

A Real-Time IoT, LLM, AI-Supported Wind Turbine Failure Prediction System

İlhan Gamze Saygi¹, Sara Mahyanbakhshayesh¹, Berdan Ak¹, Mehmet Hilal Özcanhan², Mehmet Süleyman Ünlütürk³

¹Department of Computer Engineering, Dokuz Eylul University, Izmir, 35390, Turkey

²Department of Computer Engineering, Dokuz Eylul University, Izmir, 35390, Turkey

³Department of Software Engineering, Yaşar University, Izmir 35390, Turkey

Correspondence to: Sara Mahyanbakhshayesh (saramahyan.smb@gmail.com), İlhan Gamze Saygi (ilhangamzesaygi@gmail.com)

Abstract. Wind and solar energy are two popular alternative energy sources. However, wind turbines are larger and more complex than solar panels. Accordingly, wind turbines are more exposed to environmental factors and therefore more prone to mechanical failures. Our work improves the reliability and efficiency of wind energy systems by presenting an artificial intelligence (AI)-based system that predicts mechanical gearbox and electrical failures. Historical data are aggregated with real-time sensor data to train the prediction model. Apart from related mechanical and environmental data, sensors provide real-time vibration, internal nacelle temperature, wind speed, noise and smoke levels. The developed system integrates AI and Large Language Model (LLM)-based interfaces for real-time interactive monitoring of turbines. The user interface of the developed system allows users to receive informative responses on performance, detected risks, predicted failures, and energy production levels. The developed model has been validated using 5-fold cross-validation based on *Accuracy*, *Precision*, *Recall*, *F1-Score*, and *ROC-AUC*. The model achieves approximately 89.68% *Accuracy*, 90.08% *F1-Score*, 95.65% *ROC-AUC*, and novel metric 65.13%, *Overall Performance*. The performance results demonstrate the promising potential of AI- and LLM-integrated systems for wind energy applications. Prototype data, labeled via the XGBoost model trained with SCADA data, was retrained using the LightGBM algorithm, achieving 98.37% *Accuracy*, 99.16% *F1-Score*, 98.95% *ROC-AUC* and 94.65% *Overall Performance*; the analysis indicated that the inclusion of the newly added gas and sound sensors improved the predictive performance of the student model on the pseudo-labeled prototype dataset.

1 Introduction

1.1 Background

Maintaining the reliability of wind turbines is a significant engineering challenge due to their constant exposure to harsh environmental conditions. Since traditional reactive and periodic maintenance methods are costly and operationally inefficient, Artificial Intelligence (AI) and Machine Learning (ML) based predictive maintenance has emerged as a crucial solution. By analysing real-time sensor data, these systems can detect early failure patterns in critical components, such as gearboxes and electrical systems, preventing costly downtime.

Supporting this transition, research demonstrates that combining physical thermal models with ML significantly improves fault prediction and explainability (Corley, 2021; Khan, 2023). For instance, large-scale analyses using decision tree methods on bearing and lubricant temperatures have achieved approximately 92% accuracy (Hsu, 2020). Furthermore, integrating Large Language Models (LLMs) has transformed traditional, static monitoring dashboards into interactive AI systems. Recent models, such as Generative Pre-trained Transformer (GPT)-4 and DeepSeek-V3, have demonstrated strong performance in generating structured classification workflows and synthetic datasets for fault diagnosis (Tan, 2025; Zhang, 2025). By allowing operators to query complex data in natural language, these multi-modal frameworks significantly enhance the scalability, predictive accuracy, and overall intelligence of modern energy monitoring systems.

1.2 Problem Statement

Wind turbines routinely face unexpected mechanical and electrical failures that disrupt production and drive maintenance costs upward. Existing Supervisory Control and Data Acquisition (SCADA)-based threshold methods offer limited diagnostic capability because they are unable to interpret complex correlations across heterogeneous sensor inputs. Consequently, modern predictive maintenance systems must incorporate AI-based analytics, Internet of Things (IoT) connectivity, and transparent explainability tools to achieve reliable and early fault prediction. Consequently, AI-driven, IoT-supported, and explainable predictive maintenance systems are required instead of reactive monitoring approaches.

The remainder of this study is organized as follows. Section 2 reviews the related literature and establishes our contributions relative to the state of the art. Section 3 details the materials, data preprocessing pipelines, and the system architecture of the Smart Wind Turbine Failure Prediction System (SWTFPS). Sect. 4 describes the interactive, LLM-based user interface and communication integration components. Sect. 5 presents the experimental results and a comparative model performance analysis. Sect. 6 provides a comprehensive discussion of the findings and outlines system limitations, followed by Sect. 7, which concludes the study and suggests directions for future work.

To ensure a cohesive and deeply developed scientific narrative, this study is systematically structured around two primary, interconnected core contributions rather than disconnected engineering steps: First, a **Multi-Modal Real-Time Predictive Engine** was established that bridges large-scale historical industrial SCADA datasets with live, hardware-level IoT telemetry. It is demonstrated how the predictive precision of wind turbine failure diagnostics is dramatically improved by transferring machine learning decision boundaries from industrial scales to a localized physical prototype and enriching this pipeline with newly introduced gas and acoustic sensors. Second, a **Live Conversational Decision-Support Framework** is introduced that integrates this predictive engine with an interactive, domain-grounded Large Language Model (LLM). This system successfully translates complex, high-frequency multi-modal sensor states into explainable, actionable natural language recommendations for operators. Every subsequent section of this manuscript directly supports these two pillars: Sect. 3 and Sect. 5 focus on the mathematical validation, feature selection, and transfer accuracy of the predictive engine, while Sect. 4 and Sect. 6 detail the integration, real-time communication protocols, and interactive utility of the explainable conversational interface.

2 Related Works

Another study by Zhang et al. (2023) proposed a convolutional network, a multi-scale temporal convolutional network to balance the data, achieving 93.4% accuracy. Amin et al. (2022) combined cyclostationary analysis with a Convolutional Neural Network (CNN) to detect gearbox faults using generated data, achieving 89% accuracy even under variable wind conditions. In another study, a SCADA-based system was developed. Using neural networks for temperature anomaly detection in gearboxes and generators, failures were predicted up to 17 hours in advance with high accuracy (Alzawaideh, 2021). A prior study evaluated 16 supervised models on SCADA data, identifying Random Forest (RF) and Extra Trees (ET) as the best performers for detecting faults 24 hours in advance (de Lima Munguba, 2024). Inturi et al. (2021) applied multi-level data fusion of vibration, acoustic, and oil sensors with system, achieving 92% accuracy in gearbox fault classification. Meyer et al. (2022) developed a fully automated CNN approach using vibration spectrograms, achieving 100% accuracy on laboratory data without manual feature engineering. Ogaili et al. (2024) used an eXtreme Gradient Boosting (XGBoost) classifier to detect blade faults, such as erosion and twisting, with 99.4% accuracy. Pham and Han (2025) designed a Bi-LSTM (Bidirectional Long Short-Term Memory) model with an Attention Mechanism, achieving 99.40% accuracy and providing 62-hour early warnings. Puruncajas et al. (2025) developed a normal behaviour model using SCADA temperature data, detecting gearbox anomalies 9 months before failure. Santiago et al. (2024) reviewed deep learning models for predictive maintenance, highlighting a research gap in their application to direct-drive and synchronous generators systems. Yu et al. (2021) proposed a convolutional network using wavelet packet coefficients, achieving over 99% accuracy in noisy environments. Zeng et al. (2023) introduced an estimation model to establish dynamic thresholds for anomaly detection, outperforming standard alarm systems. Zhang et al. (2023) combined CNN and (Long Short-Term Memory) LSTM to predict bearing temperatures from SCADA data, detecting anomalies earlier than single-model approaches. Finally, Zhang et al. (2014) used an Artificial Neural Network (ANN) with backpropagation on SCADA data to predict bearing temperatures, issuing warnings 3 months in advance without additional sensors.

The existing literature extensively utilises SCADA data and various deep learning architectures (CNNs, LSTMs, Transformers) for vibration analysis. However, significant gaps remain in integrating user interaction and failure prediction within a single system, because most studies rely on single-source data and present results through static dashboards rather than interactive interfaces. While pioneering studies have recently explored the utility of Large Language Models (LLMs) in wind energy systems (for instance, Tan and Carroll (2025) leverage LLMs to optimize gearbox failure prediction, and Zhang et al. (2025) propose multi-modal training frameworks for fault diagnosis) their applications are primarily restricted to offline environments, static datasets, or synthetic data generation. A major research gap persists in physically bridging live, multi-modal sensor telemetry with conversational interfaces for real-time, closed-loop operations. To address this, our SWTFPS introduces a complete, end-to-end operational integration. Unlike prior frameworks that rely on static database queries or offline diagnostic workflows, our system directly couples a physical wind turbine prototype's multi-modal IoT sensor network with cloud-hosted databases, automated high-performance ML classifiers (XGBoost/LightGBM), and an

interactive LLM-based assistant. Thus, our core novelty lies in the real-time, synchronized execution of physical edge telemetry, immediate predictive diagnostic modeling, and conversational reasoning within a single, unified monitoring platform. To address these gaps, this study delivers three key scientific and technical contributions:

- 1. Multi-Modal Feature Enrichment and Domain Adaptation Strategy:** We introduce a systematic framework to bridge industrial-scale, historical SCADA telemetry with localized IoT prototype data. We demonstrate that transferring predictive boundaries from a SCADA-trained model to a localized prototype, while enriching the model input space with previously unexploited localized sensors (acoustic and gas), systematically refines fault prediction capabilities under highly imbalanced operating conditions.
- 2. Domain-Grounded Conversational Decision-Support:** We established a novel integration of real-time machine learning predictive engines with an interactive, natural language Large Language Model (LLM) interface. This advances traditional static dashboards by transforming high-frequency multi-modal sensor inputs and failure probabilities into explainable, context-aware, and actionable maintenance recommendations for operators.
- 3. Rigorous Empirical Benchmarking and Evaluation Metric:** We conduct a comprehensive performance evaluation of eleven diverse machine learning algorithms. Furthermore, we propose and validate a novel *Overall Performance (OP)* metric that integrates both positive predictive strengths and uncertainty/error penalties (such as *LogLoss* and *Brier Error*), providing a highly balanced evaluation framework for predictive maintenance systems operating on imbalanced datasets.

3 Materials and Methods: System Architecture

The methodological framework of SWTFPS is built upon three phases: (1) the construction of **hardware** infrastructure equipped with multi-modal sensors for real-time data acquisition; (2) the development and evaluation of **machine learning models** to detect mechanical and electrical anomalies and (3) the integration of a **Large Language Model (LLM)** interface to facilitate interactive decision support and data visualization.

3.1 Materials: Custom-made Small Wind Turbine Prototype

A prototype has been designed and constructed to add newly available sensor data to traditional SCADA data. The prototype shown in Figure 1 consists of six main components:

1. The primary mechanical *Vertical Axis Wind Turbine (VAWT)* captures wind energy.
2. An *anemometer* measures real-time wind speed.
3. A *wind vane* monitors wind direction.
4. The *structural frame chassis* supports the turbine and electronic components.
5. A *DC generator* converts mechanical rotation into electrical energy.

