A Generalized Method for Fault Detection and Diagnosis in SCADA Sensor Data via Classification with Uncertain Labels

Md Ridwan Al Iqbal, Rui Zhao, Qiang Ji, Kristin P. Bennett

Dept. of Computer Science, Dept. of Electrical, Computer & Systems Engineering, Institute for Data Exploration and Applications
Rensselaer Polytechnic Institute
iqbalm@rpi.edu, zhaor@rpi.edu, qji@ecse.rpi.edu, bennek@rpi.edu

Abstract—Supervisory control and data acquisition (SCADA) systems automatically collect data from an array of sensors inside large-scale industrial machines such as wind turbines and chemical reaction chambers. We propose a new generalized method to use SCADA sensor data in real time to detect potential faults and indicate sensors associated with these faults. The problem can be regarded as a classification problem with uncertain or soft labels since the onset time of each fault is not known. A novel data transformation technique is proposed that allows any weight-sensitive classification algorithms to be used on data with soft labels. The method uses a decreasing function to assign instance weights to the instances based on the time interval before the failure. Each instance appears in the model as a positive and negative example with different weights. The method then can use any instance-weighted classification algorithm such as random forest and SVM. We compare this soft-label approach with a naive hard label approach and also propose within-outage and cross-outage testing scheme for more comprehensive evaluation. We develop a novel method for diagnosis that uses feature importance score. Experimental results on wind turbine sensor data support the effectiveness of the proposed method at both detecting and diagnosing faults. Our analysis of the calculated feature importance scores provides evidence of multiple potential root causes for a particular fault class in wind turbines that belong to the same wind farm.

Index Terms—Fault detection, Wind Turbines, SCADA, Uncertain labels

I. INTRODUCTION

Supervisory control and data acquisition (SCADA) is a system that consists of an array of sensors that monitor the condition of various subsystems of various machines such as turbines, chemical reaction chambers, and blast furnaces [1]. Continuous monitoring of SCADA sensor data is typically done via univariate rule based systems to proactively detect faults [1]. Detecting faults that are not directly captured by one sensor or that have more complex multivariate root causes remains a challenge. The sensors monitor conditions such as the temperature, pressure, voltage, angle of operation, and fluid level. Condition based monitoring is also being used instead of periodic maintenance on large scale systems [2]. Current research in condition monitoring has seen development of techniques such as vibration analysis [3] and oil particle analysis [4]. But, these methods also rely on installation of extra sensors that can be prohibitive.

We propose a soft-label classification method that can proactively detect faults without any information about exact time of fault onsets. Our approach classifies the data based on only the information that the machine has a forced shutdown at a certain time period. The method then also identifies sensors putatively associated with the root causes of faults to support rapid diagnosis. The comprehensive soft-label classification method transforms the time-series sensor data using a continuous moving window and assigns a probabilistic classification label to each window based on increasing certainty as the fault approaches. This transformed data can then be used with any type of instance-weighted classification algorithm. This allows the use of both parametric and nonparametric classification algorithms. We demonstrated using random forests, support vector machines (SVM), and k-nearest neighbor (KNN) for classification. We also propose a method to identify the root causes of faults based on soft-label random forest importance feature scores.

We examine two modeling scenarios. In-outage modeling creates a fault detection model for each machine i.e. a single wind turbine. Cross-outage modeling creates a single model for a set of machines, e.g. multiple turbines in a wind farm. This work builds on and significantly expands the capabilities of our prior SVM-based approach [5] for within-outage modeling. This comprehensive approach of addressing within- and cross-outage modeling more closely reflects how the system would be used in practice. In addition, feature importance scores from the soft-label random forest helps convert the fault detection into actionable insights for maintenance workers.

We train and test the algorithm using one year of SCADA data from a wind turbine farm with 125 turbines. Our method for detecting faults can reduce the cost of maintenance by detecting faults automatically before machine damage happens.

