Automatic Anomaly Detection in Offshore Wind SCADA Data

Steffen Dienst and Jonas Beseler

***Abstract*—We propose a computationally simple anomaly detection method that assists operators of offshore wind parks with monitoring the operation of wind turbines. Previously published methods focus on creating high quality predictive models for specific physical operational aspects of turbines, like power production or temperatures of the gearbox in relation to wind speed and other exogenous factors. SCADA systems of wind turbines typically provide many more sensor data time series than are being used for monitoring purposes. We show how this data can be used to automatically learn a large number of simple models that in sum can alert the operator about a variety of potentially defect related changes in different components. A number of different insights applicable to similar problems are provided in the conclusions. The system was developed and applied in an offshore wind park and is used to support predictive maintenance.**

I. Introduction

Operating a big number of wind power plants efficiently is a challenging task. Each individual wind energy converter (WEC) has can differ in several ways: structurally, current health conditions of components and sensors and parametrization of the controller. Physical inspections of WECs are expensive and may not even be possible for longer stretches of time due to weather related restriction. In offshore wind parks the problem is even bigger. There is be only a limited number of opportunities to fix any physical problems in an offshore wind turbine throughout the year. These facts mean that monitoring and comprehensive interpretation of the SCADA data produced by these turbines is essential for long-term economic success. Only if the operator is aware of defects or performance degradation of components as early as possible, he can efficiently operate the WEC.

The offshore wind park Global Tech I (GT1) is located about 140km of the german north-west coast and officially started production in autumn of 2015. It comprises 80 WECs of type Adwen AD 5-116 with a per turbine power rating of 5MW.

Figure 1 shows a schematic view of the GTI park topology where each WEC is represented as a pie chart showing the relative times of the day spent in a specific operation mode.

Each turbine is equipped with 313 different sensors that log the minimum, maximum, average and standard deviation for each 10 minutes interval. Additionally, there are several counters as well as event logs. In case of an error a ring memory provides details measurement data for each sensor in a higher sampling rate of 10ms. Each WEC has its own unique parametrization and is operated individually. Also, due to the harsh environmental conditions present in an offshore setting, sensors have a significant probability of breaking. In our experience, at each given time up to 2% of all sensors may be broken.

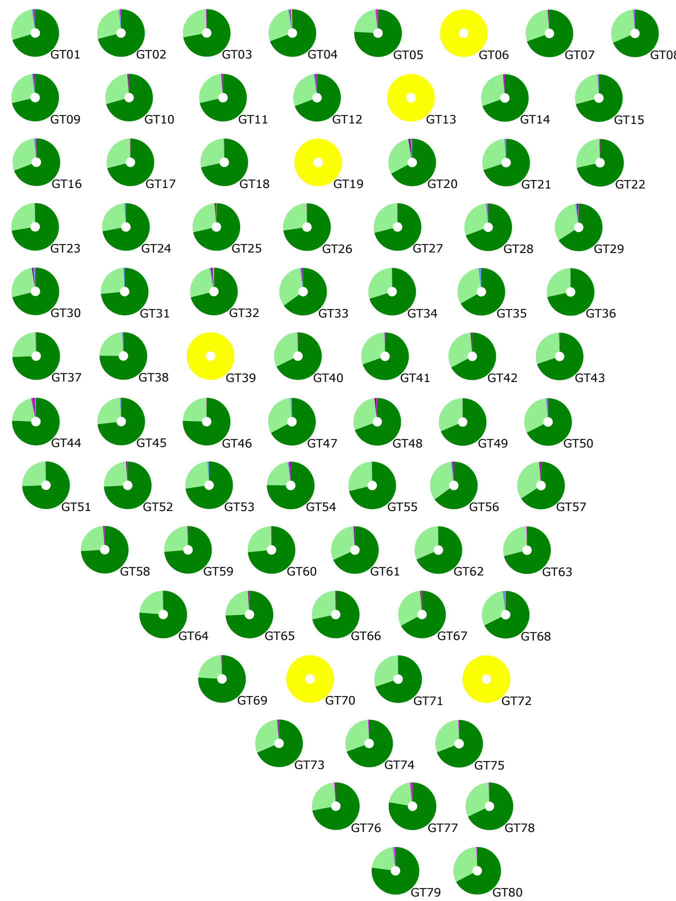

Fig. 1. Dashboard overview of the offshore park's topology. The colored arcs show the proportional time each power plant spent in a specific operation mode. Yellow represents maintenance, light green stands for 'ready' and dark green means 'in production'.