6. The *control circuitry* houses the Espressif ESP32 microcontrollers and sensors. *Dual ESP32* units operate within a network environment with a minimum bandwidth of 10 Mbps and a latency below 50 ms.

130

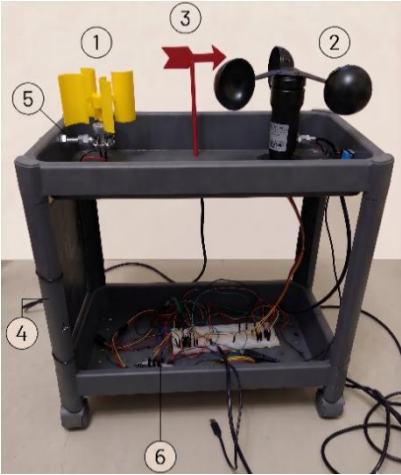


Figure 1: Custom made wind turbine prototype.

Two ESP32 microcontrollers are utilised to manage multiple real-time sensor inputs. All collected data is stored with timestamps. The circuit diagram for sensor integration is shown in Fig. 2. Three *DHT11* modules track temperature and humidity gradients across different sections of the system, while *gas and sound sensors monitor* atmospheric quality and acoustic emissions. A *piezoelectric sensor* is used to detect structural vibrations. A *line detect sensor* is used to measure the blade speed to ensure synchronisation with wind speed. An *INA219* power-monitoring module is integrated to capture precise current and voltage data from the turbine's output.

135

3.2 Materials: Data from the Prototype

140

3.2.1 Prototype Dataset and Descriptive Statistics

By carefully mapping the physical components, a real-world industrial wind turbine was modelled at a laboratory scale. The prototype utilises an array of sensors to capture critical environmental and mechanical variables and preserves data structures for ensuring operational fidelity. Our physical model bridges the gap between theoretical SCADA datasets and live IoT data.

Detailed descriptive statistics regarding the collected 2453 instance data from the prototype are presented in Table 1.

145

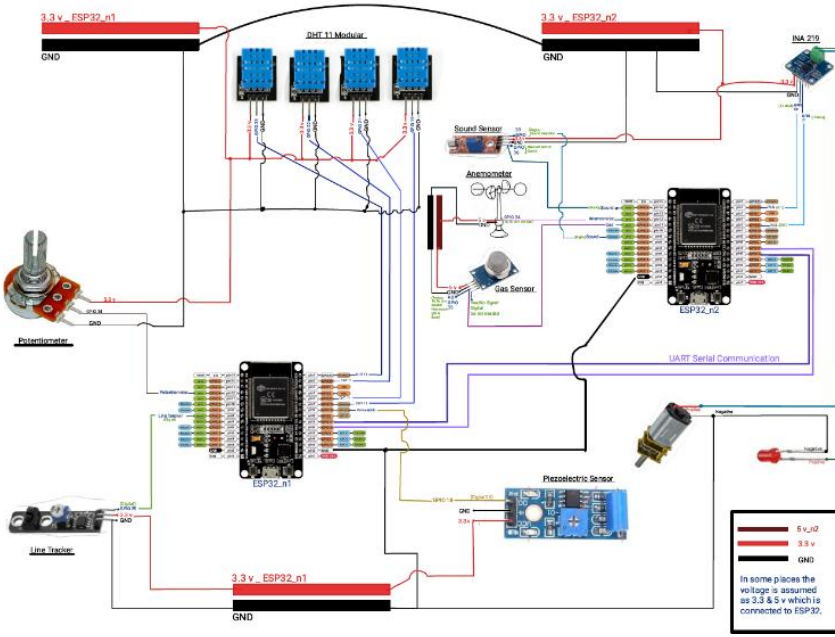


Figure 2: The circuit diagram of sensor integration on the prototype.

Table 1: Basic statistics of collected data from the prototype.

Attributes	Average	Standard Deviation	Coefficient of Variation	Min.	Max.	Range	IQR	MAPE
microphone	108.76	102.19	0.94	0	538	538	155	78.98
gas	368.79	57.15	0.15	113	545	432	65.75	12.38
wind_direction	167.95	140.4	0.84	21	360	339	309	78.33
vibration	525.49	612.26	1.17	0	2941	2941	974.75	99.85
wind_speed (km/h)	24.58	23.18	0.94	0	250.99	250.99	38.84	75.04
ambient_temperature	28.84	6.42	0.22	0	50.9	50.9	6.9	16.63
gearbox_temperature	32.1	9.81	0.31	0	60.7	60.7	12.6	23.84
nacelle_temperature	33.6	7.19	0.21	0	58.4	58.4	9.4	16.66
voltage	1.61	0.88	0.55	0	3.17	3.17	1.73	51.18
current (mA)	0.29	0.12	0.41	0	0.89	0.89	0.14	30.89
power (mW)	0.46	0.33	0.72	0	2.47	2.47	0.46	58.6

All wind speed values in Table 1 are recorded in kilometers per hour (km/h); the average operational wind speed of 24.58 km/h corresponds to approximately 6.83 m/s, which falls within standard wind turbine site averages.

The data collected from the prototype has been pre-processed. Principal Component Analysis (PCA) was used to visualise the data. To detect outliers, the $1.5 \times \text{IQR}$ (Interquartile Range) rule, rolling window analysis, K-Means clustering, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) were applied as in Fig. 3. A Min-Max scaler was used for normalization and feature scaling. Multi-class categorical features were mapped to numerical values. Non-predictive columns with zero standard deviation were removed. Features and target labels were prepared for machine learning. The dataset was initially divided into two subsets: 80% for model development and 20% as a hold-out test set for final evaluation. To reduce overfitting, the development subset was further evaluated using 5-fold cross-validation, where each

fold consisted of 80% training data and 20% validation data. *Time Window Sliding* was employed to identify and filter statistical outliers within localised temporal segments, removing extreme values while preserving the overall trend of the turbine's operational signals. The imbalanced dataset was handled through *Data-Level Methods*. To prevent data leakage, the dataset was *split chronologically*. A novel K-Means undersampling approach was then applied exclusively to the training set, balancing the classes by reducing “Normal” instances to representative centroids matching the “Failure” count.

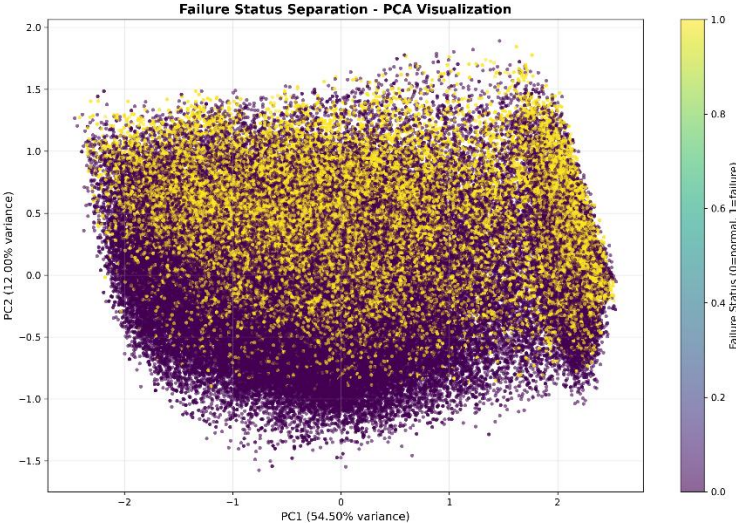


Figure 3: Final pre-processed dataset balanced via K-Means centroids.

The pre-processing pipeline ensured that the data were suitable for various machine learning algorithms, while mitigating issues such as skewness, imbalance, or feature misrepresentation.

To systematically manage the high temporal density and exceptional volume of the raw telemetry streams, a temporal aggregation preprocessing pipeline was established. Raw high-frequency signals were grouped and averaged over specified temporal windows to filter out transient sensor noise and capture robust steady-state operational trends. This step was crucial to ensure a lightweight data footprint compatible with the real-time constraints of low-power ESP32 microcontrollers and our 15-second IoT transmission cycle. However, this window-based averaging process inevitably collapses the fine-grained, continuous chronological step-by-step sequences, resulting in the loss of continuous sequential timestamp dependencies. Consequently, sequential multivariate time-series models (such as Long Short-Term Memory [LSTM] or Gated Recurrent Units [GRU]), which strictly require unbroken, chronologically ordered temporal matrices, were mathematically unsuitable for this preprocessed dataset. The predictive maintenance objective was thus formulated as a highly optimized, non-sequential classification problem over the aggregated tabular feature space.

To rigorously evaluate the transferability and diagnostic robustness of the proposed teacher-student paradigm under real-world conditions, a separate, independent testing dataset was compiled from the physical prototype. Unlike the primary dataset which utilizes high-confidence teacher-generated pseudo-labels for training, this independent testing set was manually labeled to establish an absolute physical ground-truth benchmark. The manual labeling process was executed by

directly cross-referencing our precise physical experiment logs, which meticulously recorded the exact timestamps of every physical failure injection (such as blade imbalances, mechanical gear friction, and electrical short-circuits) on the prototype turbine. This independent, ground-truth-verified dataset serves as an unseen testing environment to evaluate both the SCADA-only teacher model and the multi-sensor student model under identical physical operational anomalies.

3.2.2 Prototype Data Labelling Methodology

Given the practical and safety constraints of inducing catastrophic physical failures on the hardware prototype, obtaining direct experimental ground-truth labels was unfeasible. To address this, a model-driven pseudo-labeling approach has been adapted which serves as a pragmatic domain adaptation strategy. The optimal XGBoost classifier, pre-trained on labeled industrial SCADA data (Kasimov, 2024), was utilized as a “teacher” model to annotate the unlabeled prototype telemetry based on shared physical indicators (such as temperatures, wind speed, and voltage). Although this semi-supervised label propagation does not substitute for physical destructive testing, it establishes a mathematically structured baseline. This allows us to systematically analyze how the inclusion of previously unavailable localized sensors (gas and noise) modifies the feature space and improves the predictive capability of the “student” LightGBM model.

To label the raw prototype dataset in the absence of physical failure logs, a two-stage transfer learning (teacher-student) workflow has been implemented. This framework leverages the predictive rules learned from the large-scale, historical SCADA dataset to annotate the real-time prototype stream:

- **Stage 1 (Teacher-Driven Annotation):** The optimized SCADA-trained XGBoost model was utilized as a 'teacher' model. Crucially, because this model was not trained on the prototype's unique sensors (gas, acoustic/microphone, and piezoelectric vibration), the annotation was performed using **only the overlapping features** present in both datasets (ambient temperature, wind speed, wind direction, nacelle temperature, gearbox temperature, and voltage). The new sensor inputs were masked during this phase to prevent input mismatch or data leakage.
- **Stage 2 (Student-Model Retraining):** Once the binary status labels (Normal vs. Failure) were generated, a separate LightGBM model was trained as a 'student' model. Unlike the teacher, this model was trained on the **full prototype feature set**, including the newly introduced acoustic, gas, and vibration variables. This sequential design allowed the student model to mathematically map the correlations between these novel IoT sensor inputs and the predicted operational states, showing that the added physical dimensions significantly enhance failure prediction accuracy.

