A Report for Machine Learning

"Project: Predictive Maintenance for Wind Turbines"

Sristi Kumari Bhadani

Supervised by
Prof. Dr. Partick Levi

Ostbayerische Technische Hochschule Amberg-Weiden
Department of Electrical Engineering, Media and Computer Science

June 4, 2025

Abstract

This study presents a data-driven approach for wind turbine fault diagnosis and early warning using one year of high-resolution SCADA data from Wind Farm C in Germany. After applying DBSCAN clustering on standardized wind speed and power to remove outliers, and filtering data using a fitted ideal power curve with $\pm$3sigma bounds, a refined set of sensor features was selected through correlation-based filtering guided by domain knowledge. Two predictive maintenance strategies were developed: an unsupervised method based on KMeans clustering trained on healthy data, and a supervised method using an XGBoost classifier trained on labeled pre-failure windows. The unsupervised approach achieved a fault lead time of 3 days with a false positive rate of 1.07%, while the supervised model reached 96.1% accuracy and an F1-score of 0.868 for anomalies, detecting failures 22 hours in advance with zero false alarms on normal data. Both methods effectively fulfill early detection requirements, with the supervised model offering superior precision and operational robustness. These findings demonstrate the feasibility of deploying scalable, SCADA-based machine learning solutions for real-time wind turbine condition monitoring. Moreover, the combination of feature engineering, time-aware evaluation, and minimal false positives highlights the practical value of these methods for industrial predictive maintenance systems.

1 Introduction

Wind power is becoming a dominant source of renewable energy globally, but wind turbines often operate in harsh environmental conditions, leading to higher failure rates compared to other electromechanical systems. These failures affect both the reliability and cost-effectiveness of wind farms. Traditional maintenance practices—based on scheduled servicing or reactive repairs—often result in extended downtime and increased costs, especially when considering the logistical challenges of servicing turbines in remote areas. The complex interactions between turbine components like gearboxes, generators, and control systems further elevate the risk of faults and cascading damage.

To address these challenges, accurate fault diagnosis and early failure warnings are essential for enhancing turbine availability and reducing maintenance expenses. By detecting issues at an early stage, operators can proactively plan maintenance, avoid sudden breakdowns, and improve overall safety. However, existing predictive maintenance approaches often struggle with the complexity of turbine operations and fail to generalize effectively across different turbines or operating conditions. In this context, data-driven techniques using Supervisory Control and Data Acquisition (SCADA) data have gained attention for their potential to capture detailed turbine performance metrics and inform predictive models.

This study presents a hybrid framework for wind turbine fault prediction using SCADA data from Wind Farm C. The method integrates DBSCAN clustering and statistical power curve estimation for effective identification of abnormal operational states in noisy data. We also implement an XGBoost-based supervised learning model, optimized for accurate fault classification, and an unsupervised KMeans-based anomaly detection approach to capture deviations from normal behavior. This dual strategy enhances early fault detection while maintaining low false alarm rates. Our results demonstrate superior lead times and predictive accuracy compared to baseline models, offering a scalable and interpretable solution for predictive maintenance in wind energy systems.

2 Data Analysis

2.1 Dataset Overview

This study utilises data from Wind Farm C located in the Germany. The dataset[3] used in this study is derived from SCADA data and spans for one year . It consists of ten-minute-average values from 238 sensors. For each of these sensors avg,std ,min, max are provied. There are 27 anomaly events and 31 normal behaviour events. The sensors capture various measurements, including power production, wind speed and direction, various currents and voltages, as well as temperature measurements of bearings. Every ten minutes within a year one sample (52,560 samples) with 957 features.There are two more files given one is feature description and the other one is event file stating event start time and event end time in prediction time frame for both normal and anomaly.

In addition to the SCADA data, the dataset also gives us the information about the operational mode for each time stamp. The "status_type_id" column given in the dataset shows us different modes as shown in the below figure. For the purpose of the study only 'normal operation' is considered as normal behavior.The Purpose of the trained model is its ability to represent normal behavior of the Wind turbine. Other data points are typically easy to identify, and the wind farm operator is already alerted to the abnormal behavior via the status ID. Therefore, they are not relevant for predictive maintenance purposes and would only dilute the effectiveness of the score as mentioned in the paper [3].

Status ID	Description	Considered Normal
0	Normal operation without limitations	True
1	Derated power generation with a power restriction	False
2	Asset is idling and waits to operate again	True
3	Asset is in service mode/service team is at the site	False
4	Asset is down due a fault or other reasons	False
5	Other operational states for example system test, setup, ice build-up, or emergeny power	False

Figure 1: Status Id Description

Source: [3]

2.2 Data Preprocessing and Normal Operation Data Extraction

In our work, we begin by selecting key features wind speed and power output from the original SCADA dataset for wind turbine C. Invalid or missing data points were filtered out to ensure only positive wind speed and power values are retained as shown in [7]. For our operation data with status_type_id as 0 or 2 is selected.

