Wind Turbine Failure Prediction Using SCADA Data

L A M Lima¹², A Blatt¹ and J Fujise¹
¹ Voltalia - Rua do Passeio 78, 14º andar, Centro, 20021-290, Rio de Janeiro, RJ, Brazil
² Electrical Engineering Program, COPPE-UFRJ, Cidade Universitaria, Centro de Tecnologia, bloco H, 21941-972, Rio de Janeiro, RJ, Brazil
E-mail: l.lima@voltalia.com

Abstract. This paper proposes a failure prediction system for wind turbines using the Normal Behavior Model (NBM) approach. By using available SCADA data, the NBMs are trained to make predictions that reflect what would be a turbine’s normal operating condition. They are able to identify when a given operating condition is abnormal, which points towards probable component degradation. Alerts are raised based on the daily-averaged prediction error to help the O&M team in identifying turbines that need maintenance. The NBMs are comprised of numerous linear models with different inputs and training sets, according to an ensemble approach that aims to avoid overfitting and to reduce the amount of false-positive predictions.

Description and insights on various development steps are presented, such as data treatment, model selection, error calculation and alerts generations. Two test cases are shown using operational data from existing wind turbines, highlighting the system’s ability to generate alerts weeks before a severe fault occurs.

1. Introduction
The large size and remote location of many wind farms puts pressure on activities related to monitoring and maintenance, since availability is an important factor for any power plant. The turbine data acquisition system (SCADA) gathers a large amount of data from dozens of sensors, which can be used in different ways [1] besides monitoring the base performance of the turbine [2]. One of the main applications of SCADA data analysis is fault prediction [3]. Faults on smaller components, such as pumps or fans, could lead to unexpected downtime, production loss or other faults. A severe failure of a bigger component, such as the generator, could result in days of lost production and a high maintenance cost. Therefore, being able to identify when a component is showing signs of upcoming failure is important to schedule a predictive maintenance on the best chosen time slot, reducing turbine downtime and maintenance costs.

There are many ways of predicting a component fault. A physical model could be used, but the complex interaction between the various turbine components is difficult to model accurately. Other approaches could be used, such as digital twins or finite-element based models, but their computational cost may still be too high, which could make online monitoring unfeasible. A model based on SCADA data analysis [4, 5, 6, 7, 8], however, yields a good representation of the system and has a lower computational burden. A large amount of operation data is available from various turbine components (temperatures, wind speed and direction, currents and voltages, vibration...), consisting on rich information about the turbine itself. This data can be used to generate models that detect a faulty behavior trend days, weeks in advance of...
a potential fault itself, and the Operation & Maintenance (O&M) team can then schedule a predictive maintenance accordingly.

A failure detection system called TCHALA (Turbine Condition, Health, and Alerts Learning Algorithm) was implemented at Voltalia to aid the O&M team in monitoring the company’s windfarms. This paper presents some results, details and insights on the chosen methods.

2. Objectives
The main objective of the implemented fault detection system is to correctly identify an upcoming fault on a turbine component, based only on available SCADA data. This information is used by the O&M team to optimize the maintenance scheduling, reducing turbine downtime, production loss and maintenance costs.

To achieve that goal, the system is comprised of different modules, mainly to perform data treatment, train the machine learning models, make predictions, identify faulty trends and finally generate alerts.

3. Methodology
The SCADA system has historical data for numerous measurement channels of the wind turbine (e.g. generated power, rotor speed, generator temperature...). This approach is based on Normal Behavior Models (NBM) [4], which are machine learning algorithms that use past data to learn what the turbine’s normal operating condition should be. An individual NBM is created for each measurement channel available for the turbine. It makes a prediction of what the channel’s normal behavior should be at a given time instant, and this prediction is compared to the actual measured value. If there is a large difference, it means the measured value is outside the normal operating range, and a potential fault may be building up. The alerts generation module analyzes how much the measured value is deviating from the predicted normal behavior and then generates alerts to tell the operator something might be wrong.

