Specification of 'normal' wind turbine operating behaviour for rapid anomaly detection: through the use of machine learning algorithms

Dissertation for the Degree of MSc in Renewable Energy Engineering

Nithiya M Streethran[1]*
Supervisors: Dr Nick Bennett[2] and Dr Iain Dinwoodie[3]

[1, 2] School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom
[3] Natural Power Consultants, Ltd., Stirling FK7 7XE, United Kingdom

25th August 2017

1. Background

There is a need to increase the economic effectiveness of wind turbines, which refers to the cost to run them relative to the electricity generation, or revenue. [1, 2] Increasing this effectiveness lowers the payback period of new wind turbines or farms, thus making wind a more economic clean energy source, attracting governments and private organisations to make more investments in wind projects. [1] It can, however, be decreased due to major component failure, frequent downtime, turbine degradation and age, which in turn increase the operation and maintenance cost and decrease the energy generation efficiency of wind turbines. [1, 3] There are difficulties and high costs involved in carrying out maintenance on wind turbines, especially for ones that operate in extreme and remote conditions, such as offshore wind farms, where the turbines tend to also exist in larger numbers. [3, 4]

Condition-based monitoring systems that continuously monitor wind turbine states increase this effectiveness by significantly reducing the maintenance costs, reportedly by 20% to 25%, as it prevents unscheduled maintenance. [2] According to the Electric Power Research Institute, reactive maintenance, which refers to running the turbine until it reaches failure, has the highest cost, followed by preventive or scheduled maintenance, which is reported to cost 24% less. [5] Meanwhile, condition-based or predictive maintenance, which prevents catastrophic failure, [1] is reported to save 47% of the cost of reactive maintenance, [5] which makes it the most cost-effective and preferred approach. Condition-based monitoring technologies include sensor-based oil and vibration analysis, which are useful for checking the oil for properties such as temperature, and rotating equipment respectively. [6] These technologies, however, tend to put emphasis on the more expensive parts of a wind turbine such as the gearbox [7] due to the high costs involved in the installation of these sensors. [2, 6] These systems, which can be purchased from the turbine manufacturer, are usually pre-installed in offshore wind turbines due to the harsh environments in which they operate. However, they can be expensive [4] and uneconomical, especially for older wind turbines in onshore wind farms, whose outputs are often less than that of an offshore wind farm.

An alternative would be to use SCADA-based analysis, where the only cost involved would be computational and expensive sensors are not required. [4, 2] A SCADA system, which stands for supervisory control and data acquisition, found pre-installed in most utility-scale wind turbines, collects data using numerous sensors at the controllers with usually 10-minute resolution, [4, 8] of various parameters of the wind turbine, such as wind speed, active power, bearing temperature and voltage. [2] Power curve analysis can be done using this data, but this analysis only detects wind turbine underperformance. [9] Meanwhile, implementing machine learning algorithms on SCADA signals to classify them as having either normal or anomalous behaviour, has the ability to predict faults in advance.

* Email address: nms31@hw.ac.uk; Matriculation number: H00158233

2. Objectives

The first objective of this project is to implement a classification algorithm on wind turbine SCADA signals to identify underperforming turbines. This involves setting-up the machine learning environment, processing operational data and reporting initial results obtained through implementing a classification algorithm on the data.

The second objective is to create an effective methodology for the integration of failures and to present and interpret results. This includes labelling the data such that each specific fault can be differentiated, evaluating the performances of several classification algorithms to find the most suitable classifier, identifying limitations and suggesting improvements to the method and how it can be adapted for use in industry.

3. Tools and datasets

This project requires a computer with Python Programming Language [10] and essential libraries installed for data processing. The computer used has a dual core processor with 2.8 GHz maximum clock speed and 4 GB RAM. Additionally, the open source scikit-learn library [11] is used for machine learning. The datasets used are that of a wind farm comprised of 25 turbines with a rated power of 2,500 kW covering a period of 30 months starting 1st November 2014, downloaded from Natural Power's database in CSV format. The first dataset is wind turbine SCADA signals timestamped with a resolution of 10 minutes, with a total file size of 452 MB, and the other dataset is corresponding downtime data for the same period, with a total file size of 4 MB. In the interests of Natural Power, the location of the wind farm and turbine model will not be disclosed in this report.