To handle noisy and anomalous data, we apply the DBSCAN clustering algorithm on the wind speed and power data as mentioned in [9]. Normalization using `StandardScaler` brings both features to a common scale, which is crucial for Euclidean distance-based clustering. DBSCAN identifies low-density noise points (labeled -1), which are removed, preserving only core data representing typical turbine operation. This cleaning step is illustrated in Figure 2 and 3 .

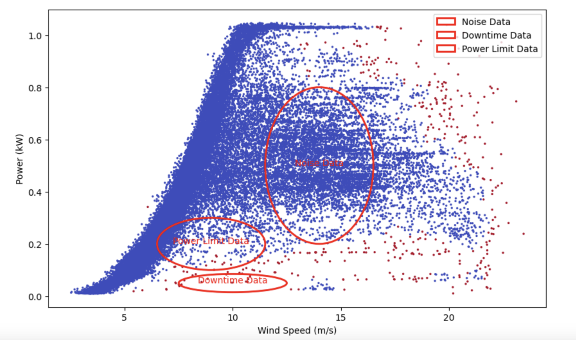

Figure 2: DBSCAN clustering: outlier removal in wind–power space. Downtime points are highlighted in red.

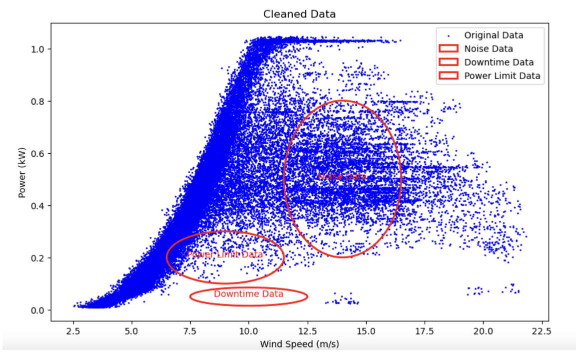

Figure 3: Cleaned data after removing DBSCAN-identified downtime points. Only core operational data is retained.

The cleaned data are then partitioned into bins based on power values (ranging from 0 to approximately 1.05), using a bin width of 0.05. For each power bin, the mean wind speed and power are computed as representative midpoints. These midpoints are used to fit a piecewise ideal power curve approximating normal turbine behavior. Specifically, a cubic polynomial models the rising section of the power curve (up to 13 m/s wind speed), reflecting the cubic power law, while the power is capped beyond this cutoff to model turbine operational limits. The piecewise function $P(w)$ is defined as:

$$P(w) = \begin{cases} a_3 w^3 + a_2 w^2 + a_1 w + a_0, & w \le w_c \\ P(w_c), & w > w_c \end{cases} \tag{1}$$

where w is the wind speed, $w_c = 13\,m/s$ is the cutoff speed, and a_0, a_1, a_2, a_3 are the polynomial coefficients determined by least squares fitting.

To define the normal operational range around this ideal curve, the average standard deviation σ of wind speed within each power bin is calculated as in [9]. Using this value, a 3-sigma band in the wind speed domain is created, forming upper and lower bounds around the ideal power curve:

$$P_{\text{lower}}(w) = P(w - 3\sigma), \quad P_{\text{upper}}(w) = P(w + 3\sigma) \tag{2}$$

Data points outside these bounds are considered anomalous and excluded.

Figure 4 shows the power data with the fitted piecewise ideal power curve and ±3sigma boundaries. Most operational points fall within this range, confirming its suitability for extracting a clean and reliable healthy dataset. We have also implemented data filtering based on the status_type_id provided in the dataset, as shown in 1, where we selected data corresponding to status IDs 0 and 2

