Azure machine learning studio and SCADA data for failure detection and prediction purposes: A case of wind turbine generator

A El-Menshawy*, Z Gul and I El-Thalji

Department of Mechanical and Structural engineering and material science, University of Stavanger, Stavanger, Norway.

* Contact author: a.elmenshawy@stud.uis.no

Abstract. Most industrial systems have supervisory control and data acquisition (SCADA) systems that collect and store process parameters. SCADA data is seen as a valuable source to get and extract insights about the asset health condition and associated maintenance operations. It is still unclear how appliable and valid insights SCADA data might provide. The purpose of this paper is to explore the potential benefits of SCADA data for maintenance purposes and discuss the limitations from a machine learning perspective. In this paper, a two-year SCADA data related to a wind turbine generator is extracted and analysed using several machine learning algorithms, i.e., two-class boosted decision tree, two-class decision forest, k-means clustering on Azure ML learning studio. It is concluded that the SCADA data can be useful for failure detection and prediction once rich training data is given. In a failure prediction context, data richness means ensuring that fault features are presented in the training data. Moreover, the logs file can be used as labelled data to supervise some algorithms once they are reported in a more rigorous manner (timing, description).

1. Introduction

The offshore wind installations in the North Sea have already passed the one-gigawatt capacity. Maintenance cost is one of the most operating expenses that operators are eagerly trying to cut it down. Offshore wind turbines can be seen as unmanned installations, which currently need several costly planned and unplanned visits. Condition monitoring systems and predictive maintenance (PdM) are also seen as potential ways to detect or predict the turbine conditions beforehand. The ultimate hope is to avoid any extract offshore visits asset damage, production shutdowns and create sufficient time intervals for maintenance planning. In the context of condition monitoring and PdM, the Supervisory Control and Data Acquisition (SCADA) systems became of high interest to provide insights for the decision support process. SCADA is built-in system of several industrial systems, e.g., wind turbines, whereas condition monitoring systems might require additional setup and instrumentations which might be costly and not allowed by the original equipment manufacturer.

A typical Supervisory Control and Data Acquisition (SCADA) collect and transmit averaged data every 10 min Kim, et al. [1]. It can be used for further examination with the help of machine learning algorithms to deduce the overall health of the turbine as well as its internal components Kim, et al. [1]. Thus, several researchers Maldonado-Correa, et al. [2], Astolfi [3], Wang, et al. [4] have investigated how can SCADA data be utilised with help of machine learning algorithms for wind turbine health
monitoring purposes. Wang, et al. [4] concluded that (i) the overall health of the turbine can be supervised by keeping track of wind speed and power output parameters, (ii) SCADA data can be useful for power prediction, optimal control settings, performance evaluation, predicting turbine faults (iii) AI model in CM of WTs is justifiable as there are many variables involved and it is not easy to establish an accurate mathematical model for such kind of complicated systems, AI techniques such as neural network (NN), Fuzzy systems logic, adaptive neuro-fuzzy inference systems (ANFIS) gives good results for condition monitoring of wind turbines. Wang, et al. [4] proposed that data mining (AI-based) and evolutionary computations could be integrated for building the models for prediction and monitoring.

Kim, et al. [1] discussed a number of measurements to develop anomaly detection algorithms and investigated classification techniques using clustering algorithms and principal components analysis for capturing faulty patterns in data.

Maldonado-Correa, et al. [2] and Astolfi [3] mentioned that Supervisory Control and Data Acquisition (SCADA) plays a vital role for an effective solution for condition monitoring as most of the wind turbines record large amounts of parameters using their SCADA data. SCADA data are crucial for industrial organizations since they help to maintain efficiency, process data for smarter decisions, and communicate system issues to help mitigate downtime. In Tautz-Weinert and Watson [5] author has thoroughly discussed different approaches to utilise SCADA data for the condition monitoring of wind turbines such as (i) trending, (ii) clustering, (iii) normal behaviour modelling, (v) damage modelling, and (vi) assessment of alarms and expert systems. It was concluded that simple trending of SCADA data has demonstrated good abilities to detect anomalies, whereas for the clustering method an extensive historical failure data is required for reliable diagnoses of failures. However, Maldonado-Correa, et al [2] on the other hand after their scientific literature review concluded that the most frequently found AI techniques for condition monitoring and wind turbine fault predictions are ANN (Artificial Neural Network) and SVM (Support Vector Machine), which according to Maldonado-Correa, et al [2] appearing in 39% and 27% of the total articles.