4. Data processing

The SCADA data has 17 fields, summarised in Table 4.1. Fields highlighted in green are average measurements recorded over each 10-minute period. Since these highlighted fields are properties of the turbines or describe its performance, they can be used as features in machine learning. Each turbine has two nacelle anemometers and wind vanes; one is used to control the turbine, and the other to monitor the first. The measurements from the anemometer and wind vane used to control the turbine are recorded again as 'ws_av' and 'wd_av', with the latter taking into account the nacelle position. Using only 'ws_av' and 'wd_av' for wind speed and wind direction, the number of features that are available for machine learning is 10.

The downtime data consists of fields summarised in Table 4.2. Each row of downtime data consists of the start and end timestamps of the downtime event, downtime categories, workorders and alarms. Downtime categories, which are turbine, environmental, grid, infrastructure and availability categories, describes the turbine's condition or cause of downtime when the maintenance work was undertaken. Each condition within each downtime category is represented by a unique identifier in the dataset. A separate spreadsheet accompanying the dataset list what each identifier stands for. All quantities in the downtime data, except the alarms, are supervised (i.e., the data recordings are input and monitored by maintenance professionals).

Each row of SCADA data requires a class which describes the state of the turbine. The chosen classes are 'normal' for normal behaviour, and 'faulty' to signify a fault. As the aim is to predict faults in advance, a category of classes, called 'before fault' will also be used. To automate the labelling process, the SCADA data can be merged with the downtime data, which has turbine categories, listed in Table 4.3, that can be used to label faults. Some of these turbine categories, such as 'OK' and 'scheduled maintenance', do not indicate a fault in the turbine, and 'other' does not specify the condition. Therefore, only the turbine categories which indicate faults, highlighted in green, are used to class the SCADA data. Prior to merging the two datasets, the downtime data is restructured such that it has the same 10-minute resolution as the SCADA data. The SCADA data was also found to have missing rows of data. Empty data

rows with only the timestamp corresponding to the missing rows were added to rectify this. Once they are merged, 14 separate labels, or columns, are added for each specific fault, which will allow for the different faults to be distinguished. The rows with a fault category are classed as 'faulty' in the corresponding column.

Table 4.1: Summary of SCADA fields for the SCADA data used in this project. The fields include timestamps with a resolution of 10 minutes, average active power, wind speed, pitch and runtime. The fields that contain measurements averaged over the 10-minute period are highlighted in green. These measurements can be used as features in machine learning as they are turbine properties.

SCADA field	Description	Unit
timestamp	In the format dd/mm/YYYY HH:MM:SS, every 10 minutes	
turbine_id	Turbine identifier (1 to 25)	
ap_av	Average active power	kW
ap_dev	Active power deviation	kW
ap_max	Maximum active power	kW
reactive_power	Reactive power	kVAr
ws_1	Wind speed measured by nacelle anemometer 1	m/s
ws_2	Wind speed measured by nacelle anemometer 2	m/s
ws_av	Anemometer wind speed (either ws_1 or ws_2)	m/s
wd_1	Wind direction measured by wind vane 1	°
wd_2	Wind direction measured by wind vane 2	°
gen_sp	Generator speed	rpm
rs_av	Rotor shaft speed	rpm
nac_pos	Nacelle position	°
wd_av	Corrected wind direction (nac_pos + (either wd_1 or wd_2))	°
pitch	Pitch angle	°
runtime	Number of seconds the turbine has operated in the 10-minute period	s

Table 4.2: Summary of fields for the downtime data used in this project. The fields include start and end timestamps for the downtime event, downtime categories, workorders and alarms.