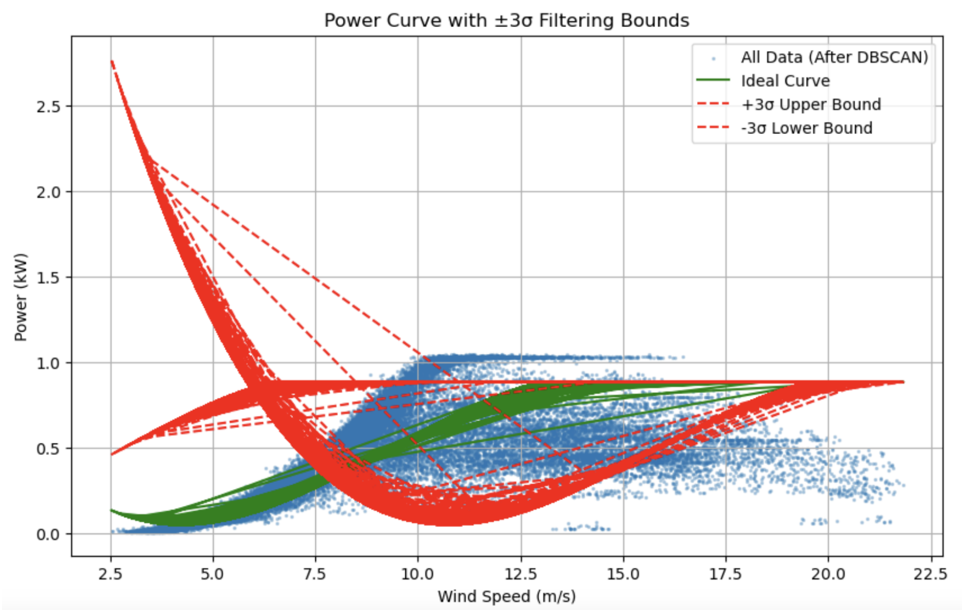

Figure 4: power versus wind speed with fitted piecewise ideal power curve (green) and ±3sigma bounds (red dashed).

This methodology effectively cleans the data by removing noise and anomalies while capturing natural variability, producing a reliable baseline for fault detection and predictive maintenance [9].

2.3 Feature Selection and Categorization

The SCADA dataset for wind turbine C comprises a large number of sensor measurements capturing different aspects of turbine operation. We have taken one turbine with asset_id 21 for our analysis but these can be extended to any turbines. Initially, sensor features were grouped into categories based on their physical meaning and function. The raw counts of sensor features (including all statistical suffixes such as `_avg`, `_std`, etc.) for each category are listed below these counts are with the overlapping features as well therefore be careful if we add this may increase the total number of columns in the dataset.

- Power-related features: 301 columns
- Wind features: 28 columns
- Temperature-related features: 316 columns
- Vibration features: 16 columns
- Counter features: 0 columns
- Angle features: 47 columns
- Hydraulic features: 131 columns
- Cooling features: 120 columns
- Motor current features: 43 columns
- Yaw motor features: 24 columns
- Generator features: 260 columns
- Gearbox features: 76 columns
- Other features: 53 columns

Before finalizing the feature set, we performed several preprocessing steps to ensure data quality and model robustness:

1. Features with low variance, indicated by being constant or having up to 3 unique values, were eliminated as they provide limited information.
2. Although the dataset provided by operators contains no missing values, many features exhibit consecutive zero values suspected to represent missing or invalid measurements. Hence, features with more than 80% zero values were removed. [7]
3. Features representing angles were transformed into their sine and cosine components to maintain continuity and prevent artificial discontinuities due to angular wraparound.
4. Features categorized as counters were dropped, as they represent cumulative counts rather than instantaneous sensor readings, making them less suitable for anomaly detection.

2.4 Correlation-Based Feature Reduction

To reduce redundancy and avoid keeping features that tell us the same thing, we looked at how strongly the features within each category were correlated, using a cutoff of 0.9. When deciding which features to drop among those that were very similar, we relied on domain knowledge to prioritize more important features in the following order: Hydraulic features come first, followed by Temperature, Cooling, Generator, Gearbox, Electrical (Power), and finally Vibration features.

To retain the most representative features from above categories, a priority scheme is defined based on the feature naming pattern giving priority to avg, then to std, then to min and finally to max.

In each highly correlated pair, the feature with lower priority is dropped. This prioritization ensures that more informative and critical sensor groups are preserved in the final feature set while eliminating redundant features with lower priority.

After filtering and overlap resolution, the final selected feature counts per category are summarized in Table 1.

Feature Category	**Filtered Feature Count**
Hydraulic	60
Temperature	57
Cooling	22
Generator	11
Gearbox	9
Electrical (Power)	65
Vibration	6

Table 1: Number of sensor features per category after correlation filtering and overlap removal.

These curated feature groups provide a comprehensive and non-redundant representation of the turbine's physical and operational state. This careful feature selection ensures that subsequent analysis and anomaly detection models utilize the most informative and robust signals, enhancing predictive maintenance capabilities.

3 Methodology and Results: Using KMeans Clustering

3.1 Training and Test Split

The sensor dataset contains multiple features representing the operational state of the asset. Non-sensor metadata columns such as timestamps, asset IDs, status IDs, and labels were excluded to focus solely on sensor measurements. To ensure consistency, only the sensor features common to both training and test datasets were selected. We have used the pre defined column "train-test" in the dataset where we first separated based on the train and prediction flag.

We focus on turbine with asset ID 21. The failure event is known by events_info file to start at

$$\texttt{failure\_start} = \text{2023-12-22 15:00:00}$$

and the pre-failure window is set as 4 days prior:

$$\texttt{prefailure\_start} = \texttt{failure\_start} - 4 \text{ days}$$

Training data consists of healthy sensor readings outside of fault and pre-failure windows. The test set comprises sensor data from the pre-failure window and the fault window itself. Only relevant sensor columns are selected, excluding metadata such as timestamps, asset IDs, and status flags.