Zhang and Lang [6] proposed a novel dynamic model sensor method for the detection of faults in wind turbines using the SCADA data. The relationship between the generator temperature, wind speed, and ambient temperature was derived. Further, a novel nonlinear system frequency analysis is adapted to extract damage-sensitive features and they achieved acceptable results on detecting the turbine generator failure and its ageing trend. Artigao, et al [7] suggested that the control system, gearbox, electric system, generator, as well as hub and blades, are the most critical components of wind turbine and Hameed, et al. [8] have presented different condition monitoring techniques for the fault prediction both on system and subsystem level.

Tao, et al. [9] applied the Grey Correlation Algorithm and Support Vector Regression (SVR) to predict the power, rotor speed, and pitch angle. However, it is hard to correlate the changes in these three predicted parameters to detect fault at specific physical components.

Lima, et al. [10] applied the Normal Behavior Model (NBM) on wind turbine SCADA data and found this model helps to avoid overfitting and reduce the false-positive predictions. However, this model is predicting abnormal behaviour for each measurement channel at a specific time instant and not to make predictions for future days or weeks. Letzgus [11] applied a change-point detection algorithm on wind turbine SCADA data and was able to detect changes in gear bearing temperature, hydraulic oil temperature, and gear oil inlet pressure, almost three months before failure was notified. Castellani, et al. [12] applied principal component analysis (PCA) and support vector machine (SVM) on SCADA data related to a Vestas V52 wind turbine generator. The incipience of the fault was detected, two weeks before the usual fault alarm detected that, as a change in the behaviour of the residuals between model estimates and measurements. It was suggested to use more resolved SCADA data, e.g., order of the second instead of 10 min. Santolamazza et al. [13] highlighted that SCADA is not currently effective to prevent critical scenarios as it often does not allow sufficient time to plan interventions. Therefore, Santolamazza et al. [13] applied feed-forward neural network (FFNN) and statistical process control (SPC) on SCADA data and tracked the generator bearing temperature and generator slip ring temperature. Yang and Zhang [14] applied deep learning algorithm on specific SCADA data: gearbox
oil temperature, and main bearing rotor side temperature. It was reported that this algorithm provides a precision predication rate of 97% and a false-positive prediction rate of less than 1%.

In summary, most of the contributions have applied single machine learning algorithms and it is hard to compare between them and decide which ML algorithm is most useful to analyse SCADA data for fault detection purposes. The performance of prediction algorithms can be evaluated by the True-Positive, False-Positive, True-Negative, False-Negative rates. Thus, there is a need for handy and robust tools to perform several algorithms at the same time and get their performance rates. Moreover, the data sets vary with their contents, the data set used in Letzgus [11] is quite illustrative even before pre-processing, while the data set used in Santolamazza et al. [13] contains high fluctuations and need several pre-processing steps. Furthermore, several ML algorithms are still not applied, e.g., boosting algorithms, on wind turbine SCADA data to explore their potentials.

The purpose of this paper is to explore the potential benefits of SCADA data for maintenance purposes and discuss the limitations from a machine learning perspective. Authors will apply common ML algorithms that have not been applied yet on SCADA data, where the Azure ML learning studio as a professional platform is used. In this paper, a two-year SCADA data related to a wind turbine generator is extracted and analysed using several machine learning (ML) algorithms: two-class boosted decision tree, two-class decision forest, k-means clustering.

In the following section, the extract SCADA data is described, the analysis methodology and associated Azure ML roadmap is represented, and the theories behind the applied machine learning algorithms are briefly presented. Then, a results and discussion section will follow to present the outcomes of each applied machine learning algorithm and discuss their accuracy, validity, and limitations. Finally, some conclusion and recommendations for further work are drawn up.

2. Methods and theory

In this section, the analysis methodology is presented in five stages, as shown in Figure 1. The first stage of analysis is about preparing multiple data files of SCADA data and uploading them into Azure ML learning studio. The second stage is to visualize the collected SCADA data with a purpose to identify the correlation (between parameters) and fluctuation (of each parameter over time) data patterns. Correlation helps to select the most important parameter(s) for further analysis. The third stage of analysis is the data pre-processing data cleaning, integrating data (signals file with logs file), dimension reduction, and feature selection are applied. The fourth stage of analysis is where you apply the selected machine learning algorithms. In this paper, we are applying: (1) two-class boosted decision tree with both PCA and feature selection methods, (2) two-class decision forest with both normal training model and tune hyperparameters training model, and (3) k-means clustering. Authors decided to try two supervised learning algorithms, i.e., boosting and decision forest, and one unsupervised, i.e., clustering with K-means. The fifth stage is about testing and evaluating the classification and clustering results. This five-stage methodology is all performed in Azure ML-learning studio.

