Diagnosing and Predicting Wind Turbine Faults from SCADA Data Using Support Vector Machines

Kevin Leahy[1], R. Lily Hu[2], Ioannis C. Konstantakopoulos[3], Costas J. Spanos[4], Alice M. Agogino[5] and Dominic T. J. O'Sullivan[6]

[1,6] *Department of Civil & Environmental Engineering, University College Cork, Cork, Ireland*
kevin.leahy@umail.ucc.ie

[2] *Salesforce Research (MetaMind), Palo Alto, CA, USA*

[1,2,5] *Department of Mechanical Engineering, University of California, Berkeley, CA, USA*

[3,4] *Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA*

Abstract

Unscheduled or reactive maintenance on wind turbines due to component failure 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. 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, fault and alarm data from a turbine on the Southern coast of Ireland is analysed to identify periods of nominal and faulty operation. Classification techniques are then applied to detect and diagnose faults 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 recall scores generally above 80% for fault detection and diagnosis, and prediction up to 12 hours in advance of specific faults, representing significant improvement over previous techniques.

1. Introduction

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 (Zaher, McArthur, Infield, & Patel, 2009). This significantly contributes to the cost of operations and maintenance (O&M), which can account for up to 30% of the cost of generation of wind power (European Wind Energy Association (EWEA), 2009). 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 or offshore locations where highly regular 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 (Godwin & Matthews, 2013). 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 (Tamilselvan, Wang, Sheng, & Twomey, 2013). However, CBM and prognostic technologies have not been taken up extensively by the wind industry, despite their supposed benefits (Godwin & Matthews, 2013).

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 (Zaher et al., 2009). By performing statistical analyses on 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 (Yang, Tavner, Crabtree, Feng, & Qiu, 2014).

A number of approaches use the turbine's power curve, the relationship between power output and hub-height wind speed for a particular turbine, as a point of reference. The power curve is modelled under normal operating conditions, and any changes in its characteristic shape can be visually diagnosed by an expert as the cause of a specific incipient fault, e.g. curtailed power output due to faulty controller values (Lapira, Brisset, Davari Ardakani, Siegel, & Lee, 2012). Other approaches compare the modelled curve to 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, Stephen, & Galloway, 2012; Skrimpas et al., 2015; Uluyol, Parthasarathy, Foslien, & Kim, 2011). However, these methods simply detect when the turbine is entering abnormal operation and do not diagnose the faults.

An expansion of the above methods is to use performance indicators other than the power curve. Work by Du et. al showed some successes in using anomaly detection to show residuals from normal behaviour in advance of a fault occurring (Du et al., 2016). By using a much wider spectrum of SCADA parameters, fault diagnosis and limited fault prediction has been successfully demonstrated by Kusiak et. al (Kusiak & Li, 2011). A number of models were built using machine learning to evaluate their performance in predicting and diagnosing faults. It was found that prediction of a specific fault, a diverter malfunction, was possible at 68% accuracy, 73% sensitivity (also known as recall - the ratio of true positives to true positives and false positives) and 66% sensitivity (the ratio of true negatives to true negatives and false positives) 30 minutes in advance of the fault occurring. Unfortunately, when this was extended out to two hours in advance, accuracy, recall and sensitivity fell to 49% and 25% and 34%, respectively. It should also be noted that, significantly, the testing set used for specific fault prediction in this paper did not represent the distribution of the underlying labelled SCADA data. Instead, there were just over twice as many "fault-free" instances as there were of a specific fault. This is in contrast to the true distribution where there could be upwards of 1,000 times more fault-free than fault instances, and does not reflect the real-world performance of the models developed. As well as this, the precision (the ratio of true positives to false positives) was not recorded.

The prediction of specific blade pitch faults was demonstrated by Kusiak and Verma, using genetically programmed decision trees (Kusiak & Verma, 2011). Here, the maximum prediction time was 10 minutes, at a 69% accuracy, 71% recall and 67% specificity. The SCADA data was also at a resolution of 1s rather than the more common 10 minutes. Once again, no precision score was given for this task.

It is clear from the literature that for relatively minor but frequent faults which 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. These faults can contribute to failures related to the power system; in a study carried out by the EU FP7 ReliaWind project, it was found that just under 40% of overall turbine downtime can be attributed to power system failures (Gayo, 2011).

In this paper, we widen the prediction capability for these types of faults. We analyse 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, and use it to detect and diagnose faults. In Section 2, we describe the turbine site and the data we use. In Section 3.1, we describe the process for labelling the data and the different types of classification performed. In Section 3 we describe the specific models used for detecting, diagnosing and predicting faults. Finally, in Section 4 we give the results obtained and evaluate the performance of our models, and compare them with previous results from the literature.

2. DESCRIPTION OF DATA AND FAULTS

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 two separate datasets taken from the turbine SCADA system; "operational" data and "status" data. The data covers an 11 month period from May 2014 - April 2015.

