Diagnosing Wind Turbine Faults Using Machine Learning Techniques Applied to Operational Data

Kevin Leahy*, R. Lily Hu†, Ioannis C. Konstantakopoulos‡, Costas J. Spanos§ and Alice M. Agogino¶
*†¶Department of Mechanical Engineering
University of California, Berkeley
‡§Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
*Department of Civil and Environmental Engineering
University College Cork, Ireland
{*leahyk, †lhu, ‡ioanniskon, §spanos, ¶agogino}@berkeley.edu

***Abstract*—Unscheduled or reactive maintenance on wind turbines due to component failures incurs significant downtime and, in turn, loss of revenue. To this end, it is important to be able to perform maintenance before it's needed. By continuously monitoring turbine health, it is possible to detect incipient faults and schedule maintenance as needed, negating the need for unnecessary periodic checks. To date, a strong effort has been applied to developing Condition Monitoring Systems (CMSs) which rely on retrofitting expensive vibration or oil analysis sensors to the turbine. Instead, by performing complex analysis of existing data from the turbine's Supervisory Control and Data Acquisition (SCADA) system, valuable insights into turbine performance can be obtained at a much lower cost.**

In this paper, data is obtained from the SCADA system of a turbine in the South-East of Ireland. Fault and alarm data is filtered and analysed in conjunction with the power curve to identify periods of nominal and fault operation. Classification techniques are then applied to recognise fault and fault-free operation by taking into account other SCADA data such as temperature, pitch and rotor data. This is then extended to allow prediction and diagnosis in advance of specific faults. Results are provided which show success in predicting some types of faults.

***Index Terms*—SCADA Data, Wind Turbine, Fault Detection, SVM, FDD**

I. Introduction

In order to reach the binding target of sourcing 16% of its annual energy use from renewables by 2020 under the EU Renewable Energy Directive, 40% of Ireland's electricity needs to come from renewable sources. Ireland has one of the best wind resources in Europe, as well as a mature wind energy industry. For this reason, it is expected that the vast majority of this target will be met by wind energy [1].

Wind turbines see highly irregular loads due to varied and turbulent wind conditions, and so components can undergo high stress throughout their lifetime compared with other rotating machines [3]. Because of this, operations and maintenance account for up to 30% of the cost of generation of wind power [4]. The ability to remotely monitor component health is even more important in the wind industry than in others; wind turbines are often deployed to operate autonomously in remote sites so periodic visual inspections can be impractical. Unexpected failures on a wind turbine can be very expensive - corrective maintenance can take up a significant portion of a turbine's annual maintenance budget. Scheduled preventative maintenance, whereby inspections and maintenance are carried out on a periodic basis, can help prevent this. However, this can still incur some unnecessary costs - the components' lifetimes may not be fully exhausted at time of replacement/repair, and the costs associated with more frequent downtime for inspection can run quite high. Condition-based maintenance (CBM) is a strategy whereby the condition of the equipment is actively monitored to detect impending or incipient faults, allowing an effective maintenance decision to be made as needed. This strategy can save up to 20-25% of maintenance costs vs. scheduled maintenance of wind turbines [5]. CBM can also allow prognostic analysis, whereby the remaining useful life (RUL) of a component is estimated. This can allow even more granular planning for maintenance actions.

Condition monitoring systems (CMSs) on wind turbines typically consist of vibration-sensors, sometimes in combination with optical strain gauges or oil particle counters, which are retrofitted to turbine sub-assemblies for highly localised monitoring. This data is sent to a central data processing platform where it is analysed using proprietary software and, if an incipient fault is detected, an alarm is raised [6]. However, CBM and prognostic technologies have not been taken up extensively by the wind industry, despite their supposed benefits [5]. A number of reasons exist for this [7], [8]. The capital cost of retrofitting sensors, as well as data collection and analysis can be quite high - upwards of €13,000 per turbine. Although CMSs have been widely successful in other applications, commercial wind turbine CMSs have not performed as well as hoped due to inherent uncertainties and inaccuracies with some CM techniques. This has led to false alarms, which can be very costly due to the downtime and manual inspections needed. More worryingly, in some cases they have not demonstrated satisfactory performance in detecting incipient faults. This can lead to catastrophic failure of components and related assemblies if maintenance is not