![Figure 1. SCADA Data Analysis Methodology on Azure ML.](image)

2.1. Datasets

Datasets are the most important part of modern machine learning applications. The quality of a dataset is directly related to the organization that creates it, and data quality is often related to its value and accuracy. However, data quality has other dimensions, such as uniqueness, completeness, validity, and consistency. The dataset consists of different files that give information about failure logs and technical
information about some of the main turbine’s components, such as the gearbox, generator, and rotor. Additional information includes meteorological data, namely wind speed and direction, air pressure, humidity, temperature, and component signals, namely generator RPMs and oil temperature in the hydraulic group [15].

**Table 1. Wind turbine SCADA data.**

| Data File                     | Variables                                                                 | #Data Points (Rows) |
|-------------------------------|---------------------------------------------------------------------------|---------------------|
| Wind Turbine Characteristics  | Power; Rotor; Gearbox; Generator; Tower; Power Curve                      | N/A                 |
| Meteorological (Metmast)     | Wind Speed and direction (2 anemometer sensors); Ambient Temperature and Air Pressure (2 sensors); Humidity; Precipitation | 87528               |
| Component Signals            | Generator RPM and Temp; Gearbox Oil Temp; Nacelle Temp; Total active and Reactive Power; Pitch Angle | 102921              |
| Logs                          | Historical Logs                                                          | 55699               |
| Failures                      | All failures                                                              | 7                   |

The Wind Turbine Characteristics file contains wind turbine main characteristics. It supplies the wind turbine’s power curve [15]. The meteorological mast file logs important meteorological signals, namely: Anemometer sensors 1 and 2; temperature and pressure sensors. The component signals file includes SCADA signals for each wind turbine’s most important components and production values. The failure logs file is a historical failure logbook for the wind farm. It logs replacement and repair processes, errors, high signal values, and component failures.

In this paper, the authors have used a SCADA data set from an onshore wind turbine located in southern Europe. We got four types of files: (1) Signals file, (2) Logs file, (3) Failure logs, and Meteorological mast files. These four types of data files cover almost two-year timeline from 01-01-2016 to 31-12-2017. In total, the data rows in Signals file are 102921, Logs file are 55699, Failure file are 07 and Meteorological file are 87528 respectively.

2.2. **Data visualization**

Before starting with model training and tuning, we had to explore the data precisely, whereas, data exploration and visualization guides the way on how data pre-processing is performed, and that is vital for the ML algorithm for many reasons such as each model requires specific input data; none sense data will produce non-sense results as widely known as “Garbage In, Garbage Out (GIGO)”; another reason is the quality of the data must be considered for less training time and better accuracy.

It is important to understand the relation between the features in the dataset [16]. Correlation explains how one or more variables are related to each other. These variables can be input data features that have been used to forecast our target variable [17]. Hence, parameter selection is important for modelling a wind turbine’s condition.

Heatmap is a helpful tool to visualize and show the correlation among parameters in a dataset. It assists the audience towards the areas that matter the most when you have a large volume of data. Heatmaps utilize dark color codes such as dark green for highly correlated features and dark red for negative correlation and lighter color tone for lightly correlated parameters. Dataset features can be positively correlated which means that when the value of one variable increases then the value of the other variable(s) also increases. It can be also negatively correlated or not correlated at all. It is also important to mention that if two variables are strongly correlated to each other, then we only need to use one of them by doing so resulting model will be simpler, and simpler models are easier to interpret [18]. On the other hand, uncorrelated variables are probably good candidates to trim from the model for better results.
2.3. Data pre-processing

Data pre-processing is about data cleaning, normalization and transformation, data integration, and data reduction tasks. The result expected after reliable chaining of data pre-processing tasks is a final dataset, which can be considered correct and useful for further data mining algorithms García, et al. [19]. The applied pre-processing steps are organised in the Azure ML Learning Studio, as shown in Figure 2. It started with ‘Add Rows’ to combines rows from several datasets for the same data type. Then, the “clean missing data” block is used to remove missing rows, columns, or even records.

