

Article

Dynamic Clustering of Wind Turbines Using SCADA Signal Analysis

Pere Marti-Puig *  and Carles Núñez-Vilaplana 

Data and Signal Processing Group, University of Vic—Central University of Catalonia,
08500 Vic, Catalonia, Spain; carles.nunez@uvic.cat

* Correspondence: pere.marti@uvic.cat

Abstract: This work explores the ability to dynamically group the Wind Turbine (WT) of a Wind Farm (WF) based on the behavior of some of their Supervisory Control And Data Acquisition (SCADA) signals to detect the turbines that exhibit abnormal behavior. This study is centered on a small WF of five WTs and uses the observation that the same signals from different WTs in the same WF coherently evolve temporally in a time domain, describing very similar waveforms. In this contribution, averaged signals from the SCADA system are used and omit maximums, minimums and standard deviations, focusing mainly on velocities and other slowly varying signals. For the temporal analysis, sliding windows of different temporal durations are explored. The signals are encoded using the Discrete Cosine Transform, which reduces the problem's dimensions. A hierarchical tree is built in each time window. Clusters are formed by pruning the tree using a threshold interpretable in terms of distance. It is unnecessary to work with an a priori known number of clusters. A protocol for enumerating the clusters based on the tree's shape is then established, making it easier to follow the evolution of the clusters over time. The capability to automatically identify WTs whose signals differ from the group's behavior can alert and program preventive maintenance operations on such WTs before a major breakdown occurs.

Keywords: hierarchical clustering (HC); dynamic clustering (DC); wind turbine (WT); SCADA data



Citation: Marti-Puig, P.; Núñez-Vilaplana, C. Dynamic Clustering of Wind Turbines Using SCADA Signal Analysis. *Energies* **2024**, *17*, 2514. <https://doi.org/10.3390/en17112514>

Academic Editor: Davide Astolfi

Received: 30 April 2024

Revised: 21 May 2024

Accepted: 22 May 2024

Published: 23 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The urgent need to transition to more sustainable energy sources directly responds to the challenges imposed by climate change and the growing global energy demand. In this scenario, wind power stands out as a promising alternative, showing remarkable technological development and capacity expansion in recent decades. Thus, it has established itself as one of the primary renewable energy sources, with increasing acceptance by society and strong support from international public policies.

WFs, made up of sets of WTs, play an essential role in transforming wind into electrical energy. The efficient management of these parks is critical not only to ensure optimal energy production but also to extend the useful life of the equipment and minimize periods of inactivity. An effective preventive maintenance system, which proactively identifies potential failures before they become major problems, is key to achieving these goals [1–3].

The present research proposes dynamically applying the Hierarchical Cluster (HC) method to SCADA data to better understand and manage turbines' operational behavior. This approach is novel because it applies clustering based on the shape of signals considered in consecutive time windows.

Groupings of WT sets with similar characteristics in the WF will allow the WF operators to understand their operation and improve their management and maintenance. Specifically, this work explores the ability to group WTs according to criteria of similarity of specific signals since this would allow the subsystems of the machines that generate them to be compared and behaviors similar to detect possible irregularities prematurely.

This can be of particular relevance when monitoring, for example, temperatures at critical points of the WT. In the context of WTs, this task presents specific difficulties since the behavior of WTs is highly nonlinear and varies enormously over time. The WTs are subject to important behavior changes due to wind fluctuation and other meteorological changes. Consequently, the variations of the shapes of the collected signals over time, or equivalently, the characteristics that represent them, also vary enormously.

As observed and proposed in [4,5], a detailed analysis of the variability and continuity of the collected signals and clustering WTs appropriately will facilitate the early detection of anomalies and the planning of maintenance interventions [6].

In the literature, clustering techniques have been used differently in the context of WTs. Liu et al.'s (2014) article [7] explores the application of WT clustering methods to improve short-term wind-power forecasts. These authors develop and validate different clustering techniques to categorize turbines based on similar performance characteristics, enabling more accurate prediction models. This research proposes using clustering to optimize energy management strategies, resulting in more accurate and efficient power generation predictions. The article [8] addresses the development of a dynamic cluster equivalent model for WTs based on the use of spanning trees. This approach allows the clustering of WTs dynamically to improve the efficiency and representativeness of WF simulation models. Such an approach uses spanning tree techniques to identify and represent the most critical connections between turbines, thus facilitating the creation of simplified yet effective equivalent models. This methodology offers a significant advancement in modeling the collective behavior of WTs, contributing to improving WF planning and operation. The article [4] presents a method for diagnosing and warning of faults in WTs using cluster analysis and a modified version of the Adaptive Neuro-Fuzzy Inference System (ANFIS). The article [9] presents an advanced methodology for early fault detection in WTs, combining operational condition clustering with optimized deep belief network modeling. This approach segments WT operations into different sub-conditions, facilitating more effective detection of possible anomalies and improving fault detection accuracy by effectively handling the WTs' nonlinear and heterogeneous operational data. The article [5] explores a new strategy for fault detection in WTs through SCADA data clustering. This methodology groups WTs based on similarities in operational data to enhance fault detection and diagnosis through comparative analysis. Finally, advanced data visualization techniques are implemented to represent the information and facilitate its graphic interpretation. This study seeks to provide a clear and detailed understanding of the dynamics within the WF using scatter plots, time plots, and other graphical tools. The HC process is then applied to identify groupings of data that will share similar properties, revealing meaningful patterns that can inform maintenance and operation decisions [7,8]. This approach improves WFs' operational performance and provides a replicable and scalable methodology for data analysis in other industrial applications, expanding companies' ability to adapt to technological and market changes.

In the context of the clustering of WTs for condition monitoring, the present work focuses on the problem of dynamic clustering based on the shape of specific signals, which is a new approach to the problem. Focusing on signals, clustering is intended to locate the potential degradation of a particular subsystem based on one signal or group of signals showing anomalies. Such capability is desirable for condition monitoring of WTs and their preventive maintenance.

As the operating conditions of WTs are very variable and variations occur on different time scales, another very critical point is that the number of clusters is variable in time and unknown a priori. For example, all the WTs in the WF often stop working due to the lack of wind, and suddenly, the recorded SCADA temperatures of all the WTs tend to converge to the ambient temperature, falling in a single cluster. Therefore, one of the requirements for the algorithm is that it does not depend on the number of clusters, K .

The popular K-means [10] and K-medoids [11] clustering techniques organize data into K mutually exclusive clusters, requiring the number K to be known a priori. The

Gaussian mixture models (GMM) [12] represent normally distributed subpopulations within an overall population and are appropriate when clusters have different sizes and different correlation structures within them. The parameters of GMM are the mixture component weights and the distributions' means and variances/covariances. Moreover, GMM can be performed *soft* by assigning the observation to multiple clusters based on the scores or posterior probabilities of the observation for the clusters. However, because of its need to know K to adjust the parameters, it does not suit the present application's needs.

Therefore, a signal of all WT in a WF is analyzed fragment-by-fragment in time windows called frames. The clustering of the signals of each WT in every frame is obtained. Clustering, considering the shapes of those signals in a temporal frame, is updated with every new frame. The detailed methodology first involves a comprehensive review and selection of the most impactful variables, such as air temperature, relative humidity, wind speed, and generated power. The next step involves a meticulous data cleaning process, where the inputs are selected and filtered to remove outliers or erroneous values. This task is crucial to maintaining the integrity of the analytical model [9,13] but is very common, so it is not focused on very much. The database employed for this work contains SCADA signals that provide data every 5 min, thus collecting 288 points daily. More relevant in our approach is that to compact the information, the Discrete Cosine Transform (DCT) is used, and only the first coefficients are considered, which are organized into vectors of features to represent the signals frame to frame. The experiments reveal that a few coefficients (3 to 5) already synthesize the information very effectively. The representative vectors of each signal are used to calculate the distances (using the Euclidean distance) between each pair of signals. Then, a binary, agglomerative, hierarchical clustering tree is built from those distances [14]. In this step, the objects (signals) are linked in pairs according to proximity, building a series of nested clusters, where the most similar elements are grouped first, and differences are incorporated as one descends the hierarchy. In agglomerative clustering, each element starts as an independent cluster and progressively merges into larger clusters based on similarity. This hierarchical tree (HT) provides an intuitive view of data structure, allowing exploration of different levels of detail in organizing elements, which is helpful in classification and pattern exploration in complex datasets because the HT can be cut to form the clusters at any particular point independently of a pre-established number of clusters [15]. Finally, a way to name the clusters is developed so that the signals with the most similarity appear in the first cluster, and as they differ more, they appear in higher clusters. This nomenclature facilitates, at least in a small WF, the monitoring of the temporal evolution of the clusters.

Notice that the proposed method works only with the SCADA averages. When working with real-world data continuously over time, specifically with SCADA data, errors are inevitable due to failures in the sensors that collect them or errors in communications. In addition, there is always noise. Statistical measurements taken by the SCADA system are typically provided every 10 min. The biggest errors in these measurements are particularly collected in the *max* and *min* statistical operations, which capture the extreme values. For this reason, *max* and *min* are less reliable SCADA measurements. Averages work as a low-pass filter and smooth out noise and errors, making them more trustworthy.

Averages also eliminate fast variations and high-frequency components that could be present in these signals but still preserve the major characteristics of the signal shape. Standard deviations provide valuable information that could be used, although the best way to take advantage of them in this context must be investigated. The low-frequency rate of SCADA data acquisition presents a significant limitation that can hinder diagnostic capabilities, particularly in detecting short-duration events. Directly identifying abnormal vibration of damaged mechanical components or anomalous electrical behavior of faulty electrical elements is challenging, with data averaging every 10 min, as discussed in studies such as [16,17]. It should be noted that condition monitoring (CM) methods based on SCADA data typically focus on detecting secondary effects of faults [18]. SCADA-based

CM methods often identify incoming faults through abnormal conditions, such as the heating or underperformance of WTs.

The SCADA system's averaging of slow-varying signals every 10 min still preserves the main information, and it does not have the devastating effect it has on fast-varying signals. For instance, practically half the SCADA system's magnitudes are temperatures, which fall in this slow-varying signal category and are taken at many points in WT's subsystems.

This work will be organized into *Materials and Methods* (Section 2) where detailed descriptions are provided of the following contents: the data used, the shape parameterization based on the DCT coefficients, a description of the hierarchical clustering employed, and the protocol to name the clusters that permits the following of the temporal evolution of clusters. Then, Section 3, *Results*, explains the interpretation of the dynamical clustering graphs and contains two experiments. The first is a study of the wind speeds recorded in the WTs' nacelles through these clustering techniques. The second is a comparative study of applying this technique to two control variables commonly used for WT prognoses, such as the rotation speed of the generator shaft and the temperature of the oil in the gearbox. Section 4, *Discussion*, deals with the main parameters of the algorithm, the limitations and some future research to improve the method, and some comparisons with other clustering techniques that require the knowledge of the number of clusters to run. Finally, the main *Conclusions* are summarized in Section 5.

2. Materials and Methods

2.1. Data Used

The present study thoroughly analyzes SCADA records of five 2.5 MW Fuhrländer FL2500 wind turbines for three years and a sampling frequency of 5 min. The system has IEC 61400-25 as its standard communication protocol for transmitting data from the wind turbines and storing it in a MySQL database. This database includes 312 analog variables from 78 different sensors. Thus, the status of various essential components, such as the transmission, generator, and converter, among many others, can be known. The data are extracted from the open-access database available at <https://github.com/alecuba16/fuhrlander>, accessed on 21 May 2024, and it is described in [19].

