A Data-Driven Artificial Intelligence Framework for Predictive Maintenance in Smart Manufacturing

Yuchen Zhao
School of Mechanical Engineering
Shanghai Jiao Tong University
Shanghai, China

Shuai Chen
School of Control Science and Engineering
Shandong University
Jinan, China

Robert L. Mitchell
Department of Mechanical and Aerospace Engineering
University of California, Irvine
Irvine, CA, USA

Daniel P. Foster
Department of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA

January 29, 2026

Abstract

Predictive maintenance (PdM) is a cornerstone capability for smart manufacturing, where maintenance actions are increasingly triggered by data and optimized against operational constraints rather than by fixed schedules. Despite rapid progress in sensing, industrial connectivity, and learning algorithms, practitioners still face persistent gaps: (i) heterogeneous and imperfect data streams; (ii) distribution shift across assets, sites, and operating regimes; (iii) uncertainty in remaining useful life (RUL) estimation and failure risk forecasting; and (iv) the translation from model outputs to actionable maintenance decisions under cost, safety, and availability requirements.

This paper proposes a data-driven artificial intelligence framework that unifies (1) an industrial data layer for acquisition, synchronization, and feature governance; (2) a modeling layer that supports both sequence-to-RUL regression and time-to-event (survival) prediction with calibrated uncertainty; and (3) a decision layer that maps forecasts to maintenance policies through cost-aware optimization. We provide core mathematical formulations, pseudocode for the training and deployment pipeline, and implementation-ready design choices for edge–cloud execution in modern smart factories. The manuscript is written to be reusable as an engineering reference: assumptions are stated explicitly, interfaces between modules are defined, and evaluation protocols are aligned with common public benchmarks and industrial constraints.

Keywords: predictive maintenance; smart manufacturing; remaining useful life; survival analysis; uncertainty quantification; industrial IoT; digital twin.

1 Introduction

Smart manufacturing systems are increasingly instrumented with vibration, current, acoustic, thermal, and process sensors, and are further connected through Industrial Internet of Things (IIoT) middleware. This creates a technically attractive setting for predictive maintenance: rather than rely on preventive schedules, engineers can forecast degradation and failure risk, then schedule interventions to minimize downtime while meeting quality and safety targets.

In practice, predictive maintenance is not a single model but a full-stack capability. The useful unit of analysis is an end-to-end workflow spanning data governance, robust modeling, and decision-making. A strong PdM solution must: handle missing and asynchronous signals; incorporate context (load, recipe, product type, environment); quantify uncertainty; survive distribution shift; and offer transparent, auditable outputs that maintenance planners can act upon. Recent surveys and application studies have stressed that success depends as much on engineering choices and evaluation discipline as on model architecture, especially in data-scarce regimes and in multi-site deployments [1–3].

This paper contributes a consolidated framework and a reference implementation blueprint. The key design principle is modularity with explicit contracts:

- **Data layer contract:** deliver aligned windows $\mathbf{X}_{t-L:t}$ with metadata $\mathbf{c}_t$ and quality flags $\mathbf{q}_t$.
- **Model layer contract:** output either a calibrated RUL distribution $p(\tau \mid \mathbf{X}, \mathbf{c})$ or a discrete-time hazard λ_t with uncertainty.
- **Decision layer contract:** map predictions to maintenance actions a_t through an explicit cost function and operational constraints.

2 Problem Setting and Notation

Consider a fleet of assets indexed by $i \in \{1, \ldots, N\}$. For each asset we observe a multivariate time series $\mathbf{x}^{(i)}(t) \in \mathbb{R}^d$ sampled at (possibly irregular) times $t \in \mathcal{T}^{(i)}$, together with contextual variables $\mathbf{c}^{(i)}(t)$ such as operating mode, control settings, production recipe, and environmental conditions.

We define a windowed representation for learning:

$$\mathbf{X}_t^{(i)} = \left[\mathbf{x}^{(i)}(t-L+1), \ldots, \mathbf{x}^{(i)}(t)\right] \in \mathbb{R}^{L \times d}, \tag{1}$$

and denote the (latent) time-to-failure from time t by $\tau_t^{(i)} \geq 0$. In run-to-failure settings, $\tau_t^{(i)}$ can be derived from the known failure time $T_f^{(i)}$ as $\tau_t^{(i)} = T_f^{(i)} - t$. In censored settings, only partial information is available: an asset may be removed from observation before failure, requiring survival modeling.