3.3 SCADA Dataset Pre-processing

The specific dataset utilised is the large-scale “**Wind Turbine SCADA Data for Early Fault Detection**”. A detailed description of the data, engineering units, and sensor types is available in Kasimov (2024). The dataset covers 89 cumulative years of operational data derived from 36 industrial-scale wind turbines across three geographically distinct wind farms, as summarised in Table 2. The SCADA dataset initially had a significant imbalance, with 87.2% “Normal” operations and only 12.8% as “Failure” events. A balancing procedure was implemented.

Table 2: SCADA dataset summary.

Metric	Wind Farm A	Wind Farm B	Wind Farm C
Total Features	86	257	957
Resolution	10 Minutes	10 Minutes	10 Minutes
Training Duration	1 Year	1 Year	1 Year
Status Labels	Comprehensive	Comprehensive	Comprehensive

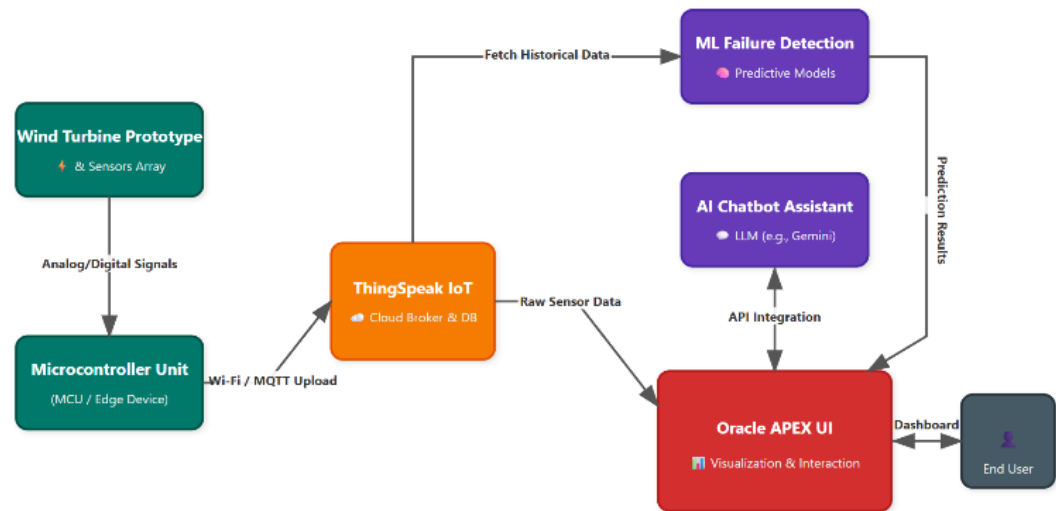
215 To simplify high-dimensional SCADA data, a feature-extraction process was implemented to capture the industrial three-phase alternating current (AC) power characteristics. The three separate voltage readings from the original dataset (*sensor_32_avg*, *sensor_33_avg*, and *sensor_34_avg*) were combined into a single “Calculated Voltage” feature by averaging them. The final SCADA feature set is shown in Table 3.

Table 3: Reduced dataset obtained from original SCADA data.

Feature Name	Description
sensor_0_avg	Ambient temperature
sensor_1_avg	Wind absolute direction
sensor_43_avg	Nacelle temperature
sensor_11_avg	Gearbox temperature
wind_speed_3_avg	Windspeed
voltage	Calculated voltage (Mean of 3 phases)
sensor_50	Total active power

220 **3.4 Methods: Our SWTFPS Architecture**

The proposed system is a multi-layered architecture (Fig. 4) that integrates traditional data with real-time data stored on a cloud service. The system transmits operational metrics to the ThingSpeak IoT platform (MathWorks, 2026). The integrated data are then analysed using machine learning algorithms and the Gemini API before being visualised in Oracle APEX.



225 **Figure 4: Overall system architecture.**

For real-time operation, the system utilises the secure communication pipeline shown in Fig. 5, which employs both the Hypertext Transfer Protocol (HTTP) and Message Queuing Telemetry Transport (MQTT) protocols. The ESP32 units act as publishers sending data to distinct topics (*WindTurbine1*, *WindTurbine2*). The *Python AI server* acts as a subscriber, continuously retrieving raw data, performing failure predictions, and writing diagnostic feedback for the user.

3.5 Evaluated Models for Decision Support

The system analyses sensor data to detect potential faults in mechanical gearboxes and electrical systems at an early stage of operation. To achieve the highest precision, eleven different machine learning models (Table 7) were evaluated.

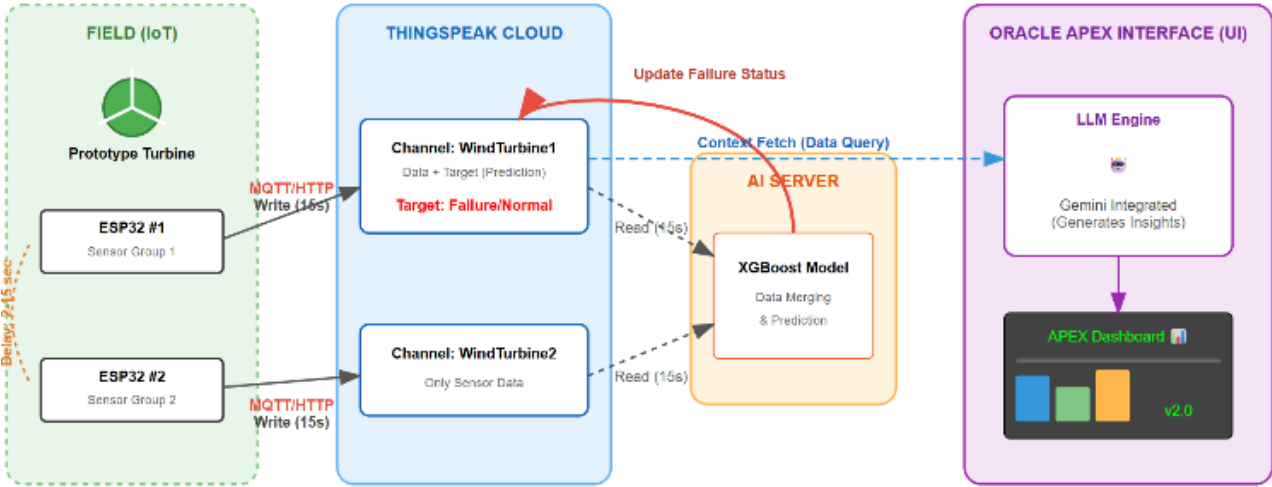


Figure 5: IoT and ML communication flow diagram.

Among the evaluated models, XGBoost emerged as the optimal choice due to its superior ability to handle non-linear sensor relationships and deliver the highest predictive performance. Ultimately, the system utilises this optimised model to provide actionable diagnostic outputs, including failure probability, risk levels, and the overall anomaly status of wind turbines. After the data pre-processing and model optimisation stages, the optimal XGBoost model was integrated into the physical system for real-time operational tests. Physical prototype tests have shown that the model successfully maintains a 15-second data collection cycle. Data transfer and analysis are completed within 5 seconds, panel updates within 3 seconds, and chat responses within 10 seconds. Synchronisation between ESP32 units was maintained, and the system returned to normal operation within 15 seconds without data loss.

245 Figure 6 illustrates the machine learning workflow, including the 80/20 train–test split, 5-fold cross-validation, and pseudo-labeling, where an XGBoost Teacher Model using 7 inputs generates labels for a LightGBM Student Model incorporating 3 additional sensor features, with final evaluation on the unseen hold-out test set.

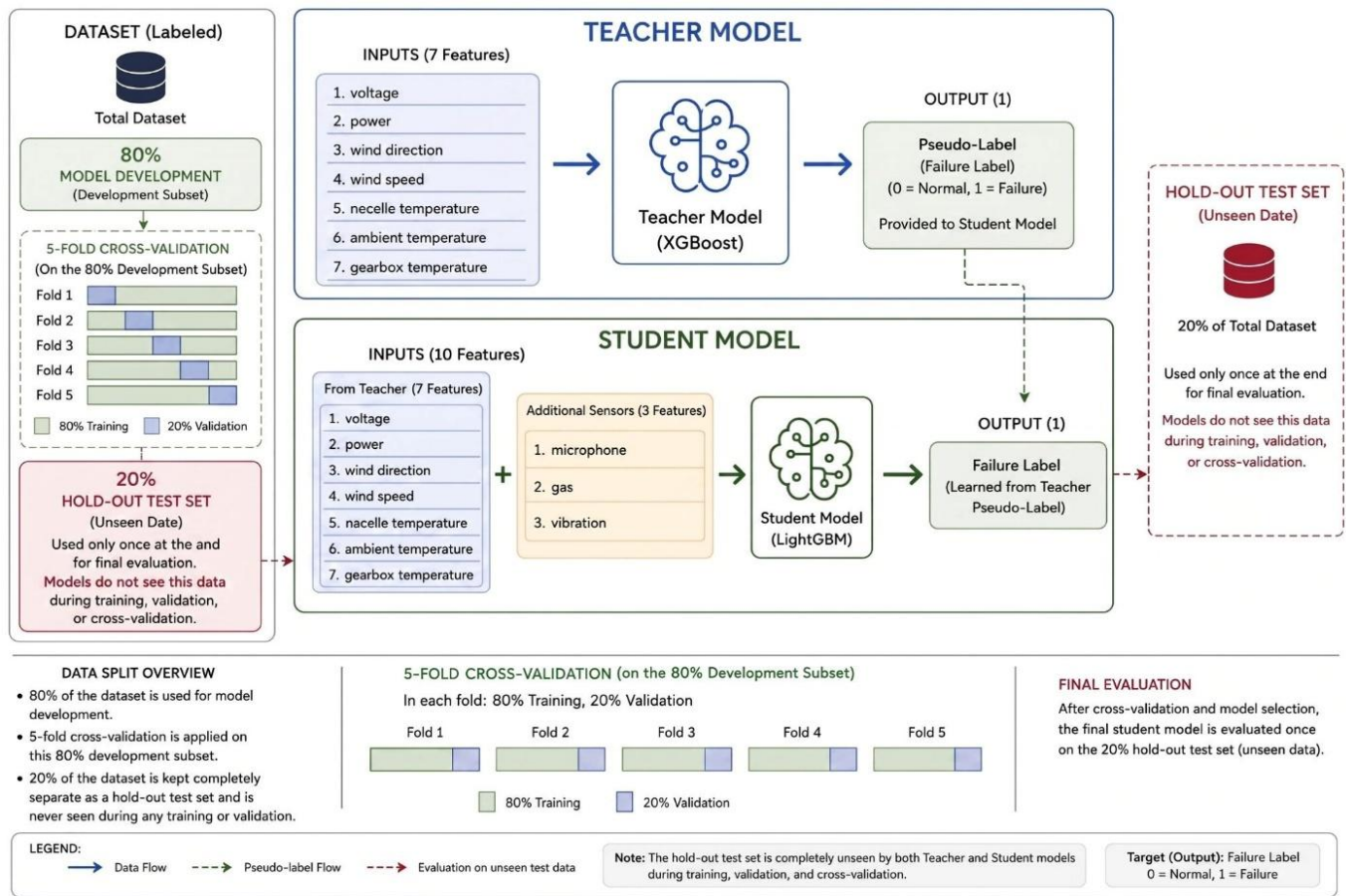
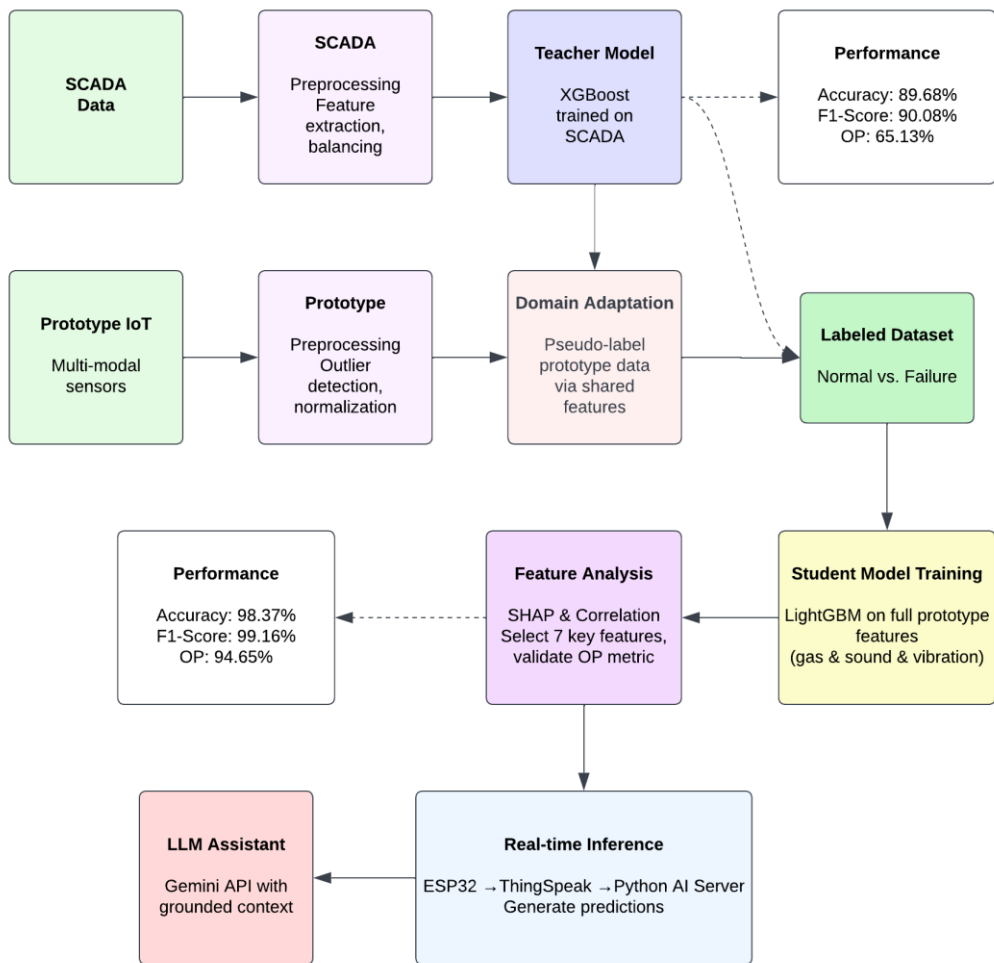