![Figure 2. The applied data pre-processing on Azure ML learning studio.](image_url)

Multiple processes have been performed to prepare the data for the modelling and tuning stage, these processes include data integration, cleaning, transformation, and reduction. Elaboration on the outcomes of those processes comes in the following paragraphs. Data Integration was needed for two reasons, the first reason is to combine dataset parts because the original source has been split into two .csv files as a preparation step which is not valid in our case and that process has been done using ‘Add Rows’ block for both Turbine Signals and Turbine Logs datasets; the other reason is to merge the aforementioned datasets into a single dataset which has been done using ‘Join Data’ block taking into account the joining method in order to optimally retain both datasets contents.

Data Cleaning as an essential and common step has been done to first remove unnecessary columns using the block ‘Select Columns in Dataset’ reducing the number of columns of Turbine Signals dataset from 81 to 36 based on the data visualisation and heat maps presentation of the subsection 3.1; secondly, to remove duplicate records in Turbine Logs dataset that are happening in the same timespan (within 10 minutes) using the block ‘Remove Duplicate Rows’; thirdly, to handle missing values using the block ‘Clean Missing Data’ taking into account the substitute method that fits the nature of the parameter (dimension); fourthly, to eliminate errors and noise that have been noticed from data exploration stage which are believed to be called as outliers and they are abnormal temperature values appeared in several parameters such as ‘Gen_Bear_Temp_Avg’ and ‘Gen_SlipRing_Temp_Avg’ due to sensors dysfunction and thanks to the block ‘Clip Values’, this issue has been resolved by clipping the peaks that normally exist in the 99 percentile of the data.

Data Transformation for Turbine Logs dataset that includes more than 3600 unique string values was the clue for creating the label column by connecting SCADA Logs with failure event in such a way that transform a log that could relate to a generator failure such as “High Temp..” or “Hot Gen…” into the numeric value ‘1’, otherwise, it will be ‘0’ for the unrelated logs, and that process has been done using the block ‘Apply SQL Transformation’ on Turbine Logs dataset.
Features selection and dimension reduction: When the dataset has too many variables, which is good in one sense that the model will have high accuracy on the training datasets but when it comes to prediction, then it will result in wrong prediction. In other words, overfitting can be avoided by reducing the model to only important variables. It is important to mention, that it is not necessary to include all correlated variables in the model which will result in overfitting. Only select the one of them by doing so resulting model will be simpler, and simpler models are easier to interpret. To investigate the statistical relationships among the parameters, Feature Selection in Azure Machine Learning will allow us to select the important variables, or it will not wrong to say that Filter Based Feature Selection is basically a method of selecting highly correlated variables in the dataset [20].

The collected SCADA data and used in this paper has over 84 parameters, where some parameters are not relevant for fault detection, and some are redundant. Thus, the authors decided to reduce the dimensions of the collected data and select the most relevant features. Two algorithms were applied: filter-based feature selection and Principal component analysis (PCA). PCA is often used to extract useful and non-redundant information from the data and is used to reduce data components to handle attributes from many the dimensions.

2.4. Machine learning algorithms for classification and clustering

Machine learning is developing rapidly every day, which is difficult for engineers from different backgrounds to cope with, however, it is needed in today's businesses as it is proved that it can solve many industrial challenges from new perspectives. Azure ML Studio is making life easier for engineers to be able to add machine learning into their toolbox. And that is what we have seen from the beginning of the discussion section and will continue to realize how handy is Azure ML when it comes to implementing different modelling methods in a single experiment.

In machine learning, there are three approaches: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning uses labelled training dataset (correct answers) to predict outcomes, while the other does not [19]. In unsupervised learning, the algorithm self-discovers the patterns or rules in the training data without any given pre-assigned labels. Clustering is an unsupervised technique that is used to divide the population or data points into several groups. Reinforcement learning uses an interactive environment of trial and error, a game-like situation, where the learner agent gets feedback (reward/penalty).

In this paper, two supervised learning algorithms, i.e., boosting and decision forest, and one unsupervised, i.e., clustering with K-means are applied. In total, six machine learning experiments have been designed and modelled into Azure ML learning studio, as shown in figure 3.
The performed machine learning experiments are summarized in Table 2. The experiments differ in three main issues: Classification/Clustering algorithm, Feature selection/reduction algorithm, and the Training model.

