American Journal of Interdisciplinary Studies

Volume: 16; **Issue:** 1
Pages: 01-40
eISSN: 3067-5146

AI and IoT Synergies in Enhancing Business Operations: A Simulation-Based Evaluation of Predictive Maintenance, Edge–Cloud Placement and Business Value

Qaium Hossain[1];

[1]. *Master of Science in Management Information Systems, Lamar University, Texas, USA;*
Email: qaiumadi33@gmail.com

Doi: 10.63125/kvspyy71
Received: *09 October 2025;* ***Revised:*** *11 November 2025;* ***Accepted:*** *10 December 2025;* ***Published:*** *02 January 2026;*

Abstract

The Internet of Things (IoT) gives businesses continuous, granular data about their physical operations; artificial intelligence (AI) turns those data into forecasts and decisions. Their combination is widely credited with lower downtime and cost, but most published figures are vendor claims or case studies whose data cannot be reproduced. This article develops a reference architecture and a synergy framework for AI–IoT integration across five operational domains, and tests the central claim, that AI and IoT together deliver more than either alone, in a transparent simulation of predictive maintenance. A synthetic fleet of 96 machines over three years has two latent degradation modes, load- and temperature-dependent wear, shocks and machine-specific failure thresholds; five sensor streams carry machine-specific baselines, load confounding, noise, drift and dropout. Gradient-boosted trees on lagged rolling features forecast failure within 72 hours with a test AUC of 0.91, against 0.83 for the best single-sensor rule and 0.75 for a model without sensor data. Six maintenance policies were simulated on identical degradation paths in five replications. The AI+IoT policy cut failures from 1.93 to 0.20 per machine-year and cost per machine-year by 56% against reactive maintenance, 37% against time-based maintenance, 18 to 21% against tuned condition-based rules and 39% against an AI-only policy, while still costing 63% more than a perfect-foresight oracle. The additive interaction between AI and IoT was negative in every replication because the separate contributions already approached the attainable ceiling; the combination captured 77% of the attainable saving, the share expected from two complementary, independent contributions. A five-year value model gives a net present value that is large against reactive maintenance, modest against an existing condition-based programme and negative for small or reliable fleets, with break-even failure intensities for each baseline; a placement model shows cloud inference cheaper and edge inference better on latency, bandwidth and autonomy. All results describe the simulated fleet; the contributions are a reproducible testbed, a decision-focused evaluation protocol, an implementation roadmap and a critical reading of circulating percentage claims.

Keywords

Internet Of Things, Industrial IOT, Artificial Intelligence, Predictive Maintenance, Edge Computing, Digital Twin, Business Operations, Simulation, Machine Learning, Business Value

Introduction

Two technologies have reshaped how organisations run their physical operations. The Internet of Things (IoT) connects machines, vehicles, buildings and inventory to networks through sensors and controllers, so that their state is observed continuously rather than inspected occasionally (Atzori et al., 2010; Gubbi et al., 2013; Xu et al., 2014). Artificial intelligence (AI), in the form of machine learning on streams of operational data, turns that observation into forecasts of failure, demand and load, into optimised schedules and into automated responses (Lee, Davari, et al., 2018; Zhong et al., 2017; Mohammadi et al., 2018). The industrial version of this combination, variously called the Industrial Internet of Things, Industry 4.0 or industrial AI, has been the subject of national strategies, consultancy forecasts and a large academic literature (Kagermann et al., 2013; Lasi et al., 2014; Lu, 2017; Sisinni et al., 2018; Boyes et al., 2018). Its promise is operational: higher equipment availability, lower maintenance and energy cost, leaner inventories, faster detection of security incidents and better product quality (Porter & Heppelmann, 2014; Porter & Heppelmann, 2015; Dalenogare et al., 2018).

The claim that AI and IoT are synergistic, that their combination delivers more than the sum of their separate contributions, is central to this promise and is rarely tested. IoT without AI produces dashboards and threshold alarms; AI without IoT works on sparse transactional records; the argument is that continuous sensing supplies the signal that learning needs, and learning supplies the interpretation that sensing lacks (Lee et al., 2015; Kwon et al., 2016; Compare et al., 2020). The evidence offered for the argument is mostly of two kinds: surveys that associate adoption with self-reported performance (Dalenogare et al., 2018; Mikalef et al., 2019; Gunasekaran et al., 2017), and case studies and vendor reports that cite percentage improvements, such as a stated reduction in downtime or maintenance cost, without the data, the baseline or the method that produced them. Systematic reviews of predictive maintenance note the same gap: many studies report model accuracy on benchmark datasets, few connect accuracy to maintenance decisions and business outcomes, and fewer still compare against the simpler policies a plant could run without machine learning (Carvalho et al., 2019; Zonta et al., 2020; Nunes et al., 2023).

This article addresses the gap in three steps. First, it synthesises the literature into a reference architecture for AI–IoT integration and a framework of five synergy mechanisms, sense, predict, optimise, automate and learn, applied across five operational domains: predictive maintenance, supply chain and logistics, energy management, security and safety, and quality and customer operations. Second, it evaluates the synergy claim quantitatively in the domain where it is most often made, predictive maintenance, using a simulation whose data-generating process is fully disclosed. A synthetic fleet with latent degradation, realistic sensor behaviour and costed maintenance actions makes it possible to compare, on identical failure histories, a reactive policy, a time-based policy, condition-based rules that use IoT data without learning, a learning model that uses operational records without sensors, a learning model that uses both, and an oracle with perfect foresight. The decomposition of the resulting cost savings into an IoT effect, an AI effect and an interaction term is a direct measurement of synergy within the model. Third, it connects the operational results to business value through a five-year value model and to architecture through an analytical model of inference placement at the edge or in the cloud, and it draws out an implementation roadmap.

Four research questions structure the study. RQ1: what are the mechanisms through which AI and IoT create value in business operations, and how do they map to operational domains and key performance indicators? RQ2: how accurately can machine-learning models forecast equipment failure from IoT sensor streams relative to single-sensor rules and to models without sensors, and what lead time do they provide? RQ3: how do maintenance policies that combine AI and IoT compare with reactive, scheduled, rule-based and AI-only policies on failures, availability and cost, and how large is the interaction between AI and IoT? RQ4: under what conditions of cost, fleet size and architecture does the integration pay back, and what does this imply for implementation?

The approach has an obvious limitation that is stated at the outset and returned to throughout: all data are synthetic. The simulated fleet does not represent any real plant, and the effect sizes depend on assumptions about degradation, sensing and cost that are listed in full and varied in sensitivity analyses. The compensating advantages are that the ground truth is known, that every baseline can be run on identical disruption paths, and that the pipeline can be reproduced and modified. The article

does not reuse percentage claims from the practitioner literature as findings; where such claims are mentioned, they are identified as claims.

Background and Related Work

From connected devices to industrial AI

The IoT was first defined as a network of uniquely identifiable objects that sense and interact with their environment, and early surveys set out its enabling technologies, from identification and sensing to middleware and communication protocols (Atzori et al., 2010; Gubbi et al., 2013; Al-Fuqaha et al., 2015). Its industrial application developed in parallel: a survey of IoT in industries catalogued applications in manufacturing, logistics, healthcare and energy (Xu et al., 2014); the Industrial Internet of Things was analysed as a framework of connected assets, data flows and decision loops (Boyes et al., 2018); and reviews of its challenges emphasised interoperability, latency, security and the management of data at scale (Sisinni et al., 2018; Karkouch et al., 2016). Industry 4.0, the German strategic initiative that gave the field its most widely used label, framed the goal as cyber-physical production systems in which physical processes are mirrored and controlled digitally (Kagermann et al., 2013; Lasi et al., 2014; Lu, 2017; Wang et al., 2016b). The five-level architecture for cyber-physical systems, from smart connection through data-to-information conversion and the cyber level to cognition and configuration, remains the clearest statement of how sensing becomes decision (Lee et al., 2015), and the digital twin, a continuously updated virtual model of a physical asset or process, is its most ambitious expression (Tao et al., 2019; Kritzinger et al., 2018; Fuller et al., 2020; Qi & Tao, 2018).

Industrial AI is the application of machine learning to these systems. Reviews of intelligent manufacturing and of machine learning in manufacturing describe applications in quality, process control, scheduling and maintenance, and the data, skill and integration problems that limit them (Zhong et al., 2017; Wuest et al., 2016; Lee, Davari, et al., 2018; Lee, 2020). Deep learning for IoT data and streaming analytics has been surveyed (Mohammadi et al., 2018; LeCun et al., 2015), and deep learning for time-series classification and machine health monitoring has its own reviews (Ismail Fawaz et al., 2019; Zhao et al., 2019). The business literature frames the same developments from the firm's side: smart, connected products change competition and company organisation (Porter & Heppelmann, 2014; Porter & Heppelmann, 2015), AI projects succeed when they target specific operational processes rather than "moonshots" (Davenport & Ronanki, 2018), and the productivity effects of general-purpose technologies arrive with a lag while complementary intangible investments are made (Brynjolfsson et al., 2021). Surveys of Industry 4.0 adoption link expected and realised performance benefits to specific technologies, with IoT, cloud and analytics among the most valued (Dalenogare et al., 2018; Frank et al., 2019; Tortorella & Fettermann, 2018), and reviews stress that small and medium-sized enterprises face distinct maturity and resource constraints (Mittal et al., 2018; Moeuf et al., 2018; Ghobakhloo, 2018).

Edge, fog and cloud architectures

Where the computation happens is an architectural decision with operational consequences. Cloud computing supplies elastic storage and compute for IoT data, and the integration of cloud and IoT has been surveyed (Botta et al., 2016), but moving all raw sensor data to a remote data centre costs bandwidth and adds latency. Fog and edge computing place processing near the devices (Bonomi et al., 2012; Shi et al., 2016; Satyanarayanan, 2017; Yousefpour et al., 2019). Multi-access edge computing within 5G networks and the communication perspective on mobile edge computing extend the idea to telecommunications infrastructure (Taleb et al., 2017; Mao et al., 2017). Deep learning at the edge, including model compression and partitioning between device, edge and cloud, has been reviewed (Li et al., 2018; Chen & Ran, 2019; Zhou et al., 2019; Zhang et al., 2019a), and federated learning allows models to be trained across devices without centralising data (Nguyen et al., 2021). Edge architectures bring their own security challenges (Roman et al., 2018). For business operations the relevant trade-off is between the latency and bandwidth advantages of local inference and the scale, manageability and model quality of centralised training; this article treats that trade-off with a simple queueing model (Kleinrock, 1975).

Maintenance strategies and predictive maintenance

Maintenance strategy has a long operations-management literature. Reactive (run-to-failure), preventive (time- or usage-based) and predictive (condition-based) strategies differ in when intervention occurs and in the information they use (Swanson, 2001; Tsang, 2002; Mobley, 2002).

Maintenance optimisation models determine inspection and replacement intervals for deteriorating systems (Wang, 2002; Dekker, 1996; de Jonge & Scarf, 2020; Alaswad & Xiang, 2017), and total productive maintenance connects maintenance to overall equipment effectiveness, which combines availability, performance and quality (Nakajima, 1988; Muchiri & Pintelon, 2008). The empirical link between maintenance strategy and productivity has been documented (Swanson, 2001; Alsyouf, 2007). Condition-based maintenance rests on diagnostics and prognostics: detecting that a component has begun to degrade and estimating its remaining useful life (Jardine et al., 2006; Heng et al., 2009; Peng et al., 2010; Vachtsevanos et al., 2006). Statistical prognostics model degradation as a stochastic process, for example a Wiener process with drift, and derive residual-life distributions from sensor signals (Si, Wang, Hu, & Zhou, 2011; Si, Wang, Hu, Chen, & Zhou, 2013; Gebraeel et al., 2005); such models also inform replacement and spare-parts decisions (Elwany & Gebraeel, 2008). Data-driven prognostics with machine learning has grown rapidly, from classifier-based approaches on semiconductor equipment (Susto et al., 2015) to deep networks on vibration and run-to-failure data (Zhao et al., 2019; Lei et al., 2018); the C-MAPSS turbofan simulation and the AI4I 2020 dataset are widely used benchmarks and, notably, both are synthetic (Saxena et al., 2008; Matzka, 2020). Systematic reviews of machine learning for predictive maintenance report that random forests, gradient boosting, support vector machines and neural networks dominate, that most studies evaluate accuracy rather than maintenance outcomes, and that data availability and labelling are the chief obstacles (Carvalho et al., 2019; Zonta et al., 2020; Zhang et al., 2019b; Zhu et al., 2025). Reviews of IoT-enabled prognostics and of the challenges of predictive maintenance in Industry 4.0 add sensor cost, data quality, integration with maintenance management systems, and the difficulty of turning predictions into decisions (Kwon et al., 2016; Compare et al., 2020; Nunes et al., 2023; Bousdekis et al., 2018). Prognostic-based decision support for multi-component systems addresses the scheduling problem that follows a prediction (Van Horenbeek & Pintelon, 2013; Nguyen et al., 2015). Anomaly detection, a related task that does not require failure labels, has general and IoT-specific surveys (Chandola et al., 2009; Cook et al., 2020).

AI and IoT in other operational domains

Beyond maintenance, the same mechanisms appear in other functions. In supply chains, IoT supplies asset tracking, condition monitoring of goods in transit and real-time inventory visibility, and reviews map these to inventory, routing and sustainability outcomes (Ben-Daya et al., 2019; Manavalan & Jayakrishna, 2019; de Vass et al., 2021); big-data analytics in supply-chain management has been reviewed from several angles (Nguyen et al., 2018; Wang et al., 2016a; Tiwari et al., 2018; Govindan et al., 2018; Choi, Wallace, & Wang, 2018), and digital twins and simulation support resilience planning (Ivanov & Dolgui, 2021; Ivanov et al., 2019; Ivanov & Dolgui, 2022). In energy management, sub-metering and building-management systems feed load forecasting and set-point optimisation (Shrouf et al., 2014; Minoli et al., 2017; Ahmad et al., 2021). In security and safety, device logs and network flows feed anomaly and intrusion detection (Sicari et al., 2015; Alaba et al., 2017; Hussain et al., 2020; Meidan et al., 2018; Alsaedi et al., 2020), and IoT raises privacy and legal questions of its own (Weber, 2010). In quality and customer operations, inline inspection and product telemetry feed defect prediction and proactive service (Zhong et al., 2017; Porter & Heppelmann, 2015). Across domains, the business-analytics literature links analytics capability to performance and resilience, while cautioning that the mechanisms are seldom isolated (Gunasekaran et al., 2017; Fosso Wamba et al., 2017; Dubey et al., 2020; Papadopoulos et al., 2017; Belhadi et al., 2021; Mikalef et al., 2019), and the sustainability literature links Industry 4.0 technologies to resource efficiency (Kamble et al., 2018).

Evaluating predictive models and policies

Evaluating a failure forecaster and the policy built on it raises methodological issues that this study follows the literature in addressing. Models must be evaluated out of time, because random cross-validation on temporally dependent data leaks the future (Bergmeir & Benítez, 2012; Cerqueira et al., 2020; Kaufman et al., 2012). Failures are rare, so the area under the precision–recall curve and average precision complement the area under the ROC curve (Fawcett, 2006; Davis & Goadrich, 2006; Saito & Rehmsmeier, 2015; He & Garcia, 2009), and the probabilities fed to a decision rule should be calibrated (Platt, 2000; Guo et al., 2017; Naeini et al., 2015; Brier, 1950). Gradient-boosted trees and random forests are the strongest general learners on tabular features of this kind (Ke et al., 2017; Chen & Guestrin, 2016; Breiman, 2001; Friedman, 2001), and tree-based attribution with SHAP values explains what the model

uses (Lundberg & Lee, 2017; Lundberg et al., 2020). Model performance drifts as equipment, sensors and operating conditions change, so monitoring and retraining are part of deployment (Gama et al., 2014; Lu et al., 2019; Sculley et al., 2015; Breck et al., 2017; Mitchell et al., 2019). For policy comparison, simulation with common random numbers and replication follows standard practice (Law, 2015; Sargent, 2013), and synthetic data generated by a disclosed model allows methods to be compared under a known ground truth (Patki et al., 2016). Finally, the prescriptive-analytics literature argues that forecasts should be judged by the decisions they improve rather than by accuracy alone (Bertsimas & Kallus, 2020; Mišić & Perakis, 2020), which is the stance this article takes.

Research gap and framework

The reviewed literature supplies architectures, algorithms, maintenance models and value propositions, but rarely puts them together in one evaluated study with transparent baselines. In particular, the synergy between AI and IoT is asserted more often than measured: studies of predictive maintenance seldom report what a simple condition-based rule on the same sensors would have achieved, or what a model without sensors could do with maintenance records alone, and the practitioner literature reports improvement percentages without the baseline policy, the cost assumptions or the data. Figure 1 presents the reference architecture that this article uses to organise the integration: physical assets and sensing, connectivity and edge, a cloud data platform with a digital twin, AI services, and business operations, with data flowing upward and decisions flowing downward under a security and governance band. Figure 2 presents the synergy framework: five mechanisms, sense, predict, optimise, automate and learn, forming a closed loop, applied to five operational domains with their typical performance indicators. The predictive-maintenance row is the one evaluated quantitatively; the others are developed in the Discussion.

