

Industrial IoT for Predictive Maintenance: Addressing Key Research Gaps in Industrial Deployment

Arpit Verma, Sarvesh Kumar Srivastava, Ashish Gupta, Ashish Singh

School of Computer Application, BBD University, Lucknow, Uttar Pradesh, India

Published October 2026

Abstract

Many studies have already been conducted on the replacement of reactive and schedule maintenance with predictive maintenance in the context of the Industrial Internet of Things, and the reported results are very promising. However, after a careful inspection of this literature, several gaps emerged: most of the systems rely on a single sensor; all the data used are laboratory-based, the detection, diagnosis and prognosis phase do not communicate with each other; predictions in deep learning are not easy to explain; deployment is costly for small and medium sized industries, and action-oriented recommendations are still only a concept and not yet a reality. Rather than repeating these gaps, we are proposing an actual framework designed to address each gap. It combines vibration, thermal, acoustic, and oil-condition signals into a single edge sensor; executes detection, diagnosis and prognosis under one single pipeline and avoids needing to involve three individual models, provides explainable AI (XAI) layer to enable the prediction with a reason attached, and is based on low-cost, off-the-shelf hardware to ensure that small industrial users do not get priced out. The paper then describes the working process, the machine learning methods used, a case scenario, and the expected benefits and limitations, and concludes with the next steps towards fully prescriptive maintenance.

Keywords: Industrial Internet of Things; predictive maintenance; sensor fusion; explainable AI; Industry 4.0; edge computing

1 Introduction

Machines that continue to operate are going to wear out — it's inevitable. The important thing is how a company reacts to that wear – it can mean the difference between production cost and worker safety. When the maintenance is reactive (no maintenance work until the machine actually breaks down), it results in unplanned downtime and lost output nearly every time. Scheduled maintenance is designed to defuse the problem by replacing parts at regular times without regard to need, but that can also mean that unnecessary parts are replaced before they need to be. Both these issues can be overcome by the Industrial Internet of Things, which keeps the sensors on the machine and continues to feed condition information – vibration, temperature, sound, lubricant quality – to a system that is actually observing. With that information, fed into machine learning, it's now possible to predict a failure before it occurs. Predictive Maintenance in practice.

There is already a significant amount of research demonstrating how Predictive Maintenance, based on the Industrial Internet of Things (Industrial IoT), can be done and is effective, at least in theory. But as we examined the published literature ourselves a theme emerged: most of the published systems are performed in a laboratory setting and not a real factory; most of the systems are based

on only one sensor; most of the systems predict a failure without offering an explanation as to why; and few of the systems discuss the cost to the smaller industry of implementing this. We didn't view these as a list of things someone else would solve, but rather as requirements and created a structure to address each one.

The rest of the paper is laid out as follows. The related work is discussed first, followed by the specific gaps that informed our design decisions. The framework itself is then presented, with each part connected back to the gaps, followed by the working process of the system and the machine learning techniques employed. An illustrative case scenario, the expected benefits, an evaluation strategy and the limitations are then discussed, and the paper closes with the scope for future work and the conclusions.

This aligns with the broader trend of the field from Industry 4.0 to Industry 5.0, which is not about collecting data but ensuring that the human engineer remains part of decision making processes that demand human thought and oversight, such as when a machine needs to be taken offline for maintenance. A system that blindly returns a failure probability is not really going to support that, an engineer viewing a number without any explanation has no real way to assess the trustworthiness of the number. For this very reason, explainability and human in the loop design are also important factors we emphasized in this paper, as are sensor fusion and cost.

2 Related work

A number of large-scale surveys have already tried to map out the predictive maintenance space within Industry 4.0. Meitz et al. [1] went through 249 publications and found that the lack of standardized, production-relevant benchmark datasets keeps coming up as one of the biggest obstacles to comparing different approaches fairly. A separate, broader review of AI-enabled predictive maintenance found something similar — even with publication numbers growing fast, most studies are still only validated on lab-generated datasets, and the authors specifically called out better model interpretability as something the field still needs [5].