2.1. Operational Data

The turbine control system monitors many instantaneous parameters such as wind speed and ambient temperature, power

Table 1. 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	Ambient Temp (avg.) °C	Bearing Temp (avg.) °C
09/06/2014 14:10:00	5.8	7.4	4.1	367	541	285	17.9	25.0
09/06/2014 14:20:00	5.7	7.1	4.1	378	490	246	17.5	24.6
09/06/2014 14:30:00	5.6	6.5	4.5	384	447	254	17.6	25.1
09/06/2014 14:40:00	5.8	7.5	3.9	426	530	318	18.1	23.7
09/06/2014 14:50:00	5.4	6.9	4.5	369	592	242	18.2	24.6

Table 2. WEC Status Data

Timestamp	Main	Sub	Description
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

Table 3. Frequently occurring faults, listed by status code, fault incidence frequency and number of corresponding 10-minute SCADA data points

Fault	Main Status	*f*	No. Pts.
Feeding Fault	62	92	251
Excitation Error	80	84	168
Malfunction Air Cooling	228	20	62
Generator Heating Fault	9	6	43

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 1. This data was used to train the classifiers and was labelled according to three different processes explained in Section 3.1. The initial operational data contained roughly 45,000 data points, representing the 11 months analysed in this study.

2.2. 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 the cut-in speed, or when the turbine is in "storm" mode, i.e., when the wind speeds are above a certain threshold. 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. Each time the 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 2 for a sample of the status message data. Any main 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".

2.3. Faults Classified

As mentioned in Section 2.2, any "main status" above zero indicates abnormal behaviour, but not necessarily a fault. Although over forty different types of faults occurred in the eleven months of data, only a small number occurred frequently enough to be able to accurately classify them. These faults are summarised in Table 3. 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, *malfunction air cooling* refers to problems in the air circulation and internal temperature circulation in the turbine, and *generator heating faults* refer to the generator overheating.

3. METHODOLOGY

In this paper, we attempt three levels of classification: fault detection, fault diagnosis and fault prediction. The general methodology for all three types of classification is shown in Figure 1. As can be seen, there are four main steps following a general machine learning process, described in detail in this section.

3.1. Data Labelling

The processes for labelling the data for each classification level are given below.

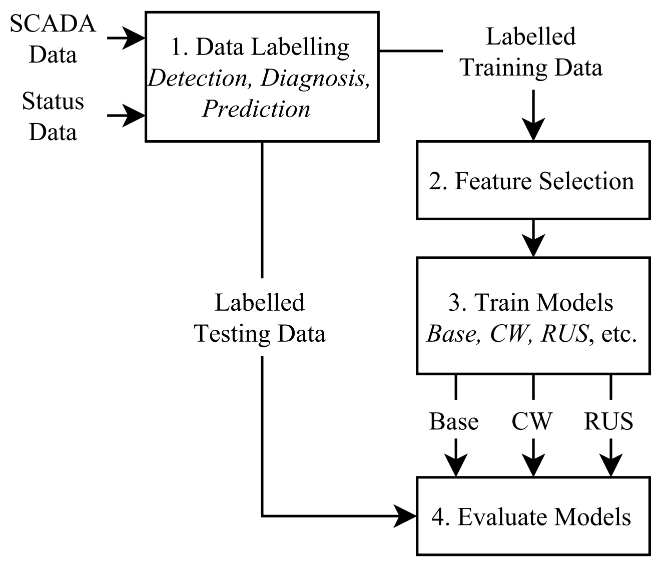

Figure 1. Methodology, following a typical machine learning approach. The abbreviations in step 3 are a selection of the approaches described in section 3.3

3.1.1. Fault Detection

The first level of classification is distinguishing between two classes: "fault" and "no-fault". The fault data corresponds to times of operation under a set of specific faults mentioned in Section 2.3. 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 time-band was selected so as to definitely capture any 10-minute period where a fault occurred, e.g., if a power feeding 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. No matter the type of fault, all faults were simply labelled as the "fault" class. All the remaining data in the operational dataset was then given the label "other", representing all other data. This is because the remaining data didn't necessarily represent fault-free data; it just meant it didn't contain the faults mentioned in Section 2.3, but could have included other, less frequent faults or times when the turbine power output was being curtailed for any one of a number of reasons mentioned previously.

3.1.2. Fault Diagnosis

Fault diagnosis represents a more advanced level of classification than simply fault detection. The aim of fault diagnosis is to identify specific faults from the rest of the data. Faults were labelled in the same way as in the previous section, but this time, each fault was given its own specific label. 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. Any data that remained, i.e., was not labelled as one of the faults mentioned in Section 2.3, was again given the "other" label, for a total of five different classes (the four fault classes as well as the "other" class). If two faults occurred concurrently, then the data point was duplicated, with each point having a different label.

