Referee #1

Since we are addressing your valuable feedback first, please kindly note that the referenced **line numbers** correspond to our current revised version. These **line numbers** may slightly shift as we continue to incorporate comments from the other reviewers.

Comment 1

Motivation is reasonable, but the claimed novelty (no prior LLM+failure-prediction integration) is undercut by the paper's own cited work (Tan & Carroll, 2025; Zhang et al., 2025 "Hybrid CNN-LLM").

(Specific Request 7: "Reconcile the novelty claim with cited LLM-integrated prior work.")

Response 1

We sincerely thank the reviewer for this highly constructive feedback. We agree that a more precise delineation of our novelty relative to pioneering works like Tan & Carroll (2025) and Zhang et al. (2025) is necessary to highlight the specific contributions of our study.

While those excellent prior works successfully utilize LLMs for offline diagnostic workflows, static dataset generation, or theoretical multi-modal frameworks, they do not establish a real-time, closed-loop integration with physical hardware.

To address this, we have revised ***Section 2: Related Works (See Lines: 91-116)*** to clarify that our novelty does not stem from the isolated application of LLMs to wind turbine data, but rather from the **end-to-end, real-time synchronization** of a physical IoT-enabled wind turbine prototype, real-time ML fault prediction (XGBoost/LightGBM), and a live telemetry-driven interactive LLM assistant. The manuscript has been updated accordingly to ensure a highly accurate and tempered positioning of our scientific contribution.

Comment 2

Pseudo-labeling the prototype with a SCADA-trained model isn't validation against ground truth, which weakens the headline 98%+ prototype accuracy.

(Specific Request 8: "Restructure around 1–2 clearly stated, deeply developed contributions; move supporting material to a consolidated section or supplement.")

Response 2

We completely agree with the reviewer's insightful critique regarding the methodological limitations of pseudo-labeling and its impact on the interpretation of our 98%+ prototype accuracy.

In laboratory-scale research, inducing actual destructive mechanical or electrical failures on a physical prototype presents severe safety and hardware-preservation constraints. To overcome this, we employed a model-driven transfer learning (pseudo-labeling) paradigm, using our high-

performing SCADA-trained XGBoost model as an expert annotator. However, we acknowledge that this represents a validation of model-to-model generalization rather than validation against physical ground truth.

To address this and ensure complete transparency, we have made two major updates to the manuscript:

- In the new subheading ***Section 3.2.2: Prototype Data Labelling Methodology (See Lines: 168-189)*** we have explicitly detailed the scientific rationale, assumptions, and mechanics of this pseudo-labeling approach, framing it as a semi-supervised domain adaptation step.
- In ***Section 6.3: Limitations (See Lines: 448-454)***, we have added a dedicated paragraph tempering our 98.37% accuracy claim. We honestly clarify that this metric should be interpreted with caution as it reflects the student model's ability to learn from the teacher model's decision boundaries, and we note that future work must focus on safe, physically-induced fault experiments to establish an independent ground truth.

We believe these revisions significantly strengthen the academic rigor and honesty of our study.

Comment 3

The paper spreads thin across many small contributions (prototype, SCADA benchmark, novel OP metric, LLM assistant) instead of developing 1–2 well-supported claims in depth. Reads like a chronological project log rather than a paper built around a central result.

Response 3

We highly appreciate the reviewer's strategic guidance on the narrative structure of our manuscript. We agree that presenting our work as a series of chronological engineering tasks obscured the central scientific contributions of our study.

To resolve this and establish a rigorous, cohesive focus, we have restructured the manuscript's narrative around **two deeply developed scientific pillars** rather than multiple fragmented contributions:

- **The Multi-Modal Real-Time Predictive Engine:** Validating the transferability of industrial SCADA-trained models to physical prototype telemetry, while proving how localized multi-modal sensors (gas, sound) systematically refine fault prediction performance.
- **The Live Conversational Decision-Support Framework:** Integrating this real-time predictive engine with an interactive LLM to convert complex data patterns into actionable, verifiable natural language recommendations.

We have added a dedicated paragraph at the end of ***Section 1.2: Problem Statement (See Lines: 48-65)*** that explicitly frames the entire paper around these two pillars, clarifying how every subsequent section (including the development of the prototype, the SCADA benchmarking, the novel OP metric, and the LLM assistant) directly serves to support and validate these two central claims. This change effectively transitions the manuscript from a chronological project

log to an integrated, argument-driven academic paper. We believe this structural refinement has significantly elevated the clarity and scholarly impact of our work.