A. Problem Description

All these conditions present significant challenges when using the measurement data to learn about the current health condition of the WECs. Due to the remote location, the difficulties and

S. Dienst, ScaDS Big Data Competence Center, University of Leipzig, Germany.
E-mail: steffen.dienst@uni-leipzig.de
Telephone: +49-341-9732238
J. Beseler, Global Tech I Offshore Wind GmbH, Germany.
E-mail: J.Beseler@globaltechone.de

costs involved in carrying out maintenances and repairs and the size of potential economic losses in case of lost production the operator was interested in using the SCADA data produced by each turbine to get a better, more complete awareness of any change in their health states. SCADA systems internally already employ a number of rules and thresholds that monitor different sensors with the main purpose of stopping the WEC if a potentially damaging condition is observed. Thresholds per sensor are insufficient, though. Many interesting changes, like continuous temperature increases in bearings or slow pressure losses typically manifest themselves within a normal value range for the respective sensors. Thresholds alone are insufficient.

Changes in the interaction of the WECs components that may indicate defects were only visible through manual inspections of sensor data visualizations by data analysts. These inspections must inevitably be incomplete, as there are $313 * 80 = 25.040$ sensors available in the GT1 park.

The task posed was: Automatically identify individual WECs with atypical measurements with high accuracy using only existing operational SCADA data from wind turbines. From a machine learning perspective this task can be formulated as: How can a system learn to distinguish between "normal" and "anomalous" operation of as many components in a WEC as possible with the least amount of additional, manually added knowledge? The main requirements for such a system were:

1) **No/as little metadata as possible** The system should not need to know about the mechanical and electrical processes or the components of a WEC.
2) **No explicit, manual definition of "normal behaviour" per sensor** In many cases the range of valid values depends on the current operation mode. Also, the range may change over time, for example due to degradation or changes in the parametrization. Requiring the operator to provide an explicit expectation per sensor is not feasible.
3) **Fast learning for quick iterations** Parameters change, components get swapped. In each case the algorithm needs to relearn. Excessive earning times are not feasible.
4) **Efficient model application** The system should not require special hardware to run. Typical off-the-shelve hardware in use by the operator should suffice.
5) **Low false alert rate** The operator is confronted with a number of automatic warning and error log messages per WEC and day. Any additional messages produced should be as relevant as possible to be useful.
6) **Complements SCADA alerts** The manufacturer has a number of internal checks in its system. Our anomaly detection algorithm should not try to replicate these rules but rather focus on monitoring as many previously unobserved aspects of the WEC's operation as possible.

B. Contributions

We presents an efficient and computationally light method to automatically learn reference models for the majority of sensors of a wind turbine. We then show how to use these models to monitor the continuous stream of SCADA data to identify anomalies by comparing sensor data with predictions made by these models. We give details about the heuristics used to identify the root causes for each anomaly. These anomalies get presented to a data analyst who can use this information to manually classify the changes and use this knowledge to reduce the potential economic impact of component malfunctions. We present a preliminary classification of behavioural changes found in the WECs of the GTI park in the spring of 2016. Finally, we list some lessons and experiences we learned from applying the presented method for the purpose of enabling predictive maintenance.

II. Related Work

We do not focus on condition monitoring in the form of vibration analysis of generators and gears, because these aspects are well understood and the market offers a broad variety of products. Rather, we are interested in using all SCADA data a wind turbine can output to identify malfunctions the operator might not be aware of, yet.