On the modeling side, one survey of deep learning architectures compared recurrent, convolutional, and hybrid model families and looked at how each one fits into different stages of the maintenance pipeline [3]. Later, broader reviews of AI- and IoT-powered smart maintenance went further and organized the literature by industrial domain — smart factories, rotating machinery, mobile robotics — and in doing so, kept flagging the same issue: detection, diagnosis, and prognosis are rarely built to work together [4].

Sensor fusion stood out to us as one of the more active responses to the single-sensor problem. One explainable predictive maintenance study on rotating machines found that fusing multiple sensors beat single-sensor approaches across every model they tested, and they paired this with LIME and Random Forest-based explanations to keep things interpretable [11]. A similar pattern showed up in a deep learning study that fused thermal imaging with electrical sensor data for network-hardware maintenance — the multi-modal fusion model caught early faults that neither signal picked up on its own [12]. A more recent review covering digital twins, multi-sensor fusion, and federated learning took this further, arguing that combining different kinds of sensing with distributed, information-sharing setups is basically where the next generation of predictive maintenance is heading [10].

Explainability is the other theme that kept surfacing. Gawde et al. [7] built a multi-sensor, multi-fault diagnosis pipeline for rotating machines and used LIME, SHAP, partial dependence plots, and

individual conditional expectation to make sense of what the model was actually doing. Pashami et al. [8] framed explainable AI (XAI) as the thing standing between a complex model and the engineer who has to act on its output — without it, the gap is just too wide. A 2025 review of human-in-the-loop XAI backed this up, finding that interactive, explanation-driven systems measurably improved maintenance decisions compared to systems that just hand over a score with no reasoning [9]. Prescriptive maintenance, on the other hand — actually recommending what to do, not just predicting that something will fail — is still fairly early. A systematic review in this space found limited conceptual clarity and very little real-world validation so far [6]. Taken together, these four strands — real-world validation, sensor fusion, explainability, and early prescriptive work — are exactly what we tried to build on for the framework in this paper.

One thing we noticed is that cost barely gets discussed in any of this. Most of the architectures in the literature are demonstrated on research-grade sensor setups or inside large manufacturing partnerships, and very few papers actually say what it would cost to install their system, let alone how it could be scaled down for a smaller operation. Given how much of industrial employment runs through small and medium-sized enterprises, that is a real gap — not a side note — which is why we treated cost as an explicit design constraint in the proposed framework rather than something to mention at the end.

3 Research gaps in existing systems

Based on all of the work, six gaps emerged that we felt guided our framework for this paper. First, most predictive models are created, and tested, on data generated from the laboratory, which is difficult to believe works as well on the real factory floor [5]. Second, although the evidence is clear, that using a combination of sensors can yield more accurate notice of fault prediction [11, 12] is not the case with most of the deployed systems. Third, detection, diagnosis, and prognosis are typically built and assessed as isolated challenges and only very few systems integrate all three in a pipeline [4]. Fourth, many deep learning models remain useful black boxes – predicting a fault, but not explaining why – which is a reason why engineers are not likely to act on the results of such models [7–9]. Fifth, no one speaks about the real costs of sensors, connectivity and cloud infrastructure, effectively excluding everyone except the big manufacturers. Sixth, moving from “this will fail” to “here's what to do about it” (prescriptive maintenance) is still an early and poorly-resolved research stream [6]. These six gaps are the very same gaps that the proposed framework is all about.

4 Proposed framework

We designed this framework so that every part of it ties back to one of the gaps from the research-gaps section — sensing, modeling, and explanation were not treated as separate boxes to tick off, but as pieces that need to work together. Figure 1, explained in the next subsection, lays out the overall data flow from sensing all the way to a recommended action.