We are focusing on a single fault class known as “CCU fault current” which is triggered by incorrect current measured from the central control unit. Simple threshold-based fault detection systems are not effective on this fault class.

The time of onset of the fault leading to a forced outage is
unknown. This lack of clear separation between a faulty state and a normal state poses a significant challenge. Our approach assumes a certain level of uncertainty about the labels and a soft and gradual change from normal state to faulty state based on a soft decay function.

Our experiments identified the most influential features for faults by calculating a feature importance score from the soft-label random forest classification model for each turbine. Our diagnosis method indicates there is evidence of multiple root causes for the same type of faults in different wind turbines.

This paper is organized as follows: We will first review the previous research on fault detection with specific focus on wind turbines. Later, the modeling method is described. Next, we describe the data itself. Finally, we examine the experimental results.

II. RELATED WORK

Classification with uncertain labels has been explored by Xue et al. [6], who proposed a noisy-label model in which a noise function is used for labeling purposes. This function is derived from reliability analysis and bootstrapped with different training sets. Another common approach to fault detection is the use of signal processing units (SPU). This approach uses process parameters and tries to divide operation into normal and faulty category using pre-specified limits and tolerances. Common methods for signal analysis in frequency domain include fast Fourier transformation and cepstrum analysis [3]. Zaher et al. [1] used neural network that models the normal state of the turbine and tries to detect anomaly based on the state. Schlechtgen et al. used adaptive neuro-fuzzy interference Systems (ANFIS) to model the normal behavior of turbines [7].

A condition-based monitoring (CBM) system must provide maximum benefit while also minimizing cost. Condition monitoring research in the past mostly focused on domain knowledge based fault detection [8]. Surveys on wind turbine failures show that gearbox, electric system and blades are the most common fault generating subsystems [9], [10]. Wilkinson and Tavner [8] applied CBM techniques on wind turbine drive train. They used the double-fed induction generator, main bearing, gearbox for their monitoring. The major trends noted in Lu et al. [11] was that time-frequency analysis tools such as wavelets signal processing are a common approach to CBM. This is due to the variation caused by external factors in the wind turbine. Zaher et al. [1] explained anomaly detection techniques used on SCADA data acquired from a wind farm, and tried to automate and simplify the operator’s task by interpreting the data available. They created one system to collate the output from anomaly detection and provided a single decision support interface.

III. FAULT DETECTION WITH UNCERTAINTY

A. Problem Formulation

There are three operational status defined in our problem formulation:

- **Normal** is when the machine’s operation satisfies its design specification with minimum risk of fault.
- **Prefault** is when the machine’s operation satisfies design specification but a fault has developed which eventually cause a forced shutdown.
- **Forced outage** is when the unit is shut down by the control system due to the fault.

In the first two states, the machine is still operating. Note only the time of the forced outage is known definitively. We know the machine has transitioned from normal to prefault prior to the forced outage, but the time of this transition is unknown.

The sensor data from each machine can be considered as a multi-dimensional time series. We can define the data as \( Z^L = \{ z_1, \ldots, z_{L-1} \} \), \( z_n = \mathbb{R}^d \), \( n = 0, \ldots, L - 1 \), where each instance \( z_n \) is a measurement of the sensors of the machine at a particular time. \( z_n \) is a vector of length \( d \) where \( d \) is the number of sensors in the time series. \( L \) is the total number of data points in the time series with a fixed interval. The machine starts at time 0 and has a forced outage at time \( L \). The goal is to detect fault of a turbine using sensor readings \( z_n \) at a time point \( n < L \).