Figure 6: AI-supported wind energy fault prediction teacher-student transfer learning model.

250 Figure 7 presents a unified, end-to-end methodology that eliminates the fragmented presentation of the workflow by connecting SCADA preprocessing, domain adaptation, student model retraining, explainability, real-time inference, and LLM-based decision support within a single integrated framework.



255 **Figure 7: Integrated SWTFPS research pipeline.**

3.6 Performance Metrics

260 The overall correctness and reliability of the models were evaluated using standard *supervised learning metrics*: *Accuracy*, *Recall*, *Precision*, *F1-Score*, and *ROC-AUC*. The mathematical formulations and detailed theoretical backgrounds of these standard machine learning evaluation metrics are further examined in (Naidu, 2023). The metrics collectively provide a comprehensive assessment of the model's ability to distinguish between faulty and healthy turbine states, especially in cases of unbalanced class distributions. In wind turbine failure prediction, the *F1-Score* is preferred because it better reflects the minority fault class in highly imbalanced datasets. The *ROC-AUC* is relegated to the background as they can be misleading in imbalanced data. To provide a more comprehensive and balanced assessment of classification performance under imbalanced operating conditions, an Overall Performance (OP) metric was additionally formulated by combining both

265 success-based and error-based evaluation criteria. While conventional metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC individually describe specific aspects of model behaviour, they may not fully reflect the trade-off between predictive capability and classification reliability in fault diagnosis applications. In particular, high Accuracy or ROC-AUC values may still coincide with elevated false alarm rates or missed fault detections in highly imbalanced datasets. Therefore, the proposed OP formulation integrates both positive performance indicators and error penalties into a single evaluation
270 framework. The OP metric is defined as in Eq. (1):

$$\text{Overall Performance} = \frac{(0.25 \times F1\text{-Score}) + (0.2 \times ROC\text{-AUC}) + (0.15 \times \text{Accuracy}) + (0.15 \times \text{Precision}) + (0.15 \times \text{Recall}) + (0.1 \times (1 - FPR))}{1 + \text{LogLoss} \times \text{Brier Error} \times \text{FNR}}, \quad (1)$$

where Accuracy, Precision, Recall, F1-Score, and ROC-AUC represent success-oriented metrics, while LogLoss, Brier Error, False Positive Rate (FPR), and False Negative Rate (FNR) represent prediction uncertainty and classification error
275 penalties. The constant value of 1 in the denominator is included to prevent numerical instability and disproportionate inflation of the metric when error terms approach zero. By jointly considering predictive strength and error behaviour, the OP metric provides a more interpretable and balanced comparison of machine learning models for real-time wind turbine fault prediction systems.

The mathematical weights assigned to the parameters in the proposed OP metric (Eq. 1) represent a key methodological
280 novelty of this work, established through a systematic literature-consensus and expert-consultation approach to align with the operational demands of industrial wind turbine diagnostics.

The F1-Score is assigned the highest weight (0.25) due to its critical role in balancing precision and recall under extreme class imbalance. The ROC-AUC is assigned 0.20 to measure threshold-independent class discrimination. Standard metrics (Accuracy, Precision, Recall) are weighted at 0.15 each to preserve baseline performance compatibility. Finally, the error
285 penalty is weighted at 0.10 to penalize high prediction uncertainty (LogLoss) and missed detections (FNR).

To validate the mathematical stability of this novel formulation, a sensitivity analysis was performed by varying the coefficients across alternative configurations (including equal-metric weighting and error-heavy schemes). Under all perturbations, the relative performance rankings of the eleven models remained completely stable, with XGBoost consistently emerging as the top performer on SCADA data (65.13% OP) and LightGBM on prototype data (94.65% OP).
290 This demonstrates that the OP formulation is a robust, objective, and highly stable novelty for evaluating real-time diagnostic systems.

3.7 Feature Importance Analysis

To optimize model performance, feature selection was executed via a two-stage hybrid pipeline combining collinearity filtering and cooperative game theory. In Stage 1, Pearson correlation analysis was performed on the raw SCADA dataset to
295 identify and filter out highly collinear and redundant variables ($|r| > 0.85$), ensuring low multicollinearity and stable model training. In Stage 2, because wind turbine thermodynamics and sensor signals involve complex, non-linear physical

300

interactions, linear selection standards like Lasso or Ridge regression are insufficient. Therefore, SHAP (SHapley Additive exPlanations) values have been computed on the remaining variables using the baseline XGBoost model. This allowed us to measure the true non-linear marginal contribution of each feature, selecting the seven key features shown in Fig. 8 for the final training set.

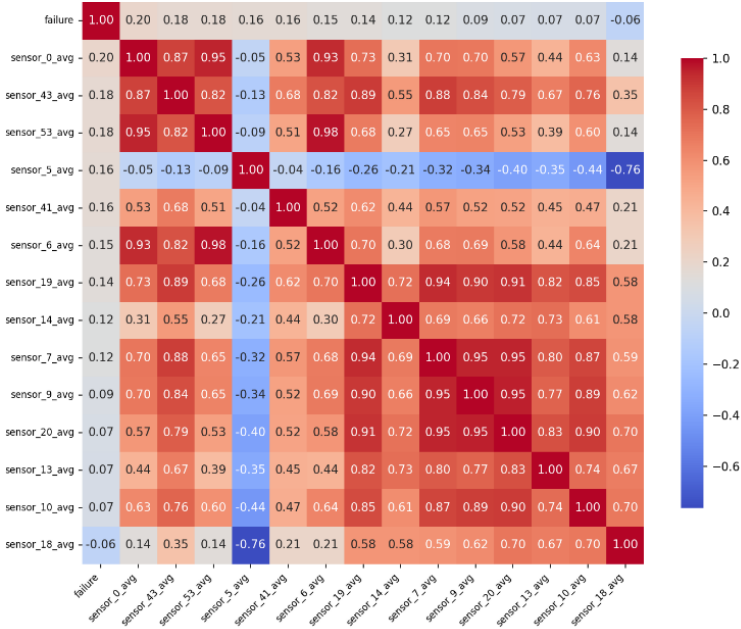


Figure 8: Heatmap of the most related features.

305

The SHapley Additive exPlanations (SHAP) plot in Fig. 9 shows the model's most impactful features, from highest to lowest. These seven features form the basis of the appropriate final training phase. The features that impact real-time failure conditions most are as follows:

310

- *sensor_0_avg* (Ambient temperature)
- *sensor_43_avg* (Nacelle temperature)
- *sensor_50* (Total active power)
- *sensor_11_avg* (Gearbox temperature)
- *wind_speed_3_avg* (Windspeed)
- *sensor_1_avg* (Wind absolute direction)
- *voltage* (Mean of voltage phases)

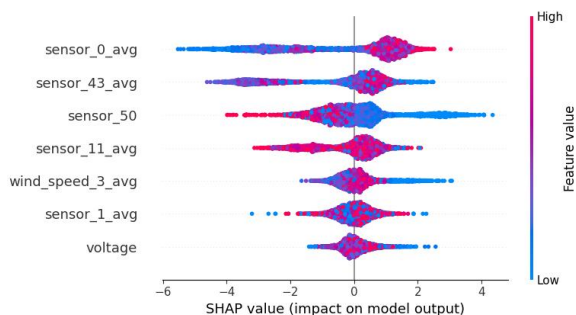


Figure 9: SHAP of XGBoost prototype model.

315 4 SWTFPS: LLM-Based Interactive User Interface

In the final phase of the SWTFPS project, a comprehensive Oracle APEX-based web interface was developed to make diagnostic data from the XGBoost model actionable for end users (Fig. 10). This platform acts as a bridge between complex AI logic and the operator.

320 The interface's technical architecture is built on modern web technologies (*HyperText Markup Language: HTML, Cascading Style Sheets: CSS, JavaScript: JS*) on the frontend and robust data management (*Structured Query Language: SQL, Procedural Language/Structured Query Language: PL/SQL*) on the backend. The system consists of the following core modules:

- Live IoT monitoring dashboard
- Instant AI failure predictions
- 325 • Interactive LLM maintenance assistant
- Transparent SHAP/Local Interpretable Model Agnostic Explanation (LIME) visual reports
- Natural Language system querying
- Fast understandable AI outputs

This integration transforms the prototype into a professional predictive maintenance platform. Supported by an LLM, it
330 delivers real-time turbine health reports and optimisation suggestions on demand.