Figure 1: AI–IoT reference architecture for business operations: five tiers with upward data flows, downward decision flows and a security and governance band (author's synthesis)

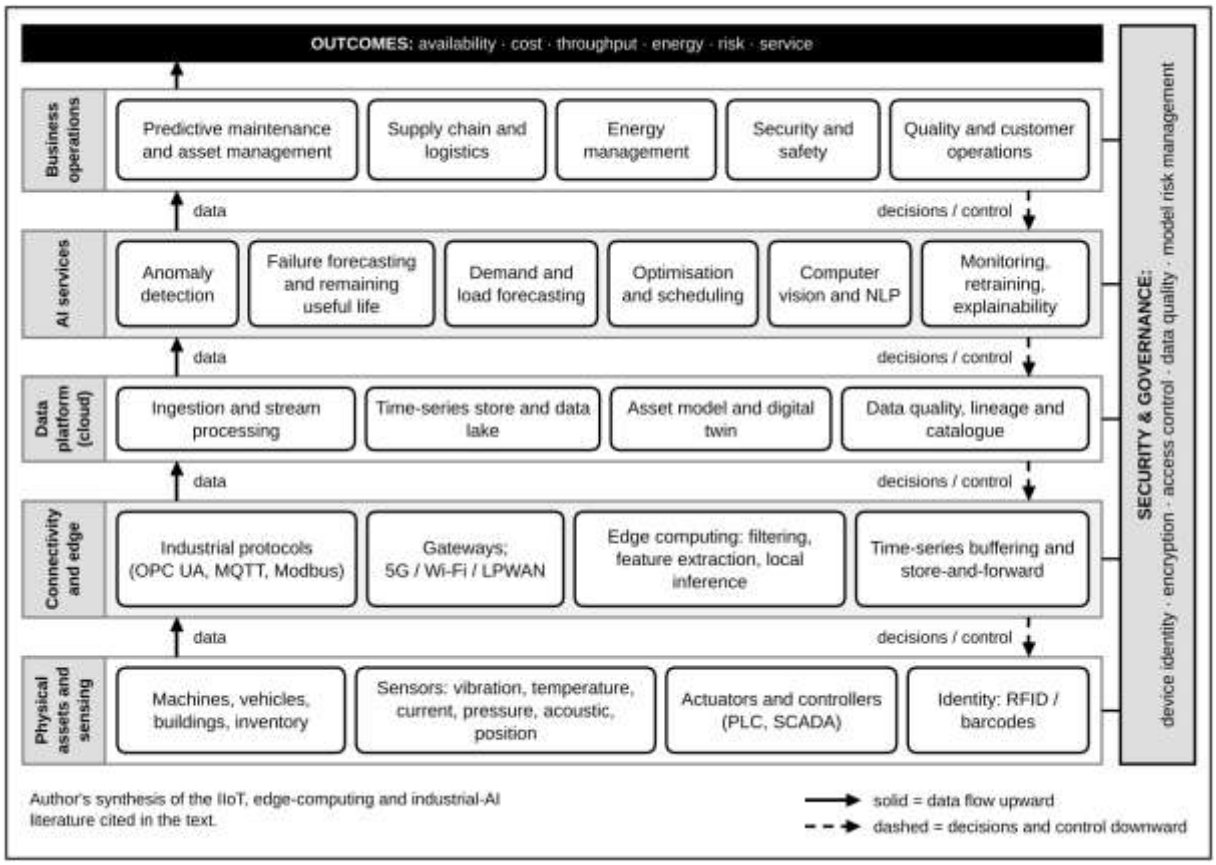
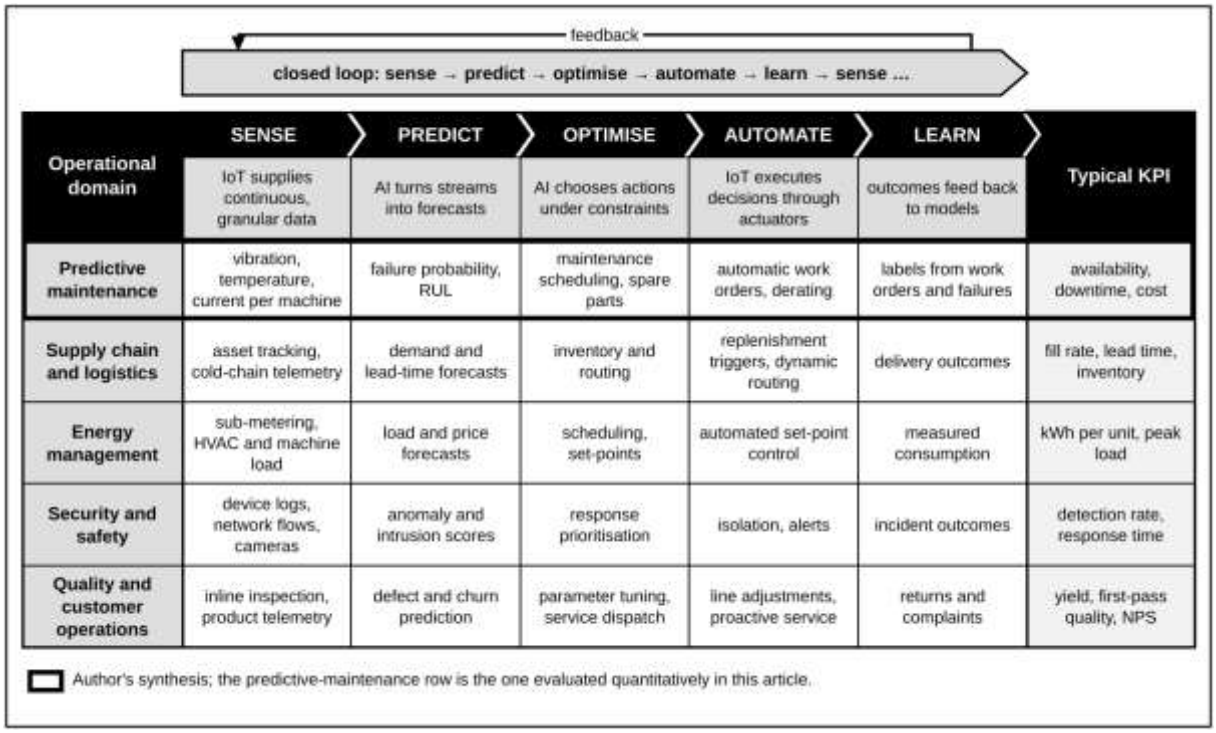


Figure 2: Synergy mechanisms between AI and IoT across five operational domains, forming a closed sense–predict–optimise–automate–learn loop (author's synthesis)



Operational domain	SENSE	PREDICT	OPTIMISE	AUTOMATE	LEARN	Typical KPI
	IoT supplies continuous, granular data	AI turns streams into forecasts	AI chooses actions under constraints	IoT executes decisions through actuators	outcomes feed back to models	
Predictive maintenance	vibration, temperature, current per machine	failure probability, RUL	maintenance scheduling, spare parts	automatic work orders, derating	labels from work orders and failures	availability, downtime, cost
Supply chain and logistics	asset tracking, cold-chain telemetry	demand and lead-time forecasts	inventory and routing	replenishment triggers, dynamic routing	delivery outcomes	fill rate, lead time, inventory
Energy management	sub-metering, HVAC and machine load	load and price forecasts	scheduling, set-points	automated set-point control	measured consumption	kWh per unit, peak load
Security and safety	device logs, network flows, cameras	anomaly and intrusion scores	response prioritisation	isolation, alerts	incident outcomes	detection rate, response time
Quality and customer operations	inline inspection, product telemetry	defect and churn prediction	parameter tuning, service dispatch	line adjustments, proactive service	returns and complaints	yield, first-pass quality, NPS

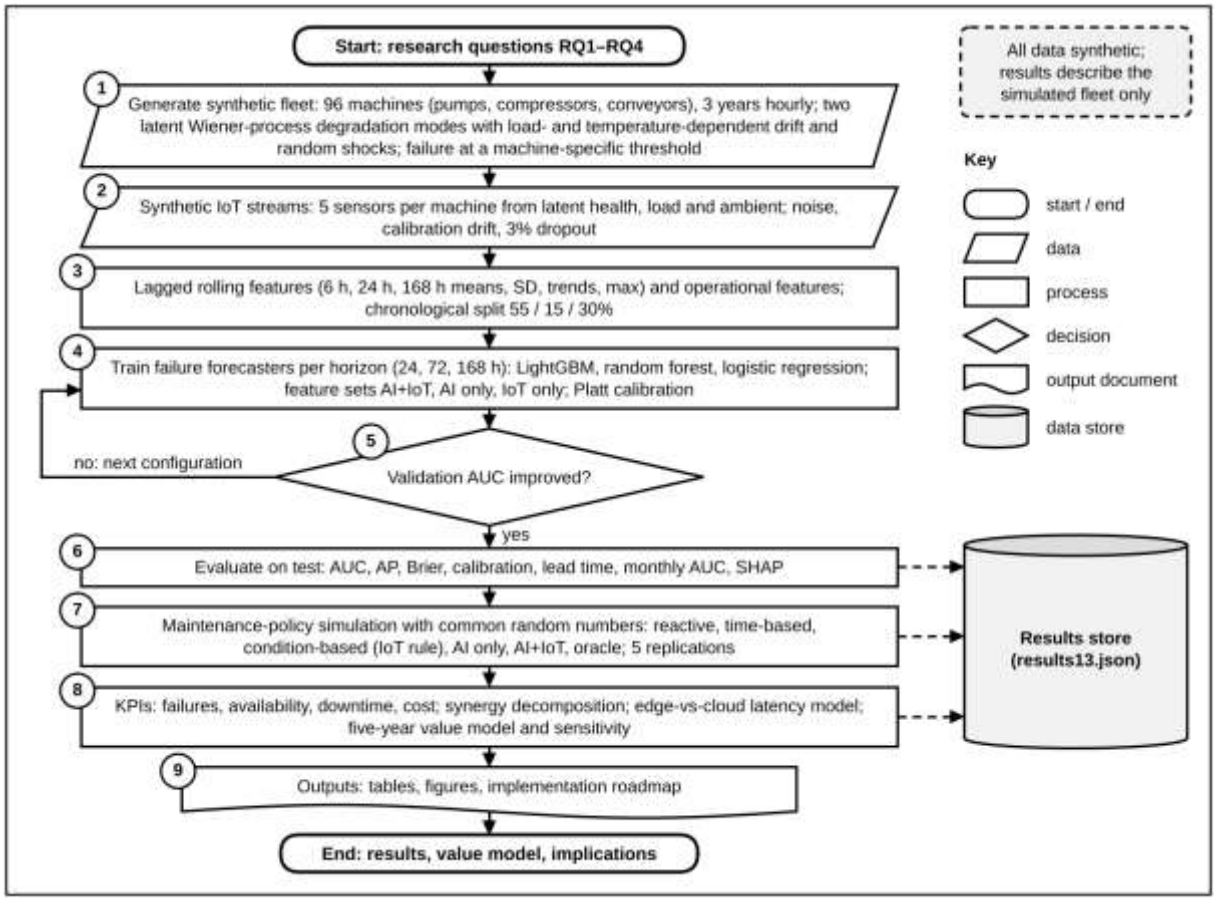
Author's synthesis; the predictive-maintenance row is the one evaluated quantitatively in this article.

Methods

Study design

The study is a simulation-based evaluation in five parts (Figure 3): generation of a synthetic industrial fleet with latent degradation and IoT sensing; failure forecasting on lagged sensor and operational features; a maintenance-policy simulation in which six policies act on identical degradation paths; an analytical edge-versus-cloud placement model; and a five-year value model with sensitivity analysis. The design follows the logic of digital-twin experiments in operations research (Ivanov & Dolgui, 2021; Tao et al., 2019) and the conventions of simulation studies, including replication, common random numbers and separate verification of each component (Law, 2015; Sargent, 2013). Five replications with seeds 2030 to 2034 were run; the first is used for detailed diagnostics and sweeps, and all policy comparisons are reported across the five. The pipeline is implemented in Python with NumPy, pandas, scikit-learn, LightGBM and SHAP (Harris et al., 2020; Pedregosa et al., 2011; Ke et al., 2017; Lundberg et al., 2020). All data are synthetic; no measurements from any real plant were used.

Figure 3: Study design as a flowchart: synthetic fleet, sensor streams, features, forecasting, policy simulation, KPIs and value model



The synthetic fleet and its degradation process

Table 1 lists the fleet and degradation parameters. The fleet has 96 machines in three types, pumps, compressors and conveyors, observed hourly for three years (26,280 hours). Each machine carries two latent degradation modes, representing a wear-type mechanism (mode A, such as bearing wear) and an electrical or lubrication mechanism (mode B). Each mode follows a Wiener process with drift, the standard statistical model of degradation signals (Si, Wang, Hu, & Zhou, 2011; Si, Wang, Hu, Chen, & Zhou, 2013; Gebraeel et al., 2005): the hourly increment is a type-specific base rate, scaled by a load factor and by an ambient-temperature factor, plus Gaussian noise whose variance scales with the drift, plus occasional shocks that add 5–15% of the failure distance at once. Mode B's drift also rises with the level of mode A, so that the two mechanisms interact. Base rates differ across machine types and across machines within a type, which gives nominal lives of the order of a year at typical load. Each machine has its own failure threshold drawn between 0.85 and 1.15 on the health scale, so that the sensor level at which it fails differs across the fleet; failure occurs when either mode reaches the threshold. After a failure or a planned intervention the modes are reset to a small residual level, which represents imperfect repair.

Load follows shift and weekend patterns, machine-specific production campaigns and random variation; ambient temperature follows a yearly and a daily cycle, and hot conditions accelerate degradation. These choices reproduce three features that make prognostics hard in practice and that distinguish the problem from benchmark datasets (Nunes et al., 2023; Compare et al., 2020): heterogeneous and partly hidden failure mechanisms, machine-specific baselines, and confounding of sensor signals by operating conditions.

Table 1: Fleet, load and degradation parameters of the synthetic testbed

Element	Specification	Value	Rationale / basis
Fleet	Machines by type	96: 32 pumps, 32 compressors, 32 conveyors	Three asset classes with different wear rates
Horizon	Hourly steps over three years; chronological split	26,280 h: fit 0–55%, validation 55–70%, test 70–100% (≈ 11 months)	Long enough for dozens of failures per window
Degradation	Two latent modes per machine (A: wear-type; B: electrical/lubrication), Wiener processes with drift	Base hourly drift A: 1/9,000, 1/7,000, 1/12,000 by type (× log-normal machine factor, σ = 0.3); B: 1/16,000, 1/11,000, 1/20,000	Statistical degradation models (Si et al., 2011, 2013); nominal lives of the order of a year
Drift modifiers	Load factor and ambient factor	A: (0.6 + 0.8 ·load^1.5); B: (0.7 + 0.6 ·load) ·(1 + 0.5 ·h_A); both × (1 + 0.3 ·max(0, (ambient − 28)/10))	Wear accelerates with duty and heat; mode B accelerates as mode A advances
Noise and shocks	Gaussian increments; random shocks	σ = 0.012 / 0.016 / 0.010 by type, scaled by √(drift/base); shock probability per hour 1/9,000, 1/6,000, 1/14,000 × (0.5 + load); shock size U(0.08, 0.25), 60% to A, 40% to B	Shocks create partly unpredictable failures
Failure	Threshold on either mode; reset after events	Threshold ~ U(0.85, 1.15) per machine; reset to U(0, 0.05) after failure, U(0, 0.08) after intervention	Machine-specific failure levels; imperfect repair
Load	Shift, weekend, campaign and random components	Day shift 1.0 / night 0.55; weekend × 0.7; campaign ± 25% with 20–60-day period; machine factor U(0.8, 1.15); noise σ = 0.05; clipped to [0.2, 1.4]	Confounds sensor signals with duty
Ambient	Yearly and daily temperature cycles	20 + 8 ·sin(yearly) + 4 ·sin(daily) °C	Seasonal acceleration of wear
Replications	Independent fleets	5 seeds (2030–2034); policies compared on identical paths within a replication	Common random numbers

IoT sensing model

Five sensor streams per machine are derived from the latent state (Table 2): vibration, temperature, motor current, pressure and acoustic level. Each is a machine-specific function of the two modes, load and ambient temperature with additive noise. Vibration responds mainly to mode A through a quadratic term and rises strongly with load, so that a heavy campaign can look like degradation; its baseline and sensitivity vary across machines by factors of up to 2.5 and 1.9. Temperature responds to both modes and to ambient conditions. Current carries mode B through its square and through intermittent spikes that appear only when mode B is advanced, which makes the electrical mechanism visible mainly in current signatures. Pressure falls with mode A and acoustic level rises with both. Vibration and temperature carry slow random-walk calibration drift, and 3% of all readings are missing at random, representing the data-quality problems documented for IoT deployments (Karkouch et al., 2016; Sisinni et al., 2018). During downtime no readings are produced. The sensing model is deliberately imperfect: no single sensor is a sufficient statistic for failure risk, which is the condition

under which multi-sensor learning can add value over a threshold rule.

Table 2: IoT sensing model: sensor equations, machine-specific characteristics and data-quality effects

Sensor	Equation (per machine m, hour t)	Machine-specific terms	What it reveals
Vibration (vib)	$b_vib + 0.9 \cdot load + k_vib \cdot (2.2 \cdot h_A^2 + 0.4 \cdot h_A) + 0.15 \cdot h_B + N(0, 0.30^2)$ + drift	$b_vib \sim U(0.8, 2.0)$; $k_vib \sim U(0.7, 1.3)$	Mode A, non-linearly; strongly confounded by load; baseline differs across machines
Temperature (°C)	$ambient + 16 \cdot load + k_t \cdot (9 \cdot h_A + 7 \cdot h_B) + N(0, 1.5^2) + 3 \cdot drift$	$k_t \sim U(0.8, 1.2)$	Both modes; confounded by ambient and load
Motor current	$b_cur \cdot load \cdot (1 + 0.15 \cdot h_A + 0.45 \cdot h_B^2) + N(0, 0.6^2)$ + spikes	$b_cur = (9 + 1.5 \cdot type) \cdot U(0.9, 1.1)$; spikes of 2–5 units with probability 0.10 per hour when $h_B > 0.6$	Mode B, mainly through spikes late in the process
Pressure	$b_pres - 0.9 \cdot h_A + 0.3 \cdot load + N(0, 0.30^2)$	$b_pres \sim U(4.6, 5.4)$	Mode A, weakly
Acoustic level	$b_ac + 6 \cdot load + 8 \cdot h_A\text{^}1.5 + 3 \cdot h_B + N(0, 2.5^2)$	$b_ac \sim U(56, 64)$	Both modes; noisy
Data quality	Random-walk calibration drift on vibration and temperature ($\sigma = 0.0005$ per hour, cumulative); 3% of readings missing at random; no readings during downtime	—	Typical IoT data-quality defects (Karkouch et al., 2016)

Features and labels

Every six hours, for every machine, 33 features were computed from the previous 168 hours of readings only (Table 3): for each sensor the 6-hour and 24-hour means, the 24-hour standard deviation, the 168-hour mean and the trend (24-hour minus 168-hour mean), plus the 24-hour maximum of vibration and current and the ratio of 24-hour to 168-hour vibration; and five operational features available without sensors, the 24-hour mean load, ambient temperature, hours since the last failure or intervention, machine type and cumulative load since the last event. The strict use of past readings follows the leakage guidance in the evaluation literature (Kaufman et al., 2012; Bergmeir & Benítez, 2012). Labels are binary indicators of a failure within the next 24, 72 or 168 hours. Three feature sets define the comparison at the heart of the study: AI+IoT uses all 33 features; AI only uses the five operational features, representing a model trained on maintenance and production records without sensors; and sensor features only uses the 28 sensor features without operational context. In addition, three single-sensor rules that require no learning, the 24-hour vibration mean, the vibration ratio and the 24-hour temperature mean, are scored as classifiers to represent the IoT-without-AI alternative.

Table 3: Feature dictionary and labels

Group	Features (computed every 6 h from the previous 168 h of readings only)	Count	Feature set membership
Sensor level and spread	For each of the five sensors: 6-h mean, 24-h mean, 24-h standard deviation, 168-h mean	20	AI+IoT; sensors only
Sensor trend	For each sensor: 24-h mean minus 168-h mean	5	AI+IoT; sensors only
Sensor extremes and ratio	24-h maximum of vibration and of current; ratio of 24-h to 168-h vibration mean	3	AI+IoT; sensors only
Operational (no sensors)	24-h mean load; ambient temperature; hours since last failure or intervention (capped at 400 days); machine type; cumulative load since last event	5	AI+IoT; AI only
Labels	Failure within the next 24, 72 or 168 hours (binary)	3	–
Single-sensor rules (no learning)	24-h vibration mean; vibration ratio; 24-h temperature mean, each scored directly as a classifier	3	IoT only

Forecasting models and evaluation

The panel was split chronologically into fit (first 55% of hours), validation (next 15%) and test (last 30%, about 11 months) windows. The primary model is LightGBM with 800 trees at most, learning rate 0.03, 31 leaves, a minimum of 200 observations per leaf, feature and row subsampling of 0.8 and early stopping on validation AUC (Ke et al., 2017; Friedman, 2001); a random forest and an L2-regularised logistic regression on standardised features serve as benchmarks for the AI+IoT feature set (Breiman, 2001; Pedregosa et al., 2011).

