Pattern Detection in Status Codes as an Optimization Tool in Offshore Wind Farms

P Doro[1], L Scerri[2], P Guillame[3], J Helsen[4]

[1,2] *Colorado State University, Fort Collins CO, 80521, USA*
pdoro@rams.colostate.edu, luke.scerri.10@um.edu.mt

[3,4] *Vrije Universiteit Brussel Ixelles, 1050, Belgium*
jan.helsen@vub.ac.be, patrick.guilluame@vub.ac.be

ABSTRACT

Renewable, sustainable energy is an evolving field and will soon become a desirable necessity to sustain an ever-growing growing population. One viable source is wind power which is trending towards large farms with many involved turbines. The sheer size of the analytical data derived from these sizeable wind farms poses both a challenge and an opportunity for farm level optimization. Data driven analytics and machine learning are making the larger and more useful data sets available to be analyzed. One method based on these techniques, pattern detection, is already used very successfully in fraud detection and many other big data industries. One source of ascertaining the state of a turbine is through the appropriate understanding of its status codes. Such codes can indicate a myriad of outcomes from operational events to alarm conditions. It is expected that these codes follow consistent patterns and being able to extract these patterns from the data can help us understand how certain sequences relate to the turbine behavior and subsequently analysis of historically linked patterns can aid in predicting certain events. For example if codes A and B and C tend strongly to occur within the same time window then following an A-B pattern one could confidently predict a corresponding C event within the time window. Such an understanding enables the error codes to reveal more than simply a snippet of information, but a productivity-enhancing, cost-beneficial operational regime. These could then be used to track and anticipate failure events. For such high level computing to occur you must take the data into a parallelized environment making it scalable to an entire wind farm over the course of several years. In this study the effects of a varied time window on how the patterns manifested themselves was analyzed by frequency of occurrence and subsequently validated by physical insights into turbine behavior. This approach and the results extracted are based on real data of a full offshore wind farm and could be harnessed as a simple yet powerful tool for large scale wind farm optimization.

1. INTRODUCTION

Wind power is a valuable, sustainable source of renewable energy. In particular it is of interest to renewable energy efforts to cluster turbines densely in offshore locations (Green 2010). Locating wind farms offshore enables expansion and scalability impossible or impractical for land based wind farms. This can yield previously unattainable levels of energy production, while minimizing ecological and social drawbacks. The drawback to having large farms offshore is twofold: minimizing maintenance costs, downtime and associated loss of revenue/power generation, as well as the challenges of remote operations and optimization of such a large amount of machinery (Helsen 2015). For wind turbines to remain a competitive energy source, every measure available should be employed to reduce the frequency of maintenance events and operational servicing (Herbert 2007). Figure 1 illustrates the many derivatives of cost per kilowatt of wind power. Offshore locations present some significant advantages, yet travel to the locations presents other problems: any maintenance activity will be magnified by the remote location. Addressing

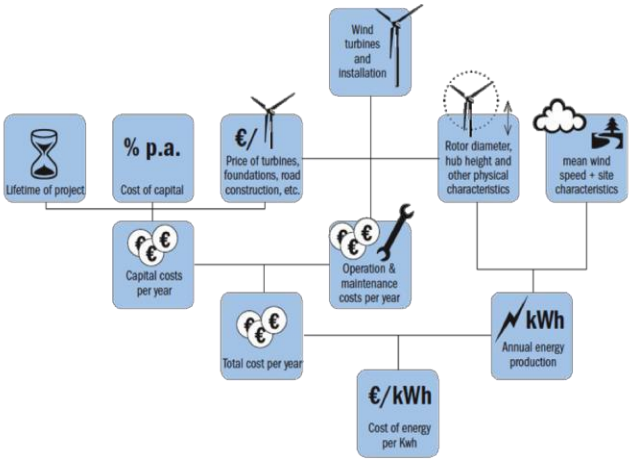

Figure 1. The Cost of Wind Power

and decreasing maintenance costs for offshore wind farms is a significant economic step toward developing wind power as a mainstream power generation source (Nelson 2006). For this reason remote diagnostics tools are particularly appealing for this application. This paper suggests and validates a methodology for pattern identification which may be used in later work as a prognostic tool. The goal of this tool is to reduce the yearly operation and maintenance costs with the goal to increase efficiency in wind power production and minimize cost per kilowatt-hour.