Table 4. Values of T and W, in hours, used for fault prediction

Case	T	W
A	1	0
B	2	1
C	3	2
D	5	2
E	12	2
F	24	12

3.1.3. Fault Prediction

Fault prediction represents an even more advanced level of classification than fault diagnosis. The aim of this level of classification was to see if it was possible to identify that a specific fault was imminent from the full set of operational data. After initial tests, it was decided to focus prediction only on generator heating and excitation faults as these showed the greatest promise for early detection. Details of this can be found in Section 4. The other faults were included in the "other" label along with the rest of the data, for a total of three classes.

For fault prediction, the times during which the turbine was in faulty operation were not labelled as such. Instead, operational data points leading up to each fault were labelled as "pre-fault", for each specific fault. When a specific fault started at time t, then all operational data points between time $t-T$ and $t-W$ were labelled as that fault's "pre-fault" data. This means that by looking at a window of time between T and W before a fault occurs, useful warning could be given of an imminent fault at least W minutes/hours before it occurs. A number of separate cases representing different values of T and W were tried to see how far in advance an accurate prediction could be made. These can be seen in Table 4. For example, for case F, all data points with timestamps between 24 and 12 hours before a generator heating fault occurred were labelled as "generator heating fault". The same was applied to excitation faults. All remaining unlabelled data points were labelled "other". Once again, if different faults occurred concurrently, the data points were duplicated and given different labels.

This would mean that, for case F, for example, if the trained classifier detected a new, live, data point as "generator fault", it would mean that a generator heating fault is likely to occur in the next 12 hours. The technician or maintenance operator would then have between 12 and 24 hours between this point being detected and the fault actually occurring to remotely or manually inspect the generator and organise any necessary maintenance actions ahead of time. If maintenance is needed, this can reduce the logistics lead time or allow it to be scheduled in conjunction with other maintenance activities for maximum economic benefit.

3.2. Feature Selection

The full operational dataset had more than sixty features, many of which were redundant, incorrect or irrelevant. Because of this, only a subset of 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, while others contained duplicate or redundant values. These were removed. A number of the remaining features which were deemed as obviously irrelevant based on some basic domain knowledge were also excluded, e.g. features relating to the cumulative uptime of the anemometer or the open/closed state of the tower base door. This resulted in 39 remaining features. A subset of *these*, 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. It was decided to scale all features individually to unit norm because some, e.g., power output, had massive ranges from zero to thousands, whereas others, e.g., temperature, ranged from zero to only a few tens.

3.3. Model Selection

Support Vector Machines are a widely used and successful machine learning algorithm for the type of classification problem seen in this study, 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 (Cortes & Vapnik, 1995; Boser, Guyon, & Vapnik, 1992). 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 have been used in other industries for condition monitoring and fault diagnosis with great success. A review by Widodo showed that SVMs have been successfully used to diagnose and predict mechanical faults in HVAC machines, pumps, bearings, induction motors and other machinery (Widodo & Yang, 2007) . CM using SVMs has also found success in the refrigeration, semiconductor production and chemical and process industries (Laouti, Sheibat-othman, & Othman, 2011). Previous work by the authors demonstrated promising early results that SVMs are successful in detecting wind turbine faults using SCADA data (Leahy, Hu, Konstantakopoulos, Spanos, & Agogino, 2016).

Table 5. Summary of different training approaches taken, showing their general category and which classification levels used these approaches.

Approach	Category	Classification Levels
Base	General	Det/Diag/Pred
CW	General	Det/Diag/Pred
RUS	Undersample	Det/Diag/Pred
CC	Undersample	Det/Diag/Pred
TL	Undersample	Diag/Pred
ENN	Undersample	Diag/Pred
SM	Oversample	Diag/Pred
EE	Ensemble	Diag/Pred
Bag-CW	Ensemble	Pred
Bag-RUS	Ensemble	Pred

Each of the labelled datasets mentioned in section 3.1 were randomly shuffled and split into training and testing sets, with 80% being used for training and the remaining 20% reserved for testing. The data in all cases was heavily imbalanced - the number of fault-free samples was on the order of 10^2 times that of fault samples. This can sometimes be a problem for SVMs, so a number of different approaches were taken to address the issue. These included adding an extra "class weight" hyper-parameter, or under/oversampling the training data being fed into the SVM. In addition to this, because fault diagnosis and prediction represented a greater classification challenge, approaches using ensemble meta-learners were used to reduce bias and variance in the results. In any case, the test data was not altered in any way so as to preserve the imbalanced distribution seen in the real world.

The various approaches used can be broken into four general categories: "general", "undersampling", "oversampling" and "ensemble methods". Because of the simpler problem it posed, fault detection was only carried out using a small subset of these methods, whereas both diagnosis and prediction were carried out using additional approaches, summarised in table 5. Additionally, fault prediction used two bagging-based approaches. These are described in detail in the sections below.