### Table 2. The performed machine learning experiments

| #  | Classification/Clustering algorithm       | Dimension Reduction | Training model                          |
|----|------------------------------------------|---------------------|----------------------------------------|
| 1  | Two-class Boosted decision tree          | Feature Selection   | Train model                            |
| 2  | Two-class Boosted decision tree          | PCA                 | Train model                            |
| 3  | Two-class decision forest                | Feature Selection   | Train model                            |
| 4  | Two-class decision forest                | PCA                 | Train model                            |
| 5  | Two-class decision forest                | PCA                 | Tune Model hyperparameters             |
| 6  | K-means clustering                       | PCA                 | Train clustering model                  |

2.4.1. Machine learning algorithm theory

Azure ML Studio has decision-tree-based algorithms for two-class classification: The Two-Class Decision Forest, Two-Class Boosted Decision Tree, and Two-Class Decision Jungle modules. These models are ensemble in nature. In ensemble learning the algorithm generate the group of base learners and combines the results which give higher accuracy. Different base learners can use different parameters, sequence, and training sets. There are two major ensemble learning methods - Bagging -
Boosting. In bagging, various decision tree models are built-in parallel and all models give the vote for the final prediction. However, in boosting, the decision tree is trained in sequence and the algorithm learns from the previous tree by focusing on incorrect observations and the new model is built with higher weight for incorrect observations from the previous sequence. Hence, the decision in getting improvised in each sequence. At the end when algorithm identifies all the observation correctly then it will combine or ensemble what it learns. It is also important to mention that these models use decision tree as a methodology, but they differ in their underlying algorithms.

**Two-class boosted decision tree**: This method creates a machine learning model that is based on the boosted decision trees algorithm. In this method, each tree is dependent on prior trees and predictions are based on the entire ensemble of trees together that makes the prediction [21].

**A two-class decision forest**: Decision forest is a fast supervised ensemble model which provides better coverage and accuracy than single decision trees. The decision forest works by building multiple decision trees and then voting on the most weighted output class occur for the result. This algorithm is a good choice in case we want to predict a target with a maximum of two outcomes [21].

Clustering is an unsupervised learning method. It is the process of dividing the entire dataset into groups (known as clusters) based on the patterns in the dataset. Data points from different clusters should be as different from each other as possible to have more meaningful clusters and data within cluster shares common properties. K-Means Clustering is a simple yet powerful algorithm where algorithm groups objects based on their feature values into K disjoint clusters. K is a positive integer number specifying the number of clusters. Now the question is why we have used clustering? There are two reasons 1) For data segregation based on data similarities and 2) To detect anomalies. In our case, we have selected clustering for anomalies detection or in other words the failures data pattern in our dataset. For example, during cluster analysis if we observe that a particular cluster or certain number of clusters have a fraction number of overall observations, it may suggest or give an indication of the anomaly within this data.

It is important to understand the evaluation model in order to draw up a good conclusion on the algorithm performance. There are eight performance indicators to evaluate any classification algorithm, as shown in Figure 4. The True Positive (TP) indicator indicates that the predicted values are correctly predicted as actual positive and False Positive (FP) indicates that negative values are predicted as positive. On the other hand, the False Negative (FN) indicates that positive values predicted as negative and True Negative (TN) where predicted values correctly predicted as an actual negative. The accuracy indicator shows the percentage of how often the classifier is correct. The Precision metric is a measure for the correctness of a positive prediction means when it predicts yes, how often it is correct. It is important to mention that Precision metric alone is not very helpful because it ignores negative class. However, Recall (or the true positive rate) also called as sensitivity metric is the measure for how many true positives get predicted out of all the positives in the dataset. F-score is a way to measure a model’s accuracy based on recall and precision. The higher an F-score, the more accurate a model is and lower the F-score model is less accurate. The F1 Score is the most used F-score, which is a weighted average score of the true positive (recall) and precision. Area Under Curve (AUC) shows the true positive rates against the false positive rate at various cut points. The higher the AUC the better the ability of the model at distinguishing between positive and negative classes. When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. When AUC lies in 0.5<AUC<1, then the ability of the classifier to distinguish between Positive and Negative classes is from moderate to high. However, when AUC = 0.5, then the classifier is not able to distinguish between Positive and Negative class points [22].
There are some challenges to measure the quality of the results provided by a clustering algorithm [23]. Contrary to supervised learning where we have ample number of performance measure to evaluate the model’s performance, clustering or unsupervised analysis on the other hand does not have a solid evaluation metric. Clusters are evaluated based on some similarity or dissimilarity measure such as distance between the cluster points. We can say that clustering algorithm has performed well when algorithm separates dissimilar observations apart and similar observations together. Hence, good cluster analysis is, when (1) the observations in the cluster (same group) share similar characteristics and (2) all cluster should have proportionate number of observations (3) determining the correct number of clusters [23]. In k-means clustering k is used as an input and the optimal value of k is important for model evaluation. There are two metrics which may give us some intuition about k such as (1) Elbow Method – It aiming to help find the appropriate number of clusters in the dataset [24] (2) Silhouette Score – It is used to measure the separation distance between the clusters.

