www.nature.com/scientificdata

scientific data

Check for updates

OPEN

DATA DESCRIPTOR

Wind turbine database for intelligent operation and maintenance strategies

Pere Marti-Puig[1,3] ✉, Alejandro Blanco-M.[1], Jordi Cusidó[1,2] & Jordi Solé-Casals[1,3] ✉

With the aim of helping researchers to develop intelligent operation and maintenance strategies, in this manuscript, an extensive 3-years Supervisory Control and Data Acquisition database of five Fuhrländer FL2500 2.5 MW wind turbines is presented. The database contains 312 analogous variables recorded at 5-minute intervals, from 78 different sensors. The reported values for each sensor are minimum, maximum, mean, and standard deviation. The database also contains the alarm events, indicating the system and subsystem and a small description. Finally, a set of functions to download specific subsets of the whole database is freely available in Matlab, R, and Python. To demonstrate the usefulness of this database, an illustrative example is given. In this example, different gearbox variables are selected to estimate a target variable to detect whether or not the estimate differs from the actual value provided for the sensor. By using this normality modelling approach, it is possible to detect rotor malfunction when the estimate differs from the actual measured value.

Background & Summary

Wind energy is essential for meeting the EU Commission's ambitious climate and energy targets for 2020, which include generating at least 20% of electricity from sustainable sources[1]. However, the operation and maintenance (O&M) costs of wind farms, which range from 10% to 35% of overall generation costs, pose a challenge[2]. By reducing these costs, wind farms can become more competitive with fossil fuels and expedite the transition to sustainable energy[3].

To ensure effective management of wind farms (WF), wind turbines (WT) are scheduled for preventive maintenance every 2500 to 5000 hours. However, relying solely on preventive maintenance is insufficient to detect and predict device conditions and anticipate potential failures. The unexpected shutdown of turbines incurs substantial costs, especially considering the logistical challenges of remote locations and the time required for component replacement and on-site repairs. Preventive maintenance schedules for wind turbines are insufficient to detect and predict device conditions and anticipate failures. The life expectancy of WT is commonly estimated at around 20 years, and on average one week of downtime per year is required due to maintenance[4]. This is particularly relevant for those turbines that have been installed in the 1990s and early 2000s that are approaching the end of their lifetime. WF operators have adopted a wide range of measures to extend the operative time of their assets, as mentioned in[5]. Identifying the root causes of failures leading to turbine downtime is essential in reducing inactivity and promptly addressing critical failures. Adopting effective methodologies and tools that assist in this process can significantly benefit wind farm owners by increasing energy production, availability, and cost savings.

Phisycal and data-based modelling. The early deterioration of WT's systems and subsystems can be detected using their physical models or building models from their generated data. The physical model approach is useful for determining and capturing how the various components of turbines work. Monitored components are modelled into systems of physical equations that describe their behaviour from a thermodynamic, electrical, or mechanical perspective. Building physical models of wind turbines requires deep knowledge and expertise in operation principles and a deep knowledge of the WT components. Sometimes, even WT owners do not have all the required information, as manufacturers do not always share the details of the turbine's inner systems.

[1]Data and Signal Processing Research Group, University of Vic–Central University of Catalonia, 08500, Vic, Catalonia, Spain. [2]Enginyeria de Projectes i de la Construcció EPC, Polytechnic University of Catalonia, 08028, Barcelona, Catalonia, Spain. [3]These authors contributed equally: Pere Marti-Puig, Jordi Solé-Casals. ✉e-mail: pere.marti@uvic.cat; jordi.sole@uvic.cat

Nonetheless, such models have been presented in various studies in the literature. For instance, in[6], a physical simulation of the loads of a turbine gearbox is proposed, showing that it can determine the effect that varying loads have on the component's lifespan. Such work requires a dynamic study of the gear conditions and Finite Element Method simulations. The works presented in[7,8] attempt a different approach by first determining the thermal network that describes gearbox conditions. Overall, physical models are a valid option for better understanding the inner workings of turbine components and generating new knowledge about them. Physical models are far more reliable than data-based ones when cause-effect relationships must be determined. However, the demanding data requirements, the availability of the necessary design parameters, and the scarce reusability of the wind turbine physical models are why the industry requires different and more flexible tools.

The main alternative to physical models are data-based models, which have risen in popularity thanks to the advancements in machine learning and statistical modelling. Only minor assumptions of the systems under analysis are needed, as the physical relations governing the operation of the various components are inferred from the data. Data sources available to study WT's behaviour include dedicated sensors that record vibrations and acoustic emissions in mechanical components, such as the gearbox and bearings of the turbine transmission[9,10]. For electrical components, current signatures can be analyzed[11]. However, these three options are particularly expensive as these sensors are not part of the standard equipment of wind turbines. Moreover, installing additional sensors poses a logistic challenge as operations need to be halted.

Wind farms can be of different sizes, ranging from small farms with a few turbines to huge farms with hundreds or thousands of turbines. On the other hand, all wind farms have different behaviours, depending on their geographical location, wind conditions, etc. Large wind farms can also have sectors with different behaviours due to their large geographical extension and the effect that the same turbines exert on each other. Many of the wind data sets currently in use are not publicly available, which challenges the reproducibility of research, particularly in commercially important areas such as wind energy forecasting. Some databases are from offshore WFs, such as[12,13]. An interesting overview of open-source wind energy data is available in[14]. Other databases include aggregated data, which lack turbine-level measurements and turbine-specific energy production. Instead, they comprise aggregated wind energy data covering various spatial scales, from wind farms to entire countries. These datasets differ from turbine-level data in their lower temporal resolution, consisting predominantly of hourly data. Our database is unique because it provides all the raw data of the system for all the WTs and a long period of time.

Condition monitoring via supervisory control and data acquisition system. Condition Monitoring Systems (CMS) employ various strategies, including machine learning techniques such as Artificial Neural Networks (ANN) and Self Organizing Maps (SOM), to analyse wind turbine data and identify deviations from normal behaviour[15]. These methodologies utilize SCADA data for condition monitoring, enabling the prediction of turbine failures. However, classification models in this context pose challenges due to imbalanced data distributions, where most examples belong to the healthy state, and only a few represent the alarm state. Furthermore, the accuracy of these models is hindered by labelling errors in the data. A more efficient maintenance approach known as condition-based maintenance (CBM) has emerged to address these issues. CBM involves ongoing surveillance and the detection of emerging faults through CMS, which acquires and pre-processes sensor data, evaluates it, and interprets the results[16]. CBM enables early detection of incipient faults and proactive planning of maintenance tasks, thus optimizing wind turbines' operation and maintenance process. The literature related to WT maintenance is rich in complex attempts to anticipate failures, ranging from signal processing analyses to physical simulations and machine learning algorithms[17,18]. Researchers often neglect the scalability of solutions, likely due to a lack of large-scale datasets, including multiple wind farms and manufacturers. Most research is developed for individual wind farms or using laboratory simulations. Rarely are algorithms tested on multiple sites characterized by various turbine technologies and environmental conditions. This is a crucial shortcoming of the literature this dataset aims to address.

A valuable data source is the utilization of the Supervisory Control and Data Acquisition System (SCADA), a network of sensors monitoring the status of the turbine. SCADA data was initially designed to provide information to verify the correct operation of turbines and not as a means to assess the health status of individual subsystems. The number of sensors monitored by SCADA can vary between turbine manufacturers, though in general, the major components of the turbine are all instrumented. The resolution of SCADA data can vary, but most commonly is 5–10 minutes, and it is of higher frequency only on very rare occasions. Physical quantities such as temperatures, speeds, pressures, and states of the turbine are included in a SCADA dataset. One valuable characteristic of SCADA data is that it is available and standardized for most turbines, meaning that algorithms for its analysis are more easily transferable from one manufacturer to another. Moreover, being part of the standard instrumentation, it does not require additional investments by the wind farm owner. The importance of SCADA data for predictive maintenance and monitoring has greatly increased in the last decade. The works in[19,20] are two of the first attempts to use SCADA data for WT condition monitoring. The methods to analyze and extract information have greatly improved from the early days. In the literature, algorithms are available to assess the health of all major components using diverse approaches based on statistical analyses, machine learning, and deep learning[17,18]. The benefits of SCADA are its wide availability, highly standardized format, and low cost. Nonetheless, its low data acquisition rate has been mentioned as an important limitation that can hinder the capability of correctly modelling the status of a turbine and detecting failures[21].

The use of machine learning, normality models or other types of modelling strategies based on data analysis can be used for O&M. The illustrative example presented in this manuscript highlights how extreme learning machines (ELMs) can be used to predict a variable from other variables in the system, which can help to detect a malfunction of the wind turbine (specifically the gearbox in the provided example), and hence the deployment of a maintenance check of the wind turbine. To advance the development of condition-based maintenance (CBM) strategies, we release a comprehensive 3-year dataset. This dataset covers all the information obtained from the SCADA system of a wind farm, which includes five 2.5 MW Fuhrländer FL2500 wind turbines. The interesting fact about the database is that it is complete, containing all the information from the wind farm's SCADA system. No variable or information has been modified, and therefore it can be a good starting point for experimenting with this type of data and its use to improve O&M strategies.

This dataset, already used in other papers (see[22–24] as examples) is presented in raw, unprocessed form, as supplied by the system. It includes all variables and alarms from the different systems and subsystems of the WTs.

Methods

The dataset contains 312 analogous variables recorded at 5-minute intervals by the wind farm's SCADA, from 78 different sensors. Wind turbines consist of nine main systems, namely Converter, Generator, Nacelle, Rotor, Tower, Transformer, Transmission, Turbine, and Yaw. Some of these systems are further divided into specific subsystems. Specifically, there are 16 identified subsystems including Hub, Hydraulic System, Main Bearing, Pitch, Power Cabinet, Roof, Rotor, Tower, Transformer, and Yaw. For a detailed overview of the relationship between the systems and their corresponding subsystems, please refer to Supplementary Table 1.

The WTs have 78 different sensors (see Fig. 1). Each sensor in the system provides data regarding a specific subsystem and presents summary information in the form of four statistical parameters calculated at 5-minute intervals. Variables from the same sensor can therefore be identified by their common name, with the only distinction being the specific term added: avg (mean), std (standard deviation), min (minimum), or max (maximum). In total, there are 78 subsystems, giving a total of 78 multiplied by 4, which equals a collection of unique variables. For a complete list of names and corresponding variables, see Supplementary Table 2.

The database also provides a comprehensive collection of alarm information. A total of 369 alarms have been identified, each of which is assigned a unique (integer) code and associated with a specific system and its corresponding subsystem, accompanied by a short descriptor. This allows the database user to easily retrieve the set of alarm codes specifically linked to a particular system of interest. This information is available in Supplementary Table 3.

Gearbox. In the WT subsystems as a whole, the gearbox is a system to be monitored because a broken or damaged gearbox is a serious and costly breakdown with prolonged downtime. The gearbox is a device that increases the speed of the slow but powerful rotation of the rotor to an optimal level for the generator. This allows the generator to convert the maximum mechanical energy of the wind into electricity. During this energy transformation, the gears of the gearbox are stressed due to the difference between the input torque and the opposite torque of the generator at the output. As a result, some parts of the gearbox experience fatigue and an increase in temperature, which hinders the effectiveness of lubrication. Detecting gearbox failures, especially in the early stages, can be difficult. In many cases, a failure in the gearbox involves the replacement of the component with a new one. This failure can be of slow onset (degradation) and, in that case, can be predicted before total failure (breakage).

In a WT gearbox, the transmission is organised in three main parts: the planetary stage, the intermediate stage, and the high-speed stage. Each stage consists of specific components, basically gear parts with a different number of teeth, which allow the speed of the rotor to be adjusted to the generator Fig. 1.

Figure 2a shows a two-speed, stall-controlled, three-bladed upwind turbine drivetrain, in which the gearbox subsystem is shown within a red box. The main components of the drivetrain are the hub, main bearing, main shaft, gearbox, brake, generator shaft and generator. In Fig. 2b, the gearbox is depicted in more detail, showing various types of bearings used depending on the load conditions and gearbox life requirements. In this example, the planetary gearbox is supported by two cylindrical roller bearings (fcCRB), and each planetary gear is supported by two identical cylindrical roller bearings (CRB). Each gearbox parallel shaft is supported by a CRB on the upwind side of the assembly and by two tapered roller bearings (TRB) mounted back-to-back on the leeward side.

