Non-stationarity in correlation matrices for wind turbine SCADA-data and implications for failure detection

Henrik M. Bette[1], Edgar Jungblut[1], and Thomas Guhr[1]

[1]Faculty of Physics, University of Duisburg-Essen, Lotharstr. 1, Duisburg, Germany

Correspondence: Henrik M. Bette (henrik.bette@uni-due.de)

Abstract. Modern utility-scale wind turbines are equipped with a Supervisory Control And Data Acquisition (SCADA) system gathering vast amounts of operational data that can be used for failure analysis and prediction to improve operation and maintenance of turbines. We analyse high freqeuency SCADA-data from the Thanet offshore windpark in the UK and evaluate Pearson correlation matrices for a variety of observables with a moving time window. This renders possible an asessment of
5 non-stationarity in mutual depedncies of different types of data. Drawing from our experience in other complex systems, such as financial markets and traffic, we show this by employing a hierarchichal k-means clustering algorithm on the correlation matrices. The different clusters exhibit distinct typical correlation structures to which we refer as states. Looking first at only one and later at multiple turbines, the main dependence of these states is shown to be on wind speed. In accordance, we identify them as operational states arising from different settings in the turbine control system based on the available wind
10 speed. We model the boundary wind speeds seperating the states based on the clustering solution. This allows the usage of our methodology for failure analysis or prediction by sorting new data based on wind speed and comparing it to the respective operational state, thereby taking the non-stationarity into account.

1 Introduction

Renewable energies are indispensable to respond to the temperature rise caused by climate change. Wind turbines are one key
15 technology paving the way towards a green and emission-less energy production. The Global Wind Energy Council (GWEC) reports a growth of 93 GW in wind installations for 2020, bringing the total up to 650 GW with an increase of the growth rate by 53 % compared to 2019. Offshore installations are on the rise, their comparatively low total increased from 29.1 GW by more than a fifth to 35.3 GW in 2019 (Lee and Zhao, 2021). The GWEC also forecasts more than 205 GW new offshore wind capacity by 2030 in its 2020 Global Offshore Wind Report (Lee and Zhao, 2020). They point out a growing acceptance
20 of the fact that the price of offshore wind power can out-compete fossil and nuclear fuels. The various reports agree that wind power cannot be the only clean energy used to reach a net zero in emissions by 2050, but onshore and especially offshore wind turbines are crucial to reach this goal.

While offshore locations usually provide steadier and higher wind speeds, this comes at the cost of harsh environmental conditions and increased difficulties for operation and maintenance. Maldonado-Correa et al. (2020) lists various authors claiming
25 that Operation and Maintenance (O&M) costs account for 20-35% of the total expenditure for offshore wind farms. The corre-

sponding numbers are lower but still significant (approx. 10-15 %) for onshore turbines (Zhao et al., 2017; Tavner, 2012).
Not surprisingly, among the many topics to be studied in the wind turbine field (Veers et al., 2019), O&M is on focus for
researchers. They follow a general trend in most industries, aiming at moving from scheduled maintenance towards condition-
based maintenance to reduce costs and efforts (Li et al., 2020). Researchers and industry alike undertake increased efforts
30   to effectively optimize O&M procedures for wind turbines and thereby reduce this cost factor (e.g., Dao et al., 2021; Ulmer
et al., 2020; Yan et al., 2019; Martin-del Campo and Al-Kahwati, 2020; Chen et al., 2021). A key to achieving this goal is the
prediction of failures in wind turbines with sufficient lead time to react and carry out preemptive maintenance instead of cor-
rectional maintenance. This reduces not only the money lost in turbine downtime, but also enables cheaper maintenance. The
idea is to optimize assets by replacing components exactly when needed (Colone et al., 2018). Furthermore, similar methods
35   could not only be used to detect component failures, but also to control problems such as, for example, yaw misalignment or
under-performance (Pandit and Infield, 2018; Meyer and Brodbeck, 2020).

One possibility for such a Condition Monitoring System (CMS) is to install additional vibrational sensors into the wind turbine.
See Stetco et al. (2019) for a review of these methods. However, the goal is to reduce costs and installing additional sensors
has its own inherent costs. This has led many researchers to develop methods for failure detection and prediction based on
40   data from Supervisory Control and Data Acquisition (SCADA) systems, which are installed in all major wind farms since
commission. This SCADA data contains many variables, usually averaged over 10 min intervals. The corresponding standard
deviations are often included. As these are usually the only available data, most developed methods try to employ them for
failure detection. The reader is referred to Maldonado-Correa et al. (2020) and Tautz-Weinert and Watson (2017) for reviews.
Common approaches include neural networks, physical models and statistical analysis. Usually historical data are used to de-
45   fine a *normal* state of the system for these methods. The process is commonly called *training*. This concept is very clearly
outlined by Jin et al. (2021). An easily understood example is modelling the power curve (Kolumbán et al., 2017; Gottschall
and Peinke, 2008) and then testing for deviations (Aziz et al., 2021). For a neural network, it might mean to predict one (or
more) variables based on other measurements and then compare with the actual measurement. An explicit example for this is
presented by Ulmer et al. (2020), while Helbing and Ritter (2018) reviews multiple neural network approaches. These authors
50   also raise two important points: First, it is often complicated or impossible to reliably label failure events in the data due to
scarcity of available log and maintenance data. Second, early detection of failures is shown to be possible with 10 min SCADA-
data by Ulmer et al. (2020) who apply convolutional neural networks. However, it is found that the averaging process naturally
leads to a loss of information. Other researchers have tried to avoid the data availability problem by using simulated high
frequency data with labelled faults for the development of their models, e.g. Pozo et al. (2018) uses a benchmark implemented
55   by Odgaard and Johnson (2013).

Pozo et al. (2018) uses principal component analysis (PCA) (Denis, 2021) to reduce the dimensionality of the data by pro-
jecting onto the first three principal components. Novelty (i.e. failure) is then detected by applying Hotelling's T^2-statistics to
compare datasets in this principal component space. PCA is a common tool for statistical analysis also applied by others to
wind turbines (Yan et al., 2019; Vidal et al., 2018) and also to the aforementioned vibration data (Daga et al., 2020), see also
60   the aforementioned reviews. The composition of the principal components of a time series dataset is equal to the structure of

the eigenvectors one obtains from spectrally decomposing the correlation matrix of that dataset. While PCA is one major statistical tool based on correlations, it is not the only one. Correlations themselves are used on industry scale to monitor condition (Siemens, 2021). Research around failure detection often uses the Mahalanobis distance (Jin et al., 2021; Ruiz de la Hermosa González-Carrato, 2018), which also involves the correlation matrix.

65 This, however, poses an interesting question: The above approaches rely in one way or the other on the assumption of stationarity, but are the correlation matrices for SCADA data obtained from a wind turbine stationary in time? Generally, this assumption does not hold true for complex systems containing many different variables, mechanisms and external influences. The stability of correlations in financial stock market data was analysed, for example, by Buccheri et al. (2013). Münnix et al. (2012) shows that the correlation matrix of this data inhabits different states over time by using cluster analysis. The stability

70 of these states was further analysed by Rinn et al. (2015) and Stepanov et al. (2015). Similar studies were done for traffic data by Wang et al. (2020). Some general correlation analysis for wind turbine data was carried out by Braun et al. (2020).

Here, we want to show that the correlations indeed change over time when looking at high frequency SCADA data from real wind turbines. To this end, we will apply cluster analysis to the correlation matrices of different SCADA signals. Furthermore, we show that one of the prime influences leading to significantly different structures in the correlation matrix stems from the

75 turbine's own control mechanisms.

Such a distinction into different operational states in the correlation structure of wind turbine data will help improve CMS systems relying directly on correlations (such as PCA and Mahalanobis distance). Furthermore, the establishment of multiple normal states might also be beneficial for machine learning techniques like neural networks or methods tackling under performance in general.

80 This paper is structured as follows: In section 2 we will introduce the dataset we work with before moving on to the theoretical background of correlation matrices and clustering in section 3. Then we will present our clustering results for a single turbine in section 4. We find proof of non-stationarity and identify the turbine control mechanism as the prime influence. Afterwards, we model the boundary wind speeds between the states in section 5. Finally, we will show that the established method works for multiple turbines without further problems and can even be improved by the increase in available data in section 6. We

85 present our conclusions in section 7.

2 Data

Our dataset includes 100 *Vestas* V90 wind turbines from the Thanet offshore wind farm south-east of Great Britain. It contains observables that are measured at approximately 5 s time intervals. To obtain aligned, synchronized data, we average the time series on 10 s time intervals resulting in a data frequency $\nu = 1/10$ s. This ensures continuity in the data even when the actual

90 measurement frequency fluctuates around 5 s. This does not hinder the calculation of correlations for our purposes. In fact, it is rather similar to any measurement process: Every sensor will in reality average over a short time span to obtain a value. Taking the mean over 10 s seems therefore more natural than, for example, using only the last value from each interval. If at some time the deviation from the 5 s interval becomes stronger or if there are actually missing data points, data points will be missing

in our averaged data as well. This might occur if there was a problem with, for example, sensors or communications. Another reason is simply a decreased measurement frequency when the turbine is switched off. This is underlined by the majority of missing values occurring in the very low wind speed regime beneath the turn-on wind speed. As the turbines are not running during these times, this wind range is not of interest for us. These missing values do not pose a problem for our analysis. For these reasons, we decided to transfer missing values from the original dataset into the 10 s data instead of replacing them by other means.

