
A METHODOLOGY FOR AUDITABLE TRUSTWORTHINESS LEVELS IN AI LIFECYCLE GOVERNANCE

Andrea Ferrario^{*1,2,3}

¹ Institute of Biomedical Ethics and History of Medicine, University of Zürich, Zürich, Switzerland

² SUPSI, Dalle Molle Institute for Artificial Intelligence (IDSIA), Lugano, Switzerland

³ ETH Zürich, Zürich, Switzerland

July 20, 2026

ABSTRACT

AI governance increasingly requires judgments about whether an AI system remains adequately trustworthy over time, whether observed changes are tolerable, and how such judgments should be documented in a transparent and contestable way. Yet existing work on AI trustworthiness remains either too high-level to support lifecycle monitoring and reassessment or too narrowly metric-driven to connect with governance needs. We therefore propose a lightweight methodology for auditable trustworthiness levels in AI governance. The methodology has two components: a formal framework for representing and learning trustworthiness levels, and a lightweight AI lifecycle governance procedure for documenting, monitoring, and reassessing them over time. The formal framework models governance-relative trustworthiness through a context-sensitive protocol of measurable dimensions and learns trustworthiness levels as interpretable rules over trustworthiness profiles. Using decision trees as an interpretable proof-of-concept model class, the methodology yields explicit trustworthiness plateaus, readable level transitions, and two simple lifecycle diagnostics: boundary margins and profile drift. The governance procedure embeds these formal objects in a conformity-oriented workflow for design-time labeling, post-deployment monitoring, reassessment, and reporting. It also assigns human responsibilities and control gates for protocol design, validation, monitoring, and reassessment. We illustrate the methodology on synthetic AI lifecycle traces involving degradation, shocks, updates, heterogeneous monitoring cadences, and system comparison. Our methodology does not replace legal or other expert judgment: it supports conformity documentation and lifecycle monitoring by providing an evidential basis for documenting and tracking AI governance-relevant changes over time.

Keywords: artificial intelligence, governance, EU AI Act, lifecycle, audit, trustworthiness

^{*}aferrario@ethz.ch

1 Introduction

AI governance depends on judgments that are neither purely technical nor purely legal. Providers, deployers, and oversight bodies must assess whether an AI system remains trustworthy after reconfiguration, updating, or shifts in its context of use. Here, we use AI trustworthiness in a governance-oriented sense: an AI system is trustworthy to the extent that, for a given intended purpose and context of use, it satisfies the socio-technical requirements documented in its design and governance. They must also determine when differences in trustworthiness across copies or variants of an AI system are material enough to warrant documentation, review, or oversight. Such trustworthiness-based questions arise across a wide range of systems, from LLM-based assistants and clinical decision-support systems to recommender agents and AI systems automating processes in banking. They are especially salient in settings where conformity assessment, monitoring, and reassessment become key AI lifecycle governance procedures, as in the case of high-risk AI systems under the EU AI Act (AIA) [EU AI Act, 2024].

A central difficulty is that current work on AI trustworthiness is pulled in two directions. On the one hand, governance frameworks articulate trustworthiness through broad requirements such as robustness, transparency, fairness, safety, and accountability [Kemmerzell et al., 2025, Mehrotra et al., 2026]. On the other hand, technical work in system engineering and MLOps operationalizes narrower subsets of these requirements through local metrics or benchmarks, often without an explicit account of how such measurements should be integrated into AI governance over time [Rabanser et al., 2026, Kreuzberger et al., 2023, Bayram et al., 2024]. As a result, trustworthiness is frequently invoked in AI governance but rarely formalized in a way that supports simple and auditable AI lifecycle monitoring. Furthermore, many governance-relevant trustworthiness judgments focus on design or review activities: they may be documented through expert procedures, checklists, or reference cases, yet they do not by themselves yield a mechanism for post-deployment and quantitative trustworthiness assignment under changing AI lifecycle conditions.

This paper addresses this gap. We propose **a lightweight methodology for auditable trustworthiness levels in AI governance**. The methodology has two components: (i) a formal framework for representing and learning trustworthiness levels, and (ii) a lightweight AI lifecycle governance procedure for documenting, monitoring, and reassessing them over time. The formal framework represents trustworthiness through documented protocols and profiles, and learns trustworthiness-level rules over those profiles. The role of learning is to translate the expert-defined scale into a reusable, auditable, and computationally lightweight rule for deployment-time monitoring. Depending on the variation generated by deployment, the informativeness of the monitored profile, and the stability of the labeling process, the learned rule may instead *confirm* the expert standard, *compress* it into a simpler operational rule, or *refine* it. The AI lifecycle governance procedure embeds these objects in a multi-phase workflow that connects pre-deployment protocol design and labeling with empirical learning, post-deployment monitoring, validation, and reassessment. In this way, the methodology provides a quantitative and auditable intermediate layer between trustworthiness-relevant measurements and lifecycle governance decisions, thereby supporting conformity assessment, technical documentation, post-deployment monitoring, and the review of governance-relevant changes that may require assessment under the logic of *substantial modifications*—see Art. 3(23) and Art. 43(4) in the AIA and [Ferrario, 2026]. More specifically, it functions as a governance translation layer: documented trustworthiness measurements are organized into trustworthiness profiles, profiles are mapped to auditable trustworthiness levels, and level transitions become available for monitoring, documentation, and reassessment across the AI lifecycle. The paper makes the following contributions:

- **Formal framework.** We introduce a formal framework for auditable trustworthiness levels of AI systems that distinguishes trustworthiness protocols, measured trustworthiness profiles, expert-defined reference labels and rules, and learned empirical trustworthiness-level rules.
- **Lightweight AI lifecycle governance procedure.** We embed the formal framework in a lightweight governance procedure that connects pre-deployment protocol design, trustworthiness learning, post-deployment monitoring and reassessment in an AI lifecycle workflow.
- **Operational monitoring and documentation tools.** We propose two simple lifecycle diagnostics—boundary margins and profile drift—together with a reporting-oriented governance artifact for documenting assumptions, learned trustworthiness rules, monitoring outputs, and governance updates over time.
- **Simulation proof of concept.** We illustrate the proposal on synthetic AI lifecycle traces that model degradation, shocks, updates, heterogeneous monitoring cadences, and system comparison, showing how auditable trustworthiness levels can support AI lifecycle monitoring. While these experiments do not provide external validation, they show how the governance objects behave under different lifecycle conditions.

2 Background

2.1 AI Trustworthiness: Common Ground

Trustworthy AI is the paradigm that seeks to identify the criteria under which AI systems should be designed, developed, deployed, and governed so as to reflect core principles that ought to guide their role in society. For example, the AI HLEG framework is grounded in principles such as human agency and oversight, technical robustness and safety, privacy and data governance, transparency, diversity and fairness, societal and environmental well-being, and accountability [EU AI HLEG, 2019]. Over time, regulators, standardization bodies, governments, and international organizations have proposed partially overlapping approaches to delineating the scope of AI trustworthiness [EU AI HLEG, 2019, EU AI Act, 2024, The White House, 2022, United Nations Educational, Scientific and Cultural Organization, 2022, OECD, 2022, ISO/IEC, 2020].

The literature on Trustworthy AI is now extensive: we refer to recent reviews for a deep dive into definitions and perspectives [Kaur et al., 2022, Kemmerzell et al., 2025, Mehrotra et al., 2026]. Here, we fix the main ideas relevant to our framework. A common underlying assumption across these approaches is that what makes an AI system worthy of trust is not a single property, but the satisfaction of *multiple socio-technical requirements* [Mehrotra et al., 2026, Kemmerzell et al., 2025, Kaur et al., 2022]. The exact formulation of these requirements varies across frameworks, depends on how abstract principles are translated into operational criteria, and remains context-sensitive across domains. For instance, the trustworthiness profile of an LLM-based self-study assistant will differ from that of a clinical decision-support system, a recommender agent, or a multi-agent AI system coordinating compliance procedures in banking. That said, commonly discussed dimensions include fairness, transparency, privacy and data governance, technical robustness and safety, accountability, human agency and oversight, and societal or environmental well-being [Kemmerzell et al., 2025, Kaur et al., 2022]. In what follows, we refer to such requirements as *trustworthiness dimensions*. AI trustworthiness depends on the choice of individual dimensions and also on how they interact, trade off, and reinforce one another over time.

A central implication is that trustworthiness must be embedded in an AI lifecycle governance framework capable of translating high-level principles into auditable practices [Mökander et al., 2022, Laux et al., 2024, Floridi et al., 2022]. The lifecycle perspective is important because AI systems are subject to change: conformity assessment establishes an initial baseline, post-deployment monitoring tracks continued compliance, and ‘substantial modification’ can function as a governance-relevant discontinuity condition. This orientation is particularly visible in the EU AI Act—see Annex IV and Article 72 therein [EU AI Act, 2024]. Thus, the challenge is to provide a sufficiently explicit representation of trustworthiness that can connect these obligations across time. The next section assesses the extent to which this challenge is addressed by existing methods and metrics for evaluating AI trustworthiness.

2.2 Existing Trustworthiness Assessment Tools, Procedures, and Audits

Recent reviews suggest that the landscape of trustworthy-AI assessment is fragmented both instrumentally and conceptually [Kaur et al., 2022, Kemmerzell et al., 2025, Mehrotra et al., 2026]. In particular, the recent scoping review of AIES and FAccT proceedings by Mehrotra et al. [2026] shows that trustworthiness is not treated as a single settled construct, but rather as a contested terrain where technical, psychological, ethical, and societal dimensions intersect.

Four families of approaches are especially relevant. First, there are governance-oriented checklists and assessment frameworks. In Europe, the Assessment List for Trustworthy AI (ALTAI) translates the AI HLEG requirements into a structured self-assessment instrument [EU HLEG AI, 2020, Radclyffe et al., 2023]. In the United States, the NIST AI Risk Management Framework and its Playbook organize AI governance around the functions Govern, Map, Measure, and Manage [National Institute of Standards and Technology, 2023]. These frameworks are valuable, but they remain primarily procedural: they do not themselves define explicit trustworthiness-level rules, nor do they provide actionable diagnostics throughout the AI lifecycle. Second, there are documentation artifacts such as model cards, datasheets for datasets, and ethical codes [Mitchell et al., 2019, Gebru et al., 2021, Loi et al., 2019]. These improve transparency and accountability, yet their primary function is documentary rather than rule-inducing, e.g., by providing a dynamic rule for repeated trustworthiness assignment during post-deployment monitoring. Third, there are toolkits targeting specific trustworthiness dimensions, such as AI Fairness 360, Fairlearn, and AI Explainability 360 [Bellamy et al., 2019, Bird et al., 2020, Arya et al., 2019]. These are useful, but they remain dimension-specific and do not yield a unified, governance-level representation of system state and its monitoring over time. Fourth, there are participatory and audit-style methodologies such as Z-Inspection, which emphasize interdisciplinary review and contextual evaluation [Zicari et al., 2021]. These approaches are valuable for surfacing ethical tensions, documenting deliberation, and structuring expert judgment across lifecycle stages. However, they do not yield a compact formal representation of trustworthiness, nor rules for assigning and tracking its levels over time.²

Taken together, four elements are rarely provided jointly. First, existing approaches only partially specify how heterogeneous trustworthiness dimensions should be integrated into one governance-relevant representation of system state. Second, they rarely provide an explicit and auditable rule that maps such a representation into trustworthiness levels such as acceptable, risky, or unacceptable functioning. Third, existing work only partially formalizes how trustworthiness dimensions are measured over time after deployment, especially when different dimensions depend

²Existing work on AI agents typically addresses only one trustworthiness layer—e.g., reliability metrics or runtime governance—without yielding a unified trustworthiness formalism for lifecycle monitoring [Rabanser et al., 2026, Phiri, 2025, Staufer et al., 2026].

on different logics, delayed outcomes, and different monitoring cadences. Fourth, current tools and procedures rarely connect pre-deployment expert labeling, post-deployment monitoring, validation, relabeling, and reassessment into one documented lifecycle workflow that can support pre-deployment conformity assessments and post-deployment monitoring processes—see [EU AI Act, 2024]. This is the goal of the formal framework that we present below.

3 Formal Framework of AI Trustworthiness

In response to the four gaps identified above, this section introduces the formal core of the paper. The aim is to define a lifecycle-ready representation of AI trustworthiness, an auditable rule for assigning trustworthiness levels, and a governance procedure that connects design-time labeling with post-deployment monitoring and reassessment.

3.1 AI Trustworthiness Commitments

Informed by the literature on Trustworthy AI, our framework rests on six commitments. First, trustworthiness is context-sensitive and normative: which dimensions matter depends on the socio-technical context (e.g., entertainment vs. healthcare), intended purpose (e.g., to recommend a song for a personalized playlist vs. predicting the likelihood of skin cancer), and application domain of the system (e.g., smartphone app vs. AI system in the clinic) [National Institute of Standards and Technology, 2023]. Second, if trustworthiness is to inform governance, it must be operationalized through indicators that can be assessed in practice [EU HLEG AI, 2020]. Third, trustworthiness is composite rather than merely dimension-wise: it depends on multiple dimensions, such as robustness, fairness, transparency, safety, or oversight, considered jointly rather than in isolation [Kaur et al., 2022, Ferrario, 2025]. A formal account therefore requires an explicit aggregation or regime-assignment rule. Fourth, objective and perceived trustworthiness must be distinguished: a system may be objectively trustworthy without being perceived as such, and vice versa [Liao and Sundar, 2022, Mehrotra et al., 2024, Ferrario, 2024]; here we focus on *objective* trustworthiness, understood as the extent to which the system satisfies the trustworthiness-relevant requirements that hold in its context of use. Fifth, trustworthiness is lifecycle-sensitive and time-dependent, because AI systems are updated, reconfigured, exposed to drift, and embedded in changing environments [Kreuzberger et al., 2023, Ferrario, 2024]. Sixth, trustworthiness formalization should be governance-oriented: the point of formalization is to support documentation, assessment, monitoring, and reassessment in auditable workflows [Floridi et al., 2022, Mökander and Floridi, 2021].

