EAI Endorsed Transactions
on Digital Transformation of Industrial Processes Review Article EAI.EU

Machine Learning for Equipment Failure Prediction in Smart Manufacturing Plants: A Review

Kamil Musiał[1], Anna Burduk[1], Karolina Iwańczyk[2]

[1]Faculty of Mechanical Engineering, Wrocław University of Science and Technology, Wybrzeże Stanisława Wyspiańskiego 27, Wrocław 50-370, Poland
[2]Streamsoft D. Chojnacki i Wspólnicy Sp.j., al. Wojska Polskiego 11, 65-077 Zielona Góra, Poland

Abstract

INTRODUCTION: Smart manufacturing plants generate continuous streams of vibration, acoustic, thermal, electrical, process-control and quality data. These data make it possible to move from reactive or calendar-based maintenance toward predictive maintenance, but they also introduce problems of missing labels, non-stationary operating regimes, rare failures and unequal costs of false alarms and missed failures.
OBJECTIVES: This review analyses how machine learning is used to forecast equipment failures and remaining useful life in intelligent production environments, with emphasis on the suitability of methods for real industrial deployment rather than benchmark accuracy alone.
METHODS: A structured narrative review was conducted over established prognostics and health management literature, Industry 4.0 predictive-maintenance surveys, and representative empirical studies using public and industrial datasets such as C-MAPSS, PRONOSTIA, IMS bearings, Bosch Production Line Performance, semiconductor ion-implantation data and IoT-enabled production-line data.
RESULTS: The reviewed literature shows that feature-based classifiers, gradient boosting, support vector machines and random forests remain competitive when labels are scarce and process knowledge is available. Deep learning improves representation learning for multivariate sequences, especially for vibration and run-to-failure data, but requires careful validation against temporal leakage and domain shift. Hybrid approaches that combine signal processing, physics, digital twins and cost-aware decision rules are increasingly important for plant-level use.
CONCLUSION: Machine learning can substantially improve failure prediction in smart factories, yet its value depends on data governance, uncertainty handling, maintainability and integration with maintenance planning. The most credible systems treat prediction as a socio-technical decision process, not as a stand-alone model.

Keywords: predictive maintenance, machine learning, smart manufacturing, Industry 4.0, equipment failure prediction, remaining useful life, anomaly detection, industrial IoT

Received on 25 May 2026, accepted on 15 June 2026, published on 29 June 2026

doi: 10.4108/dtip.13164

*Corresponding Author. Email: kamil.musial@pwr.edu.pl

1. Introduction

Unplanned equipment failure remains one of the most disruptive events in production systems. A single critical stoppage can propagate through upstream and downstream stations, increase scrap, force emergency maintenance, and compromise delivery reliability. Traditional corrective maintenance reacts after the failure has already occurred. Preventive maintenance reduces some risk by replacing or inspecting components at fixed

intervals, but it often wastes useful component life and can still miss failures that emerge between scheduled interventions [1]. Predictive maintenance (PdM) changes the logic of maintenance by estimating the current and future health of assets from measured condition and process data [2,3]. In smart manufacturing plants, this shift is supported by cyber-physical systems, industrial IoT, manufacturing execution systems, edge computing and cloud analytics [4,5].

Machine learning (ML) has become the dominant analytical family in this context because failures rarely follow a single deterministic pattern. Machines operate under multiple loads, products and environmental conditions; sensors drift; maintenance actions reset degradation trajectories; and historical labels are often sparse or delayed. ML models can learn nonlinear relationships between high-dimensional sensor histories and later events, including fault classes, anomaly scores, time-to-failure and remaining useful life (RUL). The promise is not merely earlier detection. A useful PdM model should create an actionable maintenance window: early enough to order parts and schedule labour, but not so early that healthy assets are removed prematurely.

The literature now contains many studies reporting high classification accuracy or low RUL error on benchmark datasets. These results are valuable, but they do not automatically translate into reliable factory deployments. Production plants impose stricter requirements than laboratory benchmarks. A model must tolerate missing sensor channels, varying product recipes, non-stationary operating conditions, cybersecurity constraints and limited downtime for validation. It must also express uncertainty and support maintenance decisions with asymmetric costs. A false negative can lead to breakdown; a false positive can stop production unnecessarily. For this reason, this review treats equipment failure prediction as a complete industrial system, from data acquisition to maintenance action, rather than as an isolated modelling exercise.

The purpose of this article is to review the main machine learning approaches used for forecasting equipment failures in intelligent manufacturing plants and to analyse concrete examples from the literature. The review covers classical feature-based learning, anomaly detection, deep sequence models, transfer learning, hybrid physics-ML approaches, digital-twin integration and decision-oriented evaluation. Particular attention is paid to the conditions under which each method is credible for real industrial deployment, rather than to benchmark accuracy alone. The scope is limited to equipment failure prediction in manufacturing and manufacturing-adjacent cyber-physical systems; quality prediction is discussed only where it provides useful evidence about failure detection in production lines.

This review is structured around four research questions. First, which machine learning paradigms - supervised, unsupervised, deep, hybrid or physics-informed - are best suited to which industrial data conditions, and under what circumstances does each become credible for plant-level deployment? Second, what validation and evaluation protocols are necessary to distinguish genuine progress from benchmark overfitting, and how far does current literature conform to these protocols? Third, what are the principal barriers to translating ML-based failure prediction from laboratory benchmarks to operational smart manufacturing plants, including data quality, non-stationarity, label scarcity and organizational factors? Fourth, what research gaps remain open, and what directions are most likely to close the distance between academic modelling and reliable industrial maintenance practice?

2. Review scope and analytical approach

This paper is a review article with a structured narrative design. It does not claim to be a full systematic review with database-wide screening and statistical meta-analysis. Instead, it synthesizes established foundations in condition-based maintenance, recent Industry 4.0 reviews [6-9], and highly cited empirical examples that have shaped ML practice in PdM. The scope is limited to equipment failure prediction in manufacturing or manufacturing-adjacent cyber-physical systems. Quality prediction is discussed only where it provides useful evidence about failure detection in production lines, because the boundary between process fault, product defect and equipment degradation is often blurred in smart factories.

The literature was interpreted through four questions. First, what prediction task is being solved: fault diagnosis, early warning, anomaly detection, RUL regression, maintenance-window classification or cost-based decision support? Second, what data are available: run-to-failure histories, condition-monitoring streams, machine logs, process parameters, maintenance records or quality outcomes? Third, what modelling assumption is made: supervised learning with labelled failures, unsupervised normal-behaviour modelling, self-supervised representation learning, transfer learning across machines, or hybrid physical-statistical inference? Fourth, how close is the evaluation to plant operation: random cross-validation, temporally blocked validation, machine-level holdout, cross-domain transfer, online detection delay or economic cost?

This framing is important because the same algorithm can be appropriate or inappropriate depending on the industrial setting. A random forest trained on engineered vibration features may outperform a deep neural network when there are only a few labelled failures but strong domain knowledge. Conversely, a convolutional or recurrent network may be preferable for high-frequency signals where hand-crafted features omit relevant degradation information. A one-class model may be the only feasible choice for new assets with no failure examples. The goal of the review is therefore not to rank algorithms universally. It is to explain the conditions under which each method becomes credible for failure forecasting in smart manufacturing.

3. Predictive-maintenance data in smart manufacturing plants

Smart factories provide several kinds of data for PdM. The most familiar are condition-monitoring signals, such as vibration, acoustic emission, temperature, current, pressure, oil quality and motor torque. Vibration is especially prominent in bearing, gearbox, spindle and rotating-machine studies because local defects often produce changes in frequency components before functional failure. Electrical current and power signatures are useful for motors, pumps and drives. Process-control variables, including feed rate, spindle speed, pressure set points and cycle times, describe the operating regime. Maintenance logs and work orders identify interventions, but they are frequently noisy because technicians document symptoms, actions and component replacements in inconsistent language.