There are a number of published methods to analyze SCADA data with the explicit goal of identifying malfunctioning components in wind turbines [1]. They use machine learning techniques like artificial neural networks [2], self-organizing maps [3] or support vector machines [4]. Although the learning capability of these methods are vast due to their proven successes in learning non-linear relationships, they are also computationally very demanding, which means a big upfront hardware investment is needed and the learning time can be easily in the order of days or weeks. A periodic relearning of models due to degradation or changes of components as well as parameter changes is necessary during the lifetime of a WEC. The learning time is therefore an important, especially if the number of models is large.

Also, the methods referenced typically focus on using models that focus on specific physical behaviours of wind turbines [5] [6] [7]. Among these the most prominent aspects which should be monitored are wind power curves, followed by nacelle misalignment. Both of these aspects have a direct negative impact on the revenue due to the ongoing loss of power production that can be observed if one of these two aspects shows a malfunction.

In contrast to these previous works, we address the gap of monitoring as many functional aspects of a wind turbine as possible while still keeping requirements like memory usage, computing power and computing time as low as possible. We are aware of the fact, that each functional model created by our method might possibly be improved by combining the knowledge of experts in the application as well as the data mining domain. However, we are not so much interested in improving on individual model quality but rather in providing automatic monitoring of as many aspects of a single turbine as possible with as little effort as possible.

III. Approach

The problem can be restated as: Create an explicit model for each individual sensor that predicts the expected value for each

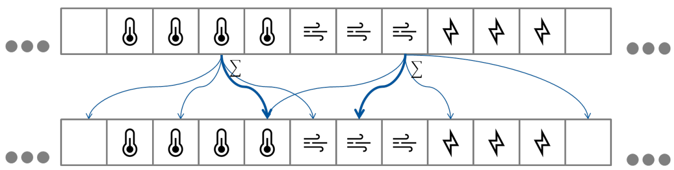

Fig. 2. Schematic representation of two LASSO regression models. Sensors represent temperatures, speeds and voltages, the targets of the blue lines signify the automatically chosen features, their width represents the regression weight.

time interval t using the measurements of a subset of all the other sensors of the same time as well as of the most recent past $t-1$, $t-2$ etc. We can then use these models to compare the measurements at each time step with the predictions. In the case of a significant divergence we have three possible explanations: 1. the model is wrong and the predictions irrelevant; 2. the measurements indeed differ from the expectation indicating a qualitative change in the behaviour of the component being measured (the model is correct and the target sensor of the model is the cause of the change); or 3. at least one of the sensors used as an input of the model is defect.

As figure 3 shows, we employed a cyclical process throughout our project that lead to higher quality results per iteration: The creation of the models, their usage to create predictions and identify divergences and the heuristics that interprets these divergences to identify the most probable root cause are fully automatic and will be described in more details in the following subsections. The operator of the parks then gets the results presented interactively. He can use different visualizations to manually classify the changes found and decide, whether and how to act upon them. The final step is feeding back findings regarding the quality of the anomalies found into settings for the model learning steps. These changes to the settings can be excluding sensors from being used as targets or inputs in models as well as changes to thresholds.

The main mechanism we used is multiple linear regression of sensor data. This method is computationally light, but it can't capture non-linear relationships. Since we are not interested in few but very accurate but rather in finding many models representing the different functional aspects present in a wind turbine, we found that the reduced accuracy of using linear approximations does not hurt the usefulness of the method. The regressions try to represent each sensor value as a weighted sum of an adequate selection of all the other available sensor's values. Figure 2 shows a schematic view of the main idea of these regression models. It shows two models, one for a temperature sensor and one for a wind speed sensor.