Note that each level of classification above fault detection used multi-class classification based on the "one-against-one" approach (Knerr, Personnaz, & Dreyfus, 1990). For all three levels of classification, the models were trained using scikit-Learn's implementation of LibSVM (Chang & Lin, 2011; Pedregosa et al., 2012). For the fault prediction case, all approaches were initially trained based on the labelled dataset representing fault prediction window A from Table 4, as described in Section 3.1.3. The best performing approach was then used to train classifiers on fault prediction windows B, C, D, E and F.

3.3.1. General Approaches

Base Case (Base)
In the base case, i.e. "vanilla" SVM, a randomised grid search was performed over a number of hyperparameters to find the ones which yielded the best results on the full set of training data. These were then verified using 10-fold cross validation. The scoring metric used for cross validation was the $F1$ score (see Eq. 4). 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 sample 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 training data from all undersampling and oversampling methods were fed into an SVM following this approach. Additionally, the meta-learners using the ensemble methods also followed this approach.

Addition of Class Weight (CW)
In this approach, an additional hyperparameter, the class weight, $c.w.$, is added during training. This is a weighted scaling factor used when calculating C for the minority class. The new value for C for the fault class, C_w, is calculated as in Eq. 1.

$$C_w = C * c.w. \tag{1}$$

Training is then performed as in the base case. A number of different class weights ranging from 1 ($C_w = C$) to 1,000 ($C_w = 1000 * C$) were added to the set of hyperparameters being searched over for this approach. There is no over- or undersampling used in this method.

3.3.2. Undersampling Methods

Random Undersampling (RUS) This approach randomly undersamples the majority fault-free class (without replacement), so that the number of fault-free samples in the training data was equal to the number of fault samples.

Cluster Centroids (CC) This undersampling method splits all the samples of the majority class into k clusters using the k-means algorithm. The centroids of these clusters are then used as the new samples for this class. In this case, the value of k used was equal to the number of samples in the minority class.

TomekLinks (TL) TomekLinks is an undersampling method based on a modification of the condensed nearest neighbour algorithm (Tomek, 1976). For our application, the fault free class was undersampled to bring the number of samples down to near the number of samples in the largest fault class.

Edited Nearest Neighbours (ENN) The Edited Nearest Neighbours method is a slight modification of the k-nearest neighbours method used to undersample from the majority class (Wilson, 1972).

3.3.3. Oversampling

SMOTE (SM) SMOTE (Synthetic Minority Over-Sampling Technique) is an algorithm that generates synthetic samples for the minority class along the line connecting each sample in the minority class to its k-nearest neighbours (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). In this case, a number of new synthetic samples were generated for each fault class to bring the number of samples in line with the number of fault-free samples.

3.3.4. Ensemble Learners

Bagging (Bag-CW, Bag-RUS) Bagging, or BootstrapAggregating, is an ensemble technique designed to reduce overall variance and avoid overfitting (Breiman, 1996). Two different bagging classifiers were trained; one using the additional "class weight" hyperparameter, as described in section 3.3.1; and another using a randomly undersampled training set, as described in section 3.3.2.

EasyEnsemble (EE) EasyEnsemble is a modification of the AdaBoost algorithm (Freund & Schapire, 1995) which uses random undersampling to addresses problems of class imbalance (Liu, Wu, & Zhou, 2006).

3.4. Model Evaluation

A number of scoring metrics were used to evaluate final performance on the test sets for fault detection and fault diagnosis, as well as the six test sets representing the time windows A-E for fault prediction. A high number of false positives can lead to unnecessary checks or corrections carried out on the turbine, and this was captured with the precision score (where a higher score represents a lower false positive rate). A high number of false negatives, on the other hand, can lead to failure of the component with no detection having taken place (Saxena et al., 2008). This is captured by the recall score, where a higher number indicates a low ratio of false negatives. The F1-Score was also used, which is the harmonic mean of precision and recall. Confusion matrices were used where appropriate to give a visual overview of performance and show absolute numbers.

The formulae for calculating precision, recall, the F1-score and specificity can be seen below:

$$Recall = tp/(tp + fn) \tag{2}$$

$$Precision = tp/(tp + fp) \tag{3}$$

$$F1 = 2tp/(2tp + fp + fn) \tag{4}$$

Table 6. Results for Fault Detection

Method Used	Pre.	Rec.	F1	Spec.
Base	.02	.78	.04	.49
CW	.04	.83	.07	.82
RUS	.02	.9	.04	.67
CC	.02	.95	.03	.5

$$Specificity = tn/fp + tn \tag{5}$$

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.

The overall accuracy of the classifier on each test set was not used as a metric due to the massive imbalance seen in the data. 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%. Specificity was not used as a metric for a similar reason, though was used in one specific case for benchmarking against specificity scores in a previous study.