4.1 System architecture overview

The framework runs across five layers. At the bottom, the multi-sensor layer attaches vibration, thermal, acoustic, and oil-condition sensors to whichever machine is being monitored. Just above that, the edge layer handles local preprocessing and fuses these signals early, which cuts down how much data needs to travel out and lets the system respond quickly. The connectivity layer then moves this processed data to the cloud using lightweight industrial protocols like MQTT and OPC-UA. The

cloud layer is where the real work happens — this is where detection, diagnosis, prognosis, and the explainable AI module all live together. Finally, the recommendation layer takes all of that and puts it in front of a maintenance engineer through a dashboard, along with the reasoning and a suggested next step.

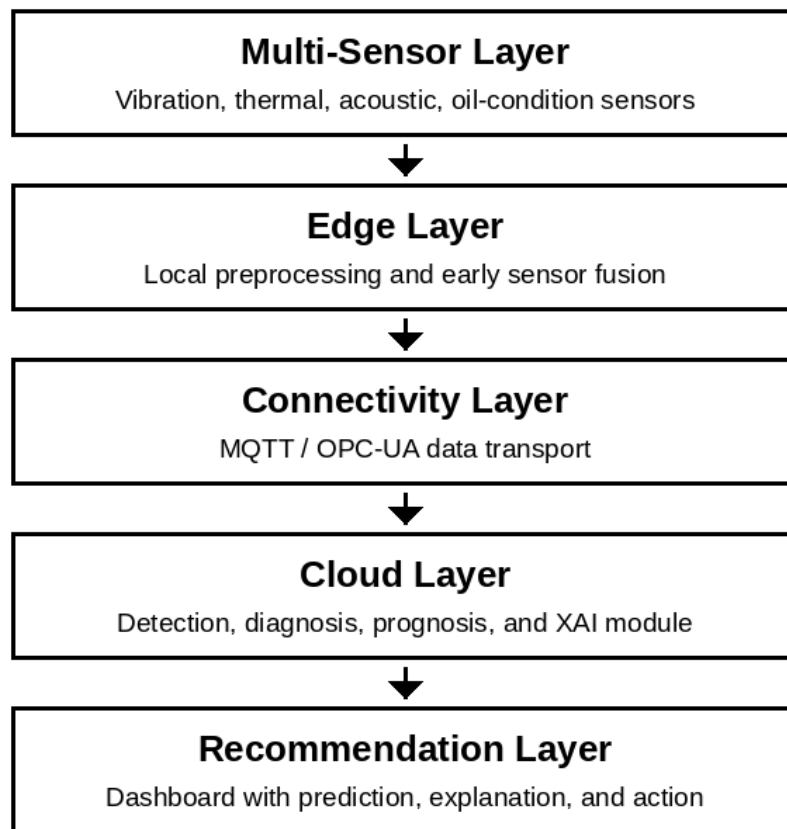


Figure 1. System architecture of the proposed framework.

4.2 Multi-sensor data fusion layer

This part directly answers the single-sensor problem from the research-gaps section. Instead of leaning on one signal, the framework fuses vibration, thermal, acoustic, and oil-condition data together. We followed the same early-fusion approach that performed well in related work [11, 12] — raw signals are time-aligned and combined right at the edge before the model ever sees them. This matters in practice because a single sensor can genuinely miss things: a bearing fault might show up as a very subtle vibration pattern that's easy to overlook, but paired with a slower, more obvious temperature rise, the combination is much harder to miss.

4.3 Edge–cloud hybrid processing for cost-effective deployment

This is where we tried to deal with cost directly. Lightweight preprocessing and anomaly screening happen on cheap edge hardware sitting right next to the machine, and only the heavier diagnosis and prognosis work gets pushed to the cloud. Splitting it this way keeps the constant data-transmission and cloud-compute bill down — which is usually what makes full predictive maintenance setups expensive in the first place — while still letting the cloud-side models use the complete historical dataset when it actually matters, i.e. when generating a diagnosis.