The prediction target is not always a literal failure event. In many plants, catastrophic failure is rare because maintenance teams intervene before breakdown. The historical data therefore contain censored observations: the component was replaced, but the true failure time is unknown. Susto et al. explicitly addressed high-dimensional and censored maintenance data by using multiple classifiers trained for different prediction horizons and connecting their outputs to operating-cost decisions [10]. This perspective remains highly relevant. Many industrial datasets are not clean sequences ending in failure. They are interrupted by inspections, preventive replacements, production changes and shutdowns.

Data quality is a recurring challenge. Missing values may arise from sensor faults, communication errors or product paths that bypass certain stations. The Bosch Production Line Performance dataset illustrates this problem at scale. It includes measurements, categorical variables and timestamp-like features collected as parts move through multiple stations; published analyses emphasize sparsity, missingness, high dimensionality, multiple product flow paths and rare positive labels [11]. These properties resemble real manufacturing more closely than many run-to-failure experiments. They also explain why simple accuracy is misleading. In rare-event settings, a trivial model can appear accurate by predicting no failures. Metrics such as Matthews correlation coefficient, precision-recall area, recall at fixed false-alarm rate and cost-weighted utility are more informative.

Another difficulty is non-stationarity. A model trained during one production campaign may degrade when product mix, tooling, operators, suppliers or environmental conditions change. Industrial datasets may contain concept drift in both the input distribution and the failure mechanism. If the data are randomly split, observations from the same machine, product family or time period may appear in both training and test sets, producing optimistic results. Stronger validation uses machine-level holdout, time-based holdout, cross-load testing or leave-one-domain-out evaluation. These protocols are essential for smart plants because PdM is often deployed on assets that are similar but not identical to those in the training set.

4. Main machine-learning paradigms for failure prediction

ML methods for PdM can be grouped by the kind of supervision they require and the representation they learn. Supervised learning maps feature vectors or sequences to labelled outcomes. Labels may be fault categories, failure within a future horizon, or RUL values. Unsupervised and semi-supervised learning model normal behaviour and detect deviations, which is attractive when failures are rare. Deep learning learns representations directly from raw or lightly processed sequences. Hybrid methods combine ML with physics, signal processing, reliability models or expert rules. In practice, successful plant systems often blend these categories rather than adopting a pure approach.

Table 1. Prediction tasks and typical machine-learning treatment in smart manufacturing PdM

Prediction task	Typical ML treatment
Failure within horizon	Binary or multi-horizon classifiers; SVM, random forest, gradient boosting, calibrated logistic models.
Fault diagnosis	Multi-class classification using signal features, CNNs, residual analysis and expert-labelled fault modes.
Anomaly detection	One-class SVM, isolation forest, PCA, autoencoders and LSTM reconstruction of normal behaviour.
RUL estimation	Regression, sequence-to-one deep learning, degradation models, quantile regression and probabilistic intervals.
Maintenance action	Cost-aware decision rules, scheduling optimization and human review of risk, time window and confidence.

4.1. Feature-based supervised learning

Feature-based learning remains central to industrial PdM. Engineers extract time-domain, frequency-domain and time-frequency features from signals: root mean square, kurtosis, crest factor, spectral energy, envelope components, wavelet coefficients and trend statistics. These features are then used by logistic regression, decision trees, random forests, gradient boosting, support vector machines (SVM), k-nearest neighbours or shallow neural networks. The advantage is transparency and data efficiency. A maintenance engineer can relate rising vibration RMS or a bearing envelope frequency to a

plausible physical mechanism. Models can be trained with fewer labelled failures than deep architectures require.

Susto et al. provide a clear example of decision-oriented supervised learning in a semiconductor manufacturing maintenance task [10]. Rather than training one classifier to predict a generic failure label, they trained multiple classifiers over different prediction horizons. The model therefore represented a trade-off between unexpected breakdowns and unused lifetime. This design is closer to maintenance reality than a single binary alarm. A warning 24 hours ahead and a warning one week ahead have different operational value. The study also showed how high-dimensional manufacturing data and censoring can be handled through a structured ML methodology, and its strongest lesson is that PdM should be optimized against maintenance cost, not only classification accuracy.

Tree ensembles and gradient boosting are especially useful for tabular manufacturing data with missing values, nonlinear interactions and mixed variable types. In the Bosch production-line case, sparse high-dimensional records and rare failure labels made model design difficult. Published work on the challenge used feature engineering, online learning and XGBoost-like approaches to identify parts likely to fail and to exploit information about stations, lines, dates and product flow paths [11]. Although the Bosch case concerns internal product failures rather than a single asset's mechanical failure, it is instructive for equipment forecasting because it exposes typical smart-factory data problems: large-scale records, routing heterogeneity, sparse station measurements and extreme imbalance.

4.2. Anomaly detection and one-class learning

In many plants, there are abundant examples of normal operation and very few labelled failures. Anomaly detection therefore becomes a practical alternative. One-class SVMs, isolation forests, Gaussian mixture models, principal component analysis, autoencoders and sequence-to-sequence models are trained on normal data and then identify deviations. The approach is appropriate for early deployment on assets where maintenance teams have not yet accumulated many breakdown histories. Its weakness is interpretability: not every statistical anomaly is a maintenance-relevant fault. Changes in product mix, speed, tooling or ambient temperature can also create unusual patterns.

Malhotra et al. introduced an LSTM encoder-decoder for multivariate time-series anomaly detection, learning to reconstruct normal time-series behaviour and using reconstruction error to flag deviations [12]. Although the study used several benchmark datasets rather than a single manufacturing plant, the method influenced PdM because multivariate sensor streams in factories share the same temporal structure. LSTM autoencoders and related models can learn temporal dependencies that static thresholds miss. They are particularly useful when failure evolves as a gradual departure from normal dynamics. However, the thresholding problem remains. A reconstruction error must be converted into an alarm policy, and that policy should be calibrated by expected downtime cost and acceptable false-alarm frequency. Anomaly detection also requires careful normal-data selection. If the training set contains early degradation, the model may learn abnormal behaviour as normal. If the training set is too narrow, normal changes in operating regime may be misclassified. Smart plants can reduce this risk by conditioning models on operational context, using regime clustering before anomaly detection, and combining anomaly scores with rules derived from process knowledge. For example, a vibration increase under a heavy load may be less alarming than the same increase under a stable light load. Context-aware anomaly detection is therefore more credible than context-free global thresholds.

4.3. Deep learning for sequential and signal data

Deep learning has reshaped PdM research because it can learn hierarchical representations from raw signals and multivariate histories. Convolutional neural networks (CNNs) capture local temporal or spectral patterns; recurrent networks such as LSTM and GRU capture temporal dependencies; temporal convolutional networks process long sequences with stable gradients; transformers model long-range dependencies using attention; and autoencoders or variational autoencoders learn compact health representations [13]. Surveys by Zhao et al., Khan and Yairi, and Wu et al. document the shift from shallow feature-based models to deep architectures in machine health monitoring and RUL prediction [14-17].

The NASA C-MAPSS turbofan degradation dataset is the most influential RUL benchmark in this area. The dataset contains simulated run-to-failure histories generated with the Commercial Modular Aero-Propulsion System Simulation, with multiple operating conditions and fault modes [18]. The task is to infer RUL from multivariate sensor trajectories. Li et al. used a deep convolutional neural network for RUL estimation and showed that CNNs can learn useful degradation representations from sensor windows [19]. Ellefsen et al. examined semi-supervised deep architectures on C-MAPSS, addressing the fact that unlabelled or partially labelled data are common in prognostics [20]. These studies demonstrate why deep models are attractive: they reduce dependence on manually selected features and can exploit temporal sensor patterns that are hard to encode explicitly.

Still, C-MAPSS also exposes the limits of benchmark-driven progress. It is simulated, the sensors are clean relative to many plant environments, and RUL labels follow a known construction. A model that performs well on C-MAPSS may not generalize to a CNC spindle, a packaging line or a compressor in a different plant. Good practice therefore requires cross-dataset evaluation,

uncertainty estimation and ablation analysis. The model should also be compared with strong feature-based baselines. Deep learning is not automatically superior; it becomes valuable when the data volume, signal complexity and maintenance objective justify the added complexity.

