PREDICTIVE MAINTENANCE FOR WIND TURBINES: A DATA-DRIVEN APPROACH

Rajvi Parekh*
Georgia Institute of Technology
rparekh43@gatech.edu

Shikhar Shiromani*
Georgia Institute of Technology
sshiromani3@gatech.edu

Atharva Bhalerao*
Georgia Institute of Technology
abhalerao9@gatech.edu

Ahmadreza Chokhachian*
Georgia Institute of Technology
chokhachian@gatech.edu

Amriteshnarayanan Kottilil Manilal*
Georgia Institute of Technology
amanilal3@gatech.edu

ABSTRACT

Effective maintenance strategies are critical for ensuring the reliability and efficiency of wind turbines, which are integral to the renewable energy sector. This project focuses on developing an early warning system to predict and anticipate faults in wind turbine subsystems, enabling operators to proactively address potential issues before they escalate into costly failures. Leveraging data science and predictive modeling, the system aims to balance precision and practicality, issuing timely alerts that maximize cost savings while minimizing unnecessary interventions. By integrating technical accuracy with cost-effective decision-making, this initiative enhances turbine reliability and contributes to the broader objective of advancing renewable energy systems. The resulting framework demonstrates the transformative potential of predictive analytics in modern industrial operations, offering valuable insights for optimizing maintenance practices.

1 INTRODUCTION

The increasing adoption of renewable energy has positioned wind turbines as a critical component of sustainable power generation. However, ensuring the reliability and efficiency of these systems presents significant challenges, particularly due to their susceptibility to mechanical and operational failures. Failures in key turbine components, such as the generator, gearbox, and hydraulic systems, can lead to significant downtime, reduced energy output, and high maintenance costs.

The objective of this study is to develop a predictive maintenance framework that anticipates turbine failures and minimizes associated costs. By leveraging operational data and advanced analytical techniques, the framework aims to identify patterns and anomalies that precede failures, enabling proactive maintenance decisions. This approach not only enhances turbine reliability but also reduces downtime and operational expenses.

The economic implications of turbine failures are substantial, as illustrated in Figure 1, which highlights the costs associated with maintenance and replacement of key components. For instance, failures in the generator or gearbox incur particularly high replacement costs, emphasizing the need for early fault detection. Furthermore, unplanned turbine downtime disrupts energy generation, reducing the reliability of wind farms and impacting energy markets.

This study is driven by the need to address these challenges through predictive analytics. By analyzing operational data at fine-grained intervals and leveraging dimensionality reduction techniques

*Equal contribution.

Subsystems	***R***	***M***	***I***
Gearbox	100,000	20,000	5,000
Generator	60,000	15,000	5,000
Generator bearings	30,000	12,500	4,500
Transformer	50,000	3,500	1,500
Hydraulic	20,000	3,000	2,000

Figure 1: Cost breakdown of replacement, maintenance, and inspection for turbine components.

such as Principal Component Analysis (PCA), we aim to uncover patterns that signal impending failures. The insights gained can inform decision-making and contribute to the development of efficient maintenance strategies.

The remainder of this paper discusses the dataset, methodology, experimental results, and insights derived from the analysis. The findings underscore the potential of data-driven approaches in enhancing the reliability and efficiency of wind turbines.

2 Related Work

The advancement of predictive maintenance strategies for wind turbines has garnered significant attention, aiming to enhance operational efficiency and reduce downtime. Traditional maintenance approaches, such as reactive and time-based maintenance, often fall short in preempting rare yet critical failures. Consequently, recent research has focused on integrating machine learning techniques with Supervisory Control and Data Acquisition (SCADA) data to develop more effective predictive maintenance models.

(2) provide a comprehensive overview of predictive maintenance methodologies in renewable energy systems, emphasizing the limitations of conventional maintenance strategies and advocating for data-driven approaches. They highlight the potential of machine learning algorithms to process vast amounts of operational data, thereby enabling the early detection of anomalies and the prediction of component failures.

In the context of wind turbines, (3) proposed a cumulative sum (CUSUM) anomaly detection method for gearbox failure prediction. Utilizing multivariate time-series data, their approach aimed to identify deviations from normal operating conditions. While the method demonstrated high sensitivity in detecting anomalies, it also exhibited a considerable rate of false positives, indicating the necessity for more robust detection frameworks.

(4) explored fault detection in wind turbines through supervised machine learning models. Their study underscored the importance of integrating SCADA data with meteorological information to enhance fault detection accuracy. The authors also emphasized the critical role of feature engineering and dimensionality reduction in managing high-dimensional datasets, which is essential for developing efficient predictive models.

The Multidimensional Anomaly Detection and Interpretation (MADI) framework, introduced by (5), offers an innovative approach to high-dimensional anomaly detection. By employing negative sampling to delineate decision boundaries between normal and anomalous behavior, this method effectively detects sparse anomalies. However, its reliance on synthetic negative samples presents challenges in achieving high precision, particularly in complex operational environments.

Deep learning techniques have also been applied to predictive maintenance in wind turbines. (6) investigated the use of deep learning models for predictive maintenance of offshore wind turbines. Their study highlights the effectiveness of deep learning in modeling complex relationships within operational data, facilitating the early detection of potential failures. However, the authors note challenges related to data quality and the need for extensive computational resources.

Similarly, (1) applied statistical process control and machine learning techniques to diagnose wind turbine faults and predict maintenance needs. Analyzing extensive sensor data, their approach successfully identified patterns indicative of impending failures, thereby enabling timely maintenance interventions.

Despite these advancements, challenges persist in the practical deployment of predictive maintenance models. Issues such as data quality, model generalization, and the implementation of real-time machine learning-driven systems in wind farms remain critical barriers. Addressing these challenges requires the development of scalable and robust predictive maintenance frameworks that can operate effectively in diverse operational conditions.

In this study, we propose a comprehensive approach that integrates high-resolution SCADA data, meteorological inputs, and detailed failure logs into a unified framework. By combining dimensionality reduction techniques, such as Principal Component Analysis (PCA), with advanced predictive modeling, our methodology aims to enhance the reliability and efficiency of wind turbine maintenance strategies. This approach builds upon prior work while addressing existing gaps in scalability, false positive reduction, and the provision of actionable insights for operational decision-making.

3 Data Collection

The dataset utilized in this study comprises operational data collected from wind turbines over a two-year period. This data is essential for understanding turbine performance and predicting failures. The primary data source is a Supervisory Control and Data Acquisition (SCADA) system, which continuously records turbine operations. The dataset also includes meteorological data and detailed failure records to provide a comprehensive view of turbine behavior.

The SCADA signals capture key time-series metrics such as rotational speed, power output, temperature, and vibrations, recorded at 10-minute intervals. This high-resolution data allows for precise tracking of operational changes over time. Meteorological data, including wind speed, wind direction, and atmospheric pressure, were collected concurrently to understand the impact of environmental conditions on turbine performance. Additionally, the failure records document the occurrence of faults, specifying the affected turbine, the timestamp of the failure, and the component involved (e.g., generator, gearbox, or hydraulic systems).

The dataset is divided into a training set, comprising 20 months of data annotated with failure events, and a testing set, covering 4 months of data with failure annotations withheld to enable unbiased evaluation of predictive models. Both datasets are available at fine-grained (10-minute) intervals and aggregated daily levels. Missing data were addressed through mean imputation or nearest-timestamp matching to ensure the robustness of the analysis.

This rich and diverse dataset provides the foundation for analyzing failure patterns, exploring operational trends, and developing predictive maintenance strategies.

4 Exploratory Data Analysis

Exploratory Data Analysis (EDA) was performed to understand the dataset's structure, uncover patterns, and derive actionable insights that informed subsequent modeling and analysis. The dataset, comprising SCADA signals, meteorological data, and failure logs, presented high-dimensional, multivariate time-series data, making EDA an essential step to evaluate operational trends, failure dynamics, and the potential for predictive maintenance.