Regulations and standards. The International Electrotechnical Commission (IEC) standards 61400-25 series provide comprehensive guidelines for monitoring and control systems in wind power plants. These standards are particularly relevant in the context of failure detection in wind turbines, especially when considering the implementation of IEC61400 as a SCADA naming standard in wind farms. A detailed examination of each standard's significance is essential for framing research in this area. The most important ones are:

a) IEC 61400-25-2:2015 - Information Models: This standard specifies information models related to wind power plants. The information models typically include the four attributes detailed in Table 1.
b) IEC 61400-25-3:2015 - Information Exchange Models: Focusing on methods and models for information exchange within wind power plants, this standard outlines essential protocols and communication patterns for efficient data transfer. The most important aspects are listed in Table 2.
c) IEC 61400-25-4:2008 - Mapping to Communication Profile: This standard addresses the mapping of information and exchange models to specific communication profiles.
d) IEC 61400-25-5:2006 - Conformance Testing: Providing guidelines for testing the conformance of wind power plants to the IEC 61400-25 series, this standard is essential for validating that the monitoring and control systems adhere to international standards.
e) IEC 61400-25-6:2010 - Logical Node Classes and Data Classes for Condition Monitoring: This part of the series specifies logical node classes and data classes for condition monitoring of wind power plants.

In the context of the IEC 61400-25-2:2015 standard, labels or naming conventions for various parameters are standardized to ensure consistency and interoperability. For parameters such as Gearbox Temperature, Main

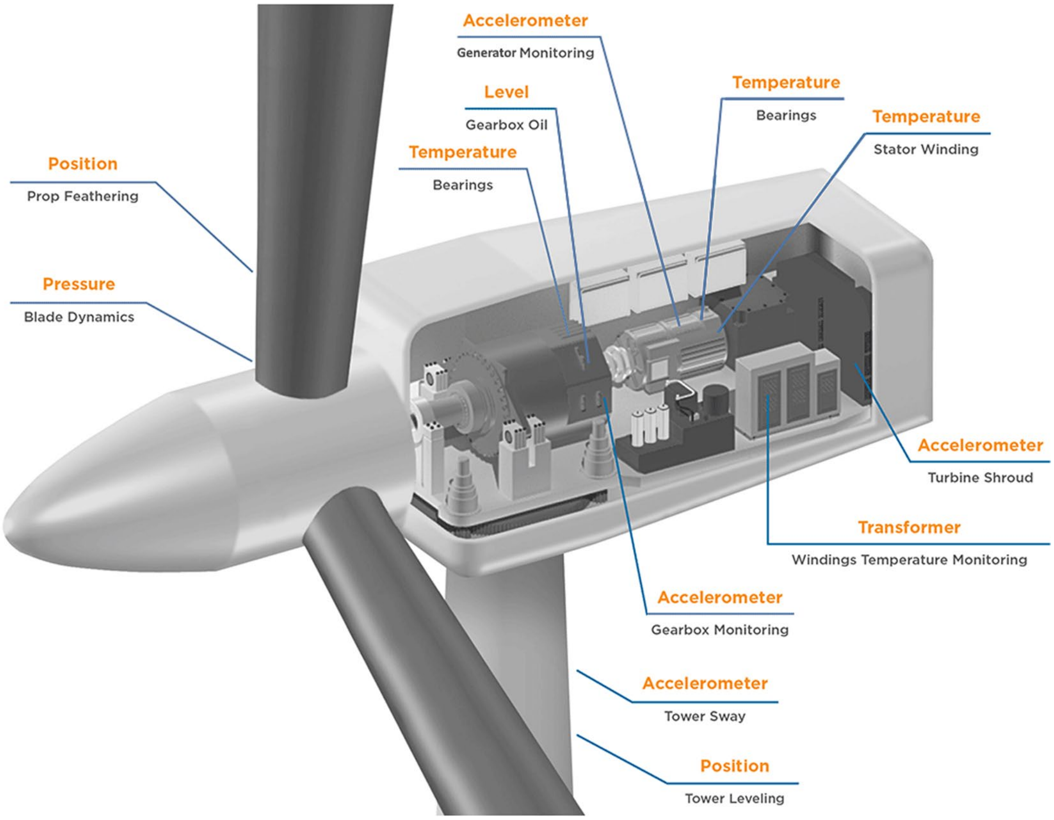

Fig. 1 Wind Turbine system and sensors. Adapted from TE connectivity (http://www.te.com/).

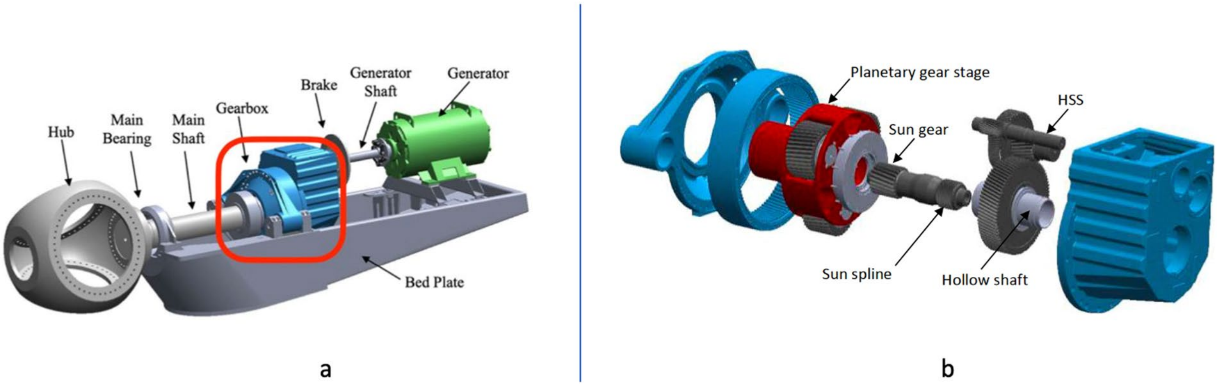

Fig. 2 (**a**) Drivetrain configuration. (**b**) Gearbox configuration.

Bearing Temperature or Active Power, the labels would typically adhere to a structured format that includes several components to accurately describe the data point. Here are some examples of how these labels might be structured:

- Gearbox Temperature:
  Label: *WTurbine1.GBX.Temp*
  WTurbine1 is the wind turbine identifier.
  GBX is an abbreviation for Gearbox.
  Temp indicates temperature measurement.

- Main Bearing Temperature:
  Label: *WTurbine1.MBear.Temp*
  WTurbine1 is the wind turbine identifier.
  MBear is an abbreviation for Main Bearing.
  Temp indicates temperature measurement.

Data Attributes	Represent specific pieces of information, like wind speed, turbine rotational speed, power output, and temperature. Each attribute is defined with a specific data type and range.
Object Models	Collections of data attributes and methods that represent a specific component of the wind turbine, such as a blade, gearbox, or generator. Object models define how data is organized and related within the system.
Hierarchical Structure	This structure facilitates the organization and retrieval of data, ensuring that the relationships between different components of the wind turbine are logically represented.
Standardized Naming Conventions	Specifies naming conventions for different data elements. This uniformity is crucial for interoperability and easy integration with various SCADA systems.

Table 1. Attributes of the information models.

Communication Protocols	Protocols and methods for data exchange, ensuring compatibility and efficiency in communication between different systems and devices.
Data Exchange Patterns	The standard outlines various patterns of data exchange such as request-response, publish-subscribe, and report by exception. These patterns define how data is transmitted, received, and processed.
Security and Reliability	Aspects of secure and reliable data transmission are covered. This includes encryption, authentication, and error-checking mechanisms to ensure that data exchange is secure and error-free.
Interoperability Guidelines	Provides guidelines for ensuring that different systems and devices can work together seamlessly. This is especially important in environments where components from different manufacturers need to communicate with each other.
Real-time Data Exchange	It emphasizes the capability for real-time data exchange, which is critical in operational monitoring and control, as well as in failure detection and response scenarios.

Table 2. Protocols and communication patterns.

- Active Power:
  Label: *WTurbine1.Gen.PwrAct*
  WTurbine1 is the wind turbine identifier.
  Gen is an abbreviation for Generator.
  PwrAct indicates Active Power measurement.

These labels are illustrative and follow a logical format, but the exact naming convention may vary depending on the specific implementation and configuration of the SCADA system in use. The key is to maintain a consistent and descriptive naming scheme that aligns with the guidelines of the IEC 61400-25-2:2015 standard, facilitating clear identification and management of data points across the wind power plant's monitoring and control systems.

Data Records

The data is available in the repository located at the Figshare repository[25], but also made available on GitHub at https://github.com/alecuba16/fuhrlander. It offers a valuable opportunity to access the raw, unprocessed data generated by the wind farm's SCADA system. Unlike many publicly available datasets, which often offer limited or filtered information, our dataset contains the complete, unaltered data directly downloaded from the server. We have deliberately chosen not to pre-process the dataset so that users can explore and analyse the data with methods of their choice. The dataset, stored within the 'dataset' folder, is in JSON format. It encompasses a comprehensive collection of data obtained from five wind turbines, specifically turbines 80, 81, 82, 83, and 84, spanning a duration of three years from 2012 to 2014. However, there is a data gap due to a temporary failure of the SCADA system. This is something that may occur in real applications, making it more difficult to work in these environments. The recorded data has a frequency of 5 minutes, encompassing four indicators for each of the 78 sensors, ultimately resulting in a total of 312 variables. For every 5-minute interval, the dataset includes minimum, maximum, mean, and standard deviation values for each sensor, providing a comprehensive overview of their respective measurements.

Table 3 presents a partial extract of the variables, showing the identifier (ID), the time (TIME) and the weather systems (MET). The complete list of variables is given in Supplementary Table 2.

Additionally, the dataset incorporates valuable information regarding warning and alarm events. These events are accompanied by details specifying the affected system and subsystem, along with concise descriptions elucidating the nature of the event. This supplementary information enhances the dataset's value by offering insights into potential system anomalies and critical occurrences.

Table 4 contains the 15 initial alarms recorded in the dataset. Each row represents an individual alarm, with an identifier (ID) represented by an integer value. In addition, the table indicates the system (SYS) and subsystem (SUBS) associated with the occurrence of each alarm. Lastly, a concise description (DESC), provided by the system, is included to give more context and information about the alarm event.

Technical Validation

The dataset originates from a historical dump of the Smartive company's RDS, a relational database system used to store information from various monitored wind plants. The RDS tables were updated through a push mechanism using an OPC standard driver connected to the Wind Farm. Specifically, the wind farms were equipped with the Mita-teknik SCADA platform[26], which notified the driver whenever a variable aggregation of 5 minutes was

System	Variable Group	Stat Type	Signal Name
ID	id	single	turbine_id
TIME	time	single	date_time
WMET	wmet_MetAlt1_Hum	min	wmet_min_MetAlt1_Hum
		avg	wmet_avg_MetAlt1_Hum
		sdv	wmet_sdv_MetAlt1_Hum
		max	wmet_max_MetAlt1_Hum
	wmet_DewPTmp	min	wmet_min_DewPTmp
		avg	wmet_avg_DewPTmp
		sdv	wmet_sdv_DewPTmp
		max	wmet_max_DewPTmp
	wmet_MetAlt1_Press	min	wmet_min_MetAlt1_Press
		avg	wmet_avg_MetAlt1_Press
		sdv	wmet_sdv_MetAlt1_Press
		max	wmet_max_MetAlt1_Press

Table 3. Extract from the list of variables, divided into systems and variable groups, for the ID, TIME and WMET systems.

ID	SYS	SUBS	DESC
0	Turbine	Control Cabinet	System OK
5	Turbine	Control Cabinet	Vibration
7	Turbine	Control Cabinet	Turbine is serviced
9	Turbine	Control Cabinet	Remote stop
13	Turbine	Control Cabinet	Manual stop
16	Turbine	Control Cabinet	Emer.stop cont.panel
23	Turbine	Control Cabinet	Repeating error
30	Nacelle	Control Cabinet	Nacelle temp.
31	Nacelle	Control Cabinet	Nacelle temp. stop
41	Turbine	Control Cabinet	UPS battery low
45	Turbine	Power Cabinet	Main ctrl. Supply
55	Turbine	Control Cabinet	Main ctrl.man.reboot
66	Turbine	Control Cabinet	Fire alarm
93	Turbine	Control Cabinet	Service hatch
100	Turbine	Control Cabinet	Repeated grid error

Table 4. List of the first 15 alarms provided by the manufacturer. Each alarm is identified by a numeric ID. Note that ID numbers are integers, ordered from lowest to highest, but not consecutive.

completed with its value. The Smartive-developed driver then stored the value in the RDS historical table. This data path represents the sole source of truth, as there is no alternative means to access the raw wind turbine data. The engineer responsible for the driver implemented a range check based on the SCADA variable information endpoint provided by the mita-teknik hub before storing the raw data. To facilitate the use of this dataset, we have made it publicly available in a self-contained JSON format, eliminating the need for third-party software or specific drivers like parquet or SQL. To validate the dataset's accuracy, we performed a comparison with the original RDS by sampling several rows from different turbines and time intervals.

