Fleet-oriented pattern mining combined with time series signature extraction for understanding of wind farm response to storm conditions

P-J. Daems[1], L. Feremans[2], T. Verstraeten[14], B. Cule[3], B. Goethals[2], and J. Helsen[1†]

[1†]*Acoustics and Vibrations Research Group, Vrije Universiteit Brussel, Pleinlaan 2, Brussel, Belgium.*
[†]*E-mail: jan.helsen@vub.be*

[2] *Department of Mathematics & Computer Science, Universiteit Antwerpen*

[3] *Department of Accountancy and Finance, Universiteit Antwerpen*

[4] *Artificial Intelligence Laboratory, Vrije Universiteit Brussel*

Keywords: Pattern mining, condition monitoring, wind turbine, dynamic event, fleet, wind farm

Abstract

Offshore wind turbine installations are rapidly spreading around Europe and all over the world. These turbines are typically installed in large wind farms combining turbines of the same type. Farm owners target maximal performance of the farm in general and particularly predictability of behaviour. The latter is getting increasingly important since offshore wind farms are being managed more and more as conventional power plants driven by the electricity market supply and demand considerations. The context of zero subsidy farms exposes farm operators to fluctuations in electricity market prices. As such, deep understanding of farm behaviour is essential to come up with a good strategy to deal with these fluctuations.

This paper focusses on the automated extraction of farm-wide response to storm conditions. The input data for the analysis are status logs and SCADA 1-second data. The status logs record the important turbine controller events. Typically they consist of a number, a time of occurrence, and a time of deactivation. The number is linked to a detailed description. The SCADA data consists of time series of the most important sensors in the turbine: power produced, RPM, wind speed,… The advantage of the 1-sec data over the traditional 10-minute averages is that the dynamic event content is much more preserved. Data of several offshore wind farms is used in the analysis to have a solid dataset. In total, 5 years of data of more than 50 turbines is used.

We show a novel farm-wide pattern mining approach that extracts events occurring for multiple turbines in the same time period. This allows us to identify those events that are predominantly driven by global wind excitations (e.g., gusts) or grid events (e.g., low voltage ride through). From the extracted events we lift out the storm conditions. For these conditions a further investigation of the time series data is done. Using event detection algorithms we extract the signatures of the stop events that each turbine is performing from the time series data. We show that the extreme change in wind speed and wind direction leads to an excessive misalignment of the turbines in the farm, followed by a stop of those turbines. The extracted patterns are compared to the time signatures to show their correlation and complementarity. As such, the typical turbine response to this event is identified. This can serve as input for identification of novel controller approaches by the farm owner and turbine manufacturer to deal with this problem.

1 Introduction

Wind turbines target a typical lifetime of 20 years. During those years the machine should produce as much energy as possible, while keeping downtime due to failure minimal. As such predictability of future behaviour is key. On the one hand, condition monitoring approaches are investigated extensively to achieve better failure predictions and thus increased uptime [1,2,3]. A variety of sensor types can be used for turbine condition monitoring ranging from accelerometers, current sensors to oil particle count devices. An increasing attention has been given in literature to the use of data of the supervisory control and data acquisition (SCADA) system as input data for performing condition monitoring, since this would avoid the need to place additional sensors on the turbine [4,5,6]. On the other hand performance analysis, e.g. through power curve modelling, target an increase of turbine output [7,8,9]. Today, performance analysis is typically performed on data of the SCADA system of the turbine sampled at 10 minute intervals. These values are highly interesting to keep an overview of general trends. However, the 10-minute interval is challenging if one intends to investigate the dynamic events the turbines are exposed to, such as emergency stops. Another data source also provided by the SCADA system offers potential in this regard: the status logs [10]. These logs provide a record of the most important turbine controller actions. Typically, they are characterized by a status code number, a descriptive message, a time of activation, a time of de-activation and a duration. Since the trigger times are registered directly this data source is very limitedly influenced by time-shifts linked to limited sampling rates. As such the status logs show high potential for event sequence investigations to better understand turbine response to triggers for the controller. Gonzalez et al. define an optimized taxonomy for categorizing the status logs and investigate the relation between component faults being followed by failure occurrences in others [11]. Pattern mining techniques are currently being explored to get detailed insights in these sequences. Kusiak et al. used frequent pattern mining techniques to identify and predict sequences in the status logs [10]. Giu et al. define two methods for alarm analysis: a time-based and probability-based approach and illustrate this approach on 2 years of field data from two populations of onshore turbines [12]. Feremans et al. similarly determined frequent item sets and used those to derive association rules to determine patterns near alarms [13]. These investigations show that there is potential for the use of pattern mining approaches for status code analysis to determine and better understand the sequences governing the turbine controller responses to different triggers. This paper builds further on this and introduces the fleet concept. Instead of focussing on individual turbines, we target the detection of patterns over multiple turbines in the farm. In other words similar responses of multiple turbines in the farm to joint or individual triggers.