One of the first steps in EDA involved comparing 10-minute interval data with daily aggregated data to assess the impact of data granularity on trend detection and anomaly analysis. As shown in Figure 2, the 10-minute data captured fine-grained temporal fluctuations, which are critical for identifying short-term anomalies that often precede failures. However, the data was inherently noisy, making it less suitable for high-level trend analysis.

In contrast, the daily aggregated data smoothed out the short-term noise, revealing longer-term trends and gradual operational shifts. This allowed for easier identification of sustained deviations in key metrics, such as temperature and rotational speed, which are linked to impending failures. The analysis demonstrated that daily data is more suitable for strategic decision-making and monitoring, while 10-minute data is essential for real-time anomaly detection.

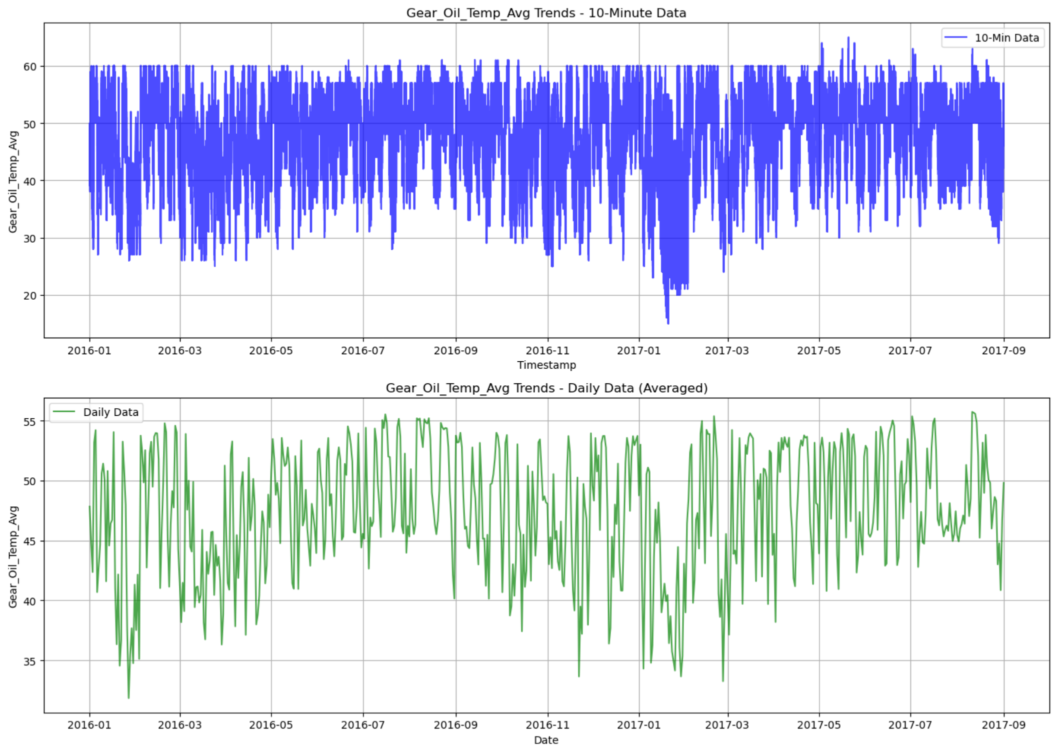


Figure 2: Comparison of 10-Minute and Daily Data Granularity.

The failure dataset provided detailed records of faults categorized by turbine components, such as gearboxes, generators, and hydraulic systems. An analysis of failure distribution revealed that generators and gearboxes accounted for the majority of failures, as shown in Figure 3. These components are critical to turbine functionality, and their high failure rates highlight the need for targeted predictive maintenance strategies.

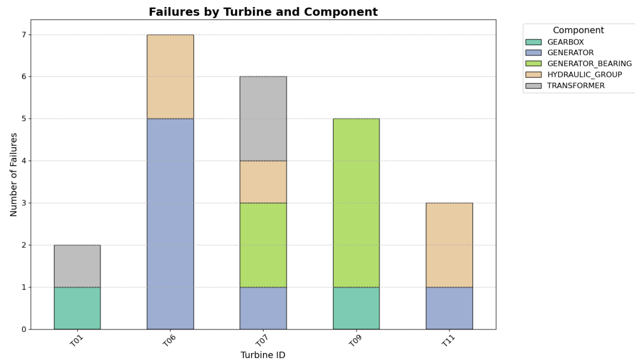
Furthermore, the analysis at the turbine level showed that certain turbines experienced repeated failures within short time intervals. This suggested potential design flaws, operational inconsistencies, or unique environmental stressors affecting specific turbines. Understanding these patterns is crucial for optimizing maintenance schedules and improving turbine reliability.

To address the high dimensionality of the SCADA data, Principal Component Analysis (PCA) was applied to reduce the dataset to its most informative components while preserving variance. Initially, PCA was performed to reduce the data to two dimensions, allowing visualization of turbine behavior before and after failures. These 2D PCA plots, as shown in Figure 4, revealed distinct behavioral shifts near failure events, particularly in the generator and gearbox components.

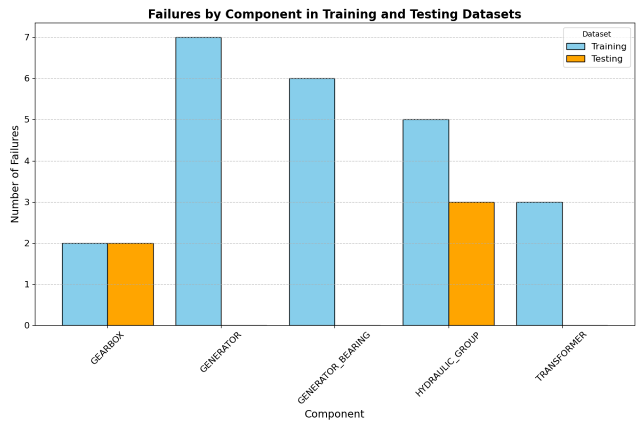
To further explore multi-dimensional trends, PCA was extended to three dimensions, enabling the visualization of operational dynamics across multiple time windows, as shown in Figure 5. These 3D plots provided a comprehensive view of how turbine behavior evolves before and after failures, highlighting the temporal and component-specific nature of operational anomalies.

The PCA analysis confirmed that turbine failures are preceded by distinct operational changes, providing valuable signals for early warning systems. Additionally, the variance captured by the principal components suggested that dimensionality reduction does not significantly compromise the information required for predictive modeling.

Temporal analysis of key variables, such as `Gear_Oil_Temp_Avg` and `Gen_RPM_Avg`, revealed gradual trends that align with long-term wear and tear, as well as short-term anomalies indicative



(a) Failures by Component.



(b) Failures by Turbine.

Figure 3: Failure Analysis and PCA Visualizations in One Row.

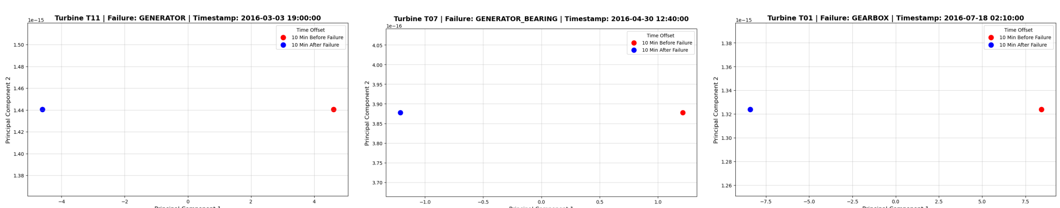


Figure 4: 2D PCA Visualization: Behavioral Changes Before and After Failures.

