GJR Publication
OPEN ACCESS
Global Journal of Research in Engineering & Computer Sciences
ISSN: 2583-2727 (Online)
Volume 06 | Issue 04 | July-August | 2026
Journal homepage: https://gjrpublication.com/gjrecs/
Research Article

Industrial AI Lifecycle Debt: An Event-Based Model for Drift, Revalidation, and Sustainment in IT/OT Environments

*Daniel Ward

Department of Computer Science, Southern New Hampshire University, Manchester, New Hampshire, United States.

DOI: 10.5281/zenodo.21949424 **Submission Date: 20 June 2026 | Published Date: 15 Aug. 2026**

Abstract

Industrial artificial intelligence investment cases often price acquisition, integration, compute, and expected operational benefit while treating post-deployment sustainment as an implementation detail. This paper develops the Industrial AI Lifecycle Assurance Cost (AILAC), Total Lifecycle Burden (TLB), and the Industrial AI Lifecycle Debt Premium (AILDP) to estimate event-triggered costs of preserving model validity, safety, cybersecurity, recoverability, and usefulness in information technology/operational technology environments. An integrative evidence synthesis was combined with conceptual model development, six deterministic scenarios, break-even analysis, and a Monte Carlo experiment using a randomly generated seed. Costs were expressed as five-year discounted present values using an illustrative 8% rate and base-year revenue normalization of 100. Across 20,000 simulations, median AILAC was 1.933 units, median TLB was 2.331, and median break-even benefit was 3.959; the 5th-95th percentile interval for break-even benefit was 2.839-5.631. Event-response cost, event frequency, and control authority had the strongest positive sensitivity relationships with TLB, while operational maturity reduced burden. Results demonstrate model behavior under stated assumptions rather than empirical industry cost rates. The contribution is an architecture- and authority-sensitive framework that treats drift, sensor and process change, pipeline updates, supplier change, cybersecurity reassessment, validation, release, and fallback as permanent investment obligations.

Keywords: *Industrial Artificial Intelligence; Operational Technology; Model Drift; MLOps; Lifecycle Assurance; Technical Debt; IT/OT Convergence.*

1. INTRODUCTION

Industrial organizations increasingly apply artificial intelligence and machine learning to predictive maintenance, quality inspection, energy management, anomaly detection, scheduling, and operator decision support. The investment case is usually written around acquisition cost, integration effort, compute, and a projected operational benefit. Once the model reaches production, however, it becomes coupled to a physical measurement system, a data pipeline, an operating envelope, a software runtime, a set of users, and frequently an external supplier. Those dependencies continue to change after commissioning. Sensors are replaced or recalibrated, equipment wears, recipes and product mixes change, historian schemas evolve, model runtimes are patched, vendors deprecate interfaces, and the organization changes the amount of authority granted to the model.

Machine-learning systems, therefore, create a different kind of lifecycle obligation from conventional deterministic software. Sculley et al. described how apparently simple models can conceal system-level coupling, data dependencies, feedback loops, configuration debt, and external-world change [1]. Production studies subsequently showed that operationalizing ML requires recurring data preparation, experimentation, deployment evaluation, monitoring, and cross-functional response rather than a one-time handoff from data science to operations [2]. In manufacturing, reference architectures and maintenance methodologies similarly treat monitoring, retraining, deployment, and model management as continuing functions [3], [4].

The problem is especially consequential in information technology/operational technology (IT/OT) environments. An inaccurate recommendation can waste engineering time; a stale quality model can increase scrap; an unavailable cloud inference service can remove a relied-upon decision aid; and an autonomous model can influence the physical process

before a person or independent safety barrier can intervene. The relevant cost is not only the compute required to retrain a model. It includes diagnosis, data qualification, process-owner review, regression testing, hardware-in-the-loop or shadow-mode evaluation, cybersecurity reassessment, release approval, rollback readiness, fallback operation, operator training, and evidence that the complete model-data-runtime-interface bundle remains valid.

Existing research documents these components but does not provide a transferable industrial total-cost curve. Drift studies establish that model performance can decline under changing process, sensor, and environmental conditions, but they rarely translate degradation into realized plant-level profit and loss. MLOps research identifies recurring work but does not provide a stable full-time-equivalent ratio per industrial model. Technical-debt studies provide rubrics and observable indicators rather than a universal monetary interest rate. Economic work models predictive-maintenance payoffs, stale-model loss, retraining cost, or update timing, but usually omit the full OT validation, safety, cybersecurity, and release burden.

This paper addresses that integration gap. It distinguishes Industrial AI Lifecycle Assurance Cost (AILAC), the recurring and event-triggered cost required to keep an AI capability valid and supportable, from Industrial AI Lifecycle Debt (AILD), the accumulated future liability created when required assurance is deferred or when architectural coupling, undocumented dependencies, weak fallback, or supplier lock-in make later change more expensive. It then defines Total Lifecycle Burden (TLB) as assurance cost plus residual lifecycle exposure and defines the Industrial AI Lifecycle Debt Premium (AILDP) as the incremental burden of a proposed architecture and authority level relative to a less-dependent feasible counterfactual.

The study asks four questions. First, which recurring and event-triggered burdens are documented for industrial AI after deployment? Second, how can process volatility, dependency coupling, supplier and connectivity dependence, AI control authority, and operational maturity be represented in a transparent lifecycle model? Third, how do representative architecture-authority scenarios change AILAC, TLB, and the break-even benefit required to justify deployment? Fourth, which model inputs have the strongest sensitivity relationships with lifecycle burden? The contribution is not a claim that the illustrative coefficients estimate an industry-wide cost rate. It is a reproducible decision structure that makes permanent assurance obligations visible before an industrial AI investment is approved.

2. RELATED LITERATURE AND RESEARCH GAP

2.1 Drift, Measurement, and Industrial Validity

The strongest direct industrial evidence concerns degradation under changing conditions. Jourdan et al. analyzed a manufacturing dataset spanning 36 months and found that concept and sensor drift reduced accuracy across the evaluated algorithms [5]. In a later CNC process-monitoring study, changed environmental, equipment, and operating conditions materially degraded deep-learning performance; even a simulated change in sensor orientation produced substantial deterioration [6]. These findings show that a model remains valid only relative to a measurement configuration and operating context, not merely to a stored model artifact.

Operational studies also demonstrate that drift can carry decision value before an error occurs. Soller et al. used production-machine sensor streams and reported that feature-drift analysis detected machine alarms approximately 15 minutes earlier on average [7]. Chen et al. found that corrosion-induced data drift in aging-bridge monitoring reduced damage-classification accuracy from about 90% to 80% at the 75-year horizon, while an updated model restored accuracy to 96% [8]. These studies support monitoring and adaptation, but neither supplies a general manufacturing dollar-loss curve. The evidence is strongest for performance degradation and early warning potential, not for realized financial loss or observed safety incidents.