Table 4: Forecasting models, feature sets and evaluation protocol

Element	Setting
Panel	417,792 machine-6-hour rows in replication 1; label prevalence 0.54% (24 h), 1.61% (72 h), 3.71% (168 h) in the test window
Split	Chronological: fit hours 0–14,453; validation 14,454–18,395; test 18,396–26,279 (no shuffling)
Primary model	LightGBM: up to 800 trees, learning rate 0.03, 31 leaves, min 200 rows per leaf, row and column subsampling 0.8, L2 = 2; early stopping (60 rounds) on validation AUC; one model per horizon and feature set
Benchmarks	Random forest (120 trees, min 50 rows per leaf, √p features); logistic regression (L2, C = 1) on standardised features; three single-sensor rules scored as classifiers
Calibration	Platt scaling on the validation window; calibrated probabilities used for policies, Brier score and reliability curve
Discrimination metrics	AUC and average precision on the test window; AUC by machine type; monthly AUC without retraining
Decision diagnostics	Lead time: hours from the first exceedance of a probability threshold (0.05–0.5) within 168 h before a failure to the failure, for all 171 test failures of replication 1; detection rate at each threshold, by failure mode
Explanation	TreeSHAP global importance (mean |SHAP|) of the 72-h AI+IoT model on 3,000 test rows
Policy models	The 72-h AI+IoT and AI-only LightGBM models drive the AI+IoT and AI-only policies; thresholds tuned on the validation window

Scores were calibrated by Platt scaling on the validation window (Platt, 2000; Guo et al., 2017), and only calibrated probabilities were used by the maintenance policies. Discrimination is reported as AUC and average precision, calibration as the Brier score and a reliability curve (Fawcett, 2006; Davis & Goadrich, 2006; Saito & Rehmsmeier, 2015; Brier, 1950; Naeini et al., 2015). Two diagnostics connect accuracy to decisions: lead time, the number of hours between the first time the calibrated 72-hour probability exceeded a threshold and the subsequent failure, computed for all 171 test failures of the first replication within a 168-hour window and separately for the two failure modes; and monthly AUC through the test window without retraining, as a drift check (Gama et al., 2014). SHAP values of the 72-hour model identify which signals carry the forecast (Lundberg & Lee, 2017; Lundberg et al., 2020), and AUC by machine type and by failure mode shows where it is weakest. Table 4 summarises the models and the evaluation protocol.

Maintenance-policy simulation

The policy simulation (Table 5) re-runs the degradation and sensing process through the test window with common random numbers, so that every policy faces the same shocks, the same noise and the same latent paths; it starts from a burn-in under reactive maintenance that is identical across policies. Six policies were compared. REACTIVE repairs after failure. TIME intervenes at a fixed interval since the last event. CBM triggers an intervention when the 24-hour vibration mean exceeds a fleet-wide threshold, and CBM-REL when it exceeds a multiple of the machine's own post-repair baseline; both are IoT rules without learning. AI-ONLY triggers when the calibrated probability of failure within 72 hours from the no-sensor model exceeds a threshold, and AI+IoT does the same with the full model; both evaluate every six hours. ORACLE intervenes when a latent mode comes within 0.05 of its threshold, an upper bound that no observable policy can reach. Triggered interventions are scheduled within a planning window of up to 24 hours, which represents the time needed to arrange a technician and parts; a failure during the window is counted as a failure. An intervention on a machine whose worse mode is below half its threshold is counted as a false alarm, that is, an unnecessary intervention. The cost model (Table 6) uses an unplanned repair cost of 12,000 and 36 hours of downtime per failure, a planned intervention cost of 3,500 and 8 hours of downtime, and lost contribution of 450 per downtime hour; these are stylised magnitudes in a common currency unit, chosen so that an unplanned failure costs roughly four to five times a planned intervention, which is the order reported in the maintenance-management literature (Mobley, 2002; Swanson, 2001), and they are varied in the value sensitivity analysis. Each policy family's operating parameter, the interval, the vibration threshold, the ratio or the probability threshold, was tuned on the validation window by minimising cost per machine-year, and the tuned policy was then evaluated on the test window; this prevents a policy from being tuned on the period on which it is judged. Because the tuned probability thresholds of the two AI policies lay at the lower edge of the grid first searched, the grid was extended downwards to 0.005 in the first replication and the optimum re-examined. For the first replication the full parameter sweeps were also run on the test window to draw cost–availability frontiers, and the planning window was varied between 6 and 96 hours. Key performance indicators are failures, planned interventions, false alarms, downtime hours, availability, unplanned and planned downtime, cost and its components, and failures and interventions per machine-year.

Table 5: Maintenance policies compared in the simulation

Policy	Information used	Trigger rule (evaluated every 6 h unless stated)	Tuned parameter (validation window)	Represents
REACTIVE	None	Repair after failure only	—	Run-to-failure
TIME	Hours since last event	Intervene when hours since last failure or intervention ≥ interval	Interval ∈ {500, 750, 1,000, 1,500, 2,000, 3,000} h	Preventive maintenance without sensors
CBM	Vibration sensor	Intervene when the 24-h vibration mean exceeds a fleet-wide level	Level ∈ {2.6, 3.0, 3.4, 3.8, 4.2, 4.6}	IoT monitoring with a conventional alarm threshold
CBM-REL	Vibration sensor and post-repair baseline	Intervene when the 24-h vibration mean exceeds a multiple of the machine's baseline (mean of hours 24–96 after the last event)	Multiple ∈ {1.2, 1.3, 1.4, 1.5, 1.7, 2.0}	IoT monitoring with a per-machine adaptive threshold
AI-ONLY	Operational records (load, ambient, hours since event, type, cumulative load)	Intervene when the calibrated 72-h failure probability from the no-sensor model exceeds a threshold	Threshold ∈ {0.02, 0.03, 0.05, 0.08, 0.12}; extended to 0.005–0.015 in replication 1	Machine learning on maintenance and production records without IoT
AI+IoT	All 33 features	Intervene when the calibrated 72-h failure probability from the full model exceeds a threshold	Threshold ∈ {0.03, 0.05, 0.08, 0.12, 0.18, 0.25, 0.35, 0.5}; extended to 0.005–0.02 in replication 1	Proposed AI–IoT predictive maintenance
ORACLE	Latent health (unobservable)	Intervene when either mode is within 0.05 of its failure threshold	—	Upper bound with perfect foresight
All triggered policies	—	Interventions scheduled uniformly within a planning window of 1–24 h; no new trigger within 48 h of an event; failure during the window counts as a failure	Planning window varied 6–96 h in replication 1	Scheduling realism

Table 6: Cost and time parameters of the maintenance simulation

Item	Value	Note
Unplanned repair cost per failure	12,000	Parts, labour, expediting
Downtime per failure	36 h	Diagnosis, parts lead time, repair
Planned intervention cost	3,500	Scheduled parts and labour
Downtime per planned intervention	8 h	Scheduled stop
Lost contribution per downtime hour	450	Applies to planned and unplanned downtime
Planning window	1–24 h	Time to arrange technician and parts
False alarm	Planned intervention on a machine whose worse mode is below half its threshold	Counted; costed as a planned intervention
Value model (five years)	Sensors, connectivity and installation 2,500 per machine; gateway 4,000 per 40 machines; integration 150,000; platform 600 per machine-year; model operations and monitoring 120,000 per year; discount rate 8%	magnitudes in a common currency unit; varied in the sensitivity analysis together with a failure-intensity scaling of the savings

Synergy decomposition

The synergy claim was tested with a two-by-two decomposition of cost savings. With cost per machine-year under REACTIVE as the reference, the IoT effect is the saving of the better condition-based rule, the AI effect is the saving of AI-ONLY, and the combined effect is the saving of AI+IoT. The interaction, the combined saving minus the sum of the two separate savings, is positive when AI and IoT deliver more together than apart. The decomposition was computed in each replication and summarised across them.

Edge-versus-cloud placement model

Inference placement was analysed with an M/M/1 queueing model (Kleinrock, 1975). Each machine generates one inference request per minute. Three placements were compared: local inference on edge gateways, one gateway per 40 machines as in the value model, each with a service rate of 40 requests per second (25 milliseconds for feature extraction and inference on modest hardware), a 5-millisecond network delay and a fixed cost per gateway-hour; cloud inference on raw streams, with a service rate of 500 requests per second, an 80-millisecond round trip, a charge per request and 2 kilobytes of upstream data per request; and a hybrid in which features are extracted on the gateways at a lower fixed cost and only 50 bytes per request are sent to the cloud for inference. Bandwidth is charged per gigabyte. For fleets of 10 to 1,600 machines the model gives utilisation, mean latency, hourly cost and upstream bandwidth. The parameters are orders of magnitude drawn from the edge-computing literature (Shi et al., 2016; Satyanarayanan, 2017; Chen & Ran, 2019); the purpose is to show the shape of the trade-off, not to price any vendor's service.

Value model

A five-year net-present-value model converts the difference in cost per machine-year between a baseline policy and AI+IoT into fleet savings, against capital costs of 2,500 per machine for sensors, connectivity and installation, 4,000 per gateway for every 40 machines and 150,000 for integration with maintenance and production systems, and operating costs of 600 per machine-year for the platform and 120,000 per year for model operations and monitoring, at an 8% discount rate. Savings are taken from the first replication's tuned policies, with the mean and range over the five replications reported alongside. Sensitivity analyses vary lost-production cost per hour, unplanned repair cost, sensor cost

and fleet size, and compute the value against reactive, time-based and both condition-based baselines separately, because the relevant baseline is the policy a plant runs today. Because the simulated fleet fails more often than many real fleets (about 1.9 failures per machine-year under reactive maintenance), a further sensitivity scales the simulated saving per machine-year by a failure-intensity factor between 0.05 and 1, on the assumption that savings are proportional to the failure intensity of the fleet, and reports the break-even factor at which the net present value is zero for each baseline. The payback period and five-year return on investment are reported alongside the net present value. These magnitudes are assumptions, chosen to be plausible rather than representative of any industry, and the sensitivity analysis is the substantive result.

Statistical analysis and transparency

Differences between AI+IoT and each other policy in cost per machine-year and availability were analysed as paired observations across the five replications with paired t-tests and reported with their ranges; with five replications the tests have little power and are descriptive. Forecasting metrics are reported for the first replication, with the 72-hour AUC across all five. The simulation script and the complete results file accompany the article so that every number can be regenerated and every assumption changed.

Results

The simulated fleet

Under run-to-failure operation the five synthetic fleets produced 524 to 541 failures over three years, 1.82 to 1.88 per machine-year, of which 29 to 34% were mode-B (electrical or lubrication) failures and the rest mode-A (wear-type) failures. Figure 4 shows what the forecaster and the maintenance rules faced in the first replication. The two latent modes of one pump rise at different rates, interact, jump at shocks and reset after each failure or intervention (Figure 4a); the machine's own threshold, 1.06, differs from its neighbours'. Vibration tracks mode A but is pushed up and down by load and shifted by calibration drift, and temperature follows the seasons and the shifts as much as the degradation (Figure 4b); current carries mode B mainly through spikes late in the process (Figure 4c). The consequence for any fleet-wide rule is in Figure 4d: the mean vibration in the 24 hours before failure, across 400 of the 536 failures, ranged from 2.4 to 5.6 between the 5th and 95th percentiles with a median of 4.36, so that a single level that catches the quiet machines before they fail would stop the loud ones several times a year for nothing. This is the condition that gives machine learning on multiple sensors something to add.

Forecasting accuracy (RQ2)

Table 7 reports discrimination and calibration on the test window of the first replication, 126,144 machine-6-hour rows with a 72-hour failure prevalence of 1.61%. The LightGBM model on all 33 features reached an AUC of 0.926 at 24 hours, 0.913 at 72 hours and 0.891 at 168 hours, with average precision of 0.122, 0.165 and 0.247; across the five replications the 72-hour AUC was 0.906 to 0.931. The random forest was within 0.01 of LightGBM at every horizon and slightly ahead at 168 hours (0.895), and the regularised logistic regression was 0.01 to 0.03 lower (0.893 at 72 hours), so the gain comes from the features more than from the learner. The two no-learning rules and the no-sensor model mark out the two halves of the synergy argument. The best single-sensor rule, the 24-hour vibration level, reached an AUC of 0.834 at 72 hours; the relative vibration rule, which compares a machine with its own recent past, was much weaker (0.671), because degradation here is slow relative to the 168-hour reference window and the ratio is dominated by load changes; the temperature rule reached 0.734. The AI-only model, trained on load, ambient temperature, hours since the last event, machine type and cumulative load, reached 0.751 at 72 hours (0.722 to 0.770 across replications): age and duty carry real information about failure risk, but far less than the sensors. The model on sensor features alone reached 0.893, 0.02 below the full model, so the operational features add a small but consistent increment on top of sensing. Figure 5 shows the ROC and precision–recall curves at 72 hours and the decay of AUC with horizon; the ranking of feature sets is the same at every horizon, and the gap between learning on all sensors and the best rule (0.07 to 0.08 AUC) is larger than the gap between learners (0.01 to 0.03)

Figure 4: Degradation and sensing dynamics in the synthetic fleet (replication 1): latent modes, vibration and temperature, current with mode-B spikes, and the spread of vibration at failure across machines

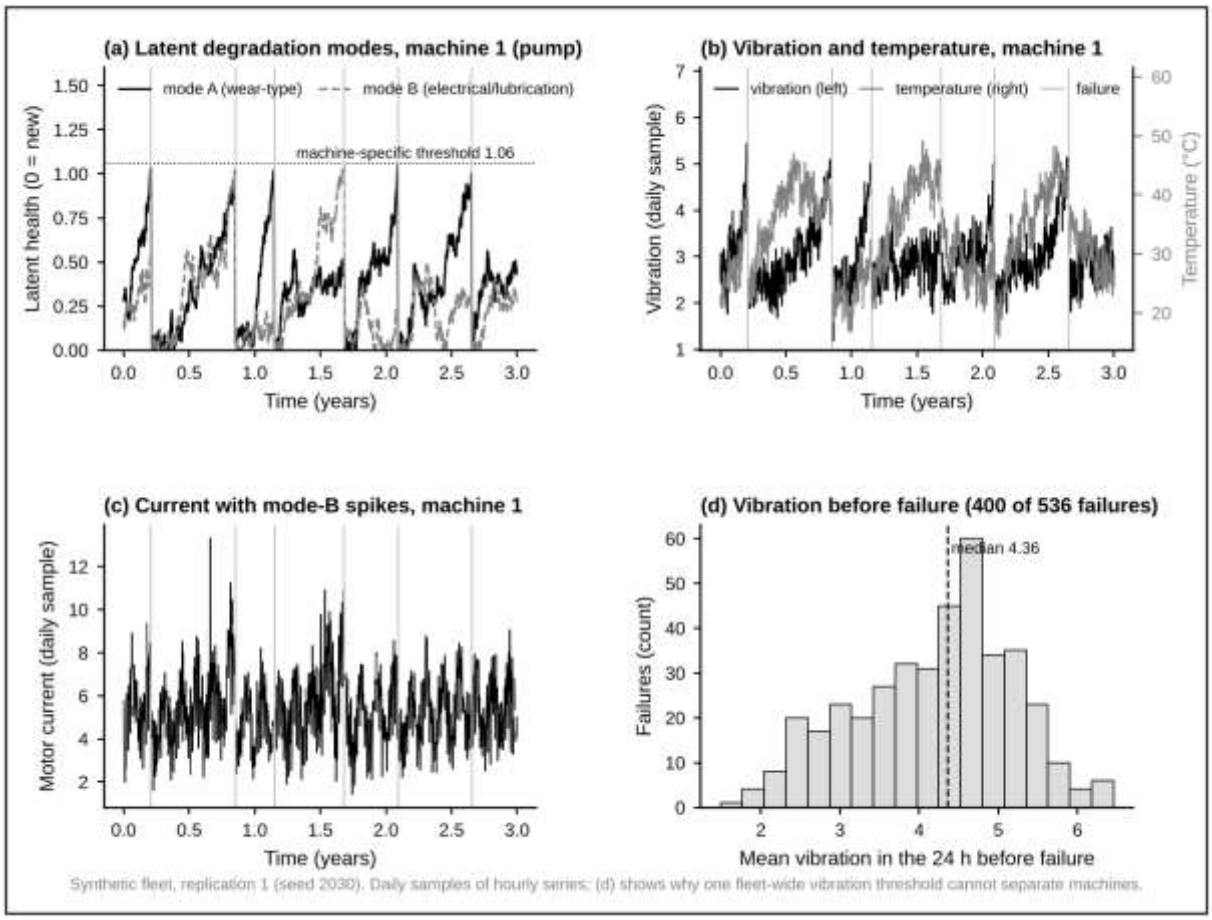
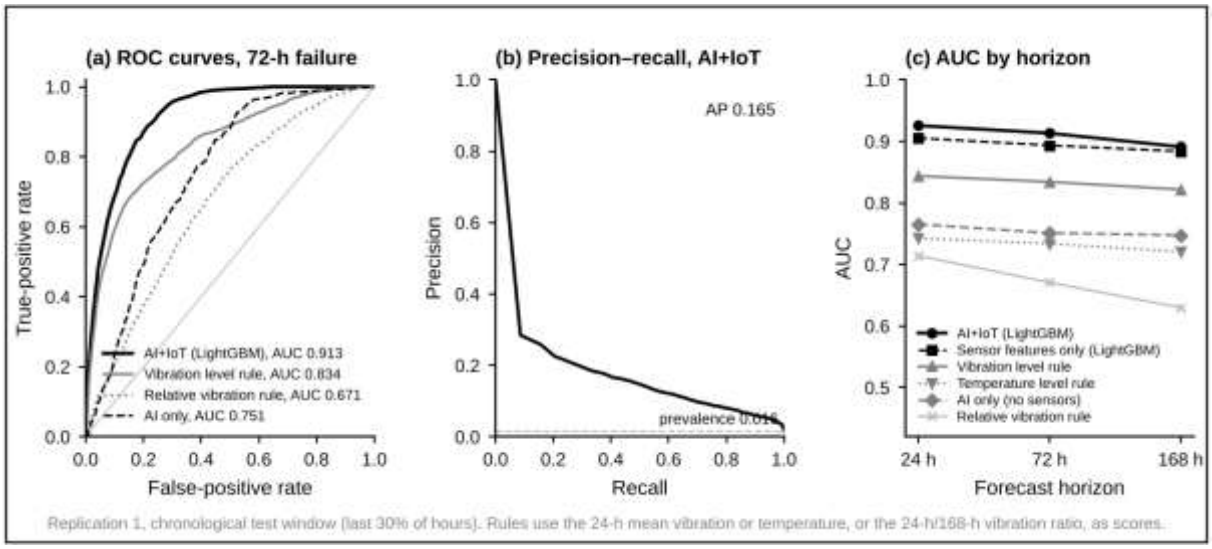


Table 7: Forecasting results on the test window: AUC and average precision by horizon, model and feature set (replication 1, with cross-replication ranges for the 72-hour models)

Model and feature set	24 h ahead (AUC / AP)	72 h ahead	168 h ahead
LightGBM, AI+IoT (33 features)	0.926 / 0.122	0.913 / 0.165	0.891 / 0.247
Random forest, AI+IoT	0.916 / 0.123	0.910 / 0.168	0.895 / 0.270
Logistic regression, AI+IoT	0.901 / 0.058	0.893 / 0.133	0.882 / 0.220
LightGBM, sensor features only (28)	0.905 / 0.063	0.893 / 0.149	0.884 / 0.241
LightGBM, AI only (5 operational features)	0.765 / 0.059	0.751 / 0.034	0.747 / 0.076
Rule: 24-h vibration level	0.844 / 0.050	0.834 / 0.113	0.822 / 0.199
Rule: vibration ratio (24 h / 168 h)	0.714 / 0.012	0.671 / 0.029	0.630 / 0.056
Rule: 24-h temperature level	0.742 / 0.016	0.734 / 0.043	0.721 / 0.089
LightGBM, AI+IoT, 72 h across 5 replications (min–max)	–	0.906–0.931 / 0.165–0.207	–
LightGBM, AI only, 72 h across 5 replications (min–max AUC)	–	0.722–0.770	–
Test prevalence	0.54%	1.61%	3.71%
Brier score, calibrated LightGBM AI+IoT	0.0050	0.0145	0.0315

Figure 5: Discrimination at the 72-hour horizon and AUC by horizon for the AI+IoT model, the sensor-only and no-sensor models and the single-sensor rules (replication 1, test window)



Calibration, drift, lead time and explanation

Calibrated probabilities were reliable across their range (Figure 6a): the decile means of predicted probability tracked the observed 72-hour failure rate from 0.0004 to 0.113, with the top decile predicting 0.113 against an observed 0.107, and the Brier score was 0.0145 against a prevalence of 0.0161. Monthly AUC without retraining stayed between 0.838 and 0.950 over the eleven months of the test window (Figure 6b), with no downward trend, which is expected because the synthetic process is stationary; the dip in month 3 coincided with the lowest monthly prevalence and is sampling variation rather than drift. Real deployments should expect drift and monitor for it, a point taken up in the Discussion.