A. LASSO-Regression

Multiple linear regression for applications with many input features tends to produce over fitted models (low bias, high variance). Therefore, an essential step to produce linear models with adequate low variance feature selection is an essential step. Tibshirani published a method called **L**east **A**bsolute **S**hrinkage and **S**election **O**perator [8] that accomplishes both: it produces a multiple linear regression with an automatic feature selection. It does this via a regularization factor in the regression formula

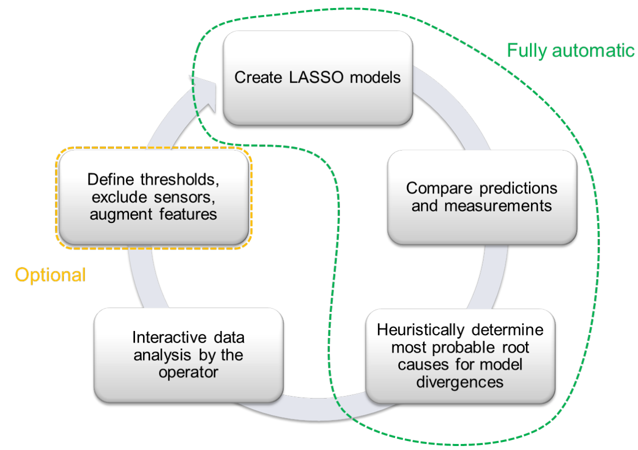

Fig. 3. Our method follows a cyclic process: the models are learned using the measurement data in a reference date range, the results are interpreted by a data analyst and any necessary changes to the algorithm parameters lead to a re-learning of models.

1, where d is the number of input dimensions, y the regression target, $\mathbf{X}$ are the input values and λ the maximum sum of the feature weights:

$$\min_{\beta} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 s.t. \sum_{j=1}^{d} |\beta_j| < \lambda \quad (1)$$

The effect of λ is a shrinking of weights in β towards zero, which effectively means that the majority of inputs will get a zero weight, which means they are irrelavant to predict the target value y.

Thanks to several improvements in the implementation of the underlying optimization algorithm in [9] and [10] this method can be used to create hundreds of individual LASSO regression models per minute.

B. Model Learning

Our model learning method has two steps: derive additional features ("virtual sensors") from the sensor measurements and then learn multiple individual LASSO models for each sensor. Each model can use data of all sensors available. In addition, we add time lagged versions of each sensors. This is useful for physical processes that manifest themselves with time delays, like for example temperature propagation in the nacelle or pressure changes due to increases in currents of pumps.

For each sensor, we try to learn more than one model to capture as many similarities between sensors as possible. To avoid reusing the same preditive sensors, after having learned a regression model we exclude the best predictors of this model and run the model finding code again. If the resulting model's quality is too low, we discard this model.

Table I shows one learned model. The most prominent relationship it captured is the direct linear dependency of the current used by the coolant pump and the water pressure produced. In addition, the algorithm selected other sensors with decreasing weights, that don't seem to be related to the

TABLE I. EXAMPLE OF A LEARNED LASSO MODEL TO PREDICT THE SENSOR "COOLANT PUMP MOTOR CURRENT 1". COLUMNS SHOW THE SENSOR NAME OF THE PREDICTOR, THE TIME LAG APPLIED TO THE SENSOR TIME SERIES AND THE RELATIVE WEIGHT β.

Name	**Time lag**	**Relative weight**
Coolant Pressure 1	0 min	0.8856
Inverter Case Temperature	-20 min	0.0477
Temperature Drawing-Off Air	-20 min	0.0176
Axis 3 Contouring Error	-20 min	0.0133
Temperature Drawing-Off Air	0 min	0.0127
Temperature Drawing-Off Air	-10 min	0.0083
Axis 3 Battery Discharge Current	-20 min	0.0067
Axis 1 Battery Discharge Current	-10 min	0.0047

water pumping process. This is purely because the learning data available resulted in slight model quality improvements if these sensors were used in the prediction. If these models were used primarily to learn about the relationship of the sensors and the underlying processes these additional predictor were not benefitial. But, since we are interested in identifying sensors with anomalous measurements, each additional model that uses a sensor increases the chance, that this model will produce diverging predictions which gives our root cause finding heuristics more hints about the true culprit.

Algorithm 1 shows pseudo code for the details of the learning step.