3.2 Data Normalization

Sensor measurements vary widely in scale and units. To prevent any single feature from dominating the clustering, data normalization was applied using `StandardScaler`, which standardizes each feature to have zero mean and unit variance. The scaler was fit on the training data and then applied to both training and test datasets to maintain consistent scaling.

3.3 Training the KMeans Model

K means clustering model with a single cluster ($n\_clusters = 1$) was trained using the normalized training data representing healthy (normal) operation. The model identifies the centroid of normal behavior in the multi-dimensional sensor space, effectively summarizing typical healthy conditions.

3.4 Computing Anomaly Scores

For each data point in the test set (including pre-failure and fault periods), the anomaly score was computed as the Euclidean distance between the normalized test sample and the learned cluster centroid. Mathematically, this distance is:

$$\text{distance} = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2}$$

where x_i are the feature values of the test sample, and c_i are the coordinates of the cluster centroid. A small distance indicates similarity to normal data, while a large distance suggests deviation and potential anomaly.

3.5 Smoothing Anomaly Scores

To reduce noise in the anomaly scores caused by sensor fluctuations, a rolling average with a window size of 3 was applied to smooth the scores. This smoothing helps in highlighting meaningful trends over time.

3.6 Threshold Determination

A threshold to separate normal from anomalous behavior was derived from the smoothed anomaly scores during the baseline healthy period (pre-fault). The threshold corresponds to the 99th percentile of the baseline anomaly scores, ensuring that 99% of normal data points fall below this value. Scores exceeding this threshold are considered anomalous.This threshold needs to be fine tuned according to the use case.

3.7 Anomaly Detection

The earliest timestamp at which the smoothed anomaly score crosses the threshold is recorded as the anomaly detection time, enabling early warning prior to the actual fault onset.

3.8 Validation and False Positive Rate

The model was validated using an independent baseline normal dataset without faults for the same turbine. Anomaly scores were computed on this baseline data, and the threshold applied to estimate the false positive rate, the proportion of normal samples incorrectly flagged as anomalies. A low false positive rate (approximately 1%) demonstrates the model's effectiveness in distinguishing true anomalies while minimizing false alarms.

3.9 Results and Observations

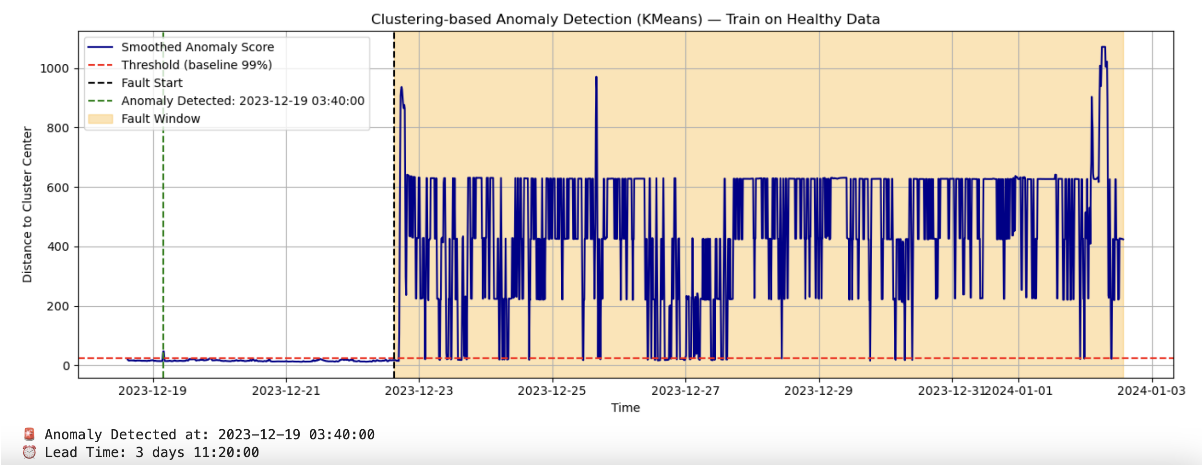

Figure 5: Smoothed anomaly score (blue) over time with threshold (red dashed line). Black dashed line indicates fault start. Green dashed line marks earliest detected anomaly, approximately 3 days and 11 hours before failure. Orange shading highlights fault window.

As seen in Figure 5, the anomaly score remains stable and below threshold during healthy operation. A distinct anomaly breach occurs several days before the fault start, providing valuable lead time for maintenance intervention.