Machinery data-driven prognostics techniques attempt to track behavior of a particular process or machine, and use historical trends and patterns to predict and optimize the process. Wind turbines, due partially to their necessarily large scale, inherently represent a large upfront capital investment. To protect this investment, there already exist a myriad of data systems developed to monitor their performance and activity. This data can take many forms, but commonly present as a system of status codes, used to convey recurring operational states.

Status codes are used to convey a bulk amount of information in single number indicator format. Status code systems are usually in excess of thousands of codes available for the nuances of a single turbine unit, and will range in type of data communicated. Examples of such codes may be as simple as "Power on", or reflect an operational status such as "Wind speed high" or, more severely, indicate a failure mode, such as "Emergency shutdown". More generally they can be defined as a change in turbine parameter (Kusiak 2011). Each is relayed at a particular frequency and duration, and often they are sent in groups indicating operational states for more than one sub-system within the turbine unit. For example, a packet may only send one status code, or it could contain multiple status codes such as;

- Error - a full system indicator
- Emergency shutdown - a separate, full system indicator
- High wind speed - an environmental indicator
- Generator 2 temperature information – an operational subsystem

These dynamic groupings are what gives rise to the fundamentals of this paper and the interest in locating relationships or patterns within the data. An understanding of these codes and their patterns, including temporal patterns of the codes, can provide insight to the dynamic characteristics of an operating turbine. There exists a data analytics opportunity for low cost remote optimization (Kusiak 2016). The analytical techniques involved will address the comparative status codes from an entire farm of turbines and the effect of widening the sampling time window to determine how patterns are linked temporally and more generally how they evolve over larger time windows. Previous work in this area has been limited to analysis of a single turbine or a single row. This paper presents for the first time a farm level and even multiple farm level analysis. As these patterns rarely manifest in normal operational data it is crucial to have input from a number of turbines (Kusiak 2006). Realization of the full utility of this technique's predictive ability is realized as the input of data from turbines grows.

2. APPROACH

In order to extract these patterns from the full farm of turbines, each contributing to a stream of incoming data, a pattern detection process was developed that was based on time windows and had no selection bias. When acquiring such large amounts of data certain problems immediately present themselves.

2.1 Analysis Architecture

Firstly, simply to accurately record and store this data requires advanced data basing tools. For this application No-SQL (Not only Structured Query Language) was implemented as the tool of choice and was integrated into the system. This data basing tool provides excellent accessibility and reliability for large data sets. It also provides the security of data retrieval even in the event of a node failure. Additionally, it provides scalability far beyond traditional and well known SQL infrastructure (Helsen 2015). Secondly, parsing and processing this data is computationally intensive. Non-parallelized querying of related values from these tables proved to be far too tedious for reasonably scalable analysis to occur. To solve this problem, Apache Spark (Spark) was implemented as the parallel coding environment. Spark enabled the creation of Resilient Distributed Datasets (RDD) which lowers the processing time significantly through distributed computing (Zaharia 2010). For this application a custom Spark process was created, simply employing capabilities of executing many serial RDD transformations resulting in a much faster parallel process. On the cluster of nodes originally set up on for use of the No-SQL database the hardware groundwork was already laid for parallel computing. This was an essential step in realizing the desired results, as previous attempts proved too slow for reasonable analysis times. Using the same framework, but lacking parallel computing, which was attempted in proof of concept trials, proved to be too slow or required a bias on the data. Here, a bias indicates a preselection of code sequences (patterns) anticipated to be critical. Applying such a bias corrupts the resulting pattern detected as not all combinations can be considered. Spark's parallelized computing environment such a bias is not necessary. Spark also demonstrates scalability to larger data sets. More specifically, in many parallelized algorithms, the amount of data to be processed start to create a bottleneck, such that at a certain quantity of data the technique becomes computationally impractical. This scope of this work already encompasses an entire farm of data and shows promise to scale further. The Spark framework was interfaced with a python script from which analytical parameters were input.