It is important to note that, while the NBMs make predictions for the turbine channels, these are predictions of how they should be behaving at a specific time instant given the available measurements. The models are not used to make predictions for future days or weeks (forecasting), since this would require different algorithms, analyses, and would be in general a completely different problem [9]. A schematic block diagram is shown in Figure 1, showing generally how a prediction error is calculated for a given channel. More detail about the individual models comprised in the NBMs are presented in the following sections.

![Figure 1: Simple block diagram illustrating the principle of the failure prediction system. Each NBM uses other channels to make a prediction about a target channel (in this example, channel 42), and the prediction is compared to the actual measured value.](image-url)
The overall system is comprised of two main modules: training and prediction.

3.1. Training Module
The training module encompasses all steps needed for the complete setup of the failure prediction system. After it is done, the system is able to make predictions, calculate error and generate alerts for the operator. It does not need to be run everyday; it is usually run once every 2 weeks to update the models’ training with more available data.

3.1.1. Data Treatment In order to generate predictions and alerts, first it is necessary to train the NBMs, and to do this, the SCADA data must be treated. If the models are trained using input data which contains wrong values, missing data or such, their prediction will be less accurate. Usually, the SCADA data follows a 10-minute frequency for all measurement channels. Sample data points are shown in Table 1.

| Timestamp      | OilTmp | ShaftTmp | AmbTmp | BearingTmp | AmbTmp(t-1) | ... |
|----------------|--------|----------|--------|------------|-------------|-----|
| 01-09-2019 14:10:00 | 35.2   | 37.4     | 29.5   | 36.8       | 29.2        | ... |
| 01-09-2019 14:20:00 | 36.1   | 37.6     | 28.9   | NaN        | 29.5        | ... |
| 01-09-2019 14:20:00 | 35.8   | 37.0     | 80000  | 37.1       | 28.9        | ... |

The implemented data treatment steps are:

- Missing data removal
- Duplicate data removal
- Maintenance dates removal
- Abnormal operating condition data points removal
- Outlier removal
- Constant channels removal
- Data shifting
- Normalization
- Training / validation split

It is very common to have missing data due to communication or other data acquisition issues. If a given data point contains missing values and is given as input for the training algorithm, the model being trained will not have full visibility of the turbine condition at that time, so it could learn the turbine behavior in a wrong or inaccurate fashion. Therefore, any data point that has any missing values is discarded. The second data point shown in Table 1 would be discarded according to this criteria. There are approaches that artificially fill these missing values to avoid discarding the entire data point but, when there is a large dataset available, this is usually not a problem. Similarly, the data acquisition system could generate duplicate entries that also need to be removed.

Since the objective is for the NBMs to learn the normal turbine behavior, any datapoints generated when the turbine is under maintenance are removed. Here, an additional step is taken, and all datapoints 7 days prior to a maintenance date are also removed. When maintenance is performed, it can be assumed that a problem was occurring with the turbine and some component
needed to be replaced due to failure or degradation. Such component probably deteriorated until it got to an unsustainable situation, so the turbine could have spent some time operating with the abnormal component. The removal of some days worth of data prior to a maintenance date is aimed at maximizing the amount of data that represents the turbine operating at normal condition, with healthy components. Abnormal operating conditions, such as curtailment, ramp up/down or stop are also removed, since they clearly do not represent the turbine normal operation.

Outlier removal is applied to remove possible extreme points in the dataset that originate from data acquisition issues. These are simple numerical values orders of magnitude different from the usual measurement channels operating range, that do not have any physical meaning. An example is shown in the third data point of Table 1, which would be discarded according to this criteria. Some channels that represent statuses or flags that are constant throughout the normal operation are removed, since they do not add any information to the system.

The amount of removed data changes drastically depending on turbine-specific conditions. For turbines with reasonably stable operation, this process can lead to an amount of removed data of 5% or less. On the other hand, especially for turbines which have undergone maintenance several times, as much as 50% of the data may end up being removed. Since there are usually several months or even years worth of data, this still leads to a significant number of data points remaining, so the training process does not seem to be much affected by the removal process.