Usage Notes

We downloaded the package *fuhrlander-master.zip* from the repository[25] or the GitHub located at https://github.com/alecuba16/fuhrlander. After downloading the package, we proceeded to extract its contents directly into the designated working directory. The extracted content included the following files: *LICENSE*, which contains the licensing information, and *README.md*, which provides instructions and essential information about the package, the python program *export_variable_info_from_json.py* and the directories *dataset, matlab* and *r*.

The program called *export_variable_info_from_json.py* is used to extract alarm information from the dataset. Within the *dataset* directory, you will find five zipped files, along with a single file in.json format. This particular. json file contains comprehensive information about the wind farm, including details related to alarms and wind turbines. Each of the zipped files contains individualised data for a specific wind turbine.

To facilitate the use of the information in the database, we have provided additional resources in the form of functions and examples. These resources are located in the directories *matlab* and *r*, respectively. The *matlab* directory contains functions and examples specifically designed to support the use of database information in the MATLAB environment, while the *r* directory provides corresponding resources for use in the R programming language.

To exploit the full potential of the database and the accompanying functions described in this section, we have prepared four illustrative examples. Although the examples have been implemented in MATLAB, they can easily be adapted for use in Python or R programming environments.

The first example demonstrates the necessary dependencies and paths that must be built into the development environment to efficiently access the supplied data and functions. This example assumes that the directory structure provided in the repository remains intact after unpacking. By following the prescribed directory structure, developers can easily access the data and functions needed to facilitate their work.

As a quick start to help users use the dataset, we have implemented outlier detection methods in the repository, as detailed in Section 1 of the Supplementary Material. Users can choose between two options: (i) to remove values outside the range mean ± 3 standard deviations, or (ii) to remove values outside the range median ± 2 absolute deviations from the median. This process is applied to each block of data and the remaining data are used to calculate the mean or the median, depending on the user's choice. However, users can explore alternative strategies by downloading the data without applying any pre-processing.

The dataset includes all subsystem variables, sorted according to the time events they capture. In addition, warnings and alarms that occurred during the period are provided separately. Users have the possibility to download only the variable data or both the variable data and the associated warnings and alarms, as described in Section 1 of the Supplementary Material.

This dataset has already been used in a number of previous works, demonstrating its value in improving maintenance strategies. For more information on the content and potential applications of the dataset, we encourage users to consult the relevant publications[22–24,27]). These resources will provide valuable context and guidance on how to make the most of the dataset.

```
%% EXAMPLE 1 Set paths & import functions

dataset_path="/Users/../fuhrlander-master/dataset";
matlab_path="/Users/../fuhrlander-master/matlab";
addpath(dataset_path,matlab_path);

import turbine_data.*; %for turbine_data.m functions
```

In the second example, we illustrate the process of reading the file named *wind_plant_data.json* and storing the wind farm information in a MATLAB structure. In this particular example, we will refer to the structure as 'WF'. Next, we illustrate how to retrieve the alarm identifiers associated specifically with the 'Gearbox' subsystem. By accessing the organised data within the WF structure, a vector of numeric alarm identifiers related to the specified subsystem can be obtained. This approach can be similarly applied to acquire information on alarm identifiers linked to any wind turbine system or subsystem.

```
%% EXAMPLE 2 Getting alarm info

% Decode JSON-formatted text. WindFarm information (alarms and WTs) in the struct
    WF.
WF = jsondecode(fileread(strcat(dataset_path,'/wind_plant_data.json')));

gearbox_alarms_ids=WF.alarm_dictionary.alarm_id(WF.alarm_dictionary.alarm_subsystem
    =="Gearbox");
```

The third example shows how to obtain the raw data of the WT80. This data is compressed in the file *turbine_80.json.bz2* and the function *get_turbine_data* decompresses it. The SCADA system of these machines provides the 4 statistical measurements in intervals of 5 minutes, so the Data Rate must be 300 s as it is required in seconds. The data of the desired WT is obtained by simply changing the identifier of each WT (80, 81, 82, 83, 84). The result is a table named WT80 in this example 3.

```
%% EXAMPLE 3 How to download WT80 raw data

WT_id=80; % Allowed ids: 80, 81, 82, 83, 84
WT_file=strcat(dataset_path,'/turbine_', num2str(WT_id), '.json.bz2');
DR=300; % Data rate, in seconds

[iserror,WT80,msg]=get_turbine_data(WT_file,[],DR,"mean",true,true);
if(iserror)
    error(msg)
end
```

Because of the importance of the statistical distribution of variables, the following example explains how to obtain a summary of the statistics of any selected variable. A Matlab function has been prepared for this purpose:

```matlab
%% EXAMPLE 4 How to obtain summary statistics

function summary_table = summary_WT(WT,v_name)

if isnumeric(v_name)
    v_name = WT.Properties.VariableNames(v_name);
    % to ensure we only have one variable
    v_name = v_name{1};
    data = table2array(WT(:,v_name));
else
    data = table2array(WT(:,v_name));
end

% create a table with summary statistics and percentiles
summary_stats = [mean(data), std(data), min(data), max(data)];
% Calculate desired percentiles
percentiles = prctile(data, [25, 50, 75]);
variable_names = {'Mean','Std_Deviation','Min','Max','Percentile_25','Percentile_50
    ','Percentile_75'};
summary_table = table(summary_stats(1), summary_stats(2), summary_stats(3),
    summary_stats(4), percentiles(1), percentiles(2), percentiles(3), 'VariableNames
    ', variable_names);
fprintf(['Summary statistics of the variable ',v_name,':\n'])
fprintf('\n')
disp(summary_table)
% boxplot
figure('Name',v_name)
boxplot(data)
title(['Boxplot of the variable ', v_name],'Interpreter','none','FontSize',15)
end % end of the function
```

The summary statistics for the variable 'wgdc_avg_TriGri_PhV_phsA' of the WT80 can be obtained by the following instruction:

```matlab
% Generating the table ST containing the summary statistics for the selected
    variable (column 11).
% The variable can also be passed using the name in the database:
% ST = summary_WT(WT80,'wgdc_avg_TriGri_PhV_phsA'); % the name of the variable can
    be used
ST = summary_WT(WT80,11); % or the column of the variable can be used
```

The result generated by the function is shown below (see Fig. 3).

Finally, the last example shows the process of synchronising data and alarm information. From the previous example, we will establish a connection between the selected data of the subsystem 'Gearbox' and the data corresponding to the WT80 by means of the function *get_turbine_data*.

There are two different methods to link alarms and data. In this particular case, the output of the function will be in table format. The initial 312 columns of the table represent the 312 signals derived from the SCADA system of WT80. These alarm-related columns are coded in binary format, with a value of 0 indicating no alarm activation and a value of 1 indicating alarm activation.

It is important to note that the function *get_turbine_data* has additional functionalities beyond the linking of data and alarms. In the context of this example 5, in addition to establishing the link between data and alarms, the information is integrated in time intervals of one hour (3600 seconds). Furthermore, the 'filtered_3sdv_mean' option is used to filter outliers present in all variables using the 3σ-rule.

```matlab
%% EXAMPLE 5 Linking of data with alarms, aggregation of information at 1-hour
    intervals, and filtering of outliers.

DR2=3600;

[iserror,WT80,msg]=get_turbine_data(WT_file,gearbox_alarms_ids,DR2,"
    filtered_3sdv_mean",true,true);
if(iserror)
    error(msg)
end
```

Options and possibilities of the *get_turbine_data* function can be found in the Supplementary Material.

Illustrative example. In this section, we will present the results of a simple normality model applied to the WT84 turbine. It is known that this turbine experienced a major failure and remained out of service for a long period of time. Our aim is to demonstrate how a normality model, built using Extreme Learning Machines (ELM) as discussed in[23], can effectively anticipate and detect such incidents at an advanced stage.

Summary statistics of the variable wgdc_avg_TriGri_PF:

Mean	Std_Deviation	Min	Max	Percentile_25	Percentile_50	Percentile_75
−0.44035	0.77528	−1	1	−0.987	−0.95917	0.51404

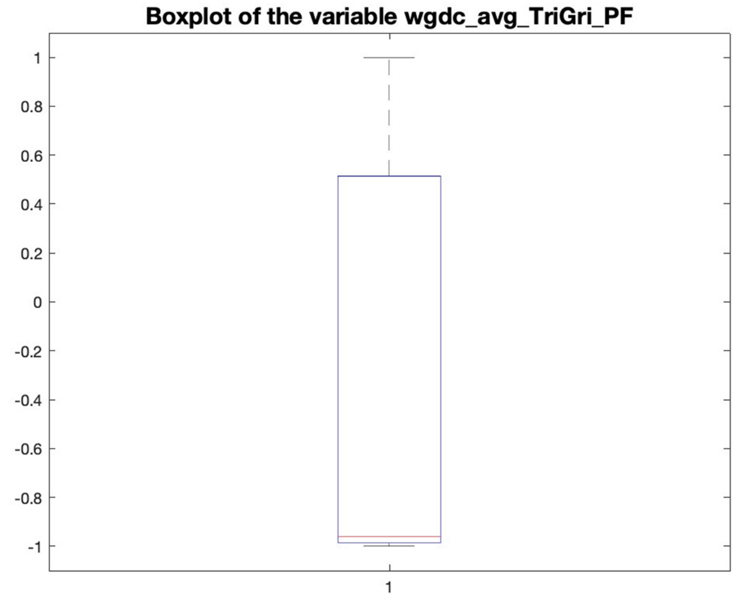

Fig. 3 At the top, the table summarises the statistics of the selected variable. At the bottom, the representation of this summary using a boxplot.

Variable Name	Description
wgdc_avg_TriGri_PwrAt	Transformer, grid side, active power
wgdc_avg_TriGri_PF	Transformer, grid side, power factor
wtrm_avg_TrmTmp_GbxOil	Temperature of the gearbox oil, in degrees Celsius.
wgen_avg_RtrSpd_WP2035	Speed of the rotor main shaft before gearbox in RPM
wnac_avg_WSpd1	Wind speed in m/s measured by the anemometer at the WT's nacelle
wtrm_avg_TrmTmp_GbxOil	Temperature of the gearbox oil
wnac_avg_ExlTmp	External temperature
wtrm_avg_TrmTmp_GbxBrg151	Speed of the rotor main shaft before gearbox in the point 151
wtrm_avg_TrmTmp_GbxOil	Temperature of the gearbox oil
Target Name	**Description**
wtrm_avg_TrmTmp_GbxBrg152	Temperature of the gearbox bearing 152, at the high speed shaft (output)

Table 5. List of variables selected to feed the ELM and the estimated target to monitor the gearbox subsystem.

In this example, we will illustrate how a part of the database can be used to develop a model that estimates a target variable within the gearbox subsystem based on a set of related variables. It is important to note that the gearbox has the longest downtime in the event of a failure. Despite the maturity and reliability of the manufacturing technology, this subsystem is prone to breakdowns and failures within a 5-year operating period due to the demanding operating conditions. Apart from the replacement costs, gearbox failure causes system downtime, which lengthens repair times, as it is one of the slowest systems to repair. The cost of gearbox replacement can be up to 14.5% of the maintenance cost of the wind turbine[28]. Consequently, gearbox failure prediction becomes a top priority, which is why it has been selected for this example.

Normality models aim to identify the point at which model predictions deviate from actual values, serving as a potential indicator of failure. In a previous study on ELMs applied to WT condition monitoring, the use of ELMs was explored[23,27]. Accordingly, we adopted the same network structure used in that work. The set of selected variables and the target variable for building the normality model can be found in Table 5. It should be noted that the target variable and the variables used remain consistent with those used in[23], although their application may differ. This section will show the same target variable as in the previous study.