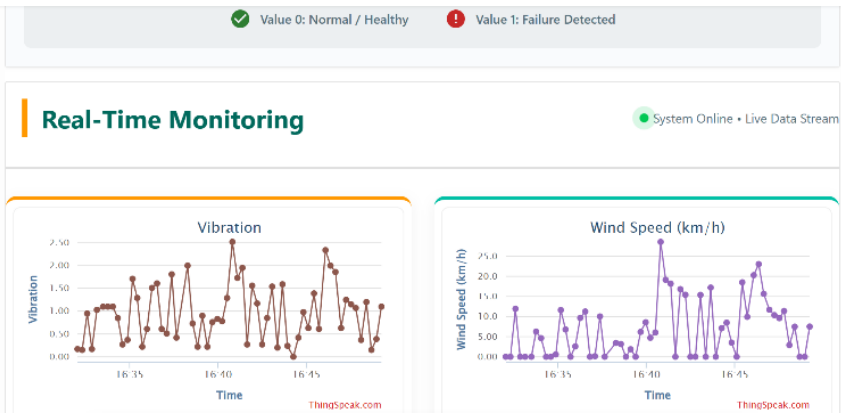


Figure 10: Oracle APEX real-time monitoring interface.

Figure 11 illustrates the typical page of a given response by the LLM interface to the user's question. The AI assistant can be updated quickly and easily by using its Retrieval Augmented Generation (RAG) capability. Examples of critical prompts and their AI assistant responses are given in Table 4.

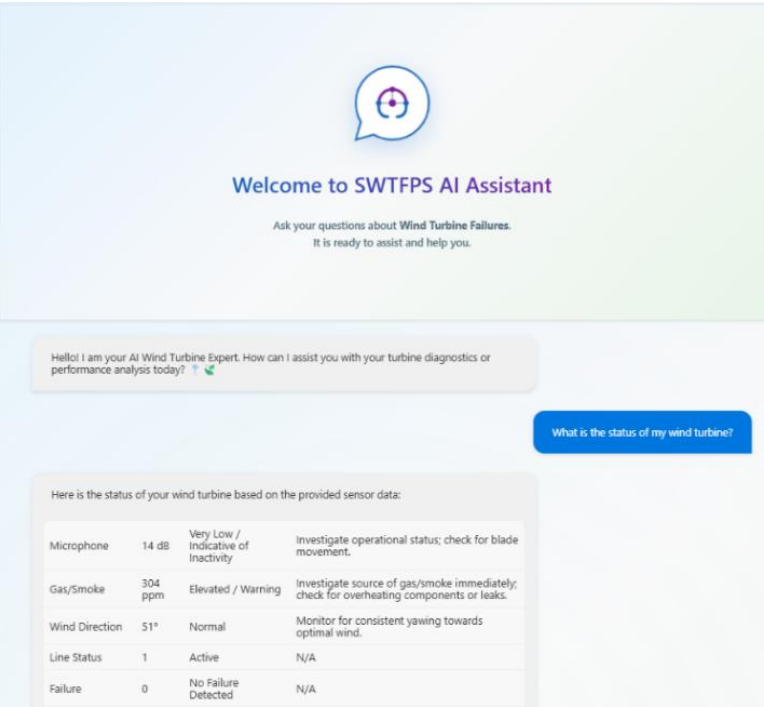


Figure 11: The SWTFPS AI assistant user interface.

Table 4 presents the most critical prompts and AI assistant responses Number 1-3, in a tabular format. When a status query is initiated, the backend fetches real-time sensor data, which the LLM then uses to interpret the system state.

340 **Table 4: Several complicated questions.**

Number	Prompt	Response
1	What is the status of my turbine?	Based on the provided sensor data, your wind turbine is in a critical state and ... • Microphone (dB): 110 -... • ... (other sensors) ... Summary and Suggestions: ...
2	What is the temperature of my turbine?	The ambient temperature around your wind turbine is 24.3°C , the gearbox temperature is 22.2°C , and the nacelle temperature is 23.4°C
3	Do you have suggestions to improve my wind turbine? Explain in detail.	Comments and Suggestions based on Sensor Data: The sensor data indicates a critical situation. ... Immediate Action Required: ...

345 A critical challenge in integrating LLMs into industrial monitoring is the risk of "hallucinations," where the model might generate plausible but physically incorrect diagnostic recommendations from raw numerical arrays. To eliminate this risk and ensure strict verifiability, the SWTFPS does not rely on the LLM to independently deduce failure mechanics from raw data. Instead, the Gemini API is explicitly grounded in deterministic domain knowledge through a RAG (Retrieval-Augmented Generation) approach.

Before the raw sensor data is transmitted to the LLM, the Python backend evaluates the data streams against a hardcoded "Sensor-Pattern-to-Action" reference matrix (Table 5). This matrix acts as a safety override mechanism, mapping specific multi-sensor operational thresholds directly to known mechanical and electrical fault typologies. When a threshold condition is met, the system injects the pre-validated domain context (e.g., "Short Circuit & Smoke Risk Detected") into the LLM's system prompt. Consequently, the LLM functions strictly as a natural language synthesizer that contextualizes pre-validated engineering facts, rather than an unguided diagnostic engine. This guarantees that all AI recommendations (such as emergency shutdowns or inspection directives) are strictly verifiable against physical engineering rules.

350

Table 5: Deterministic sensor-pattern-to-action reference matrix for LLM grounding.

Diagnostic State	Trigger Conditions	Grounded Action Directive
Disconnected Temperature Sensor	T1 = 0 OR T2 = 0 OR T3 = 0	Signal fault. Advise immediate sensor wiring inspection.
Unrealistic Wind Speed Anomaly	Wind Speed > 150 km/h	Anemometer failure. Disregard wind metric; check sensor integrity.
Open Circuit / Broken Connection	Wind > 15 km/h AND Volt > 1.0 AND Current ≤ 0.05	Generation without transmission. Inspect power lines for breaks.
Short Circuit & Smoke/Fire Risk	Current > 0.70 AND Volt < 0.5 AND Gas > 400 ppm	Critical hazard. Recommend emergency shutdown and fire inspection.

Electrical Noise / Ghost Power	Wind < 2.0 km/h AND Volt > 1.65	Inverter/Generator calibration error. No mechanical wind input.
Bearing Friction / Mechanical Wear	(T3 - T1) > 15°C AND Vibration > 1500	Thermal-mechanical anomaly. Inspect gearbox lubrication and bearings.
Blade / Rotor Imbalance Damage	Wind > 20 km/h AND Vibration > 2000 AND Power < 0.2	High kinetic input with low power and severe vibration. Halt operation.
Generator Jamming	Wind > 15 km/h AND Volt < 0.1	Mechanical rotor-generator coupling failure. Check for physical jams.
Generator Strain / Overload	Power > 1.0 AND T2 > 45°C	High load stress. Advise reducing load and checking cooling systems.
Critical Overheating	T2 > 50°C OR T3 > 50°C	Severe thermal breach. Initiate emergency thermal shutdown.

355 To provide domain-specific responses, the SWTFPS integrates a Retrieval-Augmented Generation (RAG) mechanism with an LLM-based interface. The process consists of the following stages:

- **Knowledge Base Preparation:** User-provided technical PDF documents are processed into text chunks and stored as the RAG knowledge base.
- **Semantic Retrieval:** Document chunks and user queries are converted into vector representations using the Gemini embedding model, and cosine similarity is used to retrieve the most relevant content.
- **Real-Time Data Integration:** Current turbine measurements are obtained from the ThingSpeak IoT infrastructure and combined with the retrieved information.
- **Response Generation:** The retrieved technical context, real-time sensor data, and user query are provided to the Gemini-based LLM. When a failure or anomaly is detected, the LLM relates these inputs to provide a diagnostic recommendation based on the retrieved troubleshooting information.

5 Results and Comparison of Model Performances

5.1 Model Comparison and Final Choice

To maximise the predictive capability of the models and prevent overfitting, systematic hyperparameter tuning was conducted using 5-Fold Cross-Validation, following standard optimisation methodologies established in literature. Rather than relying on default parameters, optimal configurations for each algorithm (such as tree structures for ensemble methods, learning rates for boosting models, and specific distance metrics) were identified based on the highest mean *F1-Scores*.

The final performance metrics of the algorithms post-tuning are summarised in Table 6. The results indicate that CatBoost and LightGBM, achieved the highest accuracy levels. While XGBoost and Random Forest also demonstrated strong predictive capabilities, simpler models like Logistic Regression showed limited performance. This outcome highlights the need for tuned boosting models to capture the complex, non-linear relationships inherent in industrial sensor data.

architectural structures of the specific models and the technical definitions of parameters are consistent with the literature, and Islam et al. (2025) can be considered as a reference for comprehensive explanations of these parameters. The hyperparameter optimization process was carried out independently according to the characteristics of the existing dataset.

Table 6: Accuracy and optimal hyperparameters for all evaluated models.

Model	Accuracy	Key Optimal Parameters
CatBoost	89.66%	learning_rate: 0.1, iterations: 500
LightGBM	89.58%	num_leaves: 100, n_estimators: 200, learning_rate: 0.05
XGBoost	88.98%	n_estimators: 200, max_depth: 10, learning_rate: 0.05
RandomForest	88.02%	n_estimators: 100, max_depth: 20, min_samples_split: 2
DecisionTree	82.79%	max_depth: 20, min_samples_split: 10, min_samples_leaf: 4
MLP	80.20%	hidden_layer_sizes: (128, 64), learning_rate_init: 0.01, alpha: 0.001
KNN	79.87%	n_neighbors: 7, weights: 'distance'
Logistic	69.90%	C: 0.01, penalty: 'l2'

380 The performance of the k-Nearest Neighbors (KNN) and XGBoost models was compared using the raw dataset. The results clearly show the limitations of using unbalanced data. Although the KNN model has a high overall *Accuracy* (98.43%), its *Recall* (91.87%) and *F1-Score* (93.33%) for the “Failure” class indicate a strong bias toward the majority “Normal” class. On the other hand, the XGBoost model struggles significantly with raw high-dimensional data. Its overall accuracy drops to 66.09%, and its *Precision* for the Failure class falls sharply to 20.65%, indicating it generates many false alarms. While the

385 initial baseline evaluation of the raw data showed that XGBoost performed worse than *KNN*, the model trained on the refined dataset emerged as the top performer.

According to the Confusion Matrix in Fig. 12, the XGBoost model can distinguish failure cases (Class 1) with high accuracy (1876 True Positives, **89.68%**), and an F1-Score of **90.08%**. An Area Under the Curve (AUC) of **95.65%** is observed in the Receiver Operating Characteristic (ROC) curve in Fig. 12. The high predictive capability demonstrates the model's

390 effectiveness in detecting faults and providing actionable insights.