Downtime field	Description
timestamp_start	Start time of event, in the format dd/mm/YYYY HH:MM:SS
timestamp_end	End time of event, in the format dd/mm/YYYY HH:MM:SS
turbine_id	Turbine identifier (1 to 25)
alarm_id	Ranging from 1 to 480, each corresponding to a turbine status
GridCategory_id	Identifier (0 to 3); describes the grid status (e.g., planned outage, unplanned outage, …)
InfrastructureCategory_id	Identifier (0 to 3); describes the infrastructure status (e.g., planned outage, unplanned outage, …)
EnvironmentalCategory_id	Identifier (0 to 14); describes the condition of the operating environment (e.g., icing, turbulence, …)
TurbineCategory_id	Identifier (0 to 22); describes the turbine's condition or problem (e.g., yaw system, electrical controls, …)
AvailabilityCategory_id	Identifier (0 to 2); describes the availability status (e.g., available, not available)
comment	Elaborates the condition or maintenance work undertaken
workorder_id	Recorded when maintenance work is undertaken

Figure 4.1 shows the various turbine categories quantified by downtime frequency on the left and period on the right, both per turbine per year. Looking at only turbine categories used as labels, 'pitch control' and 'electrical system' are the two categories causing the most downtime events and are in the top three in terms of the downtime period.

Table 4.3: List of turbine categories in the wind farm downtime data. The categories used as the different faults for labelling are highlighted in green. The others do not indicate a fault.

Turbine category							
id	**Name**	**id**	**Name**	**id**	**Name**	**id**	**Name**
0	Unknown	6	Generator	12	Unlogged manual stop	18	Cable unwind
1	OK	7	Yaw system	13	Customer stop	19	Hub
2	Anemometry	8	Electrical controls	14	Noise constraints	20	Rotor blades
3	Rotor brake	9	Hydraulics	15	Scheduled maintenance	21	Delayed startup
4	Main shaft	10	Electrical system	16	Tower	22	Other
5	Gearbox	11	Pitch control	17	Retrofit		

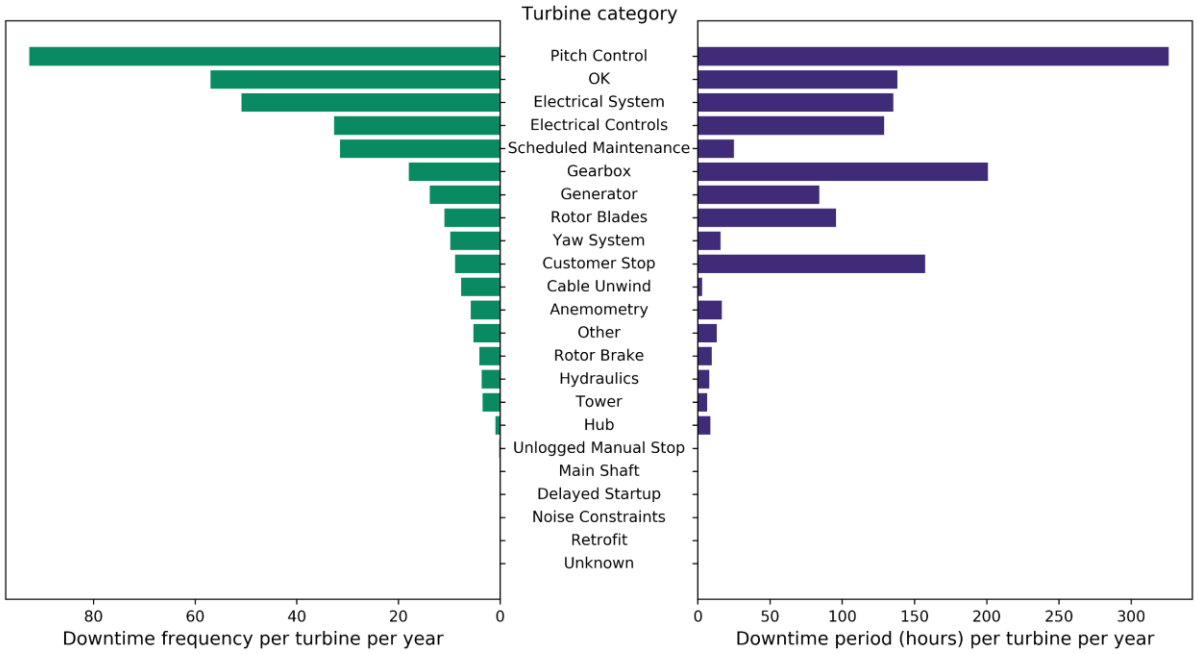


Figure 4.1: Bar chart showing the various turbine categories quantified by the downtime frequency per turbine per year on the left, and downtime period, in hours, per turbine per year on the right. This was plot using the downtime data.