In addition to the status logs and SCADA 10 minute data, for newer turbine types the complete SCADA tag set or at least a subset is becoming available at sampling rates of 1 second. The finer granularity of this data sources makes it a complementary data source for event understanding to the status log data. Event detection algorithms allow to extract and classify the dynamic events based on their time-series signatures [14]. This paper uses this approach to extract events from the SCADA 1 second data of the different turbines in the farm. This data is then combined with the interesting patterns mined from the status logs.

2 Methodology

2.1 Overview

This paper suggests a method building on the identification of fleet-wide patterns in the status logs and in the dynamic events detected in the SCADA 1-second data, illustrated in Figure 1. It comprises of three parts. First, event detection is done on the SCADA 1-second data to extract turbine start and stop events. Second, patterns are extracted separately from status logs to detect event sequences that are present in multiple turbines in the farm. Finally the results of the two are brought together in an event fusion step. The following subsections will discuss the different components of the methodology in detail. In the next section the methodology is illustrated by means of an experimental example.

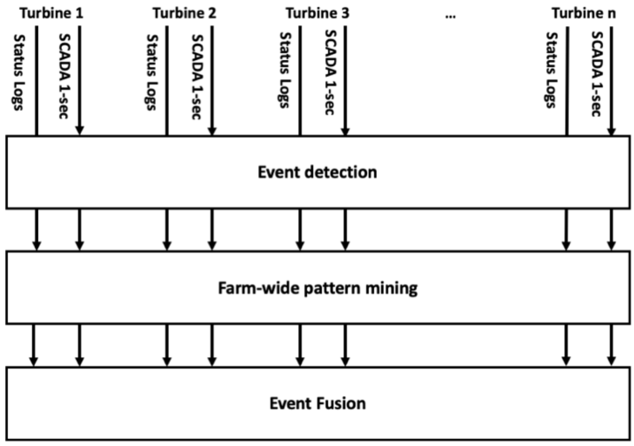

Figure 1: Farm-wide approach for event sequence extraction

Step 1: Pattern mining of status log data

Prior to the discussion of the pattern mining approach targeting itemsets used in the proposed approach some terminology is introduced. To make it more intuitive the analogy is made to analysis of customer baskets in a supermarket context.

- *Itemset:* All status log windows in the database are considered the set of all items: $I = \{i_1,i_2,\ldots,i_n\}$. A subset of I is defined as an itemset. In comparison, in the field of supermarket market basket analysis the set of items consists of all the products that the supermarket is selling.
- *Transaction:* A transaction is defined as a unique tuple that is comprised of a transaction identifier and an itemset. Thus a transaction is noted by the tuple $T_i = (tid,X)$. In this tuple tid is a unique transaction identifier. X is an itemset and thus a subset of I. Again, for the supermarket basket analysis a transaction is a set of items or products that one customer has in his basket.
- *Transaction database:* The transaction database contains all transactions. For the supermarket basket analysis the transaction database contains all baskets that were sold to customers during the period of investigation.
- *Support:* Support is linked to an itemset. It represents the number of transactions in the transaction database in which a certain itemset occurs. Typically a minimum support is defined: min_sup.
- *Frequent pattern:* Only frequent itemsets with a support higher than this minimum are considered. Similarly, a maximum support can be defined. The pattern mining

approaches used in this paper pose a threshold to the minimum support neglecting all patterns below the minimum support.

- *Closed itemset:* One of the main challenges for pattern mining approaches is the fast growth of frequent patterns. Many of the identified itemsets have subsets. An itemset of size n has 2^n subsets. If the support of the itemset is high and above the min_sup threshold, then also that of the subsets will be high, which means that it will be above the min_sup threshold too. Therefore, we target closed itemsets. An itemset can be defined to be closed if no superset is present with the same support.