2.2 Methodology and Process Flow

The status codes are initially received and encoded in the No-SQL network. Then, the scope of analysis is defined. Each analysis can specify a series of key parameters that defines the scope. This is explained in greater detail below. Then, the data tables are then broken apart and partitioned onto the nodes of the cluster and queried in parallel extracting codes and associated information that falls within the window defined. These objects are transformed to an RDD and the pattern detection algorithms using sorting and grouping commands are executed. Through this process a set of patterns, turbines affected, and frequency values is obtained. To understand how frequently recurring patterns develop over larger time scales, a sliding time window was employed. This accurately captures and compares patterns occurring over different time scales. The sliding time window described was constructed by simply defining a time gap over which patterns would be considered temporally linked. Status code data was organized chronologically and each subsequent code was either added to create a pattern or lengthen an existing pattern if the time gap between patterns is less than the prescribed amount. Otherwise the window closes and the assembled group of codes is taken to be a complete pattern for that window of time. This process is repeated on all turbines over the set total length of analysis. Time windows were selected based on divisions of standardized sampling rates for operational data in SCADA (supervisory control and data acquisition) systems. The goals of this work (as follows) are to understand three different analysis capable through altering the parameters.

1. Pattern differentiation across turbines in the same farm, but in different locations
2. Pattern evolution and growth with widening detection window
3. Detection of patterns associated with particularly critical status codes

For this analysis normal operation of turbines will be the focus. The primary goal in summary is to validate that they do indeed follow an intuitive pattern based on physical insight. Secondly, understanding the status code patterns as they represents an action the controller takes to an event. Identifying sequential events helps to characterize how the controller is responding to events. Analysis of historical data and current incoming streaming can then be used in future work to examine abnormal patterns and more complex associative techniques to analyze failure data and do supervised learning.

3. EXPERIMENTAL CASE

The dataset being analyzed in this work is based on data from a wind farm consisting of over 50 offshore turbines. The status codes were reported from a set of approximately 150 codes. For a list of codes and their meaning refer to appendix 1. It was proven that the methodology was capable of generating analysis for the entire farm with the support of a cluster of 3 high performance computing machines. Key variables that were adjusted based on the particular analysis were; the set of turbines which were being analyzed, the set of error codes to be considered, and the time gap between patterns from a single turbine. Additionally an overarching time span was set; in this case, 2 years of collected data from each turbine was analyzed. While an analysis of all turbines with all status codes was possible, it was computationally intensive and often obscured subtleties that lay in analyzing a subset.

By not pre-selecting particular patterns and tracking them, a selection bias was avoided. The patterns generated by the algorithm were a direct result of temporally related patterns for this analysis in all cases the farm was considered and the visualization was selected for clarity. Adjusting the selection of turbines allowed insight into row- or column- specific patterns. This allowed for the detection of location specific problems. For example, using a spatial analysis one could detect if an emergency related error pattern occurred on only on the edge of the wind farm, or was most often found in a particular location of a group of turbines. Expanding and collapsing the time window gave insight into the extended patterns that only manifest over longer time intervals. Using the above example, both A-B and A-B-C patterns may have very high frequencies of occurrence, however A-B-C may only occur on time windows longer than one minute.

4. RESULTS

For this analysis, normal operation of turbines will be the focus. The primary goal is understanding normal turbine behavior through use of readily available status codes, while layering in temporal-linked filtering, a technique made possible with this methodology. The visualizations presented below display the most frequently occurring status code patterns occurring within the specified time window (eg 1s) and identical time spans. ‘String’ indicates a row of turbines in a wind farm; the following number represents the turbine’s columnar position, yielding its place in a relative, rectilinear grid. Thus the leftmost column in the figure 2a represents turbine A1 exhibiting the sequence (443, 461, 447), where each code occurs within a second of the next, a total of 150 times over the 2 year time span. Turbines rows were selected arbitrarily for each visualization and while code definitions are accurate code identifier number is obscured to protect manufacturer secrecy.

Validation of the methodology and developed tool will take the form of discussing the physical significance of particular status code and their relevance to what pattern sequences are developed, compared to what could be expected. The results also confirm the feasibility of this methodology on larger data sets, as generation for each set included input data from every turbine over a two year time selected. Every analysis run amounted to less than 3 minutes of computational time, which was not previously possible without parallel computing. For this work, isolation of particular codes will be considered briefly, although only in an attempt to once again validate that such a technique produces results which are intuitive based on knowledge of the observed meaning of the status codes.