Lead time is the diagnostic that connects a probability to a maintenance decision, and Figure 6c and 6d report it for all 171 test failures (Table 8). At an alarm threshold of 0.05 on the calibrated 72-hour probability, 138 failures (81%) had an alarm within the preceding 168 hours, with a median lead of 151 hours; at 0.03, the threshold that the policy tuning later selected, 152 failures (89%) were flagged with a median lead of 163 hours; at 0.2 only 60 (35%) were flagged, with a median lead of 100 hours. The thresholds are low because the probability of failure within 72 hours is low even for a machine that is close to its threshold, and because the model is calibrated: a probability of 0.05 is three times the base rate. The failure modes differ sharply. At the 0.03 threshold, 116 of 119 mode-A failures (97%) were flagged against 36 of 52 mode-B failures (69%); at 0.2 the figures were 55 of 119 against 5 of 52. The AUC by failure mode (Figure 7b) tells the same story, 0.948 for mode A and 0.835 for mode B, and AUC by machine type ranged from 0.901 for compressors, which have the highest shock rate, to 0.926 for pumps. Mode B is harder because its sensor signature is weaker and later: it enters vibration only through a small linear term, appears in current mainly as intermittent spikes once the mode is advanced, and shocks, which add to both modes, are most frequent in the compressors.

SHAP importance (Figure 7a) shows how the model combines the two feature groups. The single most influential feature was hours since the last event, an operational feature, followed by the 24-hour acoustic mean, the 24-hour load, the 24-hour vibration mean and the machine type; the 6-hour vibration mean, the 168-hour temperature mean and the 24-hour maximum of current come next. Eight of the twelve most influential features are sensor statistics and four are operational, and the presence of load and type near the top is the model learning to discount sensor levels that are high because of duty or baseline rather than wear, which is exactly what the fixed rule cannot do. Table 8 collects these diagnostics.

Figure 6: Calibration, stability and lead time of the 72-hour AI+IoT model: reliability curve, monthly AUC, lead-time distribution and detection by threshold and failure mode (replication 1, all 171 test failures)

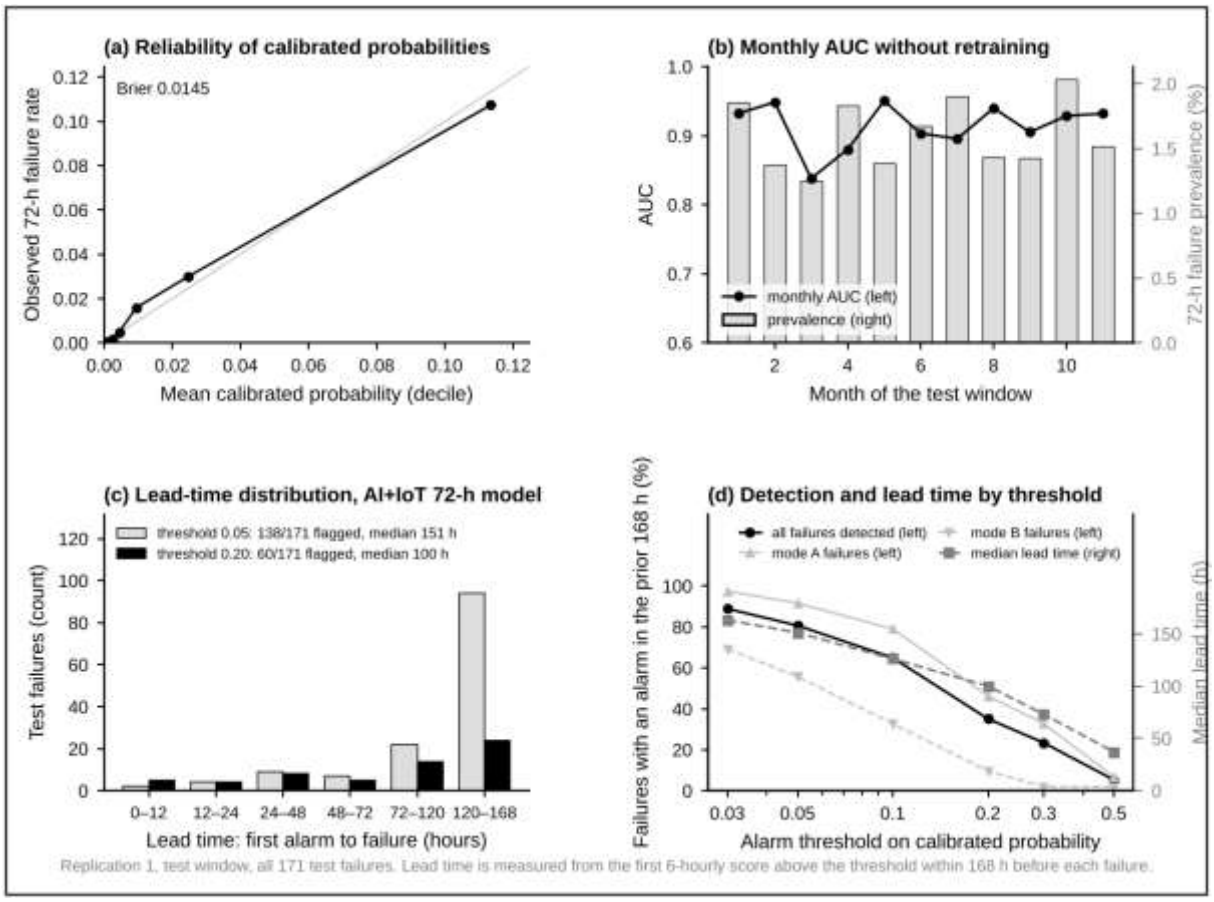


Figure 7: Global SHAP importance of the 72-hour AI+IoT model and AUC by machine type and failure mode (replication 1)

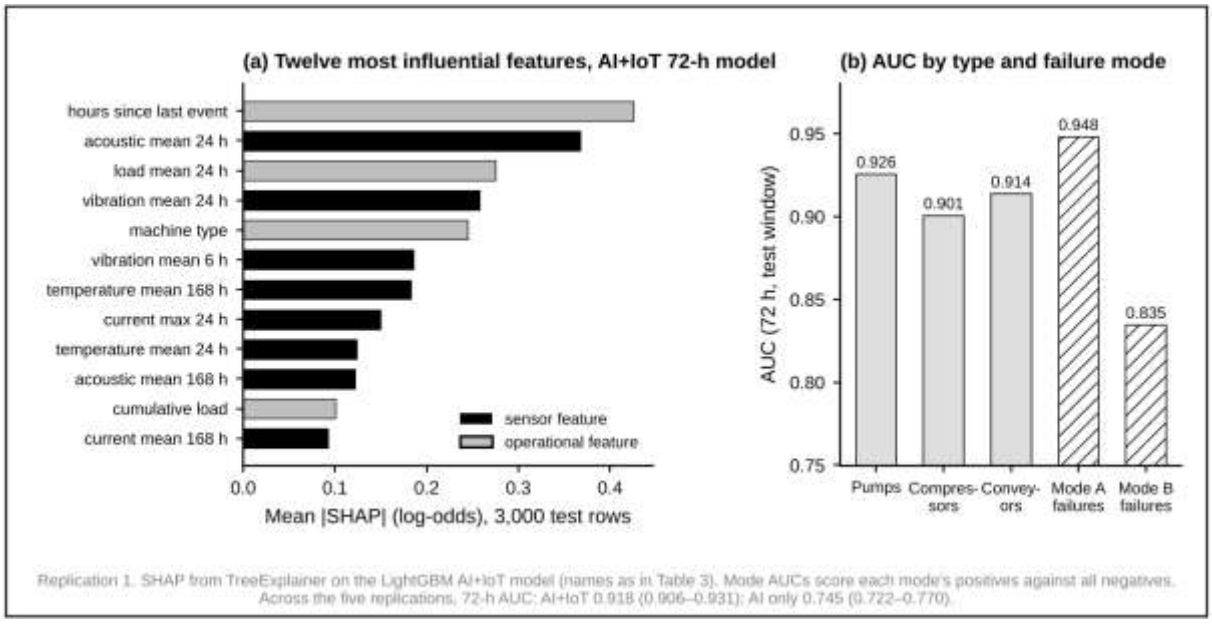


Table 8: Decision diagnostics of the 72-hour AI+IoT model: lead time and detection by threshold and failure mode, AUC by type and mode, monthly AUC and SHAP ranking (replication 1)

Diagnostic (72-h AI+IoT model, replication 1)	Result	Detail	Lead-time distribution (hours: count)
Lead time at probability threshold 0.03	152 of 171 test failures flagged within 168 h (89%); mode A 116/119, mode B 36/52	mean 137 h, median 163 h	0–12: 2; 12–24: 3; 24–48: 8; 48–72: 7; 72–120: 16; 120–168: 116
Lead time at probability threshold 0.05	138 of 171 test failures flagged within 168 h (81%); mode A 109/119, mode B 29/52	mean 128 h, median 151 h	0–12: 2; 12–24: 4; 24–48: 9; 48–72: 7; 72–120: 22; 120–168: 94
Lead time at probability threshold 0.1	111 of 171 test failures flagged within 168 h (65%); mode A 94/119, mode B 17/52	mean 109 h, median 126 h	0–12: 9; 12–24: 4; 24–48: 9; 48–72: 9; 72–120: 21; 120–168: 59
Lead time at probability threshold 0.2	60 of 171 test failures flagged within 168 h (35%); mode A 55/119, mode B 5/52	mean 93 h, median 100 h	0–12: 5; 12–24: 4; 24–48: 8; 48–72: 5; 72–120: 14; 120–168: 24
Lead time at probability threshold 0.3	40 of 171 test failures flagged within 168 h (23%); mode A 39/119, mode B 1/52	mean 80 h, median 73 h	0–12: 3; 12–24: 5; 24–48: 7; 48–72: 4; 72–120: 9; 120–168: 12
Lead time at probability threshold 0.5	9 of 171 test failures flagged within 168 h (5%); mode A 8/119, mode B 1/52	mean 64 h, median 37 h	0–12: 3; 12–24: 1; 24–48: 2; 48–72: 0; 72–120: 0; 120–168: 3
AUC by machine type: pump	0.926		
AUC by machine type: compressor	0.901		
AUC by machine type: conveyor	0.914		
AUC by failure mode: mode A failures vs all negatives	0.948		
AUC by failure mode: mode B failures vs all negatives	0.835		
Monthly AUC through the test window (no retraining)	0.838–0.950 over 11 months	prevalence 1.2–2.0%	
SHAP rank 1: hours_since_event	mean |SHAP| 0.425		
SHAP rank 2: acoustic_m24	mean |SHAP| 0.367		
SHAP rank 3: load_m24	mean |SHAP| 0.275		
SHAP rank 4: vibration_m24	mean |SHAP| 0.258		
SHAP rank 5: mtype	mean |SHAP| 0.245		
SHAP rank 6: vibration_m6	mean |SHAP| 0.185		
SHAP rank 7: temperature_m168	mean |SHAP| 0.183		
SHAP rank 8: cur_max24	mean |SHAP| 0.149		
SHAP rank 9: temperature_m24	mean |SHAP| 0.123		
SHAP rank 10: acoustic_m168	mean |SHAP| 0.122		

Maintenance policies (RQ3)

Table 9 and Figure 8 report the six policies and the oracle on the test window of the five replications, each policy tuned on the validation window. Reactive maintenance produced 167 failures per fleet in the eleven-month window, 1.93 per machine-year, 5,836 downtime hours, availability of 99.23% and a cost of 53,589 per machine-year, of which 43% was unplanned repair, the rest lost production. Every other policy cut failures by at least half, but at very different cost. The time-based policy, tuned to an interval of 2,000 hours in every replication, cut failures to 0.34 per machine-year but did so with 363 interventions per fleet, 187 of them on machines whose worse mode was below half its threshold, and its cost of 37,480 per machine-year was the highest of the preventive policies. The two condition-based rules did much better: the absolute vibration rule (tuned level 3.4 or 3.8) and the relative rule (tuned ratio 1.5 or 1.7) cut failures to 0.49 and 0.43 per machine-year with 213 and 221 interventions, raised

availability to 99.61 and 99.62%, and cost 29,895 and 28,966 per machine-year; their false alarms varied widely between replications (1 to 61 for the absolute rule), because the tuned level sits on a steep part of the cost curve. The AI-only policy was the weakest of the preventive policies: with a threshold of 0.02 to 0.05 on the no-sensor model it still allowed 0.84 failures per machine-year, made 66 unnecessary interventions and cost 38,981 per machine-year, more than the time-based policy in three replications; age and duty tell the model that a machine is at risk, but not which one, so it intervenes too late on some and too early on others.

The AI+IoT policy, tuned to a threshold of 0.03 in four replications and 0.05 in the fifth, produced 17.4 failures per fleet (0.20 per machine-year, a tenth of the reactive rate and less than half the rate of the better rule), 235 interventions, 8.6 false alarms, 2,249 downtime hours, availability of 99.70% and a cost of 23,667 per machine-year. The oracle, intervening only when a latent mode came within 0.05 of its threshold, allowed 1.4 failures, needed 183 interventions and cost 14,538 per machine-year at 99.82% availability; it is the floor that no observable policy can reach.

Table 9: Policy key performance indicators on the test window: mean (standard deviation) over five replications, with the tuned settings

KPI, mean (SD) over 5 replications	Reactive	Time-based	CBM (vibration level)	CBM (relative vibration)	AI only (no sensors)	AI+IoT	Oracle
Failures in test window	167.0 (6.3)	29.6 (3.8)	42.2 (7.4)	37.2 (9.5)	72.4 (12.9)	17.4 (7.2)	1.4 (0.9)
Failures per machine-year	1.93 (0.07)	0.34 (0.04)	0.49 (0.09)	0.43 (0.11)	0.84 (0.15)	0.20 (0.08)	0.02 (0.01)
Planned interventions	0.0 (0.0)	363.4 (3.0)	212.8 (31.9)	221.0 (34.8)	205.0 (31.4)	235.4 (20.8)	182.8 (5.0)
Unnecessary interventions (false alarms)	0.0 (0.0)	187.2 (6.8)	24.2 (26.0)	18.4 (19.6)	66.2 (19.1)	8.6 (7.5)	0.0 (0.0)
Availability (%)	99.23 (0.03)	99.53 (0.02)	99.61 (0.01)	99.62 (0.02)	99.48 (0.04)	99.70 (0.02)	99.82 (0.01)
Downtime hours (fleet)	5,836 (246)	3,580 (121)	2,959 (82)	2,851 (150)	3,959 (284)	2,249 (160)	1,332 (48)
Unplanned repair cost (thousand)	2,004 (75)	355 (46)	506 (89)	446 (114)	869 (155)	209 (86)	17 (11)
Planned intervention cost (thousand)	0 (0)	1,272 (10)	745 (112)	774 (122)	718 (110)	824 (73)	640 (17)
Lost contribution (thousand)	2,626 (111)	1,611 (54)	1,332 (37)	1,283 (68)	1,782 (128)	1,012 (72)	599 (22)
Total cost per machine-year	53,589 (2,155)	37,480 (1,080)	29,895 (748)	28,966 (1,274)	38,981 (2,384)	23,667 (1,373)	14,538 (471)
Tuned setting, replication 1	—	TIME-2000	CBM-3.8	CBMREL-1.7	AIONLY-0.02	AIIOT-0.03	—

Figure 8: Policy comparison across five replications: failures per machine-year, availability, cost per machine-year by component, and the paired cost difference of AI+IoT against each policy

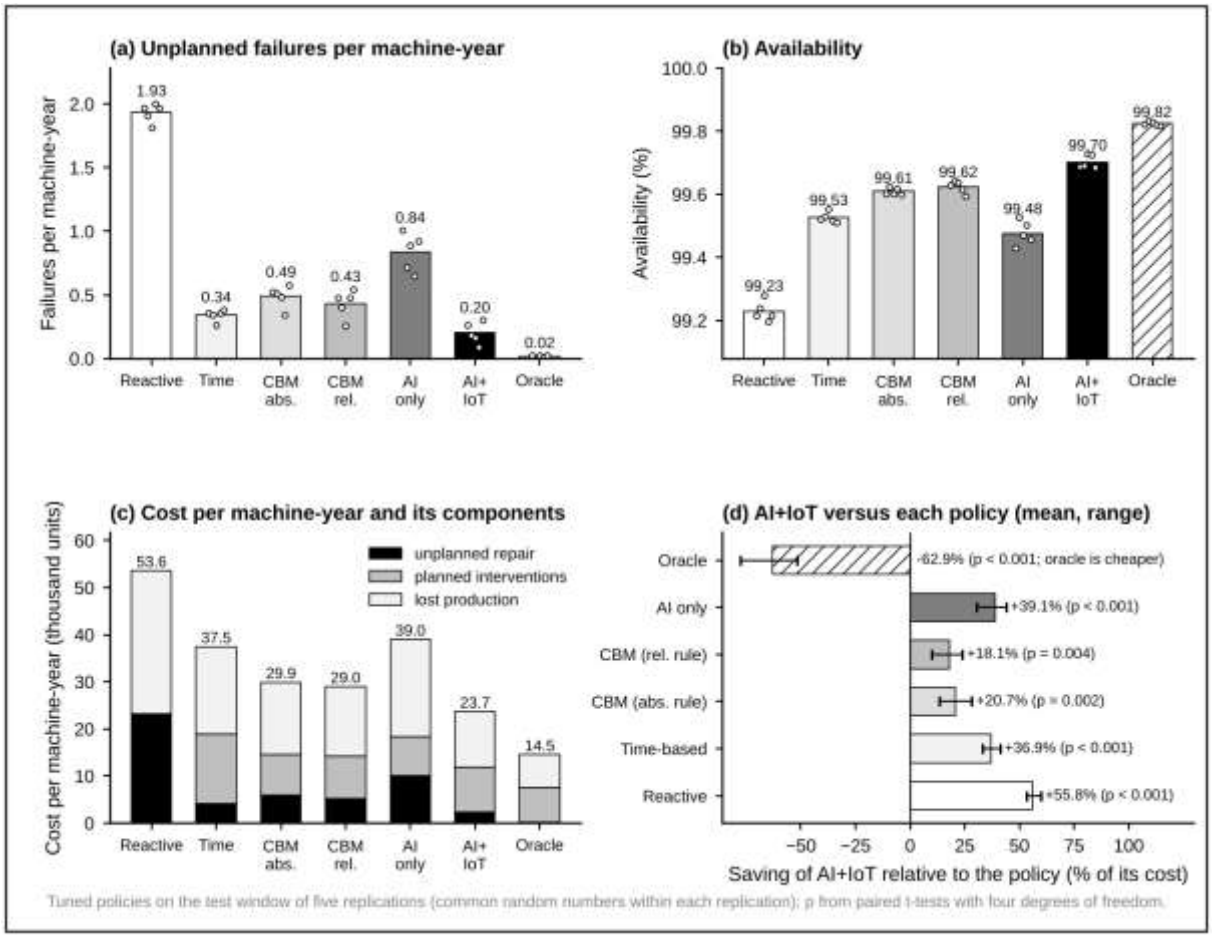


Table 10 gives the paired comparisons. Relative to each comparator, AI+IoT reduced cost per machine-year by 55.8% against reactive (range 53.1 to 59.8% across replications), 36.9% against time-based (33.3 to 41.2%), 20.7% against the absolute vibration rule (13.3 to 28.2%), 18.1% against the relative rule (9.9 to 24.0%) and 39.1% against AI-only (30.3 to 44.0%); the paired t-tests with four degrees of freedom gave $p = 0.002$ and $p = 0.004$ for the two rules and $p < 0.001$ for the rest. Availability rose by 0.47 percentage points against reactive, 0.18 against time-based, 0.09 and 0.08 against the two rules and 0.23 against AI-only. Against the oracle, AI+IoT cost 62.9% more (51.8 to 77.8%), which is the price of imperfect foresight: the oracle never intervenes early and never misses, whereas the tuned AI+IoT policy makes 29% more interventions than the oracle and still misses 17 failures per fleet; the lead-time diagnostic suggests that these are mostly mode-B and shock-driven failures.