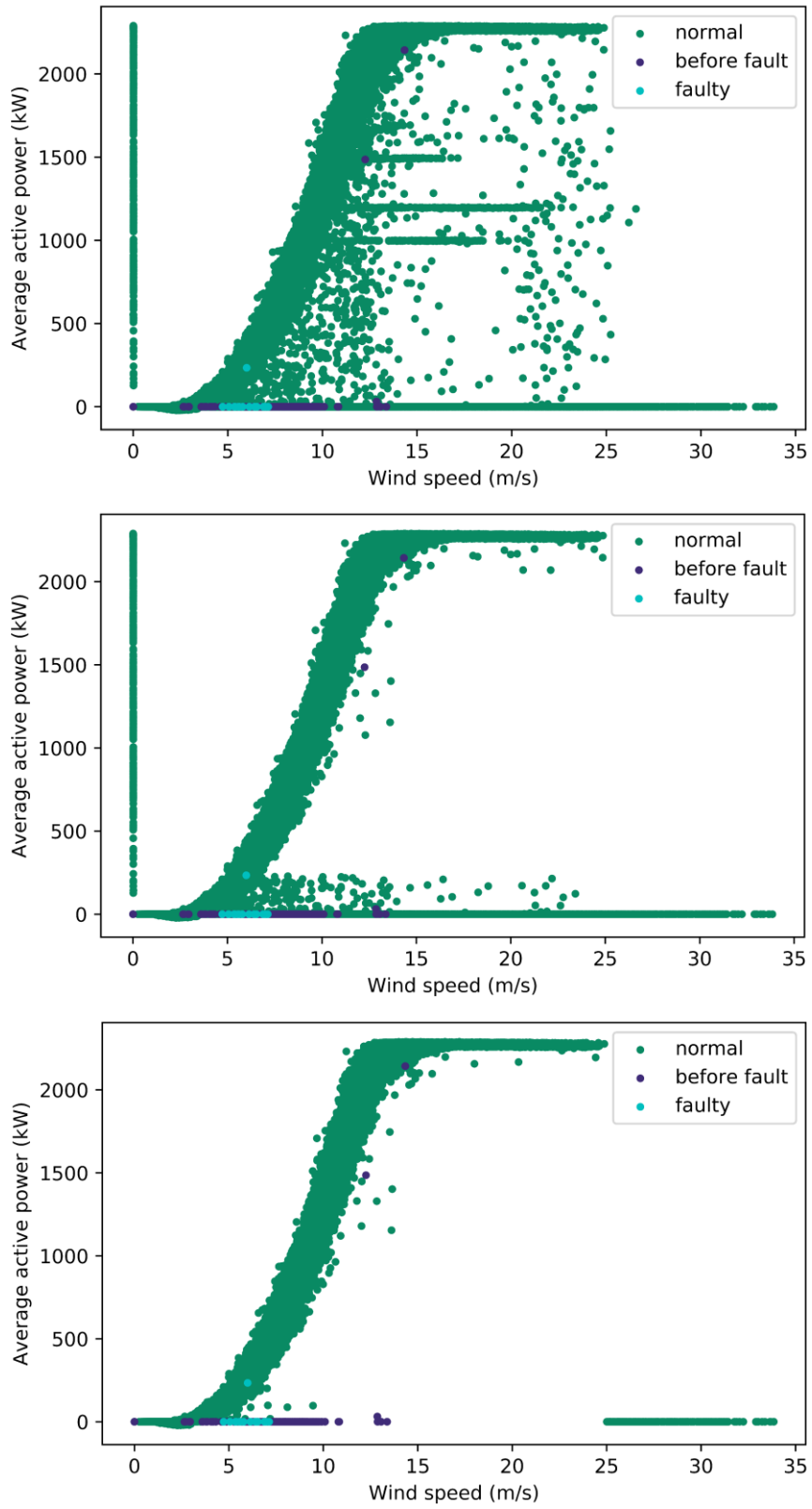
To summarise the machine learning terminology used, features refer to SCADA fields which are turbine properties, labels refer to turbine categories or type of fault, and classes refer to the state of the turbine (e.g., 'normal' or 'faulty') for each row of data at each label. The features and labels will be fit to a classifier for training as arrays X of size [rows, 10] and Y of size [rows, 14] respectively, where rows refer to the number of rows in the training data.