3.2 Protocols, Profiles, and Levels

Let A be an AI system with intended purpose I_A and context of use C_A . We use *intended purpose* in the governance-oriented sense familiar from the AIA: the use for which the AI system is intended by the provider, including the specific context and conditions of use as documented by the provider [EU AI Act, 2024]. We use *context of use* more broadly to denote the realized socio-technical environment in which the system is deployed, monitored, and governed over time. Let

$$P_A = \{d_1, \dots, d_n\}$$

be A 's *trustworthiness protocol*: a finite set of dimensions selected as relevant to the governance-relative trustworthiness of A in that context. In this paper, this trustworthiness is understood in a governance-relative sense: it denotes the extent to which an AI system satisfies the trustworthiness-relevant requirements fixed by a documented protocol for a

given intended purpose and context of use. (It is therefore objective relative to a governance standard.) Dimensions may include, for instance, robustness, fairness, transparency, or safety, but the protocol is not assumed universal. It is selected by context-sensitive admissibility criteria such as normative and operational relevance and auditability. For each dimension $d_i \in P_A$, let $k(i) \in \mathbb{N}$ denote the number of sub-metrics used to operationalize it, with index set $J_{k(i)} := \{1, \dots, k(i)\}$. Each sub-metric is normalized into $[0, 1]$, giving a dimension vector

$$\mathbf{q}_i^A(t) = (q_{i,1}^A(t), \dots, q_{i,k(i)}^A(t)) \in [0, 1]^{k(i)}.$$

Here, the normalization to $[0, 1]$ is a convention; the framework can equivalently be formulated on any common bounded scale, with thresholds and aggregation rules rescaled accordingly. Coordinates also need not have the same orientation: for some dimensions higher values may indicate better trustworthiness, whereas for others, such as fairness gaps, lower values may be preferable. Not all trustworthiness dimensions are measured in the same way. Some measures, such as explanation complexity or override rate (used to measure human agency and oversight), may be computed directly from model outputs, interface traces, or workflow logs available at time t . Other measures, such as predictive accuracy, calibration, or outcome-based fairness measures, require realized outcomes and may therefore only be measurable over a recent window of resolved cases. Accordingly, $Q_i^A(t)$ should be read as the most recent auditable estimate of dimension d_i available at time t under the measurement protocol adopted for that dimension, rather than as an instantaneous quantity computed from information available exactly at t : Table 1 gives representative examples; we refer to [Kemmerzell et al. \[2025\]](#) for an extensive survey instead. Then, a dimension-level aggregator $a_i : [0, 1]^{k(i)} \rightarrow [0, 1]$ yields the dimension score $Q_i^A(t) := a_i(\mathbf{q}_i^A(t))$. Also, these aggregation rules depend on the dimension under consideration. For instance, when aggregating consistency or robustness measures for LLM-based applications, one may use averaging [[Rabanser et al., 2026](#)]. Aggregating safety-related measures may instead require the product of risk and expected severity measures [[Kaplan and Garrick, 1981](#)]. Collecting these scores gives the observed *AI trustworthiness profile*³

$$\mathbf{Q}_A(t) := (Q_1^A(t), \dots, Q_n^A(t)) \in [0, 1]^n.$$

3.3 Learning Empirical AI Trustworthiness-Level Rules

Suppose we have a historical labeled dataset $\mathcal{D}_N = \{(\mathbf{Q}_A(t_i), Y_i)\}_{i=1}^N$, where $\mathbf{Q}_A(t_i) \in [0, 1]^n$ is the observed trustworthiness profile at time t_i , and $Y_i \in [K]$ is the trustworthiness level assigned by the audit process. (More on how to construct such a dataset at the end of this section.) The goal is to learn an empirical trustworthiness-level rule

$$\hat{T}_{A,N} : [0, 1]^n \rightarrow [K],$$

where any $\mathbf{Q}_A(t) \mapsto \hat{T}_{A,N}(\mathbf{Q}_A(t))$ belongs to a class of *interpretable stepwise functions*. A natural first restriction is to focus on rules whose partition of $[0, 1]^n$ is generated by threshold conditions on individual trustworthiness dimensions, so that the resulting region boundaries are parallel to the coordinate axes. In fact, in governance settings, trustworthiness

³At the theoretical level, we can represent the operative governance standard as a reference trustworthiness-level rule $T_A^* : [0, 1]^n \rightarrow [K]$, where $[K] = \{1, \dots, K\}$ is a fixed trustworthiness-level scale chosen ex ante as a governance parameter. (Here, objectivity is with respect to a governance standard.) Along the trajectory of the system, $T_A^*(\mathbf{Q}_A(t))$ denotes the reference trustworthiness level of the system A at time t [[Ferrario, 2024](#)].

Table 1: Representative trustworthiness dimensions, example measures, and their post-deployment collection logic, adapted from Kemmerzell et al. [2025].

Dimension	Example measures	Typical post-deployment collection / update logic
Accuracy/predictive performance	Squared error; F1 score; AUROC.	Usually updated on recent resolved cases only. It is therefore often delayed and estimated over rolling validation windows.
Calibration / uncertainty	Brier score; expected calibration error.	Usually updated only when outcomes become available, so it is typically assessed on delayed validation windows rather than live logs.
Fairness	Demographic parity; disparate impact; equalized odds; false positive error-rate balance; counterfactual fairness	Often evaluated on batches of recent labeled cases, since many metrics require protected attributes and realized outcomes.
Transparency / explainability	Faithfulness; robustness of explanations; explanation complexity; task performance; documentation review.	Some indicators can be updated directly from model behavior or documentation, while human-centered ones are usually assessed periodically.
Privacy and data governance	k -anonymity; t -closeness; differential privacy (ϵ, δ) ; membership-inference attack success rate.	Commonly assessed through periodic audits, attack simulations, or after data/model updates rather than continuously.
Technical robustness and safety	Performance drop under shift; attack success rate.	Drift indicators may be monitored semi-continuously, whereas robustness and safety are often checked through periodic tests or red-teaming.
Human agency and oversight	Override rate; escalation rate; task completion time; error rate; user satisfaction; reliance calibration.	Operational indicators can often be logged continuously, while user-centered measures are usually updated through periodic review or feedback collection.
Societal/environmental well-being	CO ₂ footprint of training and operation; number of model parameters; model runtime.	Typically documented through reporting cycles, retraining events, or sustainability audits.

judgments are naturally documented through statements of the form ‘performance below α ,’ or ‘robustness above β and fairness disparity below δ ,’ and the anchor regions used in the labeling procedure below are naturally expressed through interval-like constraints on selected coordinates. Axis-parallel partitions therefore provide a reasonable first model of how coarse governance judgments over trustworthiness profiles are translated into auditable level rules. Under this restriction, a natural candidate class is the family of decision trees $\mathcal{H}^{\text{tree}}$ [Breiman et al., 2017]. If their complexity is kept at bay, decision trees are interpretable via binary decision rules. The simplest form of the empirical learning problem becomes

$$\arg \min_{h \in \mathcal{H}^{\text{tree}}} \left\{ \frac{1}{N} \sum_{i=1}^N \ell(h(\mathbf{Q}_A(t_i)), Y_i) + \lambda \Omega(h) \right\}, \quad (1)$$

where ℓ is the loss function and $\Omega(h)$ penalizes complexity, for example via tree depth or number of leaves.⁴ Let us comment on this construction in three steps.

Safeguards against overfitting and enforcing robustness. Since any solution $\hat{T}_{A,N}$ of (1) is learned from finite historical audit data, standard *safeguards against overfitting* are required. In practice, hyperparameters controlling model complexity—for instance, tree depth, number of leaves, minimum leaf size, or pruning parameters—should be selected by out-of-sample validation. As the data are time-indexed, however, validation should respect temporal order, for instance through blocked train-test or cross-validation splits, so as to avoid time-relative information leakage.

A second requirement is *robustness of the learned trustworthiness-level rule itself*. Governance-relevant trustworthiness transitions should be stable transitions, not artifacts of brittle partition boundaries. In particular, a good learned rule should not oscillate sharply between adjacent trustworthiness levels because one dimension crosses the boundary repeatedly under minor measurement variation or small sample perturbations, e.g., decision trees can suffer from high variance [Breiman et al., 2017]. Candidate rules should therefore also be stress-tested for ‘boundary robustness:’ one should prefer rules whose level assignments remain stable under small admissible perturbations of trustworthiness

⁴One can also consider penalties that reflect the special normative status of selected trustworthiness dimensions.

profiles and under modest changes in the training data. Rules that exhibit short-horizon zig-zag transitions between levels under such perturbations should be treated as governance-inadequate and discarded [Ferrario, 2025]. Among comparably performing candidates, governance actors should prefer learned trustworthiness-level functions whose regime transitions are sparse, interpretable, and robust.

Decision trees are a choice. A K -level trustworthiness rule $\hat{T}_{A,N}$ can in principle be learned by many supervised models. Decision trees are nevertheless a natural first choice when trustworthiness levels are expected to arise from auditable threshold conditions on individual (or small groups of) dimensions. Since the goal of this lightweight framework is auditable AI lifecycle governance, a simple decision tree with standard protections against overfitting and safeguards for robustness provides an appropriate starting point. (They can be equivalently rewritten as collections of rules.) *Decision trees are not, however, always the right model class.* They should be abandoned when to-be-learned rules are too high-dimensional, interaction-heavy, or non-axis-aligned. In such settings, more complex models should be considered.

The dataset labeling procedure. A central challenge for the learning problem in (1) is the construction of the labels Y_i in the dataset $\mathcal{D}_N$ before AI deployment. Here, we assume a **labeling procedure** that produces reference labels Y_i through expert judgment and documented agreement rules during the AI design. The goal of this procedure is not to provide a fine-grained partition of the whole profile space $[0, 1]^n$, but rather to identify coarse and auditable regions corresponding to governance-relevant levels such as ‘good,’ ‘acceptable,’ and ‘risky.’ This labeling is close to chart review in medicine, structured expert adjudication in risk analysis, or severity classification in safety engineering. The same general logic is already familiar in AI governance practice: before deployment, expert committees, red-teaming exercises, or internal review procedures often identify clearly acceptable, borderline, or unacceptable system behaviors without attempting to map the entire space of possible cases in a fully granular way [Zicari et al., 2021, National Institute of Standards and Technology, 2023]. This coarseness is a starting point as the aim is to secure a documented and contestable governance basis for later learning.

Let $R_1, \dots, R_K \subseteq [0, 1]^n$ be expert-defined anchor regions associated with the K trustworthiness levels. As remarked above, these regions are naturally specified through interval-based constraints on selected trustworthiness dimensions. (This is one reason why choosing decision trees as a first model class is natural in the present setting.) These regions may be specified through interval-based constraints on selected trustworthiness dimensions by identifying combinations of dimension values that clearly correspond to acceptable or unacceptable system behavior in a given context. For instance, AI governance actors may agree that, for a given deployment context, profiles with accuracy and robustness below 0.67 and a fairness disparity metric above a threshold δ_{fair} should be labeled at least ‘risky,’ whereas profiles with accuracy above 0.85, robustness above 0.80, and fairness disparity below δ_{safe} may be labeled ‘good.’ As such rules are intentionally high-level, the collection $R_1, \dots, R_K$ need not cover the full profile space: borderline profiles may remain unassigned pending further expert review. Using these regions, together with pre-deployment observations collected in sandbox, shadow-mode, or pilot settings, AI governance actors construct a labeled dataset $\{(\mathbf{Q}(t_i), Y_i)\}_{i=1}^N$, where $Y_i \in [K]$ is the reference trustworthiness label of the profile $\mathbf{Q}(t_i) \in [0, 1]^n$ assigned by the labeling procedure.⁵

⁵Ambiguous cases may either be escalated to further review or excluded from the initial labeled set.

These characteristics arise naturally in real-world governance settings. In medicine, assigning a trustworthiness level to a clinical decision-support system may require specialist chart review, adjudication of delayed patient outcomes, subgroup analysis, and joint assessment by clinicians, data scientists, safety officers, and ethics or compliance personnel. Comparable costs arise in credit and insurance, where model-risk, compliance, and fairness teams must jointly interpret performance and disparate-impact evidence, and in industrial or public-sector applications, where engineering, legal, operational, and stakeholder expertise may be required to determine whether a system state remains acceptable. The resulting labels are often deliberately coarse because pre-deployment pilots cover only a small portion of the possible trustworthiness-profile space and because borderline cases cannot yet be resolved confidently. They may also be temporally bounded without being ill-posed: an expert label can be well justified for the initial version of a new AI product, its intended purpose, target population, workflow, and available evidence, while remaining authoritative only for a limited initial period of its lifecycle. As deployment evidence accumulates, or as the system, environment, users, or governance standard changes, such labels may need to be versioned, reviewed, refined, merged, or retired. Learning should therefore be understood as operationalizing a documented initial governance standard for routine monitoring and revealing when that standard no longer adequately represents the deployed system [Lu et al., 2018, Kulesza et al., 2019, Bernhardt et al., 2022, National Institute of Standards and Technology, 2023].