We are interested in identifying changes in the correlation structure, which emerge while the turbine is operating normally, in contrast to changes caused by failures. Therefore, we look at the following basic observables:

  – generated active power (ActivePower)

  – generated current (CurrentL1)

  – rotation per minute of the rotor (RotorRPM)

  – rotation per minute of the high speed shaft at the generator (GeneratorRPM)

  – wind speed (WindSpeed)

These observables provide a good picture of the main turbine systems. Wind speed makes the rotor move. Its rotation is transmitted via gears to the rotation of the high speed shaft at the generator. This, in turn, generates electrical current and power. While measurements of temperatures are very common and useful for failure analysis (Jin et al., 2021; Tautz-Weinert and Watson, 2017; Maldonado-Correa et al., 2020), it seems reasonable to assume that their behaviour in normal states is always coupled to mechanical variables, e.g. higher rotation speeds will lead to increased bearing temperatures. Excluded from the study at hand, further analysis of temperature correlations is nevertheless desirable for failure detection.

We used approximately three weeks of data from 5 March 2017 to 24 March 2017. The data from such a time span are still easy enough to handle while providing enough data points to obtain reasonable clustering results. In view of possible practical applications identifying normal states for failure detection, three weeks is a short enough time span to make it easily usable. There would be no need to collect huge amounts of operational data beforehand. However, it is of course necessary that the data used for identifying different operational states covers all possible states. In practice it turns out later that this means, we need a wide range of wind speeds in our data. The actual time span was chosen, because for at least one wind turbine there are no manual or automatic alarms or services during this period (cp. section 4). Two turbines have no recorded data for this time span, effectively reducing our data set to 98 turbines.

Due to confidentiality agreements we will never show absolute values of any observable. In fact, only wind speed is shown directly and is then presented in units of the nominal wind speed $\tilde{v}_{\mathrm{nom}}$ at which the turbine starts to produce its nominal power output according to the manufacturer. The tilde is introduced to mark it as the rated value provided by the manufacturer as we will later on try to infer this value also from the data.

Each of the measured signals k for each turbine l yields a time series of data points $S_k^{(l)}(t)$, $k = 1, \ldots, K$, $l = 1, \ldots, L$, $t =$

1, ..., T_{end}. Here, we assume that all times are given as unit free steps. In the case of our dataset we have $K = 5$, $L = 98$ and - assuming complete data - $T_{\text{end}} = 20 \cdot 24 \cdot 60 \cdot 6 = 172800$. The data is arranged into L rectangular $K \times T_{\text{end}}$ data matrices

$$
S^{(l)} = \begin{bmatrix} S_1^{(l)}(1) & \dots & S_1^{(l)}(T_{\text{end}}) \\ \vdots & & \vdots \\ S_k^{(l)}(1) & \ddots & S_k^{(l)}(T_{\text{end}}) \\ \vdots & & \vdots \\ S_K^{(l)}(1) & \dots & S_K^{(l)}(T_{\text{end}}) \end{bmatrix},
\tag{1}
$$

where each row is the time series of signal k.

3 Theoretical background

Our analysis to distinguish different states is based on identifying differences in the correlation matrix of observables listed in section 2. In section 3.1 we define the way in which we are calculating the correlation matrices.

To identify non-stationarity in the time series of these matrices we will use a distance measure and a clustering algorithm. These are introduced in section 3.2.

3.1 Correlation matrices

To identify changes over time in the correlation structure, the correlation matrices are calculated with a moving time window of 30 min. The time intervals do not overlap. We effectively create a time series of matrices. To this end, the signal time series $S_k^{(l)}(t)$ are divided into disjoint intervals of 30 min, i.e. the lengths of the intervals is $T = 180$ in our dimensionless time variable. We refer to these intervals as epochs. Hence, we have $T_{end}/T = 960$ epochs. To avoid notational confusion, we introduce the new time variable τ labelling the epochs. We reserve the notation $S_k^{(l)}(t)$ for the original time series and write $S^{(l)}(\tau)$ for the $K \times T$ data matrix containing the different time series for turbine l from τ to $\tau + T - 1$. The length of 30 min represents a compromise. Longer time spans would provide more data points per correlation matrix and would thereby decrease noise. However, we want to distinguish different states in time. Considering external conditions, e.g. wind, changing on short time scales of several minutes to hours, we have to choose relatively short epochs to ensure resolution of the non-stationarity . Such compromises are common when dealing with correlation matrix time series (Marti et al., 2021; Heckens et al., 2020).

The first step towards calculating correlation matrices is the normalization of $S_k^{(l)}(t)$ to zero mean and standard deviation one in each epoch. With the mean value

$$
\mu_k^{(l)}(\tau) = \frac{1}{T} \sum_{t=\tau}^{\tau+T-1} S_k^{(l)}(t)
\tag{2}
$$

and the standard deviation

$$
\sigma_k^{(l)}(\tau) = \sqrt{\frac{1}{T} \sum_{t=\tau}^{\tau+T-1} (S_k^{(l)}(t) - \mu_k^{(l)}(\tau))^2} \;,
\tag{3}
$$

of the epoch τ, the normalized time series for signal k and turbine l is given by

$$M_k^{(l)}(t) = \frac{S_k^{(l)}(t) - \mu_k^{(l)}(\tau)}{\sigma_k^{(l)}(\tau)} \quad , \quad k = 1,\ldots,K \quad , \quad l = 1,\ldots,L \quad , \quad \tau \leq t < \tau + T \quad . \tag{4}$$

The normalized $K \times T$ data matrix $M^{(l)}(\tau)$ for each epoch is then defined analogous to $S^{(l)}(\tau)$.

The Pearson correlation matrix for each turbine l during the time $\tau \leq t < \tau + T - 1$ is then easily calculated as

$$C^{(l)}(\tau) = \frac{1}{T} M^{(l)}(\tau) M^{(l)\dagger}(\tau) \quad , \tag{5}$$

where $M^{(l)\dagger}(\tau)$ denotes the transpose of $M^{(l)}(\tau)$. In the matrix $C^{(l)}(\tau)$ each element $C_{ij}^{(l)}(\tau)$ is the Pearson correlation coefficient of the signals i and j for turbine l during the epoch from τ to $\tau + T - 1$. The diagonal values are one by definition.

3.2 Clustering

We will now introduce the clustering, which allows us to sort the correlation matrices into groups (clusters) and check, whether different typical states do exist. If we can identify these, we will refer to them as *operational states*. An integer will be assigned to each of them and the algorithm will label each matrix in the time series with one such integer. Instead of a time series of correlation matrices, we then have a new integer time series $n(\tau)$ with the range $n \in \{1,\ldots,N\}$ when N is the number of clusters created.

The first outcome will then be as follows: If any decent clustering solution can be found, it is proof that typical states of the correlation structure exist. Then, analysing the resulting integer time series $n(\tau)$ can much easier reveal dependencies of the state on time or other factors.

Any method separating objects into groups needs a distance measure defined between those objects. For the correlation matrices we choose the euclidean distance (Heckens et al., 2020). The reader can imagine that all matrix entries are written into a vector, effectively arranging the columns of the matrix underneath each other, so that the standard euclidean distance between vectors can be applied. The distance between the correlation matrices for the epochs starting at τ and τ' of turbine l is then

$$d^{(l)}(\tau,\tau') = \sqrt{\sum_{i,j}(C_{ij}^{(l)}(\tau) - C_{ij}^{(l)}(\tau'))^2} = ||C^{(l)}(\tau) - C^{(l)}(\tau')|| \quad . \tag{6}$$

We choose the bisecting k-means algorithm to perform our clustering (Steinbach et al., 2000; Tan et al., 2019). This is the algorithm that was also used to determine states in the financial markets by Münnix et al. (2012) and Heckens et al. (2020). It can be described as a hybrid of standard k-means clustering (Lloyd, 1982; Jain, 2010) and hierarchical clustering (Kaufman and Rousseeuw, 1990). While the former directly divides the whole set of objects into k groups (see appendix A), the latter is performed step-wise. In each step either two groups are merged (agglomerative) or one group is divided into two (divisive). Bisecting k-means is a divisive clustering algorithm, meaning that at the start all objects belong to one big cluster and during each step one of the existing clusters is split into two new clusters. This bisection is performed by running a standard k-means on all objects within the group to be split with $k = 2$. Which of the $\tilde{N}$ currently existing clusters is split during a step, is decided

based on the average internal distance of all objects in a cluster z_n , $n = 1, \ldots, \tilde{N}$

$$d_n^{(l)} = \frac{1}{|z_n^{(l)}|} \sum_{\tau \in z_n^{(l)}} ||C^{(l)}(\tau) - \langle C^{(l)} \rangle_n|| \ . \tag{7}$$

Here, $|z_n^{(l)}|$ denotes the number of objects in cluster $z_n^{(l)}$ and $\langle C^{(l)} \rangle_n$ is the centroid of cluster $z_n^{(l)}$ defined by the element-wise mean:

$$\langle C_{ij}^{(l)} \rangle_n = \frac{1}{|z_n^{(l)}|} \sum_{\tau \in z_n^{(l)}} C_{ij}^{(l)} \ . \tag{8}$$

Each step the cluster with the largest average internal distance is bisected. The algorithm is terminated when a set number N of clusters is reached.