Physical measurement dependencies are equally important. Martins et al. emphasized metrological traceability and online monitoring of sensor calibration status [9]. Husom et al. operationalized recurring comparison of manufacturing sensor data against a reference production cycle [10]. Digital pipeline research reaches the same conclusion on the software side: persisted datasets, feature transformations, and downstream stages cannot be assumed to be compatible after schema or parameter changes [11], and continuous data validation must be treated as a first-class production function [12]. A sensor replacement, an altered sampling rate, a revised unit conversion, a historian tag change, a feature update, a new product recipe, or an operating mode can therefore trigger targeted revalidation even when the model weights have not changed.

2.2 Sustainment Workload and Hidden Technical Debt

Hidden ML debt broadens conventional software debt from code quality to the model-data-pipeline-decision system. Sculley et al. identified entanglement, undeclared consumers, feedback loops, data dependencies, configuration, and external-world change as recurring liabilities [1]. Breck et al. translated many of these concerns into a production-readiness rubric covering data tests, model tests, infrastructure, and monitoring [13]. Later empirical work found that debt patterns emerge during prototyping and persist through configuration and model-code duplication. At the same time,

industrial-practitioner research proposed debt-management processes and documentation artifacts rather than a universal dollar conversion [14].

The labor evidence is similarly component-based. Production engineers repeatedly prepare and inspect data, compare experiments, evaluate staged deployments, monitor behavior, and coordinate with product and operational stakeholders [2]. Manufacturing reference architectures place data acquisition, model development, deployment, monitoring, and feedback in a continuous operational loop [3]. Dreyfus et al. frame industrial model maintenance as drift identification, sampling, and update selection under cost constraints [4]. These activities require data scientists, controls and process engineers, operators, maintenance personnel, cybersecurity staff, quality or safety personnel, and suppliers at different points in the lifecycle.

The burden can be nonlinear because one physical or digital change may invalidate several coupled artifacts. A model update may also require a feature-definition change, a container or runtime update, a dashboard revision, an alert-routing change, an access review, and regression testing of downstream maintenance or operator workflows. Cody et al. showed that ML faults in a cyber-physical production system can propagate into nonlinear throughput effects even when transfer learning reduces local degradation [15]. The proper unit of validation is therefore the compatible release bundle and its operating context, not a serialized model file alone.

2.3 Architecture, Supplier Dependence, and Control Authority

Deployment topology changes where lifecycle work is performed do not eliminate the work. Cloud ModelOps can centralize traceability, reproducibility, training, and quality controls, but the industrial application then depends on service availability, identity, APIs, connectivity, storage, model versions, and release paths [16]. Platform-agnostic industrial MLOps research treats portability across edge, fog, and cloud as an engineering objective rather than a default property [17]. Far-edge work shows that network constraints require explicit mechanisms for distribution, observability, and remote lifecycle management [18].

The resulting placement choice is a tradeoff. Cloud centralizes heavy training and fleet operations but introduces dependencies on providers, WAN, latency, and continuity. Edge and disconnected designs preserve local latency and operational capabilities while distributing hardware, observability, and release responsibilities. Hybrid designs can keep time-bounded inference local while centralizing training and fleet visibility, but add an integration and synchronization layer. Secure-connectivity guidance for OT reinforces the idea that remote paths, identities, monitoring, isolation, and continuity must be designed as lifecycle responsibilities rather than as assumed infrastructure [19]. Vendor hosting may reduce owned compute while replacing it with recurring service, integration, supplier-governance, revalidation, and exit costs.

Control authority is a separate dimension. NIST frames trustworthy AI as a lifecycle responsibility involving validity, reliability, safety, security, resilience, accountability, and monitoring [20]. Joint guidance for AI in OT calls for safe operating bounds, continuous performance and anomaly monitoring, progressively realistic test environments, explicit fallback to non-AI operation, and defined responsibilities among operators, vendors, and governance functions [21]. Advisory systems leave actuation with the operator. Human-approved systems add authorization and timing dependencies. Supervisory systems require assurance of the model, an independent monitor, operating envelope, transfer logic, and fallback. Autonomous systems require the strongest evidence because the monitor-decision-controller-fallback chain may act before a person can intervene. Runtime-assurance research, such as AMLOP and RADICS, provides technical foundations for monitoring and backup control, but not for the overall industrial lifecycle cost [22], [23].

2.4 Partial Economic Models and the Integration Gap

Several quantitative models address individual slices of the problem. Florian et al. compare preventive and ML-enabled predictive maintenance using reliability, intervention and failure costs, sensor and IT investment, and classifier performance [24]. Mahadevan and Mathioudakis formalize a keep-versus-retrain decision as a tradeoff between stale-model loss and retraining cost [25]. Kanda et al. compare periodic and detector-triggered retraining under degradation penalties [26]. Guissouma et al. add a shadow-mode safety gate for updated supervised models [27]. Cruickshank and Kohtz estimate sustainment based on maintenance tasks, difficulty, frequency, and support arrangements [28].

These models are useful but do not optimize the whole industrial decision stack. Compute-only retraining cost omits diagnosis, data qualification, subject-matter-expert review, hardware-in-the-loop testing, cybersecurity review, release approval, rollback, and production continuity. Predictive-maintenance economics evaluates asset outcomes but not the recurring costs of maintaining the AI capability. Touch-time methods estimate labor but require a site-specific task inventory. The literature, therefore, supports a component-based, event-driven model while also establishing an important evidence boundary: there is no defensible universal annual maintenance percentage, full-time-equivalent ratio, technical-debt multiplier, or retraining cadence that travels across industrial sites.

Table 1 summarizes how the retained evidence streams are used in the proposed model. The gap is not that drift, MLOps, technical debt, or AI assurance are unknown. It is that their costs and dependencies remain fragmented across engineering, safety, cybersecurity, supplier, workforce, and economic literatures.

Table 1: Evidence streams and their role in model development

Evidence stream	Documented finding and limit	Use in this study
Industrial drift and measurement	Performance degrades under concept, sensor, equipment, and environmental change; universal thresholds and dollar losses are not available.	Represent event frequency, residual consequence, process/data volatility, and dependency-triggered revalidation.
Production MLOps and workload	Monitoring, diagnosis, retraining, staged deployment, and cross-functional coordination recur after release; no stable industrial FTE benchmark was found.	Separate fixed annual assurance workload from event-triggered response cost.
Hidden technical debt	Debt exists across data, model, pipeline, infrastructure, configuration, documentation, and organizational ownership; no universal monetary conversion exists.	Treat debt as deferred or avoidable liability and track it through an explicit register.
Architecture and supplier dependence	Cloud, edge, hybrid, and disconnected patterns redistribute rather than remove lifecycle work.	Model dependency coupling and supplier/connectivity dependence as independent burden multipliers.
Control authority and runtime assurance	Assurance expands as AI moves from recommendation to direct process influence.	Scale recurring burden, event-response cost, and conditional consequence by authority.
Economic and update models	Existing models price maintenance outcomes, stale-model loss, retraining, inspection, or labor separately.	Integrate the components into AILAC, TLB, AILDP, and a full-cost update decision rule.