In summary, the goal of learning $\hat{T}_{A,N}$ is **to translate an expert-driven governance partition that is costly, sparse, delayed, discrete-in-time, and often coarse into an empirical trustworthiness-level rule for post-deployment use**. This learned rule has five governance functions. It (i) extends costly expert judgment to routine post-deployment monitoring, (ii) interpolates between expert-adjudicated anchor cases, (iii) makes the operative governance standard auditable through an explicit rule, (iv) detects local fragility through boundary margins and profile drift—see Section 4.3, and (v) reveals when the available labels, protocol, or monitored profiles are insufficient to support stable lifecycle governance. Depending on data collection, deployment-side variation, and model-selection constraints, the learned rule may *preserve*, *compress*, or *refine* the expert-defined scale. In some settings, *expert-rule preservation* may be the most informative outcome: if deployment generates little variation across trustworthiness regimes, or if labels remain sparse, noisy, or unstable, *learning may provide no governance-relevant improvement over the expert rule*. Accordingly, the learned rule does not replace periodic expert review under changing lifecycle conditions, serving as a monitoring rule that can indicate when expert intervention is needed, once embedded in a governance framework.

4 AI Lifecycle Governance Procedure

Our formal framework provides a quantitative and auditable layer for lifecycle governance that can be aligned with conformity assessment, technical documentation, post-deployment monitoring, and reassessment procedures across AI systems. To show this, we embed it within a *lightweight AI lifecycle governance procedure* as follows.

4.1 Structure of the AI Lifecycle Governance Procedure

Figure 1 summarizes the governance procedure in three connected phases, which we now discuss in some detail.

Phase 1: Pre-deployment governance and auditing procedure. This first phase establishes the normative and methodological conditions under which trustworthiness data can later be interpreted. The AI governance procedure

begins at design-time, once the AI system’s intended purpose and deployment context have been specified [EU AI Act, 2024]. At this stage, AI governance actors define the trustworthiness protocol, namely, the set of dimensions that are relevant for judging whether the system is worthy of trust in that context. They also operationalize these dimensions through their measures and aggregation choices, and determine the trustworthiness-level scale together with the auditable labeling procedure we discussed in Section 3.3. In regulated settings, these design choices also serve as inputs to conformity documentation by making explicit which trustworthiness dimensions are being monitored, how they are measured, and how level assignments are justified—see Annex IV, Sections 1–4 in the AIA.

Phase 2: Formalization and empirical learning. Here, trustworthiness-relevant data are collected in a pre-deployment setting, for example through sandboxing, shadow mode, or controlled pilot evaluation. These observations are then turned into a historical labeled dataset $\mathcal{D}_N$ to be used for the training, validation, and selection of an empirical trustworthiness-level rule—see Section 3.3. *This phase need not always yield a learned rule that improves on the expert-defined governance standard.* Where the available data do not support stable or informative refinement—for instance because trustworthiness regimes are weakly covered or labels remain too sparse or inconsistent—the expert rule may remain the operative governance rule. As part of the AI governance framework, the learned rule should be treated as *provisional* or simply be *discarded*.

Phase 3: Post-deployment monitoring and reassessment. After deployment, the learned trustworthiness rule $\hat{T}_{A,N}$ becomes part of an AI-governance workflow. The same documented protocol used to construct the initial dataset in Phase 1 provides the basis for monitoring trustworthiness profiles over time and deciding whether new data, system updates, shocks, or other changes warrant review or reassessment.

Post-deployment profiles $\mathbf{Q}_A(t)$ and predicted levels $\hat{T}_{A,N}(\mathbf{Q}_A(t))$ are not by themselves sufficient to augment the labeled dataset $\mathcal{D}_N$. A post-deployment retraining of the trustworthiness rule requires dataset update. This update is justified only if selected post-deployment cases are sent to a further *validation procedure* and assigned new reference labels under a documented labeling protocol in Phase 1. As the target variable may change through refinement of earlier labels or changes in the criteria under which they are assigned, this validation procedure of dataset update requires explicit versioning of the labeling protocol and documentation of which cases were labeled under which governance standard. This implies that part of the trustworthiness (re-)learning problem concerns robustness to label noise, adjudication procedures, and versioned labeling standards. In practice, the audit dataset should therefore document who assigned labels, under which protocol version, and how disagreement or uncertainty was resolved. Initial design-time labels may define a coarse governance partition of the trustworthiness space, whereas later post-deployment review may revise that partition, for instance by refining, merging, or otherwise re-specifying regions under an updated governance standard. Such post-deployment validation and label refinement are not peculiar to trustworthiness learning. Comparable practices are common in machine learning: clinical models are often trained or validated against expert-adjudicated labels when no simple gold standard is available; dataset quality is routinely improved under limited annotation budgets; and, more generally, the target concept itself may evolve over time, requiring refinement of the labeling scheme [Lu et al., 2018, Kulesza et al., 2019, Bernhardt et al., 2022]. For high-risk AI systems under EU regulation, such updates should be assessed under the ‘substantial modification’ logic of the AIA—see Art. 3(23); Art. 43(4). Where the change amounts to a substantial modification of the high-risk AI system, a new conformity assessment is required; by contrast, changes that were predetermined and documented in the technical documentation at the initial conformity assessment

AI Lifecycle Governance Procedure

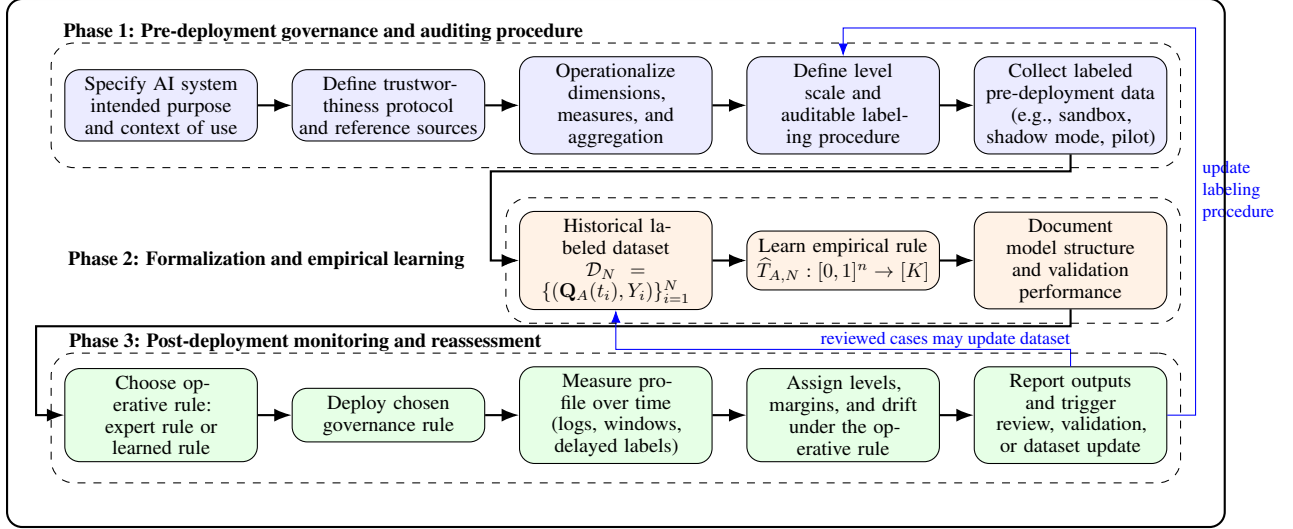


Figure 1: AI Lifecycle Governance Procedure. **Phase 1** specifies the AI system, trustworthiness protocol, measures, and labeling procedure. **Phase 2** constructs the initial labeled dataset and learns a candidate trustworthiness-level rule. **Phase 3** makes explicit the choice of operative governance rule—expert-defined or learned—and then supports post-deployment monitoring, reporting, validation, and possible dataset or protocol updates.

do not automatically count as substantial modification. Here, trustworthiness-level transitions provide a quantitative basis for documenting when system changes remain within the previously documented conformity assessment and when they may require review as they affect conformity.

One final remark. Post-deployment trustworthiness monitoring is typically *discrete* rather than *continuous*. The frequency with which $\mathbf{Q}_A(t)$ can be updated and $\hat{T}_{A,N}(\mathbf{Q}_A(t))$ monitored depends on the resources and evidence required to measure its components, as Table 1 shows. In some applications, such as recommender systems, certain dimensions may be refreshed at high frequency because engagement outcomes or system logs are produced at scale and with limited delay. In others, such as clinical decision support, some dimensions may only be updated sparsely because outcomes or expert validation become available slowly and at high cost. Accordingly, the governance procedure must document for each dimension not only its definition, but also its measurement *cadence* and any delay with which the corresponding evidence becomes available throughout all phases of the AI lifecycle governance procedure.

4.2 Human Roles, Decision Rights, and Control Measures

Each institution should assign explicit human roles and decision rights to the AI lifecycle governance procedure. While we believe that the precise job titles will vary across domains and organizations, five functions should remain distinguishable. First, a designated system or product owner is responsible for assembling the relevant evidence and ensuring that the procedure is maintained. Second, domain experts and representative users interpret the intended purpose, context of use, and domain-specific consequences of trustworthiness failures. Third, technical and data teams operationalize dimensions, maintain measurement pipelines, construct datasets, and implement candidate rules. Fourth, an independent validation, risk, compliance, safety, or ethics function challenges the selected dimensions, labels, assumptions, and validation results. Fifth, a designated governance authority—for instance, an AI governance

Phase	Main actors	Responsibilities and controls	Required output
Phase 1	System owner, domain experts, technical team, risk or compliance function	Define intended purpose, trustworthiness dimensions, measures, thresholds, labeling rules, review cadence, and escalation conditions. Require documented expert agreement and independent challenge.	Versioned protocol, labeling procedure, and responsibility register
Phase 2	Data and model teams, domain experts, independent validation function	Construct and label the dataset; assess data quality, regime coverage, class balance, temporal validation, rule stability, and performance against the expert rule. Development and approval should remain separated.	Dataset version, validation report, and decision on the operative rule
Phase 3	Monitoring team, system owner, operational users, multidisciplinary review body	Monitor profiles, levels, boundary margins, drift, incidents, and overrides. Alerts require human review. Relabeling, retraining, restriction, rollback, or continued deployment require documented approval.	Monitoring log, review decisions, and updated protocol or rule where needed
Assurance	Internal audit or independent assurance function	Periodically verify role assignment, segregation of duties, protocol versioning, evidence retention, escalation, and corrective actions.	Audit record and corrective-action plan

Table 2: Minimal allocation of human responsibilities and control measures across the AI lifecycle governance procedure. Organizational titles may vary, but evidence production, independent review, approval, monitoring, and reassessment should remain explicitly assigned.

committee or accountable senior officer—approves the operative rule and decides whether monitoring results require continued operation, intensified monitoring, expert review, system restriction, rollback, or reassessment. These functions should be supported by three general controls: segregation of duties; standard risk-management measures, including operational ownership, independent review, and documented escalation conditions; and human-authorization protocols. In summary, the proposed AI lifecycle governance procedure does not require a novel governance body, or exotic audit architectures. We believe it reasonably relies on standard organizational controls already used in model-risk management, quality assurance, safety governance, and regulated compliance. At least at this stage, it appears capable of increasing the transparency and auditability of trustworthiness judgments without materially increasing governance complexity through additional institutional layers. This does not imply that implementation is costless: measurement, expert labeling, monitoring, and review may remain resource-intensive. We summarize these considerations in Table 2.

4.3 Auditing Measures of Learned AI Trustworthiness

Once the AI system is designed and ready to be tested or deployed, governance actors need to report trustworthiness levels and other measures to support auditing of the AI system evolution. They have at their disposal (i) the measured trustworthiness profile $\mathbf{Q}_A(t) \in [0, 1]^n$, and (ii) the learned empirical trustworthiness-level rule $\hat{T}_{A,N}$. This yields the empirical trustworthiness-level trajectory $\hat{\tau}_{A,N}(t) := \hat{T}_{A,N}(\mathbf{Q}_A(t))$. In practice, two outputs provide basic information for governance: (i) the time series $(\mathbf{Q}_A(t_1), \dots, \mathbf{Q}_A(t_n))$ of the trustworthiness profile; and (ii) the time series $((\hat{\tau}_{A,N}(t_1), \dots, \mathbf{Q}_A(t_n)))$ of the predicted trustworthiness levels $\hat{\tau}_{A,N}$, together with marked events such as ‘shocks,’ system updates, or retrainings. Plotting these time series shows how the measured profile and the learned rule evolve over time. To complement these outputs, we introduce two diagnostics adapting standard MLOps-style drift monitoring and margin-to-boundary notions from active learning.

Boundary margin. The learned rule $\hat{T}_{A,N}$ defines a collection of trustworthiness regions $C_1, \dots, C_K \subseteq [0, 1]^n$. We define the boundary margin by

$$\hat{\kappa}_{A,N}(t) := \text{dist}(\mathbf{Q}_A(t), \partial C_{\hat{\tau}_{A,N}(t)}),$$

where dist denotes a distance function and $\partial C_{\hat{\tau}_{A,N}(t)}$ is the boundary of the trustworthiness region $C_{\hat{\tau}_{A,N}(t)} \ni \mathbf{Q}_A(t)$. Thus, if $\hat{\kappa}_{A,N}(t)$ is large, then $\mathbf{Q}_A(t)$ lies well inside its current trustworthiness plateau; if it is small, then $\mathbf{Q}_A(t)$ is close to a level transition. Thus, the boundary margin is a measure of (spatial) local fragility, indicating how close the system is to leaving its current trustworthiness regime. Here, closeness is contextual: the choice of dist , what counts as a small or large distance from the boundary is defined as part of the AI governance procedure documentation.