3.10 Anomaly Score Distributions During Normal and Fault Periods

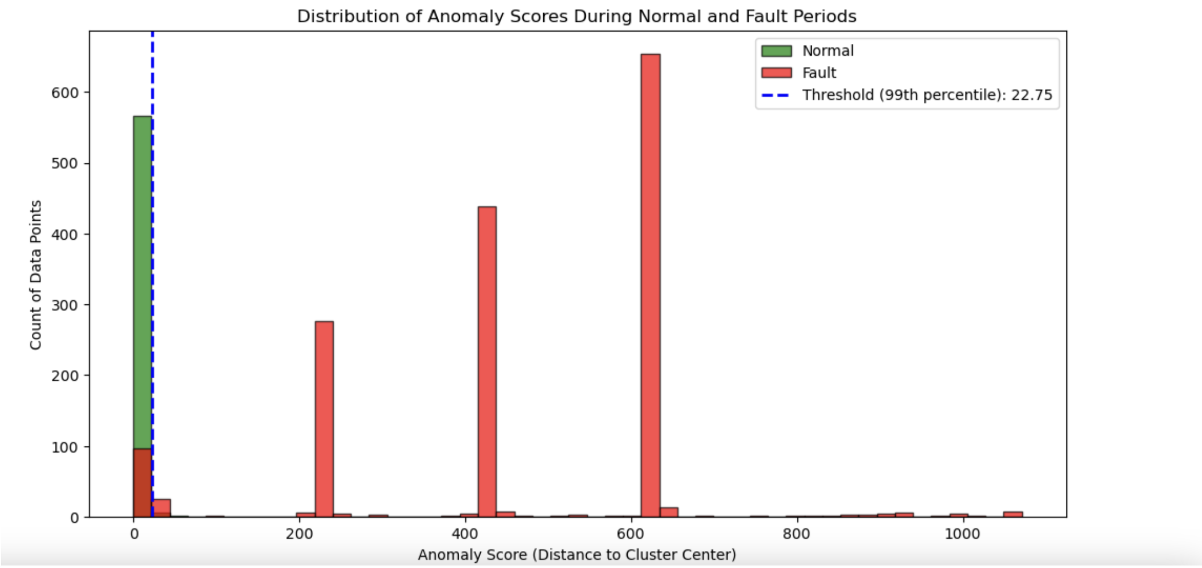

Figure 6: Histogram of anomaly scores during normal (green) and fault (red) periods. The blue dashed line indicates the 99th percentile threshold. Fault periods have significantly more high anomaly scores.

Figure 6 shows clear separation in anomaly score distributions between normal and fault periods, demonstrating the method's ability to discriminate normal states.

3.11 Baseline Normal Data: False Positive Analysis

The dataset also includes normal operation data for the same turbine. Testing on this data revealed a low false positive rate of 1.0659 percent, as illustrated in Figure 7, which demonstrates the model's reliability and its effectiveness in minimizing false alarms.

3.12 Anomaly Detection Performance Evaluation

The performance of the KMeans clustering-based anomaly detection model was evaluated using standard classification metrics. Figure 8 shows the confusion matrix, which summarizes the model's ability to distinguish between normal and fault conditions.

The classification report in Table 2 quantifies the model's precision, recall, and F1-score for both normal and fault classes, along with overall accuracy metrics.

The confusion matrix indicates that the model correctly identifies a high proportion of both normal and fault instances, with only a small number of misclassifications. The high recall for the normal class (0.99) shows the model's effectiveness in minimizing false alarms, while the fault class recall (0.93) demonstrates strong early fault detection capabilities. Overall, the model achieves an accuracy of 95%, confirming its suitability for predictive maintenance in wind turbines.