2.2. Shape Parameterization Through the DCT

The DCT will be exploited as a tool for parameterizing relatively long signals into a few parameters to compress their information. Unlike other transforms, such as the Fourier Transform, which yields complex-valued coefficients, DCT produces real-valued ones, simplifying signal processing.

To present it, let us consider the set of N points x_n , and their N DCT (of type-II) transformed coefficients y_k . The forward and backward expressions take the form:

$$y_k = \sum_{n=0}^{N-1} c_k x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right); \quad k = 0, \dots, N-1 \quad (1)$$

and,

$$x_n = \sum_{k=0}^{N-1} c_k y_k \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right); \quad n = 0, \dots, N-1. \quad (2)$$

where $c_k = \sqrt{\frac{1}{N}}$ for $k = 0$ and $\sqrt{\frac{2}{N}}$ for $k \neq 0$.

As is well known, one of DCT's key features is its ability to concentrate most of the signal energy into a few coefficients. Thus, a relatively small number of coefficients can capture much of the signal's information, making it an efficient representation for compression purposes. In many applications, such as image and video compression, DCT is applied to small blocks of the signal rather than the entire signal. This block-based processing allows for parallelization and efficient implementation. DCT, like other discrete transforms, also has fast algorithms that are being computed very efficiently. Additionally,

DCT has an inverse transform that reconstructs the original signal from its DCT coefficients. This property is essential for applications where compression is used, as it facilitates decompression to retrieve the original signal.

The compaction properties the DCT presents in the first transform coefficients will concentrate the shape characteristics of the time series of length N in a few parameters. Therefore, in the transformed domain, the N points of the signals are characterized by the L first transformed coefficients of their DCTs, where L will be much smaller than N . It is interesting to check the reconstruction capacity of only the $L = 2, 4$, or 6 DCT coefficients to reconstruct a sequence of 128 points from the following reconstruction formula.

$$\hat{x}_n = \sum_{k=0}^{L-1} c_k y_k \cos\left(\frac{\pi}{N} \left(n + \frac{1}{2}\right) k\right); \quad n = 0, \dots, N-1. \quad (3)$$

where $\hat{x}_n$ are the reconstructed samples from the first L DCT coefficients. Figure 1 shows some reconstructions for an original bloc signal of 128 points by $L = 2, 4$, and 6 DCT coefficients.

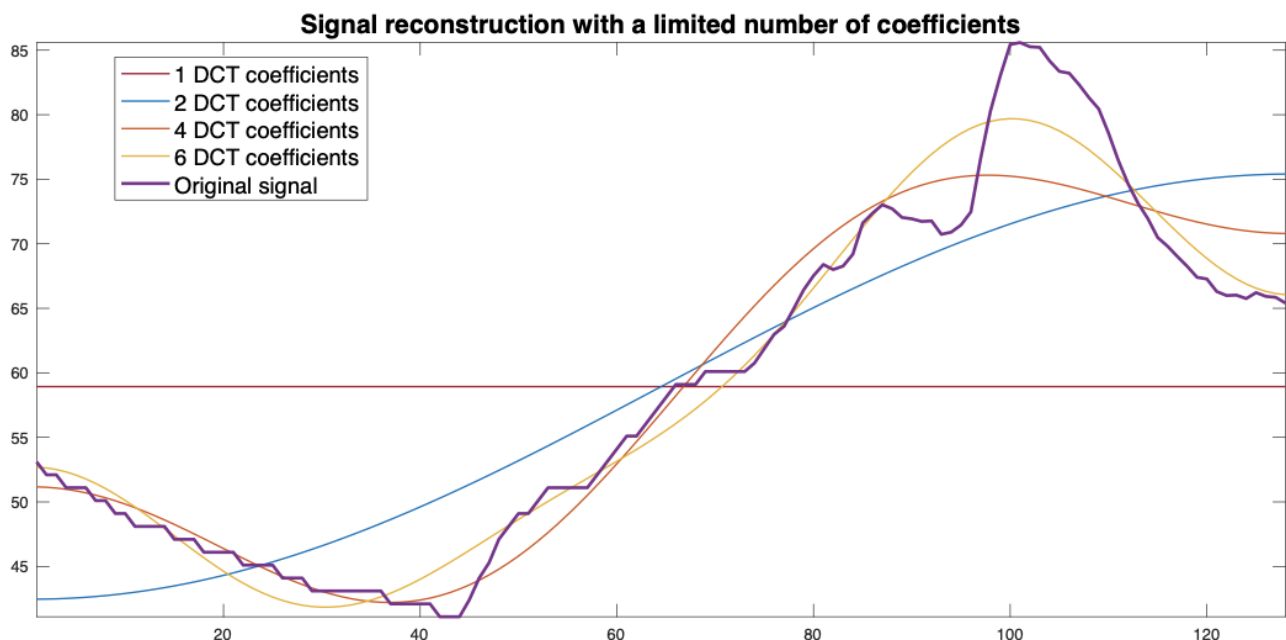


Figure 1. The graph shows the reconstruction between an original signal and approximations of it using different amounts of Discrete Cosine Transform (DCT) coefficients (from $L = 1, 2, 4$, and 6 DCT coefficients). The horizontal axis represents the time, while the vertical axis represents the signal's amplitude.

Notice in Figure 1 that the DCT concentrates energy in the first coefficients, meaning the initial coefficients capture significantly more of the original information than the latter ones. For instance, reconstructing the original signal using only the first coefficient results in a horizontal line at the mean signal value, which can be observed from, first, Equation (1) taking $k = 0$ and then from Equation (3) taking $L = 1$. Each additional coefficient incorporated into the calculation adds detail to the reconstruction.

Notice that by organizing the elements x_n and y_n of (1) in the vectors $\mathbf{x}$ and $\mathbf{y}$, the DCT can be written in matrix form as: $\mathbf{y} = \mathbf{C}\mathbf{x}$, with their elements c_{kn} taking the form:

$$c_{kn} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } k = 0 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{\pi}{N} \left(n + \frac{1}{2}\right) k\right) & \text{if } k > 0 \end{cases} \quad (4)$$

The $N \times N$ matrix $\mathbf{C}$ is unitary. Because their column vectors are orthogonal, it is fulfilled that $\mathbf{C}^{-1} = \mathbf{C}^T$. That is why the backward expression in (2) can be expressed as $\mathbf{x} = \mathbf{C}^{-1}\mathbf{y} = \mathbf{C}^T\mathbf{y}$, and the reconstructions as: $\hat{\mathbf{x}} = \mathbf{C}^{-1}\bar{\mathbf{y}} = \mathbf{C}^T\bar{\mathbf{y}}$, being $\bar{\mathbf{y}} = [y_0 \cdots y_{L-1} 0 \cdots 0]^T$.

2.3. Hierarchical Clustering

HC is a technique used in data mining and statistics to group similar data points into clusters based on their characteristics. It creates a hierarchical structure of clusters, where clusters at higher levels of the hierarchy contain fewer data points but represent broader similarities. In comparison, clusters at lower levels are more specific and may include individual data points.

There are two main types of HC: agglomerative and divisive. In agglomerative HC, each data point starts as its cluster. At each step, the two most similar clusters are merged until only one cluster remains, forming an HT-like structure. Divisive HC, on the other hand, starts with all data points in a single cluster and recursively splits them into smaller clusters until each data point is in its cluster.

This work uses agglomerative HC analysis, which follows three main steps on a data set. The first requires computing the similarity or dissimilarity between every pair of objects in the data set by calculating the distance between objects. Distance can be computed in many different ways. Standard distance metrics include Euclidean distance, Manhattan distance, and correlation-based distances, among many others. Once a distance metric is selected, the first task is to compute all the distances between all pairs of objects. Then, the distances between objects allow them to be grouped into a binary HC tree. Therefore, the second step consists of linking pairs of objects nearby using the distance information according to their proximity. As objects are paired into binary clusters, the newly formed clusters are grouped into larger clusters until an HT is formed. The third step is determining where to cut the HT to form the final clusters. This involves pruning branches off the bottom and assigning all the objects below each cut to a single cluster.

Once the HC is complete, dendrograms are often used to visualize the hierarchical structure of clusters. A dendrogram is a tree-like diagram that illustrates the order in which clusters are merged or split and can help identify the optimal number of clusters based on this structure.

HC is extremely useful in our application because it does not require specifying the number of clusters beforehand. However, it can be computationally intensive for large datasets, as it requires storing the entire dataset and computing pairwise distances between data points.

In Figure 2, the most essential parts of this process are shown. In the upper graphic, the 5 WT signals to be classified according to their shape are represented, so, in this particular case, the first three coefficients of the DCT are used to parametrize them. It is noted that each WT's signal is represented in a particular color, which is maintained in all the representations. The graph below displays the original signals' reconstructions based solely on these three coefficients. In this case, the original signals consist of 128 points, corresponding to almost 11 h. Based on vectors of only 3 components, distances between signals (objects) are calculated, and the HC dendrogram is constructed (shown in the figure below on the left). In the dendrogram, the threshold used to form clusters, a distance of 60, is also depicted to observe how the two clusters form. The figure below on the right presents the result, indicating that within the analyzed time interval and according to the threshold utilized, four signals are classified together due to similarity, while the remaining signal falls into a second cluster.

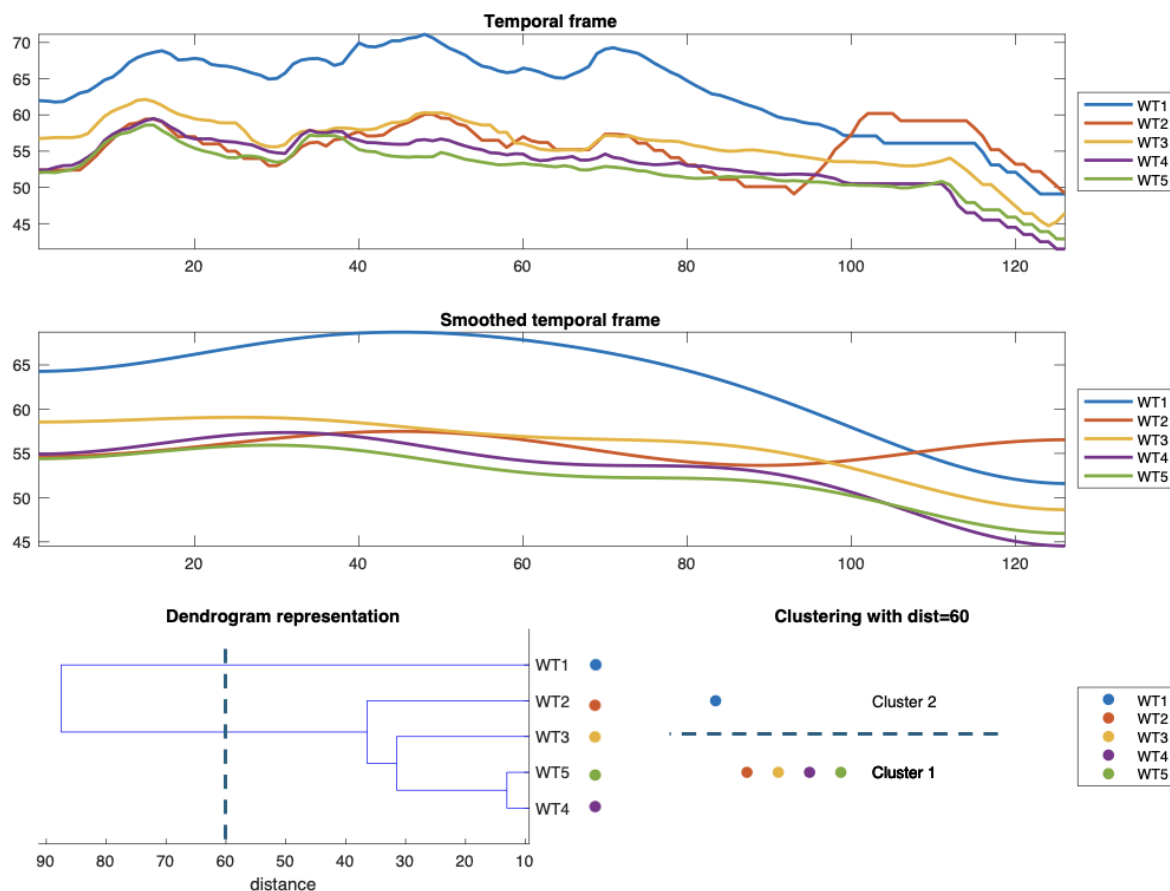


Figure 2. Representation of the main steps of clustering 5 temperature signals of 128 points using a vector of 3 DCT coefficients. Above are the original signs, and just below are reconstructions of the original signals based on just these three coefficients. On the bottom left is the HC represented as a dendrogram, and on the bottom right is the result of the clustering when using a threshold of 60.