The PdM objective is twofold:

1. **Prognostics:** estimate τ or failure risk conditional on observed data and context.
2. **Decision-making:** choose maintenance actions that minimize expected lifecycle cost while respecting constraints.

3 Related Work

Data-driven predictive maintenance spans anomaly detection, fault diagnosis, and prognostics. Contemporary smart manufacturing adds two additional pressures: (i) large-scale connectivity that enables continuous monitoring and closed-loop decision making; and (ii) rapid changes in equipment configurations and production mixes that amplify distribution shift.

Recent research highlights three themes.

(1) Smart manufacturing integration. PdM is increasingly embedded into cyber–physical production systems and digital twins, where models interact with operational planning and asset management [1, 4–6].

(2) Advanced sequence models. Beyond classical feature engineering with tree ensembles, deep models for RUL prediction now frequently use attention, temporal convolution, and hybrid signal encoders. Transformer-style architectures have been explored for RUL and for multivariate sensor fusion [7–10].

(3) Trustworthy PdM. Industrial deployments demand calibrated uncertainty, robustness, and explainability. Recent work includes distributionally robust learning, conformal prediction for reliable intervals, and post-hoc explanations tailored to rotating machinery and process industries [11–14].

4 Proposed Framework

Figure 1 summarizes the proposed end-to-end architecture.

4.1 Layer 1: Industrial Data Pipeline

The data layer converts raw signals into learning-ready examples. In manufacturing, the major obstacles are not only noise but also semantics: tags change, sensors drift, and process steps alter the meaning of identical measurements.

Acquisition and synchronization. We assume a mix of (i) high-frequency condition monitoring streams (vibration, acoustic emission) and (ii) low-frequency process variables from PLC/SCADA. A common practice is to build an event-time aligned timeline and store raw signals with immutable metadata; feature extraction can then be rerun as models evolve.

Data quality flags. For each window $\mathbf{X}_t$ we maintain a vector $\mathbf{q}_t$ marking missing segments, sensor health, and operating transients. These flags can be consumed by the model (as masks) and by governance rules (to exclude unreliable periods).

Labeling and supervision. PdM supervision can come from run-to-failure histories, maintenance work orders, inspection measurements, or alarms. The framework supports both:

- RUL regression when failure times are available and meaningful.
- Survival / hazard prediction when censoring is substantial or failure definition is ambiguous.

4.2 Layer 2: Modeling (Prognostics)

We propose a unified probabilistic interface that can be instantiated with different backbones.

Sequence encoder. Given window $\mathbf{X}_t$ and context $\mathbf{c}_t$, an encoder produces a latent representation

$$\mathbf{h}_t = f_\theta(\mathbf{X}_t, \mathbf{c}_t) \in \mathbb{R}^m. \tag{2}$$

f_θ may be a temporal convolutional network, a GRU, or an attention-based model.

Output heads. Two heads are supported.

- **RUL distribution head:** output parameters of a predictive distribution $p_\theta(\tau \mid \mathbf{h}_t)$.
- **Hazard head:** output a discrete hazard sequence λ_{t+k} for $k = 1, \ldots, K$.

Calibration and uncertainty. To make decisions under risk, we require uncertainty estimates. The framework supports: (i) quantile regression; (ii) evidential learning; and (iii) conformal prediction for distribution-free interval coverage [11, 12, 15].

4.3 Layer 3: Decision-Making (Maintenance Policy)

Forecasts become useful only after they are translated into actions.

Cost-aware thresholding. A baseline policy triggers maintenance when predicted failure risk exceeds a threshold or when the lower quantile of RUL falls below a planning horizon. Thresholds are tuned on a validation set using explicit costs: preventive maintenance cost C_{pm}, corrective maintenance cost C_{cm}, downtime cost per hour C_d, and inventory/crew constraints.

Optimization and control. For complex lines, maintenance decisions interact with production scheduling. We therefore define a Markov decision process (MDP) and optimize expected cost. In data-scarce settings, a practical alternative is to solve a rolling-horizon mixed-integer program with forecasted risks as inputs.

5 Core Formulations

This section lists the main mathematical objects used by the framework.

5.1 RUL Regression as Conditional Distribution Learning

Let $\tau \in \mathbb{R}_{\geq 0}$ denote the remaining useful life from time t. A point estimator $\hat{\tau}_t$ is often insufficient; we instead predict a distribution.