Profile drift over time. The profile drift compares the measured trustworthiness profile at time t with its past value over a fixed horizon $h > 0$:

$$D_A(t; h) := \frac{1}{\sqrt{n}} \|\mathbf{Q}_A(t) - \mathbf{Q}_A(t - h)\|_2.$$

$D_A(t; h)$ measures how much the trustworthiness profile has changed over time, capturing the extent to which the AI system at time t has moved in trustworthiness space over the last h time steps. The choice of temporal horizon is part of the AI governance procedure and must be documented. The measures $\hat{\kappa}_{A,N}$ and D_A play complementary roles: The boundary margin tells us whether the current state is close to a regime boundary. The profile drift tells us whether the system is changing rapidly over time instead.

4.4 A Simple Reporting Artifact

Table 3 shows a simple governance artifact for documenting how trustworthiness levels are defined, learned, and monitored over time that can accompany the proposed framework of AI governance. The artifact is embedded in the AI lifecycle governance procedure as follows. First, it is completed in pre-deployment settings (Phases 1–2). Then, it is updated post-deployment (Phase 3), when new measurements, delayed outcomes, review decisions, and reassessment triggers arise. By recording design choices, the learned rule, and lifecycle monitoring information, the artifact supports three functions. It supports *pre-deployment documentation* by making explicit how trustworthiness is defined, measured, and labeled before the empirical rule is learned. This is aligned with the conformity assessment documentation in the AIA—see Art. 43. It supports *post-deployment monitoring* by providing a compact schema for reporting the system’s current trustworthiness profile, level, and diagnostics over time. Finally, it supports *reassessment* by making review conditions, rule updates, and possible changes in the labeling protocol more transparent and contestable. These last two functions align with the governance of post-deployment monitoring as per the AIA—see Art. 3(23), Art. 43(4), and Art. 72.

5 Proof-of-Concept Simulation Study

We illustrate the framework through a controlled proof-of-concept simulation study comprising three experiments. *The aim of this study is not to establish external validity across domains*, but to illustrate how the proposed governance objects behave under controlled conditions that simulate governance-relevant events and actions. The simulation setup is intentionally lightweight relative to the complexity of real-world trustworthiness learning. Accordingly, the experiments should also be read as illustrating different possible outcomes of trustworthiness learning: in some cases, learning can closely recover or improve the granularity of the expert-defined rule, whereas in more demanding settings it may approximately track it or even be discarded. Further, in practice, trustworthiness protocols may involve many more dimensions and measures; for instance, [Kemmerzell et al. \[2025\]](#) discuss seven trustworthiness dimensions and more

Item	Minimal content
System and context	Intended purpose, context of use, operational constraints, excluded uses.
Trustworthiness protocol	Selected dimensions, inclusion rationale, omitted dimensions where relevant, hard/soft dimensions if used.
Measures and aggregation	Operational definitions, normalization, aggregation choices, data sources, measurement cadence, evidence windows, limitations.
Level scale and labels	Scale $1, \dots, K$, level interpretation, labeling procedure, audit-label criteria, protocol version.
Learned rule and validation	Model family, hyperparameter selection, complexity control, validation strategy, learned tree or equivalent rule.
Learning value over expert rule	Whether learning improves, compresses, confirms, or fails to justify replacing the expert-defined rule.
Data and label adequacy	Regime coverage, class balance, missingness, delayed outcomes, measurement uncertainty, label disagreement, protocol drift.
Current monitoring state	Current profile, predicted level, boundary margin, profile drift, marked shocks, updates, or retrainings.
Governance responsibility	Actor or unit responsible for maintaining the artifact, approving protocol changes, and escalating review.
Update and evidence record	Update cadence; logs, validation reports, audit decisions, expert-review notes, and protocol/version history.
Escalation and reassessment	Conditions for no action, intensified monitoring, expert review, relabeling, retraining, documentation update, or reassessment.

Table 3: Minimal governance artifact for documenting trustworthiness-level definition, learning, monitoring, and reassessment across the three AI lifecycle governance phases.

than 90 associated measures. In data-intensive applications, such as LLM-based conversational agents, learning an auditable rule may therefore require higher-dimensional profiles, multiple measures per dimension, more expressive model classes, and richer model-selection procedures than those used in the present proof of concept. Full experimental details are reported in the Appendix.

5.1 Experiment 1: Lifecycle Baseline

Experiment 1 serves as the baseline for the simulation study. It considers two AI systems with the same minimal trustworthiness protocol, consisting of two directly measured dimensions: accuracy and robustness. No within-dimension aggregation is used at this stage. The first system is a *recommender or ranking system*, for which trustworthiness-relevant evidence becomes available at high frequency, so that the resulting trustworthiness time series is relatively dense. The second is a *clinical decision-support system*, for which some trustworthiness-relevant quantities—especially outcome-based accuracy measures—are only available after a delay and therefore updated less frequently, yielding a sparser trustworthiness time series. For both systems, the two dimensions evolve under slow degradation, a small number of adverse shocks, and retraining or update events, with observed values fluctuating around latent lifecycle states. The simulated shocks represent disruptions that may affect system trustworthiness. In the recommender setting, shocks can be interpreted as sudden distribution shifts or changes in user interaction patterns that reduce ranking performance or robustness. In the clinical decision-support setting, they can be interpreted as workflow changes, or performance degradations revealed through delayed outcome review. The simulated updates represent stylized mitigation or remediation actions, such as retraining, interface adjustments, or revised monitoring procedures. Experiment 1 provides the *simplest* illustration of the framework, showing how it can support trustworthiness monitoring.

5.2 Experiment 2: Single-System Lifecycle with Asynchronous Monitoring

Experiment 2 extends the baseline to a higher-dimensional profile while preserving the single-system setting. The use case is an AI system for assessing the creditworthiness of loan applicants and the trustworthiness protocol consists of four dimensions: performance, explainability, robustness, and fairness. Each dimension is operationalized through a single metric with its own monitoring cadence.

Unlike Experiment 1, the four dimension-level time series are not observed at the same frequency: they are measured on different schedules and therefore yield arrays of different lengths. Accordingly, trustworthiness levels are evaluated only at discrete governance-review times, where each coordinate of $\mathbf{Q}_A(t)$ is given by the latest available measurement for that dimension. The dimension scores evolve under slow degradation, a small number of adverse shocks, and retraining or update events, with observed values fluctuating around latent lifecycle states. As in Experiment 1, the simulated shocks represent stylized lifecycle events, such as a deterioration in predictive performance due to portfolio shift or the emergence of fairness concerns after internal review or external auditing. Because different trustworthiness dimensions are measured through different evidential processes, the effects of such events need not become visible simultaneously across dimensions. The simulated updates represent corrective or governance actions, such as recalibration, model revision, or fairness-mitigation interventions. This experiment evaluates whether the framework remains interpretable and stable once trustworthiness profiles become higher-dimensional and asynchronously monitored.

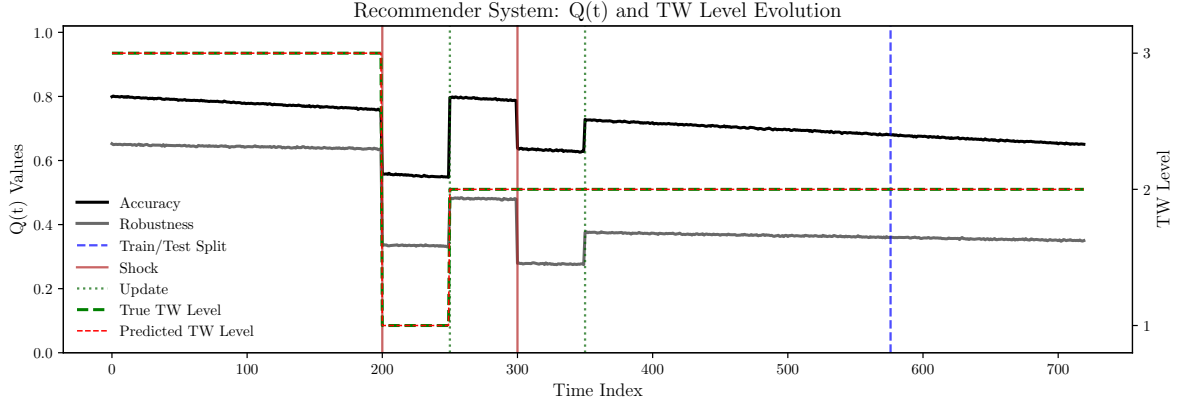
5.3 Experiment 3: AI Deployment and Governance-Context Comparison

Experiment 3 compares two deployed copies of the same clinical decision-support system in two hospitals. The copies share intended purpose, trustworthiness protocol, and monitoring cadence, but differ in both the stability of their deployment environments and the granularity of the expert-defined trustworthiness-level rules used by the local governance teams. The trustworthiness protocol consists of six dimensions: performance, calibration, robustness, fairness, explainability, and oversight quality.

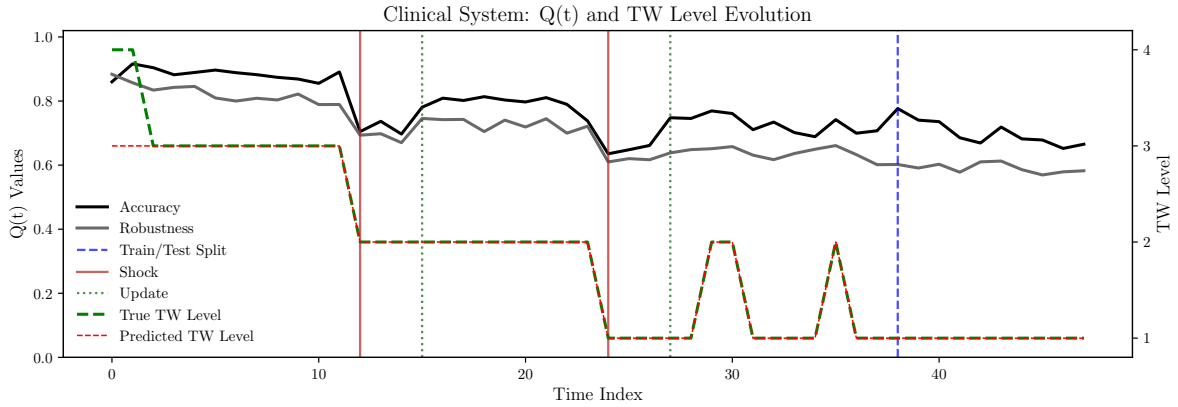
Hospital Alpha represents a stable and controlled deployment environment governed by a coarse expert-defined rule. Its measured trustworthiness profile undergoes only very slow degradation and remains within the locally accepted highest trustworthiness regime throughout the monitored lifecycle. In such a setting, lifecycle data do not contain enough regime variation to support a meaningful learning exercise. Hospital Alpha is therefore treated as a case in which *the expert-based rule remains the appropriate operative governance rule*. Hospital Beta represents a brittle deployment environment exposed to lifecycle events that combine adverse shocks and corrective updates that move the system across multiple trustworthiness regimes. Experiment 3 studies when trustworthiness-level learning is warranted in the first place, and how different institutional attitudes toward trustworthiness measurement can lead to different governance practices for copies of the same AI system. It shows that, in some settings, the methodology may recommend retaining the expert-defined rule rather than using a learned rule.

6 Results

We summarize the experimental results and report all details in the Appendix.



(a) Recommender system: trustworthiness profile and reference/predicted trustworthiness levels



(b) Clinical decision-support system: trustworthiness profile and reference/predicted trustworthiness levels

Figure 2: **Experiment 1: lifecycle baselines.** Trustworthiness-profile and level trajectories for a densely monitored recommender system and a sparsely monitored clinical decision-support system. The two systems use the same two-dimensional profile structure but differ in monitoring cadence, lifecycle dynamics, and expert-defined trustworthiness scales.

6.1 Experiment 1: Lifecycle Baseline

Experiment 1 shows that the same two-dimensional formalism yields different AI lifecycle patterns across contexts, see Figure 2. In the recommender setting, accuracy declines gradually while robustness is more shock-sensitive; the learned trustworthiness levels track the reference series almost exactly, with a sharp drop after the first shock, partial recovery after the update, and a later stabilization in an intermediate regime rather than a return to the highest level. Consistently, the boundary margin collapses near shocks, rises after updates, and then decreases slowly during the long post-update drift, while the drift indicator is nearly flat except for narrow peaks at shocks and updates. The learned partition is simple. In the clinical AI system setting, the series is more jagged and the level dynamics are coarser: the system starts in a high-trustworthiness regime, degrades after shocks, shows only brief recoveries after updates, and eventually remains in a low-trustworthiness state. Here too, boundary margin is smallest near regime changes, while drift peaks at shocks and updates and remains noisier between events because measurements are sparse. The learned partition remains geometrically simple and captures the intended coarse ordering of the four clinical trustworthiness levels.