4.1. Goal 1: Comparative analysis of turbine status code patterns in different locations

Displayed in Figure 2a is the visualization of turbines 3 rows in a wind farm. It is clear that while status codes differ slightly from turbine to turbine, all exhibit patterns that occur more frequently, indicative of a successful analysis. Notably, 443 ("Error"), then 447 ("Stop") is almost an intuitive pattern, and appears in every turbine as one of the most frequent patterns. Given the narrow time window of only a second, which prohibits the recognition of more complex patterns, this seems logical; these code combinations are common occurrences. In fact, in such a small time window, nearly all codes or code patterns are limited to the basic level. Codes 445, 449 are "Emergency", and "Pause", respectively. Codes 443, 445, 447, and 449 make up the bulk of the top patterns in every string, which again makes sense as this short time window only detects virtually concurrent patterns. Such simple codes are logical to be grouped. More interesting is the emergence of the 203 (emergency stop) status code patterns occurring more frequently for turbine A5. This clearly indicates an operational imbalance or recurring malfunction in turbine A5, which would indicate an escalating need for timely intervention or maintenance. Also interesting to note is that while all turbines should be environmentally experiencing the same wind, they do not exhibit uniform status code feedback.

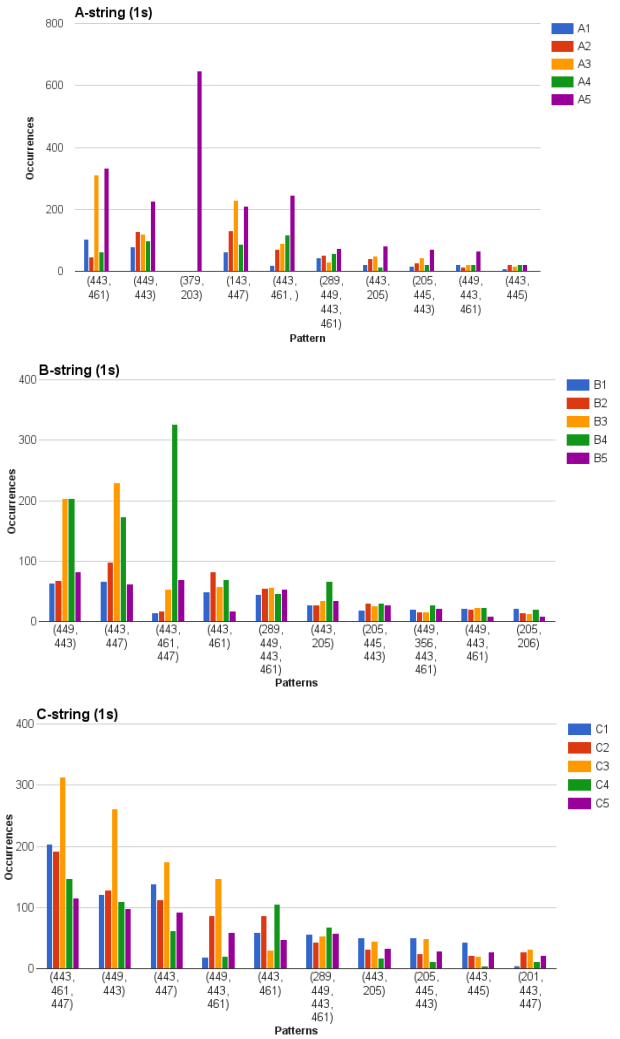

Figure 2. Top patterns from strings A-C

4.2. Goal 2: Analyzing the change in patterns with a widening time window

In Figure 3 the patterns include codes not previously included in patterns corresponding with a widening time window, and predictably, the status code groupings lengthen as a less selective filter is applied. Also notably, we recognize that common operational patterns (229-231) are thoroughly dominant at larger time scales. This is the transition from star to delta configuration which is a feature of the controller to optimize power harvesting. For this particular group of turbines it occurs over a duration of less than one minute. It is indicative of normal functioning and is a very common controller operation. From this we conclude that indeed logical patterns are being mined from the algorithm even at longer time scales For this reason this first pattern was excluded for the 5 minute trial to show greater detail of latter codes. Furthermore, the 229 and 231 codes manifest themselves in many other common operational code patterns. Again, while redundant, this validates that the patterns developed through the technique follow intuitively from physical understanding of the system, bringing credence to the methodology as a whole. As the window widens we can see which turbines or single turbines are exhibiting problematic patterns.