A more recent direction challenges the assumption that PdM models must be trained from scratch on domain-specific data. Foundation models - large neural networks pre-trained on broad corpora and subsequently fine-tuned for narrow tasks - have begun to appear in fault diagnosis research. Zheng et al. conducted a systematic empirical study of fine-tuning pre-trained large language models (LLMs) for fault diagnosis of complex systems, converting sensor readings and system-state descriptions into text prompts and fine-tuning several base LLMs on both simulated and real industrial datasets [21]. Their results show that fine-tuned LLMs can reach competitive fault classification accuracy without bespoke feature engineering, and that the gains are most pronounced when labelled examples are scarce - precisely the condition that dominates industrial maintenance settings. A complementary approach adapts multimodal LLMs to integrate heterogeneous industrial inputs: sensor time series, maintenance work orders, process recipes and technician notes can in principle be encoded in a shared representation, enabling contextual reasoning that purely numerical models cannot perform [22]. For smart manufacturing, this is relevant because failure events are often documented in unstructured text - technician logs, alarm descriptions, part-replacement records - that conventional ML pipelines discard. LLM-based diagnostics can potentially bridge the gap between structured sensor data and the natural-language knowledge that experienced maintenance engineers accumulate over years of plant operation. However, the approach is not without caveats for industrial deployment. LLMs require careful calibration to produce reliable uncertainty estimates; their inference cost is non-trivial for edge devices; and their outputs must be validated against temporal leakage and domain shift before being trusted in production environments. Research on lightweight, domain-adapted foundation models for PdM therefore represents an important open direction, but one that currently has more demonstrated potential than proven plant-level deployments.

4.4. Critical assessment of LLM-based predictive maintenance

LLM-based PdM should be interpreted as a promising but immature extension of predictive maintenance rather than as a replacement for signal-based prognostics. Its main advantage is the ability to combine heterogeneous information sources, especially technician notes, alarms, work orders, process recipes and maintenance histories that are difficult to encode in conventional numerical pipelines [21,22]. However, equipment degradation is not primarily a linguistic phenomenon. High-frequency vibration, acoustic emission, motor-current signatures and thermal transients may lose diagnostic information when compressed into textual prompts or symbolic summaries. The apparent performance of an LLM can also depend strongly on prompt design, data formatting and the way sensor histories are discretized into text. This makes reproducibility and comparison with classical PdM models difficult.

A second limitation concerns reliability. LLMs may generate plausible explanations that are not causally grounded in the physical asset, and this is problematic when the output is used to justify inspection, load reduction or replacement. In PdM, an incorrect explanation can be as damaging as an incorrect label because it may direct technicians toward the wrong component or failure mode. LLM outputs should therefore be treated as decision-support evidence, not as autonomous maintenance commands. For deployment, every LLM-based PdM study should report calibration, uncertainty, temporal leakage controls, out-of-domain testing and failure-mode coverage. The model should also be benchmarked against strong non-LLM baselines such as gradient boosting, calibrated logistic models, random forests, CNN/LSTM models and hybrid physics-informed approaches.

Deployment barriers are substantial. Large models introduce latency, memory and energy costs that may conflict with edge deployment and low-latency alarm handling. Fine-tuned models may not generalize across plants because sensor placement, sampling rates, asset age, recipe mix and maintenance vocabulary differ between factories. Cybersecurity and intellectual-property concerns are also stronger than in ordinary numerical PdM because prompts may contain production recipes, process settings, alarm histories and proprietary technician knowledge. In addition, model updates, prompt changes and retrieval-augmented pipelines require version control and audit trails; otherwise the same maintenance query may produce different recommendations over time. For these reasons, the most credible near-term use of LLMs in smart manufacturing is not fully autonomous failure prediction, but retrieval of similar historical cases, summarization of maintenance logs, explanation of model outputs and human-in-the-loop diagnostic support.

4.5. Hybrid, physics-informed and digital-twin approaches

Purely data-driven ML is limited when failures are rare, operating conditions change, or the cost of wrong predictions is high. Hybrid approaches combine data-driven learning with physics, reliability theory or expert knowledge. A model may use vibration physics to define features, reliability models to constrain RUL trajectories, or a digital twin to generate simulated degradation scenarios. Digital twins are especially relevant to smart manufacturing because they connect virtual representations of assets with live sensor data, process plans and maintenance history [23,24].

Physics-informed PdM can reduce spurious correlations. For example, a model may learn that a specific production recipe precedes failure, when the true cause is a load profile that increases bearing stress. Embedding load, speed, temperature and known degradation mechanisms helps separate causal degradation from incidental correlation. Hybrid systems can also support extrapolation beyond observed failures. A neural network trained only on historical plant data may be unreliable under a new operating envelope; a hybrid model can use physical constraints to prevent impossible health trajectories.

The current direction in the literature is not a replacement of ML by physics or vice versa, but a layered architecture. Signal processing creates robust health indicators; ML estimates risk or RUL; uncertainty models quantify confidence; maintenance optimization converts predictions into action; and human operators review the recommendation. This architecture is consistent with recent reviews that emphasize integration, multidisciplinarity and the limitations of one-size-fits-all PdM models [8,25,26].

4.6. Comparative Assessment of ML Methods

Table 2. Comparative summary of major ML approaches for equipment failure prediction

Approach	Data Requirements	Interpretability	Computational Complexity	Deployment Readiness	Typical Applications
Linear/Statistical Models	Low-moderate; requires clean, labeled numeric time-series; limited feature richness.	**High**; coefficients or simple structure allow direct interpretation of feature effects.	**Low**; fast training/inference even on modest hardware.	**High**; well-established, lightweight, easily deployed on edge devices.	Trend forecasting, simple RUL regression, basic anomaly detection.
Decision Trees / Random Forests	Moderate; handles mixed data types and some missing values; robust with moderate samples.	**High (Tree)** / **Moderate (Ensemble)**; individual trees yield clear rules, ensembles require interpretation tools (feature importance).	**Low (Tree)** / **Moderate (Ensemble)**; single-tree very fast; RF scales with number of trees/features.	**High**; widely used in industry with optimized libraries; easy to integrate into real- time systems.	Fault classification, threshold-based alerts, multivariate sensor data interpretation.
Support Vector Machines (SVM)	Moderate- high; effective in high-dimensional settings but requires careful feature scaling and kernel selection.	**Moderate (Linear SVM)** / **Low (Kernel SVM)**; linear SVMs are interpretable by weights, nonlinear kernels act as black boxes.	**Moderate- High**; QP training can be slow for large datasets; prediction speed is moderate.	**Medium**; supported by many tools, but sensitivity to hyperparameters and scaling can complicate deployment.	Vibration analysis, anomaly detection, small- to- medium feature classification tasks.
Gradient Boosting (XGBoost, LightGBM)	Moderate- high; benefits from ample labeled data and feature engineering; robust to moderate noise.	**Low**; ensemble of many trees yields high accuracy but complex decision boundaries; global feature importance is available but limited.	**Moderate- High**; iterative boosting is computationally heavier than single- tree; training can be parallelized.	**Medium**; popular in analytics (e.g. Kaggle), but requires tuning and more resources than simpler models.	RUL estimation, classification/regression on historical sensor/tabular data.
Neural Networks / Deep Learning	**High**; requires large volumes of labeled data (often hundreds to thousands of	**Low**; complex architectures make decisions opaque. Some interpretability	**High**; many parameters and layers require GPUs/TPUs for training; inference	**Low- Medium**; cutting- edge and powerful but demand specialized	Multi-sensor fusion, complex time-series RUL prediction, image-based fault detection,