To predict faults for each label, rows with timestamps up to 48 hours in advance of a 'faulty' row are classed at 6-hour intervals (i.e., up to X hours before a fault, where X = 6, 12, …, 48). The reasons for having classes of 6-hour intervals for fault detection rather than a single class is to allow action to be taken appropriate to the time before fault. For example, if it is predicted that the wind turbine could have a fault in six hours or less, it could be switched off to prevent further damage from occurring. 48 hours is enough time for maintenance professionals to travel to site and carry out inspection, and decide on what action to take. Depending on the nature of the site, this value can be modified (i.e., for an offshore wind farm which operates in harsh environments, it is more likely to take a longer time to travel to the site and complete works relative to an onshore wind farm).

Power curves are used to help with labelling as they are easier to visualise due to the distinct power curve shape which represents wind turbine performance. Figure 4.2(a) shows the labelled power curve for turbine 2 with turbine category 16, where many curtailment and

anomalous points are classed as ‘normal’. These should be removed as they deviate from the typical power curve shape which indicates normal behaviour. To filter out the curtailment, the pitch angle should be within a typical threshold for ‘normal’ data points between 10% and 90% power. Data points with power below 10% and above 90% are not included, pitch angles often deviate from 0° in these operating regions, due to the control of the turbine. To find this threshold, the most frequent pitch angles are quantified, with 0° being the most frequent. Filtering out points with a pitch angle exceeding 0°, however, distorts the power curve shape, removing a large portion of points in the region where it transitions to rated power. To prevent this, pitch angles between 0° and 10° were tested as the threshold, with 3.5° producing the best result (see Appendix A1 for full results). The effects of applying this filter can be seen in Figure 4.2(b), which still has anomalous points below 10% and above 90% power. To remove these, additional filters are applied to ‘normal’ data points at operating wind speeds, including removing zero power, and turbine categories and other downtime categories that are not faults or ‘OK’, and runtime of less than 600 s. There is a vertical line of data points at zero wind speed which is removed using a power threshold of 100 kW before the cut-in speed of 3 m/s. It is necessary to use this threshold because the nacelle anemometer wind speed, which is used to plot these power curves, is not an accurate measure of the wind speed incident on the turbine blades and removing all data points exceeding 0 kW power before cut-in results in a distorted power curve shape (see Appendix A2). The threshold is based on the minimum power before cut-in that does not distort the power curve shape for all 25 turbines. The result of applying these filters is shown in Figure 4.2(c).

Rows of data with missing features and labels are removed, as all fields must be complete for classification. Instead of deleting the rows of data corresponding to the data points removed from the ‘normal’ class, they are classed as ‘curtailment’. This is because the data points removed are specific to one label, which means they are not necessarily classed as ‘normal’ for other labels, and it is important for the classifier to learn the different states of the turbine for each fault. To summarise, the classes used are ‘normal’, ‘faulty’, ‘curtailment’ and ‘up to X hours before fault’ (where X = 6, 12, …, 48).



(a) All data points

(b) Without curtailment (i.e., pitch angle is between 0° and 3.5° for 'normal' data points between 10% power and 90% power)

(c) Additional filters (applied to 'normal' data points):

- *Power > 100 kW before cut-in (3 m/s)*
- *At operating wind speeds (3 m/s to 25 m/s):*
  - *Power ≤ 0 kW*
  - *Runtime < 600 s*
  - *Availability categories ≠ available/ non-penalising*
  - *Environmental, grid or infrastructure categories ≠ OK*
  - *Turbine categories not highlighted in Table 4.3, or ≠ OK*

Figure 4.2: Changes to the power curve of turbine 2 with the fault points corresponding to when the turbine category is 16 ('tower') through the two stages of filtering out anomalous and curtailment points labelled as 'normal'. The original power curve is shown in (a). The first stage involves a filter based on a pitch angle threshold, which produces (b). The second stage involves several additional filters to produce the final power curve (c).