In order to make this multivariate time-series into a form more usable for classification, we will use a mapping step to convert \( Z \) into the classification feature set \( X \). The classification of a time series needs to be able to detect a fault that might be slowly progressing over a period of time. An instance consisting of a single snapshot of the sensors will not be able to capture changes in the overall time series. Therefore, we consider each instance to be composed of a subsequence of samples that capture a larger time period. This subsequence will go through the data as a sliding window that will capture the time series. Let \( x_n^l \) be the instance of classification feature set \( X \) where \( l \) is the sliding window size and \( n \) is the last time point in the subsequence.

\[
    x_n^l = F(z_n, \ldots, z_{n-l+1}); \quad [1 \leq n \leq L, 1 \leq l \leq L] \tag{1}
\]

\( F(\cdot) \) is the feature construction mapping function that takes a sliding window of raw data as input. The feature construction is domain specific and will be discussed during the data description section. \( x_n^l = \mathbb{R}^m \) where \( m \) is the number of features constructed during the mapping phase. Let \( y_n^l \) be an integer variable indicating the label of subsequence \( x_n^l \).

We assume, a label \( y_n^l \) exists for the sample \( n \), that will specify the state of the turbine at that particular time point. This label could either be \( \text{Prefault} = +1 \), or \( \text{Normal} = -1 \). Our goal then is to classify the label of \( x_n^l \) to be either prefault or normal. Unfortunately, these labels are unknown on the data. As shown in Figure 1, there is an unknown moment in the time series when the machine switches from a normal state.

![Fig. 1. Time series with uncertain labels](image_url)
to a prefault state. A classification model that can learn with uncertain labels seems prudent for this problem.

B. Generalized Soft Label Classification Method

There is a hidden change-point in the data when the state shifts from normal to prefault. Figure 1 shows an illustration of the uncertain change point in the data. It is uncertain whether a certain instance $x_n^l$ is either in normal or prefault state.

We know that the finite transformed time series $X$ does not contain any samples with forced outage status. In addition, the forced outage state proceeds immediately after the last entry $x_{L-1}^l$. In other words, $X$ covers the temporal scope prior to a forced outage event. During training, we do not have the labels for each subsequence $x_n^l$. We assume that the probability of being in prefault state is non-decreasing as the sample gets closer to the outage:

$$P(y_n = 1|x_n^l) \geq P(y_k = 1|x_k^l), \forall n > k$$

(2)

The motivation of this assumption is that as time gets closer to a forced outage event, it is more likely for the machine to be going to a prefault state. Therefore, classifying a data point as partially normal and prefault seems prudent since this classification scheme preserves the uncertainty inherent in the data.

In our previous work, a soft-label support vector machine algorithm for detection was developed [5]. The soft-label SVM takes as input a training set with a probabilistic soft label. Assuming a training set $\{x_n, y_n\}_{n=1}^N$ of length $N$ data points, where $N = L - l + 1$ and $x_n \in \mathbb{R}^d$ and $y_n \in \{+1, -1\}$ is the associated classification label. We also assumed that:

$$P(y_n = 1|x_n^l) = u_n^+, \quad P(y_n = -1|x_n^l) = u_n^-$$

(3)

The probability of an instance being prefault is given by a function $p(n) = u_n^+; 1 = u_n^+ + u_n^-$. This assigned probability can be considered the probability of time $n$ being where the state of machine shifted from good to bad.

We proposed a modified SVM that can take this soft label into account during optimization. The modified SVM optimization problem is given as follows:

$$\min_w \frac{1}{2}||w||^2 + \gamma \sum_{n=1}^N (u_n^+ \xi_n^+ + u_n^- \xi_n^-)$$

s.t. $w^T \phi(x_n) + b \geq 1 - \xi_n^+$

$-w^T \phi(x_n) - b \geq 1 - \xi_n^-$

$\xi_n^+, \xi_n^- \geq 0; \quad n = 1 \cdots N$

(4)

The modified SVM (4) essentially assigns a weight to each instance based on the probability function. Thus, it can be considered an SVM that is capable of taking instance weights into account during learning. This essentially provides the intuition to generalize this approach to other learning methods that can also utilize instance weights.