3. Results and discussions
In this section, the results of data visualization (correlation heatmap), filter-based feature selection, principal component analysis, and the performed six machine learning experiments are presented and discussed.

3.1. Data exploration and visualization
The correlation analysis is performed and the heatmaps are generated as shown in Figure 5. The correlation values are graphically represented and colored to express the strength of the correlation. The dark blue represents positive correlation, while dark red represents negative correlation. As you can see in Figure 5 highlighted with green box, parameters such as “Gen_Bear_Temp”, “Gen_Phase1_Temp”, “Gen_Phase2_Temp” and “Gen_Phase3_Temp” has positive correlation with “Amp_Windspeed” but the same parameters have negative correlation with “Grd_Prod_Psblend_Avg”, “Grd_Prod_Psblelnd_Max” and “Nac_Direction_Avg” highlighted with red.

In addition to the heatmaps, manual exploration and visualization on the data have been done in order to map out the data pre-processing stage by looking into details of each dataset such as parameters pattern, noise, type, and quality. As we can see from Figure 6 which is the timestamp visualization of selected parameters of the collected SCADA data.

And the key findings of data visualization can be summarized as follows:

- Heat maps show that the Turbine Signals dataset is more representative compared to the Metmast dataset.
- The Turbine Signals dataset is relatively huge as it includes 81 dimensions (parameters).
- The Turbine Signals dataset includes outliers and missing values
- Turbine Logs dataset has more than 3600 unique string values in the Remarks column, needed to be transformed into a reasonable label for the supervised techniques discussed earlier.
Figure 5. Heat maps of Turbine Signals data

Figure 6. Timestamp visualization of selected SCADA parameters
3.2. Data pre-processing
The result of data reduction using ‘Filter Based Feature Selection’ is shown in Figure 7, where the data dimensions have been reduced from 36 to 7. The result of PCA is presented in Figure 8 where the data dimensions have been reduced from 36 to 7.

![Figure 7](image1.png)

**Figure 7.** The results of the feature selection Algorithm, Snapshot from Azure ML.

![Figure 8](image2.png)

**Figure 8.** The results of PCA algorithm, Snapshot from Azure ML.

3.3. Results of machine learning experiments
The six machine learning experiments are performed, and the results are summarised in Table 3. The accuracy, precision, AUC, true-positive rate, false-positive rate, true-negative rate, and false-negative rate are provided and compared.

| #  | ML Algorithm                          | Accuracy | Precision | AUC  | True +ve | False +ve | True -ve | False -ve |
|----|--------------------------------------|----------|-----------|------|----------|-----------|----------|-----------|
| 1  | Two-class Boosted decision tree, Filter based feature selection | 0.999    | 0.125     | 0.805 | 1        | 7         | 20564    | 12        |
| 2  | Two-class Boosted decision tree, PCA  | 0.999    | 0.200     | 0.882 | 1        | 4         | 20567    | 12        |
| 3  | Two-class decision forest, Filter based feature selection | 0.999    | 1.000     | 0.653 | 1        | 0         | 20571    | 12        |
| 4  | Two-class decision forest, PCA       | 0.999    | 0.333     | 0.768 | 1        | 2         | 20569    | 12        |
| 5  | Two-class decision forest, PCA, Tune Model hyperparameters | 0.999    | 1.000     | 0.877 | 1        | 0         | 20571    | 12        |
| 6  | K-means clustering, PCA              | N/A      | N/A       | N/A  | N/A      | N/A       | N/A      | N/A       |

3.3.1. Boosted decision tree vs. decision forest with feature selection. Figures 9 a) and b) show a comparison of the results of the two classification methods (Boosted Decision Tree and Decision Forest) based on Feature Selection.
It is noticed that with feature selection, the evaluation of the two classification methods is different in some parameters and similar in others. As we can see from Figures 9a and 9b, both algorithms predicted 1 failure which is referred to as the True Positive, and they have the same value for False Negative, which is 12, but when it comes to False Positive, Decision Forest is better as the False Positive indicator is 0 and that reflects on the Precision of the algorithm which is higher in Decision Forest. The two algorithms have the same accuracy because of the relatively huge value of the True Negative indicator, however, the AUC is higher in the case of Boosted Decision Tree.