Since the aim is to demonstrate the usefulness of the released database rather than to carry out an exhaustive study of it, we will focus on presenting the results for the WT84 turbine only. This particular turbine suffered a major failure and was out of service for a long period of time. Therefore, we will use the WT84 turbine to train a model and test it when it is working properly, as well as before the failure, in order to illustrate how simple

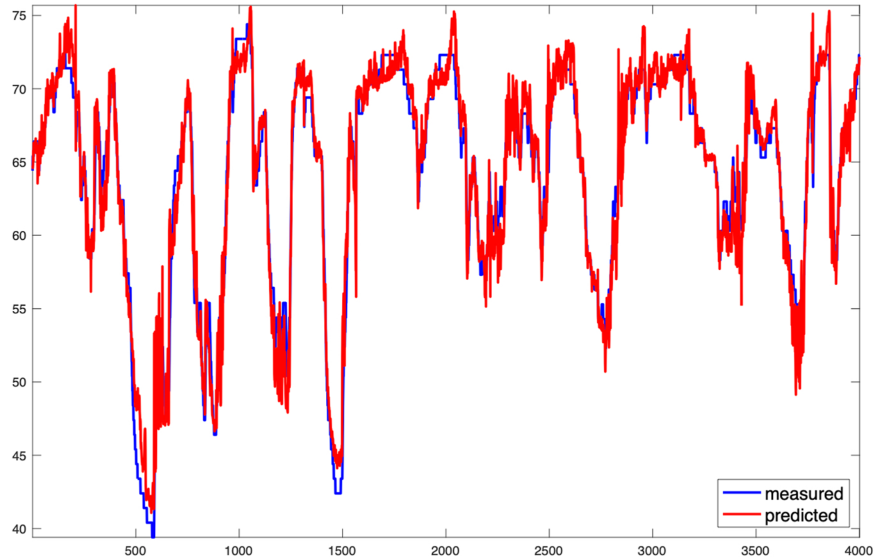

Fig. 4 Time interval of the test phase in which the model follows the signal. The horizontal axis represents the number of samples and the vertical axis the temperature in degrees Celsius.

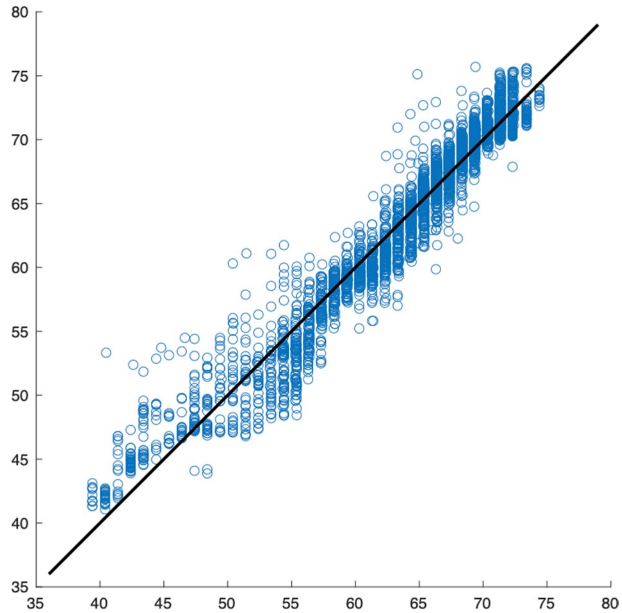

Fig. 5 Scatter plot of the above results, where the model follows the signal. The horizontal axis represents the measured signal and the vertical axis the estimated signal. The identity function is shown in black.

models can be used to monitor wind farms and often predict critical failures. This approach allows us to see how SCADA data can be used to detect the early stages of system deterioration.

Specifically, the model is trained using the first 25% of the initial WT84 data. The remaining data (75%) is used for various tests. The model consists of a feed-forward network with $H = 50$ hidden nodes and a sigmoid activation function, following the architecture presented in[23]. The model is not optimised and no formal feature selection is performed. The choice of variables is made intuitively based on knowledge generated in previous studies, as the problem was studied in[23]. The optimal network size (number of hidden nodes) is also not explored. The initial calculation of the ELM output weights is the solution presented, as different realisations yield similar results.

Since the ELM technique is robust to moderate outliers, no outlier filtering is performed in this example. The only processing applied to the raw data is a z-score normalisation of the variables and the target. The obtained normalisation constants (mean and variance) will be used to normalise and denormalise the test data.

Therefore, it is a simpler example compared to the one included in the repository (*example_elm.m*), and it can be performed using the methods of the implemented ELM class *elm_classifier.m*, also included in the repository, which facilitates the development and testing of the model.

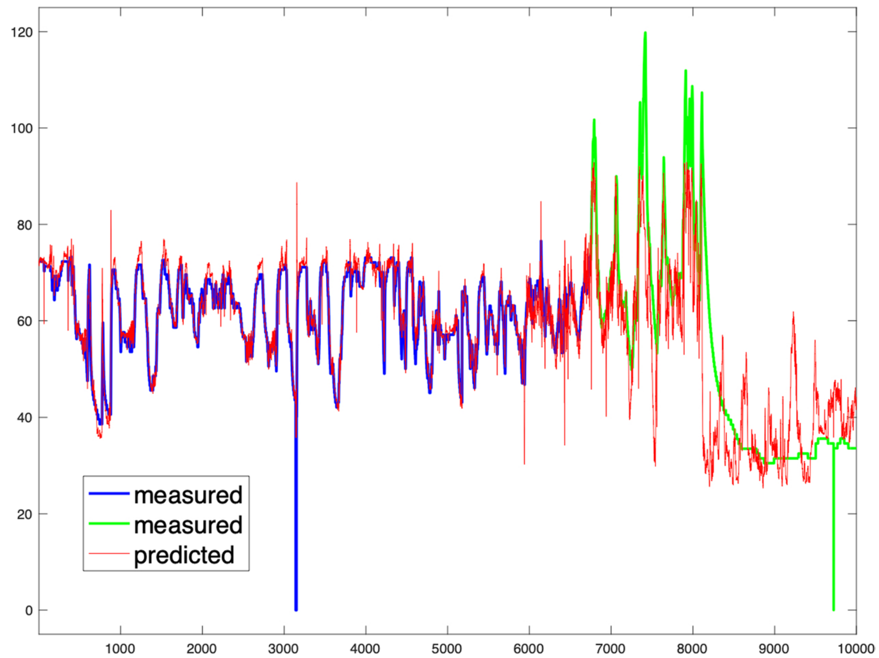

Fig. 6 Representation of the estimated and measured target within the time window that captures the onset of gear system malfunction and system failure. Note that the mismatch between the model and the system starts to be observed before sample 7000 (measured values depicted in blue color), while the failure occurs around sample 8000, after which the WT84 stops working and the target measurement tends toward room temperature values (measured values depicted in green color). Taking into account that each sample is spaced 5 minutes apart, the detection of anomalies 1000 samples (5000 minutes) before the failure indicates that the first signs could have been observed approximately 17 days before the event.

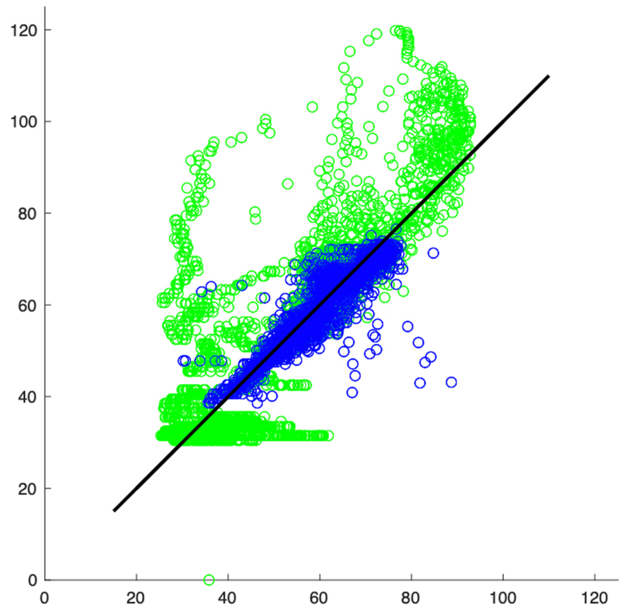

Fig. 7 Scatter plot of the above results, where the model starts to diverge of the signal. The horizontal axis represents the measured signal and the vertical axis the estimated signal. The identity function is shown in black. Blue dots correspond to the first 7000 points of the measured variable (shown in blue in Fig. 5), in which the WT84 works properly. Green points correspond to the last 3000 points (shown in green in Fig. 5) in which the WT84 had a failure. It can clearly be seen that these (green) measured values are far away from the values predicted by the ELM model.

The main difference from the *example_elm.m* model is our specific focus on WT84. We develop a dedicated model exclusively for this turbine, with a single iteration (unlike the multiple iterations used in the example to calculate output weights and identify the best solution). Additionally, we limit the training data to only the first 25% of the available data for WT84, obtained through Example 3, without incorporating any failure data. Since no optimization has been performed, we do not present accuracy measures or make other comparisons. In this context, our presentation includes a time segment during which WT84 operates correctly. Figure 4 displays this time segment, showcasing the model estimate alongside the actual measurements. Additionally, Fig. 5 presents the corresponding regression model for this segment. Furthermore, Fig. 6 illustrates the representation of the estimate and the target over a time interval spanning both before and after the gearbox failure. The failure caused a breakdown, leading to an extended period of WT84 being out of service. In the figure, the first 7000 samples, denoted in blue, represent the actual measured values. The remaining samples, depicted in green, correspond to the data points when WT84 was not functioning correctly.

For a clearer visualization of the disparity between the real measured target and the one predicted by the EML model, refer to Fig. 7. In this figure, the blue points represent the first 7000 samples, where the model produces an accurate estimate of the target. On the other hand, the green points correspond to the remaining samples, where the model noticeably deviates from the target due to the malfunction of WT84 during that period.

Code availability

The turbine dataset was generated by aggregating the SCADA data obtained from the entire wind farm. It consists of five wind turbines, all of them of the same model and manufacturer: Fuhrländer FL2500 2.5 MW. To facilitate the manipulation and pre-processing of the data, we have developed functions in the programming languages R and MATLAB to serve as an interface. These functions efficiently transform the raw data into a structured table format. In this format, each variable corresponds to a column, while each entry represents a five-minute interval of data recorded in the rows. The database and the code are freely available at[25] and at the GitHub page https://github.com/alecuba16/fuhrlander.

Received: 11 July 2023; Accepted: 13 February 2024;
Published online: 29 February 2024

References

1. Eurostat. Energy balance sheets 2011–2012, https://doi.org/10.2785/52802 (2014).
2. Milborrow, D. Operation and maintenance costs compared and revealed. *Wind Stats* **19**, 3 (2006).
3. Lei, X., Sandborn, P., Bakhshi, R., Kashani-Pour, A. & Goudarzi, N. Phm based predictive maintenance optimization for offshore wind farms. In *2015 IEEE conference on prognostics and health management (PHM)*, 1–8, https://doi.org/10.1109/icphm.2015.7245027 (IEEE, 2015).
4. Hahn, B., Durstewitz, M. & Rohrig, K. Reliability of wind turbines. In *Wind Energy: Proceedings of the Euromech Colloquium*, 329–332, https://doi.org/10.1007/978-3-540-33866-6_62 (Springer, 2007).
5. Ziegler, L., Gonzalez, E., Rubert, T., Smolka, U. & Melero, J. J. Lifetime extension of onshore wind turbines: A review covering germany, spain, denmark, and the uk. *Renewable and Sustainable Energy Reviews* **82**, 1261–1271, https://doi.org/10.1016/j.rser.2017.09.100 (2018).
6. Ding, F., Tian, Z., Zhao, F. & Xu, H. An integrated approach for wind turbine gearbox fatigue life prediction considering instantaneously varying load conditions. *Renewable energy* **129**, 260–270, https://doi.org/10.1016/j.renene.2018.05.074 (2018).
7. Corley, B., Carroll, J. & Mcdonald, A. Fault detection of wind turbine gearbox using thermal network modelling and scada data. In *Journal of Physics: Conference Series*, vol. 1618, 022042, https://doi.org/10.1088/1742-6596/1618/2/022042 (IOP Publishing, 2020).
8. Corley, B., Koukoura, S., Carroll, J. & McDonald, A. Combination of thermal modelling and machine learning approaches for fault detection in wind turbine gearboxes. *Energies* **14**, 1375, https://doi.org/10.3390/en14051375 (2021).
9. Soua, S., Van Lieshout, P., Perera, A., Gan, T.-H. & Bridge, B. Determination of the combined vibrational and acoustic emission signature of a wind turbine gearbox and generator shaft in service as a pre-requisite for effective condition monitoring. *Renewable Energy* **51**, 175–181, https://doi.org/10.1016/j.renene.2012.07.004 (2013).
10. Siegel, D., Zhao, W., Lapira, E., AbuAli, M. & Lee, J. A comparative study on vibration-based condition monitoring algorithms for wind turbine drive trains. *Wind energy* **17**, 695–714, https://doi.org/10.1002/we.1585 (2014).
11. Artigao, E. *et al*. Current signature and vibration analyses to diagnose an in-service wind turbine drive train. *Energies* **11**, 960, https://doi.org/10.3390/en11040960 (2018).
12. Zheng, C. Temporal-spatial characteristics of offshore wind energy resource dataset for the 21st century maritime silk road. *Science Data Bank* https://doi.org/10.11922/sciencedb.j00001.00142 (2020).
13. Zhang, T., Tian, B., Sengupta, D., Zhang, L. & Si, Y. Global offshore wind turbine dataset. *Scientific Data* **8**, 191, https://doi.org/10.1038/s41597-021-00982-z (2021).
14. Effenberger, N. & Ludwig, N. A collection and categorization of open-source wind and wind power datasets. *Wind Energy* **25**, 1659–1683, https://doi.org/10.1002/we.2766 (2022).
15. Lu, B., Durocher, D. B. & Stemper, P. Predictive maintenance techniques. *IEEE Industry Applications Magazine* **15**, 52–60, https://doi.org/10.1109/mias.2009.934444 (2009).
16. Márquez, F. P. G., Tobias, A. M., Pérez, J. M. P. & Papaelias, M. Condition monitoring of wind turbines: Techniques and methods. *Renewable energy* **46**, 169–178, https://doi.org/10.1016/j.renene.2012.03.003 (2012).
17. Tautz-Weinert, J. & 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).
18. Stetco, A. *et al*. 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).
19. Yang, W., Court, R. & Jiang, J. Wind turbine condition monitoring by the approach of scada data analysis. *Renewable energy* **53**, 365–376, https://doi.org/10.1016/j.renene.2012.11.030 (2013).
20. Zaher, A., McArthur, S., Infield, D. & Patel, Y. Online wind turbine fault detection through automated scada data analysis. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* **12**, 574–593, https://doi.org/10.1002/we.319 (2009).
21. Gonzalez, E., Stephen, B., Infield, D. & Melero, J. J. Using high-frequency scada data for wind turbine performance monitoring: A sensitivity study. *Renewable energy* **131**, 841–853, https://doi.org/10.1016/j.renene.2018.07.068 (2019).
22. Marti-Puig, P., Blanco-M, A., Cárdenas, J. J., Cusidó, J. & Solé-Casals, J. Feature selection algorithms for wind turbine failure prediction. *Energies* **12**, 453, https://doi.org/10.3390/en12030453 (2019).
23. Marti-Puig, P., Blanco-M, A., Serra-Serra, M. & Solé-Casals, J. Wind turbine prognosis models based on scada data and extreme learning machines. *Applied Sciences* **11**, 590, https://doi.org/10.3390/app11020590 (2021).