978-1-5090-0382-2/16/$31.00

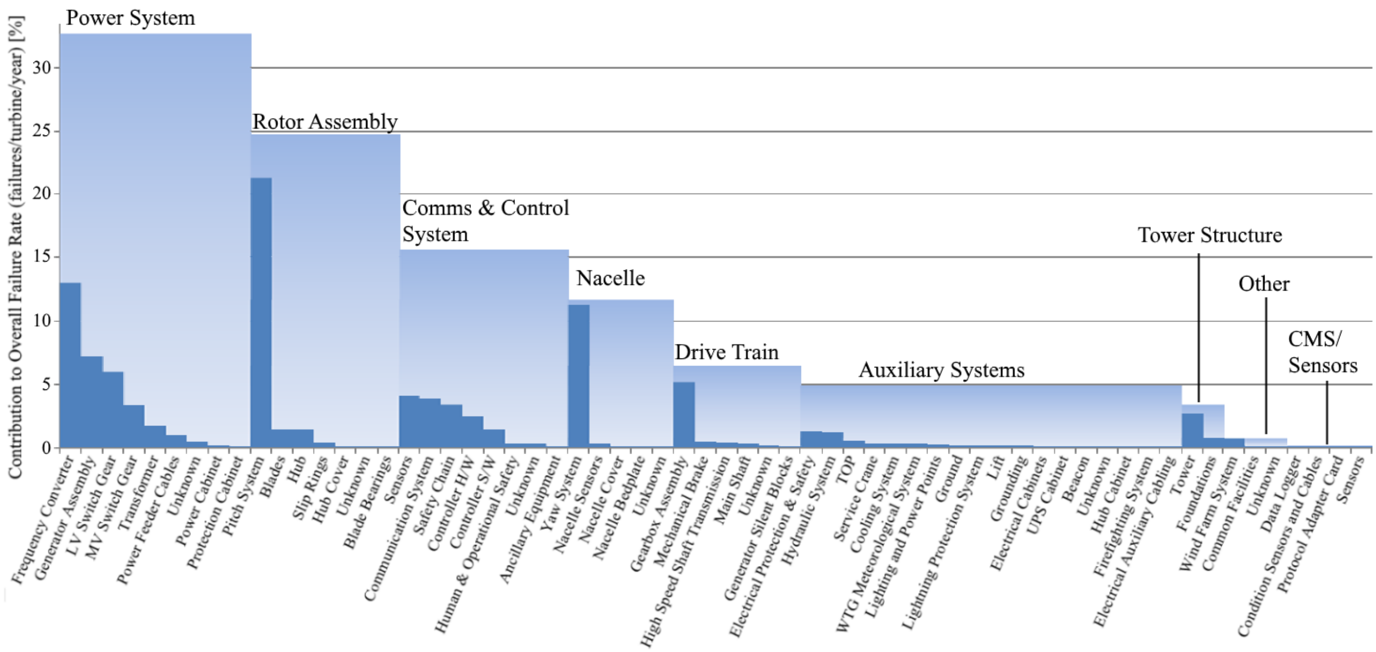

Fig. 1. Normalised failure rates of sub-systems and assemblies for all turbines surveyed [2]

performed in time.

Whereas the aim of wind turbine CMSs is to provide detailed prognostics on turbine sub-assemblies through fitting additional sensors, there already exist a number of sensors on the turbine related to the Supervisory Control and Data Acquisition (SCADA) system. In recent years, there has been a concerted effort to apply CM techniques to wind turbines by analysing data collected by the SCADA system. SCADA data is typically recorded at 10-minute intervals to reduce transmitted data bandwidth and storage, and includes a plethora of measurements such as active and reactive power, generator current and voltages, anemometer measured wind speed, generator shaft speed, generator, gearbox and nacelle temperatures, and others [3]. By performing statistical analyses on various trends within this data, it is possible to detect when the turbine is entering a time of sub-optimal performance or if a fault is developing. This is all done without the added costs of retrofitting additional sensors to the turbine [7]. There have been many different approaches to using SCADA for turbine fault detection and prediction, which we review in the following section.

In this paper, we use data from a coastal site in the South of Ireland where a 3 MW turbine has been installed at a large biomedical device manufacturing facility to offset energy costs. In Section II, we describe how a wind turbine's power curve and other SCADA data can be used for fault detection through performance monitoring, and give a brief review of methods used in the past. In Section III, we describe the turbine site and the data we use. In Section IV we describe the model used for detecting, diagnosing and predicting faults, and the results obtained. Finally, in Section V we evaluate the performance of our model against previous methods used in the literature, both in terms of accuracy and effectiveness at predicting faults.

II. Review of SCADA based CM systems

A. Wind Turbine Failure Modes

Fig. 1 shows the results of a Failure Mode Effects Analysis (FMEA) for wind turbines, after an extensive and detailed survey of the frequency of different failure modes on turbine components and sub-assemblies and their contribution to down time. As can be seen, the biggest contribution to the overall failure rate was the power system. This translated to just below a 40% contribution to overall downtime on the turbines surveyed. This data comes from a study by the EU FP7 ReliaWind project, undertaken by a consortium of stakeholders from the wind industry, technology experts and academia [2].

B. Power Curve

The relationship between power and wind speed for a specific turbine can be seen in Fig. 2 (a). This graph is known as a power curve, and shows the turbine's power output as a function of hub height wind speed. A turbine's power curve is an important metric when determining wind turbine performance. Different turbine models will have different power curves according to the operating conditions they have been designed for — typically, a certain range of wind speeds. The performance of a turbine under different wind speeds can be related to three key points on this graph [9]. u_c, the cut-in speed, is the minimum useful wind speed at which the turbine begins to generate power. u_r, the rated speed, is the speed at which maximum rated generator output is obtained. u_s, the cut-out speed, is the maximum speed at which the turbine can produce power. This is limited by engineering and safety constraints, however some turbine models allow limited power output above this through smart control of the blade pitch angle. The power curve for a particular turbine model is usually given by the manufacturer as a guaranteed performance metric [10].