	failure examples); can handle raw high-dimensional inputs (images, waveforms, IoT streams).	via saliency maps or attention.	can also be resource-intensive.	hardware; embedded deployments are challenging.	high-dimensional anomaly detection.
Bayesian Approaches (GP, BNN)	Low- Moderate; can work with smaller datasets by leveraging priors; GPs are best with small N (<10k) data points.	**Medium**; model outputs are probabilistic (uncertainty intervals), offering interpretability of confidence; model internals can be complex.	**High**; Gaussian processes scale cubically with data; Bayesian NNs require sampling/inference (e.g. MCMC, variational).	**Medium**; increasing interest for uncertainty modeling in PdM, but rarely standard in legacy systems; specialized libraries required.	Small- data RUL estimation, uncertainty quantification in predictions, components with strong prior models or physics.
Foundation Models / LLM-based Methods	**Very High**; require massive pretraining data (e.g. text corpora, technical manuals) and domain-specific fine-tuning data; potentially leverage unsupervised sensor corpora.	**Very Low**; essentially black-box; current XAI for transformers is limited; feature/attention analysis partially possible but not straightforward.	**Very High**; billions of parameters; inference and especially training demand state-of-the-art hardware and considerable latency.	**Low**; emergent research; industrial pilots exist (e.g. text analytics for maintenance logs) but full-scale real-time deployment is nascent due to resource and reliability concerns.	Text-based tasks: document summarization, fault diagnosis Q&A from manuals; possible advisory agents (e.g. multilingual log analysis); less suitable for raw sensor RUL unless combined with other methods.

The table summarizes key trade- offs among representative ML methods for PdM. Linear models (including basic statistical regressions) require relatively small amounts of structured, clean numeric time-series data and yield easily understandable parameters, making them attractive for straightforward trend modeling. They are computationally light and maturely deployed, even on edge devices.

Decision trees offer rule-based interpretability, which remains true for small trees; however, ensembles such as Random Forests improve accuracy at the cost of interpretability, though they still afford global feature-importance metrics. Trees scale very efficiently, while Random Forests scale moderately with the number of trees and features; both are widely used in industry due to optimized libraries and ease of real-time integration.

Support Vector Machines can handle complex boundaries with kernels, but non-linear kernels make them harder to explain than linear SVMs, which remain interpretable via their weights. Training can become expensive for large datasets (quadratic-programming complexity), prediction speed is moderate, and sensitivity to hyperparameters and feature scaling can complicate deployment despite broad tool support.

Gradient boosting (e.g., XGBoost, LightGBM) often achieves top predictive performance on tabular PdM data, but its ensemble nature makes it less transparent, offering only limited global feature importance. It also requires more computation than single-tree methods and careful hyperparameter tuning, which together with its resource demands place its deployment readiness at a medium level despite its popularity in analytics.

Neural networks (NNs) and deep learning excel at capturing high- dimensional sensor patterns and fused multi- modal data, but demand very large labeled datasets (often hundreds to thousands of failure examples) and extensive GPU/TPU compute; they are treated as black boxes in practice, with only partial interpretability via saliency maps or attention. Consequently, their deployment readiness is low to medium, as specialized hardware needs make embedded deployment challenging.

In contrast, Bayesian methods (e.g., Gaussian processes, Bayesian NNs) inherently quantify uncertainty and can perform well with smaller datasets (Gaussian processes especially below roughly 10,000 points) by incorporating prior knowledge. They yield probabilistic outputs useful for risk estimation and offer some interpretability through confidence intervals, but computational complexity grows sharply with dataset size - Gaussian processes scale cubically - making them costly to train as data volume increases, and their deployment remains medium, limited to specialized libraries rather than legacy systems.

Finally, foundation models and LLM-based approaches represent a new frontier: they require enormous amounts of pretraining data (text corpora, technical manuals, and potentially unsupervised sensor corpora) plus domain-specific fine- tuning data, and produce highly context-aware outputs, yet they are extremely expensive to train

and operate, with billions of parameters demanding state-of-the-art hardware. Their decision logic is effectively opaque, current explainability methods for transformers remain limited, and their deployment readiness is low-industrial pilots exist (e.g., text analytics for maintenance logs), but full- scale real-time deployment remains nascent. They are less suitable for raw sensor- based RUL prediction unless combined with other methods.

Overall, in industrial contexts practitioners must balance these factors: simple interpretable models may suffice when physics or heuristics are well-known, whereas complex deep models may be justified for very large-scale IoT systems with abundant data. The reviewed evidence suggests a pragmatic approach: use the simplest model that meets accuracy needs, to ensure real-time viability, human trust, and deployment feasibility. This table and discussion help guide such model choice by highlighting each method's practical data needs, interpretability, computational complexity, and deployment readiness.

5. Representative literature examples and lessons

The following examples illustrate how different datasets and industrial contexts shape the modelling problem. They are not interchangeable benchmarks. Each represents a different combination of asset type, data quality, failure definition and deployment relevance.

Table 3. Representative literature examples and transferable lessons

Literature example	Main lesson for smart manufacturing PdM
C-MAPSS turbofan data	Useful for RUL modelling and operating-regime handling, but simulated benchmarks do not prove factory readiness.
PRONOSTIA / IMS bearings	Shows the value of vibration features and health indicators; physical interpretability supports technician trust.
Bosch production line	Highlights sparse high-dimensional tabular data, rare positives, routing context and imbalance-aware metrics.
Semiconductor ion implanter	Demonstrates horizon-specific classifiers and cost-aware maintenance decisions under censoring.
IoT production-line RUL	Shows that real deployment is a pipeline of filtering, regime identification, forecasting and model operations.

5.1. C-MAPSS turbofan degradation

The C-MAPSS dataset is a controlled prognostic benchmark. It provides complete run-to-failure sequences in the training set and truncated test sequences for which RUL must be predicted [18]. This structure is attractive for supervised and deep RUL models because the target is explicit. Studies using CNN, LSTM, semi-supervised architectures and attention mechanisms have demonstrated progressively richer ways to encode multivariate degradation patterns [13,19,20,27]. The practical lesson for smart manufacturing is that run-to-failure data are extremely valuable but rarely available at scale. Production plants often prevent failure before it occurs. Therefore, C-MAPSS should be used to study modelling principles, not to claim direct evidence of factory readiness.

A second lesson is that operating conditions matter. C-MAPSS includes datasets with different combinations of conditions and fault modes. This resembles plant reality, where a motor, robot or spindle may operate under different products, speeds and loads. Models that ignore regime variation can confuse normal load-related changes with degradation. Regime normalization, clustering by operating mode, or inclusion of context variables improves credibility. In plant deployment, this means that PdM data models should be connected to production recipes and control states, not only raw sensor streams.

5.2. PRONOSTIA and IMS bearing data

Bearing prognosis is one of the most studied PdM topics because bearings are common, failure-prone and safety-critical in rotating equipment. PRONOSTIA is an experimental platform for accelerated bearing degradation tests and was used in the IEEE PHM 2012 data challenge [10]. IMS bearing data from the University of Cincinnati have also been widely used for vibration-based diagnosis and prognosis [29]. These datasets support studies on health-indicator construction, feature monotonicity, early fault detection and RUL estimation. They are closer to mechanical equipment than C-MAPSS, but they are still controlled experiments.

The main methodological lesson is that signal processing and ML are complementary. For bearing faults, envelope analysis, spectral bands, wavelets and statistical features often capture physically meaningful degradation. Deep networks can learn directly from vibration windows, but their outputs are more convincing when compared with known fault frequencies or trends in energy and kurtosis. In a smart factory, a bearing model should not only output a class or RUL value; it should provide enough diagnostic evidence for a maintenance technician to trust the recommendation. This is one reason explainability is more than a regulatory concern. It is part of operational adoption.

5.3. Bosch production-line data

The Bosch Production Line Performance dataset represents a different kind of smart-manufacturing problem. It records anonymized numerical, categorical and date features as components move through production lines and stations, with the goal of predicting internal failures [11]. Published analysis reports more than one million training samples, thousands of sparse features, rare positive labels, multiple stations and heterogeneous product flow paths. This example is important because many factory ML problems are tabular, sparse and imbalanced rather than clean high-frequency run-to-failure sequences.