24. Marti-Puig, P. *et al*. Detection of wind turbine failures through cross-information between neighbouring turbines. *Applied Sciences* **12**, 9491, https://doi.org/10.3390/app12199491 (2022).
25. Martínez, A. B., Solé-Casals, J. & Marti-Puig, P. Fuhrländer FL2500 2.5 MW wind turbine dataset., *Figshare*, https://doi.org/10.6084/m9.figshare.25201631.v1 (2024).
26. *Wind Park Control Concept Manual*. https://www.mita-teknik.com/media/1141/binder1.pdf.
27. Marti-Puig, P., Bennásar-Sevillá, A., Blanco-M, A. & Solé-Casals, J. Exploring the effect of temporal aggregation on scada data for wind turbine prognosis using a normality model. *Applied Sciences* **11**, 6405, https://doi.org/10.3390/app11146405 (2021).
28. Pérez, J. M. P., Márquez, F. P. G., Tobias, A. & Papaelias, M. Wind turbine reliability analysis. *Renewable and Sustainable Energy Reviews* **23**, 463–472, https://doi.org/10.1016/j.rser.2013.03.018 (2013).

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

Author contributions
Conceptualization, P.M.-P. and J.S.-C.; Data collection, J.C. and A.B.-M.; Experiments, P.M.-P., J.S.-C. and A.B.-M.; Writing - original draft, P.M.-P. and J.S.-C.; Writing - review and editing, all authors; Project administration, J.C. and J.S.-C.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41597-024-03067-9.

Correspondence and requests for materials should be addressed to P.M.-P. or J.S.-C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Supplementary material

Contents

1 Systems and subsystems

Table 1 contains the list of the systems and subsystems of the wind turbines. Note that some systems do not have subsystems while others have several. This table is derived directly from the SCADA database.

SYS	SUBS
Converter	Converter
Generator	Generator
Nacelle	Control Cabinet
Rotor blade	Control Cabinet Hub Pitch Rotor
Tower	Control Cabinet Tower
Transformer	Transformer
Transmission	Brake Gearbox Hydraulic System Main Bearing
Turbine	Control Cabinet Power Cabinet Roof
Yaw	Yaw

Table 1. Systems and the subsystems of the wind turbines.

2 Exhaustive signals list provided by the SCADA system

Table 2 contains all the signals recorded by the SCADA system. The first is the system identification, the second column is the variable group, the third column contains the statistics type of the variable, and the last column provides the signal name.

System	Variable Group	Stat Type	Signal Name
ID	id	single	turbine_id
TIME	time	single	date_time
WMET	wmet_MetAlt1_Hum	min	wmet_min_MetAlt1_Hum
		avg	wmet_avg_MetAlt1_Hum
		sdv	wmet_sdv_MetAlt1_Hum
		max	wmet_max_MetAlt1_Hum
	wmet_DewPTmp	min	wmet_min_DewPTmp
		avg	wmet_avg_DewPTmp
		sdv	wmet_sdv_DewPTmp
		max	wmet_max_DewPTmp
	wmet_MetAlt1_Press	min	wmet_min_MetAlt1_Press
		avg	wmet_avg_MetAlt1_Press

		sdv	wmet_sdv_MetAlt1_Press
		max	wmet_max_MetAlt1_Press
WGDC	wgdc_TriGri_PF	min	wgdc_min_TriGri_PF
		avg	wgdc_avg_TriGri_PF
		sdv	wgdc_sdv_TriGri_PF
		max	wgdc_max_TriGri_PF
	wgdc_TriGri_A_phsC	min	wgdc_min_TriGri_A_phsC
		avg	wgdc_avg_TriGri_A_phsC
		sdv	wgdc_sdv_TriGri_A_phsC
		max	wgdc_max_TriGri_A_phsC
	wgdc_LoVTmp	min	wgdc_min_LoVTmp
		avg	wgdc_avg_LoVTmp
		sdv	wgdc_sdv_LoVTmp
		max	wgdc_max_LoVTmp
	wgdc_TriGri_PwrAt	min	wgdc_min_TriGri_PwrAt
		avg	wgdc_avg_TriGri_PwrAt
		sdv	wgdc_sdv_TriGri_PwrAt
		max	wgdc_max_TriGri_PwrAt
	wgdc_GdcTmp_TrfGn	min	wgdc_min_GdcTmp_TrfGn
		avg	wgdc_avg_GdcTmp_TrfGn
		sdv	wgdc_sdv_GdcTmp_TrfGn
		max	wgdc_max_GdcTmp_TrfGn
	wgdc_TriGri_PhV_phsA	min	wgdc_min_TriGri_PhV_phsA
		avg	wgdc_avg_TriGri_PhV_phsA
		sdv	wgdc_sdv_TriGri_PhV_phsA
		max	wgdc_max_TriGri_PhV_phsA
	wgdc_TriGri_PwrReact	min	wgdc_min_TriGri_PwrReact
		avg	wgdc_avg_TriGri_PwrReact
		sdv	wgdc_sdv_TriGri_PwrReact
		max	wgdc_max_TriGri_PwrReact
	wgdc_TriGri_PhV_phsB	min	wgdc_min_TriGri_PhV_phsB
		avg	wgdc_avg_TriGri_PhV_phsB
		sdv	wgdc_sdv_TriGri_PhV_phsB
		max	wgdc_max_TriGri_PhV_phsB
	wgdc_TriGri_Hz	min	wgdc_min_TriGri_Hz
		avg	wgdc_avg_TriGri_Hz
		sdv	wgdc_sdv_TriGri_Hz
		max	wgdc_max_TriGri_Hz
	wgdc_TriGri_A_phsB	min	wgdc_min_TriGri_A_phsB
		avg	wgdc_avg_TriGri_A_phsB
		sdv	wgdc_sdv_TriGri_A_phsB
		max	wgdc_max_TriGri_A_phsB
	wgdc_TriGri_A_phsA	min	wgdc_min_TriGri_A_phsA
		avg	wgdc_avg_TriGri_A_phsA
		sdv	wgdc_sdv_TriGri_A_phsA
		max	wgdc_max_TriGri_A_phsA
	wgdc_TriGri_PhV_phsC	min	wgdc_min_TriGri_PhV_phsC
		avg	wgdc_avg_TriGri_PhV_phsC
		sdv	wgdc_sdv_TriGri_PhV_phsC
		max	wgdc_max_TriGri_PhV_phsC
	wgdc_TriGri_PhV	min	wgdc_min_TriGri_PhV
		avg	wgdc_avg_TriGri_PhV
		sdv	wgdc_sdv_TriGri_PhV
		max	wgdc_max_TriGri_PhV
	wgdc_TriGri_A	min	wgdc_min_TriGri_A

		avg	wgdc_avg_TriGri_A
		sdv	wgdc_sdv_TriGri_A
		max	wgdc_max_TriGri_A
WTOW	wtow_PwrPnlTmp	min	wtow_min_PwrPnlTmp
		avg	wtow_avg_PwrPnlTmp
		sdv	wtow_sdv_PwrPnlTmp
		max	wtow_max_PwrPnlTmp
WTRM	wtrm_Gbx_OilPres	min	wtrm_min_Gbx_OilPres
		avg	wtrm_avg_Gbx_OilPres
		sdv	wtrm_sdv_Gbx_OilPres
		max	wtrm_max_Gbx_OilPres
	wtrm_TrmTmp_GnBrgNDE	min	wtrm_min_TrmTmp_GnBrgNDE
		avg	wtrm_avg_TrmTmp_GnBrgNDE
		sdv	wtrm_sdv_TrmTmp_GnBrgNDE
		max	wtrm_max_TrmTmp_GnBrgNDE
	wtrm_TrmTmp_Brg1	min	wtrm_min_TrmTmp_Brg1
		avg	wtrm_avg_TrmTmp_Brg1
		sdv	wtrm_sdv_TrmTmp_Brg1
		max	wtrm_max_TrmTmp_Brg1
	wtrm_TrmTmp_GbxBrg450	min	wtrm_min_TrmTmp_GbxBrg450
		avg	wtrm_avg_TrmTmp_GbxBrg450
		sdv	wtrm_sdv_TrmTmp_GbxBrg450
		max	wtrm_max_TrmTmp_GbxBrg450
	wtrm_TrmTmp_GbxClWtBkw	min	wtrm_min_TrmTmp_GbxClWtBkw
		avg	wtrm_avg_TrmTmp_GbxClWtBkw
		sdv	wtrm_sdv_TrmTmp_GbxClWtBkw
		max	wtrm_max_TrmTmp_GbxClWtBkw
	wtrm_TrmTmp_GbxClWtFrw	min	wtrm_min_TrmTmp_GbxClWtFrw
		avg	wtrm_avg_TrmTmp_GbxClWtFrw
		sdv	wtrm_sdv_TrmTmp_GbxClWtFrw
		max	wtrm_max_TrmTmp_GbxClWtFrw
	wtrm_TrmTmp_GbxBrg451	min	wtrm_min_TrmTmp_GbxBrg451
		avg	wtrm_avg_TrmTmp_GbxBrg451
		sdv	wtrm_sdv_TrmTmp_GbxBrg451
		max	wtrm_max_TrmTmp_GbxBrg451
	wtrm_Brg_OilPresIn	min	wtrm_min_Brg_OilPresIn
		avg	wtrm_avg_Brg_OilPresIn
		sdv	wtrm_sdv_Brg_OilPresIn
		max	wtrm_max_Brg_OilPresIn
	wtrm_TrmTmp_GnBrgDE	min	wtrm_min_TrmTmp_GnBrgDE
		avg	wtrm_avg_TrmTmp_GnBrgDE
		sdv	wtrm_sdv_TrmTmp_GnBrgDE
		max	wtrm_max_TrmTmp_GnBrgDE
	wtrm_TrmTmp_GbxBrg452	min	wtrm_min_TrmTmp_GbxBrg452
		avg	wtrm_avg_TrmTmp_GbxBrg452
		sdv	wtrm_sdv_TrmTmp_GbxBrg452
		max	wtrm_max_TrmTmp_GbxBrg452
	wtrm_TrmTmp_Brg2	min	wtrm_min_TrmTmp_Brg2
		avg	wtrm_avg_TrmTmp_Brg2
		sdv	wtrm_sdv_TrmTmp_Brg2
		max	wtrm_max_TrmTmp_Brg2
	wtrm_TrmTmp_GbxOil	min	wtrm_min_TrmTmp_GbxOil
		avg	wtrm_avg_TrmTmp_GbxOil
		sdv	wtrm_sdv_TrmTmp_GbxOil