Comparing the generated power of a turbine at a given wind speed to the supplied power curve is an important way of checking if a turbine is performing correctly. However, the manufacturer typically develops this power curve according to standard guidelines, e.g., IEC 61400-12-1 [11]. They are also developed under standard conditions, using a specific methodology that is impractical to reproduce at an operating wind farm [12]. In practice, turbines are often placed on sites with varying topography and wind conditions, so any deviation from the manufacturer's power curve could be due to a number of environmental variables as opposed to indicating a problem in the turbine itself [13], [9]. By using data obtained from when a particular turbine was in fault-free optimal operation, a new power curve can be modelled and used as a visual reference for monitoring future performance. Because the topography and wind conditions at a specific site will remain largely the same over a given period of time, any changes in the characteristic shape of the power curve can be put down to changes in the turbine itself, and visually diagnosed by an expert as the cause of a specific incipient fault. An example of this is seen in Fig. 2 (b). The characteristic shape of this power curve can be attributed by an expert to curtailed power output due to faulty controller values [14].

C. Review of SCADA-based CM systems

Many attempts have been made in the past to automate the diagnostic process through use of statistical and artificial intelligence methods. A number of approaches model the power curve under normal operating conditions. This is then compared to the on-line values and a cumulative residual is developed over time. As the residual exceeds a certain threshold, it is indicative of a problem on the turbine. Gill et. al model the power curve under normal conditions using copula statistics, and demonstrate that a deviation from this could possibly be used in future to give an indication of a developing fault [10]. Skrimpas et. al use kernel methods to model the power curve and deviations from this correspond to periods of poor turbine performance due to controller errors, icing, power de-rating and operation in noise reduction mode [15]. However, the method does not differentiate between these faults. Butler et. al [16] performed a similar analysis, using Gaussian Process models to model the power curve. They were able to successfully show a performance degradation which began three months in advance of a main bearing failure on the turbine. Again, however, this method did not provide diagnostic capabilities. In [17], a novel algorithm is developed for modelling the power curve. The average power output at various different wind speed "bins" is found. A provisional power curve is then built by interpolating between these points. Next, optimal bounds are developed by shifting this curve up and down by varying degrees. All points outside these bounds are filtered out, and the process repeats itself until a satisfactory model representing nominal operation is found. Smart alarm limits were then developed to detect future anomalous behaviour. This method showed an indication of faulty operation, but did not diagnose a specific fault.

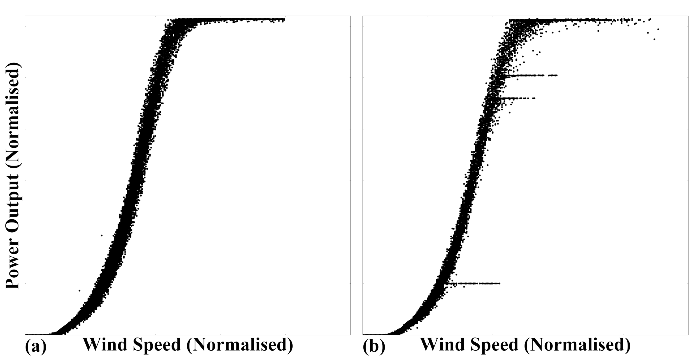

Fig. 2. Wind turbine power curve under (a) optimal operation and (b) faulty controller values

An expansion on the above methods is to use performance indicators other than the power curve. Lapira et. al trained a neural network which included additional parameters such as nacelle temperature, rotor speed, gearbox oil temperature and generator bearing temperature. Their model successfully demonstrated performance degradation leading up to a fault [14]. Another study performed performance monitoring using wind speed trended against power output, rotor speed and blade pitch angle. This gave a good performance metric for the turbine but fault diagnosis was not part of this study's scope [18].

By using a much wider spectrum of SCADA parameters, fault classification and limited fault prediction has been successfully demonstrated by Kusiak et. al [19]. A number of models, comprising of neural networks, boosting trees, support vector machines and standard classification regression trees, were built to evaluate their performance in predicting and diagnosing faults. It was found that prediction of a specific fault, a diverter malfunction, was possible at 67% accuracy and 73% specificity 30 minutes in advance. Unfortunately, when this was extended out to one hour in advance, accuracy and specificity fell to 40% and 24%, respectively. Overall, it was found that the most successful algorithm for specific fault detection was the boosting tree algorithm. The prediction of specific blade pitch faults was demonstrated in [20], using genetic programmed decision trees. Here, the maximum prediction time was 10 minutes, at a 68% accuracy and 71% specificity. The SCADA data was also at a resolution of 1s rather than the more common 10 minutes.