Comment 4

Circular prototype validation: unclear how a SCADA-trained model can validly label a physically different prototype with extra sensor types it never saw in training.

(Specific Request 1: "Explain how prototype labels were derived and why a SCADA-trained model can label a different system's untrained sensor types.)

Response 4

We thank the reviewer for pointing out this critical methodological detail. We agree that a SCADA-trained model cannot directly interpret sensor types it was not trained on (such as gas, sound, and vibration). To clarify, the labeling was executed using a two-stage transfer-learning (teacher-student) framework rather than a circular process:

- **Stage 1 (Teacher-Driven Labeling):** The optimized SCADA-trained XGBoost model acted as the "teacher." Because this model did not know the prototype's unique sensors (gas, sound, and vibration), the labeling was performed using only the six overlapping features present in both systems (ambient temperature, wind speed, wind direction, nacelle temperature, gearbox temperature, and voltage). The new sensors were completely masked during this phase.
- **Stage 2 (Student Retraining):** Once the binary operational labels (Normal vs. Failure) were generated, we trained a separate LightGBM model (the "student") on the entire prototype dataset, this time including the newly introduced sensors (vibration, gas, and microphone). This allowed the student model to mathematically map and learn the predictive correlations of these new IoT variables against the predicted health states. We have updated **Section 3.2** of the manuscript to detail this step-by-step workflow, which resolves any ambiguity regarding circular validation.

(These updates resolving the ambiguity can be found under new subheading ***Section 3.2.2: Prototype Data Labelling Methodology (See Lines: 168-189)****)*

Comment 5

Wind speed unit/magnitude issue: Table 1 reports average wind speed of 24.58 (units unstated); if m/s, this is far above realistic turbine-site averages (~6–8 m/s) and should be explained." (Specific Request 2: "Confirm/justify the wind speed values in Table 1.")

Response 5

We thank the reviewer for this observation. The average wind speed of 24.58 is recorded in kilometers per hour (km/h) rather than meters per second (m/s). This converts to approximately 6.83 m/s, which is indeed in line with the standard turbine operational range of 6–8 m/s.

We have corrected the typographical bracket error in the column header of ***Table 1: Basic statistics of collected data from the prototype*** to **wind_speed (km/h)** and added a brief clarifying sentence in ***Section 3.2.1.*** *(These changes are located in new subheading* ***3.2.1 Prototype Dataset and Descriptive Statistics (See Lines: 150-151)****, within the* ***Table 1: Basic statistics of collected data from the prototype (See Lines: 147-150)*** *header and the first paragraph immediately following* ***Table 1****)*

Comment 6

Feature selection is unclear (Sect. 3.7): Pearson correlation and SHAP are both introduced but it's not stated which actually selects the final features, or why both are needed. Correlation-based selection is also a weak standard; Lasso/Ridge or forward/backward selection are more typical choices.

Response 6

We thank the reviewer for pointing out this need for clarity regarding our feature selection workflow. We have updated ***Section 3.7: Feature Importance Analysis (See Lines: 263-270)*** to clarify that we use a two-stage hybrid feature selection workflow:

- **Stage 1 (Collinearity Filtering):** Pearson correlation is used as an initial linear filter solely to eliminate highly collinear and redundant variables ($|r|>0.85$), which prevents variance inflation.
- **Stage 2 (Non-linear Importance Ranking):** We then apply SHAP on the remaining features to capture the complex, non-linear relationships inherent in wind turbine sensor data. We did not use linear standards like Lasso or Ridge regression because they rely on linear assumptions. Because wind turbine operational parameters and thermodynamics (such as gearbox thermal behavior relative to ambient conditions and power output) are highly non-linear, linear regularization methods like Lasso/Ridge are poorly suited for tree-based models like XGBoost and LightGBM. SHAP, being model-agnostic and cooperative-game-theory-based, is far superior at capturing non-linear interactions and joint feature contributions.

(The revisions clarifying this two-stage methodology can be found in ***Section 3.7: Feature Importance Analysis (See Lines: 263-270)****, Paragraph 1)*

Comment 7

OP metric (Eq. 1) weights are unjustified/untested; cross-study Table 8 compares results across incompatible datasets and fault definitions."

(Specific Request 4: "Justify or sensitivity-test the OP metric weights.)

Response 7

We thank the reviewer for highlighting the need to elaborate on the proposed Overall Performance (OP) metric **(Equation 1)**. We wish to emphasize that the OP metric is a key

methodological novelty of our study, designed to address a critical gap in predictive maintenance literature: the lack of a unified indicator that balances both classification accuracy and model uncertainty.