Algorithm 1 High-level algorithm - Learn models

Parameter $|models|$ $\triangleright$ how many models per sensor
Parameter $quality$ $\triangleright$ minimum R^2 per regression model
procedure CREATEFEATURES($columns$)
 $features \leftarrow columns \setminus$ {filtered columns}
 Add up redundant component measurements
 Add non-linear features (squares, roots, logs,...)
 Add lagged features ($columns$ of time t_{-1}, t_{-2}...)
 Store statistical informations for each feature (average, standard deviation, extent,...)
 return $features$
end procedure
procedure LEARNMODELS($columns$, $features$)
 $|learned| = 0$
 $quality = 1.0$
 loop$\forall c \in columns$
  while $Quality_{model} > quality \wedge |learned| < |models|$ **do**
   $model \leftarrow Regression_{LASSO}$(features, c)
   $quality \leftarrow r^2$ of $model$
   $features' \leftarrow$ remove best predictor of $model$
   $|learned|++$
  end while
 end loop
end procedure
procedure LEARN($columns$)
 $features$ = CREATEFEATURES($columns$)
 return LEARNMODELS($columns$, $features$)
end procedure

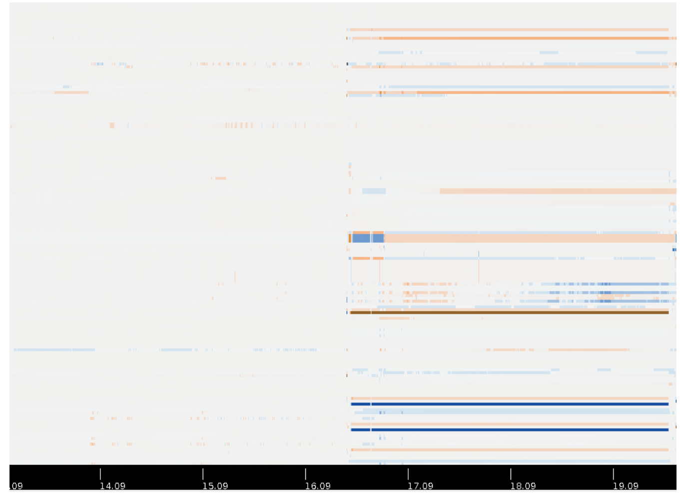

Fig. 4. Divergences between individual LASSO model predictions and real measurements for one day. Each column represents a 10min interval, each row shows the differences between a model prediction and a sensor value, blue meaning too low, orange too high. The plot shows that there are multiple time synchronized divergences starting at september, 16th, indicating a change in the interplay of the WECs components in comparison to the reference time interval used for model learning.

C. Root Cause Analysis

Figure 4 shows sample outputs of a selection of regression models learned for one wind turbine. Each column of pixels represents a 10 minutes interval, each row shows the differences between an individual models' prediction and the real sensor value, blue meaning too low, orange too high. Starting at noon in september, 16th there is a behavioural change in the operation of the wind turbine, indicated by multiple models that synchronously show large divergences between prediction and measurement.

To transform the divergences observed to reportable anomalies, first we need to detect the beginning of a new phase of diverging models. Since the quality of the automatically learned models may differ extensively the divergences observed may be intermittent and not be related to real errors in the components. Therefore, the algorithm only signals an ongoing divergence if the absolute value of the residual, the difference between model prediction and real measurement, is greate than a user defined threshold. The threshold we use is a percentage of the spread of values we observed in the learning time interval, so it is a relative measure for each sensor.

In the second step we apply a heuristic which outputs the most probable root cause for each diverging model. We can't simply assume that the sensor that is the target of the current model is misbehaving. More often it is the case that a sensor which is used as a predictor in other models causes them to diverge.

The heuristic comprises mainly two rules: detect physical sensor defects and identify qualitatively changed sensor measurements. If a sensor has a hardware defect, meaning the sensor, its electrical connection or the cable isolation is damaged, we typically observe a combination of strongly shifted values and

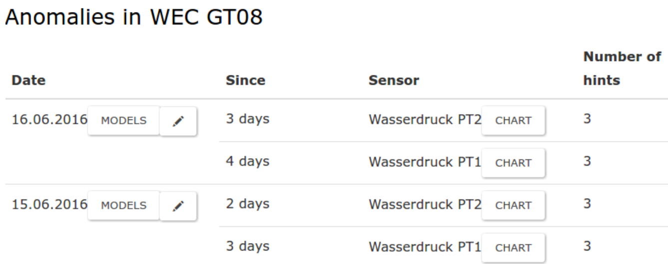

Fig. 5. Screenshot of the anomalies presentation view in the monitoring application.