The development of indicators for specific major component failures has seen more success in the literature. One effort by Butler et al. included the use of main shaft RPM, hydraulic brake temperature and pressure, and blade pitch position to build a model using Sparse Bayesian Learning. This was able to show a strong indicator of main bearing failure up to 30 days in advance [21]. Another effort used neural networks and normal behaviour models to model the generator performance. This showed that abnormalities in the residual signal were visually noticeable up to one year in advance of a full gearbox failure [3]. However, neither of these studies were able to verify their findings on a test set due to the inherent lack of data when dealing with full component failure.

It is clear from the literature that some indication of complete failure of a main component can be detected months in advance solely using SCADA data. However, for less serious, but more frequent faults which also contribute to degraded turbine performance, such as power feeding, blade pitch or diverter faults, prediction more than a half hour in advance is currently very poor. In this paper we attempt to widen the prediction capability for these types of faults. Although the most successful attempt to do this ([19], discussed above) found a boosting tree algorithm to be more successful than other methods, including Support Vector Machines (SVMs), the authors did not go into detail on the specifics of the models used. However, SVMs are a widely used and successful tool for solving classification problems. The basic premise behind the SVM is that a decision boundary is made between two opposing classes, based on labelled training data. A certain number of points are allowed to be misclassified to avoid the problem of overfitting. They are very well suited to this specific problem, where the relationship between a high number of parameters (e.g., the many different parameters collected by a SCADA system) can be complex and non-linear [22], [23]. They have been used in other industries for condition monitoring and fault diagnosis with great success. A review by [24] showed that SVMs have been successfully used to diagnose and predict mechanical faults in HVAC machines, pumps, bearings, induction motors and other machinery. CM using SVMs has also found success in the refrigeration, semi-conductor production chemical and process industries [25].

III. Data

A. Description of Data

The data in this study comes from a 3 MW direct-drive turbine which supplies power to a major biomedical devices manufacturing plant located near the coast in the South of Ireland. There are three separate datasets taken from the turbine SCADA system; "operational" data, "status" data and "warning" data. The data covers an 11 month period from May 2014 - April 2015.

1) Operational Data: The turbine control system monitors many instantaneous parameters such as wind speed and ambient temperature, power characteristics such as real and reactive power and various currents and voltages in the electrical equipment, as well as temperatures of components such as the generator bearing and rotor. The average, min. and max. of these values over a 10 minute period is then stored in the SCADA system with a corresponding timestamp. This is the "operational" data. A sample of this data is shown in Table I. This data was used to train the classifiers, and was labelled according to various filters, as explained in Section III-B. The initial operational data contained roughly 45,000 datapoints, representing the 11 months analysed in this study.

TABLE I
10 Minute Operational Data

TimeStamp	**Wind Speed (avg.)** m/s	**Wind Speed (max.)** m/s	**Wind Speed (min.)** m/s	**Power (avg.)** kW	**Power (max.)** kW	**Power (min.)** kW	**Bearing Temp (avg.)** °C
09/06/2014 14:10:00	5.8	7.4	4.1	367	541	285	25
09/06/2014 14:20:00	5.7	7.1	4.1	378	490	246	25
09/06/2014 14:30:00	5.6	6.5	4.5	384	447	254	25
09/06/2014 14:40:00	5.8	7.5	3.9	426	530	318	25
09/06/2014 14:50:00	5.4	6.9	4.5	369	592	242	25

2) Status Data: There are a number of normal operating states for the turbine. For example, when the turbine is producing power normally, when the wind speed is below u_c, or when the turbine is in "storm" mode, i.e., when the wind speeds are above u_s. There are also a large number of statuses for when the turbine is in abnormal or faulty operation. These are all tracked by status messages, contained within the "Status" data. This is split into two different sets: (i) WEC status data, and (ii) RTU status data. The WEC (**W**ind **E**nergy **C**onverter) status data corresponds to status messages directly related to the turbine itself, whereas RTU data corresponds to power control data at the point of connection to the grid, i.e., active and reactive power set points. Each time the WEC or RTU status changes, a new timestamped status message is generated. Thus, the turbine is assumed to be operating in that state until the next status message is generated. Each turbine status has a "main status" and "sub-status" code associated with it. See Table II for a sample of the WEC status message data. Any main WEC status code above zero indicates abnormal behaviour, however many of these are not associated with a fault, e.g., status code 2 - "lack of wind". The RTU status data almost exclusively deals with active or reactive power set-points. For example, status 100 : 82 corresponds to limiting the active power output to 82% of its actual current output.

3) Warning Data: The "Warning" data on the turbine mostly corresponds to general information about the turbine, and usually isn't directly related to turbine operation or safety. These "warning" messages, also called "information messages" in some of the turbine documentation, are timestamped in the same way as the status messages. Sometimes, warning messages correspond to a potentially developing fault on the turbine; if the warning persists for a set amount of time and is not cleared by the turbine operator or control system, a fault is raised and a new status message is generated. For this reason, it was decided that warning messages can be mostly ignored in this analysis, as the information is captured by the status messages. A single exception to this is mentioned in Section III-B1.