For equipment failure prediction, the Bosch case demonstrates the need to model production context. A component's route, the time spent in stations and the station-specific measurements can be more predictive than a single universal feature set. The same is true for maintenance: a robot or machine tool may show different health signals depending on product family and cycle. The case also shows why rare-event metrics matter. When less than one percent of observations are positive, accuracy is not meaningful. The Matthews correlation coefficient, precision at operational recall, and expected savings from targeted inspection provide more relevant information.

5.4. Semiconductor ion-implantation maintenance

The semiconductor ion-implanter case in Susto et al. is one of the clearest examples of moving from prediction to maintenance decision [28]. Semiconductor tools generate high-dimensional process and equipment data, while maintenance events can be censored by scheduled interventions. The authors trained multiple classifiers with different look-ahead horizons and used the outputs to support dynamic maintenance decisions. This design recognizes that maintenance value depends on timing. A model that merely says 'failure soon' is less useful than one that estimates risk across several future windows.

The broader lesson is that smart plants need prediction horizons aligned with maintenance operations. Short-horizon alarms may help avoid immediate breakdown but can create production disruption. Long-horizon predictions help spare-part logistics and production planning but have greater uncertainty. A mature PdM system should expose this trade-off. It should also support policies such as 'inspect at the next planned stop', 'reduce load and monitor', or 'replace during the next weekend shutdown'. Machine learning becomes valuable when it improves these decisions, not when it only improves a validation score.

5.5. Real production-line RUL prediction from IoT data

Recent work by Tasci et al. is notable because it addresses RUL prediction for manufacturing production lines using real-world IoT sensor data from assembly lines [30]. The study combines filtering, clustering and forecasting elements to predict RUL before production-line stoppage. This is closer to the smart-factory use case than many laboratory datasets because the model must operate with data generated by actual industrial processes. It also supports an important conclusion: practical PdM is usually a pipeline, not a single algorithm. Data cleaning, regime identification and forecasting all influence the final prediction.

Such production-line studies also raise deployment questions. A plant-level PdM model must be maintained like any other industrial software system. It requires version control, monitoring of input drift, logging of predictions and maintenance outcomes, periodic retraining and procedures for rollback. Without these elements, a model may be accurate during pilot testing but deteriorate silently after product or process changes. This operational discipline is still underreported in the academic literature, where evaluation often ends at offline performance metrics.

6. Model evaluation beyond benchmark accuracy

Evaluation is one of the weakest points in PdM research. Many studies report accuracy, F1 score, root mean squared error or mean absolute error. These metrics are useful but incomplete. A manufacturing plant needs to know whether the prediction arrives early enough, whether the false-alarm rate is acceptable, how uncertainty should change the maintenance action, and whether the model remains valid under new production conditions. For RUL models, an error of 20 hours has different implications when the true RUL is 30 hours than when it is 1000 hours. For classification models, precision and recall must be interpreted against maintenance capacity and stoppage cost.

Temporal validation is essential. If a model is trained and tested using random samples from the same machine history, the test set may include patterns that are almost duplicated in training. This produces an optimistic estimate. Stronger protocols hold out entire machines, later time periods, product families or operating regimes. For smart manufacturing, a recommended minimum is a time-ordered split plus a machine- or line-level holdout when enough data are available. For transfer studies, the source and target domains should be explicit. A model trained on one line and tested on another should not be described as general unless the domains differ in a meaningful way.

Uncertainty quantification and calibration are central to operational PdM because maintenance decisions are rarely triggered by a deterministic label alone. Three types of uncertainty should be distinguished. Aleatoric uncertainty arises from irreducible noise in sensor readings, operating conditions and degradation trajectories. Epistemic uncertainty reflects limited training data, sparse failures and model uncertainty, and should increase for unseen machines, new recipes or shifted operating regimes. Data and label uncertainty is also common in PdM because

maintenance logs are noisy, failures are censored by preventive replacement, and the exact onset of degradation is often unknown. Treating these sources as equivalent can lead to inappropriate maintenance actions: high aleatoric uncertainty may justify wider inspection windows, while high epistemic uncertainty should usually trigger model review, additional data collection or human confirmation.

Several families of methods are available. For classification tasks, calibrated logistic models, Platt scaling, isotonic regression and temperature scaling can be used to align predicted probabilities with empirical failure frequencies [31]. For RUL regression, quantile regression, probabilistic degradation models and heteroscedastic neural networks can provide prediction intervals rather than single-point estimates. Bayesian neural networks and Monte Carlo dropout approximate uncertainty over model parameters [32], while deep ensembles provide a practical and scalable alternative that often performs well under distribution shift [33]. Conformal prediction is particularly attractive for industrial PdM because it can wrap around an existing model and produce prediction sets or intervals with user-specified coverage under explicit assumptions [34]. However, conformal methods must be applied carefully to temporally dependent data; calibration sets should respect time order, asset identity and operating regime rather than relying on random exchangeable samples.

Calibration should therefore be evaluated explicitly, not inferred from accuracy. Classification studies should report reliability diagrams, expected calibration error, Brier score and negative log-likelihood in addition to precision, recall and F1 score. RUL and risk-forecasting studies should report empirical interval coverage, interval width, continuous ranked probability score or interval score, because useful intervals must be both well calibrated and sufficiently sharp [35]. In deployment, calibration should be monitored over time because sensor drift, recipe changes and maintenance-policy changes can make a once-calibrated model overconfident. A practical PdM dashboard should therefore expose not only the predicted fault class or RUL value, but also the confidence interval, calibration status and drift warning associated with the prediction.

The decision rule should be uncertainty-aware. A prediction of 100 hours of RUL with a 30–250 hour interval should not trigger the same action as a prediction of 100 hours with a 90–110 hour interval. Wide intervals may justify intensified monitoring, manual inspection or load reduction, whereas narrow intervals around a low RUL value can justify planned replacement. In this sense, uncertainty is not an optional statistical add-on; it is the mechanism that connects predictive modelling with risk-sensitive maintenance planning.

Economic evaluation should be incorporated wherever possible. Susto et al. used cost-oriented decision logic, and this remains a good model for the field [10]. A PdM system should be assessed by expected downtime reduction, avoided emergency repair, spare-part utilization, unexploited component life and production disruption. In some plants, a conservative model with moderate accuracy may deliver more value than a high-accuracy model with late warnings. The appropriate metric depends on the production system. Batch manufacturing, continuous process industries and high-mix discrete assembly have different maintenance economics.

6.1. Explainability and trustworthy AI in maintenance decisions

Given the safety-critical nature of industrial maintenance decisions, model interpretability and trustworthiness are essential complements to the calibration and uncertainty-quantification practices discussed above. Explainable AI (XAI) techniques enable engineers and technicians to understand black-box predictions and foster operator trust in automated recommendations. In PdM, both global and local explainability methods are relevant. Global explanations, such as feature-importance scores and partial dependence plots, reveal how input features generally affect model outputs across the dataset, helping engineers verify overall model behaviour. Local explanations, such as LIME, SHAP, Integrated Gradients and counterfactual explanations, explain why a specific prediction was made for a given instance. SHAP (Shapley Additive Explanations) computes each feature's contribution to a particular prediction [36], while LIME (Local Interpretable Model-agnostic Explanations) fits an interpretable model around one data point [37]. Integrated Gradients, applicable to neural networks, attribute importance by integrating gradients along input paths. Counterfactual explanations identify minimal changes to input values that would alter the model's prediction, directly suggesting actionable interventions, for example that a lower bearing temperature would have changed the predicted outcome. Attention-based methods, in models that use attention layers, can similarly highlight which inputs were most influential for a given decision.

These XAI tools have practical limitations. They may produce spurious or unstable explanations when models capture complex feature interactions, and they typically require careful interpretation by domain experts. Counterfactual explanations in particular depend on selecting meaningful candidate features and may suggest implausible or operationally infeasible changes if not properly constrained. The literature also highlights the need for explanations tailored to different stakeholders: a data scientist may require detailed attribution data, whereas a maintenance technician needs concise, actionable operational rules rather than raw feature-contribution scores.