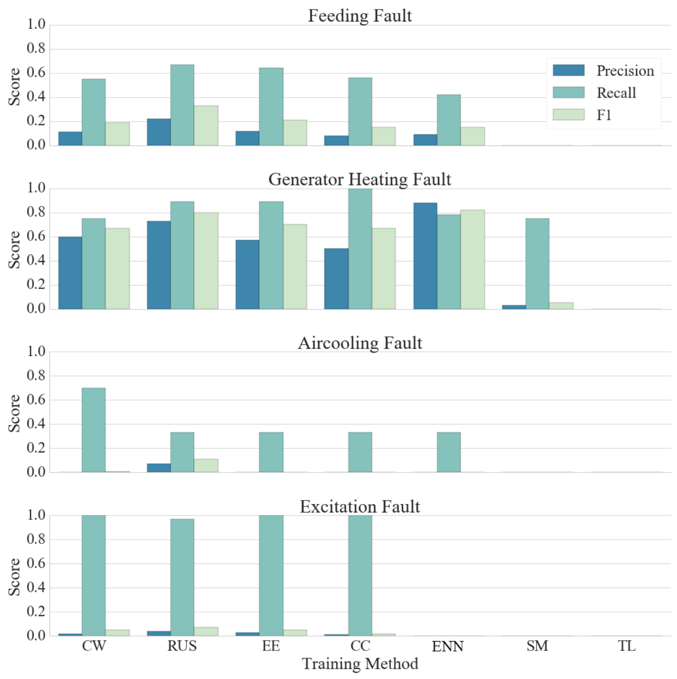

Figure 2. Precision, Recall and F1 scores for fault diagnosis across various training methods

4. RESULTS & DISCUSSION

4.1. Fault Detection

For fault detection, the recall score was generally high, ranging from .78 to .95, as seen in Table 6. However, the $F1$ score was brought down by poor precision scores all-round, representing a degree of false positives. Specificity was highest using the CW method with a score of .82, followed by RUS with .67, but suffered on the Base and CC methods, with .49 and .5, respectively. The best balanced performance was seen on CW, with good recall and specificity scores of .83 and .82, respectively. The precision of .04 was very low, but was among the highest of all results seen. The scores here were not as high as the fault detection scores in (Leahy et al., 2016), but this is to be expected due to the extra data included in this dataset which may not be clearly faulty or fault-free. Despite this, the scores were still an improvement on those obtained in (Kusiak & Li, 2011), where the best recall and specificity for fault detection were 0.84 and 0.66, respectively (compared with .95 and .82 here).

4.2. Fault Diagnosis

The scores on each fault for every fault diagnosis method are summarised in Figure 2. As can be seen, scores for the SVM trained using the CW method were generally slightly worse than RUS, apart from in the case of aircooling faults with a recall score of 0.7 (up from the randomly undersampled training set score of 0.33). This increase, however, may be because there were only 7 instances of air cooling fault in the test set, leaving it open to different test scores in each case. The CC and EE methods both performed slightly worse again, with CC slightly beating EasyEnsemble. The SVM trained on data undersampled using the ENN approach performed worse than both the EE and CC methods overall, but achieved a better F1 score on generator heating faults than the "vanilla" RUS method (0.82 vs. 0.8). However, this was down to improved precision (0.88 vs. 0.73) at the expense of recall (0.78 vs. 0.89). In the realm of fault detection and diagnostics, recall usually has a higher priority than precision due to the costs involved in missing a fault vs. a false alarm. Both TL and SM performed by far the worst overall. SM uses synthetic data to populate the minority class, so its poor performance suggests that using synthetic data is not suitable for this application.

The RUS method performed the best overall. Results for this can be seen in the confusion matrix in Figure 3. Generator heating faults showed a low proportion of false positives, as well as correctly catching 89% of faults. Excitation faults similarly showed a high proportion of caught faults at 97%, but was let down by a high number of false positives leading to a low precision score of .04. 67% of feeding faults were caught, but here also the number of false positives led to a precision score of .22.

4.3. Fault Prediction

Early results on fault prediction showed that generator heating and excitation faults showed the best promise for effec-

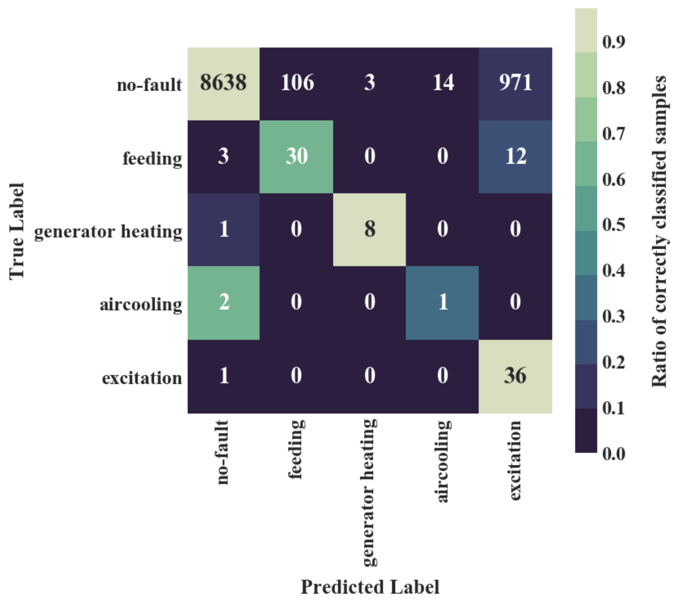

Figure 3. Confusion Matrix showing the ratio of correctly classified samples for fault diagnosis using the RUS training method