3. MATERIALS AND METHODS

3.1 Evidence Discovery and Verification

This study used an integrative evidence synthesis rather than a statistical meta-analysis. Ten targeted search blocks were executed through Elicit, whose discovery layer uses scholarly indexes including Semantic Scholar and OpenAlex. The searches addressed: (1) industrial AI lifecycle cost and risk; (2) quantitative lifecycle-cost and assurance models; (3) empirical consequences of model drift; (4) optimization of update and maintenance timing; (5) hidden technical debt; (6) sensor, process, schema, and feature dependencies that trigger revalidation; (7) cloud, edge, hybrid, and disconnected lifecycle tradeoffs; (8) assurance requirements by AI control authority; (9) recurring industrial AI sustainment workload; and (10) vendor-hosted industrial AI dependencies.

Eligible sources included English-language peer-reviewed empirical studies, methods papers, lifecycle and economic models, relevant preprints when a peer-reviewed substitute was not available, and official government guidance on AI or OT lifecycle assurance. Vendor marketing material, unsupported opinion, generic consumer-AI discussions, and model-accuracy studies without lifecycle relevance were excluded. The final synthesis retained 29 sources: 23 peer-reviewed or in-press research items, two preprints, and four official or technical reports. Original-source metadata and the specific claim used in the manuscript were checked against DOI records, publisher pages, official documents, or accessible source text. The accompanying source register identifies source type, verification basis, and claim use. Because the search was designed to build and bound a decision model, the synthesis should not be interpreted as an exhaustive systematic review of every industrial AI publication.

3.2 Model Development Procedure

Model development proceeded in five stages. First, recurring and event-triggered cost categories were extracted from the retained sources. Second, physical, digital, supplier, authority, and organizational dependencies were grouped into five normalized dimensions: process/data volatility (v), dependency coupling (d), supplier/connectivity dependence (s), AI control authority (u), and operational maturity (m). Third, the cost structure was separated into fixed assurance workload, material lifecycle-event frequency, complete event-response cost, residual consequence probability, and conditional consequence. Fourth, equations were constructed to preserve monotonic relationships supported by the literature: greater volatility, coupling, supplier dependence, and authority increase burden, while greater operational maturity reduces it. Fifth, the model was evaluated through deterministic scenarios, break-even analysis, and Monte Carlo sensitivity analysis.

The coefficient mappings are transparent analytical assumptions, not regression estimates. Their purpose is to test the behavior of the proposed structure over bounded inputs and to show where site data must be substituted. No human-subject data, proprietary plant records, or live control-system telemetry were collected.

3.3 Horizon, Units, and Accounting Boundaries

The illustrative evaluation uses a five-year horizon (H = 5), an annual discount rate of 8%, and base-year annual revenue R0 = 100. A normalized monetary value of 1.0, therefore, equals 1% of base-year annual revenue. Fixed annual assurance cost and annual expected residual loss are converted to present value using the end-of-year annuity factor in Equation (1). Implementation cost (Ci) is treated as a time-zero amount. Routine operating cost (Co) is provided as a five-year, already-discounted value covering predictable compute, storage, licenses, and ordinary platform operations.

Accounting boundaries prevent double-counting. AILAC includes recurring monitoring and assurance work, as well as the expected costs of diagnosis, remediation, validation, release, and recovery following lifecycle events. Residual lifecycle exposure (RLE) includes the expected consequences of stale, unavailable, compromised, or erroneous behavior after controls are in place. It does not include labor already counted in AILAC. AILDP compares the total lifecycle burden with a feasible reference architecture. The prior Convergence Exposure Premium model separately prices incremental cyber loss attributable to IT/OT connectivity [29]; the present model focuses on maintaining AI validity and assurance and should be integrated with cyber-loss models only when overlapping terms are removed.

$$A_H(r) = \sum_{t=1}^{H} (1+r)^{-t} \quad (1)$$

3.4 Scenario and Monte Carlo Design

Six deterministic scenarios combine different deployment architectures and authority levels: disconnected local advisory, read-only on-premises advisory, hybrid edge-cloud advisory, vendor-hosted human-approved, local bounded supervisory, and high-volatility autonomous. Scenario values are illustrative and are intended to exercise the model across increasing dependency and authority, not to rank named products or estimate an industry average.

The Monte Carlo experiment sampled 12 primitive inputs independently from bounded triangular distributions because the literature does not supply transferable empirical distributions. The experiment used 20,000 runs, Python 3.13.5, NumPy 2.3.5 default_rng with PCG64, and a randomly generated, fixed 64-bit seed (0x5D2CA2CC0D9F3664). No exogenous correlation matrix was imposed. Dependence arises only through the explicit equations, so a single draw of maturity, for example, affects fixed assurance, event frequency, event cost, and residual consequence. Pearson product-moment correlations between primitive inputs and TLB were calculated as univariate sensitivity indicators with two-sided 95% confidence intervals using Fisher's z transformation. Spearman's rank correlation was calculated as a robustness check. These statistics are not causal coefficients or variance-decomposition measures.

Table 2: Monte Carlo primitive input assumptions

Input	Triangular min/mode/max	Unit	Rationale
Process/data volatility (v)	0.200/0.550/0.900	Dimensionless 0-1	Process and data volatility
Dependency coupling (d)	0.150/0.550/0.900	Dimensionless 0-1	Coupling among sensor, pipeline, model, runtime, and consumers
Supplier/connectivity dependence (s)	0.000/0.350/0.900	Dimensionless 0-1	External connectivity and supplier dependence
AI control authority (u)	0.000/0.250/0.850	Dimensionless 0-1	Advisory through control-authorized operation
Operational maturity (m)	0.300/0.650/0.900	Dimensionless 0-1	Monitoring, ownership, validation, fallback, and change-control maturity
Baseline fixed assurance cost (F0)	0.040/0.100/0.180	Annual normalized cost	Baseline fixed lifecycle-assurance workload
Baseline lifecycle-event rate (lambda0)	0.200/0.550/0.900	Events per year	Baseline material lifecycle-event frequency
Baseline cost per event (C0)	0.080/0.220/0.450	Normalized cost per event	Diagnosis, remediation, validation, release, and recovery
Baseline residual consequence probability (q0)	0.005/0.025/0.060	Annual probability	Baseline residual consequential error/unavailability probability
Baseline conditional consequence (L0)	0.400/1.000/2.500	Normalized conditional loss	Consequence before control-authority scaling
Implementation and initial validation (Ci)	0.500/1.000/1.800	Time-zero normalized cost	Implementation and initial validation
Routine operating cost (Co)	0.150/0.450/0.900	Five-year PV normalized cost	Routine compute, licenses, storage, and platform operations

4. INDUSTRIAL AI LIFECYCLE ASSURANCE MODEL

4.1 Construct Definitions

Industrial AI Lifecycle Assurance Cost is the present value of the fixed and event-triggered resources required to preserve model validity, safety, cybersecurity, recoverability, and operational usefulness. Fixed assurance includes monitoring, data and sensor stewardship, evidence retention, supplier review, fallback exercises, and competency maintenance. Event-triggered assurance begins when drift, a change in a physical asset, a pipeline or schema change, a model or runtime update, a supplier change, a new cyber threat, or an ownership change requires investigation.