This is slightly different from the approach used in former works by Münnix et al. (2012), Rinn et al. (2015), Stepanov et al. (2015) and Heckens et al. (2020), where a threshold is introduced and all clusters are bisected until no single existing cluster has an average internal distance larger than the threshold. However, the threshold is then set based on the number of clusters wanted. It can easily be understood that the resulting clustering will be the same if one either uses our approach to produce N clusters or chooses the threshold in such a way that N clusters are produced.

Applying this algorithm, we split the set $Z^{(l)} = \{C^{(l)}(1), C^{(l)}(1 + T), \ldots, C^{(l)}(T_{\text{end}} - T)\}$ of all correlation matrices into N subsets $\{z_1^{(l)}, \ldots, z_n^{(l)}, \ldots, z_N^{(l)}\}$. The centroid of each cluster according to eq. (8) is interpreted as the mean correlation matrix of a cluster representing its typical correlation structure. Thereby, we only need to look at N matrices and a series of T_{end}/T integers $n(\tau)$ instead of as many matrices.

Later on we will see that the emerging typical correlation matrices correspond to different control settings of the turbines. This explains in a simplified way, why the hierarchical k-means works better than a normal k-means in our case. Approximately, we can describe the controller of a wind turbine as a mechanism fixing certain signals to a fixed value. This means the correlation of that signal with other non-fixed signals should vanish. The divisive clustering will first extract a group where signal A might be fixed. Then this group might be further divided into subgroups where signal B is either fixed or not. And in theory this could go on. Such a problem is very well suited for divisive clustering.

In order to check, if our clustering is sensible, we will do two things. Firstly, we will just look at the cluster centroids and see, if we can interpret them and if they are substantially different from each other. Secondly, we will calculate silhouette coefficients (Rousseeuw, 1987)

$$s^{(l)}(\tau) = \begin{cases} \frac{b(\tau) - a(\tau)}{\max(b(\tau), a(\tau))} & , |z_{n(\tau)}| > 1 \\ 0 & , |z_{n(\tau)}| = 1 \end{cases} \tag{9}$$

with the average distance to all other matrices in the same cluster

$$a(\tau) = \frac{1}{|z_{n(\tau)}| - 1} \sum_{\tau' \in z_{n(\tau)}, \tau' \neq \tau} d^{(l)}(\tau, \tau') \tag{10}$$

and the smallest average distance to a single other cluster

$$b(\tau) = \min_{m \neq n(\tau)} \left(\frac{1}{|z_m|} \sum_{\tau' \in z_m} d^{(l)}(\tau, \tau') \right) . \tag{11}$$

210 This coefficient will take values between -1 and $+1$ with larger positive values representing matrices that are well clustered and negative values showing matrices that are closer to another cluster than to their own. To get an indicator for the overall clustering we will use the average silhouette coefficient

$$S^{(l)} = \frac{T}{T_{\text{end}}} \sum_\tau s^{(l)}(\tau) . \tag{12}$$

4 State identification via clustering

215 In the following, we analyse the correlation matrix time series of one turbine, which will henceforth be referred to as turbine 1 (WT1). We are singling out this wind turbine, because the time frame of the total dataset was selected in such a way that for WT1 no problems were listed in the automatic alarm logs and manual service reports. The idea is that this will make analysis and definition of normal states easier as there was no (reported) unusual behaviour.

The correlation matrices are calculated for non-overlapping epochs of 30 minutes each. This results in 960 matrices per turbine.
220 However, due to several reasons there might be missing data in the time series. In such a case, any time stamp missing one or more of the measured variables is excluded from the calculation of the correlation matrix. Hereby, no estimation of values, which could influence the actual correlation coefficient, is necessary. More data could be used by calculating the correlation coefficients pair-wise, i.e. for any two observables just remove the time stamps where one of them is missing data. However, this does not result in a well defined positive semi-definite correlation matrix. For further analysis only those matrices, for which
225 at least half of the expected data points (90 out of 180) exist, are considered. This is not a problem as missing measurements usually stem from turbines being operational but switched off during times of very low wind speeds v smaller than turn-on wind speed v_{on}. For WT1 the average wind speed for 746 epochs with enough data is $10.01 \ \text{ms}^{-1}$, while the average for 214 epochs where no correlation matrix could be calculated is only $4.34 \ \text{ms}^{-1}$. It is obvious that these times where a wind turbine is not operating are unsuited for an identification of operational states. Of course, there might also be other reasons causing the
230 missing data, e.g. a problem with the measurement of a signal. However, as for WT1 there are no alarms or services logged, we would not know what happened in those cases anyway. Any estimation of missing values would therefore need considerable guessing. As our results show that using only the epochs with enough data points is sufficient to reach a good differentiation of operational states, we are confident that just excluding missing values instead of estimating them is a good approach for our purposes.

235 When applying the hierarchical k-means algorithm described in the previous section to the set of matrices, the first step is to decide how many clusters provide a good solution. To this end, we calculate the silhouette coefficients for solutions with 2-5 clusters. The silhouette plots can be found in fig. 1. The fifth cluster is almost imperceptible as it consists only of 3 matrices. Some descriptive statistics for these silhouette coefficients are shown in table 1. The *mean* corresponds to the average silhouette
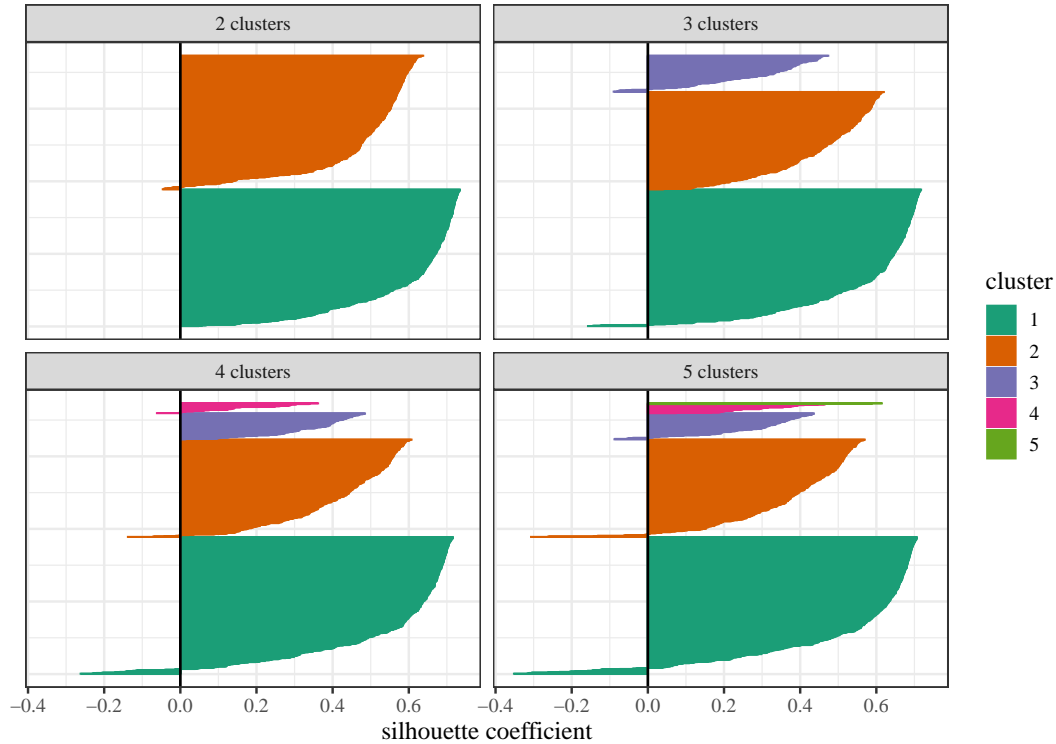
Figure 1. Silhouette plots for clustering solutions with 2-5 clusters. Each clustered element (matrix) is represented by a horizontal line the length of which is the silhouette coefficient for that element. Different clusters are colour coded.