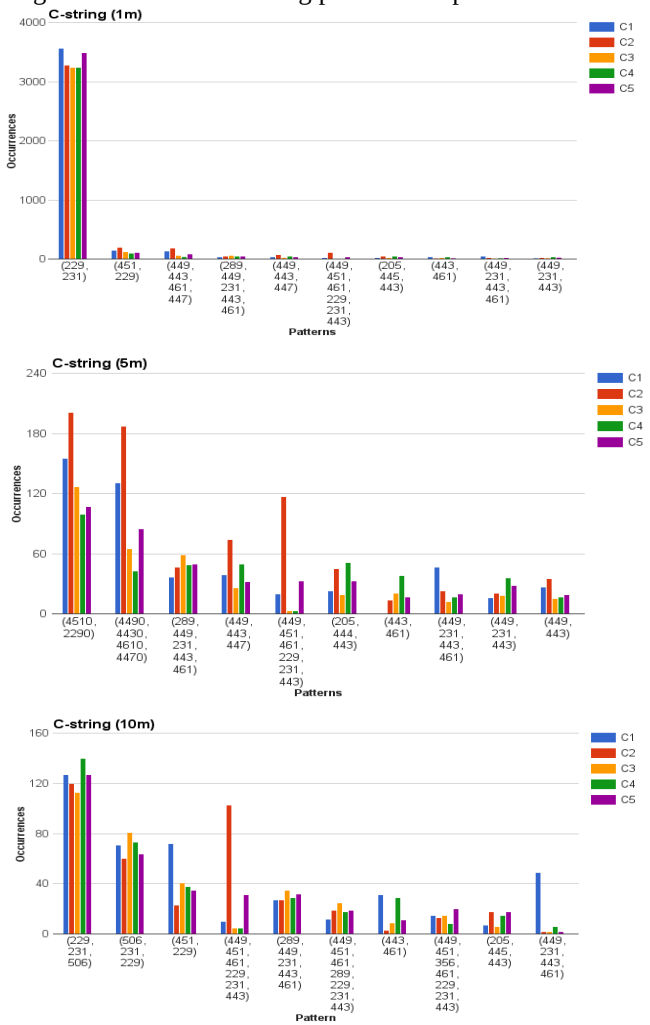

Figure 3. C-string at different time windows

4.3. Goal 3: Patterns containing critical status codes

In beginning to realize the predictive capabilities of this tool, we examine a pattern of critical interest; 203 or "Emergency stop". Emergency stops have the potential to not only diminish power output of turbines, but more critically to magnify and progress internal existing flaws. Based on the data above from a 5 minute window, C4 can be clearly recognized as a turbine with recurring issues. Other components to the pattern are again primarily operational data which is normal for this low of a time window where basic operational codes tend to dominate.