Industrial AI Lifecycle Debt is narrower. It is the future liability that arises when required assurance work is deferred, dependencies are undocumented, release bundles are not reproducible, fallbacks are weak, or architectural coupling and supplier lock-in make later changes more difficult. AILAC estimates the cost of performing the work; the debt register records the liabilities created when the work or enabling controls are absent. The Industrial AI Lifecycle Debt Premium is a prospective architectural comparison: it measures how much greater the total lifecycle burden the proposed design carries than that of the least-dependent feasible design for the same operational outcome.

Residual Lifecycle Exposure represents the expected consequence that remains after the planned assurance controls have been implemented. It includes false actions, missed conditions, stale recommendations, loss of inference service, quality or throughput effects, and other asset-specific consequences. Total Lifecycle Burden is AILAC plus RLE. Figure 1 shows the event-triggered assurance workflow.

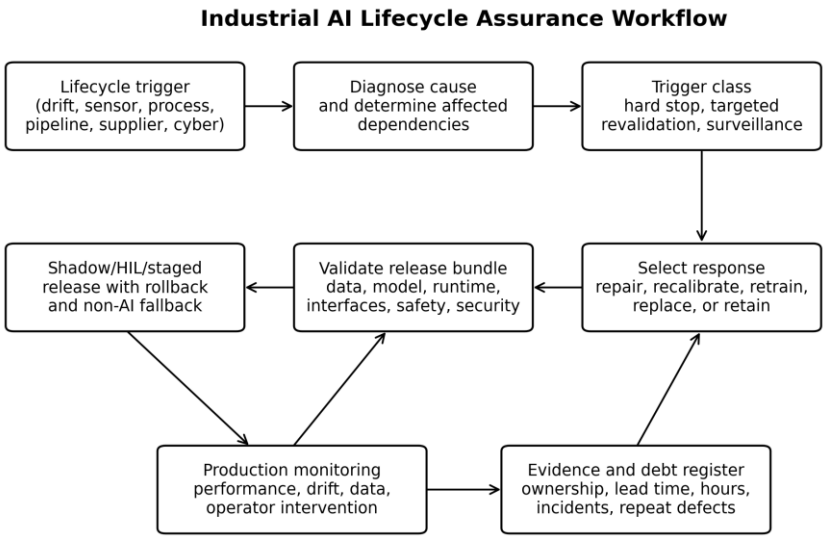

Figure 1: Event-triggered industrial AI assurance workflow. A lifecycle trigger is diagnosed and classified before repair, recalibration, retraining, replacement, or retention. The validated release bundle is introduced through shadow, hardware-in-the-loop, or staged deployment with rollback and a non-AI fallback. Monitoring and the debt register close the lifecycle loop.

4.2 Trigger Hierarchy and Required Response

Drift detection is a signal, not an automatic retraining command. The first task is to distinguish among a sensor or calibration fault, a data-pipeline break, a process change, a cyber event, and a true change in the input-outcome relationship. Table 3 converts the dependency evidence into a practical trigger hierarchy. Thresholds remain asset-specific because the literature does not supply universal values.

Table 3: Lifecycle trigger hierarchy for industrial AI

Trigger class	Illustrative conditions	Required response
Hard stop and immediate investigation	Critical sensor identity/unit change; missing or invalid required field; broken feature contract; unsupported product, recipe, or mode; calibration failure; unsafe confidence or monitor threshold.	Quarantine input or withdraw model; use approved fallback; diagnose cause; require complete revalidation before return.
Targeted revalidation	Sensor replacement or firmware/filter change; planned maintenance or rebuild; schema or feature change; new tooling/material; sustained distribution shift; model/runtime/vendor update; new operating envelope.	Identify affected dependencies; qualify data; regression-test representative conditions; review safety and cybersecurity; stage release and verify rollback.
Routine surveillance	Performance, drift, data-quality, uncertainty, latency, availability, override, and anomaly monitors remain within limits.	Continue operation; retain evidence; update only when economic and assurance thresholds are crossed.
Debt condition	Required owner, data contract, monitor, test, release bundle, rollback, fallback, or supplier exit path is absent or overdue.	Enter the liability in the debt register; estimate remediation effort and exposure; prioritize by consequence and repeat-event history.

4.3 Mathematical Formulation

Five normalized dimensions determine how the baseline terms are transformed: process/data volatility (v), dependency coupling (d), supplier/connectivity dependence (s), AI control authority (u), and operational maturity (m). Each is bounded from zero to one. F0 is the baseline annual fixed assurance cost; lambda0 is the baseline annual rate of material lifecycle events; C0 is the baseline all-in cost per event; q0 is the baseline annual residual consequence probability; and L0 is the baseline conditional consequence. Equation (2) defines the illustrative mappings.

$$F = F_0(1 + 0.50d + 0.40s + 0.80u)(1.25 - 0.50m) \quad (2)$$

$$\lambda = \lambda_0(0.50 + 1.00v + 0.40d + 0.30s)(1.20 - 0.40m)$$

$$C_e = C_0(1 + 0.60d + 0.50s + 1.20u)(1.25 - 0.50m)$$

$$q = \min\{0.50,\ q_0(0.50 + 0.80v + 0.50d + 0.40s + 0.80u)(1.20 - 0.50m)\}$$

$$L = L_0(1 + 2.00u)$$

The mappings preserve literature-consistent directionality. Greater coupling, supplier dependence, and authority increase fixed work and event-response costs. Volatility increases the event rate and residual consequence probability. Authority increases conditional consequence because the AI system can influence more consequential decisions. Maturity reduces all four burdens through ownership, monitoring, standard change control, validated fallback, and practiced recovery. The numerical coefficients are sensitivity assumptions and should be replaced by organizational data where available.

AILAC converts the fixed annual assurance cost and expected annual event-response cost into five-year present value (Equation 3).

$$AILAC_{a,u} = A_H(r)\,[F + \lambda C_e] \quad (3)$$

RLE and TLB are defined in Equation (4). RLE is separated from AILAC so that a site can distinguish the cost of residual work from the expected risk that remains after those controls are in place.

$$RLE_{a,u} = A_H(r)\,qL, \qquad TLB_{a,u} = AILAC_{a,u} + RLE_{a,u} \quad (4)$$

The lifecycle debt premium compares the proposed architecture-authority combination with a feasible reference (Equation 5). A positive value indicates an additional lifecycle burden, not necessarily an irrational choice; the additional operational benefit must be weighed against the premium.

$$AILDP_{a,u} = TLB_{a,u} - TLB_{0,u_0} \quad (5)$$

The minimum five-year expected benefit required to reach break-even is the sum of implementation, routine operating cost, and total lifecycle burden (Equation 6).

$$B^*_{a,u} = C_i + C_o + TLB_{a,u} \quad (6)$$

4.4 Architecture and Authority Interpretation

Architecture and authority are modeled independently because a local system can be autonomous and a cloud system can be advisory. Table 4 summarizes the assurance emphasis for the six deterministic cases. The categories are a practical analytical rubric, not a universal standards taxonomy.