Our proposed method generalizes soft-label learning to any classification methods that can learn with instance weights. The method trains with each instance by being both positive and negative with an associated weight. Each instance is presented twice during training phase with a positive label and a negative label respectively. We use a mapping function that takes the original data, and then create a new data set that has each instance two times and assigns an associated instance weight. This mapping function creates a data set $\{x_n, y_n\}_{n=1}^{2N}$

$$\begin{align*}
x_n^+ = x_n, y_n^+ &= +1, \quad u_n^+ = p(n) \\
x_n^- = x_n, y_n^- &= -1, \quad u_n^- = 1 - p(n)
\end{align*}$$

(5)

Thus, we have a new training data set that is twice the length of the original and we also have the associated instance weights. This forms a new classification problem.

A key aspect of the method is how to assign the instance weight $u_n^\pm$. We have chosen a sigmoid function $\tanh()$ to set the probability. In [5], we provide some alternative probability functions that could also be used. The advantage of choosing the hyperbolic tangent function is the fact that it provides exponentially decaying value from 1 to 0, which fits our goal to set a probability of failure of a particular instance. Moreover the steepness of the decay can be tuned with a parameter $\alpha$. Thus our final probability will be:

$$p(n) = \tanh(\alpha n) \quad n > 0, \quad \alpha > 0.$$ 

(6)

For a small value of $\alpha$, the resultant value of the function would be almost linear as the time interval increases. This parameter was tuned using a validation set.

Now the next step is to select an instance-weighted classification method. We studied 3 of the most popular classification algorithms that support instance weights: random forest [12], KNN [13] and SVM [14].

C. Fault Diagnosis with Feature Importance Score

Our diagnosis method finds important features from the soft-label classification model. Recall the data set consists of sensor values over time. Therefore, the we can suggest potential causes of faults by identifying important sensors that are correlated with the transition from the normal to prefault state. We identify features that have the most influence in the soft-label classification model. Out of the 3 different classification algorithms we have tried, random forests [12] had the best performance. Random forests algorithm also has the advantage of providing a feature importance score [7].

The score is calculated with the mean parent to child impurity reduction over all component trees. This score can then be used to rank the features with the most importance in the classification of prefault/normal. Once we calculate a feature important score for all the features, we can analyze this new feature score data set for potential root causes. One important goal of our diagnosis approach is that we need to identify if different machines have different root cause for the same fault type. We can run the classification for each machine separately in order to investigate this. The feature scores can then be analyzed with PCA or clustering to see if there multiple groups of influential features.
IV. FAULT DETECTION ON WIND TURBINES

A. Data Description

A wind turbine typically includes various mechanical and electrical components, assemblies and systems that may cause failure that are continuously monitored using SCADA. The SCADA data comes from a global wind energy company, who operates a large number of wind farms worldwide. The SCADA data is collected from 125 1.6MW wind turbines from a single farm over the time period from June 2013 to May 2014. There are 38 forced outage events. There is a total down time of 2305 hours with maximum and minimum down time of 544.7 hours and 1.7 hours respectively. There are 55 sensor values collected via SCADA measuring different quantities such as power generation, tower vibration, temperature, wind speed, blade angle etc. The sensor information is provided as an averaged value over 10 minutes. We excluded sensors that are mostly constant or that only contain a few distinct values. We also excluded channels with mostly missing values. This reduced our total number of sensors to 39. The sensors available in the data are listed in our previous paper [5].

B. Data Preparation and Normalization

We used a 6 hours sliding window to generate our instances. If we have L sensor readings and sliding window size l, this creates \( L - (l + 1) \) instances.

The first set of features we have are the raw sensor values of the sliding window. Since each sample is taken after 10 minutes, this gives us 36 instances in the 6 hour sliding window. The raw features then consist of 36 samples with 39 sensors each, \( 36 \times 39 = 1404 \) features.

