PROJECT REPORT

MACHINE LEARNING

Predictive Maintenance for Wind Turbines

Submitted by

Darshan Morkar

Supervised by

Prof. Dr. Patrick Levi

Ostbayerische Technische Hochschule Amberg-Weiden

Department of Electrical Engineering, Media and Computer Science

June 5, 2025

Abstract

This report explores the application of both supervised and unsupervised machine learning algorithms to real-time sensor data from wind turbines, with a focus on Wind Turbine C. The objective is to enable predictive maintenance by detecting early signs of failure, minimizing unexpected breakdowns. Supervised learning models were trained using labeled time-series data from different turbines, covering normal and faulty conditions across 95 datasets. In parallel, unsupervised methods were implemented to detect anomalies without prior labeling, allowing the system to flag unusual behavior patterns. The trained models were then evaluated on data from Wind Turbine C, which was excluded from the training phase. This approach tested the models' ability to generalize across turbines and identify failures in unseen data, improving the reliability and robustness of wind turbine monitoring systems.

1 WP1 - Data Analysis, Feature Engineering on the data set

1.1 Introduction and motivation:

The dataset is very high-dimensional with 957 features (depending on the wind farm located in Germany). The dataset is anonymized, and the overall dataset is balanced, as 44 out of the 95 datasets contain a labeled anomaly event and the other 51 datasets represent normal behavior. We have features called time_stamp, asset_id, id (which is the name given to the turbine itself), and status_type (which is the status of the wind turbine at the given time stamp). There are 6 different status IDs, stated below:

- 0 - Normal Operation - The turbine is in normal power production mode.
- 1 - Derated Operation - Derated power generation with a power restriction
- 2 - Idling - Asset is idling and waits to operate again, where the turbine is about to start considered to be normal
- 3 - Service - Asset is in service mode / service team is at the site.
- 4. Downtime - asset is down due to a fault or other reasons. This is where the turbine has stopped.
- 5 - Other - Other operational states

1.2 Motivation for Predictive Maintenance:

Wind turbines operate under different, varying environmental conditions, which makes them easily affected by sensor failures and unexpected breakdowns. These reasons might lead to huge maintenance costs. Some tests scheduled for early prediction of problems in wind turbines mostly lead to inefficiency, either performing unnecessary maintenance or failing to detect problems. Predictive maintenance helps to identify early signs of failure of sensitive sensors and helps in saving costs for unnecessary maintenance fees and also maximizes energy.

1.3 Data Cleaning and Preprocessing:

1.3.1 Handling Missing Values with Forward and Backward Fill:

In machine learning, data cleaning and preprocessing play a very important role, as SCADA systems used in wind turbines mainly have many missing values and duplicates of values, and some may be misleading informative features that are not that important for the model. to overcome missing values bfill() and ffill() can be used ffill() fills missing values using the last known value. bfill() this fills the next known value. No important data can be lost here.

1.3.2 Removing Constant Columns:

Some columns have duplicate values, which do not contribute any meaningful information to the learning of the model and usually mislead the model. nunique() is used to count those values and to remove those unwanted values The dropduplicates() built-in function has been used.

2 Related Work and Feature Engineering on the data set

2.1 Importance of Sensors and Data:

In machine learning, choosing the exact timeline where the anomaly occurred is challenging most of the time. In real time, the data points that sensors record are very precise, and to observe whether the whole turbine is in working condition or not, that has to be recorded continuously, like a 10-minute break or a 20-minute break, so it can be identified quickly by abnormal sensor values.

2.2 Analyzing Turbine Stoppage Using Sensor Data:

In this report, the visualization from dataset 35 is used for supervised and unsupervised learning. This dataset had a stoppage period of 6 days. The reason for the dataset to get an anomaly is "Turbine had several short standstills (max 8 min) with failure 'Schwingungen Umrichter Drehmomenten Level 1'" which

directly indicates that a sensor like rotor speed is affecting the wind turbine. From the below converter cabinet temperature and rotor speed sensors graph, it can be stated that the stoppage is occurring in the timeline between June 2023 and July 2023; the timestamp is being sliced for better visualization and better generalization of the model.