Beyond interpretability, broader trustworthy-AI dimensions are crucial in maintenance contexts. Frameworks such as the NIST AI Risk Management Framework identify validity and reliability, safety, security and resilience, privacy, explainability, transparency, accountability and fairness as core attributes of trustworthy

AI systems [41]. In practice, this means that maintenance AI must be rigorously tested for robustness, including resilience to sensor noise and adversarial perturbations, accompanied by transparent documentation of how models are built and what data they use, and supported by audit trails that record how predictions translate into maintenance actions. Fairness is generally less visible in purely technical fault-prediction tasks, but practitioners must still guard against systematic biases, such as models that under-serve assets in certain plant locations or production lines because of uneven historical data coverage. Safety remains paramount: automated recommendations must never introduce hazardous operating states, and ML predictions should remain subordinate to certified protection and control systems, as noted in Section 7. Human oversight is equally essential, with operators able to contest or override AI-generated suggestions and to feed corrections back into the system for future model improvement.

Taken together, XAI methods such as SHAP and LIME help satisfy transparency requirements by explaining individual decisions, but maintaining a robust and safe PdM system also depends on ongoing validation, fail-safe design and governance frameworks that connect explanations to maintenance practice. Explanations alone do not guarantee trustworthiness; they become valuable only when domain experts use them to audit predictions, detect failure modes, and adjust models so that the system remains aligned with operational constraints and regulatory requirements. These explainability requirements are particularly pressing for LLM-based diagnostic approaches (Section 4.4), where the reasoning behind a generated recommendation is inherently difficult to trace back to specific sensor signals, and where XAI methods may need to be adapted to operate on text-based or multimodal model inputs.

7. Implementation architecture in intelligent production plants

A deployable ML-based PdM system requires more than a trained model [38]. The architecture begins with sensor selection and data acquisition. Sensors must capture degradation-relevant signals at sufficient frequency and reliability. Edge devices can preprocess high-frequency vibration or acoustic signals to reduce bandwidth and latency. Plant historians, programmable logic controllers, SCADA systems and manufacturing execution systems provide operational context. Maintenance management systems supply work-order and replacement history. The central challenge is to align these sources in time and asset identity.

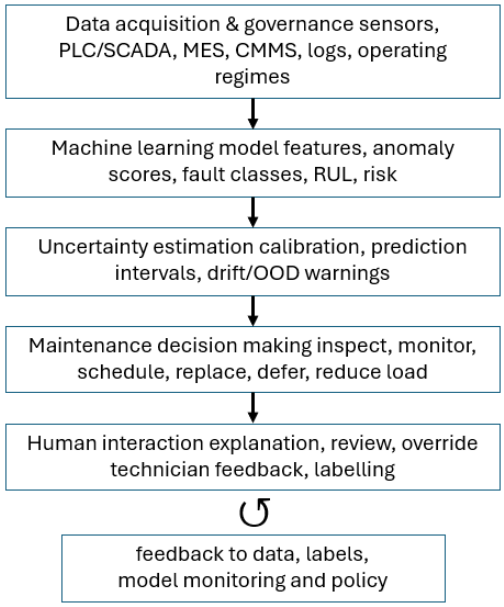

Figure 1. Conceptual framework for ML-based predictive maintenance in smart manufacturing. The framework links data acquisition, machine-learning inference, uncertainty estimation, maintenance decision making and human interaction in a closed feedback loop.

Figure 1 summarizes the proposed data-to-decision framework. Data acquisition provides sensor streams, process context and maintenance history, but these data become operationally useful only after synchronization, cleaning and asset-level contextualization. The machine-learning layer transforms these inputs into anomaly scores, fault classes, RUL estimates or risk horizons. The uncertainty-estimation layer then qualifies the prediction by providing calibrated probabilities, prediction intervals, drift indicators or out-of-domain warnings. Maintenance decision making converts the prediction and its uncertainty into an action, such as continued monitoring, inspection at the next planned stop, load reduction, spare-part ordering or component replacement. Human interaction closes the loop: technicians interpret explanations, accept or override recommendations, add diagnostic feedback and improve future labels. This feedback is essential because maintenance actions alter future degradation trajectories and therefore change the data-generating process itself.

Data engineering is often the largest part of the work. Sensor timestamps must be synchronized; missing data must be marked; maintenance events must be transformed into labels; and production states must be joined to condition data. A bearing vibration window is not interpretable without speed and load. A motor current signature is not comparable across recipes unless operating context is included. For tabular production-line data, product route and station information may be decisive. Effective feature stores for PdM therefore combine raw measurements, engineered health indicators, operating-regime tags and maintenance outcomes.

Model deployment can occur at the edge, on-premise servers or cloud platforms. Edge deployment is useful for low-latency anomaly detection, data reduction and cybersecurity-sensitive assets. Cloud deployment supports large-scale training, fleet comparison and computationally intensive models. Many plants use hybrid deployment: edge devices compute health indicators and alarms, while central systems train models and perform fleet analytics. The choice depends on latency, bandwidth, data governance and integration with existing automation infrastructure.

Human integration is critical. Maintenance technicians need explanations, trend plots, confidence intervals and recommended actions. A dashboard showing only a probability score is insufficient. Useful interfaces display the recent sensor trend, the operating regime, similar historical cases, likely failure mode and suggested inspection. The system should also allow technicians to feed back whether the alarm was correct. This creates a learning loop. Without feedback, labels remain poor, and model improvement stagnates.

Governance must include cybersecurity and safety. PdM data may reveal production capacity, process settings and proprietary know-how. Models should therefore respect access control and plant network segmentation. Safety-critical assets require additional validation and fail-safe design. ML predictions should not override safety systems. They should support maintenance planning and early diagnosis while remaining subordinate to certified protection and control mechanisms.

8. Discussion: what the literature implies for smart manufacturing

The reviewed literature supports five conclusions. First, ML is most valuable when it is matched to the data situation. Deep learning is suitable for rich sequential sensor data and large labelled datasets, while classical ML often remains preferable for small, tabular, high-dimensional or explainability-sensitive problems. The C-MAPSS and bearing literature show the strength of deep and signal-based models; the Bosch and semiconductor cases show the continuing importance of engineered features, imbalance handling and cost-aware classifiers.

Second, operating context is not optional. Smart plants are dynamic. Machines process different products, operate at different speeds and undergo maintenance actions that change the degradation trajectory. Models that ignore context risk learning shortcuts. The Bosch case is a strong reminder that production flow and timing can dominate the predictive structure [11]. In equipment PdM, this means that models should ingest production recipes, load states and maintenance history alongside sensor measurements.

Third, benchmarks should be treated as methodological test beds, not proof of deployment readiness. C-MAPSS, PRONOSTIA and IMS have enabled consistent comparison and rapid progress. They are indispensable for research. Yet real factories have noisier labels, fewer failures and stronger drift. A credible paper or industrial pilot should explain how the method behaves under missing data, domain shift and temporal validation. It should also report baselines that maintenance teams can understand.

Fourth, the field is moving from prediction to decision. Early PdM studies often focused on whether a model can detect a fault or estimate RUL. Current smart manufacturing requires a stronger connection to scheduling, spare parts, production planning and risk. A probability of failure is not the final output; it is an input to a maintenance decision. Cost-aware models, prediction intervals and planning models therefore deserve more attention than marginal improvements in benchmark RMSE.

Fifth, organizational adoption is as important as modelling. A plant may reject a technically accurate model if alarms are unexplained, if technicians cannot verify the diagnosis, or if the model disrupts production plans without clear benefit. Successful PdM programs usually start with critical assets, clear failure modes, measurable maintenance costs and a feedback process. They grow by proving reliability and integrating with existing maintenance workflows. ML should augment engineering judgement, not bypass it.