B. Data Labelling

In order to properly train a classifier, it is important that the data is correctly labelled. In this paper, we attempt three levels

TABLE II
WEC STATUS DATA

Timestamp	Main Status	Sub Status	Status Text
13/07/2014 13:06:23	0	0	Turbine in Operation
14/07/2014 18:12:02	62	3	Feeding Fault: Zero Crossing Several Inverters
14/07/2014 18:12:19	80	21	Excitation Error: Overvoltage DC-link
14/07/2014 18:22:07	0	1	Turbine Starting
14/07/2014 18:23:38	0	0	Turbine in Operation
16/07/2014 04:06:47	2	1	Lack of Wind: Wind Speed too Low

of classification: fault/no-fault, where samples are classified as faulty or fault free; fault diagnosis, where samples are classified as a specific fault, or fault-free; and fault prediction, where we attempt to classify data as a specific fault up to one hour in advance of the fault occurring. This is achieved by splitting the initial operational data into labelled "no-fault", "all faults", "specific fault" and "fault prediction" datasets. The process for labelling data is explained in this section.

1) No-Fault Dataset: For all three levels of classification, a common "no-fault" data set is needed consisting of nominal fault-free operation. To create this, three different filters were applied to the full set of 10-minute operational data. First, WEC status codes corresponding to nominal operation were selected. These are "*0 : 0 - Turbine in Operation*", "*2 : 1 - Wind Speed too Low*" and "*3 : 12 - Storm Wind Speed*". Operational data with timestamps that corresponded to at least 30 minutes *after* these statuses came into effect and 120 minutes *before* they changed were chosen. These time-bands were found empirically and eliminate any transients that may arise from going from fault-free to faulty operation or vice-versa.

Next, all operational data corresponding to RTU statuses where power output was being curtailed were filtered out. This left only one status, "*0 : 0 - RTU in operation*". For this, a time band of only 10 minutes was chosen, as any transients in the electrical control system would not last very long.

Finally, times corresponding to a single specific warning message (main warning code "*230 - Power Limitation (10h)*") were filtered out. This warning corresponds to slightly limited power output during nominal operation for one of a number of reasons, including turbine noise control during certain hours, an increase in internal temperatures on a hot day, or grid regulation. When this message is generated, there follows a 10-hour period where turbine power output may or may not be curtailed. Although considered a part of normal operation, for the purposes of this study it was decided to filter this out to give a clearer distinction for fault classification. It may be included in future work. After all three stages of filtering, the no-fault data contained roughly 28,000 points of 10 minute operational data.

To verify that only data from when the turbine was in nominal operation was included in the "no-fault" dataset, the power curve of the filtered data was plotted to check that it conformed to the nominal shape as seen in Fig. 2 (a). An algorithm for filtering out power curve anomalies, as developed in [17], was used to highlight the data points which were outside the estimated bounds of nominal operation. Fig. 3 (a) shows the power curve generated from operational data before any filtering took place. The points in red are those which the algorithm marked as anomalous. In this case there were 4,000 such points. Fig. 3 (b) shows the filtered no-fault dataset. After the three step filtering process, there were still roughly 400 points, highlighted in red, which the algorithm identified as abnormal. Because these make up less than 1% of the overall no-fault dataset and because, in practice, turbine data will always contain noise, it was decided to include these in the fault-free data initially.

2) All Faults Dataset: In order to classify fault/no-fault operation, it was also necessary to develop a set of labelled fault data. For this, a list of frequently occurring faults was made. For these faults, status messages with codes corresponding to the faults were selected. Next, a time band of 600s before the start, and after the end, of these turbine states was used to match up the associated 10-minute operational data. The 10 minutes timeband was selected so as to definitely capture any 10-minute period where a fault occurred, e.g., if a power feeding fault fault occurred from 11:49-13:52, this would ensure the 11:40-11:50 and 13:50-14:00 operational data points were labelled as faults. The faults included are summarised in Table III. Note that the fault frequency refers to specific instances of each fault, rather than the number of data points of operational data associated with it, e.g., a generator heating fault which lasted one hour would contain 6 operational data points, but would still count as one fault instance. *Feeding faults* refer to faults in the power feeder cables of the turbine, *excitation errors* refer to problems with the generator excitation system, *mains failure* refers to problems with mains electricity supply *to* the turbine, *malfunction aircooling* refers to problems in the air circulation and internal temperature circulation in the turbine, and *generator heating faults* refer to the generator overheating.

3) Specific Fault Datasets: For specific faults, the same methodology for all faults was used, but this time single status

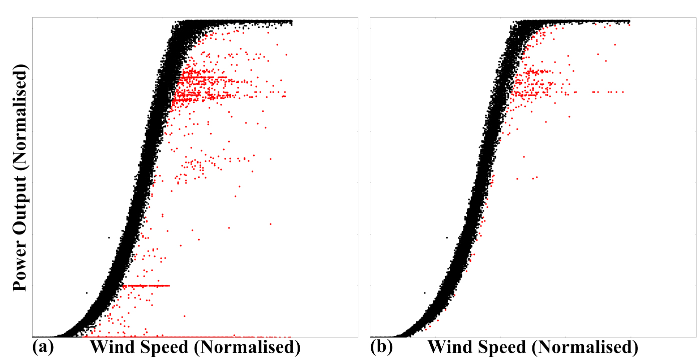

Fig. 3. Power curves for (a) Initial set of operational data (b) Fault-free operational data. Abnormal-looking values are shown to the right of the curve.