Gaussian likelihood (heteroscedastic). A common choice is

$$p_\theta(\tau \mid \mathbf{h}_t) = \mathcal{N}\big(\mu_\theta(\mathbf{h}_t),\, \sigma_\theta^2(\mathbf{h}_t)\big), \tag{3}$$

trained by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{NLL}}(\theta) = \sum_{(t,i)} \left[\frac{(\tau_t^{(i)} - \mu_\theta(\mathbf{h}_t^{(i)}))^2}{2\sigma_\theta^2(\mathbf{h}_t^{(i)})} + \frac{1}{2} \log \sigma_\theta^2(\mathbf{h}_t^{(i)}) \right]. \tag{4}$$

Quantile regression for robust intervals. For a quantile level $\alpha \in (0, 1)$, the pinball loss is

$$\rho_\alpha(u) = \max(\alpha u, (\alpha - 1)u), \quad u = \tau - \hat{q}_\alpha, \tag{5}$$

and the training loss is $\mathcal{L}_\alpha = \sum \rho_\alpha(\tau - \hat{q}_\alpha)$. Predicting multiple quantiles yields asymmetric prediction intervals that are often more stable under heavy-tailed noise.

5.2 Survival Modeling via Discrete Hazard

When censoring is present, we predict the hazard λ_k for future steps $k = 1, \ldots, K$:

$$\lambda_k = \mathbb{P}(T = k \mid T \geq k, \mathbf{h}_t), \qquad 0 < \lambda_k < 1. \tag{6}$$

The survival function over a horizon is

$$S(k) = \mathbb{P}(T > k \mid \mathbf{h}_t) = \prod_{j=1}^{k} (1 - \lambda_j), \tag{7}$$

and the probability mass at step k is $p(T = k) = \lambda_k \prod_{j<k}(1 - \lambda_j)$. This representation enables straightforward incorporation of censoring by maximizing the appropriate likelihood.

5.3 Decision Objective: Expected Cost of Maintenance

Let $a_t \in \{\text{do nothing}, \text{inspect}, \text{pm}\}$ denote an action at time t. A minimal cost model for a single asset can be written as

$$\min_\pi \; \mathbb{E}_\pi \Big[\sum_{t=0}^{\infty} \gamma^t \big(C(a_t) + C_{\text{fail}} \, \mathbb{I}[\text{failure at } t] \big) \Big], \tag{8}$$

where π is a policy mapping observations to actions and $\gamma \in (0, 1]$ is a discount factor.

In many plants, a simpler and more auditable approach is to optimize maintenance trigger thresholds under explicit constraints; the probabilistic outputs above allow engineers to trade off false alarms and missed failures.

6 Pseudocode

Algorithm 1 outlines the training and deployment loop.

```
Algorithm 1: End-to-end data-driven AI framework for predictive maintenance.
  Input  : Raw sensor streams $\{\mathbf{x}^{(i)}(t)\}$, context $\{\mathbf{c}^{(i)}(t)\}$, maintenance logs, horizon $L$,
           forecast steps $K$
  Output: Deployed prognostics model and a maintenance decision interface
  Data layer:
  1. Ingest raw signals; standardize tag metadata; record sampling rates and time offsets.
  2. Align timestamps; construct windows $\mathbf{X}_t^{(i)} \in \mathbb{R}^{L\times d}$ and masks/quality flags $\mathbf{q}_t^{(i)}$.
  3. Generate labels: RUL targets $\tau_t^{(i)}$ for run-to-failure; censoring indicators for survival.
  Model layer:
  1. Train encoder $f_\theta$ and output head(s) to minimize $\mathcal{L}_{\mathrm{NLL}}$ and/or quantile losses.
  2. Calibrate uncertainty (temperature scaling, conformal prediction, or evidential regularization).
  3. Validate under distribution shift (new regimes, new assets) and log failure modes.
  Decision layer:
  1. Define cost parameters $(C_{\mathrm{pm}}, C_{\mathrm{cm}}, C_d)$ and operational constraints.
  2. Tune trigger rules or solve a rolling-horizon optimization using predicted risks.
  3. Deploy edge–cloud inference with monitoring (drift, coverage, latency).
```

7 Framework Diagram and Illustrative Tables/Figures

7.1 System Architecture Diagram

7.2 Evaluation Metrics Table (Template)

7.3 Illustrative Plot

8 Experimental Protocol (Benchmark-Oriented)

To enable reproducible comparisons, the framework recommends separating evaluation into: (i) within-asset forecasting (same asset over time), (ii) cross-asset generalization (new units), and (iii) cross-site generalization (new plant). Public benchmarks such as NASA C-MAPSS and industrial acoustic anomaly datasets can be used for development, while production deployment should prioritize domain-specific validation.