Notably, in Figure 2, the 128 points are reconstructed using only 3 DCT coefficients.

2.4. Dynamic Evolution of Clusters and Cluster Nomenclature Protocol

A good visualization of the temporal evolution of the clusters over time in this type of problem is considered challenging. Because the signals, and therefore the feature vectors representing them, can vary significantly over time, the hierarchical trees (HTs) built before cluster formation also undergo considerable variation. Although the signals from the same WTs tend to fall into the same clusters, these cluster assignments can change from frame to frame. This means that, for instance, in frames $k - 1$, k , and $k + 1$, the cluster containing signals A, B, and C may be labeled as Clusters 1, 2, and 3, respectively, complicating the dynamic monitoring of the clusters, even in a small park like the one under consideration. Therefore, even if the clusters are well formed, the fact that Cluster 1 with signals 1 and 2 changes its name in the next frame to become Cluster 3 (also with signals 1 and 2) can make dynamic system monitoring difficult. A cluster nomenclature protocol has been developed based on the distance of the cluster in the HT, as represented in the dendrogram of Figure 3, so that Cluster 1 will be the one with the lowest distance in its highest node, Cluster 2 the next one, and so on, according to such distance. In Figure 3, note that the clusters are formed based on the distance used to prune the tree, represented by a vertical red line. Once the clusters are formed, they are named, starting with the one with the lowest distance to its junction point in the dendrogram (represented by the horizontal double-headed arrows) and continuing as those distances increase. This naming protocol stabilizes cluster names frame by frame, making them much easier to track. Figure 4 illustrates the disordered

numbering of the clusters caused by the significant variation that the HTs (represented in the dendrograms) can present frame by frame. The index k represents time and the proposed order to facilitate tracking. Each of the WTs is identified with a particular color. The left part of the graphic, part (a), exemplifies the default arrangement, while the right part, part (b), is the proposed arrangement.

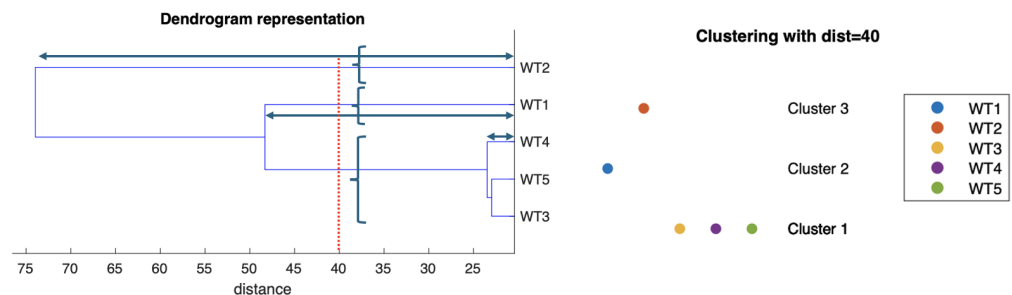


Figure 3. Once the clusters have been formed, they are sorted in increasing order of the horizontal distances represented by the two-pointed arrows. The vertical line represents the distance used to prune the HT and form the clusters.

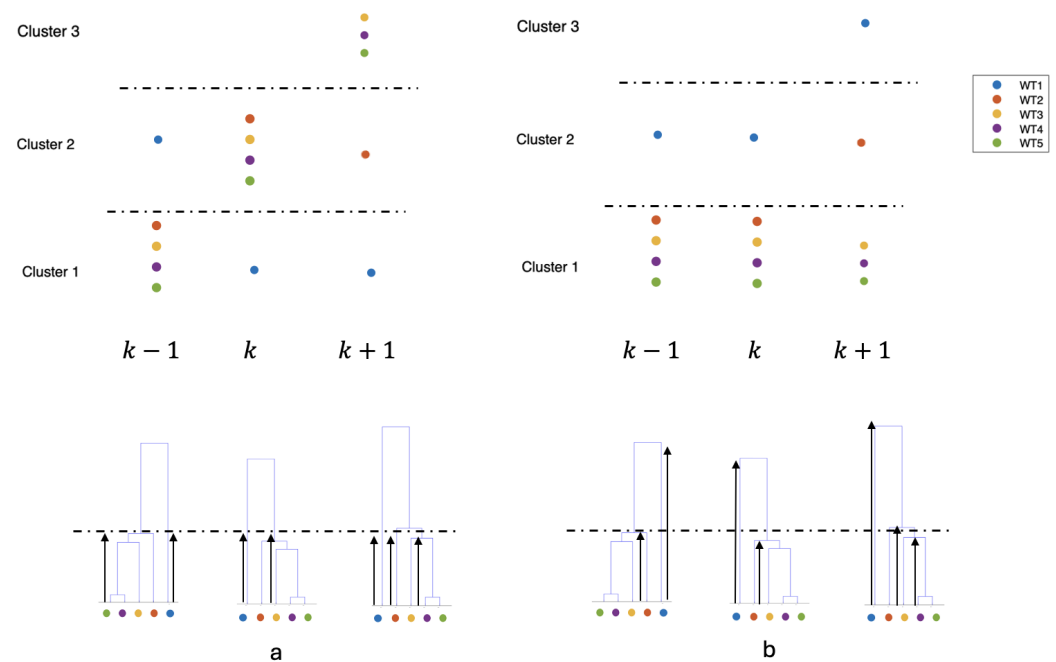


Figure 4. (a) Representation of the ordering of the clusters obtained by default after cutting the HTs. (b) The arrangement proposed, based on the distances represented by black arrows on the dendrograms, is aimed at facilitating the interpretation of the evolution of the clusters over time. It is noted that each WT is identified using a particular color.

Sorting clusters is not just for tracking over time. Once a pruning threshold has been set for the hierarchical algorithms, the WTs clustered in higher clusters are those whose signals differ more from the rest (the distance separating them from the other signals is greater). In contrast, those that remain in low clusters are much more similar. This outcome is due to the cluster nomenclature protocol illustrated in Figure 4b. According to this nomenclature, we start by assigning the number 1 to the cluster that presents the smallest distances between its WTs, Cluster 2, the cluster that requires raising the pruning distance less to fall into Cluster 1, and so on. For this reason, the most different signals of the set fall in the high clusters. The most distinct signal does not necessarily have to be related to the most damaged component. However, such clustering indicates to supervisors which components to observe more closely, which is valuable in predictive maintenance.

Figure 5 shows the process' main steps in a flow diagram.

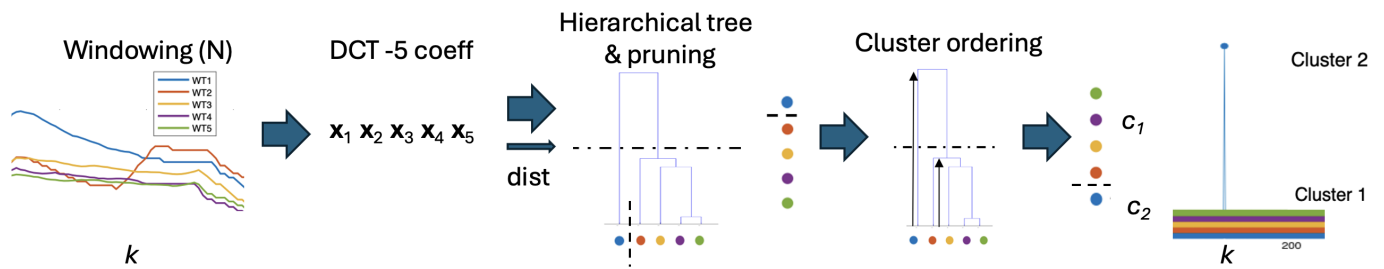


Figure 5. Flow diagram of the process's main steps. From left to right. In a time interval k , the signals to be clustered are windowed with a window of N samples. Each signal fragment is coded with a vector formed by the first five coefficients of its DCT. Pairwise distances between vectors are computed, and the hierarchical tree is constructed. This tree is pruned to a distance $dist$, and the number of clusters is determined automatically. The clusters are numbered depending on the shape of the hierarchical tree to facilitate the interpretation of the evolution of the clusters over time. The process is repeated for the next time window $k + 1$.

3. Results

3.1. Interpretation of Dynamic Clustering Graphs

The dynamic clustering procedure hinges on the configuration of three pivotal parameters. These parameters, namely the size of the temporal window, the number of DCT coefficients, and the chosen distance for HT pruning, are the backbone of the procedure, each playing a crucial role in the clustering process. As observed, the number of temporal samples taken determines the window size for signal capture and observation. In the initial test, a power of 2, $N = 128$, was chosen, corresponding to time intervals of 640 min and nearly 11 h in the analyzed park. This enabled efficient calculation of the DCT using fast algorithms, sufficient for monitoring temperature signals with slow variations occurring twice a day. Figure 1 illustrates that the number of coefficients considered influences the amount of shape information incorporated from the analyzed signals. The DCT concentrates power in the initial coefficients, as demonstrated by reconstructions with limited coefficients. Incorporating additional coefficients introduces more detail into the reconstructed shapes. However, for effective clustering, it is often unnecessary to use many coefficients; typically, 2 to 5 coefficients suffice. Nevertheless, with larger N frame sizes or rapidly varying signals, more coefficients may be beneficial despite increasing the computational cost of distance calculations. However, in our problem context, this increased dimensionality is not a significant drawback, as large parks are unlikely to have more than 300 WTs. Perhaps most crucial, the third parameter is the distance used to determine cluster membership. This distance parameter prunes the HTs and determines the number of clusters formed in each analyzed frame. Choosing a very low value results in most vectors being classified into clusters, whereas selecting a very high value tends to classify most objects into a single cluster, as significant differences are required to assign them to distinct groups. In addition to these three parameters, the type of distance used to construct the HTs could be considered; however, in these initial studies, the Euclidean distance was chosen exclusively and used consistently throughout the work.

Figures 6 and 7 show the evolution of the temperature signal *wgen_avg_GnTmp_phsC* during 100 consecutive realizations (44.44 days). In both cases, each set of 5 vertical points represents the clustering of these signals from each machine at time frame k and corresponds to 128 points (640 min). The difference between the two representations is the distance with which the clustering algorithm has been set to perform the pruning. In the case of Figure 6, the HTs are pruned at a distance of 120, meaning that to split the signals into different clusters, their vectors must have a distance greater than 120.