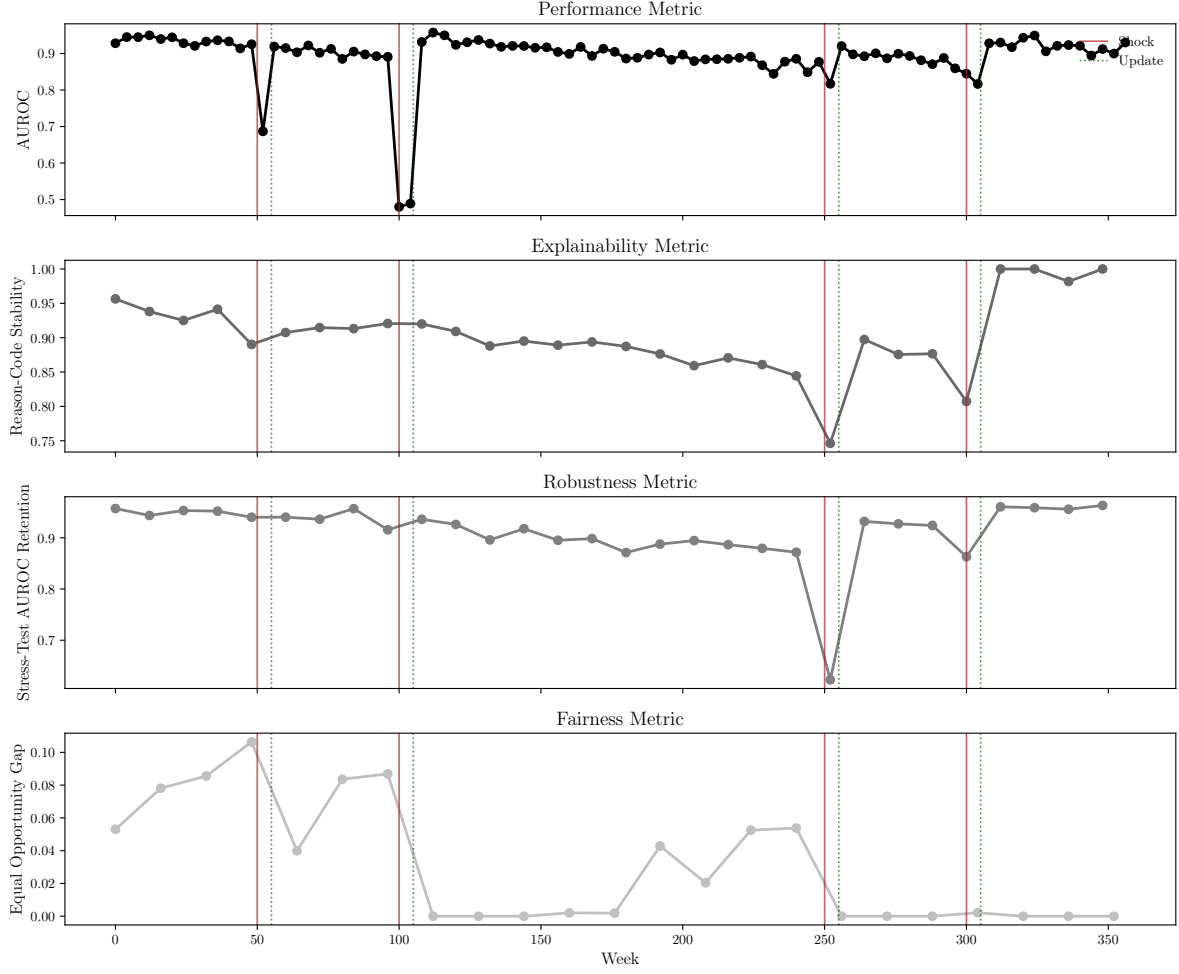
6.2 Experiment 2: Single-System Lifecycle with Asynchronous Monitoring

Experiment 2 shows that the framework remains interpretable once trustworthiness is monitored through four asynchronously measured dimensions in a lending use case, see Figure 3. During the training phase, all four trustworthiness dimensions exhibit slow degradation punctuated by adverse shocks and subsequent updates. Nevertheless, the system remains at the highest trustworthiness level for most of training, which suggests that the repair procedures associated with the updates were effective at preserving the overall trustworthiness regime. The main temporary drops from trustworthiness level 4 occur around the shock events introduced during development, pilot testing, or shadow-mode evaluation, after which the system returns to its previous regime.

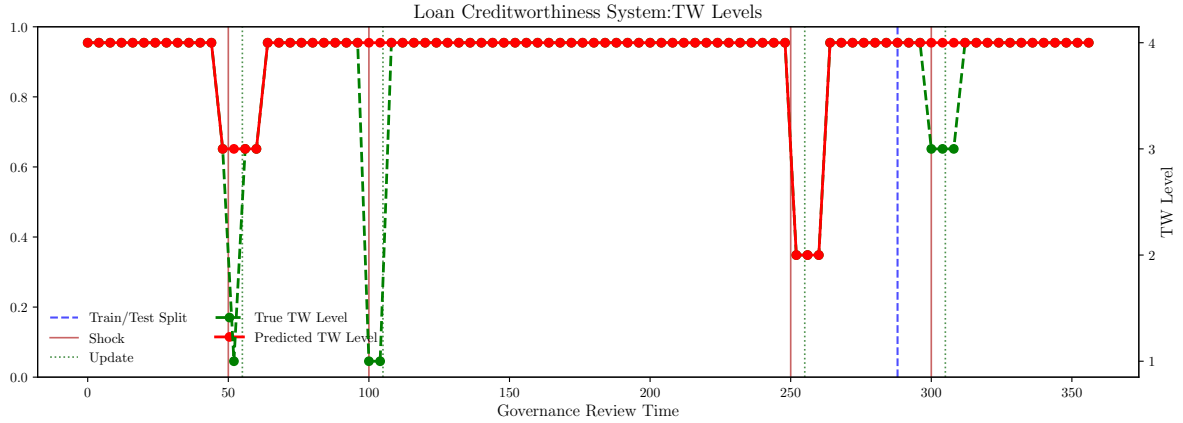
This stability largely continues in the post-deployment phase. The selected tree retains reasonable plain accuracy after deployment, but balanced accuracy and macro-F1 are lower because the lower trustworthiness regimes are weakly represented. The boundary-margin series further suggests that, even when the system is assigned to the highest learned trustworthiness regime, its profile can remain close to a learned transition away from that regime. In the selected tree, these transitions are mainly driven by fairness and explainability thresholds, reflecting the compression of the expert-defined four-level scale into a simpler operational rule. Finally, the raw-metric panel highlights that, because the four dimensions are sampled at different cadences, the effects of a shock and its corresponding update do not always appear simultaneously across dimensions. This is especially visible for fairness and robustness, whose measured recovery may occur only at the next scheduled observation.

6.3 Experiment 3: AI Deployment and Governance-Context Comparison

Experiment 3 shows that the usefulness of empirical trustworthiness-level learning depends on both lifecycle variability and institutional labeling granularity, see Figure 4. Hospital Alpha represents a stable controlled environment governed by a coarse local expert rule. Its measured trustworthiness profile undergoes only very slow degradation and all observations remain in trustworthiness level 4. In this case, learning a trustworthiness-level rule from lifecycle data would be uninformative: the appropriate operative rule is therefore the expert-defined governance rule, not a learned classifier. Hospital Beta is governed by a more refined expert-defined rule and exposed to lifecycle events—adverse shocks and corrective updates—that generate substantially more lifecycle variation. Its reference trajectory occupies all four levels, making empirical learning governance-informative. However, the learned rule does not fully recover the refined expert scale: it performs reasonably on the training segment but deteriorates on the held-out post-deployment segment, and it compresses the four-level reference scale into a coarser operational rule. Thus, the governance framework does not simply produce a learned rule, but it also reveals when this rule is too coarse, too fragile, or insufficiently sensitive to rare low-trustworthiness regimes. The diagnostics support this interpretation. Hospital Alpha shows only small profile drift, consistent with slow degradation inside a locally accepted highest-regime envelope. Hospital Beta exhibits stronger variability in trustworthiness space, including profiles lying directly at learned regime boundaries. Thus, two hospitals deploying copies of the same AI system may reasonably differ in both local governance granularity and the value of empirical learning. In the Alpha environment, learning adds little beyond the expert rule. In the Beta environment, learning is useful because its limitations reveal compression, boundary fragility, and post-deployment instability that warrant further review.

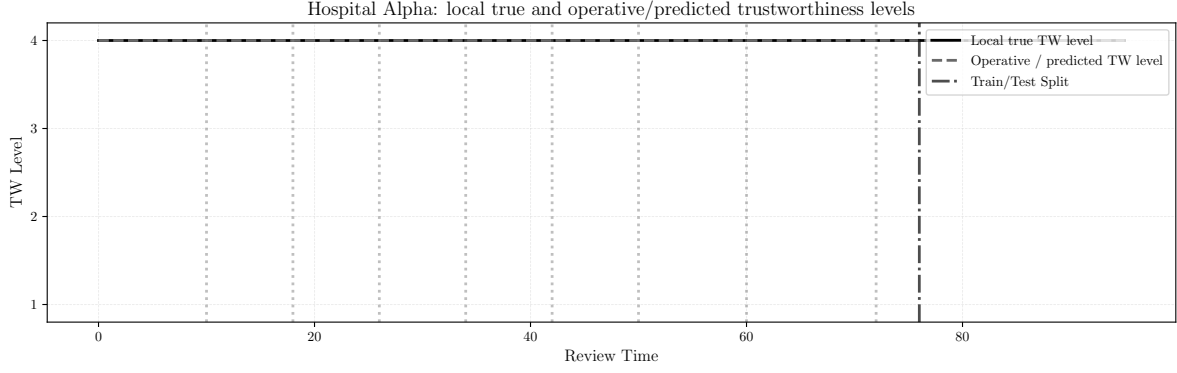


(a) Asynchronously observed raw trustworthiness metrics

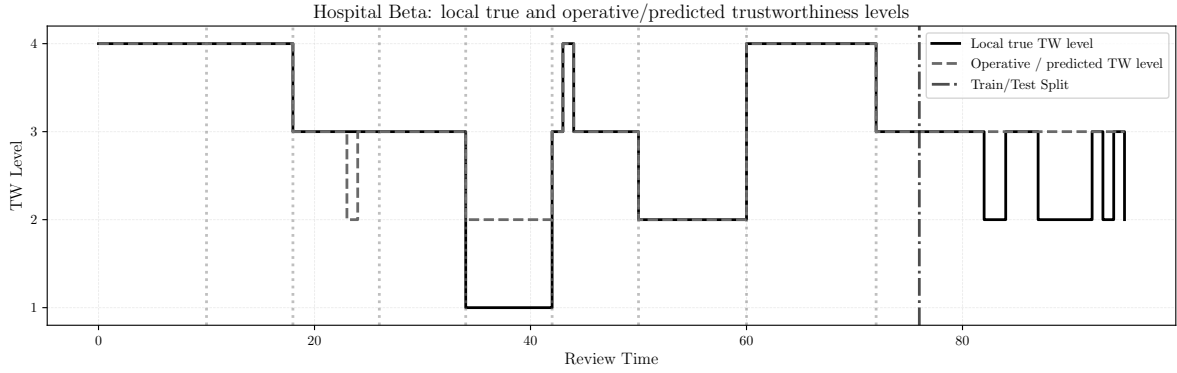


(b) Latest-available trustworthiness profile and reference/predicted trustworthiness levels

Figure 3: Experiment 2: single-system lifecycle with asynchronous monitoring. The upper panel reports the raw trustworthiness metrics observed at different monitoring cadences. The lower panel reports the resulting latest-available trustworthiness profiles and their reference and predicted trustworthiness levels in the lending use case.



(a) Hospital Alpha: expert-defined trustworthiness levels



(b) Hospital Beta: expert-defined and predicted trustworthiness levels

Figure 4: **Experiment 3: AI Deployment and Governance-Context Comparison.** The two panels compare simulated copies of the same clinical decision-support system. Hospital Alpha remains in the highest trustworthiness regime under a coarse local expert rule, whereas Hospital Beta moves across multiple regimes under a more refined expert-defined rule and lifecycle events involving adverse shocks and corrective updates.

7 Discussion and Conclusions

We introduced a lightweight methodology for auditable trustworthiness levels in AI governance that comprises (i) a formal framework for learning trustworthiness levels, and (ii) an AI governance procedure for documenting, monitoring, and reassessing trustworthiness throughout the AI lifecycle.

This methodology offers two main contributions. First, it makes trustworthiness judgments more explicit and contestable: dimensions, measures, aggregation choices, labeling procedures, learned rules, and monitoring outputs can all be documented and revised over time. Second, it connects pre-deployment governance with post-deployment monitoring by converting a comparatively static design-time governance standard into a dynamic monitoring layer that can be repeatedly applied to new trustworthiness profiles. *The resulting rule should therefore be understood as a tool for AI lifecycle governance rather than as a stand-alone compliance mechanism.* It provides a contestable evidence-based layer that can inform conformity assessment, post-market monitoring, and reassessment by making level assignments, transitions, margins, drift, and protocol updates explicit over time for different types of AI systems [EU AI Act, 2024].

A few possible criticisms deserve clarification. First, *the formal framework is algorithmically lightweight.* This is a feature of our approach. In governance settings, simplicity supports auditable threshold-based reasoning, facilitates

documentation, and makes regime transitions easier to review and contest. Thus, our goal was not to introduce a new state-of-the-art machine learning algorithm. At the same time, our framework is not tied to one fixed model class. More expressive models become appropriate when expert-defined trustworthiness regions are no longer threshold-based; the trustworthiness protocol becomes high-dimensional and includes multiple measures per dimension; richer data collection enables more complex robustness or scoring analyses; or governance actors wish to impose trustworthiness-sensitive penalties or stronger stability constraints under a broader range of shocks and update strategies. Depending on the structure of the expert rule, the available data, and the simplicity constraints imposed during model selection, the learned rule may preserve, compress, or refine the original scale.

Second, *learning may add little or nothing beyond the expert-defined rule*. This can happen when deployment generates too little informative variation across trustworthiness regimes, when difficult cases are filtered by existing governance interventions, when monitored profiles are weakly informative, or when labels are sparse, noisy, or unstable over time. In such settings, the most responsible outcome may be to retain the expert rule as the operative governance standard.