2.2 Frequent pattern mining

Frequent pattern mining algorithms target the discovery of all itemsets in the transaction database with a support higher than the defined minimum support. This cannot be done brute force since it would result in enumerating $2^{|I|}$ subsets to check all possible subsets and count occurrences. To overcome this, most frequent itemset mining algorithms exploit the anti-monotonic property of itemset support in a branch-and-bound algorithm. For any superset of an itemset, it holds that if the support of the itemset is smaller than the minimum support, this will also be the case for the superset. This property can be intuitively explained as follows. Assume we have three items x, y and z and assume x to occur in 8 transactions, y in 4 transactions and z in 2 transactions. In case we mine for patterns with minimum support of 3, then only itemsets of size 1 {x} and {y} are frequent. For the itemsets of size 2 possibilities are {x,y}, {x,z} and {y,z}. We can deduct that definitely the support of {x,y} will be smaller or equal to that of {x} and {y} individually. It could be that {x,y} is present in all 4 transactions containing y which would make it a frequent itemset. However, for itemsets of length 2 containing z it is already known that the frequency will not be big enough for making them frequent.

Most popular itemset mining algorithms build further on this property. The Apriori algorithm introduced by Agrawal et al. uses a breadth-first approach similar to the concept introduced above [15]. Zaki defined the Eclat algorithm to exploit a depth-first approach, which is based on re-arranging the transactional database and computing the set of transactions for each item [16]. This allows to calculate the support by computing a set of intersections. Han et al. developed the FP-growth algorithm based on a divide-and-conquer strategy to keep an efficient data structure for the database [17]. For each recursive step the database is filtered by the current itemset prefix, which shrinks its size, and speeds up support computation. In this paper, the Eclat algorithm is used, more specifically the method by Borgelt [18]. We are not sure about the fact that indeed the correct sequence in which the status codes occur is kept when they are stored. As such we opt to use itemset based mining approaches and not take the sequence in the identified itemsets into account.

2.3 Data-windowing

Status log data is time series data. The activation and de-activation time of each status log is recorded. As such each status code will be part of an activation time series and a de-activation time series. For simplicity the number of the status code is used as value. As such two time series of status values and corresponding timestamps are generated. To make these time series compatible with the pattern mining algorithms itemsets need to be created. This is done by windowing the data using a sliding window. In previous work on single turbine pattern mining, we used overlapping fixed windows spanning 2 hours [13]. However, overlapping windows result in higher support values than there are occurrences in the original time series, and need correcting. For farm-wide event mining we therefore use non-overlapping sliding windows spanning 24 hours.

2.4 Farm-wide support

We provide two definitions for computing farm-wide support that are applicable, in general, to a fleet's of devices, where each device logs events over a long period of time.

Vertical support is the number of windows where the patterns occurs in a single turbine. The farm-wide vertical support of a pattern, is the none-zero average of the vertical support for each wind turbine in the farm.
For example, if we create windows spanning 1 day, an average vertical support of 50 for the extreme wind direction singleton pattern, means this pattern occurs 50 times on average in a turbine.

Horizontal support measures the number of time the pattern occurs in different wind turbines during the same window, or period.
The mean non-zero horizontal support is computed by averaging the horizontal support over all windows.
For example, if we create windows spanning 1 day, an average horizontal support of 5 for the extreme wind direction singleton pattern, means this pattern occurs on average in 5 wind turbines.

We remark that certain patterns, such as pattern related to lightning, have very low vertical support, and high horizontal support.

For mining patterns having a high non-zero mean value of vertical support, we create a vertical transaction database for each wind turbine. We then mine closed itemsets in each transaction database (or turbine) and set a constraint on minimal vertical support. We then compute the non-zero mean of vertical support for each pattern in the entire fleet by looking at the the number of occurrences in all vertical transactional databases. For example, assumining 10 turbines, a total span of 1 year and a window of 1 day, we create 10 databases, each consisting of 365 transactions.

For mining patterns having a high none-zero mean value of horizontal support, we create a horizontal transaction database for each day, thereby creating a transaction for all wind turbines. We then mine closed itemsets in each horizontal transaction database and set a constraint on minimal horizontal support. We then compute the non-zero mean of horizontal support for each pattern in the entire fleet by looking at the number of occurrences in all horizontal transactional databases. For example, assuming 10 turbines, a total span of 1 year and a window of 1 day, we create 365 databases, each consisting of 10 transactions.