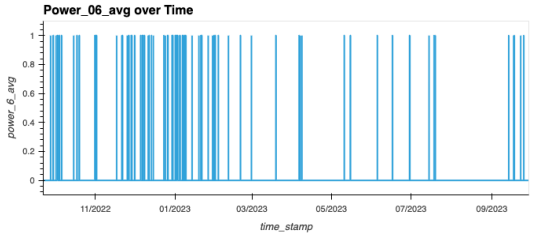

Figure 1: Visualization active power

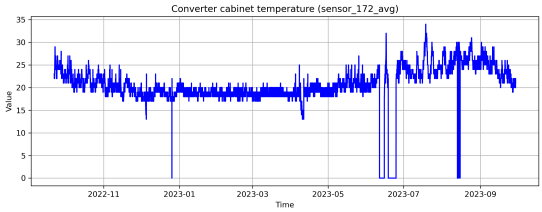

Figure 2: Visualization of Converter cabinet temperature

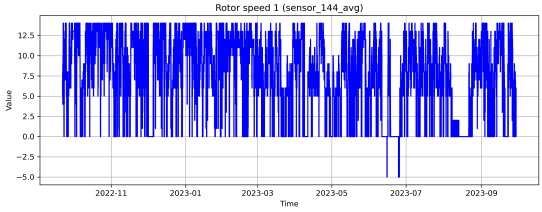

Figure 3: Visualization Rotor speed

2.3 Data Preprocessing:

In this project, extensive preprocessing was carried out for feature selection which was a major step involved in this project. The dataset included four sensor types: average, minimum, maximum, and standard deviation for each individual sensor. Here, for this project, only average, minimum, and maximum sensors were used because the temperature sensor on average, shows the general operating level, and it also help to identify overheating trends alongside; maximum shows the worst-case scenario in advance, like signaling overheating before failure. The standard deviation indicates an early sign of imbalance in

the sensor, but the minimum sensor is often close to downtime, so sometimes the minimum sensor often misleads the model even during normal operation.

2.4 Correlation Heatmap:

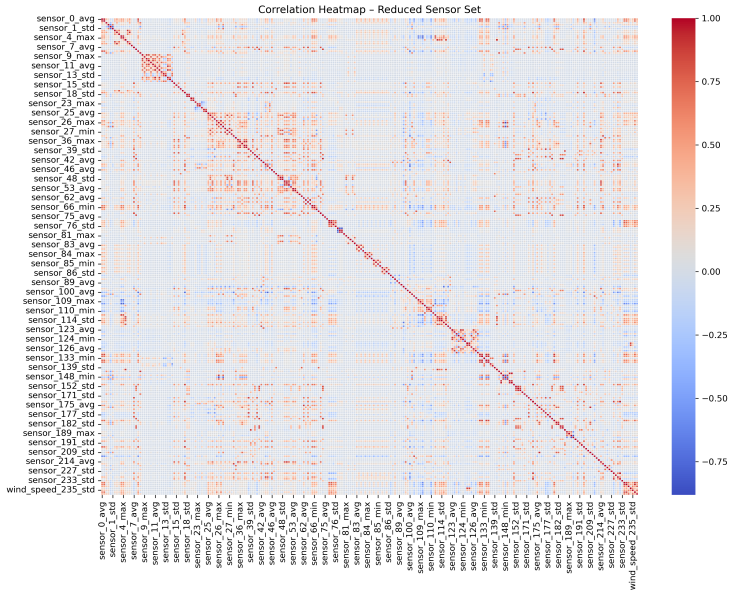

Figure 4: Visualization of reduced sensor values

2.5 Feature Reduction:

From an initial set of 957 features, a covariance heatmap shown the most redundant features and removed the highly correlated sensor because that can lead to the curse of dimensionality and due to this, the model may slow, overfit or lose generalization, so it has reduced the features to 253 by removing highly correlated ones. Further reduction was done using low variance filtering and SelectKBest with mutual information classification, helping retain the most informative features.