Third, *the framework does not eliminate the need for expert judgment in assigning trustworthiness labels*. This is not peculiar to our proposal, but a standard feature of governance-oriented assessment in domains where acceptable, borderline, and unacceptable system states must be determined under context-specific normative and operational constraints. Our framework assumes that reference labels are produced under a documented adjudication procedure that is specific to the relevant institution, domain, and governance setting: it does not attempt to standardize such procedures across all contexts. For this reason, what matters is that the procedure used to generate labels at any given institution be documented and versioned.

Finally, *the framework is only as informative as the trustworthiness object that an institution can actually construct and maintain*. In real-world settings, the main difficulty may lie less in learning the trustworthiness-level rule than in defining a meaningful protocol, aggregating heterogeneous measures, obtaining sufficiently informative profiles over time, and producing stable reference labels. Where dimensions are weakly measured, labels are sparse or inconsistent, classes are highly imbalanced, or protocol versions drift over time, the learned rule may become coarse, unstable, or effectively uninformative. The methodology is therefore best understood as an institutionally dependent governance layer: it becomes actionable only when the underlying measurement, aggregation, labeling, and review procedures are themselves documented and sufficiently mature. Future work should test the framework on real domain-specific cases, compare decision trees with other interpretable model classes, and refine validation and relabeling procedures under institution-specific protocols. Further work should also study robustness under label noise and protocol drift, and extend the approach to multi-agent settings and institutional workflows.

Acknowledgments

We acknowledge partial support by the Swiss National Science Foundation (SNSF), grant no. 229061.

8 Appendix

8.1 Simulation Study: Details on the Implementation

Common Simulation Logic for Trustworthiness Dimensions. Across the experiments, each trustworthiness-relevant quantity is generated from the same basic lifecycle model. Writing $z(t)$ for the raw or latent value of one dimension at time t , we simulate

$$z(t) = \text{clip}_{[0,1]} \left(b - d\tau_t + \varepsilon^{(1)}(t) + \sum_j s_j \mathbf{1}\{t \geq t_j^{\text{shock}}\} + \sum_k u_k \mathbf{1}\{t \geq t_k^{\text{update}}\} + \varepsilon^{(2)}(t) \right),$$

where b is a baseline level, d a drift coefficient, $\tau_t = t/(T-1)$ a normalized time index, s_j and u_k are the magnitudes of shocks and updates, and $\text{clip}_{[0,1]}$ truncates the result to the unit interval. The term $b - d\tau_t + \varepsilon^{(1)}(t)$ represents a noisy drift component, namely a slowly degrading baseline trajectory perturbed by small process-level fluctuations. The second noise term $\varepsilon^{(2)}(t)$ represents additional observation-level variability before clipping. Both noise terms follow a normal distribution with zero mean; their standard deviations differ across experiments.

The generated quantity $z(t)$ is then interpreted as a trustworthiness coordinate according to the metric convention of each experiment. In Experiment 1, the simulated accuracy and robustness values are already the two trustworthiness coordinates. In Experiment 2, the simulated raw metrics are first passed through a normalization layer that maps them into governance coordinates in $[0, 1]$ before the trustworthiness profile is constructed. This normalization layer makes heterogeneous metrics comparable at the level of the governance profile while preserving their metric orientation. Thus, performance, explainability, and robustness are normalized so that higher values indicate better quality, whereas fairness is represented as an Equal Opportunity Gap, for which lower values indicate better fairness. In Experiment 3, the same convention is used: all coordinates except fairness are interpreted as higher-is-better quality measures, while fairness is represented as a gap coordinate.

Common decision-tree learning procedure. Trustworthiness-level learning is implemented as an ordinal multi-class classification task. The labels Y_i are discrete trustworthiness levels, ordered from lower to higher trustworthiness. The recommender case in Experiment 1 uses three levels, whereas the clinical case in Experiment 1, Experiment 2, and Hospital Beta in Experiment 3 use four levels. Hospital Alpha in Experiment 3 is the exception: because all observations remain in level 4 under the local expert rule, no empirical classifier is learned and the expert-defined rule remains operative.

For each learned rule, the available data are split chronologically into a training segment containing the first 80% of observations and a test segment containing the remaining 20%, so that evaluation preserves temporal order. On the training segment, we fit a `DecisionTreeClassifier` and tune its main structural hyperparameters by temporal cross-validation using `TimeSeriesSplit`. The tree itself is trained as a classification model, using either the gini or entropy split criterion. Because the labels are ordered, model selection is driven primarily by mean absolute error over the predicted and reference levels, which treats errors between adjacent levels as less severe than errors across more distant levels. Accuracy, balanced accuracy, and macro-F1 are also recorded as complementary multi-class performance measures. The common search grid varies the split criterion, the maximum number of leaves, the maximum tree depth,

and the cost-complexity pruning parameter `ccp_alpha`. Table 4 reports the common hyperparameter grid used across the learned experiments.

Table 4: Common decision-tree hyperparameter grid used for all learned trustworthiness-level rules.

Hyperparameter	Candidate values
<code>criterion</code>	{ <code>gini</code> , <code>entropy</code> }
<code>max_leaf_nodes</code>	{3, 4, 5, 6, 8, 10, 12, <code>None</code> }
<code>max_depth</code>	{2, 3, 4, 5, 6, 8, <code>None</code> }
<code>ccp_alpha</code>	{0, 10^{-4} , 10^{-3} , 10^{-2} , 10^{-1} }

After hyperparameter tuning, the best tree is refit on the full training segment and then used to generate predicted trustworthiness levels over the entire lifecycle trajectory.

8.2 Experiment 1

This appendix provides the technical details of Experiment 1, the two-dimensional lifecycle baseline introduced in Section 5.1, and displays boundary margin and profile drift over time in Figure 5. For the profile-drift diagnostic, we use a seven-step horizon for the dense recommender trajectory and a one-step horizon for the sparse clinical trajectory. Experiment 1 uses two directly measured trustworthiness dimensions, accuracy and robustness, with no within-dimension aggregation.

Simulation design. The recommender trajectory is generated at a dense monitoring cadence, whereas the clinical trajectory is generated at a sparse cadence; both follow the common simulation logic above, but differ in step count, event timing, and expert-defined trustworthiness scale. Table 5 summarizes the main design choices. The two noise terms use the `noise_std` parameter specified for each trajectory. For the dense recommender trajectory, we set `noise_std` = 0.001, so the process-level perturbation has standard deviation 0.0005 and the observation-level perturbation has standard deviation 0.001. For the sparse clinical trajectory, we set `noise_std` = 0.02, so the corresponding standard deviations are 0.01 and 0.02, respectively.

Table 5: Experiment 1: simulation design summary. The recommender system is simulated over 720 dense daily-like observations; shocks take place at steps (200, 300) and updates at (250, 350). The clinical system is simulated over 48 sparse monthly-like observations; shocks take place at steps (12, 24) and updates at (15, 27).

System	Dimension	Baseline b	Drift d	Cadence	Obs.	Shock effects	Update effects
Recommender	Accuracy	0.80	0.15	dense / daily-like	720	−(0.20, 0.15)	(0.25, 0.10)
Recommender	Robustness	0.65	0.05	dense / daily-like	720	−(0.30, 0.20)	(0.15, 0.10)
Clinical	Accuracy	0.90	0.18	sparse / monthly-like	48	−(0.15, 0.10)	(0.10, 0.10)
Clinical	Robustness	0.85	0.16	sparse / monthly-like	48	−(0.10, 0.10)	(0.05, 0.05)

Expert-based trustworthiness-level rules. Table 6 reports the expert-defined trustworthiness-level rules used to generate the reference labels in Experiment 1. The recommender and clinical settings use different governance scales, reflecting their different intended purposes and risk profiles.

Learning results. Table 7 shows that, in the dense recommender setting, the learned trustworthiness rule reproduces the expert-based levels perfectly on both the training and test segments. In the sparse clinical setting, performance

Table 6: Experiment 1: expert-based trustworthiness-level rules.

System	Level	Rule	Interpretation
Recommender	1	$q_{acc} < 0.60$	risky
Recommender	2	$q_{acc} \geq 0.60$ and $q_{rob} < 0.60$	acceptable
Recommender	3	$q_{acc} \geq 0.60$ and $q_{rob} \geq 0.60$	good
Clinical	4	$\min(q_{acc}, q_{rob}) \geq 0.85$	excellent
Clinical	3	$0.75 \leq \min(q_{acc}, q_{rob}) < 0.85$	good
Clinical	2	$0.65 \leq \min(q_{acc}, q_{rob}) < 0.75$	acceptable
Clinical	1	$\min(q_{acc}, q_{rob}) < 0.65$	risky

remains strong overall, with near-perfect training accuracy and perfect test accuracy, while balanced accuracy and macro-F1 on the training segment are lower because some trustworthiness levels are only weakly represented.

Table 7: Experiment 1: predictive performance of the learned trustworthiness-level rules. In both settings, temporal cross-validation selected the same simple tree configuration (gini, max_depth=2, max_leaf_nodes=3, ccp_alpha = 0). Abbreviations: tr=train, te=test, Acc=accuracy, Bal.Acc=balanced accuracy.

System	MAE (tr / te)	Acc. (tr / te)	Bal. Acc. (tr / te)	Macro-F1 (tr / te)
Recommender	0.00 / 0.00	1.00 / 1.00	1.00 / 1.00	1.00 / 1.00
Clinical	0.05 / 0.00	0.95 / 1.00	0.75 / 1.00	0.73 / 1.00

In both cases, temporal cross-validation selected the same shallow tree, which is consistent with the baseline role of Experiment 1 as an intentionally simple and highly interpretable proof of concept. In the sparse clinical baseline, the expert-defined governance scale contains four levels, whereas the cross-validated tree recovers an effective three-regime operational rule. We treat this as evidence that, under strong simplicity pressure and limited support for one regime in the simulated trajectory, the learned rule may compress rather than refine the expert scale.

8.3 Experiment 2

This appendix provides the technical details of Experiment 2, the single-system lifecycle baseline introduced in Section 5.2 and displays boundary margin and profile drift over time in Figure 6.

Simulation design. Experiment 2 simulates a single loan-creditworthiness AI system over 360 time steps. Four latent trustworthiness dimensions are generated using the common simulation logic above; the dimensions are performance, explainability, robustness, and fairness. Performance, explainability, and robustness all start from a high baseline ($b = 0.95$) and drift downward. The fairness coordinate is modeled as a fairness-gap trajectory, with lower values indicating better fairness, and evolves with a negative drift, so that the gap tends to widen over time in the absence of corrective updates. The resulting raw metrics are then normalized into governance coordinates in $[0, 1]$, and governance-review profiles are constructed at the union of all measurement times by carrying forward the latest available observation for each dimension. Table 8 summarizes the main design choices. For all four latent metric trajectories in Experiment 2, we set `noise_std` = 0.01. Thus, the process-level perturbation $\varepsilon^{(1)}(t)$ has standard deviation 0.005, while the observation-level perturbation $\varepsilon^{(2)}(t)$ has standard deviation 0.01. The subsequent normalization step introduces no additional noise; it only maps the simulated raw metric values into governance coordinates in $[0, 1]$.

Table 8: Experiment 2: simulation design summary. Shocks occur at time steps (50, 100, 250, 300) and updates at (55, 105, 255, 305). Governance-review profiles are evaluated at the union of all metric timestamps, yielding 90 review points.

Dimension	Raw metric	Baseline b	Drift d	Cadence	Obs.	Shock effects	Update effects
Performance	AUROC	0.95	0.20	every 4 steps	90	$-(0.25, 0.40, 0.05, 0.05)$	$(0.25, 0.45, 0.10, 0.10)$
Explainability	Reason-Code Stability	0.95	0.18	every 12 steps	30	$-(0.10, 0.20, 0.10, 0.05)$	$(0.10, 0.22, 0.15, 0.20)$
Robustness	Stress-Test AUROC Retention	0.95	0.10	every 12 steps	30	$-(0.15, 0.25, 0.25, 0.05)$	$(0.15, 0.25, 0.30, 0.10)$
Fairness	Equal Opportunity Gap	0.05	-0.30	every 16 steps	23	$(0.15, 0.15, 0.05, 0.05)$	$-(0.20, 0.30, 0.20, 0.15)$

Expert-based trustworthiness-level rules. Table 9 reports the expert-defined trustworthiness-level rule used to generate the reference labels in Experiment 2. Here, $q_{\text{perf}}, q_{\text{exp}}, q_{\text{rob}} \in [0, 1]$ are normalized so that higher values indicate better quality, whereas $q_{\text{fair}} \in [0, 1]$ is a normalized fairness-gap coordinate for which lower values indicate better fairness.

Table 9: Experiment 2: expert-based trustworthiness-level rules.