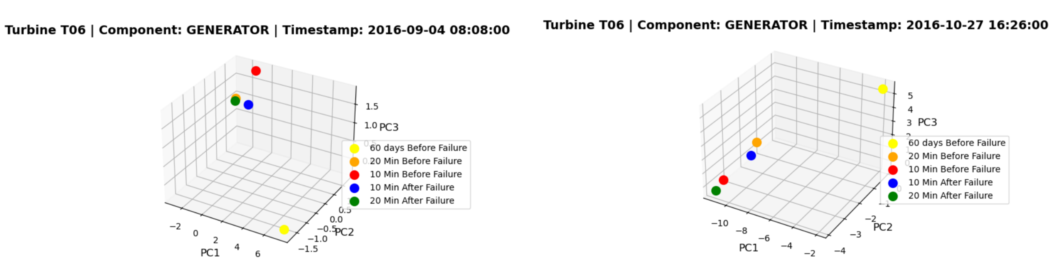


Figure 5: 3D PCA Visualization: Behavioral Changes Before and After Failures.

of sudden failures. These findings emphasized the importance of combining temporal trends with component-specific failure patterns for more accurate predictions.

The EDA also highlighted operational shifts that occur within minutes before a failure, as captured in both 10-minute data and PCA visualizations. These shifts, such as abrupt changes in temperature or rotational speed, can serve as early indicators of impending faults, enabling preemptive maintenance.

EDA provided a robust foundation for understanding turbine operations and failure dynamics. The granularity analysis emphasized the complementary roles of 10-minute and daily data in capturing anomalies and trends, respectively. The failure analysis identified high-risk components and turbines, while PCA visualizations uncovered distinct behavioral patterns before and after failures. These insights shaped the design of predictive models and informed the development of actionable maintenance strategies.

5 METHODOLOGY

5.1 TURBINE-LEVEL MODELING WITH CORE FORECASTING ARCHITECTURES

The first experimental setup was done with a turbine level modeling i.e. models will be trained separately for every turbine's data. This setup was done to understand on a lower level which core forecasting model would be most suitable for the task. The most suitable models can then be used to develop generic model for all the turbines.

5.1.1 CORE FORECASTING ARCHITECTURES:

Random Forest
Random Forest is an ensemble learning method that builds multiple decision trees using bootstrapped samples and aggregates their predictions through majority voting (for classification) or averaging (for regression). This makes it robust to overfitting and suitable for high-dimensional datasets, especially when the number of features is similar to the number of data points. A key advantage is its ability to provide feature importance, helping identify critical signals or conditions contributing to failures. It does not require feature scaling or normalization, simplifying preprocessing. However, it can be computationally expensive with large datasets or many trees and lacks interpretability at the individual tree level.
Random Forest is ideal for robust and accurate predictions across turbines with varying operational characteristics. It performs well on datasets with missing values and mixed data types without extensive preprocessing. Scaling and PCA are unnecessary, preserving data interpretability and feature relations.

XGBoost
XGBoost is a highly efficient implementation of gradient boosting that builds trees sequentially, with each tree correcting the errors of the previous ones. It includes advanced features like L1 and L2 regularization to prevent overfitting, making it effective for high-dimensional datasets such as those used in wind turbine failure forecasting. XGBoost excels in predictive accuracy and computational efficiency, especially with large datasets, and handles missing values internally while supporting parallel processing, making it suitable for real-time applications.
Despite its strengths, XGBoost requires careful hyperparameter tuning (e.g., learning rate, max depth, subsampling ratios) to achieve optimal performance. It is more computationally intensive compared to simpler models like Random Forest or AdaBoost. XGBoost is best suited for scenarios where high accuracy is critical, particularly in imbalanced datasets where minimizing false negatives (missed failures) is essential. Scaling is recommended for faster convergence during optimization, but PCA is optional and should be used if dimensionality reduction significantly improves computational efficiency.

AdaBoost
AdaBoost (Adaptive Boosting) combines multiple weak learners, typically decision stumps, to create a strong learner by iteratively adjusting weights on misclassified samples. This makes it effective in identifying subtle patterns, such as those leading to turbine failures. Its simplicity and ability to enhance weak classifiers are key advantages. However, AdaBoost is sensitive to noise and outliers, as it increases the weight of misclassified samples during training, and may not scale well to large datasets compared to Random Forest or XGBoost.
AdaBoost is most effective on clean, moderately sized datasets. It is suitable when interpretability

of weak learners is desired or when simpler models suffice for capturing failure patterns in turbines. Scaling is unnecessary since AdaBoost uses decision trees that are insensitive to feature magnitudes. PCA may be considered for computational efficiency if the dataset has extremely high dimensionality but is generally unnecessary.

5.1.2 SETUP AND DATA PROCESSING:

1. **Converting data to 1-day frequency:**
   The 10-minute frequency data is converted to a daily frequency data. This is done to eliminate any noise in the data and diminish unnecessary periodicity in the data. The wind turbines would not operate at optimal efficiencies 24 hours a day. Different times during a day will see different nature in the wind speeds and directions, thus affecting the readings of turbine sensors and the operating conditions of the turbines in general which do not directly associate with failure in the turbine. Also, there might be downtimes within a day where the turbine is not operating. Using a daily frequency data will eliminate these issues.
2. **Maintaining the intrinsic properties of features while changing frequencies:**
   We simply do not compute mean of the data while converting to daily frequency. Most features are either minimum, maximum, average and standar deviation readings of a property like rotor speed, wind speed, temperatures etc. in a certain duration. So, these natures of the features are maintained while converting the data from 10-minute frequency to 1-day frequency.
3. **Rolling averages and lagged features:**
   Rolling window averages over a week and lagged features are introduced to mitigate periodic nature in the data and also have a relevence of historical sensors that lead to failures. This feature engineering leads to a lot of new features, hence architectures that can handle large feature size are crucial for this task.
4. **Target encoding:**
   Any data records on the day of failure and the previous 30 days are encoded with the target failure = 1. And the rest of the data points are encoded with failure = 0. This 30 days period is called as a failure window.
5. **Output interpreted as a failure warning:**
   Naturally, the forecasted output from the models will also be 0's or 1's. But we need failure warning before a potential failure. So the output objective have to be modified to be interpreted as a confident failure warning. And then these failure warnings are evaluated as correct failure warnings, incorrect failure warnings and early failure warnings.

5.1.3 INTERPRETING FAILURE WARNINGS:

Confident failure warnings:
Since the total failure window is 30 days, any failure flag (y_pred = 1) of 6+ consecutive days can be considered as a confident failure warning.
If we receive a confident failure warning, the failure flags raised in the following 25 - 30 days can be considered a part of the same failure warning and a new warning is not raised.
A small perturbation in the y_pred string in any direction, $0 \rightarrow 1$ or $1 \rightarrow 0$ can be ignored as an anomaly and that data point's prediction can be considered same as the previous continuous sequence.

As illustrated in Figures 6, 8 the numbers show the data points that make up a confident failure warning and which failure warning do they belong to. As seen in Figure 8, the positive failure flags after the second failure warning are also considered part of that failure warning, since they are generated within a 25-30 day duration from a confident failure warning. The skip symbol indicates that these predictions can be considered anomalous and part of the continuous sequence, which raises a second failure warning in figure 6

Correct failure warning: Major part of the predicted failure warning window coincides with the true failure warning window. Shown with a green symbol in fig 8
Incorrect failure warning: There is no true failure window present during or just after the predicted failure warning window. Shown with a red symbol in fig 8

Early failure warning: There is a failure window predicted just before a true failure window and the overlap is very small. Shown with an orange symbol in fig 8