2.6 Domain Knowledge Integration:

After ranking features by their importance to the target variable, cross-verified those sensors with normal distribution of values using histograms and KDE for each particular sensor. The selected features were then cross-verified with domain knowledge to ensure they align with known reasons behind turbine

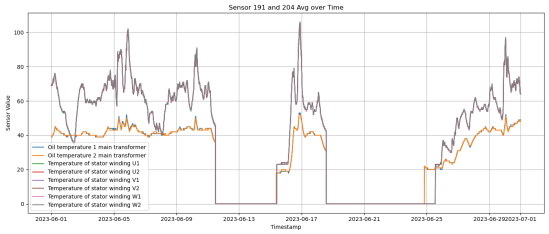

Figure 5: Visualization of temperature sensor

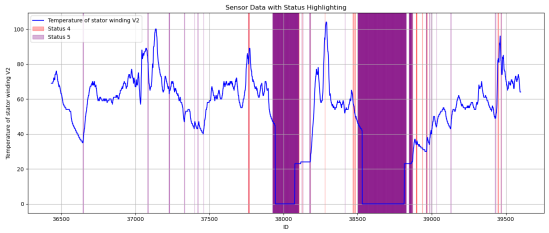

Figure 6: Visualization of temperature sensor

anomalies. This step was crucial for building a robust model that not only performs well statistically but also makes practical sense in real-world wind turbine operations.

2.7 Reason for chosen sensor:

In the visualization shown in Figure 5, there are 8 features that are selected after processing. From the graph it can be clearly seen that all sensors are going up in a certain time frame. Here, as an early prediction, the sensor values shown abnormal behaviour by reaching a certain threshold. The first five sets of sensor values have reached 100, which is the early prediction and that's causing the anomaly to occur in the coming timeline. These points are going to match when we apply the unsupervised clustering method in that prediction; the one type of cluster matches with these peak points.

The graph shown in figure 6 is of a temperature sensor. The red line indicates downtime by status ID 4, and the purple portion is the maintenance period where the turbine was in service mode. that's why this sensor has been selected, and the rest applied classification and clustering to each sensor analyzed this way.

2.8 Overview of the project:

This report focuses on the predictive maintenance approach for wind turbines. The main goal of the model is to train the model on one dataset and set it on the other dataset so we can assess how well the model is performing on unseen data for the same features selected from one dataset, simulating real-

world deployment of each turbine behaving differently. Both supervised and unsupervised methods are built to see the failure detection.

3 Machine Learning Methods and Methodological design

In this study, both supervised and unsupervised machine learning methods were used. Based on the results of these models, it can be generalized which method is better for the dataset. Using these two methods, the model would be more robust and help the model to detect failure patterns through supervised learning and discover unseen and abnormal behavior through unsupervised learning.

3.1 Supervised Learning:

The supervised learning approach uses the approach of labeled dataset, where the normal and faulty operations are already known. Supervised is about a classification model where this model is trained to distinguish normal and anomaly points. Mainly decision trees and random forest are used for classification.

Here Random Forest Classifier was chosen due to its interpretability and strong performance on high-dimensional data. After feature selection, the classification model was trained. Depending on the feature, it resists overfitting.

To evaluate our model, it's better to train on one dataset and test on a completely different dataset. It confirms whether the model is trained properly or not and performing well on different unseen environments for detecting anomalies; this cross-validation method is critical.

To apply supervised learning methods, The SCADA dataset is already labeled by 'status_type_id' This column represents the state of the turbine, which is stated in 6 different types.

- Normal State (0): The turbine is in normal power production mode, represented by status type id = 0.
- Derated Phase (1): Derated power generation with a power restriction, represented by status type id = 1.
- Idling Phase (2): represented by status type id = 2.
- Service (3): Asset is in service mode / service team is at the site, represented by status type id = 3.
- Downtime (4): The asset is down due to a fault or other reasons, conformly considered to be an anomalous state represented by status type ID = 4.
- Other Phase (5): status type id = 5, indicating other operational states that signal potential issues due to downtime in the turbine.

Here, Normal behaviour: Status Id's - 0 and 2 Anomalous behaviour: Status Id's - 1,3,4,5

By converting these into binary classification labels (0 for normal behaviour and 1 for Anomalous behaviour) first its trained and tested on only one dataset.