TABLE III
FREQUENTLY OCCURING FAULTS, LISTED BY CORRESPONDING STATUS CODE

Fault	Main Status Code	Fault Incidence Frequency	Operational Data Points
Feeding Fault	62	92	251
Excitation Error	80	84	168
Malfunction Aircooling	228	20	62
Mains Failure	60	11	20
Generator Heating Fault	9	6	43

codes for each fault code in Table III were used. Again a time band of 600s before the start and after the end of each fault status was used to match up corresponding 10-minute operational data.

4) Fault Prediction Datasets: In order to try and predict specific faults, the time band around which faults were classified was extended by varying degrees. These were: 10, 20, 30, 60, 120 and 360 minutes before a specific fault. This meant the operational data points leading up to a specific fault were also included in that fault class.

IV. METHODOLOGY

For all three levels of classification, an SVM was trained using scikit-Learn's LibSVM implementation [26], [27]. Each dataset was randomly shuffled and split into training and testing sets, with 80% being used for training and the remaining 20% reserved for testing. Because the original operational dataset had 60+ features, only a subset of 30 specific features were chosen to be included for training purposes. It was found that a number of the original features corresponded to sensors on the turbine which were broken, e.g., they had frozen or blatantly incorrect values. The most relevant of the remaining features were then selected for inclusion based on the authors' domain knowledge. A subset of *these* features, corresponding to 12 temperature sensors on the inverter cabinets in the turbine, all had very similar readings. Because of this, it was decided to instead consolidate these and use the average and standard deviation of the 12 inverter temperatures. This resulted in 29 features being used to train the SVMs, which were all scaled individually to unit norm. This was because some features, e.g., power output had massive ranges from 0 to thousands, whereas others, e.g., temperature, ranged from 0 to only a few tens.

A randomized grid search was then performed over a number of hyperparameters used to train each SVM to find the ones which yielded the best results. These were then verified using 10-fold cross validation. The scoring metric used for cross validation was a mean of the weighted precision and recall (see the end of this section for an explanation of these terms). The hyperparameters searched over were C, which controls the number of samples allowed to be misclassified, γ which defines how much influence an individual training example has, and the kernel used. The three kernels which were tried were the simple linear kernel, the radial-basis (Gaussian) kernel and the polynomial kernel. The data was heavily imbalanced - there were on the order of 10^2 more no-fault samples than fault-class samples. Two different approaches were tried to mitigate this effect. In the first approach, a class weight, $c.w.$ is added to the minority class when calculating C for that class. In this way, the new value for C for the fault class, C_w, can be seen in Eq. 1. A number of different class weights were added to the set of hyperparameters being searched over for this approach, which can be seen in Table IV.

TABLE IV
HYPERPARAMETERS USED IN RANDOMIZED GRID SEARCH

Hyperparameter	Values
C	0.01, 0.1, 1, 10, 100, 1000
$c.w.$	1, 2, 10, 50, 100
Kernel	Linear, Polynomial, R.B.F.
γ	1E-3, 1E-4, 1E-5

$$C_w = C * c.w. \quad (1)$$

The second approach instead selected a balanced set of data to train on; after the full set of data was split into training and test parts, the training set was further split to include the same number of fault-free instances as fault instances. The test data was not altered in any way so as to preserve the imbalance seen in the real world.

A number of scoring metrics were used to evaluate final performance on the test set. These were specificity, precision, recall and F1-Score, the harmonic mean of precision and recall. The overall accuracy of the classifier on the test set was not used as a metric due to the massive imbalance in the data sets. For example, if 4990 samples were correctly labelled as fault-free, and the only 20 fault samples were also incorrectly labelled as such, the overall accuracy of the classifier would still stand at 99.6%. The formulae for calculating specificity, precision, recall and the F1-score can be seen below:

$$Recall = tp/(tp + fn) \quad (2)$$

$$Precision = tp/(tp + fp) \quad (3)$$

$$F1 = 2tp/(2tp + fp + fn) \quad (4)$$

$$Specificity = tn/fp + tn \quad (5)$$

where tp is the number of true positives, i.e., correctly predicted fault samples, fp is false positives, fn is false negatives, i.e., fault samples incorrectly labelled as no-fault, and tn is true negatives.