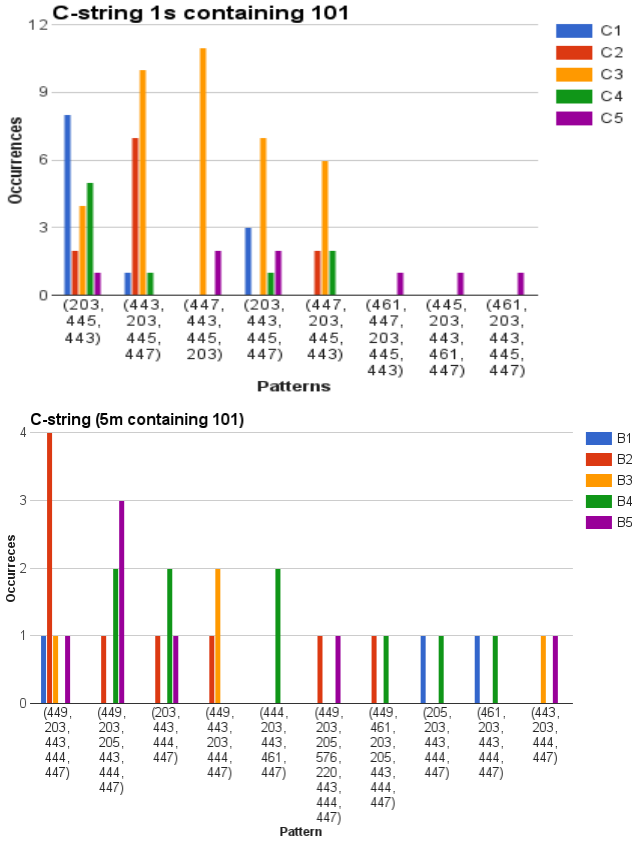

Figure 4. C-string top patterns containing 203

5. DISCUSSION AND CONCLUSION

We find that the pattern detection via parallel processing of temporal influenced code sequences algorithm is assumed to be valid, as patterns that are known to occur based on physical insight. It is also observed, based on the results that the patterns detected can logically expand to include progressions of status codes that are not intuitively apparent. It should also be noted that only sequential patterns were noted and no pattern partitioning was integrated. This will also be a topic for future work with more advanced algorithms. Indeed the expansion of patterns was by design in hopes to draw out a pattern long enough as to suggest a "unique signature" for a particular event. Understanding of the progressions of these error codes and signatures can be used in future work as an early indicator for potential failure modes.

Prognostics techniques can combine data from a particularly troublesome string or row, with other forms of data (SCADA, weather information, maintenance records, etc) and the origin of flaws can be inferred. While 203 was highlighted here, clearly other operationally limiting patterns could also be identified and more closely managed. Another pattern of particular interest is patterns which include the code 289, "High wind speed". It is intuitive that as wind speed increases so too, does the maximum power output of the farm. This is a desired outcome. Kusiak (2016) found that current management and maintenance practices, as well as control methodologies of wind turbine farms are falling far short in realizing maximum potential output. Maximum wind speed where power is being generated is a desirable outcome, yet it then becomes critical to track turbines where high wind speed status code is associated with error or emergency code. This tool also addresses the interesting variability in patterns between individual power producing units in the same farm, or even the same row. While the degree of variation is relatively slight, it continues to provide evidence to the theory that taking data from all turbines in the wind farm can yield new and interesting information about the operational and maintenance status of the wind farm.

Implementation of this tool is low cost and supported. Future work can delve further into examining relationships between codes related to critical fault and pattern family development. It seems likely that expansion and further work with this tool presents an effective and efficient solution to the problem of remote optimization in offshore wind farms. This techniques is already used in genome sequencing and fraud detection, but based on the results of this work has applicability in an industrial context not limited to wind turbines

REFERENCES

Green, Richard, & Nicholas Vasilakos. (2011). The economics of offshore wind. *Energy Policy* 39 (2), 496-502.

Helsen, Jan, De Sitter Gert & Jordaens Peter. (2015) Long-Term Monitoring of Wind Farms Using Big Data Approach. *Industrial electronics, IEEE Transactions*

Herbert, GM Joselin, et al. (2007). A review of wind energy technologies. *Renewable and sustainable energy Reviews*, 11 (6), 1117-1145.

Kusiak, Andrew, and Shital Shah. (2006). Data-mining-based system for prediction of water chemistry faults. *Industrial Electronics, IEEE Transactions*, 53 (2), 593-603.

Kusiak, Andrew, and Anoop Verma. (2011). Prediction of status patterns of wind turbines: A data-mining approach. *Journal of Solar Energy Engineering*, 133 (1).

Kusiak, Andrew. (2016). "Renewables: Share Data on Wind Energy." *Nature* 529, (7), 19-21.

Nelson, D. B., M. H. Nehrir, and C. Wang. (2006). Unit sizing and cost analysis of stand-alone hybrid wind/PV/fuel cell power generation systems. *Renewable energy* 31, (10), 1641-1656.

Zaharia, Matei, Chowdhury M, Franklin M J, Shenker S, Stoica I (2010). Spark: Cluster Computing with Working Sets. *HotCloud* 10 (10-10) June, 22.

APPENDIX

Appendix 1. List of status codes included and their meaning

- **201 –** Too many Auto restarts
- **203 –** Emergency stop
- **229 –** Generator 2 in
- **231 –** Generator 2 out
- **289 –** High wind speed
- **279 –** Engage brake
- **379 –** Error
- **443 –** Emergency
- **445 –** Stop
- **447 –** Pause
- **451 –** Run
- **461 –** 60 second auto -restart