4.4 Integrated end-to-end pipeline

Prior surveys kept pointing out that detection, diagnosis, and prognosis are usually built as separate, disconnected pieces [4], so we built ours as three stages of one pipeline instead. Detection flags that something looks off; diagnosis takes the same fused multi-sensor features and figures out what kind of fault it probably is; and prognosis estimates the Remaining Useful Life (RUL) of whatever part is affected. Because all three stages are working off the same fused feature representation, the diagnosis and prognosis outputs stay consistent with each other — they're not coming from two models that have never spoken to one another.

4.5 Explainable AI module

This part exists because of how often “black box” came up as a complaint in the papers we reviewed [7–9]. The cloud layer includes an explainability module built on SHAP and LIME, and for every diagnosis it produces, it also generates a ranked list of which sensor features actually drove that prediction — displayed right next to the prediction itself, not buried in a separate report. So an engineer looking at a bearing-fault alert can actually see that it was mostly a vibration-frequency anomaly and not the temperature reading, which makes the output easier to trust and supports the kind of human-in-the-loop decision-making that [9] found genuinely improves maintenance outcomes.

4.6 Toward prescriptive recommendations

Prescriptive maintenance is still an open problem, as [6] makes clear, so we didn't try to fully solve it. Instead, the recommendation layer takes each diagnosed fault and its estimated RUL and maps it to a short list of candidate actions — schedule an inspection within a given window, order a specific part, or reduce load until the next planned stoppage — pulled from a rule base built around the fault categories the system already knows about. We see this as a reasonable first step toward prescriptive maintenance, not a finished answer, since fully automated prescriptive reasoning is still very much an open research question.

5 Working process

Once it is running, the framework loops continuously (Figure 2). Sensors on the machine keep recording vibration, temperature, acoustic, and oil-condition data without a break. At the edge, this data gets cleaned up, time-aligned, and fused, and a lightweight anomaly screen checks whether anything looks off compared to normal operation. If it does, the fused features get sent up to the cloud, where the diagnosis model works out the likely fault type and the prognosis model estimates how much useful life is left. From there, the explainable AI module generates a feature-importance explanation for that specific diagnosis, and the recommendation layer turns it into a suggested action. All of it — the prediction, the explanation, and the recommendation — shows up together on the engineer's dashboard, so the loop goes from a raw sensor signal all the way to something the engineer can actually act on.

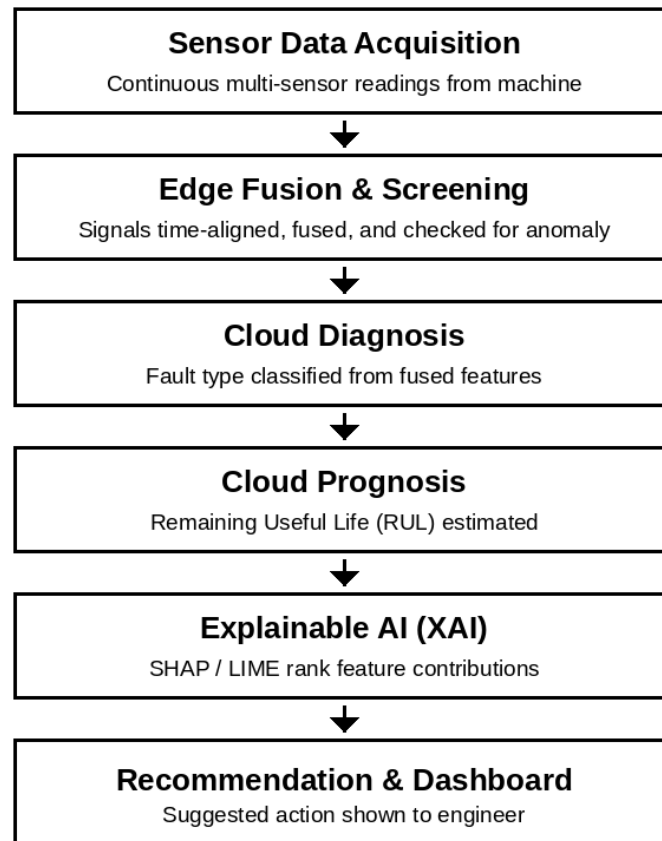


Figure 2. Working process of the proposed framework, from sensor data to recommended action.

