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

*Saygi et al. wes-2026-95*

## Summary

The manuscript presents SWTFPS, a system combining a custom small-scale prototype with a public industrial SCADA dataset, an XGBoost/LightGBM fault-classification pipeline, and an LLM-based conversational interface for querying turbine status. Models are trained and compared on the SCADA data, and the best SCADA-trained model is then used to generate labels for the prototype's own sensor stream, which is used to train a second model reported with very high accuracy.

## 1. Scientific significance — Poor to Fair

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

## 2. Scientific quality — Poor

- **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.
- **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.
- **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.
- OP metric (Eq. 1) weights are unjustified/untested; cross-study Table 8 compares results across incompatible datasets and fault definitions.
- Positive: solid 11-model benchmarking, code/data available.
- 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.
- 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.
- If kept as a main contribution, needs real evaluation: grounded case studies or some correctness/usefulness assessment.

## 3. Presentation quality — Fair

- Over-segmented structure (e.g., Sect. 2 has only subsection 2.1 — flatten it).
- Inconsistent citation style ("[20]" vs. author-year).
- Overall ordering follows project chronology rather than argument structure; reorganizing around 1–2 core claims would fix most of this.

## Recommendation

Major revision. SCADA-level ML benchmarking is solid, but the LLM component — presented as a main achievement has no supporting evaluation or grounding.

## Specific requests to authors

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

2. Confirm/justify the wind speed values in Table 1.
3. Clarify the feature-selection logic in Sect. 3.7; consider standard alternatives to correlation filtering.
4. Justify or sensitivity-test the OP metric weights.
5. Temper the Table 8 cross-study comparison.
6. Ground the LLM assistant's outputs in domain knowledge and evaluate them with concrete case studies, not just illustrative prompts.
7. Reconcile the novelty claim with cited LLM-integrated prior work.
8. **Restructure around 1–2 clearly stated, deeply developed contributions; move supporting material to a consolidated section or supplement.**