a reduced variance. We understand that this effect is the result of the translation of analogue measurements to digital values which results in the reporting of physically implausible values because the currents and voltages the sensor observes are either too low or too high in comparison to normal operation. We detect these sensor defects by comparing the variance of the measurements of each sensor with the variance observed in the learning time interval of the model as well as comparing the value range. If a sensor is stuck, the ratio of these values tends to become abnormally small.

If the first rules is not sufficient to explain the divergence of a single model the second rule is used. It acts on the observation, that, if a component changes its behaviour its sensor measurements it will have a high correlation with the residual of each model this sensor gets used in. Normally, the differences of a models predictions and the real measurements are unexplained random noise without any discernible structure. If a sensor has any visual changes in its behaviour these changes will be visible in the "shape" of the residuals. The similarity of the sensor's values and the residual increases, which can be observed as an increase in the correlation between these two. If the correlation of a predictor with the residual of a model is high it is flagged as a probable explanation of a diverging model.

Both rules get applied to each diverging model. For each sensor that is flagged as possibly changed we count the number of distinct hints, where each model that outputs this sensor as the most probable cause is one hint. If a sensor has more hints than the user requires, it is flagged as anomalous and will be reported to the user.

D. Result Presentation

The results are presented to the operator in a form similar to figure 5. For every flagged anomaly the operator may directly see the interactive measurements visualization tool to inspect and compare sensor data. Also, the application shows a button that presents more detailed outputs of the evaluation heuristics of section III-C. This view is useful to either see details about divergent models that have too few hints to be flagged as an anomaly or to manually retrace the decisions made by the root cause analysis heuristics.

IV. Results

Of the 313 sensor per WEC, 250 were not constant when the operating mode was "production". Constant sensors were related to the braking system and other components unused during normal operation, so they were excluded for learning the behaviour of WECs during production.

Given access to a history of four calendar months we used the algorithm to learn approximately 750 models per WEC, an average of three different models per sensor. The average learning time per WEC was approximately one minute. Many of the relationships between sensor automatically discovered could be manually validated by the operator. Models that were clearly wrong were discarded and the misleading sensors excluded from the model learning process (refer to figure 2 for details). Reasons for wrong models were for example counters. They were tended to be used to explain trends, because the were monotonously increasing during the learning data time interval. Afterwards, when using these models to predict sensor values some of those counters were reset by the WEC's controller, which lead to the detection a large number of misleading anomalies. We will test the usefulness of counters as soon as a longer history of data is available.

The models learned were subsequently used to inform the operator about changes in the WEC. Since we don't have a fully annotated dataset that lists every change that should have been detected, we can't give a detailed validation regarding the recall of our method.

The precision of the results depends on the severity of the detected change as indicated by the subjective evaluation of the operator. The majority of anomalies could be explained by sensor hardware defects. In fact, about 1-2% of all sensor per WEC were defect at any given time. Some of the more interesting anomalies detected lead to the discovery of leaks in the coolant system, increasing temperature trends in rotor bearings, erratic pressure measurements in the brake system and misalignments of the nacelle relative to the wind. Many of these detected anomalies were not previously known or reported by the SCADA system.

All anomalies can be explained in terms of differences in the operation of the WEC in comparison between the time interval used for model learning and the most recent past. However, not every change leads to an insight that suggests hardware failures. Instead, they could be explained by having used a non-representative learning time interval. Chapter V-A gives more detailed explanations.

Overall, the automatic anomaly detection lead to a number of actionable insights over the course of spring 2016.

V. Conclusions

The automatic anomaly detection algorithm proved useful for detecting a number of previously unknown defects and performance degradation in the Global Tech I offshore wind park. Even though the core of the method is simple and the approximations found will not be the best possible, the results that were produced proved invaluable for improving the efficiency of the predictive maintenance efforts of the GT1 offshore wind park.