6 Machine learning techniques employed

For diagnosis, we went with a Random Forest classifier to assign fault types from the fused features — it has solid baseline performance and, just as importantly, it plays well with SHAP-based explanations. For prognosis, we used a Long Short-Term Memory (LSTM) network to track the degradation trend over time and estimate Remaining Useful Life, since LSTMs are built for exactly this kind of gradually-developing, time-series pattern. Where thermal image data is available, we also bring in a Convolutional Neural Network (CNN) to pull out visual fault signatures, which lines up with the multi-modal CNN-based approach that [12] showed improves early detection. The explainability module runs SHAP and LIME on top of all of this, so we are not giving up prediction accuracy just to get something easier to explain.

Table 1 lays out how each technique lines up with a specific stage of the pipeline, along with why we picked it over the alternatives.

Table 1. Machine learning techniques used in the proposed framework.

Technique	Pipeline stage	Reason for selection
Random Forest	Diagnosis (fault classification)	Strong baseline accuracy; compatible with SHAP explanations
LSTM	Prognosis (RUL estimation)	Captures gradual, time-series degradation trends
CNN	Diagnosis (thermal / image data)	Extracts visual fault signatures from thermal images

Technique	Pipeline stage	Reason for selection
SHAP / LIME	Explainability layer	Ranks feature contributions behind each prediction

7 Illustrative case scenario

Here's a scenario to make this more concrete. Say a manufacturing facility fits a critical set of rotating machines with vibration, temperature, and acoustic sensors under this framework. Edge devices keep fusing and screening the data, and at some point the anomaly screen flags a deviation on one machine. The cloud pipeline diagnoses an early-stage bearing fault and puts the remaining useful life at roughly two weeks. The explainability module then shows that this diagnosis came mainly from a specific vibration-frequency band, with a smaller contribution from a gradual temperature rise — which gives the maintenance engineer an actual reason to trust the alert, rather than just a number to take on faith. The recommendation layer suggests getting the bearing inspected within the next five working days instead of waiting around for the next scheduled maintenance slot. This is basically the pattern we kept seeing across the literature: multi-sensor, explainable predictive maintenance lets you step in earlier and with more confidence than single-sensor or opaque systems typically allow [7, 11, 12]. Table 2 gives an illustrative (hypothetical) example of the output for this scenario.

Table 2. Illustrative (hypothetical) model output for the bearing-fault scenario.

Output field	Example value
Vibration reading	7.8 mm/s RMS (normal range: below 4.5 mm/s)
Temperature reading	82°C (gradual rise over 10 days)
Diagnosed fault type	Bearing wear (mechanical fault class)
Estimated Remaining Useful Life	13–16 days
Prediction confidence	91%
Top contributing feature (XAI)	Vibration, 2–4 kHz band (62% contribution)
Recommended action	Schedule bearing inspection within 5 working days

Now picture a second, smaller facility going through the same thing — this one helps show the cost side of the framework. Instead of putting a full cloud-connected sensor suite on every single machine, this facility only equips its three highest-risk machines with the complete multi-sensor edge node. The lower-risk equipment gets a reduced setup — just temperature and vibration — but it still plugs into the same cloud diagnosis and explainability pipeline. This kind of tiered rollout is only possible because of the edge-cloud split from the edge–cloud processing section, and it means a facility that can't afford to sensor-up everything at once doesn't have to. They can start small and expand later.