We remark that mining many small databases with constraints requires moderate resources, assuming the usage of an efficient algorithm for mining closed itemsets. The result of this step will be a set of farm-wide patterns together with their farm-wide support. In addition the time of occurrence and turbine of occurrence of each pattern is stored.

Step 2: Event detection in SCADA 1-second data

In this paper we focus on start and stop behaviour of wind turbines. As such these event types are extracted from the SCADA 1-second data. For this, we use an identification and classification approach we previously developed and documented [14]. In essence, this approach learns the time signature of typical annotated start and stop events. After this training step, it is able to automatically extract these events from time series data.

Step 3: Event fusion

In this paper, we aim to gain better insights into the co-occurrence of turbine response throughout a wind farm. To do this we merge the event sequences identified by the pattern mining approaches with the events extracted from the time series. The goal is to use the time series data as a way to go deeper in phenomenon understanding. We use the pattern mining to scan through large datasets of several years to identify farm-wide events. Once the events are identified we validate them using the events detected in the time series and perform an in-depth root-cause analysis using the raw SCADA 1-second data.

3 Experimental case

3.1 Overview

The experimental case follows the philosophy described in the method section. We use multiple years of data of multiple offshore wind farms. The dataset consists of 5 years of status log data of one farm with more than 40 turbines and 6 months of status log data and SCADA 1-second data for a farm with more than 40 turbines. The experimental case consists of two steps. First, the 5 years of status log data of the first farm is used to identify interesting farm-wide events. For this paper, we specifically target the example of a heavy storm. Once the farm pattern for a storm is identified, that specific farm pattern is searched in the status log data of the second farm. This implies transfer learning: patterns identified on one farm are transferred to another farm. To make it additionally challenging the turbines types and brands of the two farms are different. Once the storm pattern is identified, it is verified based on the events of the SCADA 1-second data that indeed a storm took place. Then the turbine response is further investigated in detail using the SCADA 1-second data.

3.2 Pattern mining on farm 1

The status log data has a long tail distribution. A very limited part of the status log data forms 99% of the total data. These status logs typically lead to trivial patterns, as we illustrated in [19]. Thus we aim to filter these trivial operational codes out. All status log data was first cleaned by removing standard operational messages depicting instantaneous values of power produced, rpm, pitch,… Moreover, to avoid trivial patterns only closed patterns with a minimal length of 5 were retained. Additionally, a non-zero mean horizontal support of 10 is required. Windows of one day are considered.

Code	Description
410	Yaw cable twist has been reset to a specified value°
412	Yaw cable was untwisted
460	Automatic restart after predefined time [s]
552	Start of automatic yawing action
560	Yaw control setting has been changed
568	Stop of automatic yawing action
712	Extreme direction of the wind with angle° at Wind speed x m/s
426	Nacelle position has been reset unexpectedly

Table 1: Pattern linked to storm conditions identified in status log data of farm 1

Table 1 illustrates an important pattern found from the fleet-wide pattern mining analysis. The pattern showed a non-zero mean horizontal support of 11. This means that the pattern was showing up at 11 turbines at the same time. Moreover, it occurred on a non-zero mean vertical support of 9.5 times with a standard deviation of 9 over the five-year data span. We expect the pattern might be occurring close to storm conditions. To validate this hypothesis, we try to find similar patterns in the data of farm 2.

3.3 Pattern searching in farm 2

The turbines of farm 2 are from a different manufacturer who uses a different ontology/taxonomy for defining his status log tags. This implies that the status codes of the first farm cannot be directly used in the pattern query for farm 2. To overcome this, generic text components were extracted from the status log descriptions in Table 1. These so-called key words were then used for the search in the status log data of farm 2. In the status log data of farm 2 the pattern illustrated in Figure 2 was found that links closely to the pattern of Table 1. Again, the subcomponents of excessive deviation between the wind direction and nacelle direction are found (orange dot in Figure 2). This links to the tag "Extreme direction of the wind with angle° at Wind speed x m/s" in Table 1. Moreover, there is extreme deviation between the needed and current yaw angle requiring to start yawing (green dot in Figure 2). This links to the tag "Nacelle position has been reset unexpectedly". These two indicate that the turbines are completely misaligned to where the wind is coming from. As such the turbines perform a yawing action, indicated by the tag "Yaw speed high" in Figure 2 and the tags "Start of automatic yawing action", "Stop of automatic yawing action" and "Yaw control setting has been changed". Finally, it seems that in many cases the targeted yaw angle is not achievable without prior unwinding of the power cable. In Table 1 this is identified by the tags "Yaw cable was untwisted" and "Yaw cable twist has been reset to a specified value°". In Figure 2 this is identified by the tag "Cable auto unwind". The comparison of the tasks shows that there is a high correlation between the status code pattern that was found in farm 1 and in farm 2.