This raw data set is augmented with extra features based on the raw data. The first type of feature that we have constructed is a pair-wise covariance of the sensors. We will construct \( 39 \times 39 \) number of covariance features from all pairs of sensors. Each sensor has \( l \) data points in the sliding window of \( x_{ni} \) that correspond to the sensor values at all the time points from \( n - (l + 1) \cdots n \). These features will then be part of classification feature set \( x_n \). These covariance values can significantly improve performance by incorporating the temporal correlation between sensors.

We have also applied customized normalization on some of the sensors and added these values as 900 new features. Some sensor values have specific relationship that can be exploited to make the values more aligned. A good example of this is the 3 voltage phase sensors. All 3 phases are closely related each other so it makes more sense to have 1 unified mean and standard deviation for all 3 phases. Examples of custom normalization include:

1) \( V_a, V_b, V_c \) are three-phase voltage value, we normalize each phase by subtracting the mean.

\[
\bar{V}_a = V_a - \bar{V} \tag{7}
\]

\[
\bar{V} = (V_a + V_b + V_c)/2 \tag{8}
\]

2) \( T_n \) is temperature of nacelle, we normalize by subtracting ambient temperature.

3) Ratio between \( \text{gen}_rpm \) and \( \text{rotor}_rpm \) should remain the same, we normalize \( \text{gen}_rpm \) by dividing with \( \text{rotor}_rpm \).

The next phase is to normalize the raw and covariance feature set. However, instead of normalizing all the turbines using the global mean and standard deviation, we can also normalize each turbine individually. This can help reduce the cross-turbine differences seen in the data.

V. EXPERIMENT SETUP

A. In-outage vs Cross-outage modeling

We are proposing two different types of experimental design: in-outage and cross-outage. There are different benefits of both approaches and wind farms can use both approach for different purposes.

The in-outage modeling creates a separate classification model for each turbine and uses only the data from that turbine for training and testing. This approach allows the diagnosis of faults in each turbine individually. This approach is very valuable for diagnosis analysis of faults covered in section III-C. In this testing scheme both training and testing can see the same turbine data but from different non-overlapping time periods. The testing and training sets are constructed based on the forced outage at time \( L \). We use the day \( L - 1 \) for testing, day \( L - 2 \) for training, days \( L - 3 \) to \( L - 6 \) for training, days \( L - 7 \) to \( L - 12 \) for testing and days \( L - 13 \) to \( L - 30 \) for training. This chronological split is appropriate because there is little variation between two adjacent points in an overlapping time window. Therefore, a random split between training and test set will not be a good measure of performance. Figure 2 shows how the data is divided in the in-outage test. The in-outage test is performed separately for each of 38 outages in the data set. The results of these experiments are aggregated. One practical limitation of the in-outage model is that faults must occur in each turbine in order to create a model.

The cross-outage modeling creates only one model for all the turbines. The model can be deployed at wind farm even when faults have not been observed in all turbines. The model can be robustly trained with more data. In our data, we have 34 outages for final training and testing, so they are randomly split into two equal groups and if an outage data is used for training, it is not used for testing at all. A validation set of 4 outages are reserved for validation and hyper-parameter selection. This testing scheme is much more robust but also much harder, since prediction during testing must be performed without any training data from that particular turbine. The cross outages test is performed 10 times with training and testing turbines selected randomly and the average performance is reported.

B. Hard Label Classification for Testing

To evaluate the results, We need to define hard label classifications to use for testing purposes regarding when the transition to prefault has happened. Thus, we worked with wind turbine engineers to come up a hard labeling scheme for testing. In this scheme, the first two days before a forced outage are considered to be prefault. While the data points
that are more than 7 days away from the outage are treated as normal or -1. The days 3 to 7 days before the outage are more uncertain in terms of labeling. These uncertain days are not used for testing but are used for training by the soft-label methods. After assigning the hard labels, we can use standard supervised learning techniques for comparison. Figure 2 shows an illustration of the hard label classification method and how the data is divided for in-outage testing.