Table 4. Architecture-authority scenarios and assurance emphasis

Scenario	Architecture and authority	Primary lifecycle assurance obligation
Disconnected local advisory	Local inference; controlled periodic updates; no persistent external path.	Validate recommendations, local data and runtime; preserve manual process; manage distributed hardware and controlled release.
Read-only on-premises advisory	Historian replica or approved read-only source; centralized local platform.	Assure data replication, feature parity, access, monitor coverage, and rollback without a control path.
Hybrid edge-cloud advisory	Local inference; centralized training, registry, and fleet visibility.	Assure edge-cloud synchronization, version pinning, outage behavior, observability, and reproducible local release.
Vendor-hosted human-approved	External service proposes action; authorized person approves each action.	Assure provider, identity, API behavior, latency, approval interface, command blocking, logs, continuity, and exit.
Local bounded supervisory	AI acts within a defined local operating envelope with independent monitor.	Assure model, monitor, authority limits, transfer timing, backup control, handback, and operator situational awareness.
High-volatility autonomous	AI controls or initiates action without per-action approval in a changing environment.	Full lifecycle safety case, independent runtime assurance, automatic fallback, frequent revalidation, and strongest recovery evidence.

5. RESULTS

5.1 Deterministic Scenario Results

Table 5 reports the scenario results. The disconnected local advisory design establishes a reference TLB of 0.409, normalized to five-year present-value units. Read-only on-premises advisory operation increases the lifecycle premium to 0.315. Hybrid edge-cloud advisory operation increases the premium to 1.331 because the scenario adds synchronization, supplier, and distributed release obligations while retaining low control authority. The vendor-hosted human-approved case reaches a premium of 2.632. Human approval limits direct actuation but does not remove provider, API, identity, latency, workflow, revalidation, and fallback obligations.

The local bounded supervisory scenario has low supplier dependence but higher authority, producing a TLB of 3.642 and a break-even benefit of 5.442. The high-volatility autonomous scenario produces TLB of 12.113 and a break-even benefit of 14.563. This does not mean autonomous AI is always uneconomic. It means that, under the stated assumptions, greater process volatility, coupling, authority, and lower maturity create a large permanent assurance and residual-exposure burden that a correspondingly large operational benefit must match.

Table 5: Deterministic scenario outcomes

Scenario	AILAC	RLE	TLB	AILDP	Break-even benefit B*
Disconnected local advisory	0.378	0.031	0.409	0.000	1.139
Read-only on-premises advisory	0.661	0.064	0.724	0.315	1.724
Hybrid edge-cloud advisory	1.595	0.145	1.740	1.331	3.190
Vendor-hosted human-approved	2.692	0.349	3.041	2.632	4.591
Local bounded supervisory	2.987	0.654	3.642	3.232	5.442
High-volatility autonomous	9.721	2.391	12.113	11.703	14.563

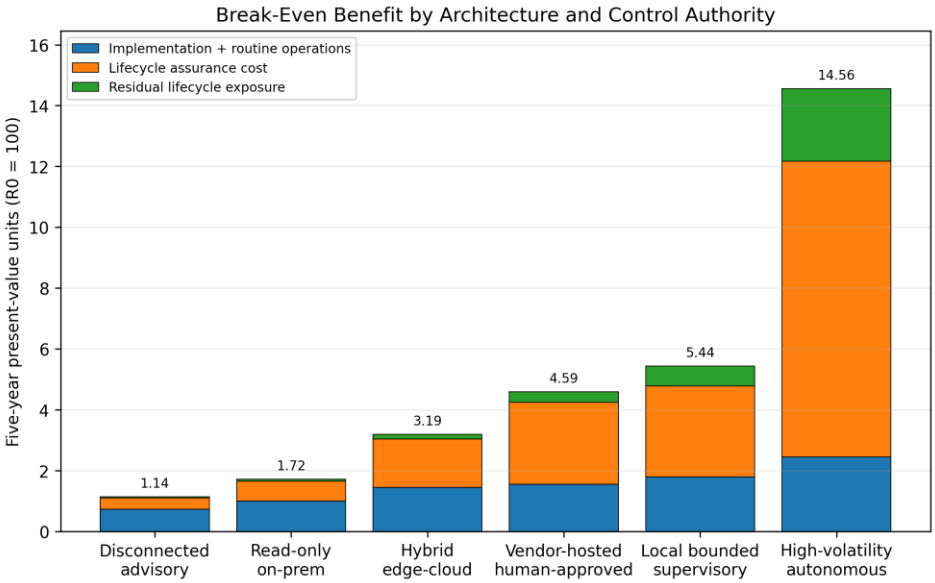

Figure 2: Five-year normalized cost and break-even benefit across deterministic scenarios. Each total combines implementation and routine operating cost, Industrial AI Lifecycle Assurance Cost, and residual lifecycle exposure. Values are illustrative model outputs, not empirical industry benchmarks.

5.2 Monte Carlo Results

Across 20,000 simulations using the randomly generated fixed seed, median AILAC was 1.933, with a 5th-95th percentile interval of 1.103-3.377. Median RLE was 0.346, with a 5th-95th percentile interval of 0.112-0.919. Median TLB was 2.331, and median AILDP relative to the disconnected reference was 1.921. The median break-even benefit was 3.959 normalized five-year present-value units; the 5th and 95th percentiles were 2.839 and 5.631. Figure 3 displays the break-even distribution. These values describe the behavior of the selected bounded assumptions. They are not estimates of the average cost, failure rate, or return of industrial AI.

Table 6: Monte Carlo output summary (N = 20,000; seed = 0x5D2CA2CC0D9F3664)

Output	Mean	Median	SD	5th percentile	95th percentile
Industrial AI Lifecycle Assurance Cost	2.051	1.933	0.714	1.103	3.377
Residual Lifecycle Exposure	0.408	0.346	0.260	0.112	0.919
Total Lifecycle Burden	2.460	2.331	0.813	1.373	3.963
Lifecycle Debt Premium	2.050	1.921	0.813	0.963	3.554
Break-even benefit	4.061	3.959	0.868	2.839	5.631

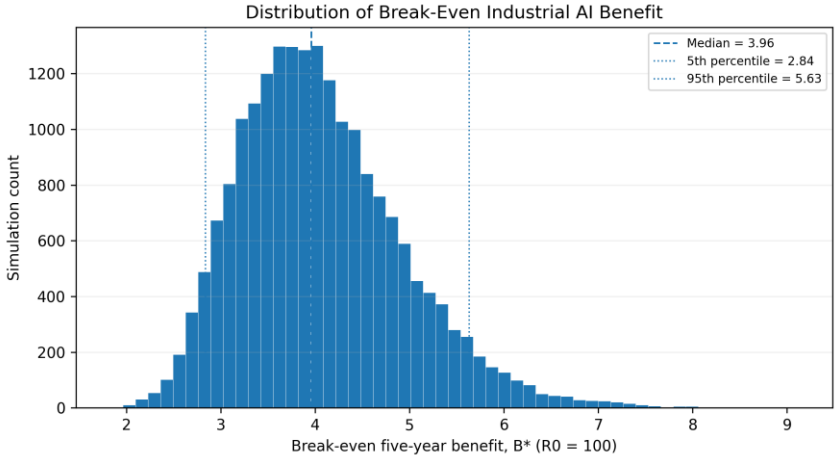

Figure 3. Distribution of the five-year break-even expected benefit B* across 20,000 simulations. The vertical markers identify the median and 5th-95th percentile interval. The distribution represents model behavior under the assumptions in Table 2.