		max	wtrm_max_TrmTmp_GbxOil
	wtrm_TrmTmp_GnClWtFrw	min	wtrm_min_TrmTmp_GnClWtFrw
		avg	wtrm_avg_TrmTmp_GnClWtFrw
		sdv	wtrm_sdv_TrmTmp_GnClWtFrw
		max	wtrm_max_TrmTmp_GnClWtFrw
	wtrm_TrmTmp_GbxBrg151	min	wtrm_min_TrmTmp_GbxBrg151
		avg	wtrm_avg_TrmTmp_GbxBrg151
		sdv	wtrm_sdv_TrmTmp_GbxBrg151
		max	wtrm_max_TrmTmp_GbxBrg151
	wtrm_TrmTmp_Gbx	min	wtrm_min_TrmTmp_Gbx
		avg	wtrm_avg_TrmTmp_Gbx
		sdv	wtrm_sdv_TrmTmp_Gbx
		max	wtrm_max_TrmTmp_Gbx
	wtrm_TrmTmp_GbxBrg152	min	wtrm_min_TrmTmp_GbxBrg152
		avg	wtrm_avg_TrmTmp_GbxBrg152
		sdv	wtrm_sdv_TrmTmp_GbxBrg152
		max	wtrm_max_TrmTmp_GbxBrg152
	wtrm_TrmTmp_GnClWtBkw	min	wtrm_min_TrmTmp_GnClWtBkw
		avg	wtrm_avg_TrmTmp_GnClWtBkw
		sdv	wtrm_sdv_TrmTmp_GnClWtBkw
		max	wtrm_max_TrmTmp_GnClWtBkw
	wtrm_Brg_OilPres	min	wtrm_min_Brg_OilPres
		avg	wtrm_avg_Brg_OilPres
		sdv	wtrm_sdv_Brg_OilPres
		max	wtrm_max_Brg_OilPres
WTUR	wtur_PwrRedCau	min	wtur_min_PwrRedCau
		avg	wtur_avg_PwrRedCau
		sdv	wtur_sdv_PwrRedCau
		max	wtur_max_PwrRedCau
	wtur_PwrRedSp	min	wtur_min_PwrRedSp
		avg	wtur_avg_PwrRedSp
		sdv	wtur_sdv_PwrRedSp
		max	wtur_max_PwrRedSp
	wtur_PwrRedNoi	min	wtur_min_PwrRedNoi
		avg	wtur_avg_PwrRedNoi
		sdv	wtur_sdv_PwrRedNoi
		max	wtur_max_PwrRedNoi
	wtur_ExtPwrSpUtil	min	wtur_min_ExtPwrSpUtil
		avg	wtur_avg_ExtPwrSpUtil
		sdv	wtur_sdv_ExtPwrSpUtil
		max	wtur_max_ExtPwrSpUtil
	wtur_ExtPwrReactSp	min	wtur_min_ExtPwrReactSp
		avg	wtur_avg_ExtPwrReactSp
		sdv	wtur_sdv_ExtPwrReactSp
		max	wtur_max_ExtPwrReactSp
WROT	wrot_TmpPwrSply_ValBl2	min	wrot_min_TmpPwrSply_ValBl2
		avg	wrot_avg_TmpPwrSply_ValBl2
		sdv	wrot_sdv_TmpPwrSply_ValBl2
		max	wrot_max_TmpPwrSply_ValBl2
	wrot_TmpHtSinkPco_Bl2	min	wrot_min_TmpHtSinkPco_Bl2
		avg	wrot_avg_TmpHtSinkPco_Bl2
		sdv	wrot_sdv_TmpHtSinkPco_Bl2
		max	wrot_max_TmpHtSinkPco_Bl2
	wrot_TmpCpt_Bl3	min	wrot_min_TmpCpt_Bl3
		avg	wrot_avg_TmpCpt_Bl3

		sdv	wrot_sdv_TmpCpt_Bl3
		max	wrot_max_TmpCpt_Bl3
	wrot_A_ValBl2	min	wrot_min_A_ValBl2
		avg	wrot_avg_A_ValBl2
		sdv	wrot_sdv_A_ValBl2
		max	wrot_max_A_ValBl2
	wrot_TmpHtSinkPco_Bl1	min	wrot_min_TmpHtSinkPco_Bl1
		avg	wrot_avg_TmpHtSinkPco_Bl1
		sdv	wrot_sdv_TmpHtSinkPco_Bl1
		max	wrot_max_TmpHtSinkPco_Bl1
	wrot_TmpCpt_Bl2	min	wrot_min_TmpCpt_Bl2
		avg	wrot_avg_TmpCpt_Bl2
		sdv	wrot_sdv_TmpCpt_Bl2
		max	wrot_max_TmpCpt_Bl2
	wrot_A_ValBl3	min	wrot_min_A_ValBl3
		avg	wrot_avg_A_ValBl3
		sdv	wrot_sdv_A_ValBl3
		max	wrot_max_A_ValBl3
	wrot_TmpPwrSply_ValBl1	min	wrot_min_TmpPwrSply_ValBl1
		avg	wrot_avg_TmpPwrSply_ValBl1
		sdv	wrot_sdv_TmpPwrSply_ValBl1
		max	wrot_max_TmpPwrSply_ValBl1
	wrot_A_ValBl1	min	wrot_min_A_ValBl1
		avg	wrot_avg_A_ValBl1
		sdv	wrot_sdv_A_ValBl1
		max	wrot_max_A_ValBl1
	wrot_V_ValBl2	min	wrot_min_V_ValBl2
		avg	wrot_avg_V_ValBl2
		sdv	wrot_sdv_V_ValBl2
		max	wrot_max_V_ValBl2
	wrot_V_ValBl1	min	wrot_min_V_ValBl1
		avg	wrot_avg_V_ValBl1
		sdv	wrot_sdv_V_ValBl1
		max	wrot_max_V_ValBl1
	wrot_TmpHtSinkPco_Bl3	min	wrot_min_TmpHtSinkPco_Bl3
		avg	wrot_avg_TmpHtSinkPco_Bl3
		sdv	wrot_sdv_TmpHtSinkPco_Bl3
		max	wrot_max_TmpHtSinkPco_Bl3
	wrot_TmpPDU_ValBl2	min	wrot_min_TmpPDU_ValBl2
		avg	wrot_avg_TmpPDU_ValBl2
		sdv	wrot_sdv_TmpPDU_ValBl2
		max	wrot_max_TmpPDU_ValBl2
	wrot_RotSt_Bl1_PDU	min	wrot_min_RotSt_Bl1_PDU
		avg	wrot_avg_RotSt_Bl1_PDU
		sdv	wrot_sdv_RotSt_Bl1_PDU
		max	wrot_max_RotSt_Bl1_PDU
	wrot_TmpCpt_Bl1	min	wrot_min_TmpCpt_Bl1
		avg	wrot_avg_TmpCpt_Bl1
		sdv	wrot_sdv_TmpCpt_Bl1
		max	wrot_max_TmpCpt_Bl1
	wrot_TmpPDU_ValBl3	min	wrot_min_TmpPDU_ValBl3
		avg	wrot_avg_TmpPDU_ValBl3
		sdv	wrot_sdv_TmpPDU_ValBl3
		max	wrot_max_TmpPDU_ValBl3
	wrot_V_ValBl3	min	wrot_min_V_ValBl3

		avg	wrot_avg_V_ValBl3
		sdv	wrot_sdv_V_ValBl3
		max	wrot_max_V_ValBl3
	wrot_TmpPwrSply_ValBl3	min	wrot_min_TmpPwrSply_ValBl3
		avg	wrot_avg_TmpPwrSply_ValBl3
		sdv	wrot_sdv_TmpPwrSply_ValBl3
		max	wrot_max_TmpPwrSply_ValBl3
WGEN	wgen_GnTmp_phsA	min	wgen_min_GnTmp_phsA
		avg	wgen_avg_GnTmp_phsA
		sdv	wgen_sdv_GnTmp_phsA
		max	wgen_max_GnTmp_phsA
	wgen_GnTmp_phsB	min	wgen_min_GnTmp_phsB
		avg	wgen_avg_GnTmp_phsB
		sdv	wgen_sdv_GnTmp_phsB
		max	wgen_max_GnTmp_phsB
	wgen_Spd	min	wgen_min_Spd
		avg	wgen_avg_Spd
		sdv	wgen_sdv_Spd
		max	wgen_max_Spd
	wgen_RtrSpd_WP2035	min	wgen_min_RtrSpd_WP2035
		avg	wgen_avg_RtrSpd_WP2035
		sdv	wgen_sdv_RtrSpd_WP2035
		max	wgen_max_RtrSpd_WP2035
	wgen_GnTmp_phsC	min	wgen_min_GnTmp_phsC
		avg	wgen_avg_GnTmp_phsC
		sdv	wgen_sdv_GnTmp_phsC
		max	wgen_max_GnTmp_phsC
	wgen_RtrSpd_IGR	min	wgen_min_RtrSpd_IGR
		avg	wgen_avg_RtrSpd_IGR
		sdv	wgen_sdv_RtrSpd_IGR
		max	wgen_max_RtrSpd_IGR
WNAC	wnac_WVaneDir1	min	wnac_min_WVaneDir1
		avg	wnac_avg_WVaneDir1
		sdv	wnac_sdv_WVaneDir1
		max	wnac_max_WVaneDir1
	wnac_WSpd2	min	wnac_min_WSpd2
		avg	wnac_avg_WSpd2
		sdv	wnac_sdv_WSpd2
		max	wnac_max_WSpd2
	wnac_Dir	min	wnac_min_Dir
		avg	wnac_avg_Dir
		sdv	wnac_sdv_Dir
		max	wnac_max_Dir
	wnac_Wdir1	min	wnac_min_Wdir1
		avg	wnac_avg_Wdir1
		sdv	wnac_sdv_Wdir1
		max	wnac_max_Wdir1
	wnac_WVaneDir2	min	wnac_min_WVaneDir2
		avg	wnac_avg_WVaneDir2
		sdv	wnac_sdv_WVaneDir2
		max	wnac_max_WVaneDir2
	wnac_WSpd1	min	wnac_min_WSpd1
		avg	wnac_avg_WSpd1
		sdv	wnac_sdv_WSpd1

		max	wnac_max_WSpd1
	wnac_Wdir2	min	wnac_min_Wdir2
		avg	wnac_avg_Wdir2
		sdv	wnac_sdv_Wdir2
		max	wnac_max_Wdir2
	wnac_ExlTmp	min	wnac_min_ExlTmp
		avg	wnac_avg_ExlTmp
		sdv	wnac_sdv_ExlTmp
		max	wnac_max_ExlTmp
	wnac_NacTmp	min	wnac_min_NacTmp
		avg	wnac_avg_NacTmp
		sdv	wnac_sdv_NacTmp
		max	wnac_max_NacTmp
WCNV	wcnv_InvTmp_ClWtrFwd	min	wcnv_min_InvTmp_ClWtrFwd
		avg	wcnv_avg_InvTmp_ClWtrFwd
		sdv	wcnv_sdv_InvTmp_ClWtrFwd
		max	wcnv_max_InvTmp_ClWtrFwd
	wcnv_IGBTTmp	min	wcnv_min_IGBTTmp
		avg	wcnv_avg_IGBTTmp
		sdv	wcnv_sdv_IGBTTmp
		max	wcnv_max_IGBTTmp
	wcnv_InvTmp_ClWtrRet	min	wcnv_min_InvTmp_ClWtrRet
		avg	wcnv_avg_InvTmp_ClWtrRet
		sdv	wcnv_sdv_InvTmp_ClWtrRet
		max	wcnv_max_InvTmp_ClWtrRet
	wcnv_HtSnkTmp	min	wcnv_min_HtSnkTmp
		avg	wcnv_avg_HtSnkTmp
		sdv	wcnv_sdv_HtSnkTmp
		max	wcnv_max_HtSnkTmp

Table 2. Complete variable list divided into systems and variable groups. Each variable group contains four statistical indicators, min/max/sdv/avg calculated from higher frequency sensors and summarized every 5 minutes.