Table 1. Minimum, first quartile, median, mean, third quartile and maximum of silhouette coefficients for the clustering solutions with 2-5 clusters for correlation matrices of WT1.

clusters	min	1st Qu.	median	mean	3rd Qu.	max
2	-0.046	0.473	0.571	0.540	0.664	0.734
3	-0.157	0.379	0.537	0.491	0.640	0.718
4	-0.262	0.343	0.508	0.465	0.631	0.716
5	-0.352	0.314	0.479	0.439	0.626	0.707

coefficient from eq. (12). All statistics provided decrease with increasing cluster number, implying that a few different states
240 are sufficient to describe the dynamics of the analysed correlation matrices. In the plots we can see some negative coefficients
implying elements that would fit better into a different cluster. Such imperfections are to be expected when using heuristics like
clustering algorithms. It is however clear, that all solutions provide a good grouping with largely positive silhouette coefficients.
This is a strong indication that the correlation matrix is not stationary, rather different states can be detected via clustering.
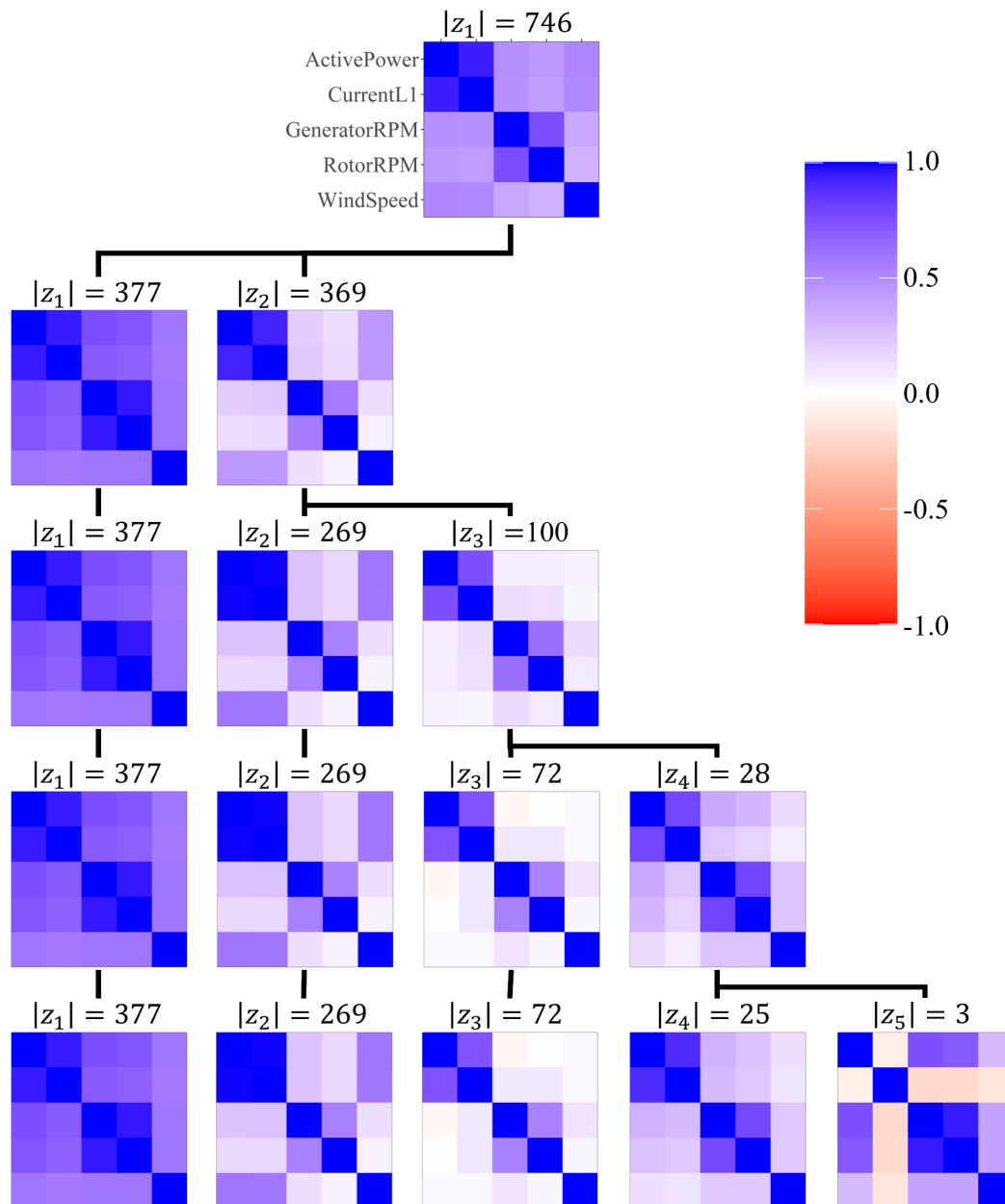
Figure 2. Cluster centroids as calculated in eq. (8) for WT1 for different numbers of clusters. The colour indicates the value of the correlation coefficient. Black lines connect child and parent clusters of the hierarchical algorithm and the number of cluster elements is given as $|z_n|$.

As mentioned before, we also look at the cluster centroids to see if the matrices show indeed different behaviour and if this

245 distinction facilitates clear identification. Figure 2 shows the matrices calculated via eq. (8) in a dendogram for the hierarchical clustering. The solutions for two and three clusters show distinctly different structures in the matrices, whereas the fourth cluster stems from cluster three, but is structurally very similar to cluster one, only differing in the strength of the mean correlation. The introduction of a fifth cluster only produces a very small cluster with only three elements. The algorithm does not identify new groups, but rather starts to classify outliers. While the average silhouette coefficient favours two clusters, we continue our

250 analysis with three clusters as we have seen that up to this point structural differences in the matrices occur and we will later see that these can be interpreted very reasonably. Here, we also point out that structural differences in the matrix have a stronger influence on the structure of the eigenvectors, i.e. principal components, than differences in average correlation strength. They are therefore more important to distinguish when using methods like PCA and Mahalanobis distance.

The classification of the matrices for three clusters is shown as an integer time series in fig. 3. All three states appear to have a

255 certain stability. Consecutive epochs often belong to the same cluster. However, there is no obvious behaviour in dependency of the time. State 3 appears far less often than states 1 and 2. To get a better idea what each state might represent, we look at the matrices for the cluster centroids calculated according to eq. (8) once more. They are seen in the third row of fig. 2. Generally, as the differences between the matrices are quite clear we can conclude - in accordance with the silhouette coefficient - that the clustering does indeed separate the matrix time series into meaningful groups. The correlation matrices are non-stationary.

260 In every cluster the strongest correlations are clearly visible between the observable pairs ActivePower-CurrentL1 and RotorRPM-GeneratorRPM. This was to be expected as these pairs are very directly linked. Apart from this we can see that for cluster 1 both of these pairs and the WindSpeed are all correlated with each other. Put differently, higher wind speed leads to faster rotation and thus to higher power. In cluster 2 this changes and the observables RotorRPM and GeneratorRPM decouple from the others. Cross-correlations between these two and any other measurement vanish. The remaining cross-correlation between

265 WindSpeed and the two observables ActivePower and CurrentL1 also vanish in cluster 3. Clearly, the three different states of the correlation structure identified via clustering are meaningful: They show distinctly different correlation behaviour between different observables.

To interpret the meaning of the clustering solution, it is helpful to look at turbine control systems. The basic functionality of such a system is, for example, described by Schütt (2014). The specific functionality varies for individual turbine types, so it

270 is likely that not all turbine types will show the same operational states. The turbine control system of the *Vestas* V90 turbines analysed in this paper is one with variable pitch (*Vestas OptiTip*™) and variable speed (*Vestas OptiSpeed*™). We can connect the three clusters to different operational states of this turbine type, which are seperated by boundary wind speeds v_{on}, v_1, v_2 and v_{nom}. For very low wind speeds just above the turn-on wind speed $v_{\text{on}} \leq v < v_1$ the generator rotation is kept constant at the lowest possible value defined by the maximum slip in the generator. This results in a correlation structure as seen in cluster

275 2. Already for slightly higher wind speeds $v_1 \leq v < v_2$ the system controls the rpm proportional to the wind speed to operate at maximum aerodynamic efficiency of the rotor. This corresponds to cluster 1. With even more wind, but still not enough to reach nominal power output $v_2 \leq v < v_{\text{nom}}$ the turbine operates at fixed nominal rpm by controlling the torque. The rotational variables decouple again as for very low wind speeds and the correlation structure corresponds to cluster 2 again. Of course,

fluctuations in wind speed will cause the rotation to fluctuate around the nominal value leading to some noise in the correlation
280 structure. If wind speed is high enough to allow full power output $v_{\mathrm{nom}} \leq v$ the nominal power output is reached and therefore
kept constant alongside the rpm. This results in correlations as seen in cluster 3. All boundary wind speeds $v_{\mathrm{on}}, v_1, v_2, v_{\mathrm{nom}}$ are
turbine dependent and usually not public knowledge.
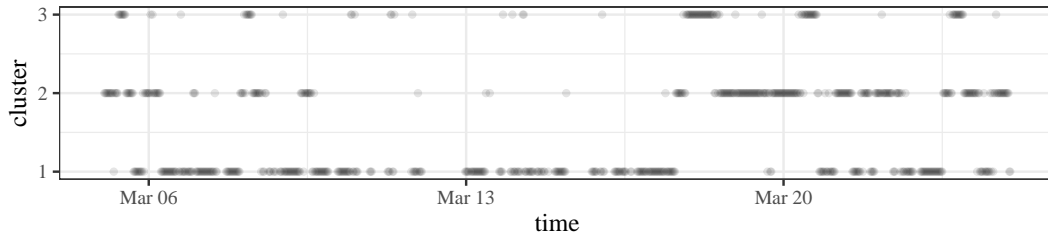
Figure 3. Cluster identifier n over time for WT1. Each dot represents a 30 min epoch.