5.3 Sensitivity and Break-Even Interpretation

Baseline all-in cost per lifecycle event had the strongest positive correlation with TLB (r = 0.512, 95% CI [0.501, 0.522]), followed by baseline event frequency (r = 0.441, 95% CI [0.430, 0.453]) and AI control authority (r = 0.401, 95% CI [0.389, 0.412]). Operational maturity had the strongest negative relationship (r = -0.284, 95% CI [-0.296, -0.271]). Dependency coupling, process/data volatility, and supplier/connectivity dependence had smaller positive correlations of 0.216, 0.193, and 0.189. Spearman's results preserved the same substantive ordering.

The sensitivity ranking has two implications. First, the economic question is not simply how frequently a model is retrained. It is the complete cost of responding safely to a material event. Second, authority must be priced independently of hosting. A local supervisory or autonomous system can impose a greater assurance burden than a cloud advisory model because the consequence and validation requirements differ. Maturity reduces burden by lowering recurring work, event frequency, event cost, and residual consequence within the model; it is therefore an investment variable, not merely a governance descriptor.

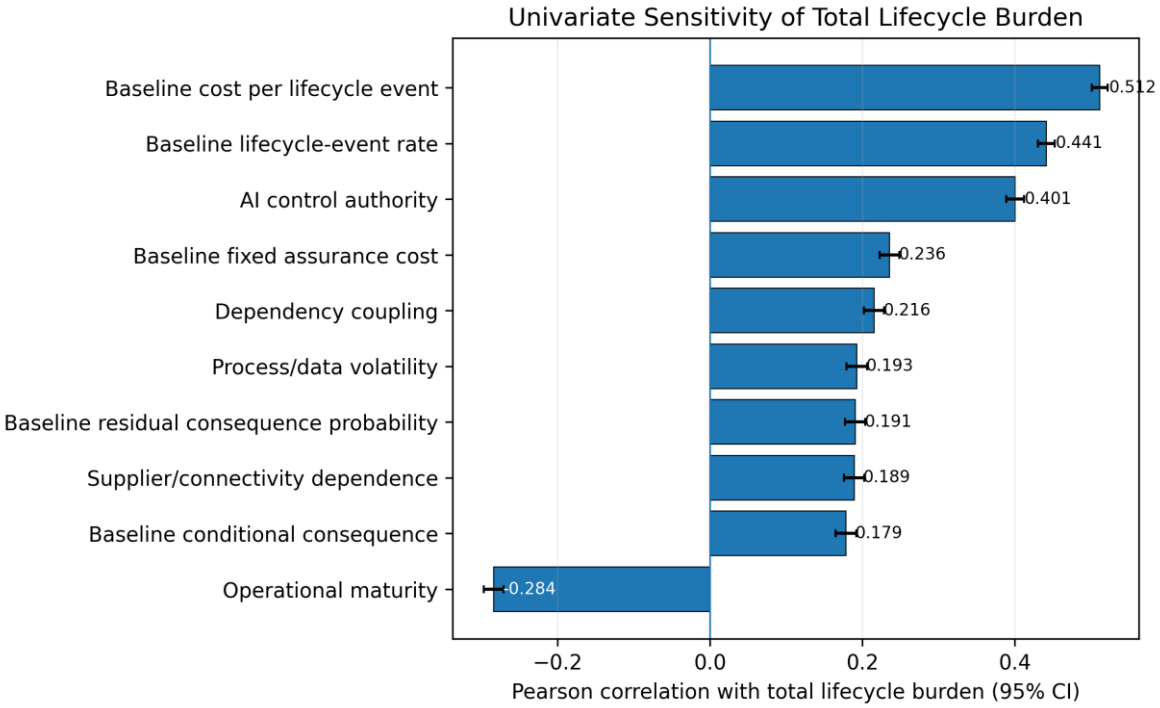

Figure 4. Pearson correlations between primitive inputs and Total Lifecycle Burden, with two-sided 95% confidence intervals. Inputs were sampled independently; correlations are sensitivity indicators rather than causal effects or variance shares.

The update decision should also include the whole industrial release cost. Equation (7) authorizes an update only when the expected reduction in stale-model loss exceeds diagnosis, data preparation, training, validation, release, and rollback-reserve cost. A drift alert may therefore lead to repair, recalibration, a data-pipeline correction, continued monitoring, or model withdrawal rather than retraining.

$$E[\Delta L_{\text{stale}}] > C_{\text{diagnosis}} + C_{\text{data}} + C_{\text{training}} + C_{\text{validation}} + C_{\text{release}} + C_{\text{rollback}} \quad (7)$$

6. DISCUSSION

6.1 Industrial AI Is a Maintained Production Capability

The central finding is structural rather than numerical: industrial AI creates a permanent assurance obligation. A model remains useful only while its measurement system, process context, data pipeline, runtime, users, suppliers, and authority controls remain within a validated state. The literature documents these dependencies separately; the model places them in a common cost structure. The lack of transferable cost benchmarks does not justify omitting the burden. It requires organizations to estimate the burden based on site-specific event frequency, role-based touch time, test requirements, recovery objectives, and asset consequence.

This distinction changes the investment case. A pilot may demonstrate accuracy without revealing the cost of five years of monitoring, drift triage, sensor changes, pipeline upgrades, cyber reviews, supplier changes, release management, and operator competency. A business case that includes only implementation and routine hosting can therefore yield a false positive, even when the model performs well during commissioning. The break-even benefit in Equation (6) forces these obligations into the approval decision.

6.2 Architecture and Authority Must Be Chosen Separately

Architecture changes the location and ownership of lifecycle work. Vendor hosting can centralize model operations, but the organization still owns the operational outcome. It must manage provider versions, identity, APIs, data movement, connectivity loss, incident coordination, evidence, portability, and fallback. Edge and disconnected systems reduce some external dependencies while increasing distributed hardware, local release, and observability work. Hybrid designs may be operationally attractive, but their additional synchronization and compatibility layers must be budgeted for.

Authority changes the assurance case in a more fundamental way. Advisory AI can usually be disabled without altering the existing control function. Human approval adds authorization, interface, latency, and workload requirements. Supervisory and autonomous systems require assurance of the entire monitor-decision-controller-fallback chain, including intervention timing and safe handback. This is why authority was among the three strongest sensitivity variables, even though supplier dependence was modeled separately.