Table 10: Paired comparisons of AI+IoT with each policy over five replications, and the synergy decomposition

Comparison	Cost per machine-year, mean difference	Difference (% of comparator): mean (range)	Paired t-test (n = 5)	Availability difference (pp)	Paired t-test
AI+IoT − Reactive	−29,922	−55.8 (−59.8 to −53.1)	t = −31.79, p = 0.000	+0.47	t = 33.12, p = 0.000
AI+IoT − Time-based	−13,813	−36.9 (−41.2 to −33.3)	t = −25.52, p = 0.000	+0.18	t = 21.24, p = 0.000
AI+IoT − CBM (vibration level)	−6,228	−20.7 (−28.2 to −13.3)	t = −7.48, p = 0.002	+0.09	t = 9.30, p = 0.001
AI+IoT − CBM (relative vibration)	−5,299	−18.1 (−24.0 to −9.9)	t = −5.96, p = 0.004	+0.08	t = 6.90, p = 0.002
AI+IoT − AI only (no sensors)	−15,314	−39.1 (−44.0 to −30.3)	t = −12.81, p = 0.000	+0.23	t = 13.35, p = 0.000
AI+IoT − Oracle	+9,129	+62.9 (+51.8 to +77.8)	t = 14.88, p = 0.000	−0.12	t = −12.88, p = 0.000
Synergy decomposition (mean of 5): saving vs reactive, per machine-year	IoT rule 24,784; AI only 14,608; sum 39,393; AI+IoT 29,922	additive interaction −9,471 (range −14,567 to −4,138)	attainable (reactive − oracle) 39,051		
Shares of the attainable saving (mean of 5)	IoT rule 63%; AI only 37%; AI+IoT 77%	independent-capture benchmark 1 − (1 − IoT)(1 − AI) = 77%	AI+IoT minus benchmark −0.4 pp (range −8.5 to +6.6)		
Conditional gains (mean of 5), per machine-year	AI given the IoT rule: 5,138 (18% of the rule's cost)	IoT given AI only: 15,314 (39% of the AI-only cost)			

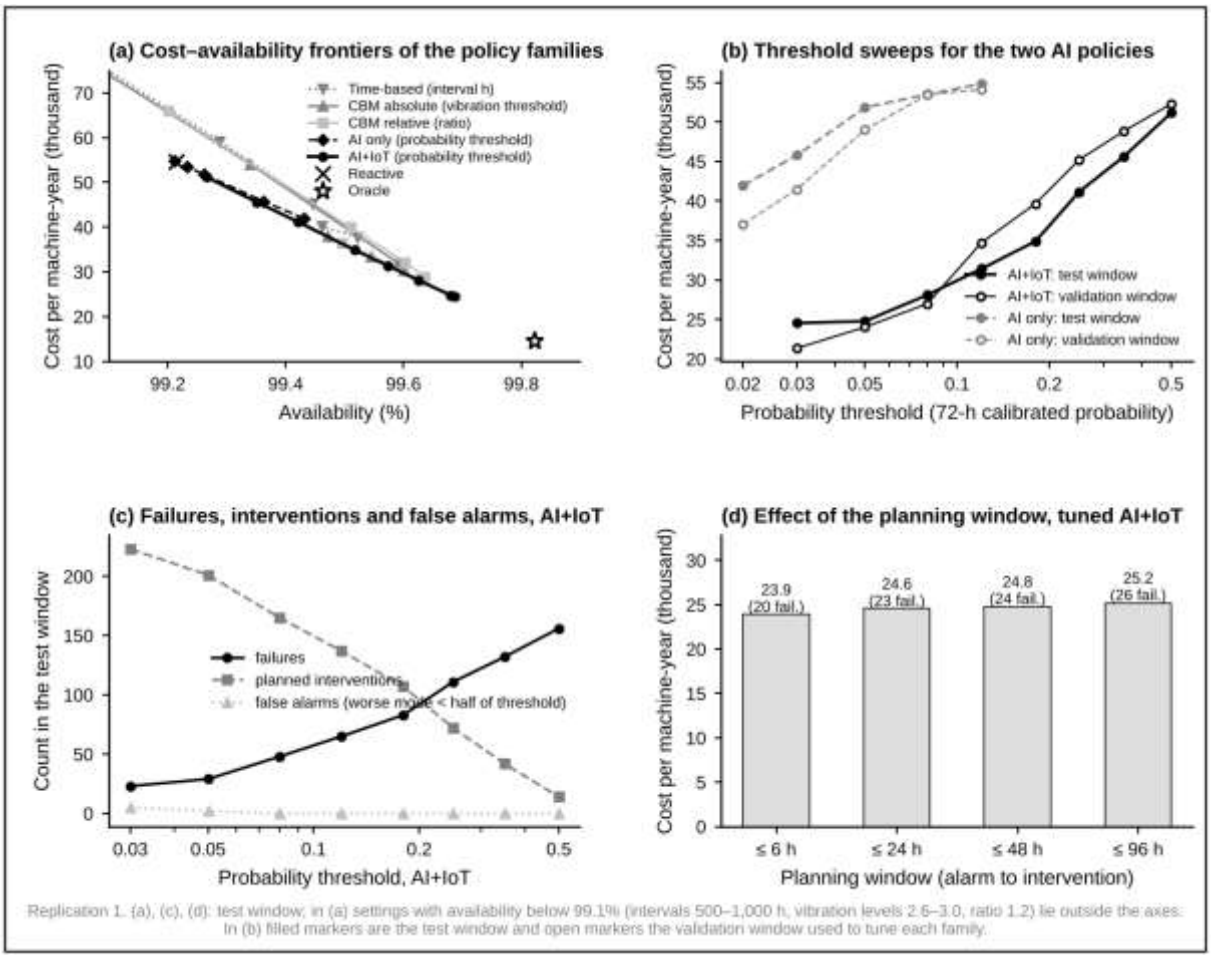
Parameter sweeps and frontiers

Figure 9 and Table 11 show how each family behaves across its parameter in the first replication. Every family traces a U-shaped cost curve: intervening too often multiplies planned cost and false alarms (an interval of 500 hours produced 1,534 interventions and 1,454 false alarms; a vibration level of 2.6 produced 2,992 interventions), and intervening too rarely returns the fleet toward reactive performance. In the cost–availability plane (Figure 9a) the families form nested frontiers: at any availability the AI+IoT frontier lies below the rule frontiers, which lie below the time-based and AI-only frontiers, and the best setting of each family lies at the lower-right end of its curve. The AI+IoT threshold sweep (Figure 9b, c) shows the trade-off in counts: raising the threshold from 0.03 to 0.5 reduced interventions from 223 to 14 and false alarms from 5 to 0, while failures rose from 23 to 156 and cost from 24,551 to 51,182 per machine-year, close to reactive. Because the tuned thresholds of both AI policies lay at the lower edge of the grid first searched, the grid was extended downward in the first replication. On the validation window the AI+IoT cost was 29,248 at 0.005, 25,760 at 0.01, 23,297 at 0.015, 21,785 at 0.02, 21,376 at 0.03 and 24,025 at 0.05, so 0.03 is an interior optimum; on the test window the curve is flat between 0.015 and 0.05 (24,046 to 24,785), and a threshold of 0.02 would have been marginally cheaper than the one selected, which is the expected size of the tuning penalty. For the AI-

only policy the validation optimum was 0.02 (37,043) with 0.015 and 0.03 both worse, and a threshold of 0.005 was catastrophic (11,839 interventions), because the no-sensor model assigns a low but non-negligible probability to almost every ageing machine.

The planning window had a small effect (Figure 9d): allowing up to 6, 24, 48 or 96 hours between the alarm and the intervention gave 20, 23, 24 and 26 failures and costs of 23,895, 24,551, 24,795 and 25,207 per machine-year. The reason is the lead time: with a median lead of 163 hours at the tuned threshold, a four-day scheduling window still leaves most alarms time to act, and the failures that the policy misses are mostly those with little or no lead, which a shorter window would not catch either.

Figure 9: Parameter sweeps in replication 1: cost–availability frontiers of the five policy families, threshold sweeps for the AI policies on validation and test windows, failures, interventions and false alarms against the AI+IoT threshold, and the effect of the planning window



Replication 1. (a), (c), (d): test window; in (a) settings with availability below 99.1% (intervals 500–1,000 h, vibration levels 2.6–3.0, ratio 1.2) lie outside the axes. In (b) filled markers are the test window and open markers the validation window used to tune each family.

Table 11: Parameter sweeps on the test window of replication 1 (with validation-window cost), including the extended threshold grid for the AI policies and the planning-window variants

Policy family	Parameter (interval h, level, ratio, probability threshold or planning window)	Failures	Planned interventions	False alarms	Availability (%)	Cost per machine-year, test	Cost per machine-year, validation
AI+IoT	0.005 (extended grid)	3	377	114	99.64	29,980	29,248
AI+IoT	0.01 (extended grid)	7	303	41	99.69	25,569	25,760
AI+IoT	0.015 (extended grid)	11	273	13	99.70	24,545	23,297
AI+IoT	0.02 (extended grid)	14	254	9	99.70	24,046	21,785
AI+IoT	0.03	23	223	5	99.69	24,551	21,376
AI+IoT	0.05	29	201	2	99.68	24,785	24,025
AI+IoT	0.08	48	165	0	99.63	28,116	27,009
AI+IoT	0.12	65	137	0	99.57	31,421	34,723
AI+IoT	0.18	83	107	0	99.52	34,894	39,661
AI+IoT	0.25	111	72	0	99.42	41,167	45,214
AI+IoT	0.35	132	42	0	99.35	45,628	48,840
AI+IoT	0.5	156	14	0	99.27	51,182	52,312
AI+IoT	planning window ≤ 6 h (threshold 0.03)	20	227	4	99.70	23,895	—
AI+IoT	planning window ≤ 48 h (threshold 0.03)	24	222	3	99.68	24,795	—
AI+IoT	planning window ≤ 96 h (threshold 0.03)	26	219	4	99.68	25,207	—
AI only (no sensors)	0.005 (extended grid)	1	11839	11797	89.05	911,228	911,751
AI only (no sensors)	0.01 (extended grid)	26	412	226	99.50	40,061	44,605
AI only (no sensors)	0.015 (extended grid)	50	277	115	99.51	37,379	40,280
AI only (no sensors)	0.02	87	184	58	99.43	41,985	37,043
AI only (no sensors)	0.03	115	115	29	99.36	45,787	41,481
AI only (no sensors)	0.05	152	39	8	99.26	51,821	49,021
AI only (no sensors)	0.08	163	15	2	99.23	53,507	53,443
AI only (no sensors)	0.12	170	3	1	99.21	54,832	54,112
CBM (vibration level)	2.6	8	2992	2892	97.20	232,830	234,361
CBM (vibration level)	3.0	14	644	475	99.34	54,064	55,449
CBM (vibration level)	3.4	32	283	70	99.59	32,060	32,519
CBM (vibration level)	3.8	45	206	9	99.60	30,308	30,884
CBM (vibration level)	4.2	71	138	0	99.54	33,425	34,463
CBM (vibration level)	4.6	96	92	0	99.47	37,914	39,969
CBM (relative vibration)	1.2	9	819	692	99.20	65,922	68,039
CBM (relative vibration)	1.3	13	465	254	99.51	39,965	41,488
CBM (relative vibration)	1.4	16	351	117	99.60	32,128	34,668
CBM (relative vibration)	1.5	22	284	38	99.64	28,920	30,735
CBM (relative vibration)	1.7	41	195	4	99.63	28,177	29,600
CBM (relative vibration)	2.0	87	112	0	99.50	36,521	37,654
Time-based	500	0	1534	1454	98.58	118,068	118,068
Time-based	750	0	1049	936	99.03	80,739	86,973
Time-based	1000	1	765	617	99.29	59,201	60,514
Time-based	1500	11	544	348	99.45	45,403	46,855
Time-based	2000	31	362	184	99.52	37,819	38,422
Time-based	3000	74	213	77	99.46	40,161	38,948

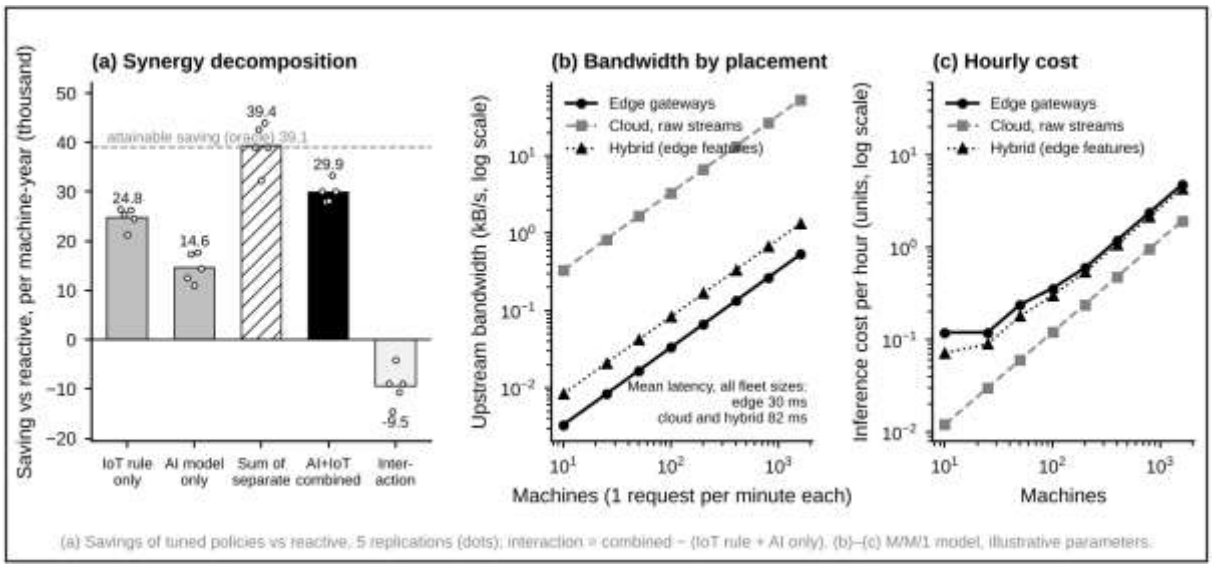
Synergy decomposition

The decomposition in Table 10 and Figure 10a is the direct test of the synergy claim within the model. Relative to reactive maintenance, the better IoT rule saved 24,784 per machine-year (mean of five replications), the AI-only model saved 14,608, and their sum is 39,393; the combined AI+IoT policy saved 29,922. The additive interaction is therefore negative in every replication, −9,471 on average (−14,567 to −4,138): on an additive scale, AI and IoT together deliver less than the sum of what each

delivers alone. The reason is a ceiling. The oracle shows that the attainable saving in this fleet is 39,051 per machine-year, almost exactly the sum of the two separate savings, so additivity would have required the combination to be perfect. Expressed as shares of the attainable saving, the IoT rule captured 63% (59 to 66%), the AI-only model 37% (30 to 44%) and the combination 77% (72 to 81%). If the two components captured independent portions of the attainable saving, the combination would capture $1 - (1 - 0.63)(1 - 0.37) = 77\%$ (72 to 81% across replications); the observed share differs from this benchmark by −0.4 percentage points on average (−8.5 to +6.6). The combination is thus consistent with complementary, independent contributions rather than with either redundancy or super-additivity.

The conditional gains are the practically relevant statement of the same result. Given an existing condition-based rule, adding the learning model on the same sensors plus operational records saved a further 5,138 per machine-year (2,717 to 7,053), 18% of the rule's cost; given an existing no-sensor model, adding sensors saved a further 15,314 (10,807 to 17,434), 39% of the AI-only cost. Sensing is the larger of the two contributions, and learning is worth more once sensing is in place than sensing is worth without learning in relative terms, because the learning model is what resolves the machine-specific baselines and load confounding that defeat a fixed rule.

Figure 10: Synergy decomposition over five replications, and the edge-versus-cloud placement model: upstream bandwidth and hourly cost by placement and fleet size



Edge versus cloud placement (RQ4)

Table 12 and Figure 10b–c give the placement model. With one gateway per 40 machines and one request per machine per minute, the gateways are lightly loaded at every fleet size (utilisation below 2%), and the mean latency of local inference is 30 milliseconds against 82 milliseconds for cloud inference, where the round trip dominates; neither placement approaches saturation in the range studied. The differences that matter are bandwidth and cost. Sending raw streams to the cloud requires 3.3 kilobytes per second for 100 machines and 53 kilobytes per second for 1,600, against 0.03 and 0.53 for edge inference and 0.08 and 1.33 for the hybrid; the hundredfold reduction for the edge and fortyfold for the hybrid is what makes local feature extraction attractive where connectivity is metered or intermittent. Cost runs the other way: for 100 machines the three gateways cost 0.36 per hour against 0.12 for cloud inference and 0.30 for the hybrid, and for 1,600 machines 4.80 against 1.93 and 4.32, because the fixed cost of gateways scales with the fleet while per-request cloud charges are small at one request per minute. The model therefore says that, for predictive maintenance at this request rate, the cloud is the cheaper place to run inference and the edge is the place to run it when latency, bandwidth or continuity of operation during network loss matter; the hybrid buys most of the bandwidth benefit at close to the edge cost. For closed-loop control at sub-second rates the request rate would be hundreds of times higher and the conclusion would reverse, which is why Figure 1 places feature extraction and

local inference at the edge and model training and fleet-level optimisation in the cloud.

Table 12: Edge-versus-cloud inference placement: utilisation, mean latency, hourly cost and upstream bandwidth by placement and fleet size (analytical model, parameters)

Machines	Placement	Gateways	Requests per s	Utilisation (%)	Mean latency (ms)	Cost per hour	Upstream bandwidth (kB/s)
10	Edge gateways (local inference)	1	0.17	0.4	30	0.12	0.00
10	Cloud, raw streams	—	0.17	0.0	82	0.01	0.33
10	Hybrid: edge features, cloud inference	1	0.17	0.0	82	0.07	0.01
50	Edge gateways (local inference)	2	0.83	1.0	30	0.24	0.02
50	Cloud, raw streams	—	0.83	0.2	82	0.06	1.67
50	Hybrid: edge features, cloud inference	2	0.83	0.2	82	0.18	0.04
200	Edge gateways (local inference)	5	3.33	1.7	30	0.60	0.07
200	Cloud, raw streams	—	3.33	0.7	82	0.24	6.67
200	Hybrid: edge features, cloud inference	5	3.33	0.7	82	0.54	0.17
800	Edge gateways (local inference)	20	13.33	1.7	30	2.40	0.27
800	Cloud, raw streams	—	13.33	2.7	82	0.96	26.67
800	Hybrid: edge features, cloud inference	20	13.33	2.7	82	2.16	0.67
1600	Edge gateways (local inference)	40	26.67	1.7	30	4.80	0.53
1600	Cloud, raw streams	—	26.67	5.3	82	1.93	53.33
1600	Hybrid: edge features, cloud inference	40	26.67	5.3	82	4.32	1.33

Business value

Table 13 and Figure 11 translate the policy results into a five-year value. For 96 machines, the capital cost of the AI+IoT programme under the assumptions of Table 6 is 402,000 and the operating cost 177,600 per year. Against reactive maintenance the saving of 29,922 per machine-year gives 2.88 million per year, a net present value of 10.4 million at 8% (9.7 to 11.6 million across replications) and a payback of under two months; against time-based maintenance the figures are 1.27 million per year and a net present value of 4.0 million. These numbers are large because the simulated fleet fails almost twice per machine-year under reactive maintenance and each failure costs 28,200 in repair and lost production; they describe a fleet in poor condition, not a typical plant. The comparison that most plants face is with a condition-based rule they already run, and there the value is an order of magnitude smaller: against the absolute vibration rule the saving is 552,722 per year in the first replication (597,878 on average, 353,332 to 2.2 million in net present value across replications), the net present value is 1.10 million and the payback 1.07 years; against the relative rule the saving is 348,111 per year, the net present value 0.28 million and the payback 2.4 years, and in one replication the net present value was negative.