Level	Rule	Interpretation
1	$q_{\text{perf}} < 0.50$ or $q_{\text{fair}} > 0.35$	risky
4	$q_{\text{perf}} \geq 0.75, q_{\text{exp}} \geq 0.75, q_{\text{rob}} \geq 0.75$, and $q_{\text{fair}} \leq 0.20$	excellent
3	$q_{\text{perf}} \geq 0.65, q_{\text{exp}} \geq 0.65, q_{\text{rob}} \geq 0.65$, and $q_{\text{fair}} \leq 0.25$, provided that Level 4 does not apply	good
2	otherwise	acceptable

Learning results. Table 10 reports the predictive performance of the learned trustworthiness-level rule in the asynchronous lending setting. First, we note that the resulting reference-label distribution is highly imbalanced. Across the 90 governance-review profiles, 78 observations are assigned to level 4, while levels 1 and 2 occur three times each and level 3 occurs six times. Under the chronological 80/20 split, the training segment contains all four levels but only sparse support for levels 1, 2, and 3; the held-out segment contains only levels 3 and 4. On the training segment, the learned rule performs well, with low ordinal error ($\text{MAE} = 0.11$), near-perfect accuracy (0.96), and good balanced accuracy (0.75) and macro-F1 (0.71). Temporal cross-validation selected a shallow tree (gini , $\text{max_depth}=2$, $\text{max_leaf_nodes}=3$, $\text{ccp_alpha} = 0$), which is consistent with the aim of learning a compact and auditable governance rule. Performance on the test segment is weaker but still reasonable for a controlled proof of concept: test accuracy remains 0.833, while the larger ordinal error ($\text{MAE} = 0.167$), lower balanced accuracy (0.500), and lower macro-F1 (0.455) indicate that the held-out portion of the trajectory contains more difficult or weakly represented trustworthiness regimes. As in Experiment 1, the cross-validated tree appears to recover a simplified operational rule relative to the expert-defined governance scale. We interpret this not as a defect of the framework, but as evidence that, under strong simplicity pressure and limited support for some regimes, the learned rule may merge neighboring expert-defined levels rather than preserve the full granularity of the original scale. Thus, Experiment 2 illustrates a characteristic governance use of the framework: the learned rule is simple and largely tracks the dominant high-trustworthiness regime, but it also reveals that rare low-trustworthiness states may be insufficiently supported under the available lifecycle trajectory.

Table 10: Experiment 2: predictive performance of the learned trustworthiness-level rule. Temporal cross-validation selected a shallow tree configuration (gini , $\text{max_depth}=2$, $\text{max_leaf_nodes}=3$, $\text{ccp_alpha} = 0$). Abbreviations: tr=train, te=test, Acc.=accuracy, Bal.Acc.=balanced accuracy.

MAE (tr/te)	Acc. (tr/te)	Bal. Acc. (tr/te)	Macro-F1 (tr/te)
0.11 / 0.17	0.96 / 0.83	0.75 / 0.50	0.71 / 0.45

8.4 Experiment 3

This appendix provides the technical details of Experiment 3, the AI deployment and governance-context comparison introduced in Section 5.3.

Simulation design. Experiment 3 simulates two copies of the same clinical AI system over 96 sparse monthly-like observations each. Hospital Alpha follows a slow-degradation trajectory with no nonzero event effects. Hospital Beta follows the common lifecycle simulation logic used in the previous experiments, with gradual drift, process-level noise, observation-level noise, adverse shocks, and corrective updates. The key contrast is therefore between a stable controlled deployment environment and a brittle environment in which signed lifecycle events move the system across multiple trustworthiness regimes. Figure 7 displays boundary margin and profile drift over time in Hospital Beta.

The measurement convention is aligned with Experiment 2. Performance, calibration, robustness, explainability, and oversight quality are normalized so that higher values indicate better quality. Fairness is represented by an Equal Opportunity Gap, so lower values indicate better fairness. Thus, the fairness coordinate is not an inverted fairness score: increasing q_{fair} represents a widening fairness gap. Table 11 summarizes the main design choices. For Hospital Alpha, we set `noise_std` = 0.004, so the process-level perturbation has standard deviation 0.002 and the observation-level perturbation has standard deviation 0.004. For Hospital Beta, we set `noise_std` = 0.01, so the corresponding standard deviations are 0.005 and 0.01, respectively.

Table 11: Experiment 3: simulation design summary. Both deployed copies are simulated over 96 sparse monthly-like observations. Hospital Alpha follows a slow-degradation trajectory with no nonzero event effects and is evaluated under a coarse local expert rule. Hospital Beta follows the common lifecycle simulation logic with signed lifecycle effects at time steps (10, 18, 26, 34, 42, 50, 60, 72). For higher-is-better dimensions, positive effects are corrective updates and negative effects are adverse shocks; for the fairness-gap coordinate, positive effects widen the gap and negative effects shrink it.

Dimension	Raw metric	Baseline b	Alpha drift d	Beta drift d	Hospital Beta signed lifecycle effects
Performance	AUROC	0.93	0.035	0.12	(0.01, -0.20, 0.10, -0.25, 0.30, -0.15, 0.25, -0.10)
Calibration	Calibration Score	0.91	0.030	0.17	(0.01, -0.10, 0.05, -0.10, 0.25, -0.05, 0.15, -0.25)
Robustness	Stress-Test AUROC Retention	0.92	0.025	0.12	(0.01, -0.10, 0.10, -0.15, 0.25, -0.05, 0.05, -0.15)
Fairness	Equal Opportunity Gap	0.11	-0.035	-0.16	(-0.01, 0.05, -0.05, 0.05, -0.10, 0.15, -0.25, 0.15)
Explainability	Reason-Code Stability	0.87	0.040	0.20	(0.01, -0.20, 0.10, -0.15, 0.30, -0.10, 0.20, -0.25)
Oversight quality	Oversight Quality	0.88	0.035	0.19	(0.02, -0.15, 0.15, -0.20, 0.25, -0.15, 0.25, -0.15)

For Hospital Beta, the negative drift coefficient for the fairness coordinate means that the Equal Opportunity Gap tends to increase over time under the common simulation equation $q(t) = \text{clip}_{[0,1]}(b - d\tau_t + \dots)$. Similarly, negative signed event effects on the fairness coordinate shrink the gap and therefore improve fairness, whereas positive signed event effects widen the gap and therefore worsen fairness.

Expert-based trustworthiness-level rules. Hospital Alpha and Hospital Beta use the same ordered four-level governance scale, but differ in the granularity of their expert-defined trustworthiness-level rules. Hospital Alpha uses a coarse local rule under which all observed profiles remain in level 4. This reflects a stable deployment environment in which slow degradation does not challenge the locally accepted highest trustworthiness regime. Hospital Beta uses a more refined rule, reported in Table 12, which adapts the threshold conditions to the six-dimensional clinical trustworthiness protocol. The Beta rule deliberately combines hard failure conditions on core dimensions with stricter conjunctive conditions for the upper trustworthiness regimes. This makes the learning problem more demanding than in

the previous experiments while preserving the assumption that governance-relevant trustworthiness judgments can be represented through auditable threshold conditions.

Table 12: Experiment 3: refined expert-based trustworthiness-level rule for Hospital Beta. All dimensions except fairness are normalized so that higher values indicate better quality. Fairness is measured as an Equal Opportunity Gap, so lower values indicate better fairness.

Level	Rule	Interpretation
1	$q_{\text{perf}} < 0.58$ or $q_{\text{cal}} < 0.56$ or $q_{\text{rob}} < 0.56$ or $q_{\text{fair}} > 0.50$	risky
4	$q_{\text{perf}} \geq 0.85$, $q_{\text{cal}} \geq 0.85$, $q_{\text{rob}} \geq 0.85$, $q_{\text{fair}} \leq 0.15$, $q_{\text{exp}} \geq 0.60$, and $q_{\text{over}} \geq 0.75$	excellent
3	$q_{\text{perf}} \geq 0.65$, $q_{\text{cal}} \geq 0.65$, $q_{\text{rob}} \geq 0.75$, $q_{\text{fair}} \leq 0.25$, $q_{\text{exp}} \geq 0.45$, and $q_{\text{over}} \geq 0.55$, provided that Level 4 does not apply	good
2	otherwise	acceptable

Hospital Alpha uses a coarser local rule: profiles are assigned level 4 unless a broad hard-stop condition is violated, namely $q_{\text{perf}} < 0.70$, $q_{\text{cal}} < 0.68$, $q_{\text{rob}} < 0.70$, $q_{\text{fair}} > 0.40$, $q_{\text{exp}} < 0.55$, or $q_{\text{over}} < 0.55$, in which case they are assigned level 3. In the simulated Alpha trajectory, none of these hard-stop conditions is reached. Hospital Beta occupies all four regimes, with 31 observations at level 4, 37 at level 3, 20 at level 2, and 8 at level 1.

Learning results. The learning strategy is conditional on the observed lifecycle variation. For Hospital Alpha, no empirical trustworthiness-level rule is learned. Since all observed profiles remain in level 4 under the local coarse rule, a supervised classifier trained on these data would collapse to the constant rule $\hat{T}(\mathbf{Q}(t)) = 4$. This is not a meaningful refinement of the expert-defined governance standard. Rather, it shows that in a stable controlled environment, the expert rule remains the appropriate operative rule and the ML exercise adds no governance-relevant information.

For Hospital Beta, an empirical trustworthiness-level rule is learned from the regime-varying lifecycle data. As in the previous experiments, the data are split chronologically into a training segment containing the first 80% of observations and a test segment containing the remaining 20%. Temporal cross-validation selects a shallow decision tree with 3 leaves (`gini`, `max_depth=2`, `max_leaf_nodes=3`, `ccp_alpha = 0`). Table 13 reports the predictive performance of the resulting rule.

Table 13: Experiment 3: predictive performance of the operative or learned trustworthiness-level rule. Hospital Alpha uses the expert-based level-4 assignment because its local rule and trajectory contain no observed regime variation. Hospital Beta uses a learned decision tree selected by temporal cross-validation. Abbreviations: tr=train, te=test, Acc.=accuracy, Bal.Acc.=balanced accuracy.

Scope	MAE	Acc.	Bal. Acc.	Macro-F1
Hospital Alpha operative rule (by construction)	N/A	N/A	N/A	N/A
Hospital Beta (tr)	0.11	0.89	0.75	0.68
Hospital Beta (te)	0.50	0.50	0.50	0.33

In this revised simulation, the learned Hospital Beta rule does not reproduce the refined expert-defined labels perfectly. Instead, it compresses the four-level expert scale into a coarser operational rule. The reference Hospital Beta labels are distributed across all four levels, whereas the learned rule predicts only levels 2, 3, and 4, with predicted counts 18, 47, and 31, respectively. This means that the rare level 1 regime is not recovered by the selected shallow tree. The deterioration on the held-out segment further shows that a compact learned rule may fail to preserve rare but important low-trustworthiness states when the lifecycle trajectory is unstable and the most severe regime is weakly represented.

The diagnostics reinforce the contrast between the two copies. Hospital Alpha has small but nonzero profile drift, with mean drift 0.0058 and maximum drift 0.0103, reflecting very slow degradation within the locally accepted highest

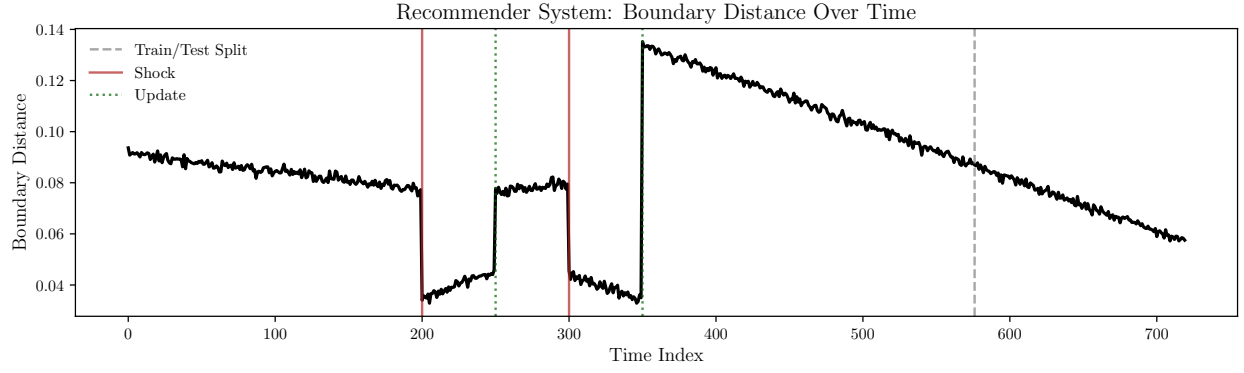
regime. Boundary margins under a learned tree are not reported for Hospital Alpha because no learned rule is used. Hospital Beta has a mean boundary margin of 0.0500, with a minimum boundary margin of 0.0000, showing that some profiles lie directly on learned regime boundaries. Hospital Beta also has substantially larger profile drift, with mean drift 0.0261 and maximum drift 0.2402. These values reflect the signed lifecycle events and resulting movement of the brittle deployment through trustworthiness space, especially in the later held-out segment where trustworthiness levels become less stable.