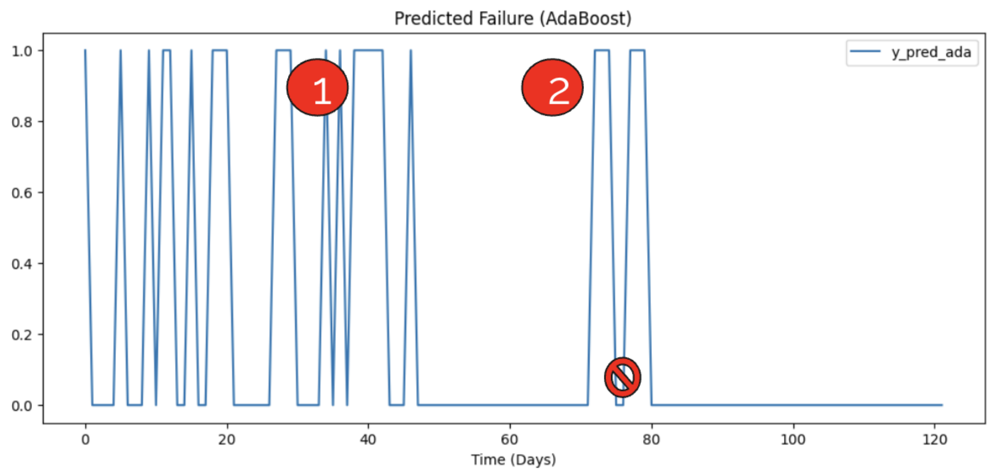


Figure 6: Failure Warning Interpretations

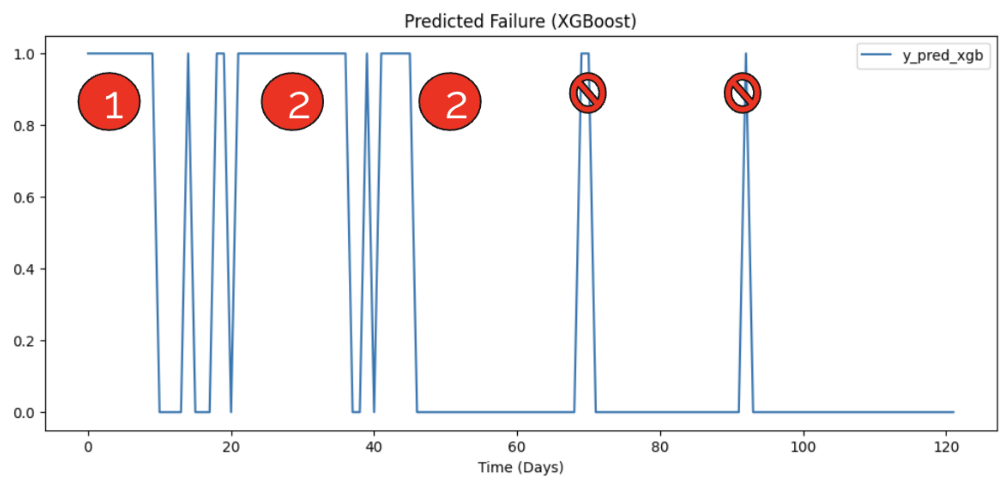
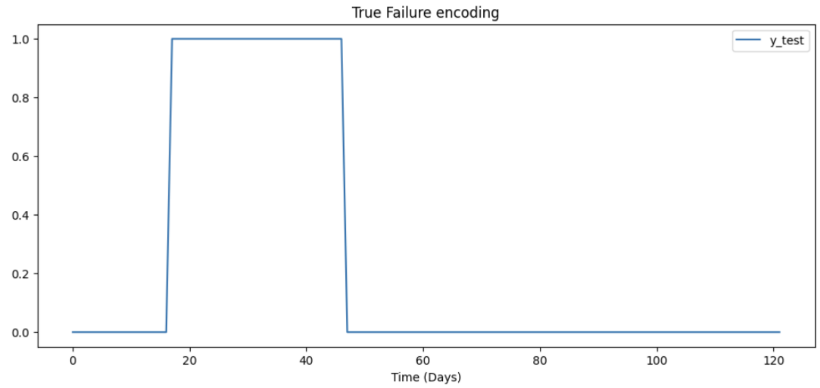


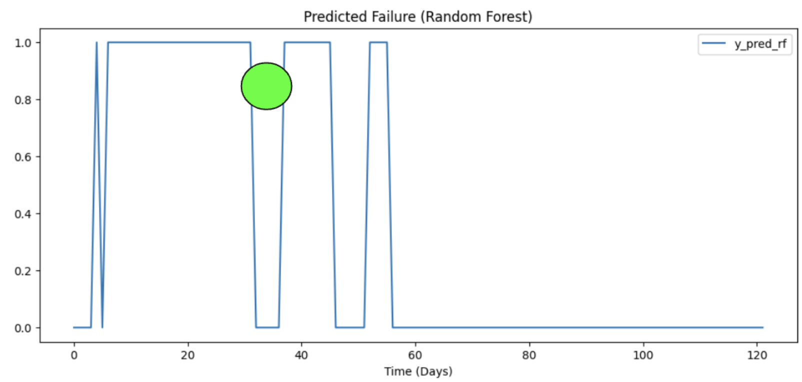
Figure 7: Failure Warning Interpretations

5.2 Cumulative sum anomaly detection(3)

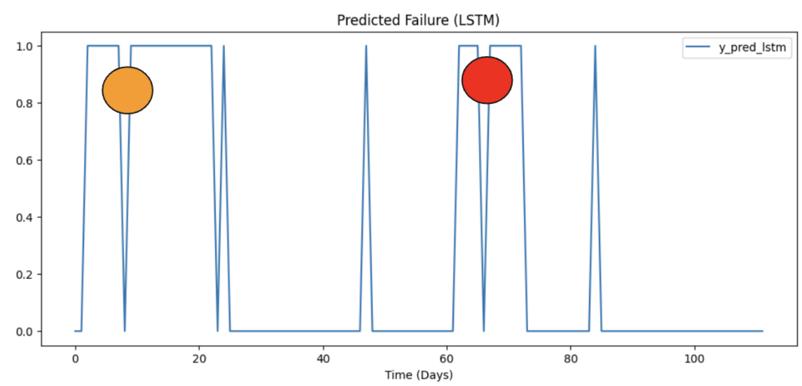
The method has already been published, and we want to summarize it in this project. The method is an unsupervised approach that calculates multivariate anomaly scores using a Minimum Spanning Tree (MST)-based technique to identify anomalies in high-dimensional SCADA data by approximating geodesic distances. These anomaly scores are accumulated over a fixed window while subtracting an offset value to reduce false alarms, ensuring only significant anomalies are tracked. A control limit (threshold) is determined from training data in an unsupervised manner to balance detection performance and economic factors, such as the costs of missed detections versus false positives. The accumulated anomaly scores are compared against this control limit, and when the score exceeds the limit, an alarm is triggered, enabling early detection of failures with a lead time of several days to weeks. This approach combines unsupervised symptom accumulation, similar to a CUSUM method, with MST-based anomaly detection to provide timely warnings for wind turbine failures. The performance of this method on a specific type of failure over all five turbines is exhibited in Figure9.



(a) True values



(b) Predicted values



(c) Predicted values

Figure 8: Failure warnings
Green shows correct failure warning, red shows incorrect failure warning and orange shows early failure warning

This method is capable of flagging all failures with a lead time of up to 90 days before each failure, which is one of its key strengths. However, as shown in the plot, there are multiple flags (represented by black solid horizontal lines) that do not correspond to actual failures, resulting in false positives. This limitation is a significant drawback of the current approach, as false positives occur frequently. It is anticipated that machine learning-based models can address this issue more effectively, reducing false positives while maintaining early detection capabilities.