The sensitivity analysis (Figure 11) shows where the case holds. Scaling the simulated savings by a

failure-intensity factor, the net present value against reactive maintenance stays positive down to a factor of 0.10, against time-based maintenance to 0.22, against the absolute rule to 0.50 and against the relative rule to 0.80. A plant whose fleet fails a quarter as often as the synthetic one, about 0.5 failures per machine-year under reactive operation, would still justify the programme against reactive or time-based maintenance but not against a competently tuned condition-based rule. Fleet size matters through the fixed integration and model-operations costs: against the absolute rule the net present value is −0.20 million for 24 machines, 0.23 million for 48, 1.10 million for 96 and 13.2 million for 768, with payback falling from 2.2 years at 48 machines to 0.56 years at 768. Lost contribution per downtime hour and unplanned repair cost scale the value almost linearly (against the absolute rule, 0.08 million at 100 per hour and 2.68 million at 1,000; 0.31 million at a repair cost of 4,000 and 2.36 million at 25,000), because the saving comes from avoided failures. Sensor cost matters least: raising it from 1,000 to 10,000 per machine moved the net present value against the rule from 1.24 to 0.38 million. The value of AI+IoT is therefore decided by the failure intensity and failure cost of the fleet, the policy already in place and the fleet size over which fixed costs are spread, and only secondarily by the price of sensing.

Figure 11: Five-year net present value of moving from each baseline policy to AI+IoT against failure intensity, fleet size, downtime cost and repair cost (replication 1, assumptions of Table 6)

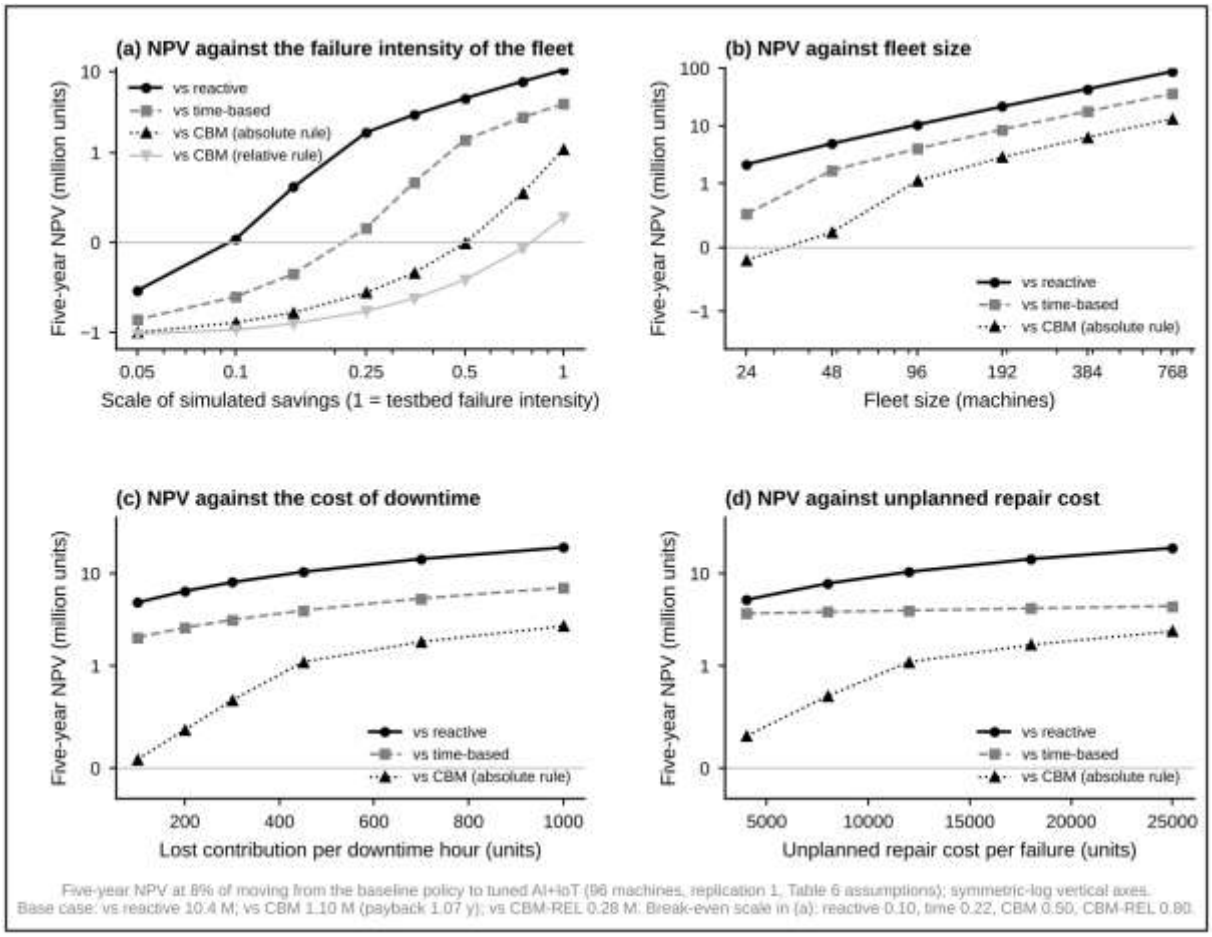


Table 13: Value model: base cases against four baseline policies (replication 1 and mean of five replications) and sensitivity to failure intensity, downtime cost, repair cost, fleet size and sensor cost

ase (replication 1, 96 machines unless stated)	Saving per year	Capex	Opex per year	Net per year	NPV (5 y, 8%)	Payback (years)	5-year ROI
AI+IoT vs reactive, replication 1	2,884,778	402,000	177,600	2,707,178	10,406,976	0.15	3267%
AI+IoT vs reactive, mean of 5 replications (NPV range)	2,872,522			2,694,922	10,358,043 (9,656,125 to 11,647,821)	0.15	3252%
AI+IoT vs time-based, replication 1	1,273,722	402,000	177,600	1,096,122	3,974,498	0.37	1263%
AI+IoT vs time-based, mean of 5 replications (NPV range)	1,326,044			1,148,444	4,183,406 (3,641,107 to 4,905,021)	0.35	1328%
AI+IoT vs CBM (vibration level), replication 1	552,722	402,000	177,600	375,122	1,095,754	1.07	367%
AI+IoT vs CBM (vibration level), mean of 5 replications (NPV range)	597,878			420,278	1,276,047 (353,332 to 2,200,404)	0.96	423%
AI+IoT vs CBM (relative), replication 1	348,111	402,000	177,600	170,511	278,801	2.36	112%
AI+IoT vs CBM (relative), mean of 5 replications (NPV range)	508,700			331,100	919,986 (−69,673 to 1,592,403)	1.21	312%
Sensitivity: Failure-intensity scale of savings = 1.0					vs reactive 10,406,976; vs time 3,974,498; vs CBM 1,095,754; vs CBM-REL 278,801	1.07 (vs CBM)	
Sensitivity: Failure-intensity scale of savings = 0.75					vs reactive 7,527,456; vs time 2,703,097; vs CBM 544,039; vs CBM-REL −68,675	1.70 (vs CBM)	
Sensitivity: Failure-intensity scale of savings = 0.5					vs reactive 4,647,935; vs time 1,431,696; vs CBM −7,676; vs CBM-REL −416,152	4.07 (vs CBM)	
Sensitivity: Failure-intensity scale of savings = 0.35					vs reactive 2,920,223; vs time 668,856; vs CBM −338,704; vs CBM-REL −624,638	25.36 (vs CBM)	
Sensitivity: Failure-intensity scale of savings = 0.25					vs reactive 1,768,415; vs time 160,296; vs CBM −559,390; vs CBM-REL −763,629	none (vs CBM)	
Sensitivity: Failure-intensity scale of savings = 0.15					vs reactive 616,607; vs time −348,265; vs CBM −780,076; vs CBM-REL −902,619	none (vs CBM)	
Sensitivity: Failure-intensity scale of savings = 0.1					vs reactive 40,703; vs time −602,545; vs CBM −890,419; vs CBM-REL −972,115	none (vs CBM)	
Sensitivity: Failure-intensity scale of savings = 0.05					vs reactive −535,201; vs time −856,825; vs CBM −1,000,762; vs CBM-REL −1,041,610	none (vs CBM)	
Sensitivity: Lost contribution per hour = 100					vs reactive 4,842,025; vs time 2,028,939; vs CBM 84,933		
Sensitivity: Lost contribution per hour = 200					vs reactive 6,432,011; vs time 2,584,813; vs CBM 373,739		
Sensitivity: Unplanned repair cost = 25,000					vs reactive 18,884,830; vs time 4,435,878; vs CBM 2,364,549		
Sensitivity: Fleet size (machines) = 24					vs reactive 2,128,900; vs time 520,781; vs CBM −198,905	56.61 (vs CBM)	
Sensitivity: Fleet size (machines) = 48					vs reactive 4,886,925; vs time 1,670,686; vs CBM 231,315	2.18 (vs CBM)	
Sensitivity: Fleet size (machines) = 96					vs reactive 10,406,976; vs time 3,974,498; vs CBM 1,095,754	1.07 (vs CBM)	
Sensitivity: Fleet size (machines) = 192					vs reactive 21,447,077; vs time 8,582,122; vs CBM 2,824,634	0.75 (vs CBM)	
Sensitivity: Fleet size (machines) = 384					vs reactive 43,523,279; vs time 17,793,368; vs CBM 6,278,393	0.62 (vs CBM)	
Sensitivity: Fleet size (machines) = 768					vs reactive 87,675,684; vs time 36,215,862; vs CBM 13,185,911	0.56 (vs CBM)	
Sensitivity: Sensor cost per machine = 1,000					vs reactive 10,550,976; vs CBM 1,239,754		
Sensitivity: Sensor cost per machine = 2,500					vs reactive 10,406,976; vs CBM 1,095,754		
Sensitivity: Sensor cost per machine = 5,000					vs reactive 10,166,976; vs CBM 855,754		
Sensitivity: Sensor cost per machine = 7,500					vs reactive 9,926,976; vs CBM 615,754		
Sensitivity: Sensor cost per machine = 10,000					vs reactive 9,686,976; vs CBM 375,754		
Break-even failure-intensity scale (NPV = 0)					vs Reactive 0.10; vs Time-based 0.22; vs CBM (vibration level) 0.50; vs CBM (relative vibration) 0.80		

Discussion

Principal findings

Four findings answer the research questions. On RQ1, the literature supports a closed loop of five mechanisms, sense, predict, optimise, automate and learn, that recurs across maintenance, supply chain, energy, security and quality operations (Figure 2); the mechanisms are the same, the data, decisions and performance indicators differ, and the predictive-maintenance instance is the one where the loop has been studied most and claimed most. On RQ2, learning on multiple IoT streams forecast failure within 72 hours with an AUC of about 0.91 in a fleet designed to be hard, against 0.83 for the

best single-sensor rule and 0.75 for a model on operational records alone; the gain over the rule came from combining sensors and discounting load, baseline and type, and the gain over the no-sensor model came from sensing itself. Lead time was long for wear-type failures and short or absent for the electrical mode and for shock-driven failures, which is where the residual failures of the best policy concentrate. On RQ3, the AI+IoT policy reduced cost per machine-year by 56% against reactive, 37% against time-based, 18 to 21% against tuned condition-based rules and 39% against an AI-only policy, with consistent direction in all five replications, and still cost 63% more than a perfect-foresight oracle. The synergy claim in its strong form, that the combination exceeds the sum of its parts, was not supported: the additive interaction was negative because the separate contributions already approached the attainable ceiling, and the combination captured almost exactly the share of attainable savings expected from two complementary, independent contributions. The claim in its practical form, that each component adds substantial value given the other, was supported. On RQ4, the business case depends on the failure intensity and failure cost of the fleet, on the baseline policy and on fleet size; it is overwhelming against reactive maintenance in a failure-prone fleet and marginal against a well-tuned condition-based rule in a small or reliable one; and inference placement is a cost-versus-latency-and-bandwidth trade-off in which the cloud wins on cost at one request per minute and the edge on bandwidth, latency and autonomy.

Relation to the literature

The forecasting results are in line with the reviews of machine learning for predictive maintenance, which report that tree ensembles and simple feature engineering on rolling statistics perform well on condition-monitoring data and that the choice of learner matters less than the features and the evaluation protocol (Carvalho et al., 2019; Zonta et al., 2020; Zhu et al., 2025). The small gap between LightGBM, random forest and logistic regression here (0.02 to 0.03 AUC) supports the argument of Rudin (2019) that simpler and more interpretable models deserve consideration in high-stakes operational settings, and the SHAP ranking shows that even the gradient-boosted model's reasoning, discounting sensor levels by duty and type, is one that a maintenance engineer would recognise. The study did not resample the 1.6% positive class, as is often done with techniques such as SMOTE (Chawla et al., 2002; He & Garcia, 2009), because the policies need calibrated probabilities and resampling distorts them (Guo et al., 2017); threshold tuning on the validation window is the appropriate way to handle imbalance when the decision cost is known. Spectral and envelope features, which the vibration-analysis literature regards as essential for bearing faults (Lei et al., 2018; Jardine et al., 2006), and automated feature extraction (Christ et al., 2018) were not used, and neither kernel methods (Cortes & Vapnik, 1995) nor sequence models (Hochreiter & Schmidhuber, 1997; Lim et al., 2021; Ismail Fawaz et al., 2019) were compared; these would be the natural extensions on real high-rate data, and the simulated signals are too simple to reward them.

The policy results add to the maintenance-optimisation literature (de Jonge & Scarf, 2020; Dekker, 1996; Wang, 2002; Alaswad & Xiang, 2017) an explicit comparison of learning-based condition-based maintenance with the rule-based and time-based policies that most plants actually run. The finding that a tuned condition-based rule captures about 63% of the attainable saving, and the learning model a further 13 percentage points, is consistent with the modest incremental gains that field studies of prognostics report once a monitoring programme is in place (Heng et al., 2009; Vachtsevanos et al., 2006; Bousdekis et al., 2018), and with the argument that the value of prognostics lies in the decision policy rather than in the accuracy metric (Susto et al., 2015; Compare et al., 2020; Nunes et al., 2023). The cost of false alarms, which the time-based policy and the aggressive rule settings made visible (hundreds of interventions on healthy machines), is the mechanism by which an alarm system loses the confidence of its users (Peng et al., 2010; Jardine et al., 2006), and the sweeps show that it is controlled by the threshold rather than by the model. The large gap to the oracle is a reminder that the forecasting problem is not solved by AUC 0.91: a week of lead time for most failures still leaves about a tenth of failures, mostly sudden ones, undetected at useful lead, and this is also the pattern reported for real fleets, where the share of random failures bounds what condition monitoring can achieve (Mobley, 2002; Tsang, 2002; Muchiri & Pintelon, 2008).

The statistical treatment deserves a note. Five replications give paired tests with four degrees of freedom, and the p-values should be read as descriptive; the ranges across replications are the more

useful statement (Demšar, 2006; Nadeau & Bengio, 2003). The use of common random numbers within each replication is what makes the comparison sharp, following the simulation-methodology literature (Law, 2015; Sargent, 2013), and the chronological split with validation-window tuning follows the time-series evaluation literature (Bergmeir & Benítez, 2012; Cerqueira et al., 2020). The pipeline was built on the standard scientific Python stack (Harris et al., 2020; Virtanen et al., 2020; Pedregosa et al., 2011).

Application to other operational domains

Table 14 carries the framework of Figure 2 into the four domains that were not simulated, drawing on the reviews cited in the Background. The structure of the maintenance result transfers: in each domain the IoT contribution is a stream of granular observations that a simple rule can already exploit, the AI contribution is a model that discounts confounders and combines streams, and the value depends on the decision policy and the cost structure rather than on the model's accuracy.

In supply chains and logistics, IoT supplies asset tracking, cold-chain telemetry and inventory visibility (Ben-Daya et al., 2019; de Vass et al., 2021; Manavalan & Jayakrishna, 2019), and AI supplies demand, lead-time and disruption forecasts and the optimisation that uses them (Nguyen et al., 2018; Seyedan & Mafakheri, 2020; Brintrup et al., 2020). The resilience literature has shown by simulation that the value of visibility lies in earlier and better-targeted responses to disruption, the ripple effect, rather than in forecasting alone (Ivanov, 2017; Dolgui & Ivanov, 2021; Hosseini, Ivanov, & Dolgui, 2019; Katsaliaki et al., 2022), and that the baselines matter: persistence and simple exponential-smoothing forecasts are hard to beat on operational demand series (Syntetos et al., 2016; Makridakis et al., 2022; Hyndman & Athanasopoulos, 2021), just as the vibration rule was hard to beat here. The pandemic literature made the same point about intertwined networks and the limits of prediction (Ivanov & Dolgui, 2020; Queiroz et al., 2022; Cohen & Kouvelis, 2021), and reinforcement-learning approaches to inventory and routing are the supply-chain analogue of the policy simulation reported here (Rolf et al., 2023; Bertsimas & Kallus, 2020). In energy management, sub-metering and building and machine telemetry supply the data, load and price forecasts supply the predictions, and scheduling and set-point control supply the decisions (Shrouf et al., 2014; Minoli et al., 2017; Ahmad et al., 2021); control loops at second-to-minute rates are the clearest case for edge placement. In security and safety, device logs, network flows and cameras supply the observations, anomaly and intrusion detection supply the predictions (Chandola et al., 2009; Meidan et al., 2018; Hussain et al., 2020; Alsaedi et al., 2020), and the false-alarm cost dominates even more than in maintenance, because every alarm consumes analyst time (Sicari et al., 2015; Alaba et al., 2017; Roman et al., 2018). In quality and customer operations, inline inspection, process parameters and product telemetry feed defect prediction and root-cause analysis (Wuest et al., 2016; Zhong et al., 2017; Mourtzis et al., 2016), and smart connected products feed proactive service (Porter & Heppelmann, 2014; Porter & Heppelmann, 2015; Wortmann & Flüchter, 2015); the characteristic difficulty is that labels, field returns and complaints, arrive late, so the learn mechanism must be designed before the predict mechanism can be trusted (Gilchrist, 2016; Mittal et al., 2018).