8 Expected benefits

Since the framework was built around the six gaps from the research-gaps section, we'd expect it to come with a few real advantages over single-sensor, black-box systems. Multi-sensor fusion should cut down false alarms and improve diagnostic accuracy, which lines up with what the fusion studies we cited found [11, 12]. The edge-cloud split should bring deployment costs down, which matters a lot for smaller industries trying to get in on this. Because detection, diagnosis, and prognosis are connected rather than isolated, the outputs should stay consistent with each other instead of

contradicting one another. The explainability module should help with engineer trust and actual adoption — [9] found exactly this with human-in-the-loop explanation. And the rule-based recommendation layer, even though it's not a complete solution, is at least a practical step toward prescriptive maintenance that doesn't require solving the whole problem first.

There's also a longer-term benefit we think is worth mentioning: the layered design is meant to grow. New sensor types can be added to the fusion layer without having to redesign diagnosis or prognosis, and new fault categories can be added to the recommendation layer's rule base as more real operational data comes in. So the system is meant to improve a bit at a time rather than needing a full rebuild every time something changes.

9 Evaluation strategy

To actually know whether this framework delivers on what we're claiming, we wouldn't want to rely on just one accuracy number — a proper evaluation needs to look at several things together. Diagnosis performance should be measured with precision, recall, and F1-score across fault classes, since in a maintenance setting, missing a critical fault (low recall) is usually worse than a few false alarms. Prognosis performance would come down to root-mean-square error (RMSE) between the predicted and actual Remaining Useful Life. To see how much the sensor fusion itself is actually contributing, we would compare the fused model's false-positive rate against single-sensor baselines trained on the same data — the same comparison approach used in [11] and [12]. And the explainability module shouldn't just be checked for technical correctness; it needs real feedback from maintenance engineers on whether the explanations actually made them trust a given alert more, which is the same human-in-the-loop evaluation approach that recent XAI reviews point to [9].

10 Limitations and challenges

We should be upfront that this framework isn't without its own problems. Fusing multiple sensors means more complexity in collecting and synchronizing data than a single-sensor setup would ever have. Edge hardware is cheaper than running everything in the cloud, but it's still an upfront cost, and for a very small operation that cost can still sting. Network latency and cybersecurity don't go away just because the architecture is smarter — if anything, more connected sensors usually means more exposure. Hooking this up to older, legacy machinery that was never built with sensor mounts or digital interfaces in mind can also be genuinely difficult in practice. And the rule-based recommendation layer from the section on prescriptive recommendations is really just a starting point — it is not a finished prescriptive-maintenance system, and its rule base would need real input from domain experts before it could be trusted in any specific deployment.

Data governance is worth a mention too. Continuous multi-sensor monitoring adds up to a lot of operational data over time, and any real deployment would need clear rules on how long raw sensor data sticks around, who gets to access it, and how it stays secure moving between the edge and the cloud — especially if more than one facility is sharing the same cloud setup.

There's also the question of model drift over the long run. As machines age or get repaired, what counts as “normal” in the sensor data can shift, and a model trained on older data will slowly start to drift away from reality. So a production version of this framework would need some kind of periodic retraining and revalidation built in — the model can't just be trained once and left alone, which does

add its own ongoing maintenance burden to a system that is, ironically, meant to reduce maintenance burden elsewhere.

11 Future scope

There are a few directions we think this could go next. The most obvious one is testing this pipeline on actual factory floor data instead of simulated or lab signals — that's the single biggest thing that would address the lab-versus-field gap from the research-gaps section. Adding 5G connectivity could also help cut down the latency between edge detection and cloud diagnosis even further. Further down the line, replacing the rule-based recommendation layer with a learned, data-driven prescriptive model — building on the recent prescriptive maintenance work in [6] — would push this framework closer to something that can actually support decisions automatically. And on the practical side, packaging up the edge-layer components into a low-cost, pre-configured hardware kit could make it a lot easier for small and medium-sized industries to actually adopt this instead of just reading about it.