We can see that our reasoning for the turbine at hand is correct by plotting the cluster state over the mean wind speed of the
epoch instead of the time stamp in fig. 4. For now disregarding the interval $v_{\mathrm{on}} \leq v < v_1$ due to lack of enough data, cluster 1
285 represents low wind speeds, cluster 2 intermediate wind speeds and cluster 3 high wind speeds, where nominal power output
can be reached. Some exceptions are to be expected due to either wrong sorting during the clustering or wind speeds changing
during the 30 min epoch. Also, the response time of a wind turbine controller is finite in the range of seconds or minutes and
therefore the correlation structure does not respond instantly to fluctuations and changes in wind speed (Behnken et al., 2020).
One example are the epochs sorted into cluster 3 whose wind speeds seem to lie in the range of cluster 1. They are all moved
290 into the fourth cluster if we take another step in the algorithm. Its centroid is shown in the dendogram fig. 2 and exhibits a
structure very close to cluster 1. Such mismatches occur due to the heuristic nature of the algorithm and noise and fluctuations
in the data, which results in matrices lying on the edge between two clusters. We will see in the following section that we can
use the silhouette coefficient to identify them. The dependency on wind speed is also in accordance with the stability in time
as periods with stable wind speeds are common (Weber et al., 2019).
295 We have seen that the correlation matrix is non-stationary in time. The clustering has revealed a primary influence of the control
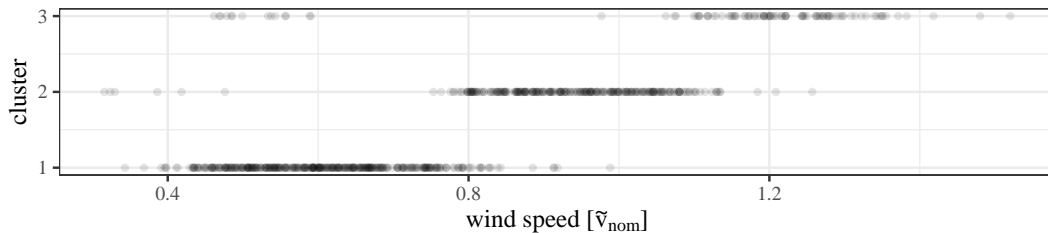
Figure 4. Cluster identifier n over wind speed for WT1. Each dot represents a 30 min epoch.
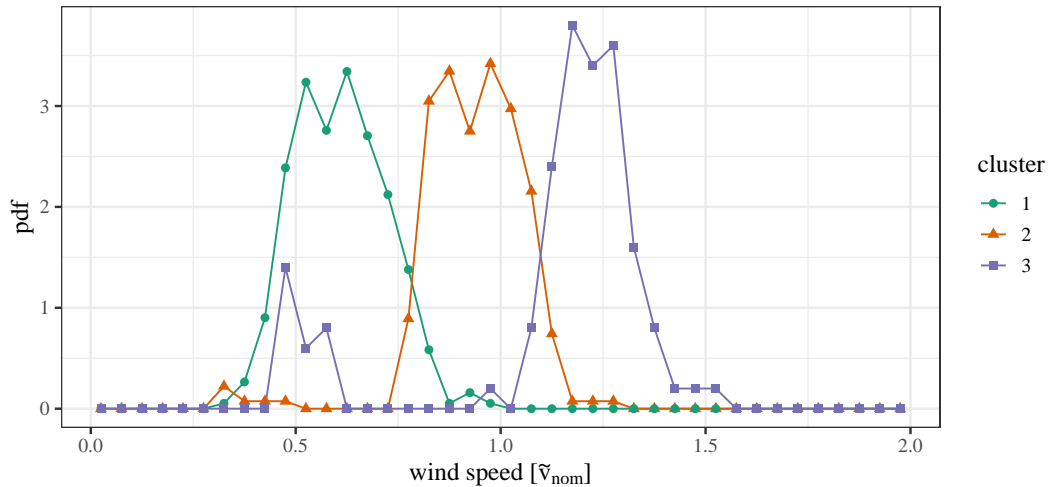
Figure 5. Probability density functions for the 30 min epoch mean wind speed per cluster. The width of bins is $0.05\tilde{v}_{\mathrm{nom}}$.

strategies in dependence of the wind speed. Further analysis of matrices with more and different variables as well as an attempt to distinguish other influencing factors could be attempted in the future.

5 Cluster prediction by wind speed

Having established a strong influence of the control system onto the structure of the analysed correlation matrices, we will now
300 try to predict which correlation matrix state, i.e. operational state, the turbine should be in based on the wind speed. Being able to do this is crucial to use different states for failure prediction. New (live) data can be matched to the expected normal state based on wind speed.

We look at the distribution of wind speeds in the different states, analyse them and then predict the boundary wind speeds that separate the distinct groups.

305 In fig. 5 one can see the empirical probability density functions (pdf) for wind speeds per cluster state. As already expected from fig. 4, we can clearly distinguish the different regimes. However, we identify much more clearly what we are calling mismatches: epochs that are sorted into cluster 3, but have mean wind speeds in the range associated with cluster 1. They make up the left peak of the distribution for cluster 3. Furthermore we can see a small peak in the probability density function for cluster 2 lying at very low wind speeds beneath the distribution for cluster 1. These could be reasonable as the rotation of the
310 generator shaft is kept at a minimum rotational speed needed for operation of the turbine for very low wind speeds as discussed in the previous section. However, the data in the very low wind regime is sparse and not as reliable as the turbines often move in and out of operation during these times due to shutting off below a certain minimal wind speed, therefore we will disregard this first boundary v_1 for now.

Before modelling the boundaries between the distributions, we try to compensate for mismatches due to matrices lying at
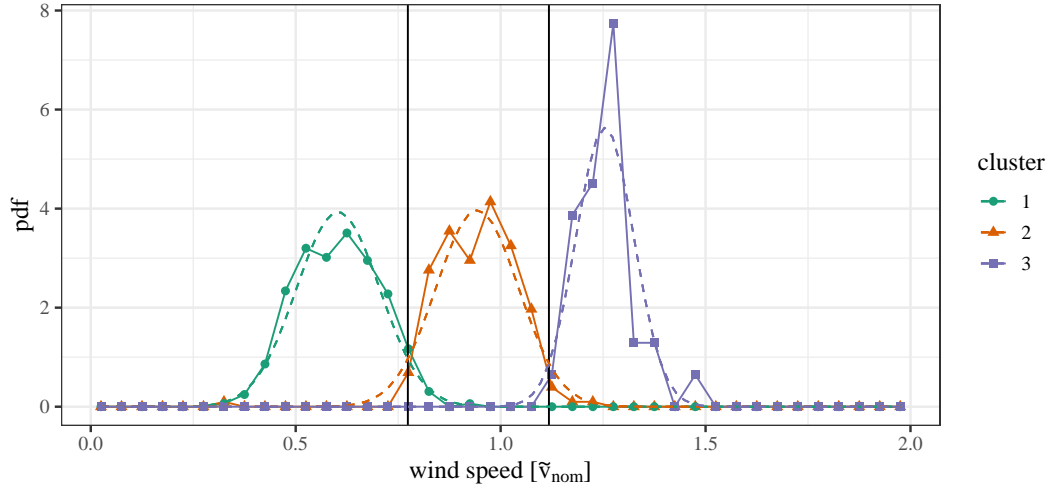
Figure 6. Probability density functions for the 30 min epoch mean wind speed per cluster using only epochs with a silhouette coefficient above the first quartile of all silhouette coefficients. For each cluster a normal distribution was fitted. Then black vertical lines indicate the intersections of these distributions and thereby the boundary wind speeds. The width of bins is $0.05\tilde{v}_{\mathrm{nom}}$.

315 the edge of two clusters, matrices being wrongly sorted, or singular outliers. This can easily be done by using the silhouette coefficient we have introduced before. It gives an indication of how good a member of a cluster fits into this cluster compared to the other clusters. This means that any 30 min epoch that has been sorted into cluster i but should rather be in cluster j will have a very small or negative silhouette coefficient. We can use this fact and remove from the calculation of the probability density function all epochs with a silhouette coefficient below the first quartile of all silhouette coefficients, which can be seen

320 in table 1. The resulting probability density functions can be seen in fig. 6. The second peak at low wind speeds for cluster 3 disappears. This indicates that our reasoning of a mismatch was correct. The persistence of the small peak at very low wind speeds for cluster 2 on the other hand shows that it indeed points toward a control of the rotational speeds in this regime.

The empirical distributions are noisy due to the finite amount of data points. This is especially true for cluster 3, which contains the least epochs. However, it is very clear that every cluster is representing a wind speed regime. There are now two

325 basic approaches to defining the boundaries between these regimes. One can simply look at the empirical data and define for each value of wind speed the maximum likelihood state based on the empirical probability density function. Secondly, one can fit a distribution to the data and calculate the intersections of these, which represent the boundaries. For now, we choose to fit distributions as it turns out that a normal distribution is a good choice for each wind speed regime (see fig. 6) and the other method could be heavily influenced by noisiness in the empirical data. Of course, this will dismiss the smaller

330 peak of cluster 2. It should be taken into account if enough data exists in this regime (cp. section 6). For now, cluster 1 has a mean of $0.603\tilde{v}_{\mathrm{nom}}$ and a standard deviation of $0.101\tilde{v}_{\mathrm{nom}}$. Clusters 2 and 3 are centred at $0.943\tilde{v}_{\mathrm{nom}}$ and $1.255\tilde{v}_{\mathrm{nom}}$ with standard deviations of $0.101\tilde{v}_{\mathrm{nom}}$ and $0.071\tilde{v}_{\mathrm{nom}}$ respectively. These values lead to boundaries at $v_2 = 0.774\tilde{v}_{\mathrm{nom}}$ and $v_{\mathrm{nom}} = 1.118\tilde{v}_{\mathrm{nom}}$. Interestingly, the last value shows that v_{nom} as calculated in our analysis is larger than $\tilde{v}_{\mathrm{nom}}$. We want
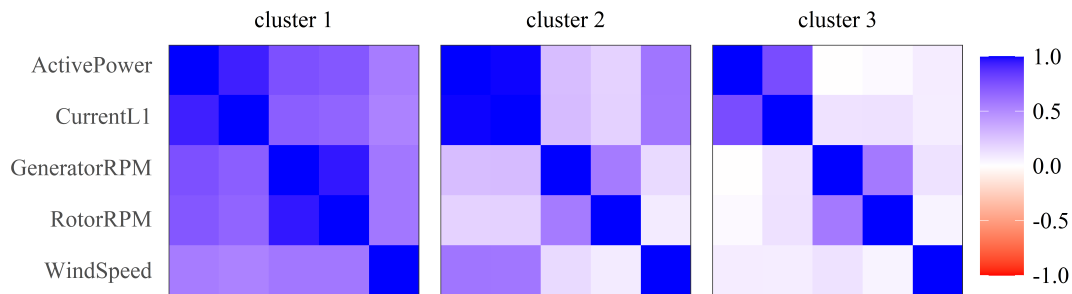
Figure 7. Matrices corresponding to the group centroids after sorting with the epochs according to the calculated boundaries for WT1. The mean matrices were calculated for all epochs, not only those used to determine the boundaries. The colour indicates the value of the correlation coefficient.

to point out that it is not proven that a normal distribution will always provide the best fit. For example, if much larger wind
335 speeds exist in the data set, the distribution for cluster 3 would have much heavier tails in the large wind speed regime.