tive prediction. Feeding and air cooling faults showed very poor performance, possibly due to a separating hyperplane for these faults being hard to find in the limited data available for these particular faults. For this reason, it was decided to focus on generator heating and excitation faults.

As described in Section 3.3, the various training methods were first tried on the labelled dataset representing fault prediction window A from Table 4. The best performing of these was then used on the other fault prediction windows.

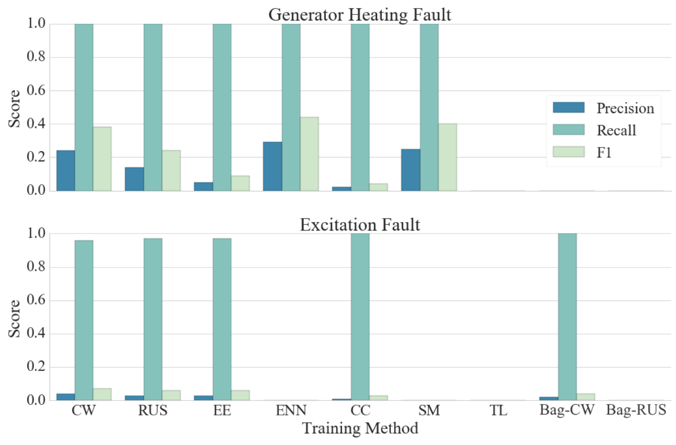

Figure 4. Precision, Recall and F1 scores on fault prediction for various training methods

The scores for each training method across the different faults using prediction window A are shown in Figure 4. Surprisingly, the best test scores were not seen on any of the ensemble classifiers, but on the SVM trained with the CW method, using a linear kernel. The full results for CW can be seen in Table 7. As can be seen, the recall score is very good, but again the $F1$ is brought down by poor precision. RUS came in close behind, but with lower precision on both faults. EE and CC both performed worse, with lower scores on precision for generator heating faults. ENN and SM both performed better than CW on generator heating faults, but were let down by a zero $F1$ score on excitation faults. TL performed very badly, with $F1$ scores of zero for both faults. Both bagging methods also performed very poorly. The poor performance of all three ensemble methods was surprising, and may not have performed as well as hoped because of the heuristic feature selection used. With a feature selection algorithm, test scores may improve.

Table 7. Results for Fault Prediction Using CW Method on Fault Prediction Window A

Faults	Pre.	Rec.	F1
Generator Heating Fault	0.24	.98	0.38
Excitation Fault	0.04	0.96	0.07

The test results from CW for various cases of time in advance of a fault, as described in Section 3.1.3, can be seen in Figure 5. The recall score for both generator heating and excitation faults stays relatively high for all cases. Both have a recall score of .97-.99 for cases A & B. This falls to around 0.8 for cases C & D for generator heating faults, but rises again to above 0.9 for cases E & F. Excitation faults start to drop just below 0.8 for cases E and F, but this is still quite high. These results show that good indicators of a developing fault are seen up to 12-24 hours in advance of a fault solely looking at 10-minute SCADA data. Previous work done in (Kusiak & Li, 2011) showed a recall score of just .24 one hour in advance of a specific fault, and this was the furthest window tested. It should also be noted that the results in that work represented a test set which did not fully sample the fault-free class. In (Kusiak & Verma, 2011), the maximum prediction time was 10 minutes, with a recall score of .71 when detecting blade pitch faults, using SCADA data at a 1s resolution (rather than the 10 minute data sued in this study). Hence, these results are extremely promising.