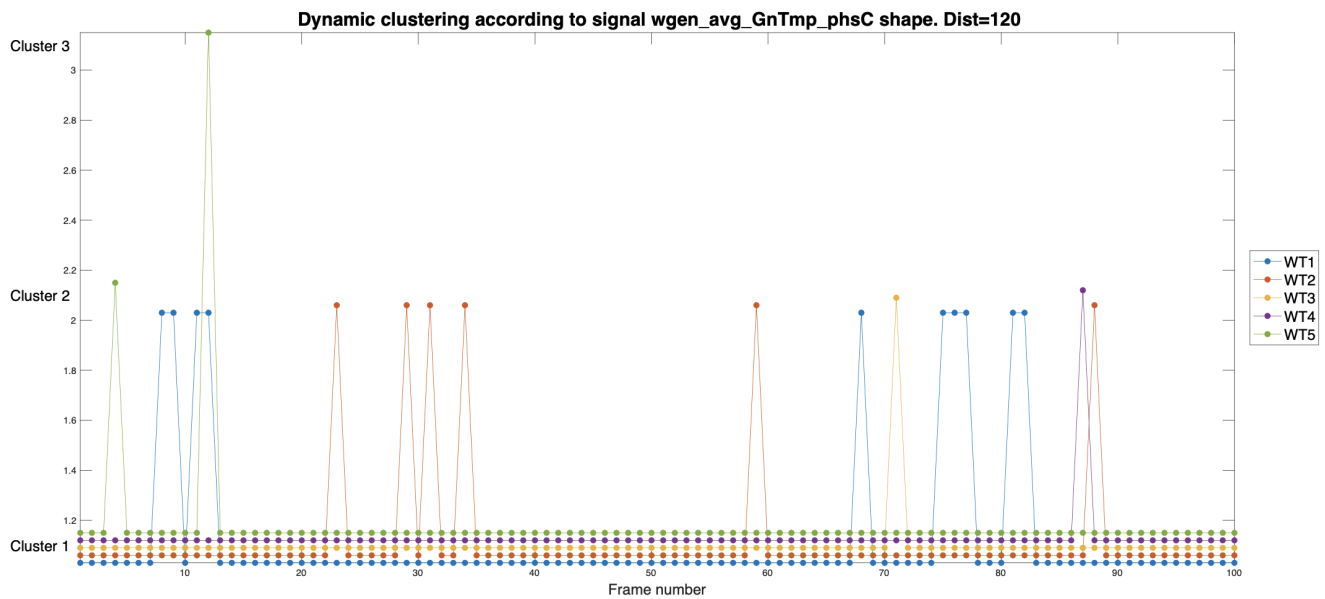


Figure 6. Cluster's dynamic evolution for the temperature signals $wgen_avg_GnTmp_phsC$. Each point represents a time window of 128 samples (640 min). The distance used to prune the HT is 120.

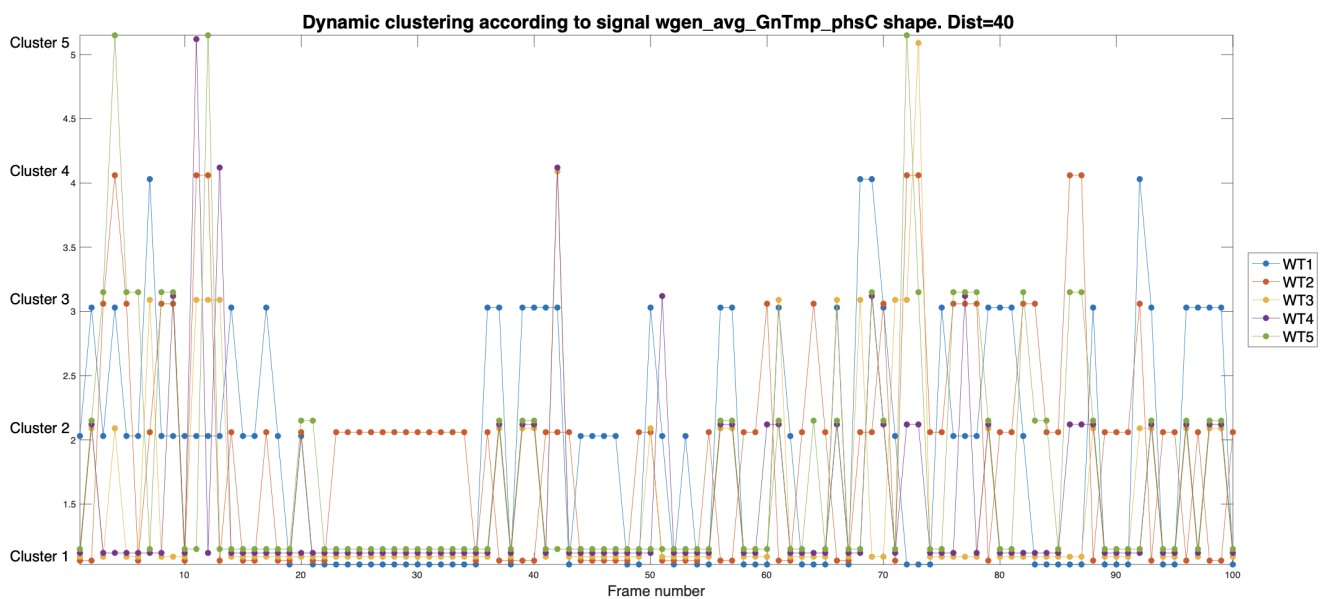


Figure 7. Cluster's dynamic evolution for the temperature signals $wgen_avg_GnTmp_phsC$. Each point represents a time window of 128 samples (640 min). The distance used to prune the HT is 40.

3.2. Analysis of Wind Speeds Measured in the WT Nacelle

In this section, the similarity of the wind speeds measured in the nacelles of each WT is studied using the dynamic clustering tools that have been developed. It is known that for being a small WF, the WTs are close; therefore, the wind conditions to which they are exposed are very similar, although not identical. For the analysis, the time frames of 128 samples are considered and encoded through 5 DFT coefficients. Clustering is performed for all the recorded wind speeds in the database. The HT is built from the DCT vectors using distances of 20, 15, and 10 to evaluate cluster formations. The results are shown graphically in Figure 8, where the upper part displays cluster formation with a pruning distance of 20, the central part with a pruning distance of 15, and the bottom part with a pruning distance of 10. These distances are generally small. In the case of wind speed, it is observed that pruning at a value of 20 practically results in a single cluster. As

the pruning distance decreases, such as to 15, it is noticed that although most of the signals continue to be classified in the first cluster, WT1 (represented in blue) begins to appear in Cluster 2, and occasionally WT5 (represented in green) also appears outside cluster 1. This indicates that the wind conditions in WT1 and, to a lesser extent, in WT5 differ from the others. When pruning is performed at a distance of 10, it is further confirmed that the turbines with more differentiated wind conditions are WT1 and WT5. However, sporadically, some other WTs also appear outside cluster 1.

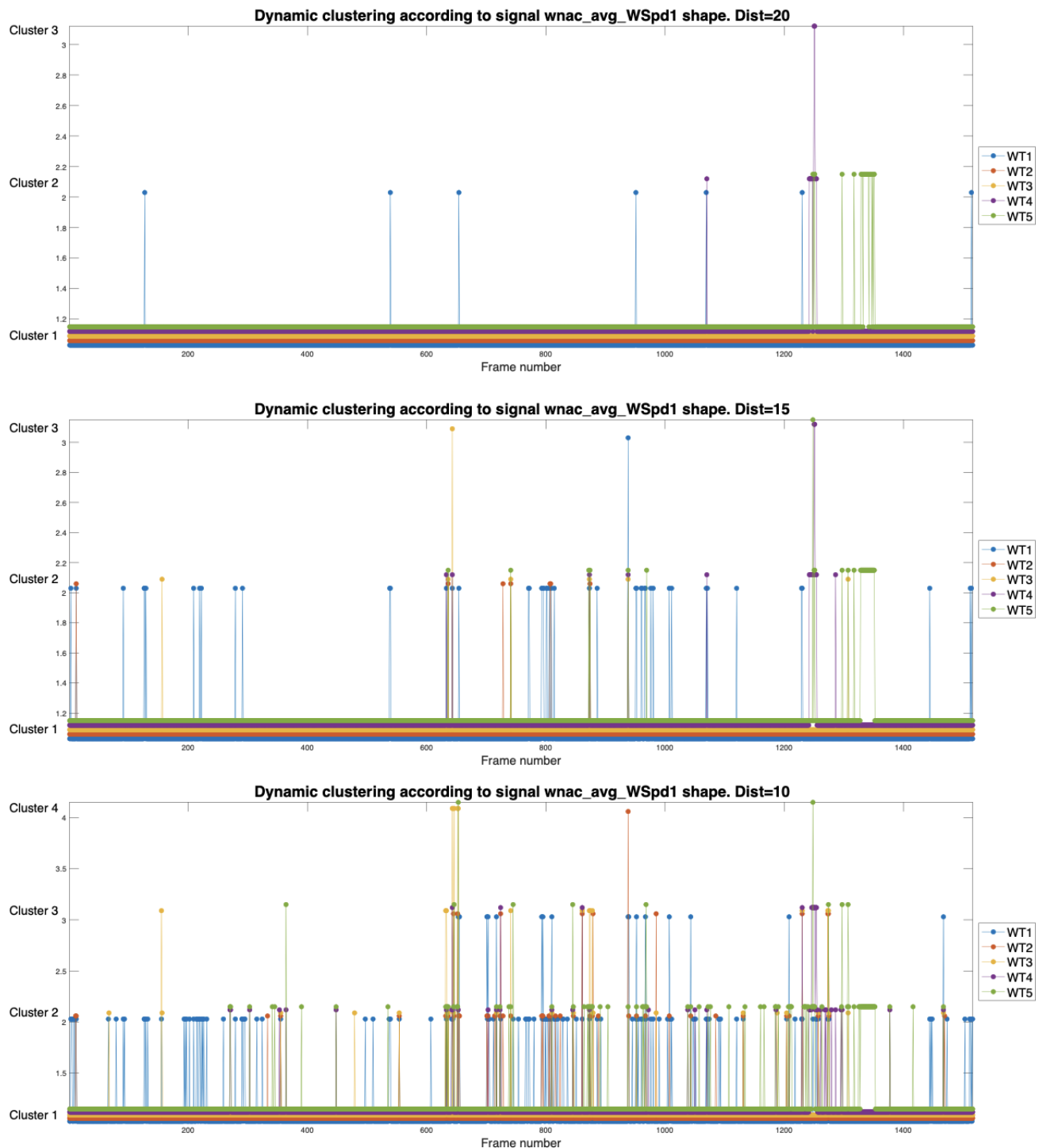


Figure 8. Dynamic Clustering Analysis of the Wind Speeds measured in the nacelles by pruning the HT at 20, 15, and 10 distances, respectively.

The consistency of the clustering results is observed by comparing them with the wind speed-power curves estimated directly from the SCADA data. Figure 9 illustrates this comparison from the work conducted with the same database in [20]. The WT curves are distinguished by the same color code used in the referenced work, while the wind-power curve (WPC) provided by the WT's manufacturer is depicted in black. Figure 9 reveals a close correspondence between generated active power and wind speed, a relationship expected in wind energy production [21]. It is noticeable that almost all curves of the different WTs overlap, except for WT1 in blue. Consequently, wind speed frames from WT1 often fall into a different cluster than those from other WTs. The curves show that WT1 (in blue) is the most efficient of the group. From the clustering of the wind signals, it is observed that the wind measured in WT1 falls more frequently in high clusters, meaning that its signal (shape) is also the most different. Observing the signals in time, the blue signal generally has higher amplitudes. The reason for these results is not perfectly known. It could be that WT1 was in a place with slightly better wind conditions. Given that wind serves as the system's input, it is sensible that differences in input wind are reflected in wind-power curve estimation through clustering techniques. It is also observed that wind turbines (WTs) with similar wind-power curves (WPCs) classify the wind signals into the same clusters.