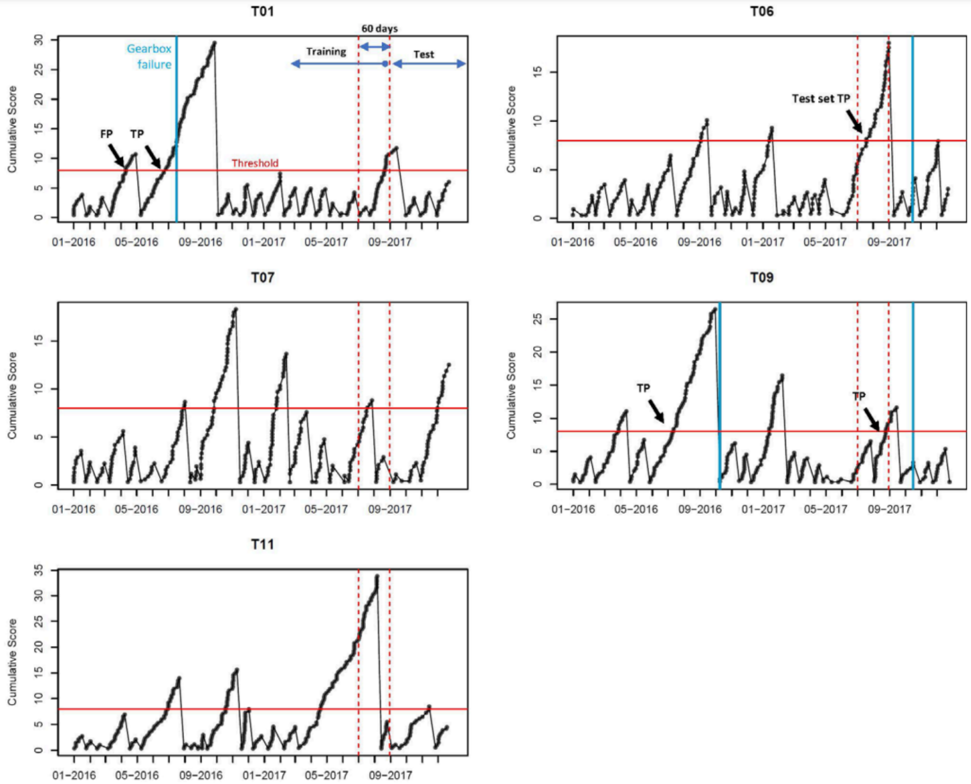


Figure 9: Cusum results over multiple turbines

5.3 Random forest with post-processing on 10-min data frequency

The following is a list of remarks on this project:

- There is often a gap between predicted output and effective alarm flagging.
- It is easier to train machine learning models on specific types of failures (e.g., hydraulic group) rather than a generalized failure model.
- Data should reveal signs of failure before the event, and these signs should be consistent over timestamps.
- Balanced datasets (e.g., 50/50) help improve prediction performance, and training/testing datasets should have the same length for better interpretability.

This methodology focuses on leveraging historical data to identify patterns that precede failure events. The approach includes the following steps:

Data Preparation: Historical data for a specific period (l days) before known failure events is divided into two parts:

- The first $l/2$ days are labeled as `y=0` (non-failure).
- The last $l/2$ days are labeled as `y=1` (failure).

It is illustrated in Figure 10

Rolling Window Evaluation: Testing data is evaluated using a rolling window framework, where the test dataset is divided into consecutive windows, each analyzed for patterns of failures which is illustrated in Figure 11

Separation Performance Index (SPI): To determine if a testing window should be flagged, the SPI metric is computed as:

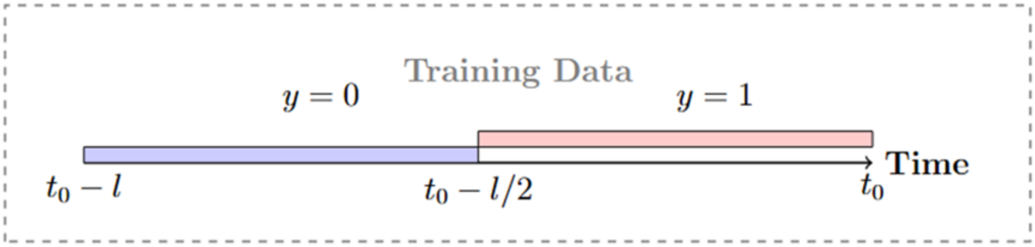


Figure 10: Failure Preprocessing

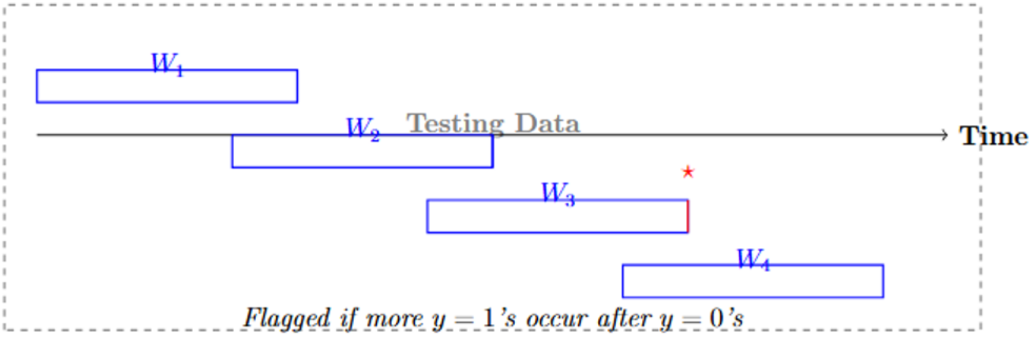


Figure 11: Rolling window evaluation framework

$$m = \left\lfloor \frac{n}{2} \right\rfloor$$

$$y_{\text{pred,first}} = [y_1, y_2, \ldots, y_m]$$

$$y_{\text{pred,second}} = [y_{m+1}, y_{m+2}, \ldots, y_n]$$

$$\text{Zeros in First Half} = \sum_{i=1}^{m} I(y_i = 0)$$

$$\text{Ones in First Half} = \sum_{i=1}^{m} I(y_i = 1)$$

$$\text{First Half Ratio} = \frac{\text{Zeros in First Half} - \text{Ones in First Half}}{m}$$

$$\text{Zeros in Second Half} = \sum_{i=m+1}^{n} I(y_i = 0)$$

$$\text{Ones in Second Half} = \sum_{i=m+1}^{n} I(y_i = 1)$$

$$\text{Second Half Ratio} = \frac{\text{Ones in Second Half} - \text{Zeros in Second Half}}{n - m}$$

$$\text{Metric}(W_i) = \text{First Half Ratio} + \text{Second Half Ratio}$$

$$\text{Flag if Metric}(W_i) \geq \text{Threshold}$$

The window is flagged if the metric exceeds a defined threshold. It is exhibited with an example in Figure 12

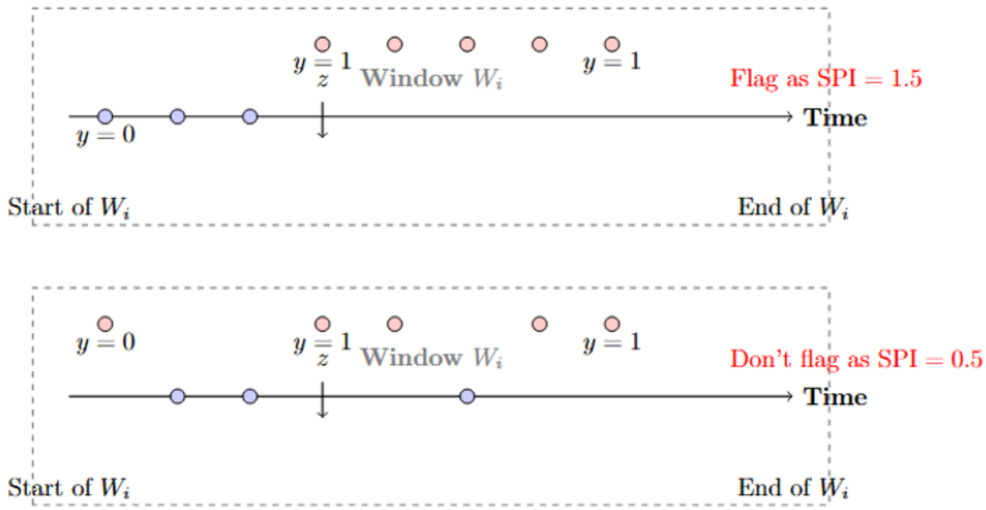


Figure 12: Separation Performance Index (SPI): For instance, if the threshold is 0.8, the top one is flagged as failure, but the bottom one is not.

5.4 Multidimensional Anomaly Detection and Interpretation (MADI)

The Multidimensional Anomaly Detection and Interpretation (MADI) framework is designed to detect anomalies in high-dimensional data by leveraging negative sampling (5). The methodology begins with defining the observed data, representing normal behavior, as the positive class. A critical component of MADI is the generation of synthetic negative samples, which are uniformly distributed outside the bounds of the observed data. These samples are separated from the positive data by a margin δ, ensuring they represent potential anomalies while minimizing overlap with the normal data.