6.3 Maturity Reduces the Cost of Change

Operational maturity reduced TLB in the simulation because it represents ownership, sensor and data contracts, versioned artifacts, monitoring, test environments, release gates, rollback, fallback, and practiced response. Mature organizations may experience the same external process change as immature organizations but detect it earlier, diagnose it faster, know which dependencies are affected, and introduce a validated release with less production disruption. Maturity can therefore reduce the event frequency experienced by the system, the cost per event, and the residual consequence.

A practical implementation should maintain a lifecycle debt register for every production model. At minimum, the register should identify the model owner; physical and digital dependencies; validated operating envelope; data contract; monitors and thresholds; change triggers; test and release evidence; fallback and rollback; supplier and connectivity dependencies; mean time and labor by event type; repeat incidents; and retirement conditions. Missing or overdue controls should be treated as debt rather than a normal assurance cost. This creates an empirical path for future calibration: hours per model-year, hours per event, revalidation lead time, production interruption, and residual loss can be collected from tickets and change records.

6.4 Relationship to IT/OT Convergence Economics

The lifecycle model complements rather than duplicates the Convergence Exposure Premium. CEP asks how much incremental cyber loss a proposed connectivity architecture creates relative to a less-exposed counterfactual [29]. The present paper asks how much it costs to keep an industrial AI capability valid, secure, recoverable, and useful after deployment, including changes that occur even in a disconnected system. Model drift is therefore not the same as connectivity debt. It is a lifecycle event whose cost can be amplified by connectivity, supplier dependence, architecture, and control authority. A complete investment case may include implementation, routine operations, AILAC, residual lifecycle exposure, connectivity debt, and incremental cyber loss, provided the accounting boundaries prevent the same labor or loss from being counted twice.

6.5 Managerial and Engineering Implications

For executives and finance teams, the model converts post-deployment assurance from an unpriced technical concern into a break-even benefit requirement. For plant and engineering leaders, it makes sensor, process, recipe, equipment, and operating-envelope changes formal AI impact events. For data and ML teams, it expands retraining cost from compute to the complete validated release. For cybersecurity teams, it makes model, pipeline, access, vendor, and remote-support changes part of the threat and evidence lifecycle. For operators and safety personnel, it links authority to independent monitoring, fallbacks, overrides, and the preservation of non-AI capabilities.

The most defensible approval rule is not to select the most connected or autonomous design. It is to select the least-dependent architecture and lowest authority that can produce the documented operational outcome, then increase dependence or authority only when the incremental benefit exceeds the resulting lifecycle premium. This rule does not reject industrial AI. It prevents a successful pilot from becoming an unfunded permanent obligation.

7. LIMITATIONS AND FUTURE RESEARCH

This study has several limitations. First, the literature synthesis was integrative and targeted rather than an exhaustive systematic review. Elicit was used for discovery, and the final sources were verified at the claim and metadata levels, but the search may have omitted relevant sector-specific studies. Second, the source base documents performance

degradation and lifecycle activities more strongly than realized industrial dollar loss, safety incidents, annual labor ratios, or universal retraining intervals. The absence of those benchmarks is part of the identified gap, but it limits external calibration.

Third, the five-year horizon, 8% discount rate, triangular distributions, scenario inputs, and coefficient mappings are transparent analytical assumptions. They are not parameters estimated from a multi-site longitudinal dataset. Fourth, Monte Carlo primitive inputs were sampled independently. Real organizations may exhibit correlation among volatility, coupling, supplier dependence, authority, cost, maturity, and consequence. Fifth, the scalar residual-exposure formulation does not represent threshold effects, correlated process failures, safety cascades, or time-dependent deterioration. Event trees or state-transition models would be appropriate where those effects are material.

Sixth, AILDP compares total lifecycle burden with a reference architecture; it does not prove that every increment is avoidable technical debt. The debt register is required to separate necessary assurance from deferred or poorly designed liability. Seventh, the authority taxonomy is a practical governance rubric rather than a universal standard.

Future research should collect work tickets, time records, model-monitor events, asset changes, release records, operator interventions, and outcome data across industrial sites. Labor should be reported by role, as hours per model year and hours per event. Studies should compare disconnected, edge, on-premises, hybrid, and hosted designs serving the same use case; separate advisory, human-approved, supervisory, and autonomous authority; and quantify the cost and performance of fallback. Bayesian or survival models could estimate event frequency, while correlated simulation and real-options analysis could value staged increases in authority or connectivity. The strongest validation would follow industrial AI systems from commissioning through sensor, process, model, vendor, and workforce changes.

8. CONCLUSION

Industrial AI is not a deploy-once asset. Its value depends on a permanent capability to monitor the measurement system and model, diagnose change, qualify data, retrain or repair when justified, revalidate the complete release bundle, reassess cybersecurity and suppliers, preserve fallback, and maintain operator competence. Existing literature documents these obligations but does not provide a universal industrial lifecycle-cost curve.

The proposed AILAC, TLB, and AILDP framework makes those obligations explicit. Under the illustrative scenarios and simulations, all-in event-response cost, lifecycle-event frequency, and AI control authority were the strongest positive sensitivity drivers, while operational maturity reduced burden. The model does not estimate an industry rate; it provides a reproducible structure for replacing hidden post-deployment assumptions with site-specific inputs.

The practical conclusion is that every industrial AI proposal should identify its validated operating envelope, lifecycle triggers, full update cost, authority level, fallback, supplier dependencies, and retirement condition before approval. The model that worked at commissioning has not completed its business case. It has begun a lifecycle.

DATA AVAILABILITY

The manuscript is based on publicly available literature and a normalized model-based analysis. The accompanying data package contains the complete Python script, software versions, a randomly generated fixed 64-bit seed, an assumption register, six deterministic scenario inputs and outputs, the 20,000-row Monte Carlo dataset, sensitivity statistics, a simulation summary, figures, and a source-verification register.

FUNDING

This research received no external funding.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

AUTHOR CONTRIBUTIONS

Daniel Ward: conceptualization, methodology, investigation, formal analysis, software, visualization, writing - original draft, and writing - review and editing.

ACKNOWLEDGMENTS

The author acknowledges the use of Elicit as a tool for literature discovery and evidence organization. All manuscript claims and references were based on the original publications or official source records rather than on AI-generated summaries.