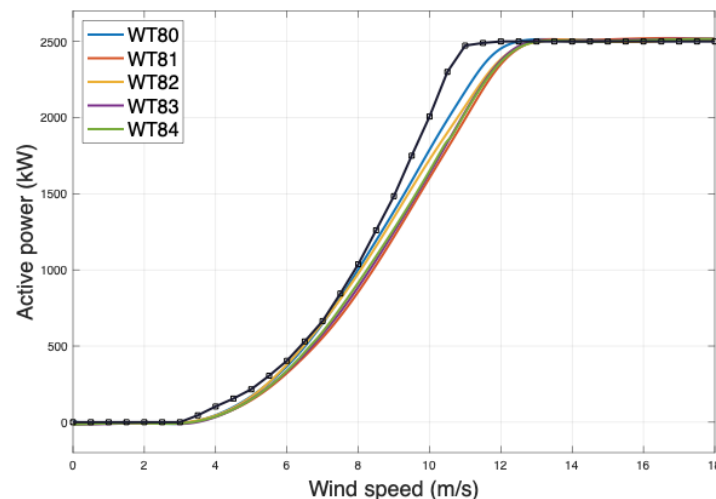


Figure 9. Wind-power curves of the WTs estimated from SCADA data.

3.3. Analysis of Clustering between Generator Speed and Gearbox Oil Temperature

Monitoring and comparing critical variables is crucial in the preventive maintenance of WTs. By way of example, in this section, two variables are analyzed and compared through the dynamic clustering: the generator shaft's rotation speed, $wgen_avg_Spd$, and the gearbox oil's temperature, $wtrm_avg_TrmTmp_GbxOil$. This comparison makes it possible to detect abnormal patterns indicative of wear or malfunction, facilitating proactive interventions before serious failures occur. The objective is to detect the WTs whose critical variables suffer more variations concerning those of the WF and to identify the machines where priority maintenance should be applied.

Figure 10 presents the dynamic study of the variable $wtrm_avg_TrmTmp_GbxOil$, which measures the temperature of the oil in the gearbox subsystem and, to some extent, is related to the degradation of oil in the gearbox. Distinctive patterns are identified between the five WTs studied (WT1 to WT5). WT1 predominantly shows a low wear profile, clustering in Clusters 1 and 2. In contrast, WT2 and WT3 exhibit more significant variability, with sporadic episodes reaching Cluster 3, which may indicate more significant wear and tear events. The behavior of WT4 was similar to that of WT1, suggesting less wear. WT5 presents numerous appearances in Cluster 2 and 3, suggesting that it is subject to more significant wear than the set of WTs.

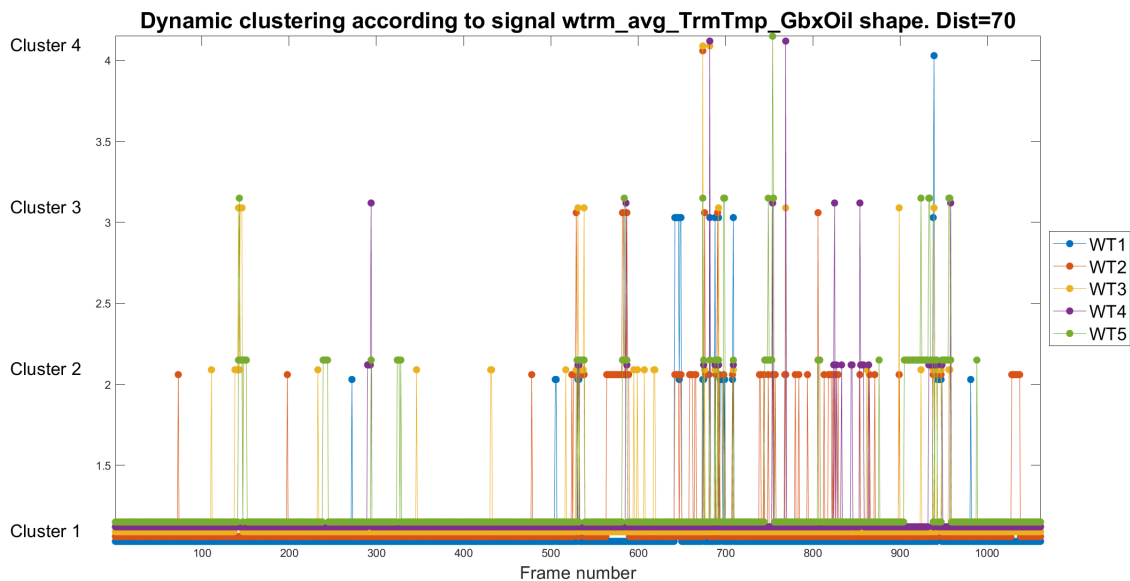


Figure 10. Dynamic clustering of *wtrm_avg_TrmTmp_GbxOil*, the mean gearbox oil temperature performed by pruning the HTs with a distance of 120.

Figure 11 presents the dynamic study of variables *wgen_avg_Spd*. Given the variability and the dynamic range of the analyzed magnitude, the HTs can be at a distance of 350 in this study. Regarding the speed of rotation of the generator, it is observed that WT1 remains clustered most of the time in Cluster 1 and sporadically in Cluster 2, which aligns with the results obtained in the previous oil analyses. WT2 and WT3 have intermittent appearances in Cluster 4, which may imply a relationship between high operating speeds and an increase in wear. WT4 confirms its low wear profile with consistently low rotational speeds. The WT5 shows a more significant number of appearances than the rest of the WTs in Clusters 2 to 4, which is also seen in the analysis of the oil temperature.

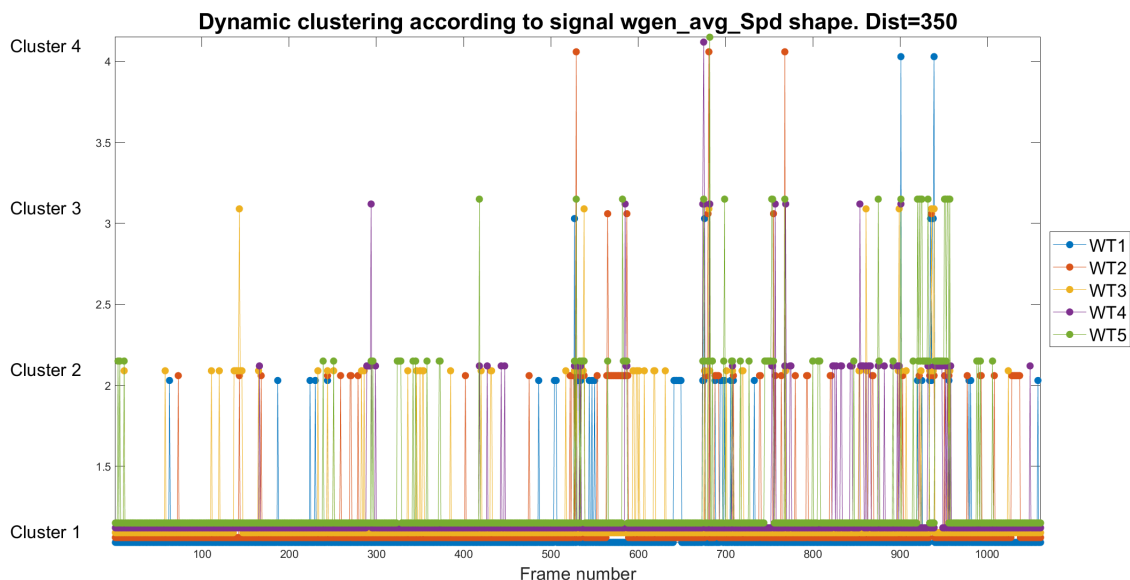


Figure 11. Dynamic clustering of *wgen_avg_Spd* performed by pruning the HTs with a distance of 350.

The two figures show many similarities in the distribution of the clustering of the signals of each WT, whether the study is carried out from the shaft's speed of rotation or the oil's temperature. In the absence of other indicators, when considering predictive maintenance, it is a good strategy to prioritize those machines that present many clusterings in high clusters in the dynamical clustering analysis of their critical variables.

The gearbox of a wind turbine has the function of increasing the speed of rotation coming from the rotor (which rotates slowly) until reaching the speed required by the generator to produce electricity efficiently. During this process, the friction between the gears and other mechanical components generates heat. This heat in the oil leads to wear in it, which can lead to poor operation.

Having shown the fact that the two graphs are so similar, it indicates a correlation in which the higher the speed of rotation, the higher the temperature will be. That is why, for example, WT2 and WT3 demonstrate a more apparent direct relationship when expanding the graphics between frames 500 and 600 in Figure 12, where one can observe the remarkable similarity of the signals in the oil and the rotational speed.

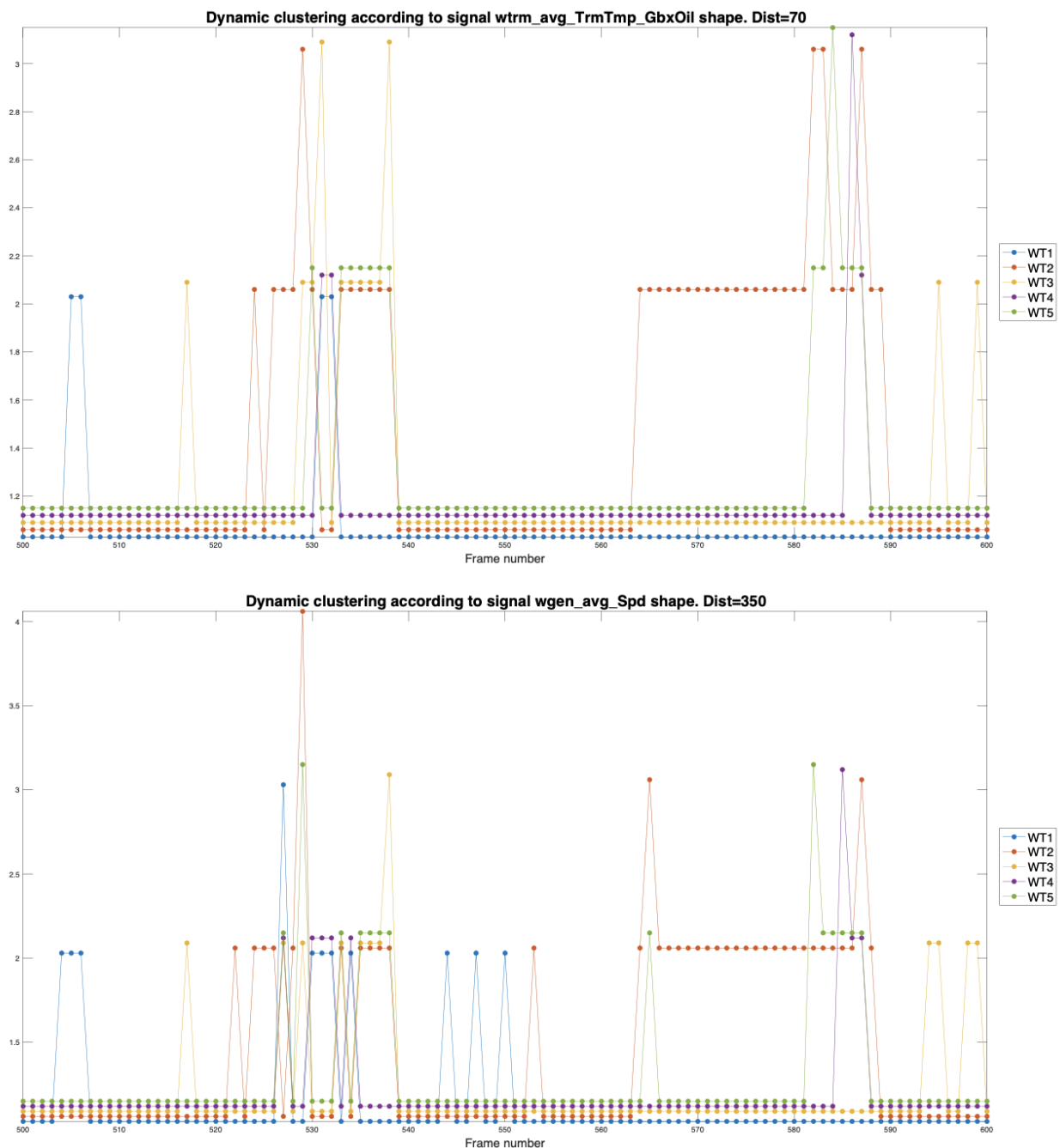


Figure 12. Comparison of gearbox oil temperature and generator speed of the different WTs studied. The analysis focuses on selected intervals for detailed inspection, revealing key patterns in operating behavior that could influence equipment efficiency and maintenance.