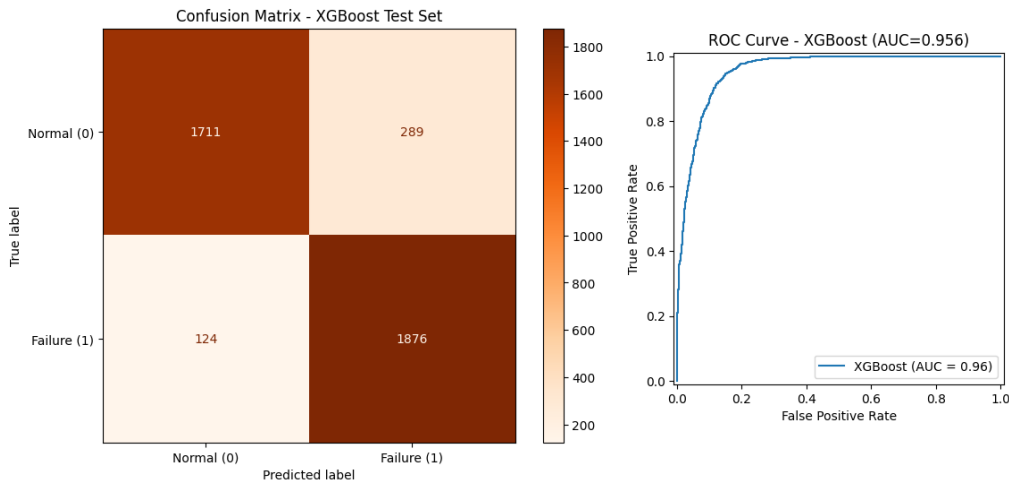


Figure 12: Confusion matrix and ROC curves of the XGBoost.

5.2 Explainability and Interpretation Capability

Following the initial assessment of the large-scale dataset, the SWTFPS was tested on a small-scale, refined subset of approximately 20,000 samples. The F1-Score was prioritised as the primary evaluation metric over ROC-AUC, as it better mitigates the imbalance in the wind turbine data. Again, the XGBoost model was selected for its superior performance. It achieved the highest *F1-Score* among the evaluated models, demonstrating a robust balance between Precision and Recall, in a small-scale experimental trial as well.

As detailed in Table 7, the *XGBoost* model provides a superior balance of high accuracy, fast inference speed, and architectural simplicity, making it the most reliable and cost-effective solution.

Table 7: Model performance comparison.

Model	Accuracy	F1-Score	ROC-AUC
XGBoost	89.68%	90.08%	95.65%
Stacking	89.58%	89.85%	96.39%
LightGBM	88.78%	89.32%	95.70%
CatBoost	88.23%	88.79%	95.61%
RandomForest	87.78%	88.22%	95.59%
MLP	80.18%	80.80%	88.37%
SVM	77.68%	77.89%	85.31%
KNN	77.83%	77.81%	85.39%
NaiveBayes	71.15%	71.11%	77.65%
DecisionTree	84.05%	84.28%	84.05%
Logistic	69.58%	69.64%	75.91%

Transforming the data into a noise-free, balanced dataset significantly improved the performance of all eleven evaluated models. However, the results show that all tree-based ensemble methods (Stacking Ensemble, Random Forest, LightGBM, CatBoost, and XGBoost) outperformed the simpler models (KNN, Logistic Regression, etc.).

5.3 Independent Empirical Validation and Sensor Ablation Analysis

To systematically address potential pseudo-label bias inherent in the teacher-student procedure and to directly quantify the predictive value added by each newly introduced localized IoT sensor (Sound, Gas, and Vibration), an independent validation study was conducted. Both the transferred SCADA-equivalent Teacher model (XGBoost) and the multi-sensor Student model (LightGBM) were evaluated against the same manually labeled, ground-truth-verified independent prototype dataset. The system has been systematically evaluated across eight distinct sensor configurations, starting from the SCADA-equivalent baseline and incrementally incorporating the localized sensors. The resulting performance metrics are presented in Table 8.

Table 8: Ablation study evaluating the individual and joint contributions of localized IoT sensors against manually labeled, ground-truth-verified independent prototype data.

Scenario	Sensor Configuration	Accuracy	Precision	Recall	F1-Score	OP Score
Scenario 1	Base Model	97.24%	97.07%	92.56%	94.76%	94.85%
Scenario 2	Base + Vibration	99.37%	99.53%	98.14%	98.83%	98.66%
Scenario 3	Base + Sound	98.49%	98.56%	95.81%	97.17%	97.20%
Scenario 4	Base + Gas	97.37%	97.55%	92.56%	94.99%	94.89%
Scenario 5	Base + Vibration + Sound	99.50%	100.00%	98.14%	99.06%	98.75%
Scenario 6	Base + Vibration + Gas	98.87%	98.13%	97.67%	97.90%	97.77%
Scenario 7	Base + Sound + Gas	99.12%	98.60%	98.14%	98.37%	98.26%
Scenario 8	Full-Model	99.62%	99.53%	99.07%	99.30%	99.09%

The empirical necessity of multi-modal sensor fusion and the validity of our teacher-student paradigm are illuminated by the independent validation in Section 5.3, which addresses the 'mimicry bias' and domain-shifting challenges inherent to the framework. While global SHAP results (Fig. 14) show that gas and acoustic metrics are prioritized during model splits, Table 8 validates this synergy against physical ground-truth labels. If the student merely mimicked the SCADA-only teacher under this domain discrepancy, its performance would be capped by the teacher's error rate; however, the progression from a SCADA-only baseline Recall of 92.56% and F1-Score of 94.76% (Scenario 1) to a Recall of 99.07% (+6.51%) and F1-Score of 99.30% (+4.54%) in the Full Student Model (Scenario 8) proves the student successfully resolves the domain shift, capturing anomalies invisible to SCADA. Ablation analysis reveals that while the vibration sensor (Scenario 2) yields the most dominant standalone improvement (F1-Score: 98.83%, +4.07%; OP: 98.66%) by capturing mechanical rotor dynamics, the acoustic and gas sensors provide vital diagnostic redundancies. The acoustic sensor (Scenario 3) detects high-frequency micro-cracking before vibrations propagate (F1-Score: 97.17%, +2.41%), while the gas sensor (Scenario 4) captures kinetic-free electrical fire risks despite sparse standalone gains (+0.23% F1-Score). Ultimately, multi-modal integration in the Full Model (Scenario 8) achieves peak performance, reducing the false negative rate on true physical failures to just 0.93% and mathematically expanding predictive maintenance limits.

6 Discussion and Limitations

6.1 Interpretation of Results

The XGBoost model optimised with 50% data balancing and the selection of seven key features demonstrated the highest performance (**89.68% Accuracy, 90.08% F1-Score, 95.65% ROC-AUC**). This finding demonstrates that data quality is more critical than data quantity in fault prediction and that this compact approach is highly effective for real-time monitoring on hardware such as ESP32.

The global SHAP analysis revealed that ambient and nacelle temperatures are the primary drivers of predictions, which align well with turbine thermodynamics. To further complement the evaluation, local dynamics were examined using LIME (Fig. 13). The evaluated scenarios demonstrated the model's reliability in accurately detecting failures driven by thermal

anomalies (*Case 1*). The model also recognises stable conditions (*Case 2*) and, critically, prioritises high-temperature warnings even when conflicting standard power metrics suggest normal operation (*Case 3*).

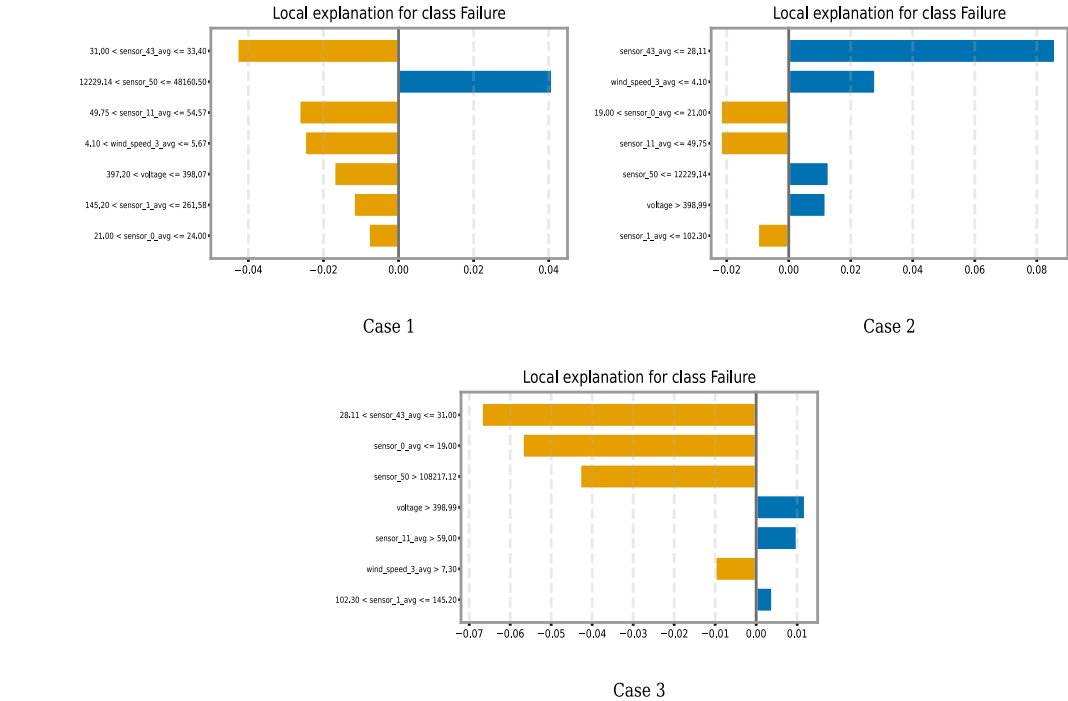


Figure 13: LIME cases of XGBoost.

To provide a more balanced evaluation of predictive capability and classification reliability under imbalanced conditions, the proposed Overall Performance (OP) metric introduced in Sect. 3.6 was additionally employed, and the result in Table 9 was obtained. As a result of the evaluation based on the overall performance metric, it is seen that the XGBoost model, whose performance was previously proven by the **F1-Score** ranking, stands out among the other models with its high **overall performance** of **65.13%**. The obtained result was compared with the results of similar studies in literature. Table 9: Model overall performance comparison.

Model	Accuracy	Precision	Recall	F1-Score	Log Loss	Brier Error	ROC-AUC	FPR	FNR	OP
XGBoost	89.68%	86.65%	93.80%	90.08%	25.36%	7.73%	95.65%	14.45%	6.20%	65.13%
Stacking	89.58%	87.57%	92.25%	89.85%	24.63%	7.48%	96.39%	13.10%	7.75%	64.94%
LightGBM	88.78%	85.17%	93.90%	89.32%	25.72%	7.94%	95.70%	16.35%	6.10%	64.40%
CatBoost	88.23%	84.70%	93.30%	88.79%	26.87%	8.19%	95.61%	16.85%	6.70%	63.18%
RandomForest	87.78%	85.12%	91.55%	88.22%	28.88%	8.78%	95.59%	16.00%	8.45%	61.07%
MLP	80.18%	78.32%	83.45%	80.80%	43.27%	13.87%	88.37%	23.10%	16.55%	47.12%
SVM	77.68%	77.15%	78.65%	77.89%	47.85%	15.52%	85.31%	23.30%	21.35%	42.88%
NaiveBayes	71.15%	71.21%	71.00%	71.11%	62.86%	20.15%	77.65%	28.70%	29.00%	34.16%
Logistic	69.58%	69.49%	69.80%	69.64%	58.95%	20.03%	75.91%	30.65%	30.20%	33.87%
KNN	77.83%	77.87%	77.75%	77.81%	117.78%	15.55%	85.39%	22.10%	22.25%	31.04%
DecisionTree	84.05%	83.09%	85.50%	84.28%	574.90%	15.95%	84.05%	17.40%	14.50%	11.91%

The results obtained in the articles in the literature are included in Table 10. However, the success and error metrics provided in the literature are insufficient to fill the table, and *Overall Performance (OP)* values could not be calculated. Similarly, Puruncajas et al.'s Artificial Neural Network (ANN) model and CPEM Study's Combined Probability Estimation model did not provide any scores and were therefore not directly included in the table.