3 Description of the alarms

Table 3 contains the list of alarms and warnings, which are mixed together in the database. Therefore we will use the name 'alarm' to refer to the set of alarms and warnings. The alarms are tagged by a numerical identifier (ID). There is a total of 369 different alarms. Each alarm is associated with a main system (SYS) and a subsystem in the system (SUBS). Moreover, there is a short description of each alarm (DESC).

ID	SYS	SUBS	DESC
0	Turbine	Control Cabinet	System OK
5	Turbine	Control Cabinet	Vibration
7	Turbine	Control Cabinet	Turbine is serviced
9	Turbine	Control Cabinet	Remote stop
13	Turbine	Control Cabinet	Manual stop
16	Turbine	Control Cabinet	Emer.stop cont.panel
23	Turbine	Control Cabinet	Repeating error
30	Nacelle	Control Cabinet	Nacelle temp.
31	Nacelle	Control Cabinet	Nacelle temp. stop
41	Turbine	Control Cabinet	UPS battery low
45	Turbine	Power Cabinet	Main ctrl. Supply
55	Turbine	Control Cabinet	Main ctrl.man.reboot
66	Turbine	Control Cabinet	Fire alarm
93	Turbine	Control Cabinet	Service hatch
100	Turbine	Control Cabinet	Repeated grid error

102	Turbine	Control Cabinet	Phase drop
103	Turbine	Control Cabinet	Vector surge
110	Turbine	Control Cabinet	Voltage high
111	Turbine	Control Cabinet	Voltage low
113	Transformer	Transformer	Trafo overtemp.
114	Transformer	Transformer	Trafo temp. stop
120	Turbine	Control Cabinet	Frequency high
121	Turbine	Control Cabinet	Frequency low
128	Turbine	Control Cabinet	Transient grid error
130	Turbine	Control Cabinet	L1-L2-L3 120
134	Turbine	Control Cabinet	Critical frequency
154	Transformer	Transformer	Trafo min. temp.
155	Transformer	Transformer	Trafo min temp. stop
200	Nacelle	Control Cabinet	Outdoor temp. low
201	Nacelle	Control Cabinet	Outdoor temp. high
202	Turbine	Control Cabinet	Wind < power
203	Turbine	Control Cabinet	Wind > power
205	Tower	Control Cabinet	Tower resonance time
206	Rotor	Control Cabinet	Ice warning
220	Turbine	Roof	Diff WindSpeedSens>SHH
221	Turbine	Roof	W.direc.nonidentical
226	Turbine	Roof	WindSpeedSensorsDefect
230	Turbine	Roof	WindSpeedSens1 defect
231	Turbine	Roof	WindSpeedSens2 defect
233	Turbine	Control Cabinet	WindSpeed<StartCond
235	Turbine	Control Cabinet	Wind vanes defect
236	Turbine	Control Cabinet	Out.temp low stop
237	Turbine	Control Cabinet	Out.temp hi. Stop
238	Turbine	Control Cabinet	Anemo.test rpm high
239	Turbine	Control Cabinet	Anemo.test rpm low
241	Turbine	Control Cabinet	Weather sensor com.
243	Turbine	Control Cabinet	Light.prot. term.box
250	Turbine	Control Cabinet	WindSpeed > SH
251	Turbine	Control Cabinet	WindSpeed > SHH
260	Turbine	Control Cabinet	Wind vane 1 defect
261	Turbine	Control Cabinet	Wind vane 2 defect
265	Turbine	Control Cabinet	Weather #1 Com.
266	Turbine	Control Cabinet	Weather #1
267	Turbine	Control Cabinet	Weather #1 Iced
320	Turbine	Control Cabinet	WP2035 (R) overspeed
346	Turbine	Control Cabinet	(G)oversp. operation
347	Turbine	Control Cabinet	Max. service (G)rpm
348	Turbine	Control Cabinet	WP2035 com. timeout
349	Turbine	Control Cabinet	WP2035 com.pack.err
350	Turbine	Control Cabinet	WP2035 <> rotor rpm
414	Transmission	Brake	MBS Not Open
415	Transmission	Brake	MBS Pads Wear
416	Transmission	Brake	MBS Pads Wear Warn
431	Generator	Generator	BP50 GenSpeedRed<min
474	Turbine	Control Cabinet	(B) air press. low
500	Generator	Generator	Gen Repeat Error
530	Turbine	Control Cabinet	ActivePower > SHH
531	Generator	Generator	Gen TempCoil L1 > SH
533	Generator	Generator	Gen TempCoil L2 > SH
535	Generator	Generator	Gen TempCoil L3 > SH

554	Generator	Generator	Gen WearBrush Warn
562	Generator	Generator	Gen TempCoolWatRet>SHH
601	Turbine	Control Cabinet	Current asymmetry
613	Transformer	Transformer	Trafo oil press.high
614	Transformer	Transformer	Trafo oil press.stop
640	Rotor	Hub	Lightningprot.hub
663	Transformer	Transformer	Transformer leaking
700	Yaw	Yaw	Error by yawing
701	Yaw	Yaw	Rep. error by yawing
708	Yaw	Yaw	Twisted CCW
709	Yaw	Yaw	Twisted CW
715	Yaw	Yaw	Cable autounwind
719	Yaw	Yaw	(H)yaw(M)therm.relay
730	Yaw	Yaw	Yaw sensor defect
731	Yaw	Yaw	Yaw(P) starting rate
732	Yaw	Yaw	Manual yaw active
733	Yaw	Yaw	Yaw(P) oper.time>max
734	Turbine	Control Cabinet	Nacel.pos<>wind vane
742	Yaw	Yaw	Yaw sensor A/B
743	Yaw	Yaw	Yaw thermal relay
760	Yaw	Yaw	Yaw misalignment
770	Yaw	Yaw	Yaw error inverter 1
771	Yaw	Yaw	Yaw error inverter 2
772	Yaw	Yaw	Yaw error inverter 3
773	Yaw	Yaw	Yaw error inverter 4
775	Yaw	Yaw	LubYaw Rim Time-out
776	Yaw	Yaw	LubYawBear Time-out
779	Yaw	Yaw	LubYaw Rim GreaseEmpty
780	Yaw	Yaw	LubYawBearGreaseEmpty
904	Generator	Generator	Gen FuseTripCoolWatPp
907	Nacelle	Control Cabinet	Therm. nacelle fan
916	Transmission	Gearbox	MGB Temp CoolWater > SHH
944	Transmission	Main Bearing	MGB PressSwitchCoolWat
963	Transmission	Gearbox	MGB RepeatCoolWatError
964	Transmission	Main Bearing	MGB FuseTrip OilPump
1016	Rotor	Pitch	Ethcan pitch com
1017	Rotor	Pitch	CANopen pitch com
1018	Rotor	Pitch	Ethcan inv. Com
1019	Rotor	Pitch	CANopen inv. Com
1020	Rotor	Pitch	Pitch 1 CANopen
1021	Rotor	Pitch	Pitch 2 CANopen
1022	Rotor	Pitch	Pitch 3 CANopen
1023	Rotor	Pitch	Inverter CANopen
1024	Rotor	Pitch	Ethcan pitch 1 emcy.
1025	Rotor	Pitch	Ethcan pitch 2 emcy.
1026	Rotor	Pitch	Ethcan pitch 3 emcy.
1027	Rotor	Pitch	Ethcan inv. emcy.
1028	Rotor	Pitch	Ethnet pitch receive
1029	Rotor	Pitch	Ethnet inv. receive
1030	Rotor	Pitch	Ethnet pitch send
1031	Rotor	Pitch	Ethnet inv. send
1113	Rotor	Pitch	PLU DeltaPitchAngle>SHH
1207	Transmission	Main Bearing	MBS FuseTrip HydOilPp
1210	Transmission	Main Bearing	M.bear.Level Oil < min
1213	Transmission	Main Bearing	MMBSOilPp RunTime>max

1215	Transmission	Main Bearing	MBSOilPpStartingRate
1224	Transmission	Main Bearing	MBSPress HydOil < SL
1271	Transmission	Gearbox	MGB FuseTripCoolWatPp
1272	Transmission	Main Bearing	M.bear.PressOil IN>SH
1273	Transmission	Main Bearing	M.bear. Error Pressure
1280	Transmission	Main Bearing	M.bear. Press Oil IN < SL
1306	Transmission	Main Bearing	MGB PressOil In < SL
1329	Transmission	Main Bearing	MGB FuseTripOilHeater
1359	Transmission	Main Bearing	M.bear. Temp 1 > SHH
1360	Transmission	Main Bearing	M.bear. Temp 1 > SH
1361	Transmission	Main Bearing	M.bear. Temp 1 < SL
1362	Transmission	Main Bearing	M.bear. Temp 1 < SLL
1363	Transmission	Main Bearing	M.bear. Temp 2 > SHH
1364	Transmission	Main Bearing	M.bear. Temp 2 > SH
1365	Transmission	Main Bearing	M.bear. Temp 2 < SL
1366	Transmission	Main Bearing	M.bear. Temp 2 < SLL
1367	Transmission	Gearbox	MGB TempOilSump > SHH
1368	Transmission	Gearbox	MGB Temp OilSump > SH
1369	Transmission	Gearbox	MGB Temp OilSump < SL
1370	Transmission	Gearbox	MGB Temp OilSump < SLL
1371	Transmission	Gearbox	MGB Repeat Temp Error
1372	Transmission	Gearbox	MGB TempBear151 > SHH
1373	Transmission	Gearbox	MGB TempBear451 > SHH
1374	Transmission	Gearbox	MGB TempBear150 > SHH
1375	Transmission	Gearbox	MGB TempBear450 > SHH
1376	Transmission	Gearbox	MGB TempBear152 > SHH
1377	Transmission	Gearbox	MGB TempBear452 > SHH
1378	Transmission	Gearbox	MGB FilterOil Warning
1379	Transmission	Gearbox	1379: MGB Filter Oil Stop
1380	Transmission	Main Bearing	M.bear.FuseTrip OilPp
1381	Transmission	Main Bearing	M.bear. PressOilPp<SL
1382	Transmission	Main Bearing	M.bear. FilterOilStop
1392	Transmission	Main Bearing	M.bear. FilterOilWarn
1402	Converter	Converter	Freq. conv. warning
1404	Converter	Converter	Freq.conv. emer.stop
1406	Converter	Converter	Freq.conv.grid error
1409	Converter	Converter	Freq. conv. error
1411	Converter	Converter	Freq.conv.overspeed
1412	Converter	Converter	Timeout Freq. conv.
1415	Converter	Converter	Freq.conv. <> sync.
1498	Converter	Converter	Freq.con. heat oper.
1544	Transmission	Gearbox	PT100 defective
1588	Turbine	Control Cabinet	4-20mA signal defect
1595	Turbine	Control Cabinet	Selftest
1667	Converter	Converter	Freq. conv. communi.
1668	Converter	Converter	Freq. conv. error A
1669	Converter	Converter	Freq. conv. error B
1671	Converter	Converter	F.conv. MCCB tripped
1672	Converter	Converter	F.conv. MCCB open
1673	Converter	Converter	Freq.conv.UPS defect
1674	Converter	Converter	F.conv.power high
1684	Converter	Converter	F.conv. trans. grid
1685	Turbine	Control Cabinet	Service active LVU
1689	Converter	Converter	Timeout grid connect
1702	Transmission	Main Bearing	M.bear. TempRepeatErr