4. Discussion

4.1. Configuration Parameters of the Dynamic Clustering Algorithm

The threshold (the distance) used to prune hierarchical trees has been covered extensively throughout the work. In this section, we will explore how the size of the windows and their possible overlap, as well as the number of DCT coefficients used in the coding of the signals, impact the algorithm's performance. We will visually show the results of a particular case through an experiment.

The study begins by investigating the impact of the number of coefficients used in the dynamic clustering algorithm, and the following experiment is designed. The signal *wnac_avg_WSp1* already used is selected. A window size of 128 samples and a pruning distance of 10 were chosen. Then, the algorithm is executed, varying the number L of DCT coefficients in the signal parameterization used from 2 to 5. The results for the entire signal record are shown in Figure 13. In subfigure (a), only two coefficients are used in (b) three, in (c), four, and in (d), five. Despite the difference in the number of coefficients, first of all, it can be observed that the results of the dynamic clusters are very similar, even when using only two coefficients. The most appreciable changes are observed from two, subfigure (a), to three coefficients, subfigure (b). Adding a new coefficient and moving to (4) shows that the new clustering has more WTs in higher clusters, as is appreciated in subfigure (c). It is interesting to note that the difference between using 4 and 5 coefficients is practically undetectable.

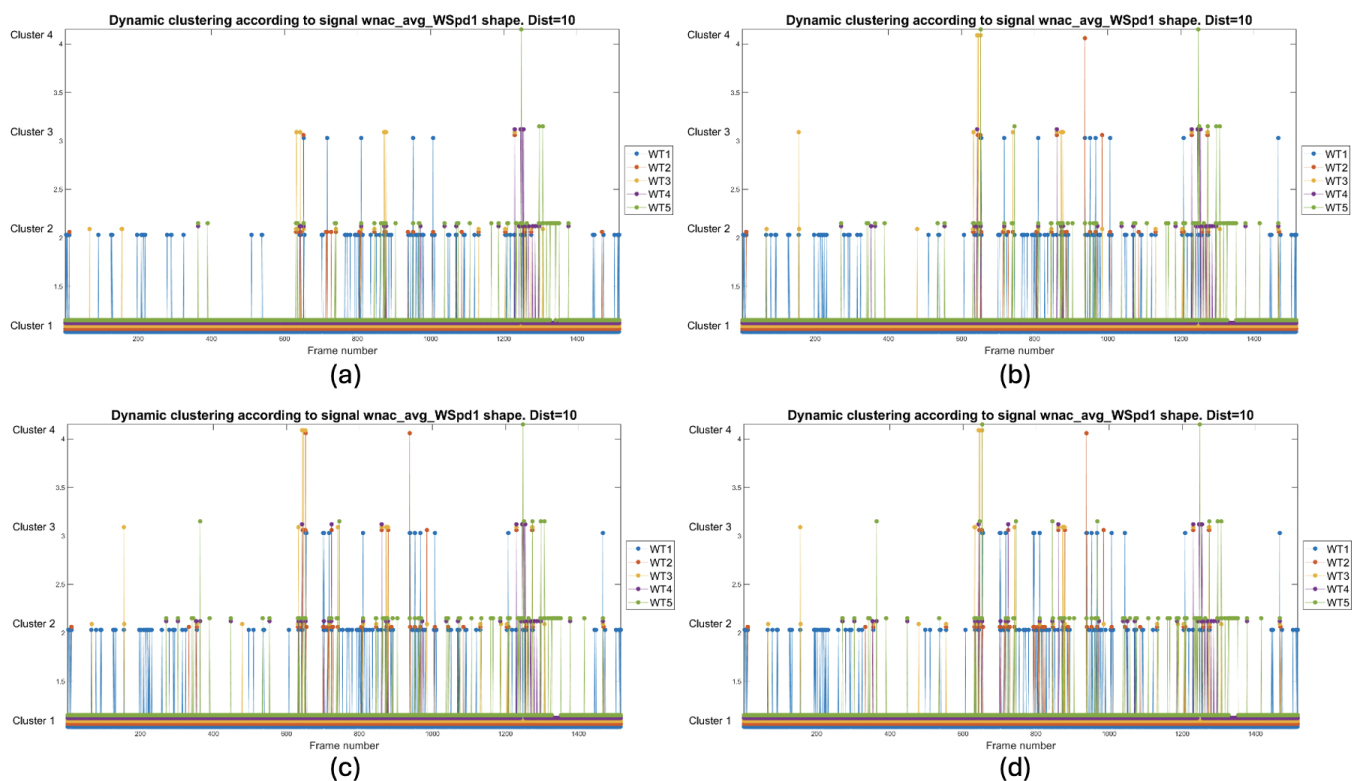


Figure 13. Clustering of the *wnac_avg_WSp1* signal keeping the window size at 128 samples, the pruning distance at 10 and varying the number of DCT coefficients to (a) 2, (b) 3, (c) 4 and (d) 5.

The second experiment is also performed on the signal *wnac_avg_WSp1* works. The number of coefficients and the pruning distance are kept constant at 5 DCTs and 10, and the size of the window varies to 32, 64, 128, and 256; the results are shown in subfigures (a), (b), (c) and (d), respectively. All signal history is processed. Notice that depending on the length of the windows, the clustering algorithm delivers more or less outputs.

This second experiment also provides very similar results regardless of the window length used. It is observed that the concentrations of clusters follow a similar temporal

distribution, and the same WTs form the clusters (see Figure 14). This result is remarkable because we obtain similar information regardless of the window size used for the analysis. Small windows of 32 samples provide much more definition since clustering is computed every 2.67 h while working with a window of 256 samples; the clustering is carried out every 21.3 h, considering a SCADA system working at a frequency of 5 min, as is the case discussed.

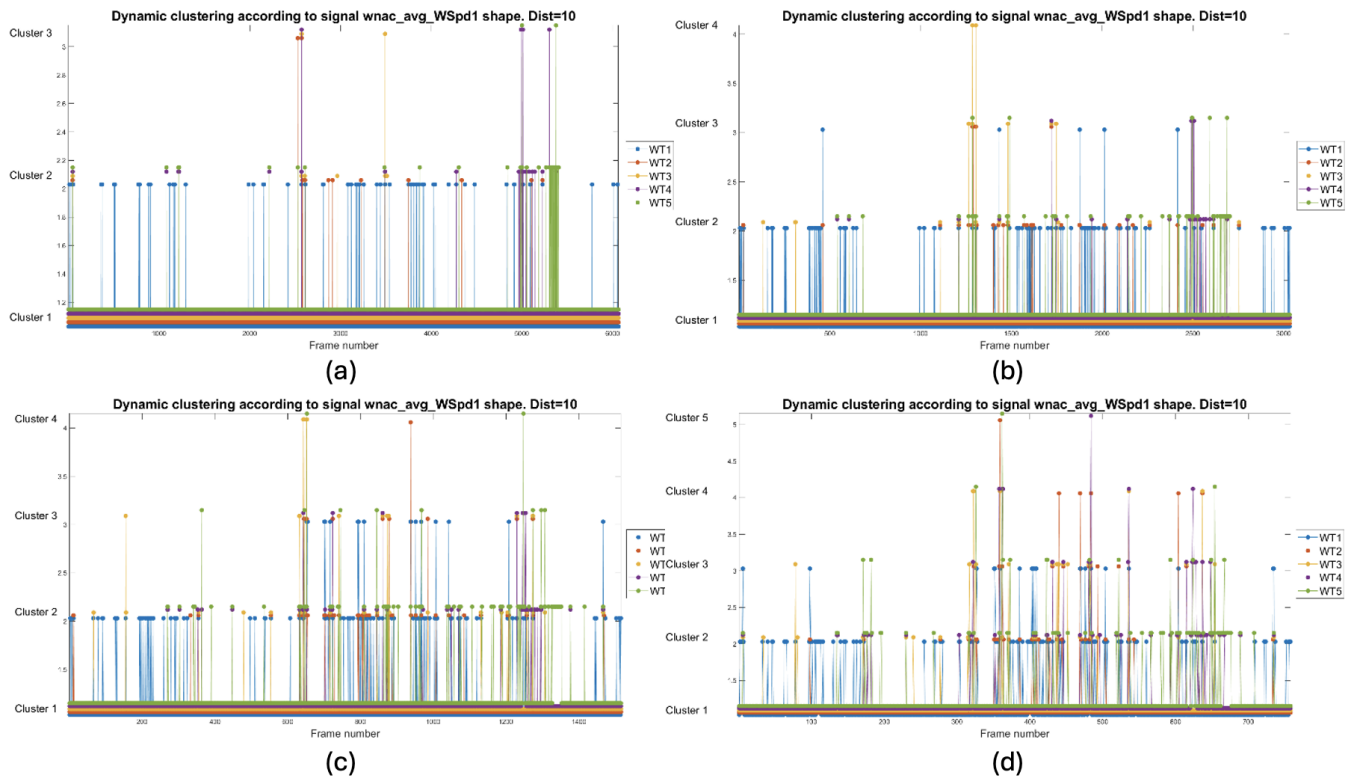


Figure 14. Clustering of the *wnac_avg_WSpd1* signal keeping 5 DCT coefficient, the pruning distance at 10 and varying the window size at (a) 32, (b) 64, (c) 128 and (d) 256 samples.

Finally, the third experiment explores window overlapping. The dynamic cluster results are presented for the same *wnac_avg_WSpd1* signal, keeping the same pruning distance (10) and 5 DCT coefficient and performing different frame overlappings. The results are presented in Figure 15 where the overlap relation appears in the subplots' title. One can appreciate that the window is of 128 samples, and the overlaps are at 50% (64 samples), 25% (32 samples), and 12.5% (16 samples). In all cases, all historical data have been processed. Notice that, as in the previous experiment, the algorithm produces very similar and consistent results quite independently of the overlapping, which indicates that the algorithm is also very robust to the overlapping and, therefore, can be omitted.

The three previous experiments show that the clustering algorithm is very robust regarding the number of DCT coefficients used, the size of the windows, and their possible overlap. The most determining parameter is the threshold used to prune the hierarchical tree. The threshold is a crucial parameter that determines whether signals (and the WTs they originate from) are classified into different clusters if they exceed it or the same cluster if they do not. This interpretation is consistently applied throughout this work. Each signal type requires a different threshold, necessitating some initial exploration to determine the appropriate value. The threshold can be seen as a versatile tool for exploring datasets in the first phase by performing clusterings with different thresholds and finding the first threshold values that begin to form the first clusters.