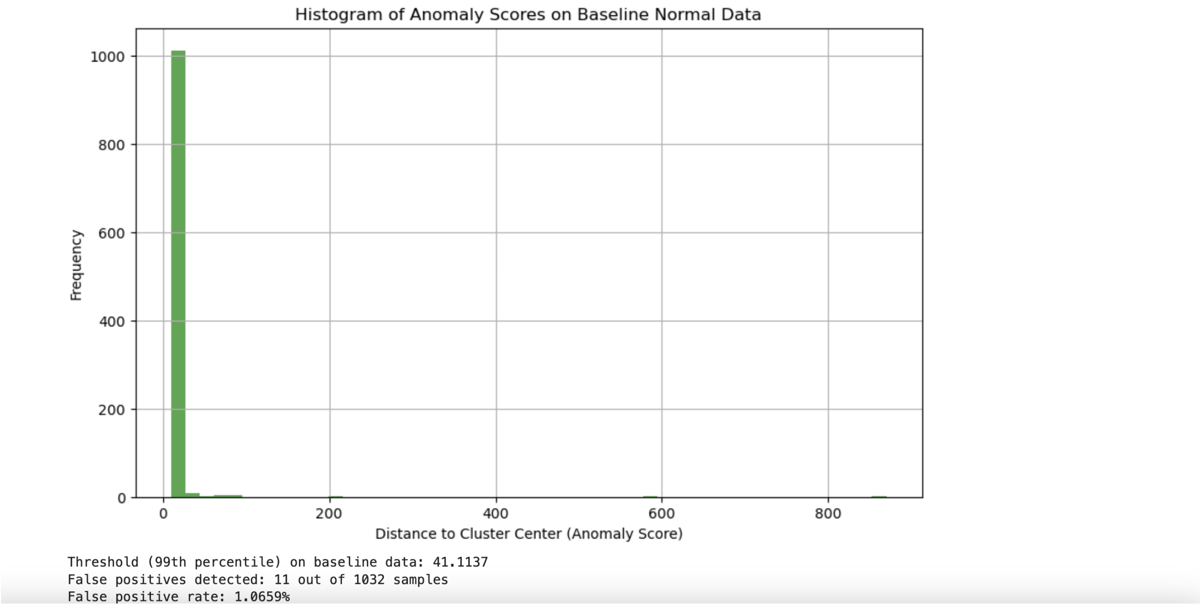

Figure 7: Histogram of anomaly scores on baseline healthy data used for thresholding. Most scores lie below the threshold, with only 1.07% false positive rate (11 out of 1032 samples).

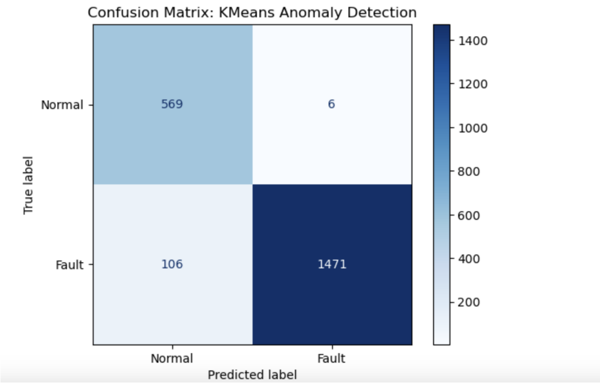

Figure 8: Confusion Matrix for KMeans Anomaly Detection

Classification Report:

	Precision	Recall	F1-score	Support
Normal	0.84	0.99	0.91	575
Fault	1.00	0.93	0.96	1577
Accuracy			0.95	2152
Macro avg	0.92	0.96	0.94	2152
Weighted avg	0.96	0.95	0.95	2152

Table 2: Classification report for the unsupervised KMeans clustering-based anomaly detection.

4 Methodology and Results: Supervised Learning Approach

4.1 Train-Test Split and Labeling Strategy

For this study, we used the predefined `train_test` column in the dataset to divide the data into training and test sets. This column specifies whether each observation should be used for model training or for prediction, which helps preserve the natural time order of the data a crucial aspect in any predictive maintenance application, where the goal is to forecast future failures based on past behavior.

To make the test set more realistic and allow for early detection evaluation, we extended it by including the 24-hour period leading up to each known failure event for the selected turbine (Asset ID 21). These pre-failure windows represent the kind of situations where a predictive system would ideally raise an alert, giving maintenance teams time to respond before a breakdown occurs.

Labeling was done as a binary classification task. In this setup, prexisting status_type_id column is not used. By default, all data points were labeled as normal (0). In the test set, any data that fell within the 24 hours before a failure was labeled as anomalous (1), marking it as part of the pre-fault period. In the training set, we used a slightly wider window of 7 days before each failure to label anomalies [7]. This gives the model more examples of how a system typically behaves in the lead-up to a failure. To make sure there's no overlap between training and test data, these pre-failure segments were removed from the training set before the labels were assigned.

Overall, this setup allows the supervised learning model to learn meaningful patterns from early warning signs in the training data and then demonstrate its ability to detect similar patterns in the test data ideally, before any actual failure occurs.

4.2 Feature Selection and Model Training

For the supervised learning approach, we followed a feature selection process similar to the one used in our unsupervised method (see Section 1). Starting with the preprocessed dataset, we excluded metadata fields such as `id`, `time_stamp`, `asset_id`, and `train_test`,`status_type_id` as these do not carry meaningful information for prediction. What remained were purely sensor-based features, which reflect the physical behavior of the wind turbines.

Since failure events are rare compared to normal operation, the dataset was highly imbalanced. To address this, we sampled 10,000 healthy data points (label = 0) and combined them with all available pre-failure samples (label = 1). This gave us a more balanced dataset, allowing the model to learn patterns associated with anomalies without being overwhelmed by the majority class. We then shuffled the combined data to avoid any unintended temporal patterns affecting the training process.

With the cleaned and balanced dataset in place, we prepared the input features and labels for both training and testing. For model training, we used the XGBoost classifier, a decision-tree-based ensemble method well-suited for tabular sensor data due to its high accuracy, built-in regularization to reduce overfitting, and robust handling of feature interactions and class imbalance. Its ability to efficiently capture nonlinear relationships while maintaining computational speed makes it ideal for anomaly detection tasks in SCADA datasets. We configured the model with 100 boosting rounds, a maximum tree depth of 5, and a learning rate of 0.1. To further help the model deal with the class imbalance, we adjusted the `scale_pos_weight` parameter based on the ratio of normal to anomaly samples in the training set.