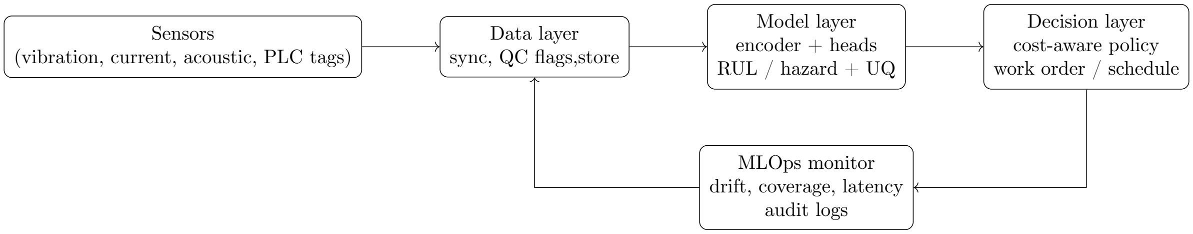

Figure 1: A modular PdM architecture with explicit interfaces between data, modeling, and decision-making.

Task	Metric	Notes
RUL regression	RMSE / MAE	Report by operating regime if possible
Uncertainty	PICP / MPIW	Coverage and width for prediction intervals
Survival	Concordance / NLL	Handle censoring explicitly
Decision quality	Expected cost	Use plant-specific cost model

Table 1: Recommended metrics aligned with both prognostics accuracy and decision utility.

8.1 Data Splits and Leakage Prevention

A major source of over-optimism in PdM is leakage due to random sampling across time. We therefore recommend chronological splits, with entire units (or batches) held out when evaluating cross-asset performance. For process industries, split by campaigns or recipes to reflect real changes.

8.2 Baselines

A defensible experimental section typically includes:

- Classical: random forest / gradient boosting on engineered features.
- Deep sequential: TCN, GRU, attention-based encoder.
- Uncertainty: quantile models and conformal intervals.

9 Deployment Considerations in Smart Manufacturing

9.1 Edge–Cloud Partitioning

High-frequency condition monitoring may require edge inference for latency and bandwidth reasons. The framework supports splitting computation: (i) feature extraction and lightweight inference at the edge; (ii) periodic retraining and global calibration in the cloud. A digital-twin layer can provide a consistent interface for simulation, scenario analysis, and what-if planning [5, 6].

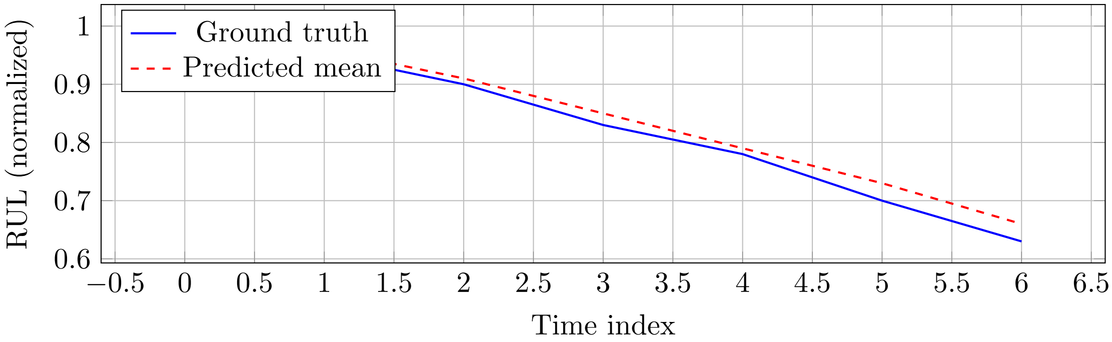

Figure 2: Illustrative RUL trajectory plot (replace with results from your chosen datasets/line).

9.2 Monitoring, Drift, and Governance

Model performance can degrade when sensors drift, maintenance practices change, or new product variants are introduced. We recommend monitoring:

- data drift (marginal distribution changes),
- concept drift (mapping from signals to failures), and
- coverage drift (prediction interval coverage).

Audit logs should retain model versions, calibration parameters, and the decision thresholds used to create work orders.