A binary classifier is then trained using these positive and negative samples to learn the decision boundary between normal and anomalous regions. The ratio of negative to positive samples, denoted as `sample_ratio`, is a key parameter that balances the sensitivity of the model. The classifier uses a loss function, such as binary cross-entropy, to minimize classification errors during training. Once trained, the model predicts anomalies by identifying data points that fall into regions classified as negative.

This approach enables robust anomaly detection without requiring a comprehensive labeled dataset for anomalies. Furthermore, the high dimensionality of the data is leveraged through the concentration phenomenon, which ensures that anomalies are sparse and likely to fall outside the normal data distribution. The MADI framework is computationally efficient and scalable, making it suitable for real-time applications, such as wind turbine anomaly detection.

5.5 Long Short-Term Memory (LSTM) Networks

The Long Short-Term Memory (LSTM) network is employed to model temporal dependencies in sequential data for turbine failure prediction. The input to the LSTM model consists of sequences of sensor readings spanning 24 time steps, corresponding to a sliding window of hourly measurements. Each sequence includes multiple normalized features representing the operational state of the turbine. The LSTM architecture is designed to capture temporal patterns in the data that indicate potential failures.

The LSTM network used in this study consists of two stacked LSTM layers, followed by dropout layers to mitigate overfitting. Each LSTM layer comprises 64 and 32 units, respectively, and uses the ReLU activation function. The dropout rate is set at 0.2 to ensure regularization. A dense output layer with a sigmoid activation function produces a probability score, representing the likelihood of turbine failure.

The network is trained using the binary cross-entropy loss function, optimized with the RMSprop optimizer for efficient weight updates. Early stopping is employed to prevent overfitting by monitoring validation loss and restoring the best weights after training stabilizes. To address the class imbalance in the dataset, various techniques, such as class weighting and oversampling, are incorporated during training.

The LSTM model processes input sequences and predicts failures based on learned patterns in the sensor data. Its ability to handle long-term dependencies makes it particularly effective for failure prediction tasks, where historical context is crucial. The results from the LSTM model are evaluated using metrics such as precision, recall, F1-score, and ROC-AUC, providing a comprehensive assessment of its performance.

6 Experimental Results

6.1 Turbine-level modeling with core forecasting architectures

In the following images of fig 13 and 14 we see the results of the 3 forecasting models Random Forest, XGBoost and AdaBoost on turbines T06 and T11. We can see that Random Forest performs better than the other two architectures. Random Forest outperforms other two in standard metrics like accuracy and f1 score and it also correctly predicts the failure warnings with minimal early warnings and incorrect warnings.

But this result is not consistent with all the turbines. In some turbines XGBoost might perform better than Random Forest or Random Forest may be better but wont predict all the failures. But overall Random Forest gave the better results, and it can be used for the generalized models which are trained on all turbines.

Also the data we had was very small to test the sensitivity of these architectures and to test whether the performance of these models changes with a change in the training and testing timespans.

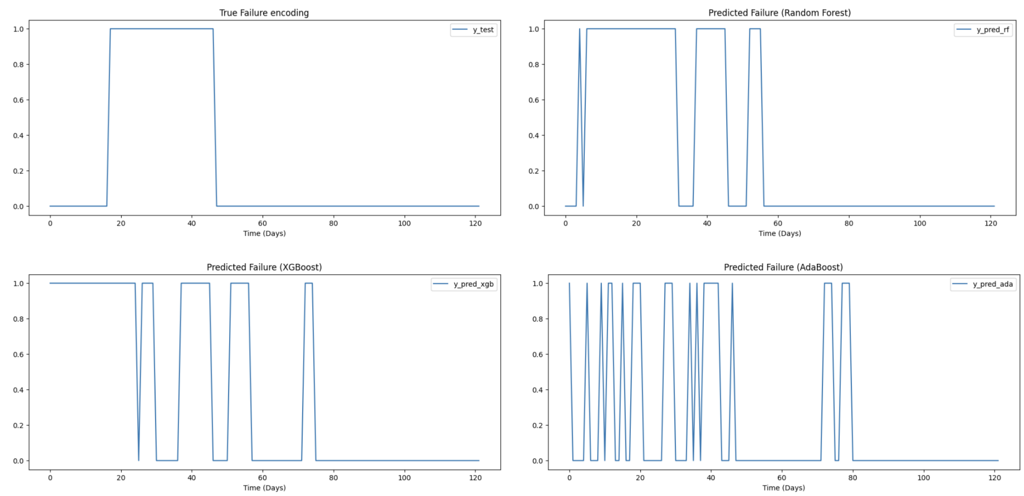
6.2 Random forest with post-processing on 10-min data frequency

Results for Turbine T09: The model detected three actual hydraulic failures with a lead time of 1–2 months, trained on data from turbine `T06`. 15

Results for Turbine T07: When applied to oil leakage failures, the model identified 3 out of 4 failure events with a look-back period of 120 days and PCA capturing 90% variance. This is shown in Figure 16.

PCA Sensitivity Analysis: Sensitivity analysis showed optimal performance metrics (accuracy, precision, recall, F1-score) at 90% PCA variance retention. 17

Classifier Comparison: Performance across various classifiers (Random Forest, Logistic Regression, SVM, Decision Tree, K-Nearest Neighbors) was evaluated, showing comparable accuracy and precision. 18 It worths mentioning that they all flagged the failures that random forest flagged, but the SVM and Logistic regression performed the best.

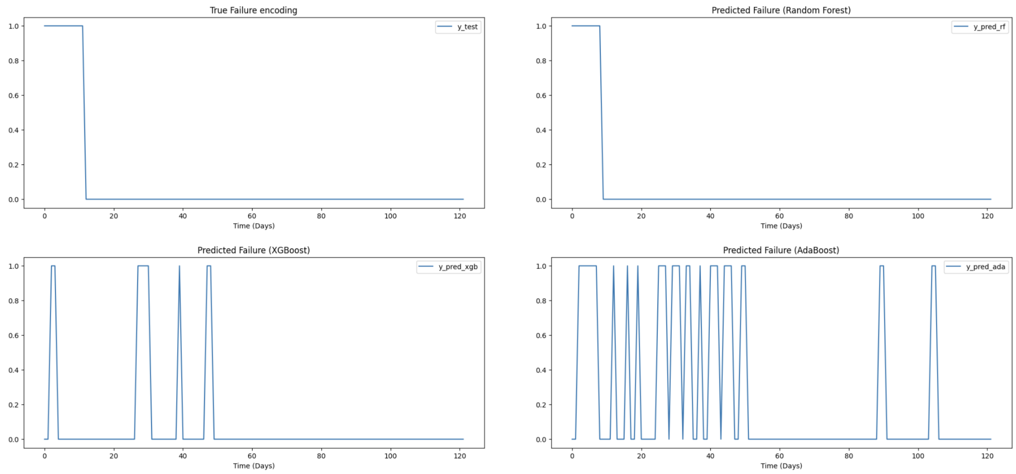


(a) True values and predictions of 3 models on T06

Model	Accuracy	Precision	Recall	F1 Score	Correct Warnings	Early Warnings	Incorrect Warnings
Random forest	0.82	0.6	0.8	0.68	1 / 1	1	0
XGBoost	0.71	0.45	0.70	0.55	1 / 1	1	1
AdaBoost	0.77	0.54	0.47	0.5	1 / 1	0	1

(b) Performance comparision for T06

Figure 13: Results of Turbine-Level Modeling for T06



(a) True values and predictions of 3 models on T11

Model	Accuracy	Precision	Recall	F1 Score	Correct Warnings	Early Warnings	Incorrect Warnings
Random forest	0.98	1.0	0.75	0.86	1 / 1	0	0
XGBoost	0.86	0.22	0.17	0.19	0 / 1	0	1
AdaBoost	0.75	0.2	0.5	0.29	1 / 1	0	1