VI. EXPERIMENTAL RESULTS

We performed a series of experiments to evaluate the effectiveness and generalization of the proposed approach. Since the data are heavily skewed towards the normal case, we use AUC (Area under Curve) as our performance evaluation measure [15] which is robust on skewed data sets.

We have used standard algorithms random forest, KNN and SVM in order to run the experiments. Matlab implementation of the algorithms have been used. Both linear and RBF kernel based non-linear SVM have been used. Validation and grid search was performed to determine algorithm parameters.

One important factor in this experiment is the shape of the decay function for the soft label method. We used the validation set to come up with the optimum value for the parameter \( \alpha \) using a simple grid search and the validation AUC. Figure 3 shows how the average validation AUC for the in-outage problem varies for different values of \( \alpha \). This performance is an indication that a lower value of \( \alpha \) provides a better performance. We found \( \alpha = 10 \) to have the best performance in the validation set. This indicates the transition from fault to the actual forced shutdown is a relatively slow process and therefore a less steep decay function yields better performance.

Table VI shows the results of uncertain soft-label classification versus hard-label classification for random forest, SVM, and nearest neighbors based on the parameters selected from the validation set. This result uses turbine specific normalization and \( \alpha = 10 \).

The soft-label method improved performance over the hard-label approach for all three methods. This can be attributed to the fact that only the soft label case utilizes the full data set without discarding a huge portion of data in the days 3-7; it also makes fewer assumption regarding the boundary between prefault and normal states. Soft-label random forests performed best overall for both in-outage and cross-outage with significant improvements in many cases. Soft-label random forests represents a significant advance over our prior soft-label SVM approach for in-outages [5] because of its consistently strong performance and superior ability to determine influential features over the SVM approach. In addition, random forests provided vastly superior performance on the more challenging cross-outage problem, which shows robustness without over-fitting. The performance advantage of random forest stems from the fact that random forest is a highly non-linear and robust method that performs well without much model selection. Moreover, random forest is also highly robust to noise in data. SVM suffers from poor AUC which is almost close to random guessing in the hard-label classification.

Table I results show that in-outage testing performs significantly better than cross-outage testing. In cross-outage, the presence of high standard deviations indicates that the model is performing very well in some turbines attaining almost perfect accuracy. But it is also performing poorly on other turbines. This indicates the same type of fault may have different causes on different turbines.
TABLE I
AVERAGE AUC ON IN-OUTAGE & CROSS TESTING

<table>
<thead>
<tr>
<th>Test</th>
<th>Mean AUC±Std. Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In-outage test</td>
</tr>
<tr>
<td>Hard Label</td>
<td>R. Forest: 0.889±0.09</td>
</tr>
<tr>
<td></td>
<td>KNN: 0.875±0.11</td>
</tr>
<tr>
<td></td>
<td>Lin-SVM: 0.773±0.15</td>
</tr>
<tr>
<td></td>
<td>RBF-SVM: 0.717±0.12</td>
</tr>
<tr>
<td>Soft Label</td>
<td>R. Forest: 0.912±0.09</td>
</tr>
<tr>
<td></td>
<td>KNN: 0.889±0.09</td>
</tr>
<tr>
<td></td>
<td>Lin-SVM: 0.841±0.15</td>
</tr>
<tr>
<td></td>
<td>RBF-SVM: 0.731±0.12</td>
</tr>
<tr>
<td></td>
<td>Cross-outage test</td>
</tr>
<tr>
<td>Hard Label</td>
<td>R. Forest: 0.727±0.11</td>
</tr>
<tr>
<td></td>
<td>KNN: 0.657±0.12</td>
</tr>
<tr>
<td></td>
<td>Lin-SVM: 0.564±0.15</td>
</tr>
<tr>
<td></td>
<td>RBF-SVM: 0.542±0.11</td>
</tr>
<tr>
<td>Soft Label</td>
<td>R. Forest: 0.753±0.09</td>
</tr>
<tr>
<td></td>
<td>KNN: 0.711±0.12</td>
</tr>
<tr>
<td></td>
<td>Lin-SVM: 0.665±0.16</td>
</tr>
<tr>
<td></td>
<td>RBF-SVM: 0.652±0.11</td>
</tr>
</tbody>
</table>

VII. Diagnosis via Influential Features

As discussed in section III-C, soft labels methods enable diagnosis of potential root causes of faults. The soft label random forest method provides influential features for each outage as part of the training process for the in-outage problem. We performed an analysis of the influential features for each turbine and found that there is one main group of turbines with similar root cause as well as outlier turbines.