V. RESULTS & DISCUSSION

The results of the hyperparameter search for the case of a balanced training set, with $c.w.$ set to 1 can be seen in Table V. The results of performing this search on the full imbalanced training set, including different values of $c.w.$, can be seen in Table VI. Training time for the case of the smaller balanced

TABLE V
HYPERPARAMETERS FOUND SEARCHING OVER FULL TRAINING SET

Test Set	C	**kernel**	γ	$c.w.$
Fault/No-Fault	1000	linear	.001	2
Feeding Faults	1000	linear	.0001	2
Excitation Faults	1000	linear	.0001	2
Generator Heating Faults	1000	linear	.0001	10

TABLE VI
HYPERPARAMETERS FOUND SEARCHING OVER BALANCED TRAINING SET ($c.w.$ SET TO 1)

Test Set	C	**kernel**	γ
Fault/No-Fault	1000	linear	.0001
Feeding Faults	100	linear	.001
Aircooling Faults	.01	linear	.0001
Excitation Faults	1000	linear	.0001
Generator Heating Faults	1000	linear	.0001
Mains Failure Faults	100	linear	.0001

TABLE VII
RESULTS FOUND USING BALNCED TRAINING SET ($c.w.$ SET TO 1)

Dataset	**Precision**	**Recall**	**F1**	**Specificity**
Fault/No-Fault	.08	.9	.15	.83
Feeding Faults	.05	.87	.09	.85
Aircooling Faults	.12	.27	.17	.99
Excitation Faults	.04	1	.08	.85
Generator Heating Faults	.56	1	.71	.99
Mains Failure Faults	.01	1	.01	.9

TABLE VIII
RESULTS FOUND USING FULL TRAINING SET

Dataset	**Precision**	**Recall**	**F1**	**Specificity**
Fault/No-Fault	1	.48	.65	1
Feeding Faults	.97	.58	.72	.99
Excitation Faults	1	.33	.5	1
Generator Heating Faults	1	1	1	1

TABLE IX
RESULTS FOUND ONE HOUR IN ADVANCE FOR THE CASES OF BALANCED AND IMBALANCED TRAINING SETS

Dataset	**Prec.**	**Rec.**	**F1**	**Spec.**
Feeding Faults (imbalanced))	.08	.97	.15	.79
Feeding Faults (balanced)	1	.28	.44	1
Excitation Faults (imbalanced)	.06	1	.11	.76
Excitation Faults (balanced)	1	.16	.27	1
Generator Heating Faults (imbalanced)	.1	1	.17	.98
Generator Heating Faults (balanced)	1	1	1	1

training set was less than one minute. For the case of the full, balanced set, training time was around 30 minutes. The PC used had an intel core i5-4300U CPU and 8GB of RAM.

A. *Fault/No-Fault Dataset*

When training the data on a balanced set, the fault/no-fault prediction performance on recall and specificity is quite high (0.9 and 0.83, respectively), as seen in Table VII. The high recall means there are very few missed fault instances. However, the SVM proved to have very poor precision, and, as a result, a low F1 score. When compared with work done in [19], our methodology has achieved a better recall and specificity (compared with 0.84 and 0.66). There was no mention of precision score in other literature, so it is hard to benchmark this. The poor precision represents a high proportion of samples incorrectly labelled as faulty compared to correctly labelled as faulty. This is a common problem with imbalanced data. As previously mentioned, a way to deal with this was to train on the full imbalanced set, and introduce a set of $c.w.$ hyperparameters to the graph search. When this was performed on the fault/no-fault set, precision and specificity were both increased to 1, but recall was reduced to 0.48, as seen in Table VIII. This shows the inherent trade-off in precision and recall.

B. *Specific Fault Datasets*

For the prediction of specific faults trained on the balanced set, performance was similar the basic Fault/No-Fault case, apart from on Aircooling Faults which showed very poor performance all-round. Predicting Generator Heating Faults showed the best promise, with a Precision of 0.56 - well above any of the others, a recall of 1 and a specificity of 0.99. It also yielded an F1 score of 0.71. It is hard to compare the case of specific faults against previous studies in [19], as the test set at this level of detection in that study was flawed; it used a balanced number of samples from each class, which improves test performance, but does not represent the true distribution of data. Nevertheless, even compared to this, our model performed better for predicting a number of specific faults - the best score in that study was a recall of 0.87 and specificity of 0.63. It was decided to train on the full set of data only for three of the five specific faults, as these had a higher frequency than the others. The results of this can be seen in VIII. Here, for the most part, the same trade off in precision and recall can be seen as in the fault/no-fault set. However, extremely good performance was seen in the prediction of Generator Heating Faults. This could be due to the fact that in the test set, there were only seven instances of generator heating faults, although there were no false positives among the roughly 5,000 no-fault instances.

C. *Advanced Prediction of Specific Faults*

When the time band before specific faults was stretched from 10 minutes to one hour, good performance on recall and specificity was still possible. The results of this can be seen in Table IX. Here again the trade-off between precision and recall can be seen when training on the imbalanced and balanced training sets. Once again, however, generator heating faults were predicted with unprecedented accuracy one hour in advance, with a perfect specificity and F1 score. These results represent significant advancement of work done in [19], where the best performance of recall and specificity at one hour in advance of a specific fault were .24 and .34, respectively.

VI. Conclusion

A new methodology for classifying and predicting turbine faults based on SCADA data was investigated. The fault classification operates on three levels: distinguishing between fault/no-fault operation, classifying a specific fault, and the prediction of a specific fault one hour in advance. The results were very promising and show that distinguishing between fault and no fault operation is possible with very good recall and specificity, but the F1 score is brought down by poor precision. A trade-off is possible using a slightly different method of training which yields improved precision but poorer recall. In general, this was also the case for classifying a specific fault, and for predicting specific faults in advance. The less than ideal overall performance could be due to the highly imbalanced nature of fault data, with the no-fault class having an overwhelming majority of samples. However, generator heating faults were classified with a perfect F1 score, and one hour prediction also yielded a perfect score.