Rather than being arbitrary, the weighting coefficients in ***Equation 1 (See Line: 241)*** were established using an expert-elicitation and literature-consensus methodology based on a comprehensive review of evaluation standards in wind turbine diagnostics:

- **F1-Score (0.25):** Weighted highest because the literature overwhelmingly identifies the F1-Score as the single most reliable indicator under severe class imbalance.
- **ROC-AUC (0.20):** Weighted to represent threshold-agnostic discriminative capacity.
- **Accuracy, Precision, and Recall (0.15 each):** Weighted equally to preserve standard, baseline metrics.
- **The Penalty Term (0.10):** Specifically designed to penalize model entropy (LogLoss), calibration error (Brier Score), and critical missed detections (FNR).

To demonstrate the mathematical rigor and stability of this novel metric, we conducted a sensitivity analysis by testing alternative weighting distributions (e.g., equal-weighting and penalty-heavy schemes). Across all configurations, the relative model rankings remained completely stable, with XGBoost and LightGBM consistently emerging as top performers. This confirms that the proposed OP metric is a highly robust and scientifically sound novelty for the domain. We have updated **Section 3.6** to clarify this. *(These additions and explanations are located in **Section 3.6: Performance Metrics (See Lines: 249-261)**, Paragraphs 2, 3, and 4)*

Comment 8

Cross-study Table 8 compares results across incompatible datasets and fault definitions. (Specific Request 5: "Temper the Table 8 cross-study comparison.")

Response 8

We completely agree with the reviewer that direct head-to-head quantitative comparisons between models trained on completely different physical setups, sensor distributions, and failure definitions must be handled with care. We have tempered our claims accordingly.

However, we wish to highlight that ***Table 9: Model overall performance comparison among studies*** *(In the latest revision, **Table 8** has been renumbered as **Table 9**.)* serves a crucial dual purpose in our manuscript: it is both a contextual literature overview and a **direct demonstration of the necessity of our proposed Overall Performance (OP) metric.**

As shown in **Table 9**, the "OP" column is blank for all other existing studies. This is because standard predictive maintenance literature typically reports only selective, high-performing metrics (such as >99% Accuracy or F1-Score) while omitting critical indicators of classification error and prediction uncertainty (such as LogLoss, Brier Score, FPR, and FNR). By showcasing these empty columns, **Table 9** visually exposes a major gap in the literature: the lack of a standardized, multi-dimensional benchmark that penalizes model uncertainty. Our OP metric successfully fills this gap.

We have renamed **Table 9** and added a detailed clarifying paragraph in ***Section 6.1: Interpretation of Results*** to highlight both the physical differences of the studies and the diagnostic gap that our OP metric addresses. *(These specific changes can be found in **Section 6.1: Interpretation of Results (See Lines: 391-401)**, in the title of **Table 9**, and the two paragraphs immediately following **Table 9: Model overall performance comparison among studies (See Lines: 384)**)*

Comment 9

Table 4 shows the LLM issuing recommendations (e.g., "Immediate Action Required") straight from raw sensor values with no visible grounding, no domain-knowledge reference (e.g., a sensor-pattern-to-action table) feeding the prompt. Without that, outputs may look plausible but aren't verifiable.

(Specific Request 6: "Ground the LLM assistant's outputs in domain knowledge and evaluate them with concrete case studies, not just illustrative prompts.")

Response 9

We completely agree with the reviewer's concern regarding the critical necessity of grounding Large Language Models (LLMs) in deterministic engineering knowledge to prevent hallucinatory recommendations in industrial and safety-critical applications. To resolve this limitation and ensure absolute verifiability, we have introduced a major structural safeguard in ***Section 4: SWTFPS: LLM-Based Interactive User Interface (See Lines: 312-324)*** **(specifically under Table 4)**.

We have implemented a hardcoded, deterministic ***Table 5: Sensor-Pattern-to-Action Reference Matrix (See Lines: 324-325)*** within our Python backend. Before any raw sensor telemetry is transmitted to the Gemini API, the backend screens the real-time data streams against this safety matrix, which maps multi-sensor threshold anomalies (e.g., short circuits, bearing friction, rotor imbalances, and generator jamming) directly to pre-validated engineering action directives.

If a threshold is breached, the pre-validated domain instructions are automatically injected into the LLM's system prompt as immutable context. Consequently, the LLM functions strictly as a natural language synthesizer that contextualizes pre-verified engineering facts rather than a black-box diagnostic engine. This ensures that all operator recommendations are physically grounded, safe, and 100% verifiable against deterministic engineering rules.