In the current case for WT1 the model fitted works very well. Compared to the clustering solution we have 9.9% of changes in group assignment. If one only looks at epochs with silhouettes above the first quartile this number reduces to 3.7%. This is clear as the epochs previously characterized as mismatches obviously change their cluster allocation when applying the model to all epochs. The mean correlation matrices of the states as sorted by the model are shown in fig. 7. They clearly exhibit the
340 different structures discussed above produced by the control system of the turbine showing that our state prediction works.

6   Application to multiple turbines

For a single wind turbine we identified different operational states in the correlation structure and presented a model to distinguish these states based on wind speed. To be useful for failure analysis and prediction, our findings need to be general characteristics and not be specific for one turbine. We proceed and test our methodology for all turbines in the data set. We want to emphasize this does not mean applying the previously fitted model to all turbines, rather performing a cluster analysis and fitting the boundaries for each turbine separately.

An easily comparable indicator for the quality of the proposed methodology is the relative numbers of cluster allocation changes from the model compared to the clustering itself. We already discussed that for WT1 at the end of the previous section. This number will drastically increase if either of the two steps in the calculation does not work well on a turbine: If the clustering algorithm returns a solution that is not grouped by wind speed, sorting on the basis of wind speed will change the allocation of many epochs. If they are clustered according to wind speed, but the boundaries are less sharp than for WT1, it will again result in many changed allocations.
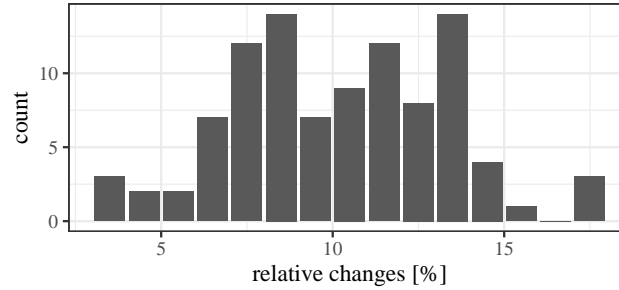
Figure 8. Histogram counts showing the frequency of relative changes in state allocation when comparing clustering and individual models per turbine.
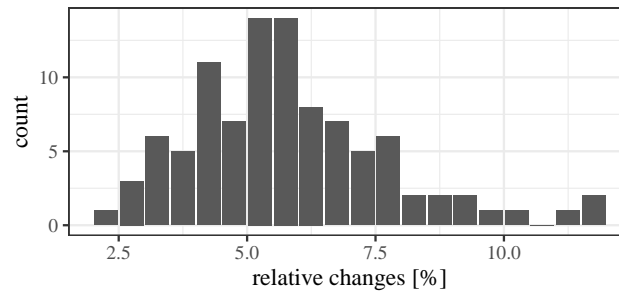
Figure 9. Histogram counts showing the frequency of relative changes in state allocation per turbine when comparing the clustering solution and the maximum likelihood model based on data from all turbines.

A histogram over all calculated allocation changes for the 98 turbines is shown in fig. 8. We can see that the changes for WT1 lie in the lower end of changes, as expected due to it not showing any alarms or failures in the chosen time span. This does not hold true for the other turbines, most of which exhibit a few more changes. However, there are multiple turbines which show no more changes than WT1. Also, for those that do the fitted boundaries still work remarkably well. Some allocation changes are always expected for the reasons discussed above and especially near the boundaries the distinction between two states is not always perfect. Concluding that our proposed method works well for all turbines in our dataset, we proceed with an optimization.

Looking at multiple turbines it stands to reason that the probability density functions for wind speeds per cluster should be much smoother than for one turbine as we have, in our case, approximately 98 times as many data points. This is indeed so as can be seen in fig. 10. Furthermore, we can see that the previously assumed Gaussian fit would not work well anymore. Especially the distribution shown for cluster 1 is skewed. Also, the peak in the distribution for cluster 2 beneath wind speeds of $0.4\tilde{v}_{\mathrm{nom}}$ becomes more explicit. This second point is explained in sections 4 and 5 and actually underlines our reasoning: For very low wind speeds rotation is kept constant and with more data from all turbines we can distinguish this regime more clearly than before.

At a first glance, the skewness of the probability density functions does not fit our theory so well. If the controller would work perfectly and instantly and the wind speeds were constant during each epoch, we would expect the distinctions between the operational states to be represented by rectangular functions as probability densities. In non-perfect conditions these would overlap and smooth out to be something similar to a gaussian distribution as assumed before, but they should not become skewed. However, the reason can be found in the underlying distribution of wind speeds in the inlay in fig. 10. It follows roughly the expected Weibull distribution. The deviations could be explained by combinations of influences in the environment of the wind farm, overlying of different Weibull distributions for different wind directions or maybe even measurement effects due to the sensor being placed behind the rotor. Some differences might also be introduced by the removal of low silhouette coefficients as these will appear often in the regimes of the boundaries between states where two correlation structures might be mixed during an epoch. This is of interest for future studies. For now, we can take away that the skewness of this underlying distributions might lead to the skewness in the cluster probability density functions. To check this we replace the histogram count of epochs $h_i(v_w)$ for wind speed v_w and cluster i with the rescaled count

$$\tilde{h}_i(v_w) = \frac{h_i(v_w)}{h_{\text{total}}(v_w)}$$

by dividing with the total histogram count of epochs $h_{\text{total}}(v_w)$ for that wind speed and all clusters. This basically removes the effect of the underlying wind speed distribution by transforming it into an equal distribution. The resulting probability density functions for each cluster can be seen in fig. 11. Indeed, we can now see symmetric areas, showing behaviour very similar to rectangular functions for cluster 1 and 3 with cluster 2 being smoothed out to a more Gaussian curve, because its wind regime is quite narrowly bounded by overlaps with the other states. This strongly underlines the existence of three regimes corresponding to operational states of the turbines.

As the functions for all turbines are much smoother and the bin size can be reduced, we can apply the direct maximum likelihood method instead of fitting a continuous curve to decide cluster allocation based on the epoch wind speed. This leads to three instead of the previous two boundary wind speeds to account for the appearance of cluster 2 in the very low wind regime. The values of the boundaries are $v_1 = 0.38\tilde{v}_{\text{nom}}$, $v_2 = 0.80\tilde{v}_{\text{nom}}$ and $v_{\text{nom}} = 1.10\tilde{v}_{\text{nom}}$. The resulting histogram of changes due to model allocation compared to the clustering is presented in fig. 9 and shows less changes compared to fig. 8. One reason for this is the taking into account of the very low wind speed regime in cluster 2. Such a method without need for fitting could be easily transferred to other data and turbine types providing high usability for failure prediction.

Overall we conclude that the results formerly shown for WT1 are easily transferred to multiple wind turbines. Furthermore, the model to decide state allocation based on wind speed can be optimized by taking into account more turbines and thereby more data.