3.1.1 Evaluation Metrics for Supervised Learning:

The evaluation used standard metrics:

- Accuracy: overall correctness of predictions
- Precision: how many predicted anomalies were actually correct
- Recall: how many actual anomalies were successfully detected
- F1-score: balance between precision and recall

  The results showed strong performance, confirming that the model could generalize well across different turbines and data conditions.

3.1.2 Anomaly-Detection Metrics:

True Positive Rate (TPR), models this result shows that actual anomalies detected by the model. High TPR indicates most failure cases which is crucial for our early fault prediction value.

False alarm rate (FAR) = FP / (FP + TN). Having A low FAR is crucial for industrial settings like wind turbines , lead to costly shutdowns can occur due to false alarms.

3.1.3 Performance on Training-Set Turbines:

As supervised learning models are trained on labeled data, here from table 1 it's been seen that it has performed well on train_test split within the dataset 35. Accuracy achieved is 95%. When we tested on different dataset 16, the overall accuracy was 89% but poorly performed on testing dataset like by achieving 0.78 and F1-score of 0.70.

Train Dataset	Precision	Recall	F1-score
df35	0.96	0.80	0.87

Table 1: Train test split within the dataset

Test Dataset	Precision	Recall	F1-score
df16	0.75	0.78	0.70

Table 2: Comparison of accuracy score on different testing dataset

3.1.4 ROC–AUC:

To evaluate model the Receiver Operating Characteristic (ROC) curve and False Positive Rate (FPR) have been used to observe all classification thresholds.

As we know the AUC of 1.0 represents perfect classification, while 0.5 reflects random guessing. In the graph shown the model has excellent discriminative performance by score of 0.97.

Since AUC is independent of the decision threshold, it is especially valuable for comparing models and understanding performance without committing to a fixed cut-off.

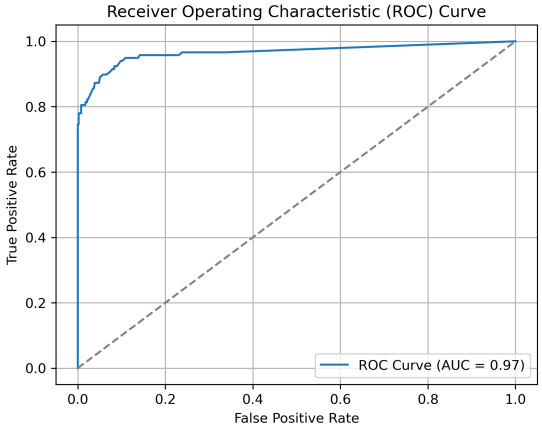

Figure 7: Visualization of temperature sensor

3.2 Unsupervised Learning

Unsupervised is used to detect unlabeled anomalies. There are many clustering methods like k-means, DBSCAN, Gaussian Mixture Model, and hierarchical clustering.

3.2.1 DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

In DBSCAN, the number of clusters need not be specified like in KMeans, and here DBSCAN and KMeans have been used. It's mainly structured to identify arbitrary shape and outlier clusters. It mainly focuses on abnormal points that are normally ignored by KMeans. This method is well suited for which has fewer irregularities in data points. Here the timeline chosen has very few anomalies. From Figure 8 and Figure 9, there are two clusters Blue ones are normal points, and red ones are anomalies. the downtime status type ID

is 4, which is analysed in figure 5 from selected features that can be seen from those red points the time_stamp and status_type_id.

DBSCAN produced a higher silhouette score of 0.699, indicating well-defined clusters with clear density separation. Its ability to handle noise and irregular cluster shapes made it particularly effective for this SCADA dataset, especially in identifying dense anomaly segments without need to pre-define the number of clusters.

To visualize the results of DBSCAN in a high-dimensional feature space, t-SNE (t-distributed Stochastic Neighbor Embedding) was used to reduce dimensionality to 2D. The t-SNE plots clearly showed distinct cluster boundaries, with isolated points correctly detected by DBSCAN as anomalies. This combination offered a powerful toolset for understanding complex turbine behaviors and visually validating anomaly detection results.