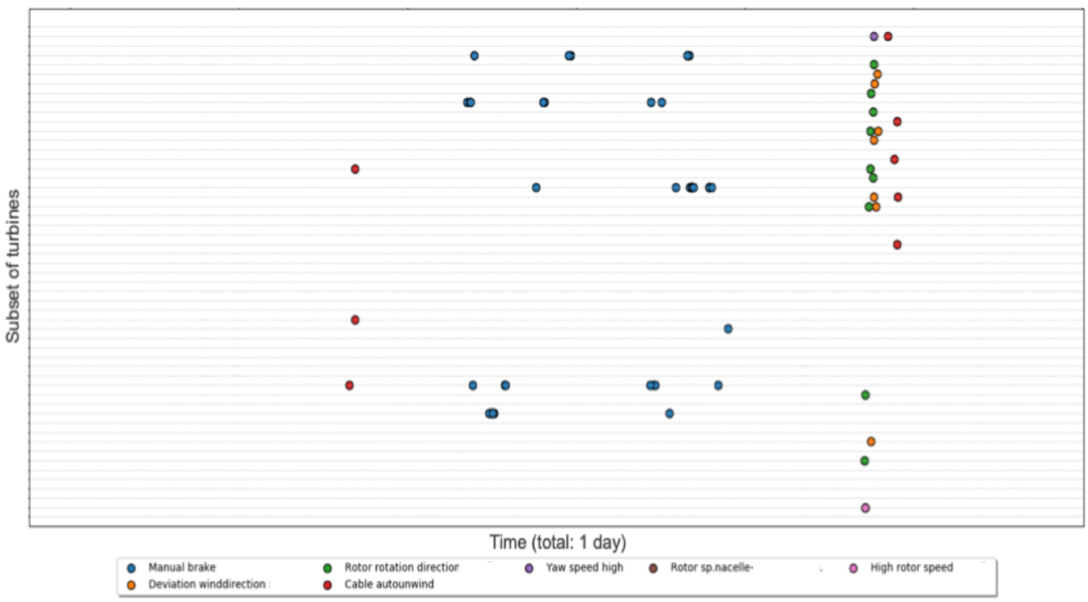

Figure 2: Status code representation for a subset of the turbines of farm 2 over the course of 1 day.

3.4 SCADA 1-second event analysis on farm 2

The status codes were thus used to identify the time stamps of the interesting events. As mentioned before, we expect the event sequence that was identified using the pattern mining approach to be linked to storm conditions. In Figure 3 it can been seen that although the majority of the turbines shows a certain behavior not all turbines are showing exactly the same response. This means that there must be local differences between the loading the turbines are exposed to. These local changes are investigated further using the time series SCADA 1-second data. Figure 3 shows the different start-up and shutdown events identified from the SCADA 1-second data. The blue zone indicates the time period for which the farm event is expected to take place. The shutdown events are clearly present for all turbines in the farm in contrast to the status codes where only a limited set of machines were showing events. This is due to the fact that the status log for stop is only recorded if the blade pitch evolves to 90°. Moreover, from the events in Figure 3 it is clear that several other farm-wide events occur later on. Once the turbines perform their final start-up no more events occur. This indicates that the event sequence is located in this limited time window.