Table 14: Applying the synergy framework to the five operational domains: IoT data, AI tasks, decisions, evidence base and what the simulation implies

Operational domain	IoT data	AI task	Decision and automation	Evidence base (reviews)	What the simulation implies
Predictive maintenance	Vibration, temperature, current, pressure, acoustic; work orders	Failure forecasting, RUL, anomaly detection	Scheduling within a planning window; spare parts; derating	Jardine et al. (2006); Lei et al. (2018); Carvalho et al. (2019); Zonta et al. (2020)	Value is in the policy, not the model: tune thresholds to cost; compare with a condition-based rule; expect a gap to perfect foresight
Supply chain and logistics	Asset and shipment tracking, cold-chain telemetry, inventory sensors	Demand, lead-time and disruption forecasting	Replenishment, routing, allocation under uncertainty	Ben-Daya et al. (2019); Nguyen et al. (2018); Ivanov & Dolgui (2021)	Same structure: forecast → scenario → optimisation; persistence baselines are strong
Energy management	Sub-metering, HVAC and machine load, ambient conditions	Load and price forecasting; anomaly detection in consumption	Scheduling, set-points, demand response	Shrouf et al. (2014); Minoli et al. (2017); Ahmad et al. (2021)	Edge placement for control loops; cloud for fleet-level optimisation
Security and safety	Device logs, network flows, cameras, access events	Intrusion and anomaly detection	Isolation, alerts, response prioritisation	Sicari et al. (2015); Hussain et al. (2020); Meidan et al. (2018)	False-alarm cost dominates; tune thresholds to response capacity
Quality and customer operations	Inline inspection, process parameters, product telemetry	Defect prediction, root-cause analysis, churn	Parameter tuning, proactive service	Zhong et al. (2017); Wuest et al. (2016); Porter & Heppelmann (2015)	Labels from field returns arrive late; design the feedback loop first

A critical reading of percentage claims

Practitioner reports, consultancy surveys and a good deal of the student and conference literature on AI and IoT in business operations cite round-number improvements, such as a stated 25 to 30% reduction in downtime or maintenance cost, a 50% reduction in energy waste or a 90% detection rate for security incidents, usually without the baseline policy, the cost assumptions, the sample or the

method that produced them. The present results suggest three questions to put to any such figure. First, improvement relative to what? The same AI+IoT policy produced a 56% cost reduction against reactive maintenance, 37% against scheduled maintenance and 18 to 21% against a tuned condition-based rule, in the same fleet on the same disruption paths; a percentage without its baseline is not a result. Second, at what cost of false alarms? Every preventive policy here could be tuned to near-zero failures at the price of hundreds of unnecessary interventions, and a downtime reduction reported without the intervention count or the planned downtime is incomplete. Third, over what failure intensity and fleet? The value model showed the same operational improvement generating a net present value of 10 million against one baseline and a negative one against another at a smaller fleet or a lower failure rate. Academic surveys that associate adoption with self-reported performance (Dalenogare et al., 2018; Mikalef et al., 2019; Tortorella & Fettermann, 2018; Kamble et al., 2018) cannot answer these questions, and the field would benefit from reporting standards for predictive-maintenance studies that require the baseline policy, the intervention counts and the cost model (Nunes et al., 2023; Zonta et al., 2020), as the forecasting community has done through its competitions (Makridakis et al., 2022).

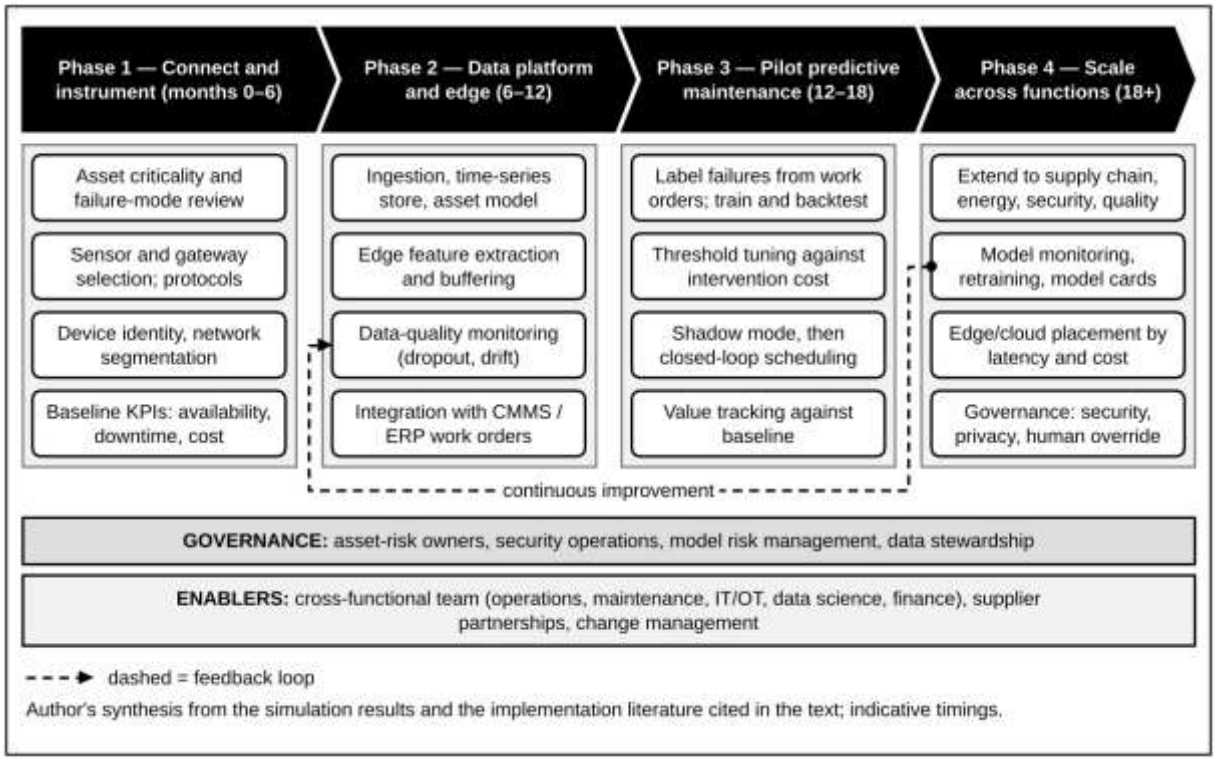
Implementation roadmap

Figure 12 sets out a phased roadmap derived from the simulation results and the implementation literature (Moeuf et al., 2018; Ghobakhloo, 2018; Frank et al., 2019; Davenport & Ronanki, 2018). Phase 1 connects and instruments: an asset-criticality and failure-mode review decides which machines and which sensors, since the results show that the mode a sensor set cannot see sets the floor on failures; device identity and network segmentation are designed in from the start (Sicari et al., 2015; Roman et al., 2018); and baseline KPIs, failures, availability, downtime and cost per machine-year under the current policy, are measured, because the value of the programme will be judged against them. Phase 2 builds the data platform and the edge: ingestion, a time-series store and an asset model (Tao et al., 2019; Qi & Tao, 2018; Fuller et al., 2020), feature extraction and buffering on gateways, data-quality monitoring for dropout and drift (Karkouch et al., 2016), and integration with the maintenance-management system so that work orders and failures become labels. Phase 3 pilots predictive maintenance on the instrumented assets: labels are taken from work orders, models are trained and backtested with chronological splits, thresholds are tuned against the plant's own intervention and failure costs rather than at 0.5, the model runs in shadow mode beside the existing rule, and value is tracked against the Phase 1 baseline. Phase 4 scales across functions, to supply chain, energy, security and quality, adds model monitoring, retraining and documentation (Sculley et al., 2015; Breck et al., 2017; Mitchell et al., 2019; Lu et al., 2019), decides edge or cloud placement by request rate, latency and bandwidth, and installs the governance that the next section describes. The feedback arrow from value tracking to data-quality monitoring is the learn mechanism applied to the programme itself.

Governance, security and privacy

The architecture of Figure 1 carries a governance band for a reason. IoT devices extend the attack surface of the operation and are often weakly authenticated and rarely patched (Sicari et al., 2015; Alaba et al., 2017; Hussain et al., 2020); the data they produce can identify workers and customers (Weber, 2010; Roman et al., 2018); and the models trained on them can fail silently when the process drifts (Gama et al., 2014; Lu et al., 2019). Three practices follow from the results. Thresholds and policies should be owned by the asset-risk owners who bear the cost of failures and false alarms, not by the data-science team, because the sweeps show that the threshold is a business decision. Models should be monitored for calibration and drift with the same diligence as the sensors, since a probability that is no longer calibrated silently moves the policy along its cost curve; monthly AUC and reliability curves, as in Figure 6, are inexpensive to compute. And human override should be designed in: the oracle gap means that the best policy still fails, and the engineer who can see a spike that the model has discounted must be able to act. Federated and edge learning (Nguyen et al., 2021; Zhou et al., 2019) and model documentation (Mitchell et al., 2019) are the technical complements to these practices.

Figure 12: Implementation roadmap for AI–IoT synergies in business operations: four phases with governance and enabler bands (author's synthesis; indicative timings)



Limitations

Table 15 lists the limitations by component. The most important is that all data are synthetic. The degradation process, the sensing model and the cost parameters were chosen to be plausible and difficult, not estimated from any plant, and the effect sizes depend on them: a fleet whose failures are mostly sudden would show a smaller gain for every preventive policy and a smaller synergy, and a fleet with cleaner sensor signals would show a smaller gap between the rule and the learning model. The simulated failure intensity, 1.9 per machine-year under reactive operation, is high, and the failure-intensity sensitivity is the honest statement of what the value results mean for fleets that fail less often. The sensing model has fixed functional forms and five sensors, and the features are rolling statistics; the forecasting results therefore understate what signal processing can extract from high-rate vibration data and overstate the simplicity of real signals. The policy simulation ignores crew, spare-parts and production-schedule constraints, so the scheduling flexibility of the preventive policies is overstated. The edge-versus-cloud model uses rates and prices, and the value model's investment and operating costs are assumptions; both are meant to show the shape of trade-offs, not to price a deployment. Five replications give wide ranges, and the policy comparisons were tuned once on a validation window and tested once, so the reported costs include a tuning penalty that is visible in the extended-grid results. Only predictive maintenance was simulated; the application of the framework to other domains rests on the literature. Finally, the author's characterisations of cited studies are from reading them, not from re-analysis, and the reader should consult the originals before relying on any specific attribution.

Table 15: Limitations by component, their likely effect on the results and how each could be tested with real data

Component	Assumption or limitation	Likely effect on results	How to test with real data
Degradation	Two Wiener-process modes with shocks; machine-specific thresholds	Determines how predictable failures are; shocks cap achievable lead time	Fit degradation models to condition-monitoring histories; estimate the share of sudden failures
Sensing	Five sensors with fixed functional forms, load confounding, drift and 3% dropout	Sets the gap between rules and learning; real signals may be weaker or richer	Measure signal-to-noise and confounding on pilot machines
Features	Rolling statistics every 6 h; no spectral features	Understates what signal processing can extract	Add spectral and envelope features where vibration is sampled at high rate
Models	LightGBM tuned lightly; no sequence models	Accuracy may be conservative	Benchmark sequence models on the same splits
Policies	Single-component interventions; planning window; no spare-parts or crew constraints	Overstates scheduling flexibility	Add crew, parts and production-schedule constraints
Costs	Stylised repair, intervention and lost-production costs	Drives all value results; sensitivity reported	Use plant accounting data; re-run the value model
Replication	Five fleets; policies tuned on validation, tested once	Low statistical power; wide ranges	More replications; longer windows
Edge/cloud	M/M/1 with rates and prices	Shape of trade-off only	Measure request rates, service times and tariffs
Scope	Only predictive maintenance simulated; other domains argued from literature	Synergy measured in one domain	Replicate the decomposition in supply-chain or energy settings

Future research

Four directions follow. First, replication on real condition-monitoring histories with the same protocol, a chronological split, a tuned condition-based rule and a no-sensor model as baselines, a costed policy simulation and a decomposition against an oracle estimated from hindsight, would test whether the shares of attainable saving found here, roughly two-thirds for sensing with a rule and three-quarters for sensing with learning, hold in practice. Second, the synergy decomposition should be run in a supply-chain or energy setting, where the oracle is a perfect forecast and the policies are replenishment or dispatch rules, to see whether the pattern of complementary, independent contributions recurs. Third, the forecasting side should be extended with spectral features, sequence models and multi-horizon architectures on high-rate data, and with explicit models of the electrical failure mode that was hardest to see here, including current-signature analysis. Fourth, the policy side should add resource constraints, opportunistic grouping of interventions and the costs of alarms on human attention, and should treat the threshold as a control variable that adapts to the state of the maintenance queue, which is where reinforcement learning may have a role. In all four, the testbed and results file released with this article can serve as a starting point and as a check on whether a proposed method improves on the baselines that matter.

Conclusion

AI and IoT are synergistic in business operations in the sense that matters to a manager: each adds substantial value given the other. In a transparent simulation of predictive maintenance on a 96-

machine fleet with hidden, interacting degradation modes and confounded, drifting sensors, learning on IoT streams forecast failures within 72 hours with an AUC of about 0.91, against 0.83 for the best single-sensor rule and 0.75 for a model without sensors, and a maintenance policy driven by its calibrated probabilities cut cost per machine-year by 56% against reactive maintenance, 37% against scheduled maintenance, 18 to 21% against tuned condition-based rules and 39% against an AI-only policy, while raising availability from 99.2 to 99.7%. The claim that the combination exceeds the sum of its parts was not supported: the separate contributions of sensing and learning already approached the ceiling set by a perfect-foresight oracle, and the combination captured about three-quarters of the attainable saving, the share expected from two complementary contributions. What the results do support is a sequence, sensing and a well-tuned rule first, learning on top, with thresholds set against the plant's own costs and with the false-alarm count reported alongside the failure count. The business case is decided by the failure intensity and failure cost of the fleet, the policy already in place and the fleet size, not by the price of sensors, and it is marginal where a competent condition-based programme already exists in a small or reliable fleet. Inference belongs in the cloud for cost and at the edge for bandwidth, latency and autonomy, with a hybrid that extracts features locally as the usual compromise. All of these statements describe a synthetic fleet; their value is that every assumption behind them is disclosed and can be changed, which is more than can be said for most of the percentages that circulate about AI and IoT in business operations.

REFERENCES

[1]. Ahmad, T., Zhang, D., Huang, C., Zhang, H., Dai, N., Song, Y., & Chen, H. (2021). Artificial intelligence in sustainable energy industry: Status quo, challenges and opportunities. *Journal of Cleaner Production*, *289*, Article 125834. https://doi.org/10.1016/j.jclepro.2021.125834

[2]. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, *17*(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

[3]. Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. *Journal of Network and Computer Applications*, *88*, 10–28. https://doi.org/10.1016/j.jnca.2017.04.002

[4]. Alaswad, S., & Xiang, Y. (2017). A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliability Engineering & System Safety*, *157*, 54–63. https://doi.org/10.1016/j.ress.2016.08.009

[5]. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, *8*, 165130–165150. https://doi.org/10.1109/ACCESS.2020.3022862

[6]. Alsyouf, I. (2007). The role of maintenance in improving companies' productivity and profitability. *International Journal of Production Economics*, *105*(1), 70–78. https://doi.org/10.1016/j.ijpe.2004.06.057

[7]. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

[8]. Belhadi, A., Kamble, S., Jabbour, C. J. C., Gunasekaran, A., Ndubisi, N. O., & Venkatesh, M. (2021). Manufacturing and service supply chain resilience to the COVID-19 outbreak: Lessons learned from the automobile and airline industries. *Technological Forecasting and Social Change*, *163*, Article 120447. https://doi.org/10.1016/j.techfore.2020.120447

[9]. Ben-Daya, M., Hassini, E., & Bahroun, Z. (2019). Internet of things and supply chain management: A literature review. *International Journal of Production Research*, *57*(15-16), 4719–4742. https://doi.org/10.1080/00207543.2017.1402140

[10]. Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences, 191*, 192–213. https://doi.org/10.1016/j.ins.2011.12.028

[11]. Bertsimas, D., & Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, *66*(3), 1025–1044. https://doi.org/10.1287/mnsc.2018.3253

[12]. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC '12)* (pp. 13–16). ACM. https://doi.org/10.1145/2342509.2342513

[13]. Botta, A., de Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, *56*, 684–700. https://doi.org/10.1016/j.future.2015.09.021

[14]. Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2018). Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance. *Journal of Intelligent Manufacturing*, *29*(6), 1303–1316. https://doi.org/10.1007/s10845-015-1179-5

[15]. Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. *Computers in Industry*, *101*, 1–12. https://doi.org/10.1016/j.compind.2018.04.015

[16]. Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 1123–1132). IEEE. https://doi.org/10.1109/BigData.2017.8258038

[17]. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

[18]. Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review, 78*(1), 1–3. https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2

[19]. Brintrup, A., Pak, J., Ratiney, D., Pearce, T., Wichmann, P., Woodall, P., & McFarlane, D. (2020). Supply chain data analytics for predicting supplier disruptions: A case study in complex asset manufacturing. *International Journal of Production Research*, *58*(11), 3330–3341. https://doi.org/10.1080/00207543.2019.1685705

[20]. Brynjolfsson, E., Rock, D., & Syverson, C. (2021). The productivity J-curve: How intangibles complement general purpose technologies. *American Economic Journal: Macroeconomics*, *13*(1), 333–372. https://doi.org/10.1257/mac.20180386

[21]. Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. da P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, Article 106024. https://doi.org/10.1016/j.cie.2019.106024

[22]. Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning, 109*(11), 1997–2028. https://doi.org/10.1007/s10994-020-05910-7

[23]. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), Article 15. https://doi.org/10.1145/1541880.1541882

[24]. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

[25]. Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. *Proceedings of the IEEE*, *107*(8), 1655–1674. https://doi.org/10.1109/JPROC.2019.2921977

[26]. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

[27]. Choi, T.-M., Wallace, S. W., & Wang, Y. (2018). Big data analytics in operations management. *Production and Operations Management*, *27*(10), 1868–1883. https://doi.org/10.1111/poms.12838

[28]. Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing*, *307*, 72–77. https://doi.org/10.1016/j.neucom.2018.03.067

[29]. Cohen, M. A., & Kouvelis, P. (2021). Revisit of AAA excellence of global value chains: Robustness, resilience, and realignment. *Production and Operations Management*, *30*(3), 633–643. https://doi.org/10.1111/poms.13305

[30]. Compare, M., Baraldi, P., & Zio, E. (2020). Challenges to IoT-enabled predictive maintenance for Industry 4.0. *IEEE Internet of Things Journal*, *7*(5), 4585–4597. https://doi.org/10.1109/JIOT.2019.2957029

[31]. Cook, A. A., Mısırlı, G., & Fan, Z. (2020). Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal*, *7*(7), 6481–6494. https://doi.org/10.1109/JIOT.2019.2958185

[32]. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

[33]. Dalenogare, L. S., Benitez, G. B., Ayala, N. F., & Frank, A. G. (2018). The expected contribution of Industry 4.0 technologies for industrial performance. *International Journal of Production Economics*, *204*, 383–394. https://doi.org/10.1016/j.ijpe.2018.08.019

[34]. Davenport, T. H., & Ronanki, R. (2018). Artificial intelligence for the real world. *Harvard Business Review*, *96*(1). https://hbr.org/2018/01/artificial-intelligence-for-the-real-world

[35]. Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 233–240). ACM. https://doi.org/10.1145/1143844.1143874

[36]. de Jonge, B., & Scarf, P. A. (2020). A review on maintenance optimization. *European Journal of Operational Research*, *285*(3), 805–824. https://doi.org/10.1016/j.ejor.2019.09.047

[37]. de Vass, T., Shee, H., & Miah, S. J. (2021). IoT in supply chain management: A narrative on retail sector sustainability. *International Journal of Logistics Research and Applications*, *24*(6), 605–624. https://doi.org/10.1080/13675567.2020.1787970

[38]. Dekker, R. (1996). Applications of maintenance optimization models: A review and analysis. *Reliability Engineering & System Safety*, *51*(3), 229–240. https://doi.org/10.1016/0951-8320(95)00076-3

[39]. Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30. https://jmlr.org/papers/v7/demsar06a.html

[40]. Dolgui, A., & Ivanov, D. (2021). Ripple effect and supply chain disruption management: New trends and research directions. *International Journal of Production Research*, *59*(1), 102–109. https://doi.org/10.1080/00207543.2021.1840148

[41]. Dubey, R., Gunasekaran, A., Childe, S. J., Bryde, D. J., Giannakis, M., Foropon, C., Roubaud, D., & Hazen, B. T. (2020). Big data analytics and artificial intelligence pathway to operational performance under the effects of entrepreneurial orientation and environmental dynamism: A study of manufacturing organisations. *International Journal of Production Economics*, *226*, Article 107599. https://doi.org/10.1016/j.ijpe.2019.107599