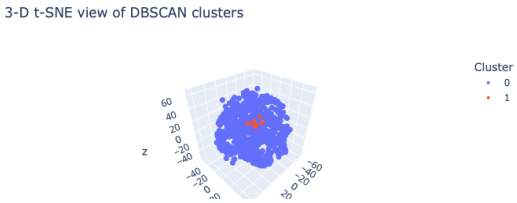

Figure 8: Visualization of DBSCAN cluster

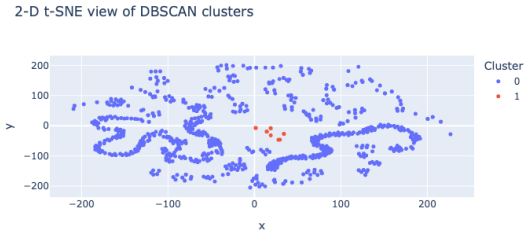

Figure 9: Visualization of DBSCAN cluster in 2D

3.2.2 K-Means Clustering:

KMeans clusters based on the similar data points. The Silhouette score, a measure of how similar a point is to its own cluster compared to other clusters, is used to validate clustering quality. The dataset used here has 6 days of stoppage; KMeans can identify different clusters corresponding to the anomaly. From the Figure 10 the green points are representing anomaly points.

K-Means Clustering: K-Means resulted in a lower silhouette score of 0.576 compared to DBSCAN. While it was still useful in grouping similar operational conditions, its performance was limited in capturing the irregular and noisy nature of real-world turbine anomalies, which often do not conform to spherical cluster shapes.

Though the visualization model is not trainable, the clustering indices can be reinitialized for better cluster distinguishability. (Threshold for k-nearest neighbors, number of clusters, k-means can be adjusted for better visualization.)

Both methods helped in validating anomaly segments, with K-Means offering more structured clustering and DBSCAN being more adaptable to irregular anomaly patterns.

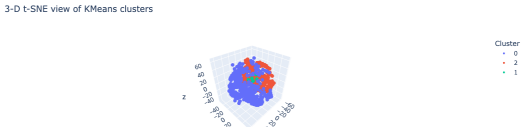

Figure 10: Visualization of KMeans cluster

4 Prefered Approach

The project was completed with both supervised and unsupervised learning, unsupervised learning was found to be more effective and practical for the following reasons:

- Better Precision in Anomaly Detection : the anomaly point are very precise giving the exact dates where the anomalies where for in initial sensor graphs, it revealed more focused patterns of abnormal behaviour.
- Density Based Approach : DBSCAN has helped to detect short sharp anomaly from the dataset within given time_stamp which has sliced after analysing feature.
- Visual Interactive : DBSCAN's clustering results can be visually understandable via time_stamp plots or dimensionality-reduced views like t-SNE, making it easier to understand why data point considered as an anomaly.
- Avoids Overfitting : supervised learning risk in overfitting but unspervised is naturally more resistant to this issue when we use it on different features.

5 Conclusion

In this project, It's been explored how machine learning can help detect early signs of failure in wind turbines using real SCADA sensor data. Started by cleaning and processing the data, then used a covariance heatmap to reduce the number of features from 957 to 253. To further narrow it down, I applied techniques like mutual information classification and SelectKBest, making sure the selected features were relevant and followed a normal distribution—important for reliable modeling.

For the supervised approach, trained a Random Forest classifier, which gave good results when tested across different turbines. It was especially effective in classifying normal vs. anomalous behavior when labeled data was available.

On the unsupervised side, DBSCAN has been used, which turned out to be very powerful. It doesn't rely on labeled data, making it more flexible and general. It was able to detect subtle changes and patterns in the sensor data that pointed to real anomalies. One of the best parts was using t-SNE to visualize the high-dimensional data—DBSCAN clearly picked out the outliers, and the clusters were easy to see. This really helped in understanding what was happening with the turbine, especially for one with short, repeated failures.

Overall, while the Random Forest model worked well when labels were available, DBSCAN stood out as the better choice for this type of data. It was more general, handled unseen data better, and gave more precise insight into when and where anomalies were happening—making it a great fit for real-world wind turbine monitoring.

6 Reference

1. https://www.mdpi.com/2071-1050/15/10/8333
2. https://www.youtube.com/watch?v=Hf875eOVrVI
3. https://www.youtube.com/watch?v=LklUVkMPl8g
4. https://www.researchgate.net/publication/382607570_Anomaly_detection_of_wind_turbines_based_on_sta