### 3.3.2. Boosted decision tree vs. decision forest with feature selection.

Similar to the representation of the previous subsection, Figures 10a) and b) show a comparison of the results of the two classification methods (Boosted Decision Tree and Decision Forest) based on Principal Component Analysis.

**Figure 9**
- a) Boosted Decision Tree (Filter based), Snapshot from Azure ML
- b) Decision Forest (Filter based), Snapshot from Azure ML

**Figure 10**
- a) Boosted Decision Tree with PCA, Snapshot from Azure ML
- b) Decision Forest with PCA, Snapshot from Azure ML
With PCA, the evaluation of both algorithms is somehow different as shown in Figures 10a and 10b. Since the precision of both algorithms is quite low and that can be deduced from the relation between the True Positive with the predicted positive (total positive), in other words, here the False Positive indicator controls the Precision indicator as both algorithms predicted the same number of failures which is 1 referred to the True Positive. And for the same reason discussed earlier, the accuracy is high in both algorithms.

3.3.3. Effect of tune model hyperparameters. Figures 11 a) and b) show the evaluation of both Two-Class Decision Forest, one normally trained and second trained with tune model hyperparameters. It can be noticed that there is a relative advantage using tune model hyper-parameters over the default trained model as the plot on the right tends to be more ideal than the plot on the left. Moreover, it can be concluded from the Area Under the Curve (AUC) tune hyper-parameters produces a bigger AUC than the experiment performed with the normal train model. In fact, this is due to the false positive figures. However, the tune model hyper-parameters technique has no improvement on the “True Positive” values (it is still 1, which means no more faults are detected).

Figure 11. a) Decision Forest with normal train model, Snapshot from Azure ML b) Decision Forest with tune model hyperparameters, Snapshot from Azure ML

3.3.4. K-means clustering. The results of experiment 6 where K-means clustering algorithm is applied, have been shown in Figures 12 and 13. The algorithm could cluster the SCADA data into 4 to 8 clusters, where it is observed that even different k values have hardly any negligible impact on cluster formation. The clusters are quite overlapping. There is no segregation among the data points, and it is hard to detect anomalies in the dataset. The data is quite overlapping or in other words, data is flat. There is no segregation among the data points, and it is hard to detect anomalies in the dataset.
4. Conclusions
The purpose of this paper is to explore the potential benefits of SCADA data for maintenance purposes and discuss the limitations from a machine learning perspective. The exploration was done over a five-step methodology. It can be concluded based on the six machine learning experiments that SCADA data has limited benefits for fault detection purposes due to several issues.

The first issue is related to the measured parameter and their coverage or contain of fault symptoms. For example, Generator bearing temperature parameter might be considered as a symptom of the generator bearing fault. However, the temperature is generally considered as a late-fault indicator and influence by several operating conditions (ambient temperature, oil temperature) that are not related to the fault mechanism. Symptoms like vibration, acoustic emission, and oil debris might provide earlier fault indication, especially for fatigue and wear failure mechanisms. Thus, the diagnostic coverage of the SCADA system shall be studied and there might be needed to have a condition monitoring system (CMS). The benefit of the SCADA system, once the CM system is used, a correlation between fault-related parameter like vibration and the environmental loading parameters like wind speed can be
determined and used for prediction purposes. A combined SCADA and CMS might enhance unsupervised learning.

The second issue is related to the synchronising between the signals data and logs data. Industry shall target to have an automatic system to synchronise the log description for each abnormal signal data point, to ensure the same timestamp without time-delay or missing input. Some of the logs, analysed in this paper, have been written almost one week after the issue is notified by the signal data. Such an issue makes the supervised learning unreliable as labels are generated at totally incorrect timestamps.

The third issue is related to data richness and whether the SCADA data is learnable. In a failure prediction context, data richness means ensuring that fault features are presented in the training data. For example, out of 102,921 data rows, only 61 were labelled as faulty/abnormal data and the rest is somehow considered as healthy or normal data. This makes the algorithms learn very well how to predict healthy data, but not the faulty ones.