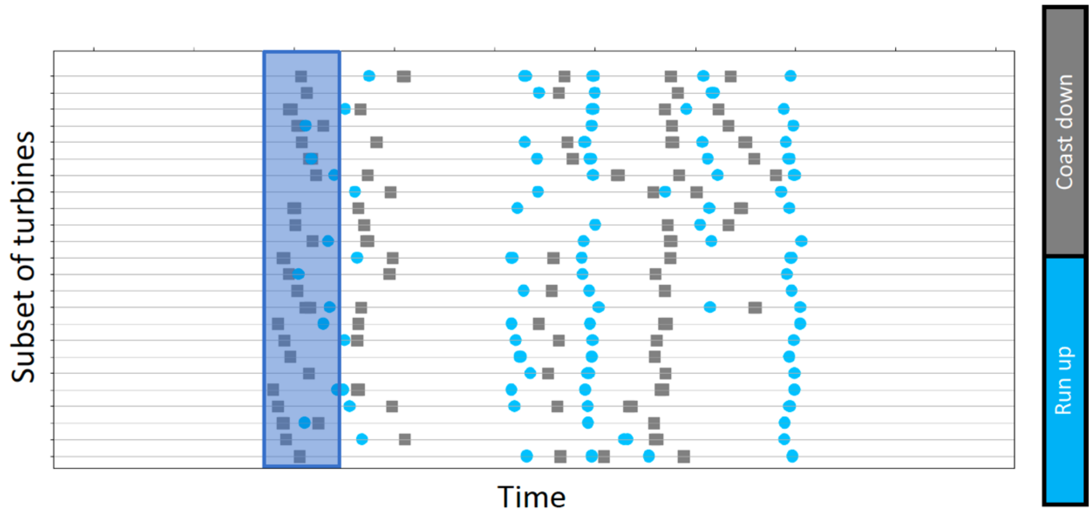

Figure 3: Events detected from SCADA 1-second data using signature detection.

3.5 Event fusion

To understand the different events that are taking place and why they result in the sequence that they do, we investigate the SCADA 1-second time series deeper. A representative turbine is taken that was showing identified pattern. Figure 4 illustrates the characteristic signals during the event: rpm, wind speed, wind direction and yaw angle. Based on the wind direction and yaw angle, it can be seen that there is a rapid change in wind direction causing an important misalignment of the turbine to the wind. This causes the machine to perform a stop. As shown in Figure 5, some turbines are experiencing a large rapid wind speed increase in addition to the sudden direction change. Many turbines are becoming strongly misaligned towards the wind at that particular moment. This matches with the presence of the tag type "Extreme wind direction angle °" in the status codes. The machine attempts to correct for the yaw misalignment by performing yawing action, as was indicated in the pattern. However,

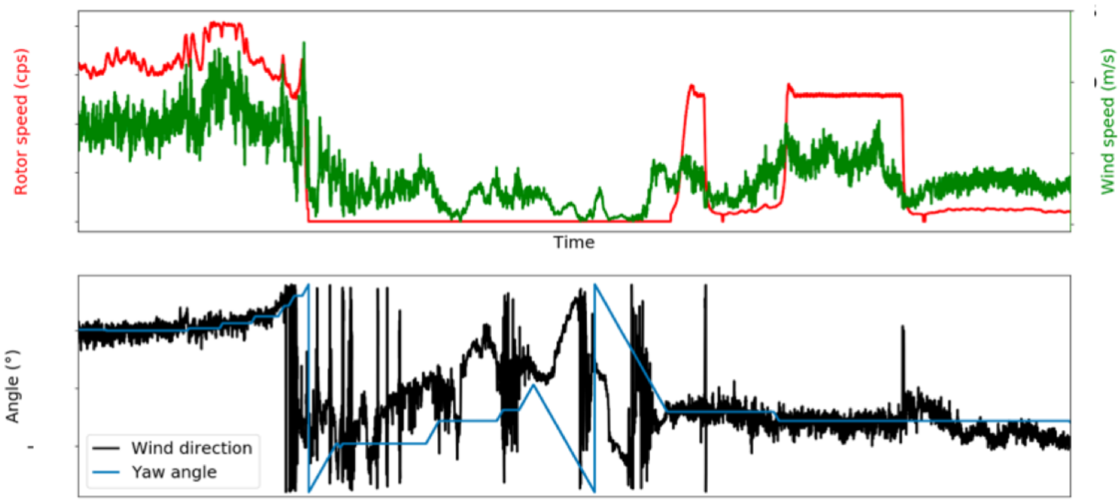

Figure 4: SCADA 1-second data of rotor speed, wind speed, wind direction and yaw angle for period of 2 hours to provide an overview of the response to storm condition.