Limitations of this review

Several limitations of the present work should be acknowledged. First, the review follows a structured narrative design rather than a fully systematic protocol with pre-registered search strings, exhaustive database screening and PRISMA-compliant reporting. The selection of studies reflects the authors' judgement about representativeness and methodological relevance, and may underrepresent work published in languages other than English or in conference proceedings not indexed in major databases. Second, the five representative datasets and associated empirical studies - C-MAPSS, PRONOSTIA/IMS, Bosch, semiconductor ion-implantation and IoT production-line data - were chosen to illustrate diverse industrial data conditions rather than to provide a statistically complete sample of the PdM literature. Findings drawn from these examples should not be generalized without considering whether the asset type, failure mechanism and data quality match the reader's industrial context. Third, the review does not include a quantitative meta-analysis of reported performance metrics. Numerical comparison across studies is complicated by differences in validation protocols, dataset versions, preprocessing choices and evaluation metrics, and aggregating such results without controlling for these factors would risk misleading conclusions. Fourth, the literature coverage is weighted toward journal articles published before mid-2024; rapidly evolving areas such as foundation models, federated learning and LLM-based diagnostics are represented by early studies that may not reflect the current state of the field. Future reviews should apply systematic protocols and include grey literature, industrial case reports and non-English sources to provide

a more complete picture of PdM practice in smart manufacturing.

9. Research gaps and future directions

Several gaps remain open. The first is reproducibility. Public datasets have enabled progress, but many industrial studies use proprietary data that cannot be shared. Researchers should at least publish preprocessing logic, validation protocols and realistic baselines. Synthetic or anonymized datasets generated from digital twins may help, but they must preserve the statistical difficulties of real plants, including missingness, drift and censoring.

The second gap is transferability. A model trained on one bearing rig, one production line or one turbine dataset often performs poorly elsewhere. Domain adaptation, transfer learning and federated learning are promising because they can reuse knowledge while respecting data ownership [39]. However, industrial transfer is not only statistical. Differences in sensor placement, sampling frequency, maintenance policy and production context must be documented. Without this documentation, transfer results are hard to interpret.

The third gap is uncertainty and calibration. Many PdM systems still output deterministic labels or RUL values. Smart plants need calibrated probabilities and intervals to plan interventions. Future work should report calibration error, decision curves and risk-sensitive metrics. It should also study how human decision-makers interpret uncertainty. A technically sound interval can still be misused if the interface encourages overconfidence.

The fourth gap is integration with maintenance optimization. Prediction should be linked to scheduling constraints, spare-part availability, crew capacity and production plans. A model that predicts a likely failure during a peak production window may suggest a different action than the same prediction during planned downtime. Combining ML with operations research is therefore a natural direction. The recent survey literature on PdM planning and Industry 4.0 supports this broader decision perspective [26,40].

A gap that has opened rapidly since 2023 concerns foundation models and LLMs. Early PdM research assumed that models must be purpose-built from domain data. Pre-trained LLMs upend this assumption by bringing broad linguistic and reasoning knowledge to a new task through fine-tuning. Zheng et al. showed that this transfer is feasible for fault diagnosis even with limited labelled examples [21], and multimodal extensions suggest that LLMs could unify sensor streams with the unstructured text that pervades industrial maintenance records [22]. Several open questions remain. It is not yet clear how reliably fine-tuned LLMs generalize across different machine types, plant layouts or product recipes - the kind of domain shift that is routine in smart manufacturing. Calibration is also poorly characterized: an LLM that outputs a confident fault label without an associated probability interval is a poor fit for cost-aware maintenance decisions. Computational cost is a further barrier; current frontier LLMs cannot be deployed on industrial edge devices without significant quantization or distillation. Future work should establish reproducible benchmarks for LLM-based PdM that include domain-shift evaluation, calibration metrics and latency constraints, so that the genuine industrial value of foundation models can be separated from benchmark-optimized performance.

Finally, sustainability impacts should be assessed quantitatively rather than treated only as a general benefit of predictive maintenance [7]. In smart manufacturing, PdM can reduce environmental impact by lowering unplanned downtime, avoiding emergency repairs, reducing scrap, extending component lifetime and improving the use of spare parts. However, these benefits are not automatic. A poorly calibrated PdM system may generate excessive false alarms, unnecessary inspections and premature component replacements, which can increase material consumption, labour intensity and production interruptions.

Future studies should therefore evaluate PdM using measurable sustainability indicators. These may include the reduction in unplanned downtime hours, the decrease in scrap rate, the number of avoided emergency interventions, the extension of component service life, the reduction in unnecessary replacements, and the change in energy consumption before and after PdM implementation. Where possible, these operational indicators should be converted into environmental indicators such as avoided material waste, avoided energy use and estimated CO2-equivalent emissions [42,43]. For example, avoided scrap can be estimated by multiplying the reduction in rejected material by the emission factor of the material used, while avoided downtime-related energy losses can be estimated from machine power consumption during stoppage, restart and recovery phases.

A quantitative sustainability assessment should also include the additional burdens introduced by PdM itself. These include the energy required for sensors, edge devices, cloud computation and data transmission, as well as the environmental impact of additional inspections or replacements caused by false alarms. The net sustainability effect of PdM should therefore be understood as the difference between avoided impacts and added impacts. In this sense, PdM is environmentally beneficial only when the savings from reduced downtime, scrap and premature failures exceed the extra impacts created by monitoring, computation and unnecessary maintenance actions.

This quantitative perspective would make sustainability claims more transparent and comparable across studies. Instead of stating that PdM supports sustainable manufacturing in general, future research should report how much downtime, scrap, energy use, component waste or CO2-equivalent emissions are actually reduced under a PdM-assisted maintenance policy compared with a corrective or preventive maintenance baseline. Such reporting would also help distinguish genuinely sustainable PdM systems from systems that improve technical

prediction accuracy but shift costs and environmental burdens elsewhere in the production system.

10. Conclusions

Machine learning has become a core enabling technology for equipment failure prediction in intelligent manufacturing plants. The literature shows strong evidence that ML can detect abnormal behaviour, classify faults, estimate RUL and support maintenance decisions when appropriate data and validation are available. Classical feature-based models remain highly relevant, especially where domain knowledge is strong and labelled failures are scarce. Deep learning offers powerful representation learning for complex time-series and signal data, as demonstrated in C-MAPSS, bearing prognosis and broader machine-health monitoring research. An emerging extension of this paradigm is the adaptation of pre-trained foundation models and LLMs for fault diagnosis, where fine-tuning on limited industrial data has shown promise as an alternative to training specialist models from scratch [21,22]. Whether this approach will prove robust under the distribution shift and latency constraints of real plant deployments remains an active research question. Hybrid approaches that combine signal processing, physics, digital twins and cost-aware decision rules are increasingly necessary for deployment.

The main conclusion is that PdM success depends less on choosing a fashionable algorithm than on building a reliable decision pipeline. The pipeline must include meaningful sensors, clean and contextualized data, appropriate labels, temporally honest validation, uncertainty estimation, economic evaluation and human-centred integration. It should also include quantitative sustainability indicators, because an accurate PdM model may still be unsuitable for deployment if it reduces downtime but increases unnecessary replacements, energy use or material waste. The reviewed examples also show that there is no universal benchmark for smart-factory failure prediction. C-MAPSS teaches sequence modelling and RUL estimation; PRONOSTIA and IMS teach vibration-based degradation analysis; Bosch teaches high-dimensional imbalanced production data; semiconductor ion-implantation teaches horizon-specific cost-aware maintenance; and recent IoT production-line studies teach the importance of end-to-end operational pipelines.

For researchers, the priority is to move beyond offline accuracy and toward reproducible, uncertainty-aware and decision-oriented evaluation. For practitioners, the priority is to begin with critical assets and well-defined maintenance decisions, then select ML methods that fit the available data. In smart manufacturing, machine learning is not a replacement for maintenance engineering. It is a way to transform dispersed sensor, process and maintenance data into timely, quantified and reviewable evidence for better maintenance action.

References

[1] Ahmad R, Kamaruddin S. An overview of time-based and condition-based maintenance in industrial applications. Comput Ind Eng. 2012;63(1):135-149. doi:10.1016/j.cie.2012.02.002.