REFERENCES

1. 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. In *Advances in Neural Information Processing Systems 28*. https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html
2. Shankar, S., Garcia, R., Hellerstein, J. M., & Parameswaran, A. G. (2024). We have no idea how models will behave in production until production: How engineers operationalize machine learning. *Proceedings of the ACM on Human-Computer Interaction, 8*(CSCW1), Article 206, 1–34. https://doi.org/10.1145/3653697
3. Raffin, T., Reichenstein, T., Werner, J., Kuehl, A., & Franke, J. (2022). A reference architecture for the operationalization of machine learning models in manufacturing. *Procedia CIRP, 115*, 130–135. https://doi.org/10.1016/j.procir.2022.10.062
4. Dreyfus, P.-A., Pelissier, A., Psarommatis, F., & Kiritsis, D. (2022). Data-based model maintenance in the era of Industry 4.0: A methodology. *Journal of Manufacturing Systems, 63*, 304–316. https://doi.org/10.1016/j.jmsy.2022.03.015
5. Jourdan, N., Sen, S., Husom, E. J., Garcia-Ceja, E., Biegel, T., & Metternich, J. (2021). On the reliability of machine learning applications in manufacturing environments. *arXiv*. https://doi.org/10.48550/arXiv.2112.06986
6. Jourdan, N., Bayer, T., Biegel, T., & Metternich, J. (2023). Handling concept drift in deep learning applications for process monitoring. *Procedia CIRP, 120*, 33–38. https://doi.org/10.1016/j.procir.2023.08.007
7. Soller, S., Hoelzl, G., & Kranz, M. (2020). Predicting machine errors based on adaptive sensor data drifts in a real world industrial setup. In *2020 IEEE International Conference on Pervasive Computing and Communications* (pp. 1–9). https://doi.org/10.1109/PerCom45495.2020.9127357
8. Chen, M., Park, Y., Mangalathu, S., & Jeon, J.-S. (2024). Effect of data drift on the performance of machine-learning models: Seismic damage prediction for aging bridges. *Earthquake Engineering & Structural Dynamics, 53*(15), 4541–4561. https://doi.org/10.1002/eqe.4230
9. Martins, A., Fonseca, I., Farinha, J. T., Reis, J., & Cardoso, A. J. M. (2023). Online monitoring of sensor calibration status to support condition-based maintenance. *Sensors, 23*(5), Article 2402. https://doi.org/10.3390/s23052402
10. Husom, E. J., Tverdal, S., Goknil, A., & Sen, S. (2022). UDAVA: An unsupervised learning pipeline for sensor data validation in manufacturing. In *Proceedings of the 1st International Conference on AI Engineering—Software Engineering for AI* (pp. 159–169). https://doi.org/10.1145/3522664.3528603
11. van der Weide, T., Papadopoulos, D., Smirnov, O., Zielinski, M., & van Kasteren, T. (2017). Versioning for end-to-end machine learning pipelines. In *Proceedings of the 1st Workshop on Data Management for End-to-End Machine Learning* (Article 2, pp. 1–9). https://doi.org/10.1145/3076246.3076248
12. Caveness, E., Chandramouli, P. S. G., Peng, Z., Polyzotis, N., Roy, S., & Zinkevich, M. (2020). TensorFlow Data Validation: Data analysis and validation in continuous ML pipelines. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp. 2793–2796). https://doi.org/10.1145/3318464.3384707
13. 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* (pp. 1123–1132). https://doi.org/10.1109/BigData.2017.8258038
14. Wang, X., Schuster, H., Borrison, R., & Kloepper, B. (2023). Technical debt management in industrial ML: State of practice and management model proposal. In *2023 IEEE 21st International Conference on Industrial Informatics* (pp. 1–9). https://doi.org/10.1109/INDIN51400.2023.10217843
15. Cody, T., Adams, S. C., Beling, P. A., & Freeman, L. J. (2022). On valuing the impact of machine learning faults to cyber-physical production systems. In *2022 IEEE International Conference on Omni-Layer Intelligent Systems* (pp. 1–6). https://doi.org/10.1109/COINS54846.2022.9854969
16. Hummer, W., Muthusamy, V., Rausch, T., Dube, P., El Maghraoui, K., Murthi, A., & Oum, P. (2019). ModelOps: Cloud-based lifecycle management for reliable and trusted AI. In *2019 IEEE International Conference on Cloud Engineering* (pp. 113–120). https://doi.org/10.1109/IC2E.2019.00025
17. Keusch, A., Blumauer-Hiessl, T., Furutanpey, A., Schall, D., & Dustdar, S. (2024). Platform-agnostic MLOps on edge, fog and cloud platforms in industrial IoT. In *Proceedings of the 20th International Conference on Web Information Systems and Technologies* (pp. 71–79). https://doi.org/10.5220/0012977500003825
18. McCormack, C., & Mitiche, I. (2025). A platform for machine learning operations for network constrained far-edge devices. *Applied System Innovation, 8*(5), Article 141. https://doi.org/10.3390/asi8050141
19. Cybersecurity and Infrastructure Security Agency, & partner agencies. (2026). *Secure connectivity principles for operational technology (OT)*. https://www.cisa.gov/resources-tools/resources/secure-connectivity-principles-operational-technology-ot
20. Tabassi, E. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)* (NIST AI 100-1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1
21. Cybersecurity and Infrastructure Security Agency, & partner agencies. (2025). *Principles for the secure integration of artificial intelligence in operational technology*. https://www.cisa.gov/resources-tools/resources/principles-secure-integration-artificial-intelligence-operational-technology

22. Wang, Y., Buysse, L., Vanoost, D., Pissoort, D., Boydens, J., & Verbeke, M. (2025). AMLOP: A framework for the operational safety assurance of machine learning leveraging model monitoring. In *2025 IEEE Conference on Artificial Intelligence* (pp. 1361–1368). https://doi.org/10.1109/CAI64502.2025.00248
23. Wheatman, B., Chen, J., Sookoor, T., & Amir, Y. (2021). RADICS: Runtime assurance of distributed intelligent control systems. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops* (pp. 182–187). https://doi.org/10.1109/DSN-W52860.2021.00038
24. Florian, E., Sgarbossa, F., & Zennaro, I. (2021). Machine learning-based predictive maintenance: A cost-oriented model for implementation. *International Journal of Production Economics, 236*, Article 108114. https://doi.org/10.1016/j.ijpe.2021.108114
25. Mahadevan, A., & Mathioudakis, M. (2023). Cost-effective retraining of machine learning models. *arXiv*. https://doi.org/10.48550/arXiv.2310.04216
26. Kanda, H., Okamura, H., & Dohi, T. (2023). A note on optimal retraining strategy for ML systems. In *2023 10th International Conference on Dependable Systems and Their Applications* (pp. 730–733). https://doi.org/10.1109/DSA59317.2023.00104
27. Guissouma, H., Zink, M., & Sax, E. (2023). Continuous safety assessment of updated supervised learning models in shadow mode. In *2023 IEEE 20th International Conference on Software Architecture Companion* (pp. 301–308). https://doi.org/10.1109/ICSA-C57050.2023.00069
28. Cruickshank, I., & Kohtz, S. (2024). *Planning for AI sustainment: A methodology for maintenance and cost management* (SYM-AM-24-079). Acquisition Research Program. https://dair.nps.edu/handle/123456789/4980
29. Ward, D. (2026). The convergence exposure premium: A risk-adjusted model for industrial digital transformation and IT/OT connectivity. *American Journal of Industrial and Business Management, 16*(8), in press.

CITATION
Ward, D. (2026). Industrial AI Lifecycle Debt: An Event-Based Model for Drift, Revalidation, and Sustainment in IT/OT Environments. Global Journal of Research in Engineering & Computer Sciences, 6(4), 115–128.
https://doi.org/10.5281/zenodo.21949424