Figure 4 shows the sorted plot of feature importance scores found by the soft label random forest. Each outage is associated with a small number of top features that have higher feature importance score. Thus we can only focus on the top 5 features from each turbine classification model.

Fig. 4. Sorted feature importance score plot

The separate feature importance scores from a particular turbine can be analyzed to determine the similarities and differences between turbines.

We performed PCA on this feature score data to show how the turbines are grouped. Figure 5 shows a bi-plot of the top 3 components from the PCA analysis. Most of the turbines cluster together in the center of bi-plot. This signifies that they have similar feature importance make up. Therefore, these turbines could potentially also have the same underlying cause for forced outage. We can clearly see some turbines are outliers with different feature makeup suggesting multiple root causes. Our analysis found Turbines 6, 9 and 21 to be strong outliers.

Table II provides the most influential features of these 3 outlier turbines as well as those determined by the mean importance of all the turbines. It shows the top 5 features with the highest importance scores. The features are named in a specific way. A feature with T200, for example, represents the sensor value that is 200 minutes away from forced outage.

The mean of the top features shows sensors hydraulic pressure and battery box axis to be important. This indicates that the most common cause of the fault in most turbines is abnormal hydraulic pressure and battery box axis. This could indicate the need for more routine maintenance to improve performance related to these sensors.

The outage model for the outliers show a different set of features. Turbine 6 shows anomalies with the battery box axis. Turbine 9 primarily shows problems with hydraulic pressures. Turbine 21 indicates totally different classes of sensors associated with cooling, high speeds, and the nacelle positions. These indicate potentially different root cause for the turbine faults.

Fig. 5. Feature importance score 3D bi-plot of top 3 components. Three outlier turbines have been marked

VIII. Conclusion

We proposed a new generalized approach for fault detection and diagnosis based on SCADA sensor data using a probabilistic uncertain-label framework usable with any instance-weighted classification algorithm. The approach was used to predict forced outages of wind turbines and diagnosis sensors associated with potential causes of these outages. We
examined the in-outage problem of predicting faults using a unique model for each turbine and the more challenging cross-outage problem of creating one model for all turbines. The cross-outage problem can predict faults from a turbine without having training data for that turbine. We tested the approach using random forests, SVMs and KNN. The proposed approach showed much better performance than hard-label classification for all three algorithms for both in-outage and cross-outage experiments. Soft-labels combined with random forests performed best overall.

The soft-label models provide a novel method to diagnosis potential root causes of failures. This can enable proactive maintenance. We determined important features by training a model for each turbine separately and calculating a feature importance score list for each of the models using the random forest results. Analysis of the feature importances showed that faults had multiple root causes in different turbines. We also concluded that the cross-outage predictive accuracy is lower because one single model may not be able to predict multiple root causes. We have identified that hydraulic pressure and temperature of the battery box are important indicators of potential problems in most of the turbines in this particular wind farm.

In the future, we plan to work on extended the uncertain-label models. One approach is develop uncertain-label tensor classification models. We believe such an approach can better incorporate the relationship between different sensors over time. We could also employ deep learning techniques on this problem in order to more create complex models. The challenge would be to extend the valuable diagnosis approach proposed here to these new modeling methods.

ACKNOWLEDGMENT

Supported by NSF grant #1331023 "EXTRÊME SQED: Data Analytics Throughout Undergraduate Mathematics".

REFERENCES