Data shifting is employed so that each data point has visibility of the channels’ past values. This is useful for building auto-regressive models [10]; since many of the channels represent physical systems that have a high inertia (kinetic or thermal), they are strongly correlated to their previous values. For example, the generator temperature does not tend to change too much from one timestamp to another due to the high amount of steel mass that results in a high thermal inertia. This way, considering the past measurement values could increase the prediction accuracy. An example is shown in the right-most column of Table 1. The data is then normalized to have zero mean and unit variance, which helps in the upcoming training process.

Finally, the data is split into a training dataset and a validation dataset. The NBM will use the training dataset to learn the turbine normal behavior, while the validation dataset is used to assess the prediction accuracy and calculate useful metrics.

3.1.2. Inputs Selection There are numerous measurement channels available to be used as prediction inputs. Therefore, it is a challenge to choose the appropriate channels to be fed to the prediction algorithm. Using every single channel as an input to the prediction models would greatly increase the overall computational cost, and it could also reduce the prediction accuracy since channels with poor correlation to the output would be included. The training dataset is used to calculate the correlation between the available inputs and the desired channel to be predicted [3], and those with highest correlation are selected. The number of inputs may vary according to the machine learning model training configuration, described in the following section. Similarly, inputs of past timestamps may be chosen or not by a given machine learning model, according to its configuration.

3.1.3. Models Creation In order to increase prediction accuracy, each NBM comprises many different models that form a committee. This is similar to an ensemble approach [11], where each model is trained with different parameters and the output is a weighted average of the individual models’ outputs. In this approach, the models are trained with different portions of the training set and have different inputs. Each model uses a random subset with between 7 and 10 of the best inputs, which were chosen based on correlation as described in the past section. For example, a given model may use inputs #1,2,4,6,7,8,9 while a different model may use inputs
This number of inputs aims at balancing accuracy and computational cost, and was chosen based on experimentation. Besides that, different model types can be used [12], such as linear regression, lasso, neural network (NN) and SVR, among others. It is expected that, by having a variety of prediction models with different characteristics, they can mitigate each other’s limitations and generate a more robust overall prediction. This is shown in Figure 1. Another advantage of using various models with different inputs is that, if a given measurement channel had communication problems and is unavailable on a given day, the models that use it as input will be inactive, while the ones that do not rely on it can still be used to make predictions.

### 3.1.4. Data Fitting and Model Selection
It was mentioned before that each model may use a different portion of the training dataset. Therefore, several partial training sets are generated and assigned to their corresponding models for training. Each of these sets has the same number of samples as the overall training set and is generated by sampling it with replacement. A grid search with cross-validation algorithm is run to find optimized model parameter values (such as, for example, the number of layers of a NN), and then the models are fit to their training set.

After training, the models make predictions based on the validation dataset, and a prediction error is calculated (mean squared error, or MSE) by comparing the predicted values with the measured values of the target channel. Each model is assigned a score, defined as the inverse of the MSE, that represents the model’s prediction accuracy. To avoid using overfit, poorly trained or very inaccurate models in the prediction committee, which would reduce the overall accuracy, a model selection algorithm removes any model with a score lower than 70% of the best one.

### 3.1.5. Daily Prediction Error Calculation
A prediction is made on the validation dataset for each data point and compared to the measured value for the target channel, resulting in a given prediction error. It is important to highlight the meaning of this error at this point. Since the models were trained using data from the turbine operating at normal condition, it is assumed that the predictions indicate what would be the results of a normal operation condition. If the measured value deviates too much from the predicted value (i.e. high error), that data point is deemed abnormal operation, or anomaly.

Since the datapoints are acquired on a 10-min frequency, ideally 144 predictions are made in a day, and a prediction error value is calculated for each one. The problem is that there is inherent statistical error in the prediction itself. It may be due to the models not being able to perfectly represent the target channel with the given inputs, from the underlying randomness of the training process, from atmospheric conditions or from the data acquisition process itself. An approach to reduce the statistical error and increase robustness is to average all prediction error values in a day [5], resulting in one daily average value that is used as baseline for the analyses.

Since the data points are acquired on a 10-min frequency, ideally 144 predictions are made in a day, and a prediction error value is calculated for each one. The problem is that there is inherent statistical error in the prediction itself. It may be due to the models not being able to perfectly represent the target channel with the given inputs, from the underlying randomness of the training process, from atmospheric conditions or from the data acquisition process itself. An approach to reduce the statistical error and increase robustness is to average all prediction error values in a day [5], resulting in one daily average value that is used as baseline for the analyses.