(b) Performance comparision for T11

Figure 14: Results of Turbine-Level Modeling for T11

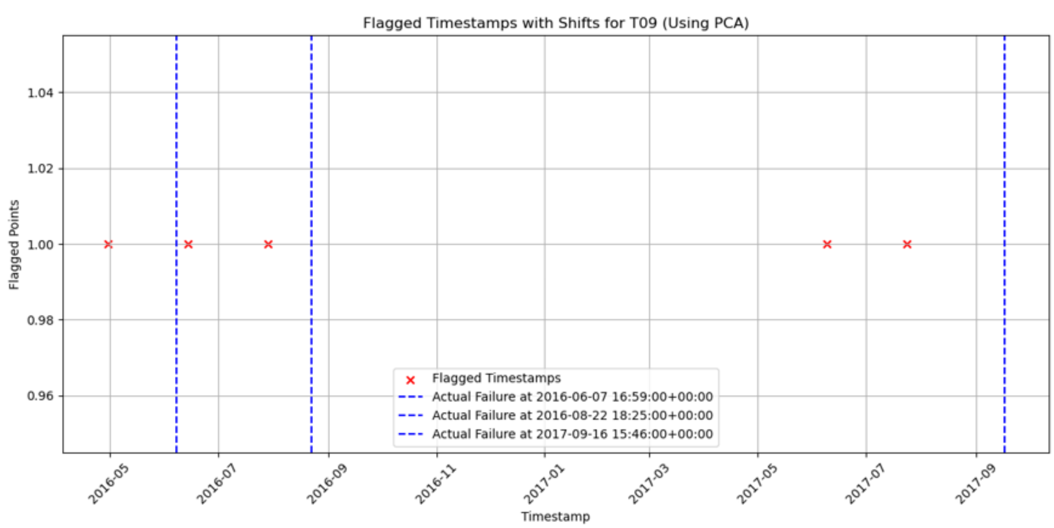


Figure 15: Results for turbine T09 on high temperature failures

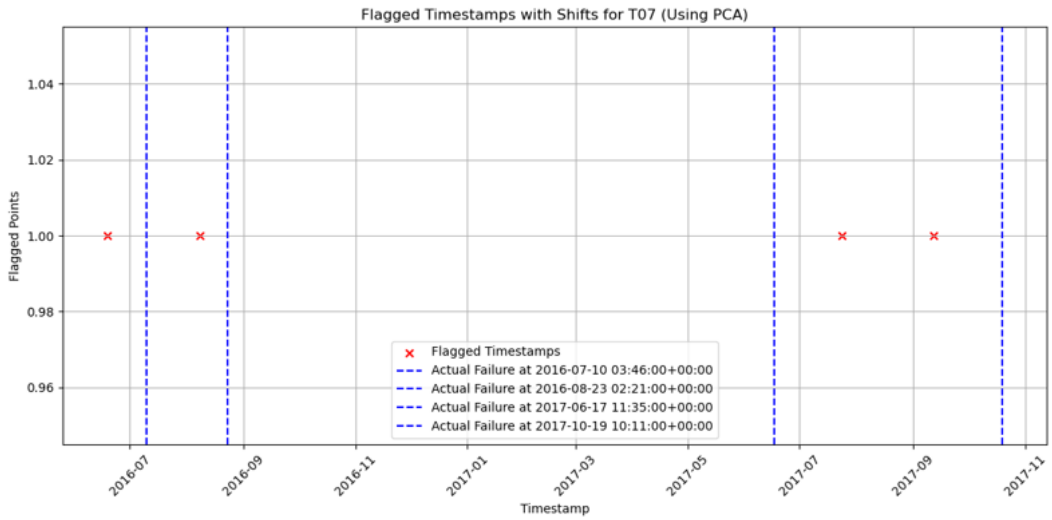


Figure 16: Results for turbine T07 on oil leakage

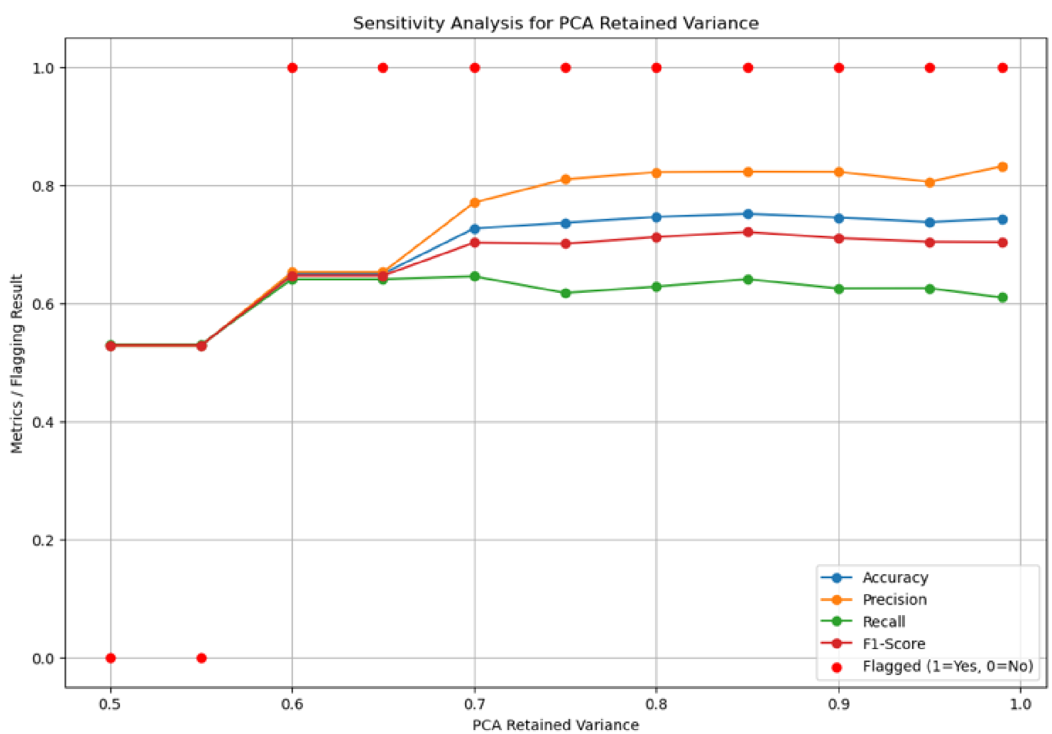


Figure 17: PCA sensitivity analysis

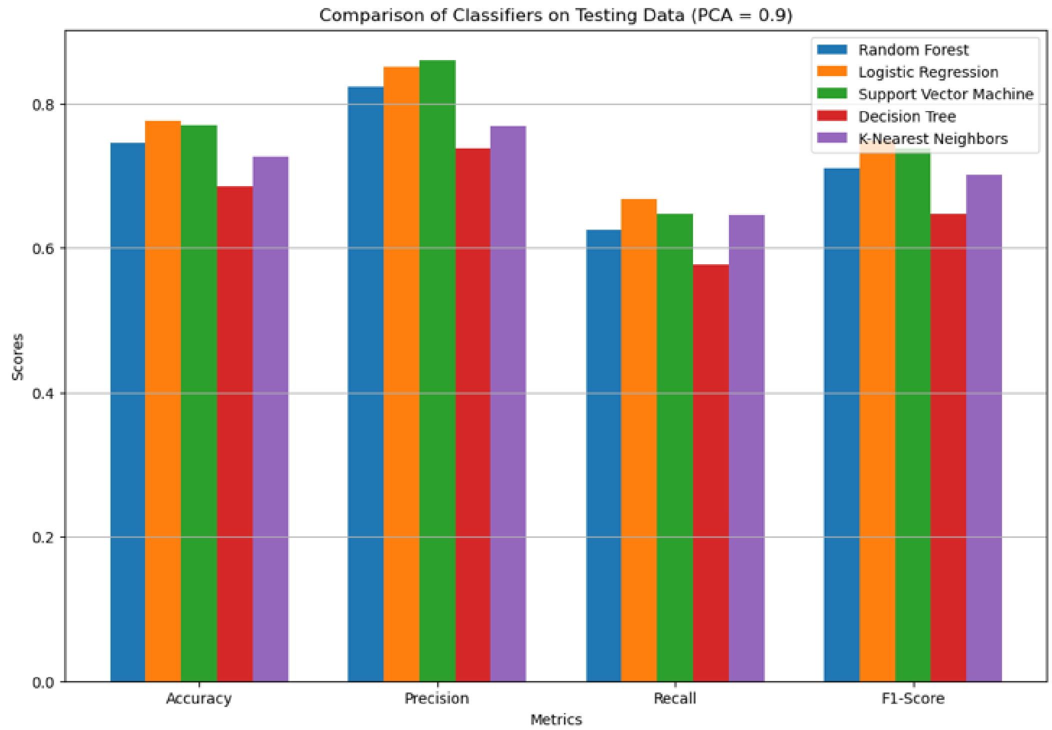


Figure 18: Classifier performance comparison

The methodology demonstrates strong predictive performance for failure events, with high accuracy across multiple types of failures and turbines. This method only shown one missing alarm on the third failure of T07, but captured all other failures within 30 or 90 days prior period. This shows the superiority of this approach over the cumulative sum method.