After training, the model was evaluated on the test data using standard classification metrics such as precision, recall, and F1-score. These metrics help us understand how well the model detects anomalies in advance while minimizing false positives. The goal was not just to catch failures, but to catch them early and only when it truly matters.

4.3 Results

4.3.1 Model Performance

The supervised learning model was evaluated on a test set containing both normal and pre-failure operational data. As shown in Table 4.3.1, the model achieved a high overall accuracy of 96.1%. More importantly, the F1-score for the minority (anomaly) class was 0.868, indicating strong predictive performance even under class imbalance.

Classification Report on Test data:

	precision	**recall**	**f1-score**	**support**
Normal	0.968	0.987	0.977	1577
Fault	0.922	0.819	0.868	288
accuracy			0.961	1865
macro avg	0.945	0.903	0.923	1865
weighted avg	0.961	0.961	0.960	1865

4.3.2 Confusion Matrix

The confusion matrix in Figure 9 further illustrates that the model correctly identified 236 out of 288 anomalous cases, with only 52 false negatives. False positives were limited to 20 instances, demonstrating strong precision for both classes.

4.3.3 Temporal Detection Behavior

To assess whether the model detects anomalies sufficiently early, we analyzed its predictions over time relative to the actual failure event. Figure 10 shows the predicted anomaly labels in a 4-day window surrounding a known fault. The shaded region indicates the 24-hour pre-failure window, during which the model is expected to detect early warning signs.

The model triggered a consistent anomaly streak (10 consecutive positive predictions) starting at approximately 15:00 on December 21, nearly 24 hours in advance of the actual failure at 15:00 on December 22. This provides a meaningful lead time for operational response. Isolated false positives were rare and temporally scattered, demonstrating that the model balances early detection with a low false alarm rate.

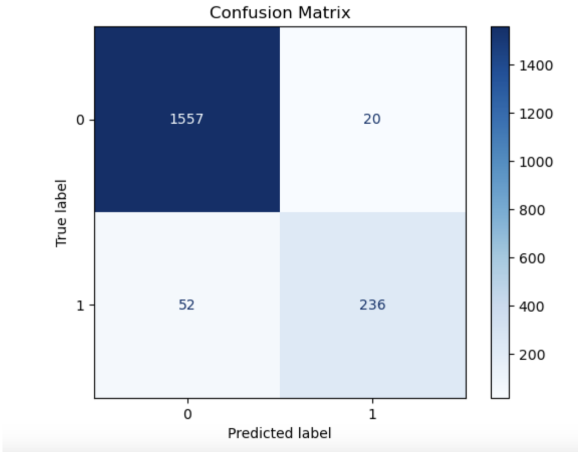

Figure 9: Confusion matrix showing true vs. predicted labels.

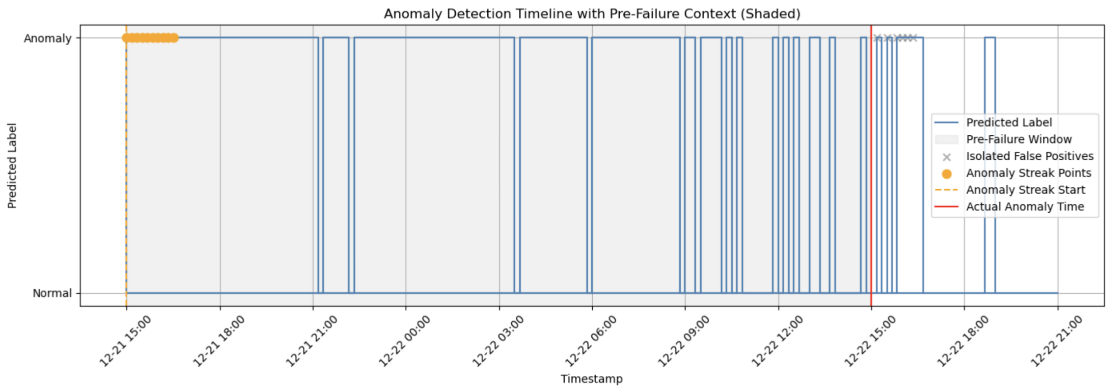

Figure 10: Anomaly detection timeline showing predicted labels, actual failure time, and pre-failure window.

4.3.4 Lead Time Evaluation

From the timeline visualization, the first true anomaly streak was detected at 2023-12-21 15:00, offering a lead time of 22 hours and 40 minutes before the failure. This confirms that the supervised model is capable of not only identifying faults, but doing so with actionable advance notice.

4.3.5 False Positive Evaluation on Normal Data

To further evaluate the reliability of the model, we tested its predictions on a separate segment of data from the same turbine where no failure events were recorded and the system was operating under fully normal conditions. This test consisted of 1,032 data points. The model produced zero false alarms, resulting in a false positive rate of 0.00%.