It is common to have communication problems or different turbine operating conditions in a given day, so most of the time the number of treated datapoints available for a day is lower than 144. The idea of daily averages is to reduce the prediction error variance, but if the number of datapoints in a day is low, the variance will not be reduced by much. An additional penalty is introduced to avoid false positives: the daily error is multiplied by a gain (between 0 and 1) that is related to the data availability for that day. This way, when a day with few datapoints results in high error even after the application of this penalty, the chance of that being a false positive caused by the inherent process randomness is reduced.

It is also useful to normalize the daily prediction error, so future failure analyses can easily identify if a daily error is relevant or not by comparing it to the error calculation process variance. The validation dataset’s daily prediction error is calculated and its normalization parameters (mean and variance) are saved for future use.
3.1.6. Training Module Results  At the end of the training process, the system is ready to make predictions, calculate error and generate alerts. The main outcomes of the training module are the trained models and their score; the training data normalization parameters (mean and variance), used to normalize the new data when a prediction should be made; and the daily prediction error normalization parameters (also mean and variance), used to normalize new predictions.

3.2. Prediction Module
The prediction module is run regularly so the operator can have an ongoing assessment of the turbine’s situation. As it was mentioned, the calculated prediction error is averaged over a day. Therefore, the prediction module is run daily: it gathers all data for the past day, makes predictions for all datapoints, calculates a daily error value and generates a possible alert accordingly.

3.2.1. Data Treatment  When new data is gathered, the same steps as described in section 3.1.1 are performed. The only difference is that the normalization is done based on the training set normalization parameters. Since the prediction models were trained with normalized data, the same normalization parameters must be used to ensure meaningful results.

3.2.2. Committee Setup and Prediction The trained models are loaded and a committee prediction is made for each data point. It is important to note that the committee prediction weighting process is based on the model scores calculated during the training phase.

3.2.3. Error Calculation  The prediction error is calculated based on the prediction and on the actual measured values of the target channel, and averaged for the day. It is then normalized according to the error normalization parameters from the training set.

3.2.4. Alerts Generation  Based on the error value for the past days, an alert may be raised according to predefined criteria. Since the error represents the deviation of the actual measured values from the predicted values, the higher the error, the more anomalous that operating condition is. An anomalous condition is an indication that a given turbine component could be deteriorating, or a failure could be building up but has not manifested yet. A high error value, resulting in an anomalous day, is the indicator that something might be wrong. An error is classified as high if it is 3 or more standard deviations away from the mean; this indicates that there is a very low probability that it was generated by the calculation method’s inherent randomness [13], and it is significant.

To avoid false positives, an alert is only raised if a high error value occurs repeatedly on the last days. Different alert levels are defined:

- Level 3: if 5 or more of the past 6 days were identified as anomalous;
- Level 2: if 3 or 4 of the past 6 days were identified as anomalous;
- Level 1: if a statistical analysis of the past 14 days results in an average prediction error between 1 and 3 standard deviations, a fault could be just starting to build up.

3.2.5. Health Score  Each channel is assigned a daily health score. It starts at 100 and is decreased by 1/3 of the alert level whenever an alert is raised. If several consecutive days have alerts, the health score will be gradually reduced. Whenever no alerts are raised, it is incremented by 0.5. This is a straightforward way to keep track of the situation and have an idea of component degradation.
3.2.6. Prediction Module Results  The main outcomes of the prediction module, saved on an internal database, are the individual predictions and prediction error; the daily prediction error; the resulting alerts; and the health score. This information is compiled in a dashboard used by the O&M team in their regular routine.

4. Results

4.1. Model Type Comparison

One of the most important aspects of a prediction system is computational time. There is a large number of turbines, each with dozens of measurement channels to be trained and predicted. Since the designed system runs every day for making predictions and every 2 weeks for updating the training, these processes must be concluded in a reasonably fast timeframe to be able to cover all channels of all turbines.