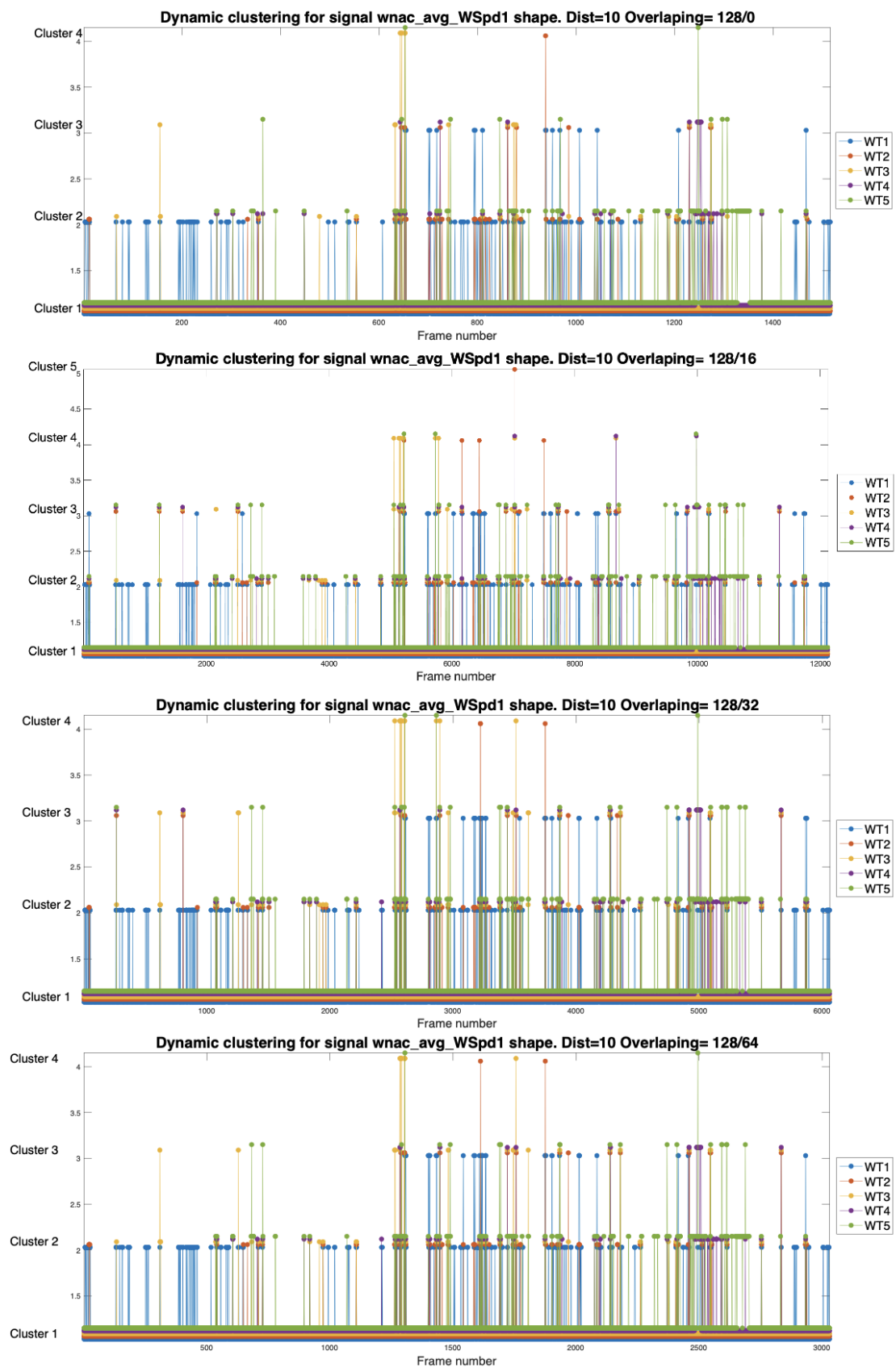


Figure 15. Clustering of the *wnac_avg_WSpd1* signal keeping 5 DCT coefficient, the pruning distance at 10, the window length of 128 samples and varying the window overlapping going from top to down to 0, 12.5%, 25% and 50% of overlapping.

4.2. On the Performance and Limitations of the Dynamical Clustering and Future Research

WTs change operating conditions at different time scales due to the wind, the seasonal weather conditions, and their strongly nonlinear behavior, as seen from the wind-power curve. The significant advantage of using thresholding for hierarchical tree pruning is that it avoids establishing a predetermined number of clusters to run the algorithms, a common requirement in centroid-based clustering techniques such as K-means and K-medoids. The present approach works with a threshold that can be interpreted as the distance between signals required to form the different clusters. Once the threshold is selected, the algorithm processes the recorded historical data covering several years, automatically finding the number of clusters.

The present approach monitors the dynamics of the WF in each time window, which means it computes a picture of the clusters in that particular time window. A challenge found is to follow the evolution of the clusters over time in a way that is easy to follow. In this work, we have introduced a procedure of assigning the name to the clusters (see Section 2.4), which preserves as much frame-by-frame information as possible and facilitates monitoring over time. This step is essential to follow the dynamics of the processes. Even for a WF of 5 WT, the information is messy and complicated to follow if this step is not done. Ordering names, as we propose, works acceptably well in a park of reduced dimensions like the one used in the experiments.

The method's main limitation, as it currently stands, is the temporal monitoring of the clusters that would form in WF with many more WTs. The protocol for naming the clusters that make it easy to follow their dynamic evolution must be improved, and some research must be performed to validate a possible, more scalable solution. Still, the method proved to be strong for the WF investigated. The system was initially designed to cluster based on a single parameter. A possible solution for clustering based on multiple parameters would be to concatenate the five coefficients of the different signals and form a larger feature vector. However, it is necessary to investigate the best normalization that should be applied to the signals since these can have different characteristics. In any case, it is also a topic for future research. Figure 16 shows the two lines to improve the method. In subplot (a), the proposal is to cluster based on more than one signal, and in (b), there is a need to find a representation of the results visually interpretable over time, especially in big WFs.

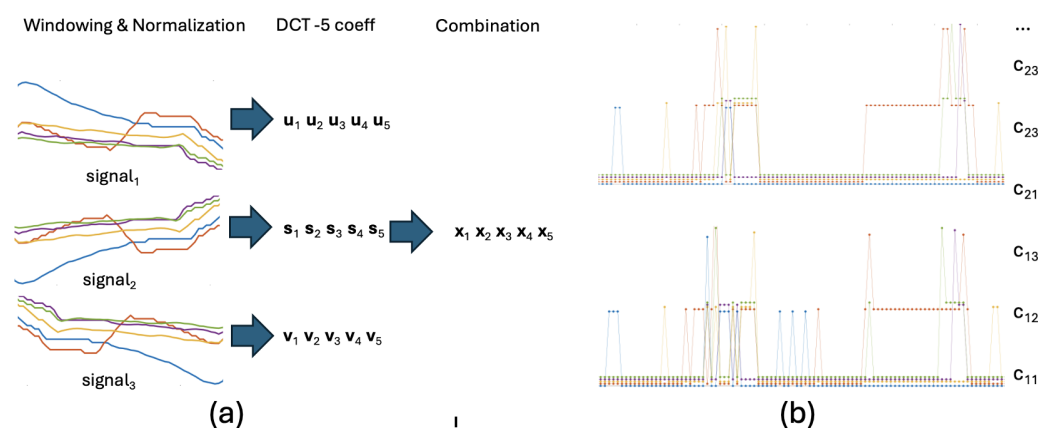


Figure 16. The two main lines of method improvement are (a) to cluster based on more than one signal and (b) to find a representation of the results visually interpretable over time for big WFs.

4.3. Dynamical Clustering Working with K-Means and K-Medoids

The number of clusters, K , varies over time and is unknown. Clearly, all the WTs in the WF often stop working due to the lack of wind, and suddenly, the recorded SCADA temperatures of all the WTs tend to converge to the ambient temperature. In that case, the signals are so close that they all follow in a single cluster. The algorithms that need to know the number of clusters a priori organize the data in the pre-established number of clusters, so a previous estimation of K is required before their execution. A robust solution

to overcome this limitation is presented in this work. Figures 17 and 18 show the dynamic clusterization of the data using frames of 128 samples and the first 5 DFT coefficients for values of $K = 2$ and 3 when using K-means K-medoids, respectively.

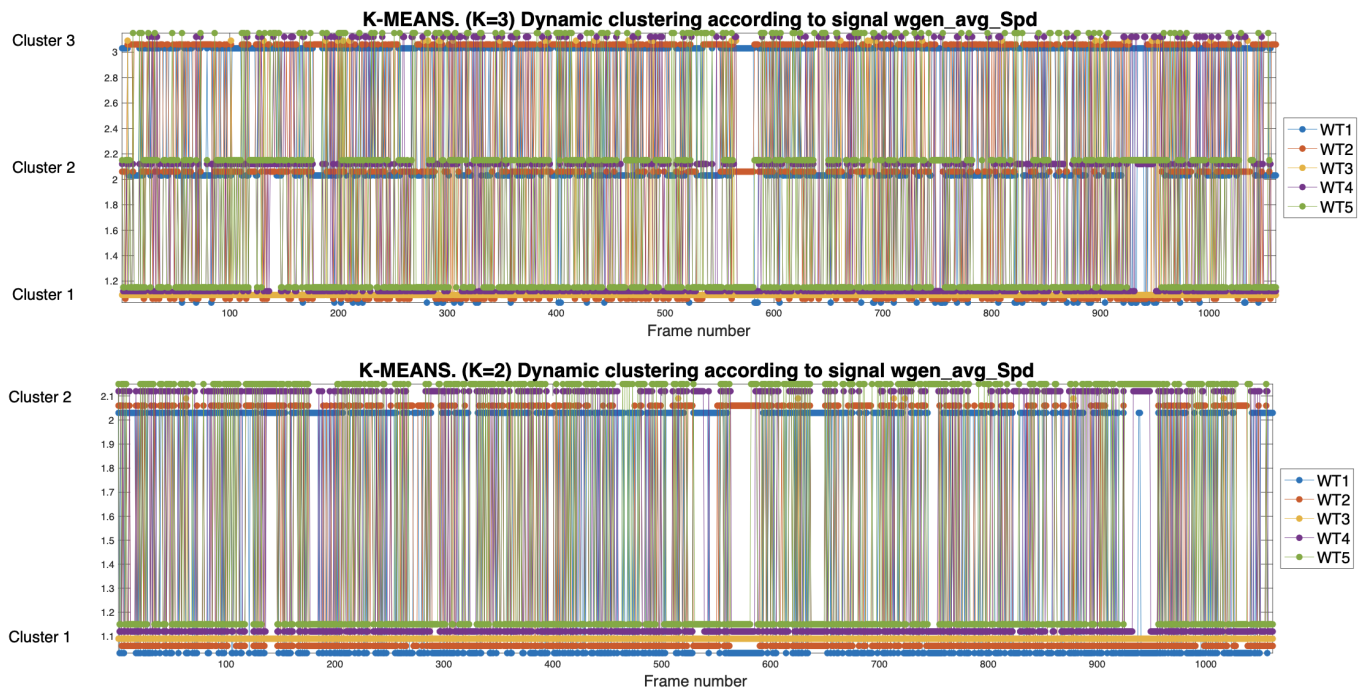


Figure 17. Dynamic clustering using the K-means algorithm for $K = 2$ (below) $K = 3$ (above).