The methodology presented in this paper trained multiple binary classifiers on separate test sets for each fault. This is not ideal as there is typically only one unified "test set" containing multiple faults in real world applications. Future work will take this into account so that a practical application of this research could be made possible using multi-class classification. It is planned to apply new techniques for using SVMs when working with the highly imbalanced nature of fault data to improve the precision and recall performance seen in this paper. As well as this, other techniques which perform well at multi-class classification such as boosting trees, logistic regression and ensemble methods will be compared. Furthermore, it is planned to obtain more data with more fault instances to verify the prediction performance obtained in this study. Advanced feature extraction and selection will also be utilised to verify that all features used in this study are relevant, and to check if there are any others which could improve performance. Finally, the precision/recall trade-off will be tuned to minimize overall operational cost by looking at the cost of specific "missed" faults vs. false alarms. This tuning can be achieved through appropriately biasing the classifier.

Acknowledgements

Ioannis C. Konstantakopoulos is a Scholar of the Alexander S. Onassis Public Benefit Foundation.

This work is supported by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a centre for intellectual excellence in research and education in Singapore.

References

[1] S. Billington and P. Mohr, "The Value of Wind Energy to Ireland," 2014.

[2] J. B. Gayo, "ReliaWind Project final report," Tech. Rep., 2011.

[3] A. Zaher *et al.*, "Online wind turbine fault detection through automated SCADA data analysis," *Wind Energy*, vol. 12, no. 6, pp. 574–593, sep 2009.

[4] European Wind Energy Association (EWEA), "The Economics of Wind Energy," Tech. Rep., 2009.

[5] 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, p. 11, 2013.

[6] Z. Hameed *et al.*, "Condition monitoring and fault detection of wind turbines and related algorithms: A review," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 1, pp. 1–39, jan 2009.

[7] W. Yang *et al.*, "Wind turbine condition monitoring: Technical and commercial challenges," *Wind Energy*, vol. 17, no. 5, pp. 673–693, 2014.

[8] W. Yang, R. Court, and J. Jiang, "Wind turbine condition monitoring by the approach of SCADA data analysis," *Renewable Energy*, vol. 53, pp. 365–376, may 2013.

[9] J. Manwell, J. G. McGowan, and a. L. Rogers, *Wind Energy Explained*, 2009.

[10] 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.

[11] International Electrotechnical Commission (IEC), "IEC 61400-12-1:2006: Wind Turbines - Part 12-1: Power Performance Measurements of Electricity Producing Wind Turbines," 2013.

[12] O. Uluyol *et al.*, "Power Curve Analytic for Wind Turbine Performance Monitoring and Prognostics," *Annual Conference of the Prognostics and Health Management Society*, pp. 1–8, 2011.

[13] S. Li *et al.*, "Using neural networks to estimate wind turbine power generation," *IEEE Transactions on Energy Conversion*, vol. 16, no. 3, pp. 276–282, 2001.

[14] E. Lapira *et al.*, "Wind turbine performance assessment using multi-regime modeling approach," *Renewable Energy*, vol. 45, pp. 86–95, 2012.

[15] G. A. Skrimpas *et al.*, "Employment of Kernel Methods on Wind Turbine Power Performance Assessment," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 3, pp. 698–706, 2015.

[16] S. Butler, J. Ringwood, and F. O'Connor, "Exploiting SCADA system data for wind turbine performance monitoring," in *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE, oct 2013, pp. 389–394.

[17] J.-y. Park *et al.*, "Development of a Novel Power Curve Monitoring Method for Wind Turbines and Its Field Tests," *IEEE Transactions on Energy Conversion*, vol. 29, no. 1, pp. 119–128, 2014.

[18] A. Kusiak and A. Verma, "Monitoring wind farms with performance curves," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 1, pp. 192–199, 2013.

[19] A. Kusiak and W. Li, "The prediction and diagnosis of wind turbine faults," *Renewable Energy*, vol. 36, no. 1, pp. 16–23, 2011.

[20] A. Kusiak and A. Verma, "A data-driven approach for monitoring blade pitch faults in wind turbines," *IEEE Transactions on Sustainable Energy*, vol. 2, no. 1, pp. 87–96, 2011.

[21] S. Butler *et al.*, "A feasibility study into prognostics for the main bearing of a wind turbine," *2012 IEEE International Conference on Control Applications*, pp. 1092–1097, oct 2012.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[23] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[24] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560–2574, 2007.

[25] N. Laouti, N. Sheibat-othman, and S. Othman, "Support Vector Machines for Fault Detection in Wind Turbines," *18th IFAC World Congress*, pp. 7067–7072, 2011.

[26] C.-c. Chang and C.-j. Lin, "LIBSVM : A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, pp. 1–39, 2011.

[27] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, jan 2012.