Table 10: Model overall performance comparison among studies.

Article	Best Model	Accuracy	Precision	Recall	F1-Score	Log Loss	Brier Error	ROC - AUC	FPR	FNR	OP
Munguba et al.	Random Forest Classifier (RF)	99.94%	99.94%	99.94%	99.94%	-	-	100.0%	-	-	-
Zhang et al. (2023)	Adaptive Multivariate Time-Series CNN (AdaMTCN)	93.40%	-	-	99.60%	-	-	-	-	-	-
Zhang et al. (2025)	Hybrid CNN-LLM	91.50%	91.00%	90.00%	90.50%	-	-	96.00%	-	-	-
Ogaili et al.	Boost + ReliefF	99.40%	99.80%	99.80%	99.80%	0.005	-	99.90%	-	-	-
Pham & Han	Bi-LSTM + Attention Mechanism	99.40%	99.25%	99.52%	99.38%	-	-	99.80%	0.65%	-	-
Irfan et al.	Perturbed-Random Forest (P-RF)	99.88%	99.88%	99.88%	99.88%	-	-	100.0%	-	-	-
Tan & Carroll	DeepSeek-V3 Random Forest	-	97.00%	97.00%	97.00%	-	-	-	-	-	-
Inturi et al.	ANFIS (Adaptive Neuro-Fuzzy)	92.00%	-	-	-	-	-	-	-	-	-
Horodyvskyi et al.	LSTM Autoencoder / MLP	94.00%	-	94.00%	-	-	-	-	11.00 %	-	-
Hsu et al.	Random Forest	95.34%	-	-	-	-	-	-	-	-	-
Amin et al.	Cyclostationary-based CNN	89.00%	-	-	-	-	-	-	-	-	-

This emphasizes that a direct quantitative comparison between the SWTFPS and the historical studies compiled in Table 10 is subject to physical and data-structural limitations. These studies employ heterogeneous datasets with different temporal resolutions (ranging from real-time streams to 10-minute SCADA intervals), physical architectures of vastly different scales, and differing operational failure definitions.

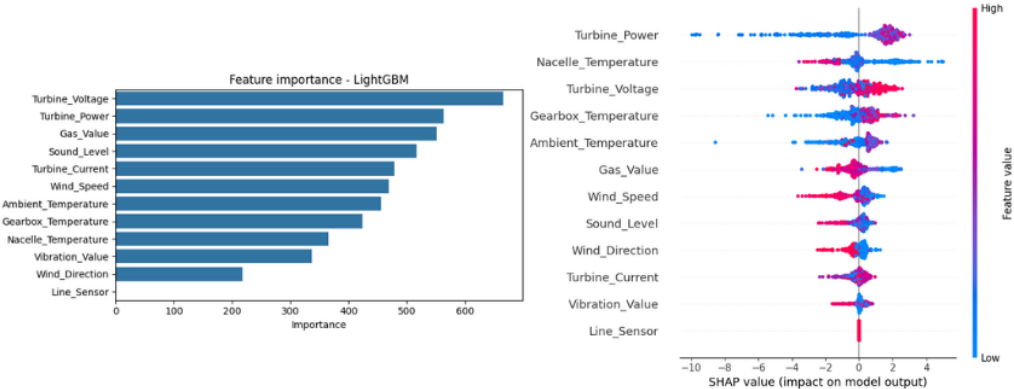
However, Table 10 serves as a vital diagnostic gap analysis. The empty “OP” column for all prior studies highlights a major limitation in current wind turbine literature: researchers frequently report isolated, success-oriented metrics (such as high accuracy) but fail to disclose critical indicators of model reliability, such as prediction uncertainty (LogLoss), calibration error (Brier Score), or false rates. This systemic omission makes calculating their Overall Performance (OP) impossible.

Consequently, Table 10 is presented not to claim absolute mathematical superiority over structurally different datasets, but to demonstrate the critical necessity of our proposed OP metric as a comprehensive, transparent, and unified benchmarking standard for future predictive maintenance research.

465 The data collected by the prototype was labeled by XGBoost which is the best model obtained from SCADA data. After the newly labeled dataset was retrained, LightGBM was selected as the best performing model with 98.37% *Accuracy*, 99.16% *F1-Score*, 98.95% *ROC-AUC*, and **94.65% OP** scores. Table 11 shows the comparison between the models trained with the data collected from the prototype. Figure 14 shows the (a) Feature Importance and (b) SHAP plots obtained from the LightGBM model results. According to the feature importance result, the newly added sensors, **gas** and **sound** level sensors, 470 were observed as *significant predictors* of failure output, while **vibration** was observed as a *moderate predictor*. The fact that the newly added sensors enabled the model to achieve a **98.86%** F1-Score (and **94.65%** OP) indicates that they have a high prediction performance above failure prediction.

Table 11: Data model overall performance comparison of labelled prototype.

Model	Accuracy	Precision	Recall	F1-Score	Log Loss	Brier Error	ROC-AUC	FPR	FNR	OP
LightGBM	98.37%	98.75%	99.58%	99.16%	10.27%	1.58%	98.95%	37.50%	0.42%	94.65%
CatBoost	98.17%	98.54%	99.58%	99.06%	5.30%	1.50%	98.66%	43.75%	0.42%	94.18%
XGBoost	97.96%	98.34%	99.58%	98.95%	5.81%	1.70%	98.87%	50.00%	0.42%	93.52%
Stacking	97.96%	98.34%	99.58%	98.95%	6.17%	1.53%	98.86%	50.00%	0.42%	93.51%
RandomForest	97.76%	98.33%	99.37%	98.85%	5.60%	1.55%	97.70%	50.00%	0.63%	93.21%
DecisionTree	98.37%	99.37%	98.95%	99.16%	58.73%	1.63%	90.10%	18.75%	1.05%	92.34%
MLP	97.76%	97.74%	100.00%	98.86%	8.56%	1.84%	91.76%	68.75%	0.00%	90.22%
KNN	96.95%	97.33%	99.58%	98.44%	21.52%	2.60%	89.68%	81.25%	0.42%	88.12%
SVM	96.74%	96.74%	100.00%	98.34%	10.55%	2.69%	93.86%	100.00%	0.00%	87.37%
Logistic	96.74%	96.74%	100.00%	98.34%	11.27%	2.84%	86.99%	100.00%	0.00%	86.00%
NaiveBayes	26.48%	100.00%	24.00%	38.71%	400.04%	68.63%	83.57%	0.00%	76.00%	50.51%



a) Feature Importance plot of LightGBM Model

b) SHAP Plot of LightGBM Model

475 **Figure 14: Feature importance and SHAP plots of LightGBM.**

Table 12 presents a systematic comparison of the proposed SWTFPS against recent literature across four key domains: SCADA-based deep learning, vibration signal processing, multimodal fusion, and LLM diagnostics, highlighting the specific research gaps our end-to-end interactive architecture addresses.

Table 12: Chronological literature review and research gap analysis.

Reference (Author, Year)	Core Methodology / System Architecture	Key Limitations & Research Gap Addressed	Diagnosis through LLM Interface
SCADA & Deep Learning Applications Alzawaideh et al. (2021); Pham & Han (2025); Munguba et al. (2024)	Predicts failures using historical SCADA data and supervised deep learning architectures (e.g., LSTM, Random Forest).	Relies on offline, single-source historical data without live edge-telemetry or interactive dashboards.	No
Vibration-Based & Signal Processing Meyer et al. (2022); Amin et al. (2022); Ogaili et al. (2024)	Extracts high-frequency vibration features using CNNs and XGBoost for automated fault classification.	Confined to a single sensor type (vibration), lacking comprehensive multi-sensor fusion and environmental parameters.	No
Multimodal & Hybrid Modelling Khan et al. (2023); Inturi et al. (2021)	Fuses multimodal sensor data (vibration, acoustic, oil) using physical thermal models and machine learning.	Requires complex, turbine-specific physical modeling and lacks real-time conversational user interfaces.	No
LLM Integrations in Diagnostics Tan & Carroll (2025); Zhang et al. (2025)	Employs LLMs (DeepSeek-V3, GPT-4) for generating synthetic datasets and optimizing offline diagnostic ML workflows.	Restricted to backend pipeline optimization without real-time, natural language decision-support for end-users.	No
Proposed Work (SWTFPS) Saygi, Ak, Mahyanbakhshayesh (2026)	Fuses live multi-modal IoT edge-telemetry with optimized ML (LightGBM/XGBoost) and a Gemini API-powered Oracle APEX dashboard.	Bridges offline analytics and live edge-telemetry, transforming static monitoring into a real-time, RAG-constrained conversational DSS.	Yes

6.2 LLM Decision-Support Case Studies

To evaluate the practical correctness, verifiability, and operational usefulness of the LLM assistant, physical anomaly simulations have been conducted on the hardware prototype. By deliberately inducing specific fault conditions, the LLM's ability to translate the grounded sensor telemetry (as defined in Section 4.1) has been assessed into accurate and actionable maintenance directives.

Case Study 1: Induced Mechanical Obstruction (Generator Jamming)

- **Physical Simulation:** An artificial, high-velocity airflow was applied directly to the anemometer using external equipment, simulating strong wind conditions. Simultaneously, a physical obstruction was intentionally placed against the turbine blades, preventing rotation and energy generation.
- **System Telemetry & Processing:** The sensors recorded a high wind speed (> 15 km/h), while the power and voltage sensors registered near-zero outputs (Volt $< 0.1V$). The Python backend successfully intercepted this physical discrepancy, matching it against the 'Generator Jamming' rule in the deterministic matrix, and flagged a mechanical fault.
- **LLM Output:** The AI assistant immediately issued a critical alert: *"Warning: A severe discrepancy is detected. High wind speeds are present, but turbine voltage is critically low (near 0V). This indicates a generator jamming event or severe rotor obstruction. Immediate Action: Halt the system and visually inspect the blades and rotor-generator coupling for physical blockages."*
- **Evaluation:** This case demonstrates high *correctness* and *usefulness*. Instead of merely reporting "0 Volts" or guessing the fault, the LLM accurately diagnosed the mechanical contradiction (wind vs. power) based on the grounded context. It provided a precise, physically verifiable recommendation that would prevent an operator from assuming the lack of power was due to low wind.