7   Conclusions

Using a matrix distance measure and clustering algorithm formerly applied to other complex systems we were able to identify different operational states in 30 min correlation matrices of high frequency wind turbine data without prior knowledge of
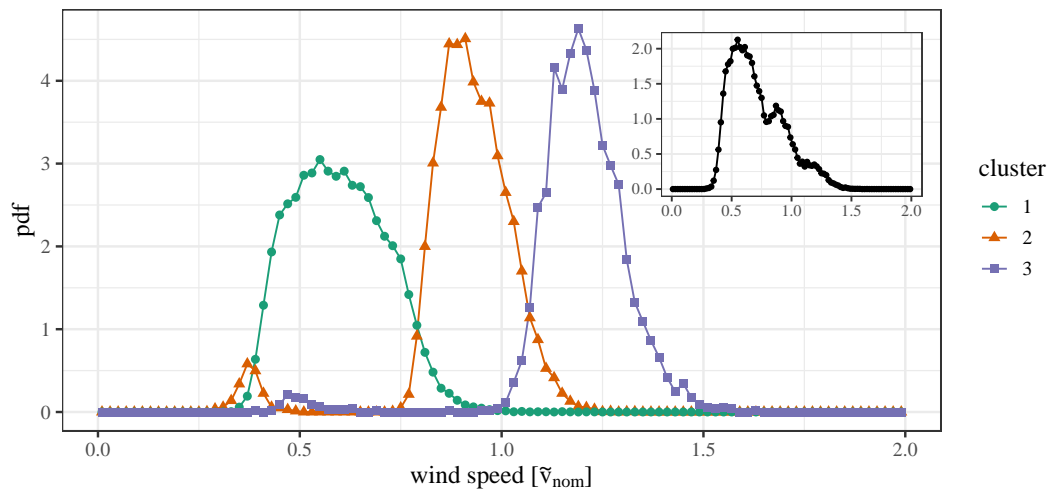
Figure 10. Probability density functions for the 30 min epoch mean wind speed per cluster considering all turbines. Only those epochs with a silhouette coefficient above the first quartile of all silhouette coefficients for each turbine were used (calculated seperately per turbine). The underlying wind speed distribution without cluster seperation is shown as inlay. The width of bins is $0.02\tilde{v}_{\mathrm{nom}}$.
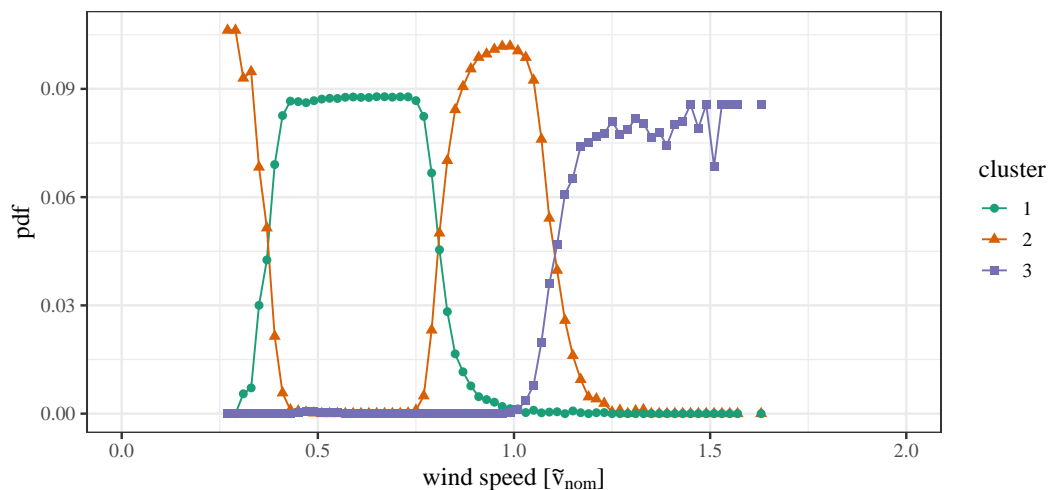
Figure 11. Probability density functions for the 30 min epoch mean wind speed per cluster considering all turbines after dividing by the total number of counts per wind speeds to rescale the underlying distribution to an equal one. Only those epochs with a silhouette coefficient above the first quartile of all silhouette coefficients for each turbine were used (calculated seperately per turbine). The width of bins is $0.02\tilde{v}_{\mathrm{nom}}$.

the control system. This demonstrates the non-stationarity of the correlation matrix of wind turbines. While the states quite often exhibit stability over a certain time period, the real dependency was identified to lie with wind speed. Furthermore, it was possible to model the boundary wind speeds between the different states - again without knowledge about the actual parameters used in the control system. This allowed us to recreate the cluster allocation solely based on the 30 min average wind speeds.

365 Being developed on one turbine, the method transferred easily onto multiple turbines. Results were even improved by an increased amount of data.

Our method enables the definition of multiple operational states for wind turbines, whereby an optimization for failure analysis and prediction could be undertaken. New or live data could be compared to the respective operational states based on wind speed after modelling the boundary wind speeds with historical data. While all techniques used for failure analysis might

370 benefit from a more confined parameter space when using multiple normal states compared to just one normal behaviour, those techniques relying on the correlation matrix - such as principal component analysis or Mahalanobis distance - are likely to benefit most. Charmingly, the proposed ansatz does not require changes in established techniques, it just requires their application to multiple subgroups and is therefore easily implemented.

We aim to quantify the possible improvement in future studies using SCADA-data with labelled failure events. This will also

375 allow training of the model based on all intervals without failures instead of a continuous time span thereby increasing the amount of available data per turbine. Furthermore, it will then be possible to study long-term non-stationarity indicated by Jin et al. (2021) where failure prediction works better if the training period of the model is a moving window. These are all interesting aspects to study in future works.

Alongside this, we will analyse the non-stationarity for other wind turbine types and different observables.

380 Our study so far has made it quite clear that the assumption of stationarity in the correlation structure of wind turbine SCADA-data is questionable and non-stationarity effects have to be considered when developing failure analysis and prediction methods.

Data availability. The data used four our calculations were provided by *Vattenfall AB* as part of the government funded research project *Wind farm virtual Site Assistant for O&M decision support – advanced methods for big data analysis* (WiSAbigdata). The legally binding project cooperation agreement prohibits the disclosure of data to any party outside the project.

385 **Appendix A: Standard k-means clustering**

The standard k-means algorithm sorts every object O_i, $i = 1, \ldots, N$ from a set of N objects into k subsets $\{z_1, \ldots, z_l, \ldots, z_k\}$. Every subset is called a cluster. Generally, the optimal number k has to be determined by a separate method Jain (2010); Kaufman and Rousseeuw (1990).

The input for the algorithm are the objects O_i, $i = 1, \ldots, N$ and a distance measure $d(O_i, O_j) \geq 0$, $i, j = 1, \ldots, N$ as well as a

390 method to compute the centroid of any cluster z_l. Note, that the distance measure must also be defined for the centroids. Then the algorithm works as follows:

1. Select k objects as starting cluster centroids.

2. Assign every object to the nearest cluster based on the distance from the object to the cluster centroid.

3. Calculate the new cluster centroids.

395   4. Repeat steps 2 and 3 until no allocation changes occur.

In this paper the objects are the correlation matrices of the subset we wish to split into two in the hierarchical approach, therefore $k = 2$ always. The cluster centroids are calculated according to eq. (8) and the distance is calculated by eq. (6).

Author contributions. TG initiated the research. HMB and TG defined the objectives. HMB developed the tools and conducted the data analysis with support from EJ. HMB wrote the first draft of the manuscript. EJ and TG contributed to the writing of the final version.

400   *Competing interests.* The authors declare that they have no conflict of interest.

Acknowledgements. We are most grateful to *Vattenfall AB* for providing the data. We acknowledge fruitful conversations with Prof. Dr. J. Peinke, Dr. M. Wächter, C. Phillipp and Dr. T. Lichtenstein. This study was carried out in the project *Wind farm virtual Site Assistant for O&M decision support – advanced methods for big data analysis* (WiSAbigdata) funded by the Federal Ministry of Economics Affairs and Energy, Germany (BMWi). One of us (H.M.B.) thanks for financial support in this project.

References

Aziz, U., Charbonnier, S., Bérenguer, C., Lebranchu, A., and Prevost, F.: Critical comparison of power-based wind turbine fault-detection methods using a realistic framework for SCADA data simulation, Renewable and Sustainable Energy Reviews, 144, 110 961, https://doi.org/10.1016/J.RSER.2021.110961, 2021.

Behnken, C., Wächter, M., and Peinke, J.: Multipoint reconstruction of wind speeds, Wind Energy Science, 5, 1211–1223, https://doi.org/10.5194/WES-5-1211-2020, 2020.

Braun, T., Waechter, M., Peinke, J., and Guhr, T.: Correlated power time series of individual wind turbines: A data driven model approach, Journal of Renewable and Sustainable Energy, 12, 023 301, https://doi.org/10.1063/1.5139039, 2020.

Buccheri, G., Marmi, S., and Mantegna, R. N.: Evolution of correlation structure of industrial indices of US equity markets, Phys. Rev. E, 88, 012 806, https://doi.org/10.1103/PhysRevE.88.012806, 2013.

Chen, W., Qiu, Y., Feng, Y., Li, Y., and Kusiak, A.: Diagnosis of wind turbine faults with transfer learning algorithms, Renewable Energy, 163, 2053–2067, https://doi.org/10.1016/j.renene.2020.10.121, 2021.

Colone, L., Reder, M., Dimitrov, N., and Straub, D.: Assessing the Utility of Early Warning Systems for Detecting Failures in Major Wind Turbine Components, Journal of Physics: Conference Series, 1037, https://doi.org/10.1088/1742-6596/1037/3/032005, 2018.

Daga, A. P., Fasana, A., Garibaldi, L., and Marchesiello, S.: On the use of PCA for Diagnostics via Novelty Detection: Inter-pretation , Practical Application Notes and Recommendation for Use, in: PHM Society European Conference, vol. 5(1), p. 13, https://doi.org/10.36001/phme.2020.v5i1.1241, 2020.

Dao, C. D., Kazemtabrizi, B., Crabtree, C. J., and Tavner, P. J.: Integrated condition-based maintenance modelling and optimisation for offshore wind turbines, Wind Energy, pp. 1–19, https://doi.org/10.1002/we.2625, 2021.