This result is particularly important in the context of predictive maintenance, where false positives can lead to unnecessary maintenance operations, increased costs, and reduced trust in the system. The model's ability to maintain such a low false alarm rate, while still detecting anomalies with sufficient lead time, demonstrates its practical suitability for deployment in real-world wind turbine monitoring systems.

5 Conclusion and Comparison of Approaches

In this project, we developed and evaluated two predictive maintenance solutions for wind turbines using SCADA data from Wind Farm C. Both approaches one unsupervised (based on KMeans clustering) and one supervised (based on the XGBoost classifier) were carefully designed to detect failure events in advance while minimizing false alarms, fulfilling the customer's core requirements.

The unsupervised approach relied solely on normal (healthy) operation data to model baseline turbine behavior. By computing anomaly scores as distances from the healthy cluster centroid, it was able to detect deviations that preceded failure events. This method showed robust early detection capability, with the first anomalies detected up to 3 days and 11 hours in advance of actual failures. It also maintained a low false positive rate of 1.07%, validating its reliability for real-world deployment. The confusion matrix illustrating this method's performance is shown in Figure 8.

The supervised learning approach used labeled pre-failure windows to train an XGBoost model to distinguish between normal and faulty states. It achieved even stronger predictive performance, with an overall accuracy of 96.1% and an F1-score of 0.868 for the anomaly class. Importantly, it provided up to 22 hours and 40 minutes of lead time while producing zero false positives on a normal baseline dataset. The corresponding confusion matrix is shown in Figure 9.

While both models demonstrated strong performance,we recommend the supervised approach using the XGBoost classifier for deployment. It offers superior precision, lower false alarm rate, and better interpretability, all of which are critical for building trust and operational efficiency in a predictive maintenance system.

However, if labeled pre-failure data is unavailable or limited, the unsupervised method still provides a viable, accurate alternative with strong early warning capabilities.

References

[1] Mahwish Anwar, Lars Lundberg, and Anton Borg. "Improving anomaly detection in SCADA network communication with attribute extension". In: *Energy Informatics* 5 (2022), p. 69. DOI: `10.1186/s42162-022-00244-5`. URL: `https://doi.org/10.1186/s42162-022-00252-1`.

[2] Liming Gou et al. "State Reliability of Wind Turbines Based on XGBoost–LSTM and Their Application in Northeast China". In: *Sustainability* (). DOI: `10.3390/su16104099`. URL: `https://www.mdpi.com/2071-1050/16/10/4099`.

[3] Christian Gück, Cyriana M. A. Roelofs, and Stefan Faulstich. "CARE to Compare: A Real-World Benchmark Dataset for Early Fault Detection in Wind Turbine Data". In: *Data* 9.12 (2024), p. 138. DOI: `10.3390/data9120138`. URL: `https://doi.org/10.3390/data9120138`.

[4] Marc-Alexander Lutz et al. "Evaluation of Anomaly Detection of an Autoencoder Based on Maintenance Information and Scada-Data". In: *Energies* 13.5 (2020), p. 1063. DOI: `10.3390/en13051063`. URL: `https://doi.org/10.3390/en13051063`.

[5] OpenAI. *ChatGPT*. `https://chatgpt.com`. Accessed: 2025-06-04. 2024.

[6] Cyriana M.A. Roelofs, Christian Gück, and Stefan Faulstich. “Transfer learning applications for autoencoder-based anomaly detection in wind turbines”. In: *Energy and AI* 17 (2024), p. 100373. DOI: 10.1016/j.egyai.2024.100373.

[7] Cyriana M.A. Roelofs et al. “Autoencoder-based anomaly root cause analysis for wind turbines”. In: *Energy and AI* 4 (2021), p. 100063. DOI: 10.1016/j.egyai.2021.100063. URL: https://doi.org/10.1016/j.egyai.2021.100063.

[8] Cyriana M.A. Roelofs et al. “Autoencoder-based anomaly root cause analysis for wind turbines”. In: *Energy and AI* 4 (2021), p. 100065. DOI: 10.1016/j.egyai.2021.100065. URL: https://doi.org/10.1016/j.egyai.2021.100065.

[9] Yujia Shi, Yuhuan Liu, and Xuan Gao. “Study of Wind Turbine Fault Diagnosis and Early Warning Based on SCADA Data”. In: *IEEE Access* 9 (2021), pp. 124600–124615. DOI: 10.1109/ACCESS.2021.3110909.

[10] Adrian Stetco et al. “Machine learning methods for wind turbine condition monitoring: A review”. In: *Renewable Energy* 133 (2019), pp. 620–635. DOI: 10.1016/j.renene.2018.10.047. URL: https://doi.org/10.1016/j.renene.2018.10.047.