1715	Turbine	Control Cabinet	4-20mA signal stop
1743	Turbine	Control Cabinet	Emer. stop 1 nacelle
1744	Turbine	Control Cabinet	Emer. stop 2 nacelle
1768	Turbine	Control Cabinet	Service mode
1769	Turbine	Control Cabinet	Main ctrl.m.shutdown
1791	Turbine	Control Cabinet	[P] reduced temp.
1792	Turbine	Control Cabinet	[P] reduced conv.
1793	Turbine	Control Cabinet	[P] reduced EU
1813	Transmission	Brake	MBS Pads Wear Stop
1814	Transmission	Hydraulic System	(H)(B)press hi. Test
1815	Transmission	Hydraulic System	(H)(B)press lo. Test
1816	Transmission	Hydraulic System	(H)(B)press Test
1817	Transmission	Hydraulic System	(B) Press. Time out
1818	Transmission	Hydraulic System	(B) Press. Delayed
1819	Transmission	Hydraulic System	(B) Press. Deviation
1820	Transmission	Hydraulic System	(B)Press.Limit valve
1821	Transmission	Hydraulic System	(B)Press.Complete
1822	Transmission	Hydraulic System	(B) Press. Undelayed
1823	Transmission	Hydraulic System	(B)Time out test
1907	Rotor	Pitch	LubPitch Rim Time-out
1919	Rotor	Pitch	Pitch 1 too slow
1920	Rotor	Pitch	Pitch 2 too slow
1921	Rotor	Pitch	Pitch 3 too slow
1925	Rotor	Pitch	Pitch 1 pos. FB
1926	Rotor	Pitch	Pitch 2 pos. FB
1927	Rotor	Pitch	Pitch 3 pos. FB
1928	Rotor	Pitch	Capacitor volt. low
1929	Rotor	Pitch	Capacitor volt.high
1930	Rotor	Pitch	Capacitor capac. low
1934	Rotor	Pitch	LubPitchRimGreaseEmpty
1942	Rotor	Rotor	Service active hub
1944	Rotor	Blade	MainSwitchOff blade1
1945	Rotor	Blade	MainSwitchOff blade2
1946	Rotor	Blade	MainSwitchOff blade3
1947	Rotor	Pitch	PLU com. error
1948	Rotor	Pitch	PLU PBU Voltage4<SLL
1949	Rotor	Pitch	PLU safety sys. rel.
1950	Rotor	Pitch	PLU run away
1951	Rotor	Pitch	PLU EndStop100 PS2
1952	Rotor	Pitch	PLU PCO Error
1954	Rotor	Pitch	PLU safety run error
1955	Rotor	Pitch	PLU PCH Error
1956	Rotor	Pitch	PLU PCH Warning
1957	Rotor	Pitch	PLU HMI serv. switch
1958	Rotor	Pitch	PLU1 EndStop -5 PS2
1959	Rotor	Pitch	PLU EndStop100 PS1
1960	Rotor	Pitch	PLU Brake Not Open
1961	Rotor	Pitch	PLU blade fixed
1963	Rotor	Pitch	PLU PBU Voltage4>SHH
1964	Rotor	Pitch	PLU1 PBUDeltaVolt>SHH
1965	Rotor	Pitch	PLU2 PBUDeltaVolt>SHH
1966	Rotor	Pitch	PLU3 PBUDeltaVolt>SHH
1967	Rotor	Pitch	PLU PBU Temp>SHH
1968	Rotor	Pitch	PLU PCOTempHeatSink>SHH
1969	Rotor	Pitch	PLU PDU Temp>SHH

1970	Rotor	Pitch	PLU PCH Temp>SHH
1986	Rotor	Pitch	PLU PCO Temp>SHH
1987	Rotor	Pitch	PLU BladeAngleDelta>SHH
1994	Rotor	Pitch	PLU Pitch Angle>SH
2027	Generator	Generator	Gen PressSwitchCoolWat
2028	Generator	Generator	Gen RepeatCoolWatError
2029	Generator	Generator	LubGen GreaseEmpty
2034	Generator	Generator	LubGen Time-out
2035	Generator	Generator	GenTempBear DE >SH<SL
2037	Generator	Generator	GenTempBearNDE >SH<SL
2040	Generator	Generator	Gen FuseTrip Heater
2046	Generator	Generator	Service act tm. Box
2047	Generator	Generator	Gen WearBrush Stop
2048	Generator	Generator	Gen TempCoil L1 < SL
2049	Generator	Generator	Gen TempCoil L2 < SL
2050	Generator	Generator	Gen TempCoil L3 < SL
2058	Generator	Generator	Gen TempCoil L1 > SHH
2059	Generator	Generator	Gen TempCoil L2 > SHH
2060	Generator	Generator	Gen TempCoil L3 > SHH
2061	Generator	Generator	Gen TempCoil L1 < SLL
2062	Generator	Generator	Gen TempCoil L2 < SLL
2063	Generator	Generator	Gen TempCoil L3 < SLL
2064	Generator	Generator	Gen TempBearNDE > SHH
2065	Generator	Generator	Gen TempBearNDE < SLL
2066	Generator	Generator	Gen TempBear DE > SHH
2067	Generator	Generator	Gen TempBear DE < SLL
2120	Turbine	Control Cabinet	Turbine maintenance
2135	Turbine	Control Cabinet	Check alarm connect.
2142	Transmission	Main Bearing	M.bear.FuseTripOilHeat
2154	Turbine	Power Cabinet	Emer.stop med.volt
2166	Turbine	Power Cabinet	Terminal box temp.
2167	Turbine	Power Cabinet	LV unit temp. high
2168	Turbine	Power Cabinet	LV unit th.rel.
2169	Turbine	Power Cabinet	MV unit MCCB open
2170	Turbine	Power Cabinet	MV unit MCCB releas.
2171	Turbine	Power Cabinet	MV unit PE-br.1 clos
2172	Turbine	Power Cabinet	MV unit PE-br.2 clos
2173	Turbine	Control Cabinet	Safety sys activated
2174	Turbine	Control Cabinet	Safety sys def.
2225	Turbine	Control Cabinet	UPS Comm. Lost
2235	Turbine	Control Cabinet	Obstr.light grid
2236	Turbine	Roof	Obstr.light one lamp
2237	Turbine	Roof	Obstr.light two lamp
2238	Turbine	Control Cabinet	Acc.nacelle Y warn.
2239	Turbine	Control Cabinet	Acc.nacelle Z warn.
2240	Turbine	Control Cabinet	Acc.nacelle Y stop
2241	Turbine	Control Cabinet	Acc.nacelle Z stop
2242	Tower	Tower	Tower torsion warn.
2243	Tower	Tower	Tower torsion stop
2244	Turbine	Control Cabinet	Acc.nacel.filt.Y war
2245	Turbine	Control Cabinet	Acc.nacel.filt.Z war
2246	Turbine	Control Cabinet	Acc.nacel.filt.Y stp
2247	Turbine	Control Cabinet	Acc.nacel.filt.Z stp
2248	Turbine	Control Cabinet	ACS watchdog
2249	Turbine	Control Cabinet	ACS error

2250	Turbine	Control Cabinet	Resid.curr.guard war
2251	Turbine	Control Cabinet	Resid.curr.guard stp
2252	Turbine	Control Cabinet	CMS watchdog
2253	Turbine	Control Cabinet	CMS warning
2254	Turbine	Control Cabinet	CMS stop
2269	Turbine	Control Cabinet	WP4086#1 acc. Alarm
2270	Turbine	Control Cabinet	WP4086#1 acc. warn.
2273	Turbine	Control Cabinet	WP4086#1 lev.1 alarm
2274	Turbine	Control Cabinet	WP4086#1 lev.1 warn.
2275	Turbine	Control Cabinet	WP4086#1 lev.2 alarm
2276	Turbine	Control Cabinet	WP4086#1 lev.2 warn.
2277	Turbine	Control Cabinet	WP4086#1 lev.3 alarm
2278	Turbine	Control Cabinet	WP4086#1 lev.3 warn.
2279	Turbine	Control Cabinet	WP4086 comm. Error
2285	Turbine	Control Cabinet	App. system err.
2290	Turbine	Control Cabinet	WP4086#1 log ready.
2291	Turbine	Control Cabinet	WP4086#1 Intern Err.
2300	Transmission	Gearbox	MGB TempBear151 > SH
2301	Transmission	Gearbox	MGB TempBear451 > SH
2302	Transmission	Gearbox	MGB TempBear150 > SH
2303	Transmission	Gearbox	MGB TempBear450 > SH
2304	Transmission	Gearbox	MGB TempBear152 > SH
2305	Transmission	Gearbox	MGB TempBear452 > SH
2306	Transmission	Gearbox	MGB Temp ErrTimeLimit
2423	Turbine	Control Cabinet	Fire protect warn. 1
2424	Turbine	Control Cabinet	Fire protect warn. 2
2425	Turbine	Control Cabinet	Fire protect warn. 3
2426	Turbine	Control Cabinet	Fire protect warn. 4
2427	Turbine	Control Cabinet	Fire protect warn. 5
2476	Turbine	Control Cabinet	App. Not configured
2491	Turbine	Control Cabinet	Repeating alarm
2701	Turbine	Control Cabinet	No PMS available
2703	Turbine	Control Cabinet	PMC stop
3025	Turbine	Control Cabinet	Stop by SCADA
3108	Turbine	Control Cabinet	TSO Stop
3109	Turbine	Control Cabinet	APC Local Mode
3110	Turbine	Control Cabinet	APC Setpoint !
3111	Turbine	Control Cabinet	RPC Local Mode
3112	Turbine	Control Cabinet	RPC Setpoint
3113	Turbine	Control Cabinet	P > P Sp
3119	Turbine	Control Cabinet	GCA Grid
3120	Turbine	Control Cabinet	GCA Grid Con. Inhibit
3121	Turbine	Control Cabinet	TSO Reconnect. Block
3122	Turbine	Control Cabinet	GCA Manual Stop
3123	Turbine	Control Cabinet	GCA Stop
5000	Turbine	Control Cabinet	WTG System OK
5131	Turbine	Control Cabinet	Wrong App 50Hz 60 Hz
5132	Turbine	Control Cabinet	Communication TEST
5135	Turbine	Control Cabinet	App. Not configured
5194	Turbine	Control Cabinet	Customer ON
5195	Turbine	Control Cabinet	Customer OFF
5196	Turbine	Control Cabinet	Service ON
5197	Turbine	Control Cabinet	Service OFF
5198	Turbine	Control Cabinet	Maintenance ON
5199	Turbine	Control Cabinet	Maintenance OFF

5213	Turbine	Control Cabinet	PLU PCO warning
5235	Rotor	Rotor	Rotor Speed > SH
5256	Rotor	Pitch	PBU Error
5257	Rotor	Pitch	PBU Warning
5345	Turbine	Control Cabinet	NacTowerUPSLightWarn
5361	Rotor	Blade	BladeCtrl Error
5362	Rotor	Blade	BladeCtrl Warning
5363	Rotor	Blade	BladeDamage
5430	Turbine	Control Cabinet	GridFQ<StartCondition
5473	Turbine	Control Cabinet	LVU LightingProtWarn1
5474	Turbine	Control Cabinet	LVU LightingProtWarn2
5494	Turbine	Control Cabinet	LVU ExternalStop
5495	Turbine	Control Cabinet	External warning
5496	Turbine	Control Cabinet	TB External stop 1
5497	Turbine	Control Cabinet	TB ExternalStop 2
5498	Turbine	Control Cabinet	TB ExternalStop 3
5499	Turbine	Control Cabinet	TB ExternalStop 4
5705	Generator	Generator	Gen Speed > SHH
5706	Generator	Generator	Gen <> RotSafSys > SH
5707	Yaw	Yaw	Gen <> RotIGR > SH
5815	Yaw	Yaw	Yaw Inv Error Repeat
5930	Yaw	Yaw	LubYaw GreaseEmpty
5931	Yaw	Yaw	LubYaw Pressure Error
5932	Yaw	Yaw	LubYaw Error

Table 3. Complete list of alarms provided by the manufacturer. Each alarm is identified by a numeric ID. Note that the ID numbers are all integers, ordered from lowest to highest, but not consecutive.

4 Options of the function *get_turbine_data*

This function is the one used to obtain the wind turbine data for specific WTs, period of time, and type of pre-processing, together with the selected alarms.

The function is called as follows:

```
[error,data,msg]=get_turbine_data(compressed_file_name, alarm_id\_list,
    frequency_seconds, combine_func, one_hot_encoding, threads, verbose)
```

Where the parameter frequency_seconds allows merging the data in blocks according to the selected frequency (in seconds), to decrease the sampling rate if needed.

The pre-processing (filtering of outliers) is controlled with the combine_functions, which has the following possible values:

- "mean": Average of aggregated data inside the block
- "median": Median of aggregated data inside the block
- "max": Max of aggregated data inside the block"
- "min": Min of aggregated data inside the block"
- "filtered_3sdv_mean": Within each aggregated block of data, a filtering process removes all values outside the range mean$\pm$3sdv. The mean of the remaining values is then calculated.
- "filtered_3sdv_median": Within each aggregated block of data, a filtering process removes all values outside the range mean$\pm$3sdv. The median of the remaining values is then calculated.
- "filtered_mad_mean": Within each aggregated block of data, a filtering process removes all values outside the range median$\pm$2MAD. The mean of the remaining values is then calculated.
- "filtered_mad_median": Within each aggregated block of data, a filtering process removes all values outside the range median$\pm$2MAD. The median of the remaining values is then calculated.

Note that in the first four options the outliers are not filtered, while in the last four, the values are filtered using two different criteria, and the mean or the median values are calculated after this pre-processing.

Finally, one_hot_encoding is used to determine how the alarm information is coded. If TRUE, a column will be generated for each alarm starting with "alarm_" and the alarm id with 0 (not active) and 1 (active). If FALSE, a single column will be generated with the name alarms_active including a comma-separated list of active alarms.