Case Study 2: Induced Thermal Anomaly (Critical Overheating)

- **Physical Simulation:** An external heat-generating device was applied directly to the nacelle temperature sensor, artificially forcing the localized temperature to rise rapidly beyond the critical safety threshold of $50^{\circ}C$.
- **System Telemetry & Processing:** As the nacelle temperature crossed $50^{\circ}C$, the system recognized a thermal breach ($T3 > 50^{\circ}C$), bypassing the standard machine learning inference with a hardcoded safety override for 'Critical Overheating'.
- **LLM Output:** The LLM responded with an emergency diagnostic message: *"CRITICAL ALERT: The nacelle temperature has exceeded safe operational limits (Current: $>50^{\circ}C$). Risk of severe component degradation or fire. Immediate Action: Initiate an emergency thermal shutdown and activate cooling protocols. Do not restart until the nacelle environment stabilizes."*
- **Evaluation:** This test verifies that the LLM reliably adheres to strict industrial safety boundaries without hallucinating secondary, non-existent mechanical faults. The output is highly *useful* as it instantly transitions a numerical threshold breach into an explicit emergency protocol, significantly reducing operator reaction time.

6.3 Limitations

Although SWTFPS is a successful prototype, it has some limitations for real-world use. Specifically, the prominent 98.37% classification accuracy reported for the prototype must be interpreted with academic caution. Because the prototype data was annotated using pseudo-labels generated by the SCADA-trained model rather than independent physical fault verification,

this high accuracy primarily reflects the model's capacity to generalize and refine the 'teacher' model's decision boundaries. Consequently, this pipeline may inherit and propagate the inherent predictive biases or limitations of the baseline classifier.

Recognizing this as a key limitation, our findings should be viewed as a proof-of-concept for multi-modal sensor enrichment rather than a definitive clinical validation of absolute physical accuracy. Second, there is a significant difference between our small prototype and massive industrial turbines; therefore, sensor patterns such as vibration and temperature may not perfectly match those of industrial systems. Third, the affordable sensors and ESP32 microcontrollers used are great for research but may not be sufficiently reliable or durable for the harsh environmental conditions of real wind farms.

7 Conclusion and Future Work

SWTFPS has successfully bridged the gap between industrial SCADA data and an IoT prototype. It demonstrates that strategic data enhancement and a balanced 7-feature model are highly effective in fault prediction. The key achievements are summarised as follows:

- **Integrated LLM:** Developed an LLM-centered interface that allows operators to communicate with the system that provides recommendations in natural language.
- **Real-Time Data Architecture:** Successfully established a live data collection that integrates traditional SCADA data with a wide array of new-generation sensor data, proving that systems can operate in full synchronization for real-time monitoring.
- **Optimised Performance via Hybrid AI:** Conducted a comprehensive comparison of multiple machine learning algorithms. By implementing Centroid-based K-Means balancing to eliminate data bias, the XGBoost model was optimized to achieve superior performance metrics: *Accuracy*: 89.68%, *F1-Score*: 90.08%, *ROC-AUC*: 95.65%, and *Overall Performance*: 65.13% while retrained model using the LightGBM algorithm, achieving 98.37% *Accuracy*, 99.16% *F1-Score*, 98.95% *ROC-AUC* and 94.65% *Overall Performance*.

To further develop the system, future work will focus on Advanced Diagnosis and Deep Learning for sub-categorization of specific mechanical faults; Edge Computing and Sensor Fusion for running models directly on offline hardware; and Digital Twin Integration for developing a 3D digital twin to provide a comprehensive, visual monitoring experience.

Code and data availability

The raw dataset analysed in this study was originally published by Kasimov (2024) and is publicly available at <https://www.kaggle.com/datasets/azizkasimov/wind-turbine-scada-data-for-early-fault-detection>. The source code used for data preprocessing, model training, and generating the figures presented in this manuscript is openly available in the GitHub repository at https://github.com/SaraMahyan/SWTFPS_Wind-Turbine-Failure-Prediction.

Author contributions

MHÖ and MSÜ contributed to the conceptualization and methodology of the study, supervised the research process, managed project administration, acquired funding, and reviewed and edited the manuscript. SM conducted the investigation, developed the software and hardware, provided resources, curated the data, and reviewed and edited the manuscript. İGS contributed to software development, conducted the investigation, curated the data, performed visualization and formal analysis, prepared the original draft of the manuscript, and reviewed and edited the manuscript. BA curated the data, performed formal analysis, contributed to software development and investigation, and reviewed and edited the manuscript.

Competing interests

The authors declare that they have no competing interests.

Disclaimer

Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Acknowledgements

The authors would like to express their sincere gratitude to Dokuz Eylül University, particularly the Department of Computer Engineering, for providing the necessary laboratory facilities, technical infrastructure, and academic support throughout this research. We are especially thankful to the faculty members and technical staff of the department for their valuable guidance, insightful suggestions, and continuous encouragement during the development of the prototype and experimental studies. We also extend our appreciation to Yaşar University for their academic contributions. This work would not have been possible without the collaborative research environment and resources provided by these institutions. The authors utilized VSCode Copilot (GPT-5.4 Mini architecture) to assist with code generation and adapting figures for color accessibility. Additionally, Gemini and Grammarly were used to refine the academic language and grammar of the manuscript. The authors thoroughly reviewed and take full responsibility for the final content.

Financial support

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

- Corley, B., Koukoura, S., Carroll, J., and McDonald, A.: Combination of thermal modelling and machine learning approaches for fault detection in wind turbine gearboxes, *Energies*, 14, 1375, <https://doi.org/10.3390/en14051375>, 2021.
- Khan, P. W., Yeun, C. Y., and Byun, Y. C.: Fault detection of wind turbines using SCADA data and genetic algorithm-based ensemble learning, *Eng. Fail. Anal.*, 148, 107209, <https://doi.org/10.1016/j.engfailanal.2023.107209>, 2023.
- Hsu, J. Y., Wang, Y. F., Lin, K. C., Chen, M. Y., and Hsu, J. H. Y.: Wind turbine fault diagnosis and predictive maintenance through statistical process control and machine learning, *IEEE Access*, 8, 23427–23439, 2020.
- Tan, Y. W. and Carroll, J.: LLMs in Wind Turbine Gearbox Failure Prediction, *Energies*, 18, 4659, <https://doi.org/10.3390/en18174659>, 2025.
- Zhang, Y., Zhao, Z., Zhang, D., and Dai, J.: Optimisation of Large Language Models with Multi-modal Data Training for Fault Diagnosis in Wind Turbine Systems, in: *Proceedings of the 2025 IEEE International Conference on Power Electronics and Energy (PEE)*, 172–176, <https://doi.org/10.1109/peed63748.2025.00038>, 2025.
- Amin, A., Bibo, A., Panyam, M., and Tallapragada, P.: Vibration based fault diagnostics in a wind turbine planetary gearbox using machine learning, *Wind Eng.*, 47, 175–189, <https://doi.org/10.1177/0309524x221123968>, 2022.
- Alzawaideh, B., Baboli, P. T., Babazadeh, D., Horodyvskyy, S., Koprek, I., and Lehnhoff, S.: Wind turbine failure prediction model using scada-based condition monitoring system, in: *2021 IEEE Madrid PowerTech*, 1–6, IEEE, 2021.
- de Lima Munguba, C. F., Ochoa, A. A. V., Leite, G. D. N. P., da Costa, A. C. A., da Costa, J. Â. P., de Menezes, F. D., and de Souza, M. G. G.: Fault detection framework in wind turbine pitch systems using machine learning: Development, validation, and results, *Eng. Appl. Artif. Intell.*, 138, 109307, 2024.
- Inturi, V., Shreyas, N., Chetti, K., and Sabareesh, G.: Comprehensive fault diagnostics of wind turbine gearbox through adaptive condition monitoring scheme, *Appl. Acoust.*, 174, 107738, <https://doi.org/10.1016/j.apacoust.2020.107738>, 2020.
- Meyer, A.: Vibration fault diagnosis in wind turbines based on automated feature learning, *Energies*, 15, 1514, <https://doi.org/10.3390/en15041514>, 2022.
- Ogaili, A. A. F., Hamzah, M. N., and Jaber, A. A.: Enhanced fault detection of wind turbine using extreme gradient boosting technique based on nonstationary vibration analysis, *J. Fail. Anal. Prev.*, 24, 877–895, 2024.
- Pham, D. A. and Han, S. H.: Advanced Machine Learning Model Based on Bi-LSTM and Attention Mechanism for Fault Detection in Wind Turbine Systems, *J. Electr. Eng. Technol.*, 1–14, 2025.
- Puruncajas, B., Castellani, F., Vidal, Y., and Tutivén, C.: Use of artificial neural networks and SCADA data for early detection of wind turbine gearbox failures, *Machines*, 13, 746, <https://doi.org/10.3390/machines13080746>, 2025.
- Santiago, R. A. D. F., Barbosa, N. B., Mergulhão, H. G., Carvalho, T. F. D., Santos, A. A. B., Medrado, R. C., and Nascimento, E. G. S.: Data-driven models applied to predictive and prescriptive maintenance of wind turbine: A systematic review of approaches based on failure detection, diagnosis, and prognosis, *Energies*, 17, 1010, 2024.

- 605 Tan, Y. W., and Carroll, J.: LLMs in Wind Turbine Gearbox Failure Prediction, *Energies*, 18, 4659,
<https://doi.org/10.3390/en18174659>, 2025.
- Yu, X., Tang, B., and Zhang, K.: Fault diagnosis of wind turbine gearbox using a novel method of fast deep graph convolutional networks, *IEEE Trans. Instrum. Meas.*, 70, 1–14, <https://doi.org/10.1109/tim.2020.3048799>, 2021.
- Zeng, X., Yang, M., Feng, C., and Tang, Y.: A generalised wind turbine anomaly detection method based on combined probability estimation model, *J. Mod. Power Syst. Clean Energy*, 11, 1136–1148, 2022.
- 610 Zhang, G., Li, Y., and Zhao, Y.: A novel fault diagnosis method for wind turbine based on adaptive multivariate time-series convolutional network using SCADA data, *Adv. Eng. Inform.*, 57, 102031, 2023.
- Zhang, Y., Zhao, Z., Zhang, D., and Dai, J.: Optimization of Large Language Models with Multi-modal Data Training for Fault Diagnosis in Wind Turbine Systems, 2025 IEEE International Conference on Power Electronics and Energy (PEE),
615 172–176, <https://doi.org/10.1109/peed63748.2025.00038>, 2025.
- Zhang, Z. Y. and Wang, K. S.: Wind turbine fault detection based on SCADA data analysis using ANN, *Adv. Manuf.*, 2, 70–78, 2014.
- Kasimov, A.: Wind turbine SCADA data for early fault detection (Version 1) [data set], Kaggle, <https://www.kaggle.com/datasets/azizkasimov/wind-turbine-scada-data-for-early-fault-detection>, 2024.
- 620 MathWorks: ThingSpeak: An IoT analytics platform service, <https://thingspeak.com>, (accessed 29 May 2026).
- Naidu, G., Zuva, T., and Sibanda, E. M.: A review of evaluation metrics in machine learning algorithms, in: *Computer Science On-line Conference*, 15–25, Springer, Cham, 2023.
- Islam, M. M., Rifat, H. R., Shahid, M. S. B., Akhter, A., Uddin, M. A., and Uddin, K. M. M.: Explainable machine learning for efficient diabetes prediction using hyperparameter tuning, SHAP analysis, partial dependency, and LIME, *Eng. Rep.*, 7,
625 e13080, 2025.