References

[1] K. Kim, G. Parthasarathy, O. Uluyol, W. Foslien, S. Sheng and P. Fleming, “Use of SCADA Data for Failure Detection in Wind Turbines,” in *ASME 2011 5th International Conference on Energy Sustainability*, Washington, D.C., 2011.

[2] K. Leahy, R. L. Hu, I. C. Konstantakopoulos, C. J. Spanos and A. M. Agogino, “Diagnosing wind turbine faults using machine learning techniques applied to operational data,” in *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, Ottawa, 2016.

[3] S. Dienst and J. Beseler, “Automatic Anomaly Detection in Offshore Wind SCADA Data,” Big Data Competence Center, University of Leipzig, Leipzig.

[4] J. Tautz-Weinert and S. J. Watson, “Using SCADA data for wind turbine condition monitoring – a review,” *IET Renewable Power Generation,* pp. 1-13, 2016.

[5] National Instruments, “Wind Turbine Condition Monitoring,” National Instruments, 18 December 2015. [Online]. Available: http://www.ni.com/white-paper/9231/en/. [Accessed 19 August 2017].

[6] F. P. García Márquez, A. M. Tobias, J. M. Pinar Pérez and M. Papaelias, “Condition monitoring of wind turbines: Techniques and methods,” *Renewable Energy,* vol. 46, pp. 169-178, 2012.

[7] J. L. Godwin and P. Matthews, “Classification and Detection of Wind Turbine Pitch Faults Through SCADA Data Analysis,” *International Journal of Prognostics and Health Management,* vol. 4, pp. 1-11, 2013.

[8] W. Yang, P. J. Tavner, C. J. Crabtree, Y. Feng and Y. Qiu, “Wind Turbine Condition Monitoring: Technical & Commercial Challenges,” *Wind Energy,* vol. 17, no. 5, pp. 673-693, 2014.

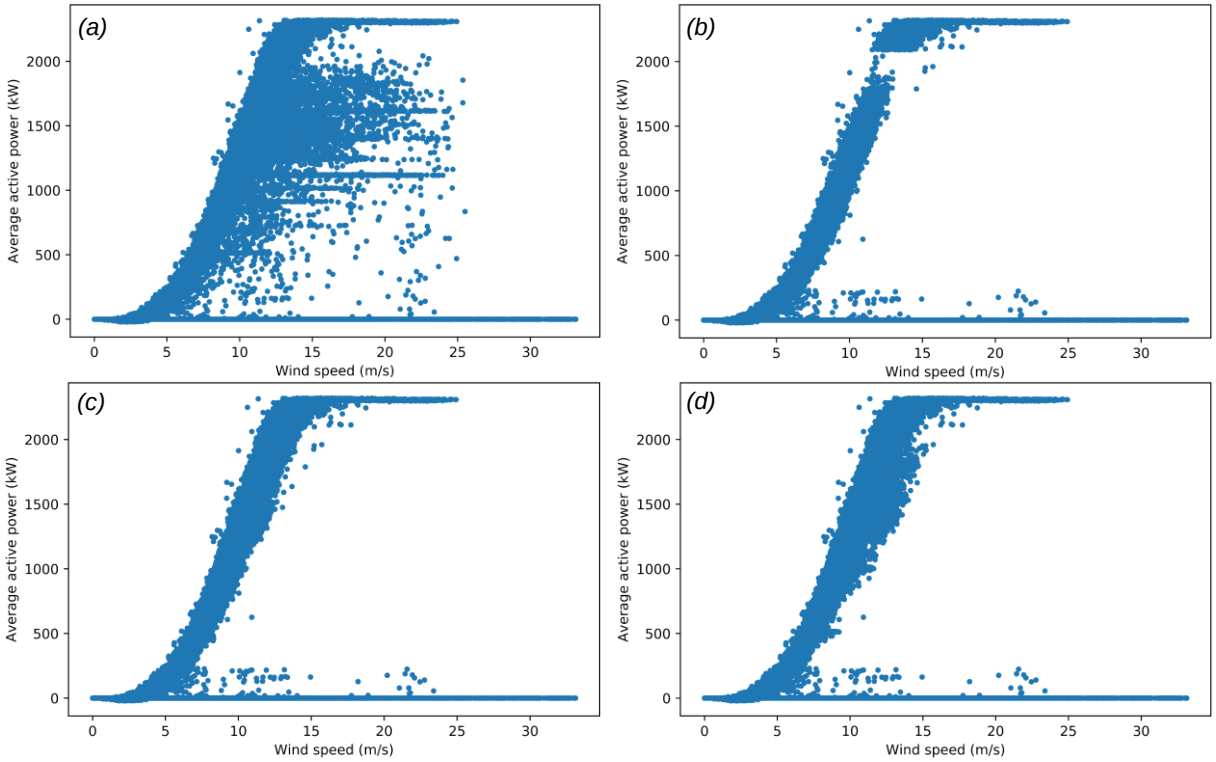
[9] S. Gill, B. Stephen and S. Galloway, “Wind Turbine Condition Assessment Through Power Curve Copula Modeling,” *IEEE Transactions on Sustainable Energy,* vol. 3, no. 1, pp. 94-101, 2012.