Different types of model have very different computational costs. Table 2 shows the amount of time spent for training a given model, without including the data loading, treatment or any other part of the algorithm. It also shows the amount of time spent for a model to make a prediction. The computer used for this benchmarking has an Intel i7-755U 2.7GHz processor and 16 Gb RAM, running exclusively this program.

Table 2: Training and prediction times for different models.

| Model Type   | Training (12 months) | Training (6 months) | Prediction (1 day) |
|--------------|----------------------|---------------------|--------------------|
| Linear       | 10 ms                | 7 ms                | 2 ms               |
| Lasso        | 20 ms                | 15 ms               | 2 ms               |
| Ridge        | 10 ms                | 7 ms                | 2 ms               |
| ElasticNet   | 25 ms                | 15 ms               | 2 ms               |
| SVR          | 400 ms               | 270 ms              | 20 ms              |
| NeuralNetwork| 3500 ms              | 2500 ms             | 30 ms              |

The difference between linear and nonlinear models is noticeable. The linear models’ training time is almost negligible, which enables the use of numerous models with different training sets and inputs. A NN, on the other hand, requires detailed cross-validation to search for best parameters (such as number of layers and number of neurons) and also multiple initializations, where the weights are randomly initialized [14]. The extra amount of computational time of data loading and treatment is estimated to be around 3s per channel. Therefore, a NN-based NBM for a given channel could take around 2 minutes or more to train with 6 months of data. If the NBM was based on linear models, it could take around 5s. Considering a general count of 100 turbines with 30 measurement channels each, the total training time for NNs would be of 100 hours, and this could easily increase if more detailed grid search or weight initializations are implemented, if more channels are monitored, or if less computational power is available. As for linear models, the total amount of time would be just over 4 hours.

Besides being incredibly fast to train, the usage of linear models allows for the ensemble approach, which would be unfeasible with NNs. This is useful to reduce the probability of overfitting and also to reduce model dependency on a given input channel, since multiple combinations of inputs are employed. Therefore, the implemented NBMs use a combination of 30 linear regression and 30 lasso models (the latter, to further reduce the chance of overfitting).
4.2. Case Study - Turbine 01
On October 20th 2019, the operator noticed that alerts had been raised for some turbine channels related to components’ temperature for 16 days in a row, indicating something was probably going on. The turbine standard alarm system had not sent any kind of alarms; probably the preset temperature limits had not yet been surpassed. Maintenance was scheduled for the 25th, and a cooling fan was found to be broken. Without this system, the operator would have had no visibility of this event, and a more severe fault could have occurred due to extended overheat of the converter or generator components.

Figure 2 shows the alerts for the generator windings. X-axis represents days, y-axis shows different turbine components. Filled rectangles represent the active alert for each component on each day, where a darker color represents a higher alert level (more severe situation). It can be seen that many temperature-related channels start giving out alerts around the same time, which makes sense since a small overall increase in nacelle temperature is expected. Unrelated channels, such as blades pitch, did not send any alert since they are unrelated to the issue.

![Figure 2: Alerts over time for Turbine 01.](image)

Figure 3 shows the normalized prediction error for the converter temperature and pitch angle. It can be seen that the prediction error for the converter temperature is around zero and jumps to around 4, staying there for the next days. As seen in Figure 2, the first alert for this channel is sent on October 5th, which is the 3rd consecutive day with a high error, according to the alert logic presented in section 3.2.4. The daily prediction error for the pitch angle stays inside the threshold, not resulting in any alerts.

4.3. Case Study - Turbine 02
This case depicts a failure that occurred before the implementation of the prediction system. The standard turbine alarm system sent a warning on 24/04/19, stating “High temperature on generator winding 1”. The turbine went on maintenance, but similar warnings kept happening...
Figure 3: Normalized prediction error for the (a) converter temperature and (b) pitch angle of blade #1. The limits (±3) over which an alert may be generated are shown.

until the generator itself had to be fully replaced on 05/05/19 due to damaged windings. If the prediction system was already in place, it could have detected the overheat when it started building up, and probably the full generator replacement could be avoided. Figure 4a shows the alerts heatmap for this situation. It can be seen that the alerts start on 15/03/19, more than a month before the turbine alarm system started reacting. The fault could have been detected before it turned into something severe. To illustrate the advantages of using the ensemble approach, Figure 4b shows the same heatmap, but using only one linear model with the entire training set. It can be seen that pitch-related alerts are raised, which is actually a false-positive. The wind speed alert shows up in both cases, and is assumed to be caused by the interdependence of the inputs. The wind speed prediction model probably uses temperature channels as inputs; since these are outside the normal operating range, this affects the wind speed prediction.