Confusion matrices for case A and D are shown in Figure 6. For excitation faults, although nearly all faults were successfully predicted in advance, there were a high number of false positives, leading to a low precision score of below 10% in both cases. Generator heating faults saw more success. There was a 40% precision score for case A, and a 22% score for case D. Although these scores are quite low, the inherent value of a SCADA-based system is that it does not require the installation of any additional sensors or other hardware, so can sit alongside existing systems with little additional cost. Any alarms generated by the system showing impend-

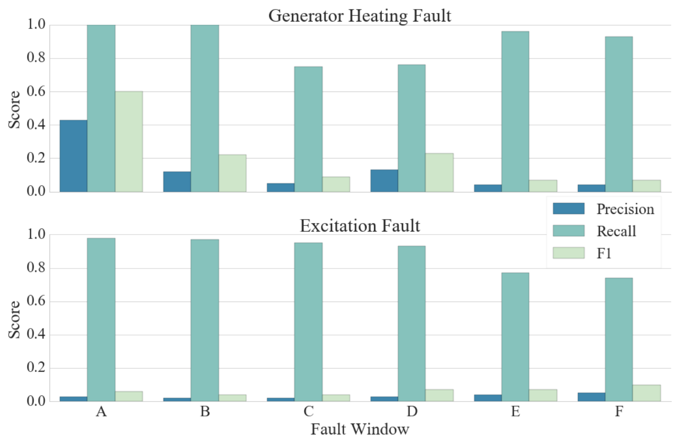

Figure 5. Precision, Recall and F1 scores for using CW training method to predict generator heating and excitation faults for the cases shown in Table 4

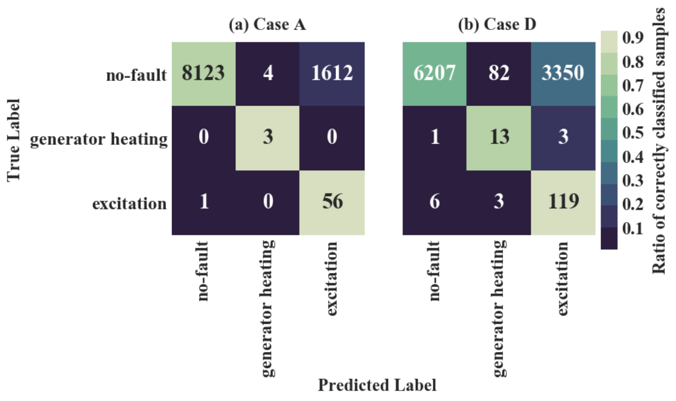

Figure 6. Confusion matrices showing the ratio of correctly classified samples for fault prediction using the CW training method on prediction cases A and D

ing faults can be remotely investigated to determine if action needs to be taken. Work by the authors in (Hu et al., 2016) has shown promising early results in improving precision scores for fault detection by using time-lagged and statistical features, and will be applied to both fault diagnosis and prediction in future work.

5. CONCLUSION

Various classification techniques based on the use of SVMs to classify and predict faults in wind turbines based on SCADA data were investigated. Three levels of fault classification were looked at: fault detection, i.e. distinguishing between faulty and fault-free operation; fault diagnosis, whereby faulty operation was identified and subsequently the nature of the fault diagnosed; and, fault prediction, where a specific fault was identified as being likely to occur in advance of the fault actually occurring. The classification techniques employed involved various different ways of training SVMs, including resampling the data as well as using ensembles of SVMs. 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. In general, this was also the case for classifying a specific fault. More importantly, predicting certain types of faults was possible up to twelve hours in advance of the fault with very high recall scores. Although F1 score was brought down by a poor precision, this still represents a significant increase over what was previously possible using 10 minute SCADA data. Improving the precision scores would represent a very important step forward in being able to rely on SCADA data for accurate fault prediction.

The data used in this study related to a single turbine over an eleven month period. This represents a limitation in what can be achieved; with additional data, i.e., from more turbines over a longer period, better fault prediction will be possible due to more positive examples being available for training. As well as this, advanced feature extraction and selection would enable even higher scores. Work done by the authors in (Hu et al., 2016) showed improved precision scores on fault/no-fault classification by using domain knowledge, temporal and statistical features, followed by using feature selection methods to find only the relevant features and speed up training time. The feature extraction and selection methods described in that work will be applied to the fault diagnosis and prediction described in this paper to improve prediction performance. Furthermore, an additional classification step will be performed to determine *which* window a potentially faulty test point is most likely to fall into, be it 2, 3, 5 or 12 hours in advance. For commercial viability, the ideal balance between precision and recall will be explored, i.e., is it cheaper to have many false positives with associated maintenance checks, or more false negatives and the associated replacement costs. An important next step is also to investigate whether trained models will still be accurate after a significant change in the turbine, e.g. after having a major component replaced.

ACKNOWLEDGEMENT

This research was funded by Science Foundation Ireland (SFI) Centre MaREI - Centre for Marine and Renewable Energy (12/RC/2302) and the Fulbright Commission in Ireland in partnership with the Sustainable Energy Authority of Ireland. Supporting funding was obtained from 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.

REFERENCES

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, 144–152. doi: 10.1.1.21.3818

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, *24*(421), 123–140. doi: 10.1007/BF00058655

Chang, C.-c., & Lin, C.-j. (2011). LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*, 1–39. doi: 10.1145/1961189.1961199

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. doi: 10.1613/jair.953

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. doi: 10.1007/BF00994018

Du, M., Tjernberg, L. B., Ma, S., He, Q., Cheng, L., & Guo, J. (2016). A SOM based Anomaly Detection Method for Wind Turbines Health Management through SCADA Data. *International Journal of Prognostics and Health Management*, *7*, 1–13.