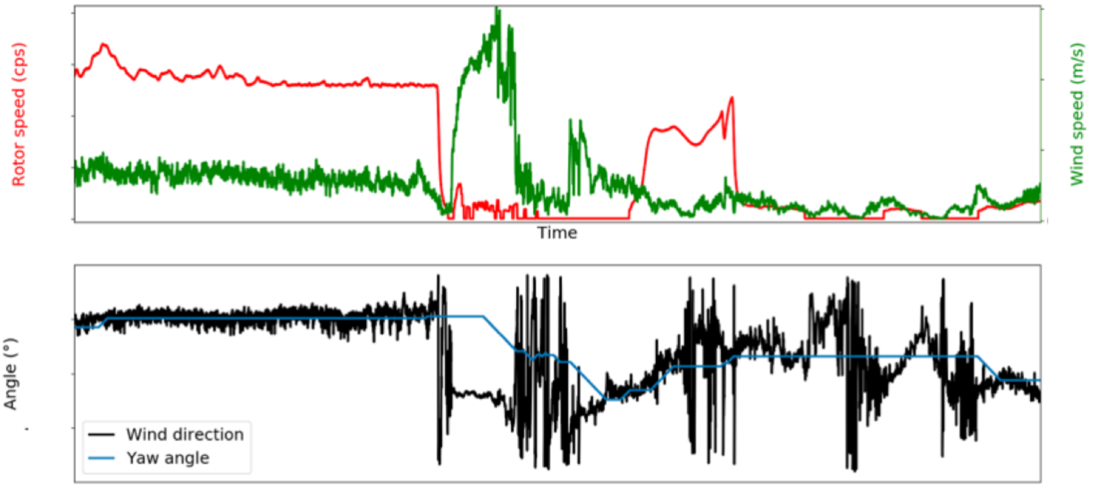

Figure 5: SCADA 1-second data of rotor speed, wind speed, wind direction and yaw angle for period of 1 hour to illustrate extreme yaw misalignment.

the misalignment remains substantial. After the sudden increase of the wind speed, the wind reduces significantly in speed. For certain turbines, as shown in Figure 4 the turbine is not able to reach the required yaw angle due to cable winding constraints. Thus, the turbine remains in idling condition and performs a cable unwinding procedure as was identified in the pattern. Figure 6 illustrates this cable unwind. The yaw angle changes linearly over time as the turbine unwinds itself. After this action the machine can again meet the required yaw angle. These actions match with the sequences identified using the pattern mining approach.

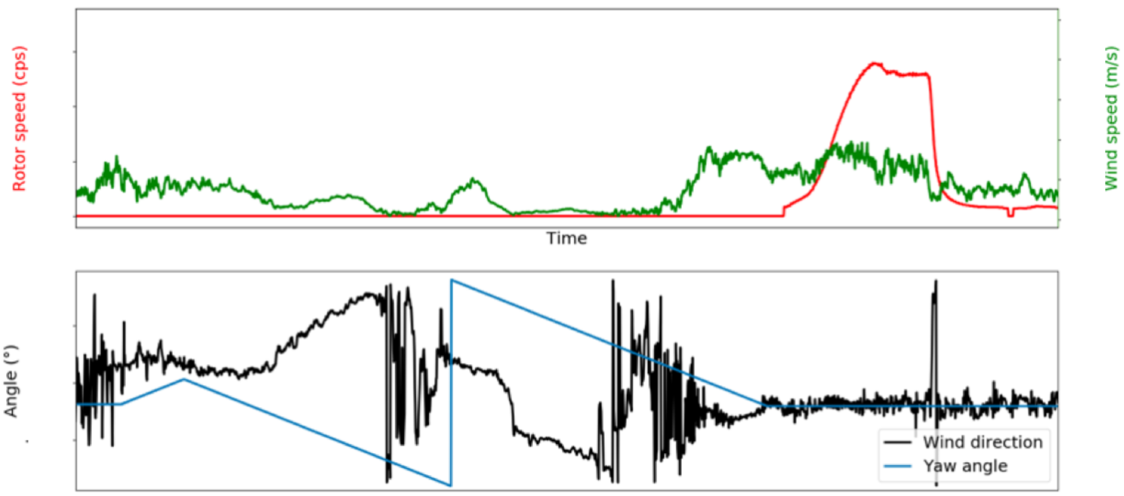

Figure 6: SCADA 1-second data of rotor speed, wind speed, wind direction and yaw angle for period of 1 hour to illustrate cable unwind.