Figure 4: Alerts heatmap for Turbine 02, using (a) the proposed ensemble approach with multiple models, and (b) a single-model approach.
5. Conclusions

In this paper, a fault prediction system is presented and discussed. The employed techniques allow for mining information from readily-available SCADA data, in order to detect possible anomalies as wind turbines operate. By monitoring turbine components with this system, it is possible to identify faults before their occur and act accordingly, reducing downtime and maintenance costs.

The implemented system uses an ensemble approach to make predictions, as each NBM is comprised of numerous individual models with different inputs and training sets. The prediction error is averaged over a day to mitigate inherent randomness and reduce the prediction sensibility. Alerts are raised based on the daily error values according to a predefined criteria, indicating to the operator that a potential fault is building up.

Two test cases were presented using real operational data from Voltalia’s wind turbines. Results showed that the system is sensible enough to capture measurement deviations that indicate upcoming failures, raising alerts weeks before something more severe would happen.

Condition monitoring is an ever-evolving field of study, and there are multiple ways to achieve results. Possibilities to improve the system are continuously identified and implemented, looking to improve prediction accuracy and reduce false-positives rate. Further developments will aim at comparing signals and predictions between different turbines, increasing the information pool available for the system. Different techniques will be studied to avoid models’ overfitting and reducing false-positives alerts. Besides that, the statistics to analyze daily prediction error and generate alerts will be improved to avoid time delays.

References

[1] Tautz-Weinert J and Watson S J 2017 Using SCADA data for wind turbine condition monitoring – a review IET Renewable Power Generation 11 382-94
[2] Gonzalez E, Nanos E, Seyr H, Valldecabres L, Yurusen N, Smolka U, Muskulus M and Melero J 2017 Key performance indicators for wind farm operations and maintenance Energy Procedia 137 559-70
[3] Schlechtingen M and Santos IF 2011 Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection Mech. Systems and Signal Processing 25(5) 1849-75
[4] Schlechtingen M, Santos I F and Achiche S 2013 Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description Applied Soft Computing 13(1) 259-70
[5] Schlechtingen M and Santos I F 2014 Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 2: Application examples Applied Soft Computing 14(C) 447-60
[6] Zhang Y, Li M, Dong Y and Meng K 2019 A probabilistic anomaly detection approach for data-driven wind turbine condition monitoring CSEE Journal of Power and Energy Systems 5(2) 149-58
[7] Kusiak A and Verma A 2012 A data-ming approach to monitoring wind turbines IEEE Transactions on Sustainable Energy 3(1) 150-7
[8] Pandit R and Infield D 2018 Gaussian process operational curves for wind turbine condition monitoring Energies 1 1631
[9] Colak I, Sagiroglu S and Yesilbudak M 2012 Data mining and wind power prediction: a literature review Renewable Energy 46 241-7
[10] Zaher A, McArthur S D J, Infield D G and Patel Y 2009 Online wind turbine fault detection through automated SCADA data analysis Wind Energy 12 574-93
[11] Heiermann J and Kramer O 2016 Machine learning ensembles for wind power prediction Renewable Energy 89 671-9
[12] Stetcu A, Dinmohammadi F, Zhou X, Robu V, Flynn D, Barnes M, Keane J and Nenadic G 2019 Machine learning methods for wind turbine condition monitoring: a review Renewable Energy 133 620-35
[13] Chandola V, Banerjee A, Kumar V 2009 Anomaly detection: A survey ACM Computer Surveys 41(3) 15
[14] Rafiq M Y, Bugmann G and Easterbrook D J 2001 Neural network design for engineering applications Computers & Structures 79(17) 1541-52