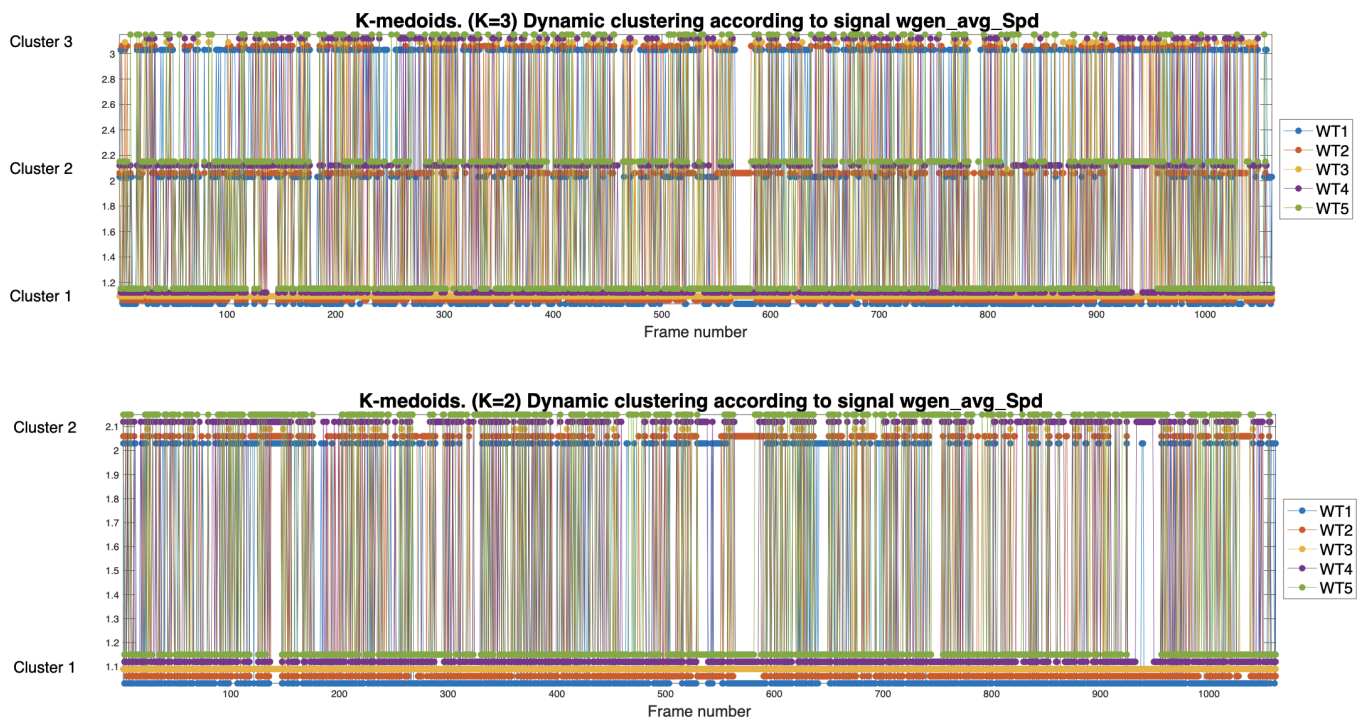


Figure 18. Dynamic clustering using the K-medoids algorithm for $K = 2$ (below) $K = 3$ (above).

5. Conclusions

Specifically, this work explores the ability to group WTs according to criteria of similarity of specific signals since this would allow the subsystems of the machines that generate them to be compared and behaviors similar to detect possible irregularities prematurely.

In this work, a new methodology is proposed to cluster the WTs of a WF into clusters. It is crucial to understand that the operation of wind turbines is highly nonlinear and time-varying, requiring dynamic monitoring over time for extended supervision. The conditions at different instants can be vastly different. The work is at an early stage, but the results obtained showcase great potential. Below, the main contributions are highlighted:

1. Clustering is based on a specific SCADA signal observed during a time interval called a frame. It is carried out frame by frame and works for any averaged signal.
2. Compressing the information of the signals is critical, so the first coefficients of the DCT are used. With the help of the DCT, each WT's signals are represented in low-dimensional vectors.
3. Widely known agglomerative hierarchical clustering techniques are used, and the Euclidean distance is employed, which is applied to the vectors of DCT coefficients. In a more advanced phase of knowledge, other distances can be explored. The advantage of these techniques is that they do not impose a fixed number of clusters. However, setting a distance (a threshold) is necessary to prune the hierarchical trees. Dendrogram-type representations can be used to explore the appropriate distance. Once such distance is decided, it is maintained and used to process all frames.
4. To maintain an interpretable temporal track of the clusters frame by frame, it is crucial to define a stable cluster nomenclature. The information generated when the hierarchical tree is built from the distances between vectors according to a previously explained criterion is used. According to this nomenclature, the most similar signals are organized in low clusters and the most different in high clusters.

Due to the investigations' initial state, different topics must be explored in more detail. For example, it should be noted that this approach may not fully capture complexities in waveforms depending on the signal analyzed and their rapid transitions, thus implying more research on how best to select the appropriate number of DCT coefficients according to signal type, the length of the frame and operational conditions. Another point that may require a significant amount of research is the study of how to understandably represent dynamic WF diagrams with a more significant number of WTs and helpfully for making decisions, which indeed goes parallel to developing a form better to name the clusters.

In this work, however, one can already see the potential of the method presented by first analyzing an individual signal, such as the wind speed measured in the nacelles of the WTs, or the comparison of two critical variables, such as a rotation speed of the generator shaft or the gearbox oil temperature. Being able to know, for all critical variables, which are the WTs that deviate the most from normality, i.e., from Clusters 1 and 2, objectively can be an invaluable help to improve the preventive maintenance of the WFs.

Applying this clustering model in WF operational management could successfully plan preventive maintenance through early detection of abnormal conditions. Case studies have validated its practical applicability by reducing downtime and associated costs, demonstrating that this approach works.

Author Contributions: P.M.-P.: Conceptualization, methodology, software, formal analysis, investigation, validation, writing, project administration and C.N.-V.: data curation, software, methodology, visualization, investigation and writing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministerio de Ciencia e Innovación of the Spanish Government (ref: PID2020-120314RB-I00).

Data Availability Statement: The data can be extracted from the freely available database at: <https://github.com/alecuba16/fuhrlander>, accessed on 21 May 2024.

Acknowledgments: The author would like to thank the Smartive company (<http://smartive.eu/>, accessed on 1 June 2021) for providing the data used in the experimental part.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DCT	Discrete Cosine Transform
HT	Hierarchical Tree
HC	Hierarchical Cluster
SCADA	Supervisory Control and Data Acquisition
WF	Wind Farm
WPC	Wind-Power Curve
WT	Wind Turbine
WT1	Wind Turbine 1
WT2	Wind Turbine 2
WT3	Wind Turbine 3
WT4	Wind Turbine 4
WT5	Wind Turbine 5

References

1. Surucu, O.; Gadsden, S.A.; Yawney, J. Condition monitoring using machine learning: A review of theory, applications, and recent advances. *Expert Syst. Appl.* **2023**, *221*, 119738. [\[CrossRef\]](#)
2. Pandit, R.; Astolfi, D.; Hong, J.; Infield, D.; Santos, M. SCADA data for wind turbine data-driven condition/performance monitoring: A review on state-of-art, challenges and future trends. *Wind. Eng.* **2023**, *47*, 422–441. [\[CrossRef\]](#)
3. Nunes, P.; Santos, J.; Rocha, E. Challenges in predictive maintenance—A review. *CIRP J. Manuf. Sci. Technol.* **2023**, *40*, 53–67. [\[CrossRef\]](#)
4. Zhou, Q.; Xiong, T.; Wang, M.; Xiang, C.; Xu, Q. Diagnosis and early warning of wind turbine faults based on cluster analysis theory and modified ANFIS. *Energies* **2017**, *10*, 898. [\[CrossRef\]](#)
5. Du, B.; Narusue, Y.; Furusawa, Y.; Nishihara, N.; Indo, K.; Morikawa, H.; Iida, M. Clustering wind turbines for SCADA data-based fault detection. *IEEE Trans. Sustain. Energy* **2022**, *14*, 442–452. [\[CrossRef\]](#)
6. Ali, M.; Ilie, I.S.; Milanović, J.V.; Chicco, G. Probabilistic clustering of wind generators. In Proceedings of the IEEE PES General Meeting, Minneapolis, MN, USA, 25–29 July 2010; pp. 1–6.
7. Liu, Y.; Gao, X.; Yan, J.; Han, S.; Infield, D.G. Clustering methods of wind turbines and its application in short-term wind power forecasts. *J. Renew. Sustain. Energy* **2014**, *6*, 053119. [\[CrossRef\]](#)
8. Teng, W.; Wang, X.; Meng, Y.; Shi, W. Dynamic clustering equivalent model of wind turbines based on spanning tree. *J. Renew. Sustain. Energy* **2015**, *7*, 063126. [\[CrossRef\]](#)
9. Wang, H.; Wang, H.; Jiang, G.; Li, J.; Wang, Y. Early fault detection of wind turbines based on operational condition clustering and optimized deep belief network modeling. *Energies* **2019**, *12*, 984. [\[CrossRef\]](#)
10. Likas, A.; Vlassis, N.; Verbeek, J.J. The global k-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [\[CrossRef\]](#)
11. Park, H.S.; Jun, C.H. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* **2009**, *36*, 3336–3341. [\[CrossRef\]](#)
12. Yang, M.S.; Lai, C.Y.; Lin, C.Y. A robust EM clustering algorithm for Gaussian mixture models. *Pattern Recognit.* **2012**, *45*, 3950–3961. [\[CrossRef\]](#)
13. Paik, C.; Chung, Y.; Kim, Y.J. Power Curve Modeling of Wind Turbines through Clustering-Based Outlier Elimination. *Appl. Syst. Innov.* **2023**, *6*, 41. [\[CrossRef\]](#)
14. Xie, W.B.; Lee, Y.L.; Wang, C.; Chen, D.B.; Zhou, T. Hierarchical clustering supported by reciprocal nearest neighbors. *Inf. Sci.* **2020**, *527*, 279–292. [\[CrossRef\]](#)
15. Lee, H.G.; Piao, M.; Shin, Y.H. Wind power pattern forecasting based on projected clustering and classification methods. *Etri J.* **2015**, *37*, 283–294. [\[CrossRef\]](#)
16. Astolfi, D.; Scappaticci, L.; Terzi, L. Fault diagnosis of wind turbine gearboxes through temperature and vibration data. *Int. J. Renew. Energy Res. (IJRER)* **2017**, *7*, 965–976.
17. Astolfi, D.; Pandit, R.; Terzi, L.; Lombardi, A. Discussion of wind turbine performance based on SCADA data and multiple test case analysis. *Energies* **2022**, *15*, 5343. [\[CrossRef\]](#)
18. Maldonado-Correa, J.; Martín-Martínez, S.; Artigao, E.; Gómez-Lázaro, E. Using SCADA data for wind turbine condition monitoring: A systematic literature review. *Energies* **2020**, *13*, 3132. [\[CrossRef\]](#)
19. Marti-Puig, P.; Blanco-M, A.; Cusidó, J.; Solé-Casals, J. Wind turbine database for intelligent operation and maintenance strategies. *Sci. Data* **2024**, *11*, 255. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Marti-Puig, P.; Hernández, J.Á.; Solé-Casals, J.; Serra-Serra, M. Enhancing Reliability in Wind Turbine Power Curve Estimation. *Appl. Sci.* **2024**, *14*, 2479. [\[CrossRef\]](#)
21. Kusiak, A.; Li, W. Short-term prediction of wind power with a clustering approach. *Renew. Energy* **2010**, *35*, 2362–2369. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.