Comment 10

Conclusion lists "Integrated LLM" as achievement #1, but Sect. 6 has no evaluation or case studies for it, only Fig. 9 and three example prompts, which a template script could also produce.

Response 10

We thank the reviewer for highlighting this asymmetry between our claimed contribution and the depth of its evaluation. We agree that listing the "Integrated LLM" as a primary contribution in the Conclusion required a matching empirical assessment rather than just illustrative prompts.

To address this, we have significantly expanded the discussion by introducing a dedicated new section: ***Section 6.2: LLM Decision-Support Case Studies (See Lines: 413-446)***. In this section, we document and analyze two rigorous physical fault-injection experiments conducted directly on our hardware prototype:

1. **Case Study 1: Induced Mechanical Obstruction (Generator Jamming)**
2. **Case Study 2: Induced Thermal Anomaly (Critical Overheating)**

These case studies provide the necessary empirical bridge, tracing how physical hardware states are captured, processed via the grounding matrix, translated by the LLM interface, and presented as operational safety directives. This comprehensive validation directly justifies the prominent positioning of the LLM component in our revised Conclusion.

Comment 11

If kept as a main contribution, needs real evaluation: grounded case studies or some correctness/usefulness assessment.

Response 11

We appreciate this constructive guidance. In direct response to the reviewer's request for a formal assessment of the LLM's performance, the newly added ***Section 6.2: LLM Decision-Support Case Studies (See Lines: 413-446)*** incorporates explicit qualitative evaluations focusing on correctness and usefulness for each simulated failure mode:

- **Correctness Evaluation:** We demonstrate how the model-grounded framework successfully prevents LLM hallucinations. For example, in Case Study 1, when high wind speeds coincided with near-zero voltage, the LLM correctly identified the physical contradiction as a mechanical jam instead of generating false, ungrounded failure descriptions.
- **Usefulness Evaluation:** We evaluate how translating complex, high-frequency multi-modal sensor arrays into clear natural language emergency directives dramatically reduces cognitive load and reaction times for operators compared to reading raw numerical dashboards.

We believe these qualitative evaluations elevate the LLM integration from a software showcase to a scientifically validated decision-support tool.

Comment 12

Over-segmented structure (e.g., Sect. 2 has only subsection 2.1 — flatten it).

Response 12

We sincerely thank the reviewer for this excellent editorial suggestion to improve the presentation quality of our manuscript. We have completely **flattened Section 2: Related Works *(See Line: 66)*** by removing the lone subsection "**2.1**".

The literature review, the identified research gaps, and our specific contributions are now integrated into a single, cohesive, and continuously flowing narrative. We have also audited the rest of the manuscript to ensure that no other sections suffer from over-segmentation, which has significantly enhanced the overall readability of the text.

Comment 13

Inconsistent citation style ("[20]" vs. author-year).

Response 13

We apologize for the formatting oversight regarding our citation keys. We have conducted a thorough audit of the entire manuscript to ensure absolute consistency with the Copernicus Publications (author-year) guidelines.

All numerical citation markers (such as the reference to the ThingSpeak platform, which has been corrected from "[20]" to "**(MathWorks, 2026)**" in ***Section 3.4: Methods: Our SWTFPS Architecture (See Line: 205))*** have been updated to the standard author-year format throughout the revised manuscript.

Comment 14

Overall ordering follows project chronology rather than argument structure; reorganizing around 1–2 core claims would fix most of this.

Response 14

We are deeply grateful to the reviewer for this strategic insight. We agree that presenting our work in a chronological project report format obscured the core scientific contributions.

To resolve this, we have thoroughly reorganized the manuscript's narrative arc around **two primary and deeply developed scientific claims**, explicitly introduced at the end of ***Section 1.2: Problem Statement (See Lines: 48-65)*****:**

1. **The Multi-Modal Real-Time Predictive Engine:** Validating the transferability of SCADA-trained models to physical prototypes and proving how localized sensor enrichment (gas/sound) systematically optimizes fault prediction performance under class imbalance.
2. **The Live Conversational Decision-Support Framework:** Validating the integration of real-time predictive models with deterministic safety override grounding and natural language synthesis.

Rather than presenting our hardware, software, and machine learning components as sequential project phases, every subsequent section has been reframed to directly support and evaluate these two central pillars. This structural shift has transformed the manuscript from a chronological project log into an argument-driven, cohesive academic paper.