[42]. Elwany, A. H., & Gebraeel, N. Z. (2008). Sensor-driven prognostic models for equipment replacement and spare parts inventory. *IIE Transactions*, *40*(7), 629–639. https://doi.org/10.1080/07408170701730818

[43]. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

[44]. Fosso Wamba, S., Gunasekaran, A., Akter, S., Ren, S. J.-F., Dubey, R., & Childe, S. J. (2017). Big data analytics and firm performance: Effects of dynamic capabilities. *Journal of Business Research*, *70*, 356–365. https://doi.org/10.1016/j.jbusres.2016.08.009

[45]. Frank, A. G., Dalenogare, L. S., & Ayala, N. F. (2019). Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics*, *210*, 15–26. https://doi.org/10.1016/j.ijpe.2019.01.004

[46]. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

[47]. Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, *8*, 108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358

[48]. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys, 46*(4), Article 44. https://doi.org/10.1145/2523813

[49]. Gebraeel, N. Z., Lawley, M. A., Li, R., & Ryan, J. K. (2005). Residual-life distributions from component degradation signals: A Bayesian approach. *IIE Transactions*, *37*(6), 543–557. https://doi.org/10.1080/07408170590929018

[50]. Ghobakhloo, M. (2018). The future of manufacturing industry: A strategic roadmap toward Industry 4.0. *Journal of Manufacturing Technology Management*, *29*(6), 910–936. https://doi.org/10.1108/JMTM-02-2018-0057

[51]. Gilchrist, A. (2016). *Industry 4.0: The industrial Internet of Things*. Apress. https://doi.org/10.1007/978-1-4842-2047-4

[52]. Govindan, K., Cheng, T. C. E., Mishra, N., & Shukla, N. (2018). Big data analytics and application for logistics and supply chain management. *Transportation Research Part E: Logistics and Transportation Review*, *114*, 343–349. https://doi.org/10.1016/j.tre.2018.03.011

[53]. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

[54]. Gunasekaran, A., Papadopoulos, T., Dubey, R., Fosso Wamba, S., Childe, S. J., Hazen, B., & Akter, S. (2017). Big data and predictive analytics for supply chain and organizational performance. *Journal of Business Research*, *70*, 308–317. https://doi.org/10.1016/j.jbusres.2016.08.004

[55]. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning* (PMLR 70, pp. 1321–1330). https://proceedings.mlr.press/v70/guo17a.html

[56]. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., . . . Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[57]. He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

[58]. Heng, A., Zhang, S., Tan, A. C. C., & Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, *23*(3), 724–739. https://doi.org/10.1016/j.ymssp.2008.06.009

[59]. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[60]. Hosseini, S., Ivanov, D., & Dolgui, A. (2019). Review of quantitative methods for supply chain resilience analysis. *Transportation Research Part E: Logistics and Transportation Review*, *125*, 285–307. https://doi.org/10.1016/j.tre.2019.03.001

[61]. Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in IoT security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, *22*(3), 1686–1721. https://doi.org/10.1109/COMST.2020.2986444

[62]. Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

[63]. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: A review. *Data Mining and Knowledge Discovery*, *33*(4), 917–963. https://doi.org/10.1007/s10618-019-00619-1

[64]. Ivanov, D., & Dolgui, A. (2020). Viability of intertwined supply networks: Extending the supply chain resilience angles towards survivability. A position paper motivated by COVID-19 outbreak. *International Journal of Production Research*, *58*(10), 2904–2915. https://doi.org/10.1080/00207543.2020.1750727

[65]. Ivanov, D., & Dolgui, A. (2021). A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0. *Production Planning & Control*, *32*(9), 775–788. https://doi.org/10.1080/09537287.2020.1768450

[66]. Ivanov, D., & Dolgui, A. (2022). Stress testing supply chains and creating viable ecosystems. *Operations Management Research*, *15*(1-2), 475–486. https://doi.org/10.1007/s12063-021-00194-z

[67]. Ivanov, D., Dolgui, A., & Sokolov, B. (2019). The impact of digital technology and Industry 4.0 on the ripple effect and supply chain risk analytics. *International Journal of Production Research*, *57*(3), 829–846. https://doi.org/10.1080/00207543.2018.1488086

[68]. Ivanov, D. (2017). Simulation-based ripple effect modelling in the supply chain. *International Journal of Production Research*, *55*(7), 2083–2101. https://doi.org/10.1080/00207543.2016.1275873

[69]. Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510. https://doi.org/10.1016/j.ymssp.2005.09.012

[70]. Kagermann, H., Wahlster, W., & Helbig, J. (Eds.). (2013). *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Final report of the Industrie 4.0 Working Group*. acatech – National Academy of Science and Engineering. https://www.acatech.de/publikation/umsetzungsempfehlungen-fuer-das-zukunftsprojekt-industrie-4-0-abschlussbericht-des-arbeitskreises-industrie-4-0/

[71]. Kamble, S. S., Gunasekaran, A., & Gawankar, S. A. (2018). Sustainable Industry 4.0 framework: A systematic literature review identifying the current trends and future perspectives. *Process Safety and Environmental Protection*, *117*, 408–425. https://doi.org/10.1016/j.psep.2018.05.009

[72]. Karkouch, A., Mousannif, H., Al Moatassime, H., & Noël, T. (2016). Data quality in Internet of Things: A state-of-the-art survey. *Journal of Network and Computer Applications*, *73*, 57–81. https://doi.org/10.1016/j.jnca.2016.08.002

[73]. Katsaliaki, K., Galetsi, P., & Kumar, S. (2022). Supply chain disruptions and resilience: A major review and future research agenda. *Annals of Operations Research*, *319*(1), 965–1002. https://doi.org/10.1007/s10479-020-03912-1

[74]. Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data, 6*(4), Article 15. https://doi.org/10.1145/2382577.2382579

[75]. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*, 3146–3154. https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree

[76]. Kleinrock, L. (1975). *Queueing systems, volume 1: Theory*. Wiley. https://openlibrary.org/books/OL5048985M

[77]. Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, *51*(11), 1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474

[78]. Kwon, D., Hodkiewicz, M. R., Fan, J., Shibutani, T., & Pecht, M. G. (2016). IoT-based prognostics and systems health management for industrial applications. *IEEE Access*, *4*, 3659–3670. https://doi.org/10.1109/ACCESS.2016.2587754

[79]. Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, *6*(4), 239–242. https://doi.org/10.1007/s12599-014-0334-4

[80]. Law, A. M. (2015). *Simulation modeling and analysis* (5th ed.). McGraw-Hill Education. https://www.mheducation.com/

[81]. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

[82]. Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001

[83]. Lee, J., Davari, H., Singh, J., & Pandhare, V. (2018). Industrial artificial intelligence for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, *18*, 20–23. https://doi.org/10.1016/j.mfglet.2018.09.002

[84]. Lee, J. (2020). *Industrial AI: Applications with sustainable performance*. Springer. https://doi.org/10.1007/978-981-15-2144-7

[85]. Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834. https://doi.org/10.1016/j.ymssp.2017.11.016

[86]. Li, H., Ota, K., & Dong, M. (2018). Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE Network*, *32*(1), 96–101. https://doi.org/10.1109/MNET.2018.1700202

[87]. Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

[88]. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering, 31*(12), 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857

[89]. Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, *6*, 1–10. https://doi.org/10.1016/j.jii.2017.04.005

[90]. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*, 4765–4774. https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions

[91]. Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence, 2*(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

[92]. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, *38*(4), 1346–1364. https://doi.org/10.1016/j.ijforecast.2021.11.013

[93]. Manavalan, E., & Jayakrishna, K. (2019). A review of Internet of Things (IoT) embedded sustainable supply chain for Industry 4.0 requirements. *Computers & Industrial Engineering*, *127*, 925–953. https://doi.org/10.1016/j.cie.2018.11.030

[94]. Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, *19*(4), 2322–2358. https://doi.org/10.1109/COMST.2017.2745201

[95]. Matzka, S. (2020). Explainable artificial intelligence for predictive maintenance applications. In *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)* (pp. 69–74). IEEE. https://doi.org/10.1109/AI4I49448.2020.00023

[96]. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, *17*(3), 12–22. https://doi.org/10.1109/MPRV.2018.03367731

[97]. Mikalef, P., Boura, M., Lekakos, G., & Krogstie, J. (2019). Big data analytics and firm performance: Findings from a mixed-method approach. *Journal of Business Research*, *98*, 261–276. https://doi.org/10.1016/j.jbusres.2019.01.044

[98]. Minoli, D., Sohraby, K., & Occhiogrosso, B. (2017). IoT considerations, requirements, and architectures for smart buildings—Energy optimization and next-generation building management systems. *IEEE Internet of Things Journal*, *4*(1), 269–283. https://doi.org/10.1109/JIOT.2017.2647881

[99]. Mišić, V. V., & Perakis, G. (2020). Data analytics in operations management: A review. *Manufacturing & Service Operations Management*, *22*(1), 158–169. https://doi.org/10.1287/msom.2019.0805

[100]. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (pp. 220–229). ACM. https://doi.org/10.1145/3287560.3287596

[101]. Mittal, S., Khan, M. A., Romero, D., & Wuest, T. (2018). A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs). *Journal of Manufacturing Systems*, *49*, 194–214. https://doi.org/10.1016/j.jmsy.2018.10.005

[102]. Mobley, R. K. (2002). *An introduction to predictive maintenance* (2nd ed.). Butterworth-Heinemann. https://openlibrary.org/isbn/9780750675314

[103]. Moeuf, A., Pellerin, R., Lamouri, S., Tamayo-Giraldo, S., & Barbaray, R. (2018). The industrial management of SMEs in the era of Industry 4.0. *International Journal of Production Research*, *56*(3), 1118–1136. https://doi.org/10.1080/00207543.2017.1372647

[104]. Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, *20*(4), 2923–2960. https://doi.org/10.1109/COMST.2018.2844341

[105]. Mourtzis, D., Vlachou, E., & Milas, N. (2016). Industrial big data as a result of IoT adoption in manufacturing. *Procedia CIRP*, *55*, 290–295. https://doi.org/10.1016/j.procir.2016.07.038

[106]. Muchiri, P., & Pintelon, L. (2008). Performance measurement using overall equipment effectiveness (OEE): Literature review and practical application discussion. *International Journal of Production Research*, *46*(13), 3517–3535. https://doi.org/10.1080/00207540601142645

[107]. Nadeau, C., & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, *52*(3), 239–281. https://doi.org/10.1023/A:1024068626366

[108]. Naeini, M. P., Cooper, G. F., & Hauskrecht, M. (2015). Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (pp. 2901–2907). AAAI Press. https://doi.org/10.1609/aaai.v29i1.9602

[109]. Nakajima, S. (1988). *Introduction to TPM: Total productive maintenance*. Productivity Press. https://openlibrary.org/books/OL2064937M

[110]. Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *23*(3), 1622–1658. https://doi.org/10.1109/COMST.2021.3075439

[111]. Nguyen, K.-A., Do, P., & Grall, A. (2015). Multi-level predictive maintenance for multi-component systems. *Reliability Engineering & System Safety*, *144*, 83–94. https://doi.org/10.1016/j.ress.2015.07.017

[112]. Nguyen, T., Zhou, L., Spiegler, V., Ieromonachou, P., & Lin, Y. (2018). Big data analytics in supply chain management: A state-of-the-art literature review. *Computers & Operations Research*, *98*, 254–264. https://doi.org/10.1016/j.cor.2017.07.004

[113]. Nunes, P., Santos, J., & Rocha, E. (2023). Challenges in predictive maintenance – A review. *CIRP Journal of Manufacturing Science and Technology*, *40*, 53–67. https://doi.org/10.1016/j.cirpj.2022.11.004

[114]. Papadopoulos, T., Gunasekaran, A., Dubey, R., Altay, N., Childe, S. J., & Fosso-Wamba, S. (2017). The role of big data in explaining disaster resilience in supply chains for sustainability. *Journal of Cleaner Production*, *142*, 1108–1118. https://doi.org/10.1016/j.jclepro.2016.03.059

[115]. Patki, N., Wedge, R., & Veeramachaneni, K. (2016). The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 399–410). IEEE. https://doi.org/10.1109/DSAA.2016.49

[116]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html

[117]. Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: A review. *The International Journal of Advanced Manufacturing Technology*, *50*(1-4), 297–313. https://doi.org/10.1007/s00170-009-2482-0

[118]. Platt, J. C. (2000). Probabilities for SV machines. In A. J. Smola, P. Bartlett, B. Schölkopf, & D. Schuurmans (Eds.), *Advances in large margin classifiers* (pp. 61–74). MIT Press. https://doi.org/10.7551/mitpress/1113.003.0008

[119]. Porter, M. E., & Heppelmann, J. E. (2014). How smart, connected products are transforming competition. *Harvard Business Review*, *92*(11). https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition

[120]. Porter, M. E., & Heppelmann, J. E. (2015). How smart, connected products are transforming companies. *Harvard Business Review*, *93*(10). https://hbr.org/2015/10/how-smart-connected-products-are-transforming-companies

[121]. Qi, Q., & Tao, F. (2018). Digital twin and big data towards smart manufacturing and Industry 4.0: 360 degree comparison. *IEEE Access*, *6*, 3585–3593. https://doi.org/10.1109/ACCESS.2018.2793265

[122]. Queiroz, M. M., Ivanov, D., Dolgui, A., & Fosso Wamba, S. (2022). Impacts of epidemic outbreaks on supply chains: Mapping a research agenda amid the COVID-19 pandemic through a structured literature review. *Annals of Operations Research*, *319*(1), 1159–1196. https://doi.org/10.1007/s10479-020-03685-7

[123]. Rolf, B., Jackson, I., Müller, M., Lang, S., Reggelin, T., & Ivanov, D. (2023). A review on reinforcement learning algorithms and applications in supply chain management. *International Journal of Production Research*, *61*(20), 7151–7179. https://doi.org/10.1080/00207543.2022.2140221

[124]. Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, *78*, 680–698. https://doi.org/10.1016/j.future.2016.11.009

[125]. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x

[126]. Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE, 10*(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

[127]. Sargent, R. G. (2013). Verification and validation of simulation models. *Journal of Simulation*, *7*(1), 12–24. https://doi.org/10.1057/jos.2012.20

[128]. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, *50*(1), 30–39. https://doi.org/10.1109/MC.2017.9

[129]. Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management* (pp. 1–9). IEEE. https://doi.org/10.1109/PHM.2008.4711414

[130]. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems, 28*, 2503–2511. https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

[131]. Seyedan, M., & Mafakheri, F. (2020). Predictive big data analytics for supply chain demand forecasting: Methods, applications, and research opportunities. *Journal of Big Data*, *7*(1), Article 53. https://doi.org/10.1186/s40537-020-00329-2

[132]. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

[133]. Shrouf, F., Ordieres, J., & Miragliotta, G. (2014). Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 697–701). IEEE. https://doi.org/10.1109/IEEM.2014.7058728

[134]. Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). Remaining useful life estimation – A review on the statistical data driven approaches. *European Journal of Operational Research*, *213*(1), 1–14. https://doi.org/10.1016/j.ejor.2010.11.018

[135]. Si, X.-S., Wang, W., Hu, C.-H., Chen, M.-Y., & Zhou, D.-H. (2013). A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. *Mechanical Systems and Signal Processing*, *35*(1-2), 219–237. https://doi.org/10.1016/j.ymssp.2012.08.016

[136]. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, *76*, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008

[137]. Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial Internet of Things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, *14*(11), 4724–4734. https://doi.org/10.1109/TII.2018.2852491

[138]. Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, *11*(3), 812–820. https://doi.org/10.1109/TII.2014.2349359

[139]. Swanson, L. (2001). Linking maintenance strategies to performance. *International Journal of Production Economics*, *70*(3), 237–244. https://doi.org/10.1016/S0925-5273(00)00067-0

[140]. Syntetos, A. A., Babai, M. Z., Boylan, J. E., Kolassa, S., & Nikolopoulos, K. (2016). Supply chain forecasting: Theory, practice, their gap and the future. *European Journal of Operational Research*, *252*(1), 1–26. https://doi.org/10.1016/j.ejor.2015.11.010

[141]. Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, *19*(3), 1657–1681. https://doi.org/10.1109/COMST.2017.2705720

[142]. Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2019). Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, *15*(4), 2405–2415. https://doi.org/10.1109/TII.2018.2873186

[143]. Tiwari, S., Wee, H. M., & Daryanto, Y. (2018). Big data analytics in supply chain management between 2010 and 2016: Insights to industries. *Computers & Industrial Engineering*, *115*, 319–330. https://doi.org/10.1016/j.cie.2017.11.017

[144]. Tortorella, G. L., & Fettermann, D. (2018). Implementation of Industry 4.0 and lean production in Brazilian manufacturing companies. *International Journal of Production Research*, *56*(8), 2975–2987. https://doi.org/10.1080/00207543.2017.1391420

[145]. Tsang, A. H. C. (2002). Strategic dimensions of maintenance management. *Journal of Quality in Maintenance Engineering*, *8*(1), 7–39. https://doi.org/10.1108/13552510210420577

[146]. Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. Wiley. https://doi.org/10.1002/9780470117842

[147]. Van Horenbeek, A., & Pintelon, L. (2013). A dynamic predictive maintenance policy for complex multi-component systems. *Reliability Engineering & System Safety*, *120*, 39–50. https://doi.org/10.1016/j.ress.2013.02.029

[148]. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[149]. Wang, G., Gunasekaran, A., Ngai, E. W. T., & Papadopoulos, T. (2016a). Big data analytics in logistics and supply chain management: Certain investigations for research and applications. *International Journal of Production Economics*, *176*, 98–110. https://doi.org/10.1016/j.ijpe.2016.03.014

[150]. Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research*, *139*(3), 469–489. https://doi.org/10.1016/S0377-2217(01)00197-7

[151]. Wang, S., Wan, J., Li, D., & Zhang, C. (2016b). Implementing smart factory of Industrie 4.0: An outlook. *International Journal of Distributed Sensor Networks*, *12*(1), Article 3159805. https://doi.org/10.1155/2016/3159805

[152]. Weber, R. H. (2010). Internet of Things – New security and privacy challenges. *Computer Law & Security Review*, *26*(1), 23–30. https://doi.org/10.1016/j.clsr.2009.11.008

[153]. Wortmann, F., & Flüchter, K. (2015). Internet of Things: Technology and value added. *Business & Information Systems Engineering*, *57*(3), 221–224. https://doi.org/10.1007/s12599-015-0383-3

[154]. Wuest, T., Weimer, D., Irgens, C., & Thoben, K.-D. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. *Production & Manufacturing Research*, *4*(1), 23–45. https://doi.org/10.1080/21693277.2016.1192517

[155]. Xu, L. D., He, W., & Li, S. (2014). Internet of Things in industries: A survey. *IEEE Transactions on Industrial Informatics*, *10*(4), 2233–2243. https://doi.org/10.1109/TII.2014.2300753

[156]. Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., & Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, *98*, 289–330. https://doi.org/10.1016/j.sysarc.2019.02.009

[157]. Zhang, C., Patras, P., & Haddadi, H. (2019a). Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials*, *21*(3), 2224–2287. https://doi.org/10.1109/COMST.2019.2904897

[158]. Zhang, W., Yang, D., & Wang, H. (2019b). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, *13*(3), 2213–2227. https://doi.org/10.1109/JSYST.2019.2905565

[159]. Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, *115*, 213–237. https://doi.org/10.1016/j.ymssp.2018.05.050

[160]. Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent manufacturing in the context of Industry 4.0: A review. *Engineering*, *3*(5), 616–630. https://doi.org/10.1016/j.eng.2017.05.015

[161]. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, *107*(8), 1738–1762. https://doi.org/10.1109/JPROC.2019.2918951

[162]. Zhu, T., Ran, Y., Zhou, X., & Wen, Y. (2025). A survey on intelligent predictive maintenance (IPdM) in the era of fully connected intelligence. *IEEE Communications Surveys & Tutorials*, *28*, 633–671. https://doi.org/10.1109/COMST.2025.3567802

[163]. Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*, Article 106889. https://doi.org/10.1016/j.cie.2020.106889