedings of the Twelfth International Conference on International Conference on Machine Learning, 1995, pp. 115–123.
  - [128] L. Wang, Z. Zhang, Automatic detection of wind turbine blade surface cracks based on UAV-taken images, *IEEE Trans. Ind. Electron.* 0046 (c) (2017) 1.
  - [129] O. Russakovsky, et al., ImageNet large scale visual recognition challenge, *Int. J. Comput. Vis.* 115 (2015) 211–252.
  - [130] J.F. Cardoso, Blind signal separation: statistical principles, *Proc. IEEE* 86 (10) (1998) 2009–2025.
  - [131] I. Abdallah, et al., Fault Diagnosis of Wind Turbine Structures Using Decision Tree Learning Algorithms with Big Data, 2018, pp. 3053–3061, no. iii.
  - [132] C. Willmott, K. Matsuura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, *Clim. Res.* 30 (1) (Dec. 2005) 79–82.
  - [133] Duke University, "How to compare regression models. [Online]. Available: <https://people.duke.edu/~rnau/compare.htm>.
  - [134] C. Tofallis, A better measure of relative prediction accuracy for model selection and model estimation, *J. Oper. Res. Soc.* 66 (8) (2015) 1352–1362.
  - [135] Is R-squared Useless? | University of Virginia Library Research Data Services + Sciences."
  - [136] E. Mahmoud, Accuracy in forecasting: a survey, *J. Forecast.* 3 (2) (Apr. 1984) 139–159.
  - [137] J.G. De Gooijer, R.J. Hyndman, 25 years of time series forecasting, *Int. J. Forecast.* 22 (3) (Jan. 2006) 443–473.
  - [138] J.S. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: empirical comparisons \*, *Int. J. Forecast.* 08 (1992) 69–80.
  - [139] L. Breiman, Bias, Variance and Arcing Classifiers, 1996.
  - [140] C. Olah, et al., The building blocks of interpretability, *Distill* 3 (3) (Mar. 2018) e10.
  - [141] N. Mechbal, J.S. Uribe, M. Rébillat, A probabilistic multi-class classifier for structural health monitoring, *Mech. Syst. Signal Process.* 60 (61) (Aug. 2015) 106–123.
  - [142] P. Caselitz, J. Giebardt, Advanced maintenance and repair for offshore wind farms using fault prediction techniques, *Security* 49 (0) (2002) 1–4.
  - [143] Andrew Ng, "The State of Artificial Intelligence." [Online]. Available: [https://www.youtube.com/watch?v=NKpuX\\_yzYs&t=128s](https://www.youtube.com/watch?v=NKpuX_yzYs&t=128s). [Accessed: 21-May-2018].
  - [144] P. Guo, D. Infield, X. Yang, Wind turbine generator condition-monitoring using temperature trend analysis, *IEEE Trans. Sustain. Energy* 3 (1) (Jan. 2012) 124–133.