4 Conclusions

This paper showed a methodology for identification of event sequences governing the response of wind turbines in wind farms. It comprised of farm-wide pattern identification and SCADA 1-second based event detection to determine the typical event sequences following farm-wide triggers. The methodology was illustrated by a storm event. Patterns were learned on the status logs of one farm and used to identify the storm event in a second farm with turbines from a different manufacturer. Moreover, SCADA 1-second data was used to show the validity of the identified pattern and to further understand the way the turbines reacted to the storm event.

Acknowledgements

The authors would like to acknowledge the IWT HYMOP project and VLAIO SIM MaSiWEC project for their support as well as the farm owners for sharing their data in the context of OWI-lab.

References

[1] FPG. Marquez, AM. Tobias, JMP. Perez, M. Papaelias: Condition monitoring of wind turbines: Techniques and methods, Renewable Energy, vol. 46, pp. 169-178, 2012

[2] Z. Hameed, YS. Hong, YM. Cho, SH. Ahn, CK. Song: Condition monitoring and fault detection of wind turbines and related algorithms: a review, Renewable and Sustainable Energy Reviews, vol. 13(1), pp.1-39, 2009

[3] B. Lu , Y. Li, X. Wu, Z. Yang: A review of recent advances in wind turbine condition monitoring and fault diagnosis, IEEE Power Electronics and Machines in Wind Applications, 2009

[4] K. Leahy, R. Lily Hu, I. C. Konstantakopoulos, C. J. Spanos, A.M. Agogino: Diagnosing wind turbine faults using machine learning techniques applied to operational data, IEEE conference on Prognostics and Health Management (ICPHM), 2016

[5] J. Tautz-Weinert, S. J. Watson: Using SCADA data for wind turbine condition monitoring – a review, IET Renewable Power Generation, vol. 11(4), 2017

[6] J. Helsen, C. Devriendt, W. Weijtjens, P. Guillaume: Condition monitoring by means of SCADA analysis, European Wind Energy Conference, 2015

[7] A. Kusiak, A. Verma: Monitoring wind farms with performance curves, IEEE transactions on sustainable energy, vol.4(1), pp.192-199, 2013

[8] C. Carrillo, AF. Obando Montano, J. Cidras, E. Diaz-Dorado: Review of power curve modelling for wind turbines, Renewable and Sustainable Energy Reviews, vol. 21, pp. 572-581, 2013

[9] M. Optis, J. Perr-Sauer, C. Philips, A.E. Craig, J. C. Y. Lee, T. Kemper, S. Sheng, E. Simley, L. Willimans, M. Lunacek, J. Meissner, J. Fields: OpenOA: An open-source code base for operational analysis of wind power plants, Wind Energ. Sci. 10.5194/wes-2019-12, 2019

[10] A. Kusiak, A. Verma: Prediction of status patterns of wind turbines: a data-mining approach, Journal of solar energy, vol. 133(1), 2011

[11] E. Gonzalez, M. Reder, J.J. Melero: SCADA alarms processing for wind turbine component failure detection, Journal of Physics: Conference Series 753 072019, 2016

[12] Y. Qiu, Y. Feng, P. Tavner, P. Richardson, G. Erdos, B. Chen: Wind turbine SCADA alarm analysis for improving reliability, Wind Energy, vol. 15, pp. 951-966, 2012

[13] L. Feremans, B. Cule, C. Devriendt, B. Goethals, J. Helsen: Pattern mining for learning typical turbine response during dynamic wind turbine events, ASME IEDETC, 2017

[14] P.J. Daems, N. Gioia, H. Vervaeck, C. Peeters, J. Verbeke, J. Helsen: Automatic detection of events critical for drivetrain health and lifetime from long-term field measurements, Conference Wind Power Drives, Aken Germany, 2019

[15] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo: Fast discovery of association rules, Advances in knowledge discovery and data mining, vol.12(1), pp. 307-328, 1996

[16] M.J. Zaki: Scalable algorithms for association mining, IEEE Transactions on Knowledge and Data Engineering, vol. 12(3), pp. 372-390, 2000

[17] J. Han, J. Pei, Y. Yin: Mining frequent patterns without candidate generation, ACM Sigmod Record, vol.29, pp. 1-12, 2000

[18] Borgelt: Pyfim toolbox, http://www.borgelt.net/pyfim.html
[19] P. Doro, L. Scerri, J. Helsen: Pattern detection in status codes as an optimization tool in offshore wind farms, European conference if the prognostics and health management society, Bilbao, 2016