6.3 LSTM

The Long Short-Term Memory (LSTM) model demonstrated high accuracy but struggled with the detection of anomalies due to class imbalance challenges. The training and validation loss curves are shown in Figure 19, highlighting the overfitting behavior as the validation loss increases significantly after 20 epochs. Despite the high overall accuracy of 73%, the recall for the minority class (anomalies) was only 7%, as shown in Table 1. This indicates the model's limitation in detecting turbine failures effectively.

The confusion matrix further confirms this challenge, with a large number of false negatives (15634). The high precision for the normal class (80%) compared to the anomaly class (13%) emphasizes the imbalance issue. Future work may involve applying oversampling, undersampling, or other class-balancing techniques to address this limitation.

Class	Precision	Recall	F1-score	Support
Normal (0)	0.80	0.89	0.84	70672
Anomaly (1)	0.13	0.07	0.09	16847
Accuracy		0.73		
Macro avg	0.47	0.48	0.47	87519
Weighted avg	0.67	0.73	0.70	87519

Table 1: Performance metrics for the LSTM model.

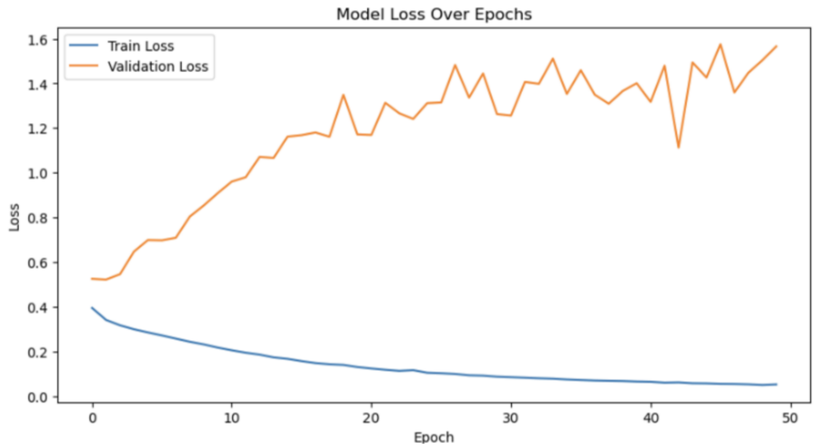


Figure 19: Training and validation loss for the LSTM model over 50 epochs.

6.4 MADI

The Multidimensional Anomaly Detection and Interpretation (MADI) framework was evaluated using AUC (Area Under the Curve) metrics, as shown in Figure 20. The average AUC for anomaly detection across multiple scenarios was 21.60%, reflecting the difficulty in identifying anomalies in the high-dimensional wind turbine data.

The execution times for MADI ranged between 9.5 and 10.5 seconds, demonstrating its computational efficiency. Despite the lower AUC values, the method holds potential for improvement with enhanced feature engineering and more sophisticated sampling strategies. These results highlight the challenges of anomaly detection in sparse and complex datasets, where defining the boundaries between normal and anomalous behavior is inherently difficult.

7 Conclusion

Comparing the cumulative sum approach with the Random Forest model developed by our team, the cumulative sum method exhibits a significant number of false positives, making it less reliable for practical applications. In contrast, the Random Forest model, coupled with post-processing techniques, successfully eliminates false positives entirely. However, it does miss one of the failures, highlighting a trade-off between precision and recall in the detection process. This outcome demonstrates the improved robustness of the Random Forest model while also pointing to potential areas for further refinement to achieve comprehensive failure detection. A promising future direction could involve incorporating temporal dependencies into the labeling process. Instead of binary labels (0 and 1), a backward integer labeling scheme could be implemented, where labels represent the number of days or intervals remaining until a failure. This approach would better capture the temporal progression of failure signals, potentially enhancing the model's ability to learn and predict failure trends over time.

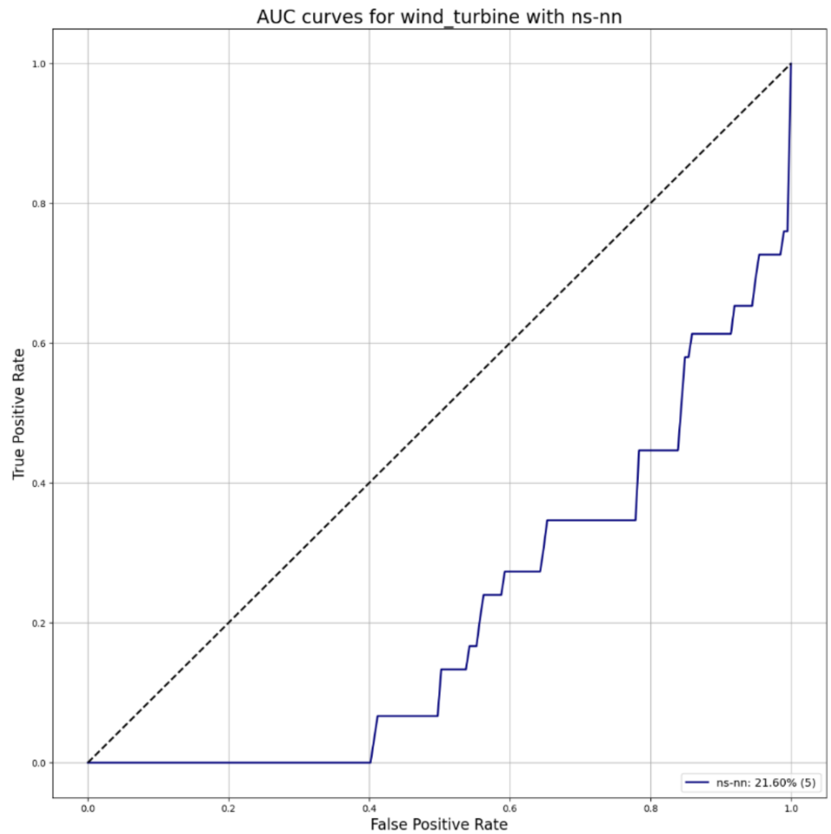


Figure 20: AUC curves for wind turbine anomaly detection using MADI.

References

[1] P. Chatterjee and L. Ross. Statistical process control and machine learning for wind turbine fault diagnosis. *Energy and AI*, 8:104–116, 2022.

[2] J. Doe and A. Smith. Predictive maintenance for renewable energy systems. *Journal of Industrial Engineering*, 2023.

[3] Effi Latiffianti, Shawn Sheng, and Yu Ding. Wind turbine gearbox failure detection through cumulative sum of multivariate time series data. *Frontiers in Energy Research*, 10, 2022.

[4] K. Lee and Y. Zhang. Fault detection in wind turbines. *Energy Systems*, 2021.

[5] John Sipple. Interpretable, multidimensional, multimodal anomaly detection with negative sampling for detection of device failure. *arXiv preprint arXiv:2007.10088*, 2020.

[6] H. Turner and M. Watson. Deep learning approaches for predictive maintenance of offshore wind turbines. *Renewable Energy*, 75:865–880, 2021.