[2] Jardine AKS, Lin D, Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech Syst Signal Process. 2006;20(7):1483-1510. doi:10.1016/j.ymssp.2005.09.012.

[3] Si XS, Wang W, Hu CH, Zhou DH. Remaining useful life estimation: a review on the statistical data driven approaches. Eur J Oper Res. 2011;213(1):1-14. doi:10.1016/j.ejor.2010.11.018.

[4] Lee J, Bagheri B, Kao HA. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manuf Lett. 2015;3:18-23. doi:10.1016/j.mfglet.2014.12.001.

[5] Lee J, Lapira E, Bagheri B, Kao HA. Recent advances and trends in predictive manufacturing systems in big data environment. Manuf Lett. 2013;1(1):38-41. doi:10.1016/j.mfglet.2013.09.005.

[6] Carvalho TP, Soares FAA, Vita R, Francisco RP, Basto JP, Alcala SGS. A systematic literature review of machine learning methods applied to predictive maintenance. Comput Ind Eng. 2019;137:106024. doi:10.1016/j.cie.2019.106024.

[7] Cinar ZM, Nuhu AA, Zeeshan Q, Korhan O, Asmael M, Safaei B. Machine learning in predictive maintenance towards sustainable smart manufacturing in Industry 4.0. Sustainability. 2020;12(19):8211. doi:10.3390/su12198211.

[8] Zonta T, da Costa CA, da Rosa Righi R, de Lima MJ, da Trindade ES, Li GP. Predictive maintenance in the Industry 4.0: a systematic literature review. Comput Ind Eng. 2020;150:106889. doi:10.1016/j.cie.2020.106889.

[9] Wen Y, Gao L, Li X. Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective. Measurement. 2022;187:110276. doi:10.1016/j.measurement.2021.110276.

[10] Susto GA, Schirru A, Pampuri S, McLoone S, Beghi A. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. IEEE Trans Ind Inform. 2015;11(3):812-820. doi:10.1109/TII.2014.2349359.

[11] Mangal A, Kumar N. Using big data to enhance the Bosch production line performance: a Kaggle challenge. arXiv:1701.00705; 2017.

[12] Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv:1607.00148; 2016. doi:10.48550/arXiv.1607.00148.

[13] Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge, MA: MIT Press; 2016.

[14] Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. Mech Syst Signal Process. 2019;115:213-237. doi:10.1016/j.ymssp.2018.05.050.

[15] Khan S, Yairi T. A review on the application of deep learning in system health management. Mech Syst Signal Process. 2018;107:241-265. doi:10.1016/j.ymssp.2017.11.024.

[16] Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: a systematic review from data acquisition to RUL prediction. Mech Syst Signal Process. 2018;104:799-834. doi:10.1016/j.ymssp.2017.11.016.
[17] Wu F, Wu Q, Tan Y, Xu X. Remaining useful life prediction based on deep learning: a survey. Sensors. 2024;24(11):3454. doi:10.3390/s24113454.
[18] Saxena A, Goebel K. Turbofan engine degradation simulation data set. NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA; 2008.
[19] Li X, Ding Q, Sun JQ. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf. 2018;172:1-11. doi:10.1016/j.ress.2017.11.021.
[20] Ellefsen AL, Bjorlykhaug E, Aesoy V, Ushakov S, Zhang H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliab Eng Syst Saf. 2019;183:240-251. doi:10.1016/j.ress.2018.11.027.
[21] Zheng S, Pan K, Liu J, Chen Y. Empirical study on fine-tuning pre-trained large language models for fault diagnosis of complex systems. Reliab Eng Syst Saf.2024;252:110382. doi:10.1016/j.ress.2024.110382.
[22] Jose S, Nguyen KTP, Medjaher K, Zemouri R, Lévesque M, Tahan A. Advancing multimodal diagnostics: Integrating industrial textual data and domain knowledge with large language models. Expert Syst Appl. 255, 124603, DOI 10.1016/j.eswa.2024.124603.
[23] ISO. ISO 23247-1:2021 Automation systems and integration - Digital twin framework for manufacturing - Part 1: Overview and general principles. Geneva: International Organization for Standardization; 2021.
[24] Tao F, Zhang H, Liu A, Nee AYC. Digital twin in industry: state-of-the-art. IEEE Trans Ind Inform. 2019;15(4):2405-2415. doi:10.1109/TII.2018.2873186.
[25] Ran Y, Zhou X, Lin P, Wen Y, Deng R. A survey of predictive maintenance: systems, purposes and approaches. IEEE Commun Surv Tutor. 2019;21(4):3783-3816. doi:10.1109/COMST.2019.2905569.
[26] ul Hassan I, Panduru K, Walsh J. Predictive maintenance in Industry 4.0: a review of data processing methods. Procedia Comput Sci. 2025;257:896-903. doi:10.1016/j.procs.2025.03.115.
[27] Ramasso E, Saxena A. Performance benchmarking and analysis of prognostic methods for CMAPSS datasets. Int J Progn Health Manag. 2014;5(2):1-15.
[28] Nectoux P, Gouriveau R, Medjaher K, Ramasso E, Morello B, Zerhouni N, Varnier C. PRONOSTIA: an experimental platform for bearings accelerated degradation tests. In: IEEE International Conference on Prognostics and Health Management; 2012. p. 1-8. doi:10.1109/PHM.2012.6221634.
[29] Qiu H, Lee J, Lin J, Yu G. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. J Sound Vib. 2006;289(4-5):1066-1090. doi:10.1016/j.jsv.2005.03.007.
[30] Tasci B, Omar A, Ayvaz S. Remaining useful lifetime prediction for predictive maintenance in manufacturing. Comput Ind Eng. 2023;184:109566. doi:10.1016/j.cie.2023.109566.
[31] Guo C, Pleiss G, Sun Y, Weinberger KQ. On Calibration of Modern Neural Networks. In: Proceedings of the 34th International Conference on Machine Learning; 2017. arXiv:1706.04599.
[32] Gal Y, Ghahramani Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In: Proceedings of the 33rd International Conference on Machine Learning; 2016. p. 1050–1059.
[33] Lakshminarayanan B, Pritzel A, Blundell C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In: Advances in Neural Information Processing Systems; 2017.
[34] Angelopoulos AN, Bates S. Conformal Prediction: A Gentle Introduction. Foundations and Trends in Machine Learning. 2023;16(4):494–591. doi:10.1561/2200000101.
[35] Gneiting T, Raftery AE. Strictly Proper Scoring Rules, Prediction, and Estimation. J Am Stat Assoc. 2007;102(477):359–378. doi: 10.1198/016214506000001437.
[36] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. NeurIPS 30; 2017, p. 4765–4774.
[37] Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?": Explaining the predictions of any classifier. KDD 2016, p. 1135–1144, doi:10.1145/2939672.2939778.
[38] Vogl GW, Weiss BA, Helu M. A review of diagnostic and prognostic capabilities and best practices for manufacturing. J Intell Manuf. 2019;30:79-95. doi:10.1007/s10845-016-1228-8.
[39] Pruckovskaja V, Weissenfeld A, Heistracher C, Graser A, Kafka J, Leputsch P, Schall D, Kemnitz J. Federated learning for predictive maintenance and quality inspection in industrial applications. arXiv:2304.11101; 2023.
[40] Hector I, Panjanathan R. Predictive maintenance in Industry 4.0: a survey of planning models and machine learning techniques. PeerJ Comput Sci. 2024;10:e2016. doi:10.7717/peerj-cs.2016.
[41] National Institute of Standards and Technology (NIST). Artificial Intelligence Risk Management Framework (AI RMF 1.0). Gaithersburg, MD: NIST; 2023. doi:10.6028/NIST.AI.100-1
[42] ISO. ISO 14040:2006 Environmental management - Life cycle assessment - Principles and framework. Geneva: International Organization for Standardization; 2006.
[43] ISO. ISO 14044:2006 Environmental management - Life cycle assessment - Requirements and guidelines. Geneva: International Organization for Standardization; 2006.