10 Discussion

The proposed framework emphasizes that PdM is a socio-technical system. Even a well-calibrated RUL estimator can fail operationally if it does not align with maintenance planning horizons, spare-part lead times, or technician availability. Conversely, modest predictive models can generate strong economic value if their outputs are stable, interpretable, and integrated into decision processes.

Two open challenges merit particular attention.

Generalization under distribution shift. Smart factories often operate fleets of nominally similar assets under different loads, operators, and environments. Models should be validated under realistic shifts, and domain adaptation or federated learning may be required when data sharing is constrained [16, 17].

Trust and accountability. Maintenance decisions have safety and financial implications. Uncertainty quantification and explainability are therefore not optional; they are prerequisites for adoption and for post-incident analysis [13, 14].

11 Conclusion

We presented a data-driven AI framework for predictive maintenance in smart manufacturing, structured around three explicit layers: industrial data engineering, probabilistic prognostics, and cost-aware decision-making. The framework provides core formulations for RUL and hazard prediction, pseudocode for the end-to-end pipeline, and implementation-ready guidance for deployment in edge–cloud smart factory environments.

Future work should focus on rigorous multi-site validation, robust learning under limited failure data, and tighter integration between prognostics uncertainty and production scheduling.

References

[1] J. Theil, C. Rosenkranz, and J. Weinand, “A review of smart manufacturing–based predictive maintenance solutions and research directions,” Forum, vol. 40, no. 1, pp. 1–44, 2025.

[2] M. Iswad, M. Nasir, L. Zhang et al., “Data-driven predictive maintenance in industry 4.0: A systematic literature review,” International Journal of System Assurance Engineering and Management, pp. 1–33, 2025.

[3] Y. Yesar, R. Sharma, and P. Kumar, “Data-driven predictive maintenance for smart manufacturing: Current challenges and future trends,” Computers & Industrial Engineering, vol. 188, p. 109918, 2024.

[4] A. Sharma, T. Hedberg et al., “A smart manufacturing predictive maintenance data architecture for interoperable and scalable analytics,” in Proceedings of the Winter Simulation Conference (WSC), 2023, pp. 1–12.

[5] G. Pitel, R. Jurdak et al., “A roadmap for predictive maintenance with digital twins in smart manufacturing,” 2023.

[6] M. Fahim, A. Rehman et al., “Digital twins for maintenance in smart manufacturing: Concepts, methods, and industrial perspectives,” Journal of Intelligent Manufacturing, pp. 1–27, 2024.

[7] Y. Zhang, H. Wang, and Q. Li, “Remaining useful life prediction with graph attention and dual attention transformer for multi-sensor degradation data,” International Journal of Fatigue, vol. 176, p. 107829, 2023.

[8] J. Liu, Y. Chen, and X. Zhao, “Denoising transformer networks for remaining useful life prediction under complex operating conditions,” Energies, vol. 16, no. 4, p. 1921, 2023.

[9] S. Kim, J. Park, and H. Lee, “An improved transformer for remaining useful life prediction with multi-scale feature fusion,” Electronics, vol. 13, no. 6, p. 1102, 2024.

[10] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” in Advances in Neural Information Processing Systems, 2017.

[11] J. Kang, Y. Lin et al., “Distribution-free prediction intervals for remaining useful life via conformal prediction,” International Journal of Prognostics and Health Management, vol. 14, no. 2, 2023.

[12] S. Feldman, S. Bates et al., "Copal: Conformal prediction after learning for time-series forecasting," in Proceedings of the International Conference on Machine Learning (ICML), ser. Proceedings of Machine Learning Research, 2024.

[13] D. Serrano, M. García et al., "Explainable artificial intelligence for predictive maintenance of rotating machinery: Methods and case studies," IEEE Access, vol. 12, pp. 34 567–34 592, 2024.

[14] T. Schmalenbach, J. Krüger et al., "From model to maintenance: A case study on deploying predictive maintenance with calibrated uncertainty in a smart factory," Schmalenbach Journal of Business Research, vol. 77, pp. 1–24, 2025.

[15] M. Rahman, S. Islam et al., "Evidential deep learning for uncertainty-aware remaining useful life estimation in industrial assets," Journal of Artificial Intelligence and Soft Computing Research, vol. 15, no. 1, pp. 55–76, 2025.

[16] R. Khan, T. Nguyen et al., "Federated learning for predictive maintenance in industrial iot: Challenges and a reference architecture," 2023.

[17] L. Gomez, J. Park et al., "Edge ai for predictive maintenance in smart manufacturing: Design patterns and benchmarking," 2024.