12 Conclusion

The goal in this paper was to move beyond the review of the gaps in Industrial IoT-based predictive maintenance — which involved relying on lab data, setting simple limits on individual sensors, poor pipeline integration, lack of explainability, high deployment cost, and limited prescriptive capability — and actually develop something that addressed each one of those. We believe that this approach, by combining multiple sensor modalities, integrating detection, diagnosis and prognosis within a single pipeline (as opposed to three separate ones), incorporating an explainable AI layer and considering cost from the start and not at the end, takes it one more step beyond the review-only papers that fill the majority of this space. We haven't hidden the fact that there are real challenges regarding getting validated in a real factory and synchronizing data with all the old systems, and there are still challenges in this space. We do believe this provides a viable path to achieving greater accuracy, reliability, and usability of Industrial IoT for predictive maintenance on smaller players in the industrial space.

References

- [1] Meitz L, Senge J, Wagenhals T, Schöler T, Hähner J, Edinger J, et al. A literature review framework and open research challenges for predictive maintenance in industry 4.0. *Computers & Industrial Engineering*. 2025;206:111193. <https://doi.org/10.1016/j.cie.2025.111193>
- [2] Bitam T, Yahiaoui A, Boubiche DE, Martínez-Peláez R, Toral-Cruz H, Velarde-Alvarado P. Artificial intelligence of things for next-generation predictive maintenance. *Sensors*. 2025;25(24):7636. <https://doi.org/10.3390/s25247636>
- [3] Serradilla O, Zugasti E, Zurutuza U. Deep learning models for predictive maintenance: A survey, comparison, challenges and prospect. *arXiv:2010.03207 [Preprint]*. 2020.
- [4] Albattat SM, Albaker BM, Alsaedi MA. A systematic review of AI and IoT-powered smart maintenance methods in industrial application domains. *Engineering, Technology & Applied Science Research*. 2025;15(5):26890–26900. <https://doi.org/10.48084/etasr.12455>
- [5] Maintenance in Industry 4.0: A systematic review. 2025.
- [6] Orosnjak M, Saretzky F, Kedziora S. Prescriptive maintenance: A systematic literature review and exploratory meta-synthesis. *Applied Sciences*. 2025;15(15):8507. <https://doi.org/10.3390/app15158507>

-
- [7] Gawde S, Patil S, Kumar S, Kamat P, Kotecha K, Alfarhood S. Explainable predictive maintenance of rotating machines using LIME, SHAP, PDP, ICE. *IEEE Access*. 2024;12:29345–29361. <https://doi.org/10.1109/ACCESS.2024.3367110>
 - [8] Pashami S, Nowaczyk S, Fan Y, Jakubowski J, Paiva N, Davari N, et al. Explainable predictive maintenance. *arXiv:2306.05120* [Preprint]. 2023.
 - [9] Amaliah NR, Tjahjono B, Palade V. Human-in-the-loop XAI for predictive maintenance: A systematic review of interactive systems and their effectiveness in maintenance decision-making. *Electronics*. 2025;14(17):3384. <https://doi.org/10.3390/electronics14173384>
 - [10] Divya D, Arunkumar ON. An information processing theory framework for intelligent fault diagnosis and predictive maintenance. *Frontiers in Mechanical Engineering*. 2025;11:1724801. <https://doi.org/10.3389/fmech.2025.1724801>
 - [11] Gawde S, et al. An explainable predictive maintenance strategy for multi-fault diagnosis of rotating machines using multi-sensor data fusion. *Decision Analytics Journal*. 2024:100425. <https://doi.org/10.1016/j.dajour.2024.100425>
 - [12] Joseph. Predictive failure detection in network hardware using thermal imaging and deep learning with sensor fusion. In: *2025 3rd International Conference on Data Science and Network Security (ICDSNS)*; Tiptur, India; 2025. <https://doi.org/10.1109/ICDSNS65743.2025.11168660>