[10] Python Software Foundation, “Welcome to Python.org,” Python, [Online]. Available: https://www.python.org/. [Accessed 24 August 2017].

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher and M. Perrot, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

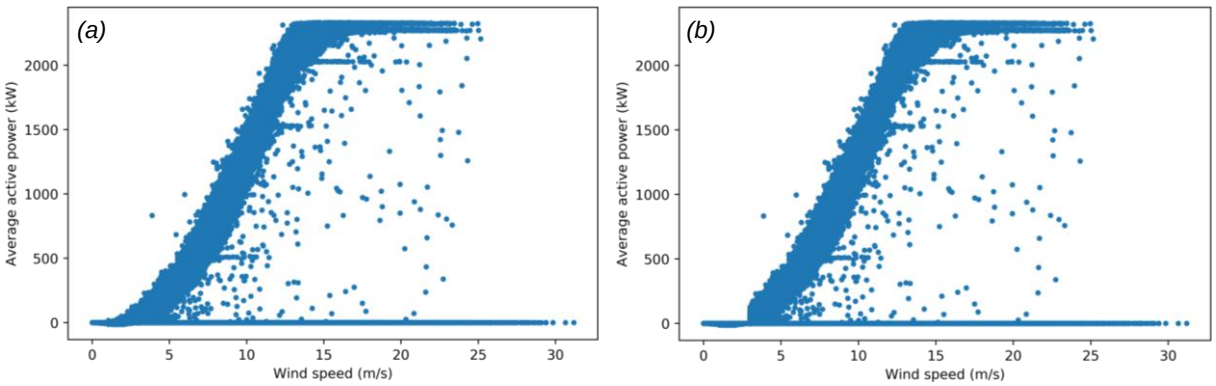
Appendix

A1. Pitch angle threshold



Power curves for turbine 1 used in selecting the pitch angle threshold. (a) is the original power curve. In (b), data points with a pitch angle not equal to 0° between 90% and 10% power were filtered out, which distorts the power curve shape. In (c), all data points have a pitch angle between 0° and 3.5°, which removes most curtailment and anomalous points while maintaining the typical power curve shape. In (d), all data points have a pitch angle between 0° and 7°, which allows some curtailment points to appear. Therefore, it was decided that the filter used in (c) is the most suitable.

A2. Power before cut-in threshold



Power curves for turbine 24 used in selecting the power threshold before cut-in speed. (a) is the original power curve. (b) is the power curve with a filter applied to remove all data points with power > 0 kW before the cut-in speed of 3 m/s. Anemometer wind speeds, which were used to plot these power curves, are not an accurate measure of the wind speed incident on the turbine blades. Therefore, a threshold of 100 kW before cut-in is applied, which maintains the power curve shape for all 25 turbines while removing anomalous points, such as the ones in turbine 2's power curve.