References

[1]. Kim, K Parthasarathy, G Uluyol, O Foslien W Sheng, S and Fleming, P. 2011 Use of SCADA data for failure detection in wind turbines. Proc. Series Energy Sustainability, ASME 5th Int. Conf. on Energy Sustainability, Washington DC, Aug. 7-10.
[2]. Maldonado-Correa J, Martin-Martinez S, Artigao E, Gomez-Lazaro E. 2020 Using SCADA Data for wind turbine condition monitoring: A systematic literature review, Energies 13(12), 3132.
[3]. Astolfi D 2021 Perspectives on SCADA data analysis methods for multivariate wind turbine power curve modeling, Machines 9(5), 100.
[4]. Wang, KS Sharma, V S & Zhang, ZY. 2014 SCADA data-based condition monitoring of wind turbines, Adv. Manuf 2(1), 61-9.
[5]. Tautz-Weinert, J and S J Watson 2017 Using SCADA data for wind turbine condition monitoring—a review, JET Renewable Power Gener 11(4), 382-94.
[6]. Zhang S and Z-Q Lang 2020 SCADA-data-based wind turbine fault detection: A dynamic model sensor method, Control Eng Pract 102, 104546.
[7]. Artigao, E Martin-Martinez, S Honrubia-Escribano A and Gomez-Lázaro E 2018 Wind turbine reliability: A comprehensive review towards effective condition monitoring development, Appl Energy, 228, 1569-83.
[8]. Hameed, Z Hong, Y S Cho, Y M Ahn H and Song, C K 2009 Condition monitoring and fault detection of wind turbines and related algorithms: A review, Renewable Sustainable Energy Rev 13(1), 1-39.
[9]. Tao L Q Siqi Y Zhang and H Shi 2019 Abnormal detection of wind turbine based on SCADA data mining, Math Probl Eng 2019, 5976843.
[10]. Lima L, Blatt A and Fujise J 2020 Wind turbine failure prediction using SCADA data, J Phys, Conf Series 1618, 022017.
[11]. Letzgus S 2020 Change-point detection in wind turbine SCADA data for robust condition monitoring with normal behaviour models, Wind Energ. Sci. 5(4), 1375-97.
[12]. Castellani F, Astolfi D and Natili F 2021 SCADA data analysis methods for diagnosis of electrical faults to wind turbine generators, Appl Sci 11(8), 3307.
[13]. Santolamazza A, Dadi D and Introna V 2021 A data-mining approach for wind turbine fault detection based on SCADA data analysis using artificial neural networks, Energies 14(7), 10.3390/en14071845
[14]. Yang L and Z Zhang 2021 Wind turbine gearbox failure detection based on SCADA Data: A deep learning-based approach. IEEE Trans Instrum Meas 70, 1-11.
[15]. Menezes D Mendes M Almeida JA Farinha T 2020 Wind farm and resource datasets: A comprehensive survey and overview, Energies 13(18), 4702.
[16]. Anderson, C 2019 Hot or Not? Heatmaps and Correlation Matrices. Available at: https://medium.com/@connor.anderson_42477/hot-or-not-heatmaps-and-correlation-matrix-plots-940088fa2806
[17]. Upadhyay, A 2020 What Is Correlation in Machine Learning? Available at: https://medium.com/analytics-vidhya/what-is-correlation-4fe0c6fbed47
[18]. Looking for Connections in Your Data – Correlation Coefficients – The Official Blog of BigML.com, Dated 13.09.2021.

[19]. García, S Luengo, J Herrera, F 2015 Data preprocessing in data mining, Intelligent Systems Reference Library. Springer, Doi: 10.1007/978-3-319-10247-4

[20]. Feature Selection in Azure Machine Learning (sqlshack.com), Dated 13.09.2021

[21]. Docs M 2020 Two-Class Decision Forest module, Available at: https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/two-class-decision-forest.

[22]. Bhandari A 2020 AUC-ROC Curve in Machine Learning Clearly Explained Retrieved 27.09.2021, Available at: https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/.

[23]. Palacio-Niño, J-O and F Berzal 2019 Evaluation metrics for unsupervised learning algorithms, Technical Report, Cornell University, https://arxiv.org/abs/1905.05667.

[24]. Liu, F and Y Deng 2020 Determine the number of unknown targets in Open World based on Elbow method, IEEE Trans Fuzzy Syst 29(5), 986-995.