European Wind Energy Association (EWEA). (2009). *The Economics of Wind Energy* (Tech. Rep.).

Freund, Y., & Schapire, R. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. *Computational learning theory*, *55*(1), 119–139. doi: 10.1006/jcss.1997.1504

Gayo, J. B. (2011). *ReliaWind Project final report* (Tech. Rep.).

Gill, S., Stephen, B., & Galloway, S. (2012). Wind turbine condition assessment through power curve copula modeling. *IEEE Transactions on Sustainable Energy*, *3*(1), 94–101. doi: 10.1109/TSTE.2011.2167164

Godwin, J. L., & Matthews, P. (2013). Classification and Detection of Wind Turbine Pitch Faults Through SCADA Data Analysis. *International Journal of Prognostics and Health Management*, *4*, 11.

Hu, R. L., Leahy, K., Konstantakopoulos, I. C., Auslander, D. M., Spanos, C. J., & Agogino, A. M. (2016). Using Domain Knowledge Features for Wind Turbine Diagnostics. In *Icmla.*

Knerr, S., Personnaz, L., & Dreyfus, G. (1990). Single-layer learning revisited: a stepwise procedure for building and training a neural network. *Neurocomputing*(68), 41–50.

Kusiak, A., & Li, W. (2011). The prediction and diagnosis of wind turbine faults. *Renewable Energy*, *36*(1), 16–23. doi: 10.1016/j.renene.2010.05.014

Kusiak, A., & Verma, A. (2011). A data-driven approach for monitoring blade pitch faults in wind turbines. *IEEE Transactions on Sustainable Energy*, *2*(1), 87–96. doi: 10.1109/TSTE.2010.2066585

Laouti, N., Sheibat-othman, N., & Othman, S. (2011). Support Vector Machines for Fault Detection in Wind Turbines. *18th IFAC World Congress*, 7067–7072.

Lapira, E., Brisset, D., Davari Ardakani, H., Siegel, D., & Lee, J. (2012). Wind turbine performance assessment using multi-regime modeling approach. *Renewable Energy*, *45*, 86–95. doi: 10.1016/j.renene.2012.02.018

Leahy, K., Hu, R. L., Konstantakopoulos, I. C., Spanos, C. J., & Agogino, A. M. (2016). Diagnosing Wind Turbine Faults Using Machine Learning Techniques Applied to Operational Data. In *Ieee international conference on prognostics and health management.*

Liu, X. Y., Wu, J., & Zhou, Z. H. (2006). Exploratory under-sampling for class-imbalance learning. *Proceedings - IEEE International Conference on Data Mining, ICDM*, *39*(2), 965–969. doi: 10.1109/ICDM.2006.68

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2012, jan). Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830. doi: 10.1007/s13398-014-0173-7.2

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. *2008 International Conference on Prognostics and Health Management, PHM 2008.* doi: 10.1109/PHM.2008.4711436

Skrimpas, G. A., Sweeney, C. W., Marhadi, K. S., Jensen, B. B., Mijatovic, N., & Holboll, J. (2015). Employment of Kernel Methods on Wind Turbine Power Performance Assessment. *IEEE Transactions on Sustainable Energy*, *6*(3), 698–706. doi: 10.1109/TSTE.2015.2405971

Tamilselvan, P., Wang, P., Sheng, S., & Twomey, J. M. (2013). A Two-Stage Diagnosis Framework for Wind Turbine Gearbox Condition Monitoring. *International Journal of Prognostics and Health Management*(Special Issue on Wind Turbines PHM).

Tomek, I. (1976). Two Modifications of CNN. *IEEE Transactions on Systems, Man and Cybernetics 6*, 769–772. doi: 10.1109/TSMC.1976.4309452

Uluyol, O., Parthasarathy, G., Foslien, W., & Kim, K. (2011). Power Curve Analytic for Wind Turbine Performance Monitoring and Prognostics. *Annual Conference of the Prognostics and Health Management Society*, 1–8.

Widodo, A., & Yang, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, *21*(6), 2560–2574. doi: 10.1016/j.ymssp.2006.12.007

Wilson, D. L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man and Cybernetics*, *2*(3), 408–421. doi:

10.1109/TSMC.1972.4309137

Yang, W., Tavner, P. J., Crabtree, C. J., Feng, Y., & Qiu, Y. (2014). Wind turbine condition monitoring: Technical and commercial challenges. *Wind Energy*, *17*(5), 673–693. doi: 10.1002/we.1508

Zaher, A., McArthur, S., Infield, D., & Patel, Y. (2009, sep). Online wind turbine fault detection through automated SCADA data analysis. *Wind Energy*, *12*(6), 574–593. doi: 10.1002/we.319