Denis, D. J.: Principal components analysis, in: Applied Univariate, Bivariate, and Multivariate Statistics, chap. 13, pp. 423–448, John Wiley & Sons, Ltd., Hoboken, 2 edn., https://doi.org/10.1002/9781119583004.ch13, 2021.

Gottschall, J. and Peinke, J.: How to improve the estimation of power curves for wind turbines, Environmental Research Letters, 3, 015 005, https://doi.org/10.1088/1748-9326/3/1/015005, 2008.

Heckens, A. J., Krause, S. M., and Guhr, T.: Uncovering the dynamics of correlation structures relative to the collective market motion, Journal of Statistical Mechanics: Theory and Experiment, 2020, 103 402, https://doi.org/10.1088/1742-5468/abb6e2, 2020.

Helbing, G. and Ritter, M.: Deep Learning for fault detection in wind turbines, Renewable and Sustainable Energy Reviews, 98, 189–198, https://doi.org/10.1016/j.rser.2018.09.012, 2018.

Jain, A. K.: Data clustering: 50 years beyond K-means, Pattern Recognition Letters, 31, 651–666, https://doi.org/10.1016/j.patrec.2009.09.011, 2010.

Jin, X., Xu, Z., and Qiao, W.: Condition monitoring of wind turbine generators using SCADA data analysis, IEEE Transactions on Sustainable Energy, 12, 202–210, https://doi.org/10.1109/TSTE.2020.2989220, 2021.

Kaufman, L. and Rousseeuw, P. J.: Finding Groups in Data, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., Hoboken, NJ, USA, https://doi.org/10.1002/9780470316801, 1990.

Kolumbán, S., Kapodistria, S., and Nooraee, N.: Short and long-term wind turbine power output prediction, https://arxiv.org/abs/1707.06497, 2017.

Lee, J. and Zhao, F.: Global Offshore Wind: Annual Market Report 2020, Tech. rep., Global Wind Energy Council, https://gwec.net/global-offshore-wind-report-2020/, 2020.

Lee, J. and Zhao, F.: Global Wind Report 2021, Tech. rep., Global Wind Energy Council, https://gwec.net/global-wind-report-2021/, 2021.

Li, Y., Peng, S., Li, Y., and Jiang, W.: A review of condition-based maintenance: Its prognostic and operational aspects, Frontiers of Engineering Management, 7, 323–334, https://doi.org/10.1007/s42524-020-0121-5, 2020.

445 Lloyd, S. P.: Least Squares Quantization in PCM, IEEE Transactions on Information Theory, 28, 129–137, https://doi.org/10.1109/TIT.1982.1056489, 1982.

Maldonado-Correa, J., Martín-Martínez, S., Artigao, E., and Gómez-Lázaro, E.: Using SCADA data for wind turbine condition monitoring: A systematic literature review, Energies, 13, 3132, https://doi.org/10.3390/en13123132, 2020.

Marti, G., Nielsen, F., Bińkowski, M., and Donnat, P.: A review of two decades of correlations, hierarchies, networks and clustering in
450 financial markets, in: Progress in Information Geometry. Signals and Communication Technology, edited by Nielsen, F., pp. 245–274, Springer Cham, https://doi.org/10.1007/978-3-030-65459-7_10, 2021.

Martin-del Campo, S. and Al-Kahwati, K.: Unsupervised Ranking of Outliers in Wind Turbines via Isolation Forest with Dictionary Learning, in: PHM Society European Conference, vol. 5(1), p. 9, https://doi.org/10.36001/phme.2020.v5i1.1164, 2020.

Meyer, A. and Brodbeck, B.: Data-driven Performance Fault Detection in Commercial Wind Turbines, PHM Society European Conference,
455 5, 7, https://doi.org/10.36001/phme.2020.v5i1.1276, 2020.

Münnix, M. C., Shimada, T., Schäfer, R., Leyvraz, F., Seligman, T. H., Guhr, T., and Stanley, H. E.: Identifying states of a financial market, Scientific Reports, 2, 644, https://doi.org/10.1038/srep00644, 2012.

Odgaard, P. F. and Johnson, K. E.: Wind turbine fault detection and fault tolerant control - An enhanced benchmark challenge, in: 2013 American Control Conference, pp. 4447–4452, IEEE, https://doi.org/10.1109/acc.2013.6580525, 2013.

460 Pandit, R. K. and Infield, D.: SCADA-based wind turbine anomaly detection using Gaussian process models for wind turbine condition monitoring purposes, IET Renewable Power Generation, 12, 1249–1255, https://doi.org/10.1049/iet-rpg.2018.0156, 2018.

Pozo, F., Vidal, Y., and Salgado, Ó.: Wind turbine condition monitoring strategy through multiway PCA and multivariate inference, Energies, 11, 749, https://doi.org/10.3390/en11040749, 2018.

Rinn, P., Stepanov, Y., Peinke, J., Guhr, T., and Schäfer, R.: Dynamics of quasi-stationary systems: Finance as an example, Europhysics
465 Letters, 110, 68 003, https://doi.org/10.1209/0295-5075/110/68003, 2015.

Rousseeuw, P. J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics, 20, 53–65, https://doi.org/10.1016/0377-0427(87)90125-7, 1987.

Ruiz de la Hermosa González-Carrato, R.: Wind farm monitoring using Mahalanobis distance and fuzzy clustering, Renewable Energy, 123, 526–540, https://doi.org/10.1016/j.renene.2018.02.097, 2018.

470 Schütt, R. J.: Control of Wind Energy Systems, in: Understanding Wind Power Technology: Theory, Deployment and Optimisation, vol. 9781118647, pp. 340–368, Wiley Blackwell, https://doi.org/10.1002/9781118701492.ch9, 2014.

Siemens: Predictive Services, https://www.youtube.com/watch?v=oAnNvbGTQFQ, 2021.

Steinbach, M., Karypis, G., and Kumar, V.: A Comparison of Document Clustering Techniques, in: KDD Workshop on Text Mining, 2000.

Stepanov, Y., Rinn, P., Guhr, T., Peinke, J., and Schäfer, R.: Stability and hierarchy of quasi-stationary states: Financial markets as an example,
475 Journal of Statistical Mechanics: Theory and Experiment, 2015, P08 011, https://doi.org/10.1088/1742-5468/2015/08/P08011, 2015.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J., and Nenadic, G.: Machine learning methods for wind turbine condition monitoring: A review, Renewable Energy, 133, 620–635, https://doi.org/10.1016/j.renene.2018.10.047, 2019.

Tan, P.-N., Steinbach, M., Karpatne, A., and Kumar, V.: Introduction to Data Mining (2nd Edition), Pearson Education, New York, 2019.

Tautz-Weinert, J. and Watson, S. J.: Using SCADA data for wind turbine condition monitoring - A review, IET Renewable Power Generation, 11, 382–394, https://doi.org/10.1049/iet-rpg.2016.0248, 2017.

Tavner, P.: Offshore wind turbines: reliability, availability and maintenance, Institution of Engineering and Technology, London, https://doi.org/10.1049/PBRN013E, 2012.

Ulmer, M., Jarlskog, E., Pizza, G., Manninen, J., and Huber, L. G.: Early Fault Detection Based on Wind Turbine SCADA Data Using Convolutional Neural Networks, in: PHM Society European Conference, vol. 5(1), p. 9, https://doi.org/10.36001/phme.2020.v5i1.1217, 2020.

Veers, P., Dykes, K., Lantz, E., Barth, S., Bottasso, C. L., Carlson, O., Clifton, A., Green, J., Green, P., Holttinen, H., Laird, D., Lehtomäki, V., Lundquist, J. K., Manwell, J., Marquis, M., Meneveau, C., Moriarty, P., Munduate, X., Muskulus, M., Naughton, J., Pao, L., Paquette, J., Peinke, J., Robertson, A., Sanz Rodrigo, J., Sempreviva, A. M., Smith, J. C., Tuohy, A., and Wiser, R.: Grand challenges in the science of wind energy, Science, 366, eaau2027, https://doi.org/10.1126/science.aau2027, 2019.

Vidal, Y., Pozo, F., and Tutivén, C.: Wind turbine multi-fault detection and classification based on SCADA data, Energies, 11, 3018, https://doi.org/10.3390/en11113018, 2018.

Wang, Z., Yao, L., Cai, Y., and Zhang, J.: Mahalanobis semi-supervised mapping and beetle antennae search based support vector machine for wind turbine rolling bearings fault diagnosis, Renewable Energy, 155, 1312–1327, https://doi.org/10.1016/j.renene.2020.04.041, 2020.

Weber, J., Reyers, M., Beck, C., Timme, M., Pinto, J. G., Witthaut, D., and Schäfer, B.: Wind Power Persistence Characterized by Super-statistics, Scientific Reports, 9, 19 971, https://doi.org/10.1038/s41598-019-56286-1, 2019.

Yan, H., Mu, H., Yi, X., Yang, Y., and Chen, G.: Fault Diagnosis of Wind Turbine Based on PCA and GSA-SVM, 2019 Prognostics and System Health Management Conference (PHM-Paris), pp. 13–17, https://doi.org/10.1109/PHM-Paris.2019.00010, 2019.

Zhao, Y., Li, D., Dong, A., Kang, D., Lv, Q., and Shang, L.: Fault prediction and diagnosis of wind turbine generators using SCADA data, Energies, 10, 1210, https://doi.org/10.3390/en10081210, 2017.