A. Lessons Learned

“Error free” reference date ranges are hard to find. The park officially opened in September 2015. Since then, parameters get changed on a near daily basis. Each new setting may also change the relationship between different sensor. We do not have a long data history, yet.

Not every model makes sense. Some sensors output by the SCADA system represent parameter settings rather than measurements. Trying to predict settings from measurements lead to low quality models. We explicitely removed these sensors from the model learning to increase the models quality.

The more models, the better. Some changes in the behaviour of the WEC will affect several components in a similar way. If we only learned the relationship between these components we might not get an anomaly if there is a change related to it. If we learn more than one model per sensor, we have a higher chance of learning different interdependencies, which in turn means a higher chance of anomalies.

Performant interactive data visualizations are essential. The automatic detection of anomalies is just the first, but an essential, step to identify problems in a WEC. Our system does not know about the technical processes and physical interdependencies of the turbines. This means, it can’t classify the changes as erroneous or normal. This decision is up to the data analyst. In our experience, low latency during interactive investigations increases the chance that the data analyst can find an interpretation of anomalies detected quickly.

Redundant components need to be added up. In our WECs there are several redundant components like motors and pumps. They are operated in an alternating fashion. For the purpose of our method, they really represent one functional component of the WEC. Since our method compares sensor time series for each time step, we had to add up the sensor values of these redundant components (for example we replaced the “motor current oil pump A” and “motor current oil pump B” with their sum).

Not all anomalies necessarily indicate a defect. A number of anomalies detected were the result of having learned from unrepresentative data due to the short history of the offshore park. For example, the power usage of components in the nacelle increased during cold weather periods, because heating was activated. This heating was not active during our reference time interval, so apparently the relationship between the currents and other sensor were flagged as anomalous. Other examples of anomalies that were not interesting were changes in the distribution of voltages and temperatures, that could be explained by parameter changes which lead to changed operation modes.

Sensors are unreliable. Sensors develop malfunctions the same way as any other hardware component of a WEC. We saw an error probability of up to 2% per sensor at any given time. The problem is, that the detection of errors relies on reliable sensors in the first place. A high quality detection algorithm for sensor defects proved essential for the viability of any automatic anomaly detection procedure.

B. Acknowledgements

This work was supported by the German Federal Ministry of Education and Research within the project Competence Center for Scalable Data Services and Solutions (ScaDS) Dresden/Leipzig (BMBF 01IS14014B), the CVtec project (01IS14016C) as well as by the Global Tech I GmbH.

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*. American Society of Mechanical Engineers, 2011, pp. 2071–2079.

[2] A. P. Verma, “Performance monitoring of wind turbines: a data-mining approach,” 2012.

[3] X. You and W. Zhang, “Fault diagnosis of frequency converter in wind power system based on som neural network,” *Procedia Engineering*, vol. 29, pp. 3132–3136, 2012.

[4] N. Laouti, N. Sheibat-Othman, and S. Othman, “Support vector machines for fault detection in wind turbines,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 7067–7072, 2011.

[5] P. Sun, J. Li, C. Wang, and X. Lei, “A generalized model for wind turbine anomaly identification based on scada data,” *Applied Energy*, vol. 168, pp. 550–567, 2016.

[6] Z.-Y. Zhang and K.-S. Wang, “Wind turbine fault detection based on scada data analysis using ann,” *Advances in Manufacturing*, vol. 2, no. 1, pp. 70–78, 2014.

[7] M. Schlechtingen, I. F. Santos, and S. Achiche, “Wind turbine condition monitoring based on scada data using normal behavior models. part 1: System description,” *Applied Soft Computing*, vol. 13, no. 1, pp. 259–270, 2013.

[8] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[9] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani *et al.*, “Pathwise coordinate optimization,” *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[10] E. Frandi, R. Ñanculef, S. Lodi, C. Sartori, and J. A. Suykens, “Fast and scalable lasso via stochastic frank-wolfe methods with a convergence guarantee,” *arXiv preprint arXiv:1510.07169*, 2015.