References

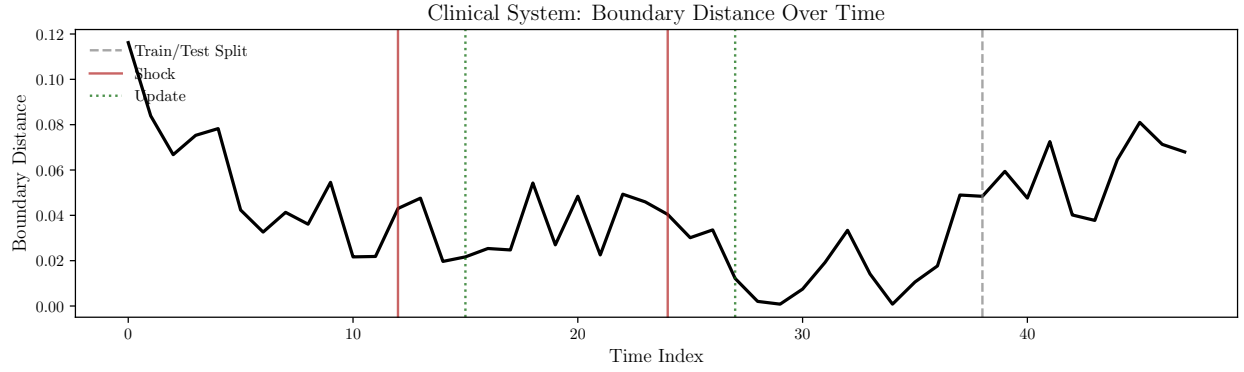
- Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sam Mourad, Paolo Pedemonte, Ramya Raghavendra, John T. Richards, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques. In *Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence*, 2019.
- Firas Bayram, Bestoun S Ahmed, and Erik Hallin. Adaptive data quality scoring operations framework using drift-aware mechanism for industrial applications. *Journal of Systems and Software*, 217:112184, 2024.
- Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Praveen Lohia, Jacqueline Martino, Sameep Mehta, Aleksandra Mojsilović, et al. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. In *IEEE International Conference on Data Mining Workshops*, pages 152–160, 2019.
- Mélanie Bernhardt, Daniel C Castro, Ryutaro Tanno, Anton Schwaighofer, Kerem C Tezcan, Miguel Monteiro, Shruthi Bannur, Matthew P Lungren, Aditya Nori, Ben Glocker, et al. Active label cleaning for improved dataset quality under resource constraints. *Nature Communications*, 13(1):1161, 2022.
- Sarah Bird, Miroslav Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in AI. *Microsoft Research White Paper*, 2020.
- Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Chapman and Hall/CRC, 2017.
- EU AI Act. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence. Technical report, European Union, June 2024. URL <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>.
- EU AI HLEG. Ethics Guidelines for Trustworthy AI. Technical report, European Commission, April 2019. URL <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>.
- EU HLEG AI. Assessment list for trustworthy artificial intelligence (ALTAI). Technical report, European Commission, 2020. Self-assessment tool for implementing the Ethics Guidelines for Trustworthy AI.
- Andrea Ferrario. Justifying our credences in the trustworthiness of AI systems: A reliabilistic approach. *Science and Engineering Ethics*, 30(6):55, 2024.

- Andrea Ferrario. A trustworthiness-based metaphysics of artificial intelligence systems. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 1360–1370, 2025.
- Andrea Ferrario. High-risk AI systems and the problem of identity in the European AI Act, 2026. Accepted as a non-archival paper at *The 2026 ACM Conference on Fairness, Accountability, and Transparency (FAccT '26)*, Montreal, QC, Canada, June 25–28, 2026.
- Luciano Floridi, Matthias Holweg, Mariarosaria Taddeo, Javier Amaya, Jakob Mökander, and Yuni Wen. CapAI-A procedure for conducting conformity assessment of AI systems in line with the EU Artificial Intelligence Act. Available at SSRN 4064091, 2022.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.
- ISO/IEC. Information technology — artificial intelligence — overview of trustworthiness in artificial intelligence. Technical Report ISO/IEC TR 24028:2020, International Organization for Standardization and International Electrotechnical Commission, 2020. URL <https://www.iso.org/standard/77608.html>.
- Stanley Kaplan and B John Garrick. On the quantitative definition of risk. *Risk analysis*, 1(1):11–27, 1981.
- Davinder Kaur, Suleyman Uslu, Kaley J Rittichier, and Arjan Durrezi. Trustworthy artificial intelligence: A review. *ACM Computing Surveys*, 55(2):1–38, 2022.
- Nils Kemmerzell, Annika Schreiner, Haroon Khalid, Michael Schalk, and Letizia Bordoli. Towards a better understanding of evaluating trustworthiness in AI systems. *ACM Computing Surveys*, 57(9):1–38, 2025.
- Dominik Kreuzberger, Niklas Köhl, and Sebastian Hirschl. Machine learning operations (MLOps): Overview, definition, and architecture. *IEEE Access*, 11:31866–31879, 2023.
- Todd Kulesza, Denis Charles, Rich Caruana, Saleema Amin Amershi, and Danyel Aharon Fisher. Structured labeling to facilitate concept evolution in machine learning, June 11 2019. US Patent 10,318,572.
- Johann Laux, Sandra Wachter, and Brent Mittelstadt. Trustworthy artificial intelligence and the European Union AI Act: On the conflation of trustworthiness and acceptability of risk. *Regulation & Governance*, 18(1):3–32, 2024.
- Q Vera Liao and S Shyam Sundar. Designing for responsible trust in AI systems: A communication perspective. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1257–1268, 2022.
- Michele Loi, Christoph Heitz, Andrea Ferrario, Anita Schmid, and Markus Christen. Towards an ethical code for data-based business. In *2019 6th Swiss Conference on Data Science (SDS)*, pages 6–12. IEEE, 2019.
- Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2018.
- Siddharth Mehrotra, Chadha Degachi, Oleksandra Vereschak, Catholijn M Jonker, and Myrthe L Tielman. A systematic review on fostering appropriate trust in human-AI interaction: Trends, opportunities and challenges. *ACM Journal on Responsible Computing*, 1(4):1–45, 2024.
- Siddharth Mehrotra, Jin Huang, Xuelong Fu, Roel Dobbe, Clara I Sánchez, and Maarten de Rijke. Understanding AI trustworthiness: A scoping review of AIES & FAccT articles. *Journal of Artificial Intelligence Research*, 85, 2026.

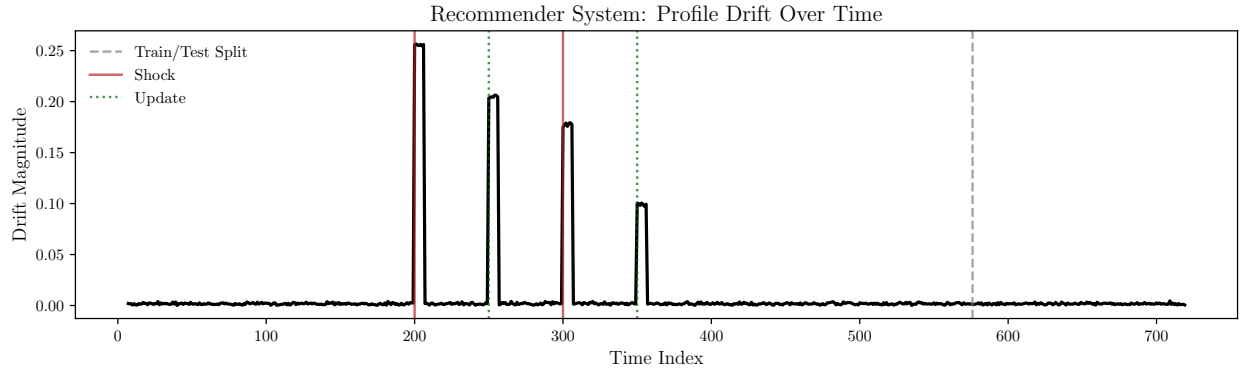
- Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229, 2019.
- Jakob Mökander and Luciano Floridi. Ethics-based auditing to develop trustworthy AI. *Minds and Machines*, 31(2): 323–327, 2021.
- Jakob Mökander, Maria Axente, Federico Casolari, and Luciano Floridi. Conformity assessments and post-market monitoring: A guide to the role of auditing in the proposed European AI regulation. *Minds and Machines*, 32(2): 241–268, 2022.
- National Institute of Standards and Technology. Artificial intelligence risk management framework (AI RMF 1.0). Technical Report AI 100-1, NIST, 2023.
- OECD. Recommendation of the council on artificial intelligence: Oecd/legal/0449. <https://legalinstruments.oecd.org/en/instruments/oecd-legal-0449>, 2022.
- Charles Chimwemwe Phiri. Creating characteristically auditable agentic AI systems. In *Proceedings of the Intelligent Robotics FAIR 2025*, pages 1–14. 2025.
- Stephan Rabanser, Sayash Kapoor, Peter Kirgis, Kangheng Liu, Saiteja Utpala, and Arvind Narayanan. Towards a science of AI agent reliability. *arXiv preprint arXiv:2602.16666*, 2026.
- Catherine Radclyffe et al. The assessment list for trustworthy artificial intelligence: A review and recommendations. *Frontiers in Artificial Intelligence*, 6:1020592, 2023.
- Leon Staufer, Kevin Feng, Kevin Wei, Luke Bailey, Yawen Duan, Mick Yang, A. Pinar Ozisik, Stephen Casper, and Noam Kolt. The 2025 AI agent index: Documenting technical and safety features of deployed agentic AI systems. *arXiv preprint arXiv:2602.17753*, 2026.
- The White House. Blueprint for an AI bill of rights: Making automated systems work for the american people. <https://www.whitehouse.gov/ostp/ai-bill-of-rights/>, 2022.
- United Nations Educational, Scientific and Cultural Organization. Recommendation on the ethics of artificial intelligence. <https://unesdoc.unesco.org/ark:/48223/pf0000381137.locale=en>, 2022.
- Roberto V Zicari, John Brodersen, James Brusseau, Boris Düdler, Timo Eichhorn, Todor Ivanov, Georgios Kararigas, Pedro Kringen, Melissa McCullough, Florian Möselein, et al. Z-inspection®: A process to assess trustworthy AI. *IEEE Transactions on Technology and Society*, 2(2):83–97, 2021.



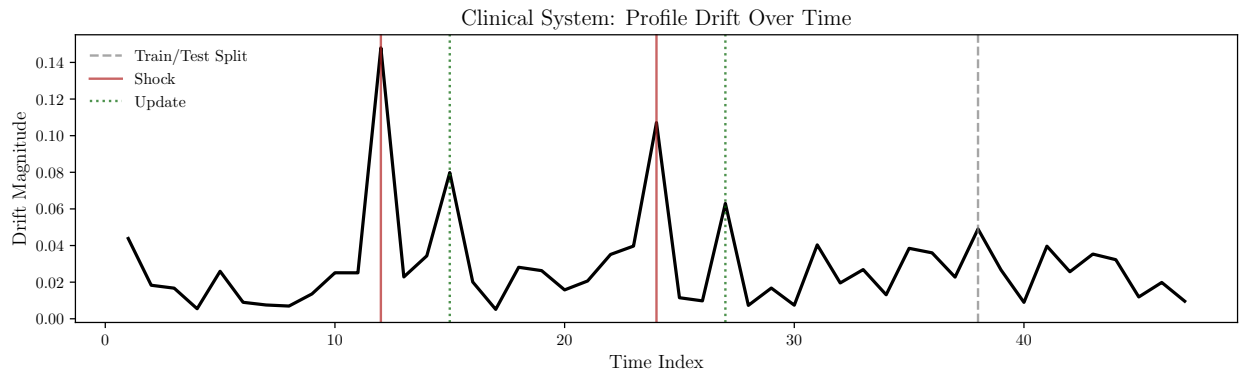
(a) Recommender system: Boundary margin over time



(b) Clinical decision-support system: Boundary margin over time

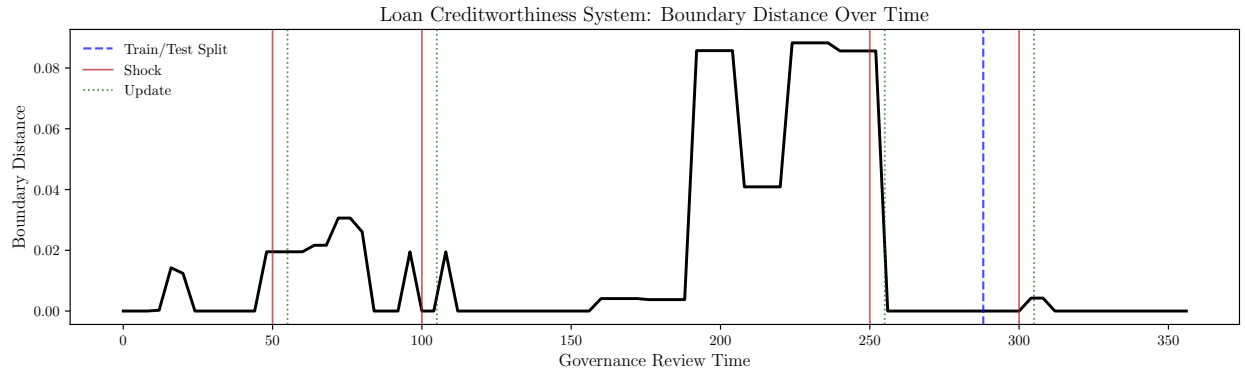


(c) Recommender system: Profile drift over time

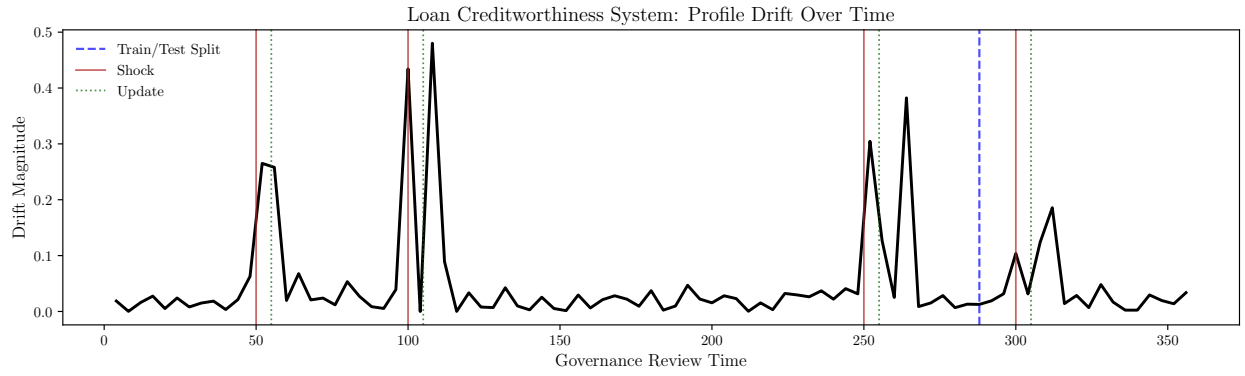


(d) Clinical decision-support system: Profile drift over time

Figure 5: Experiment 1. Reporting boundary margin and profile drift over time.

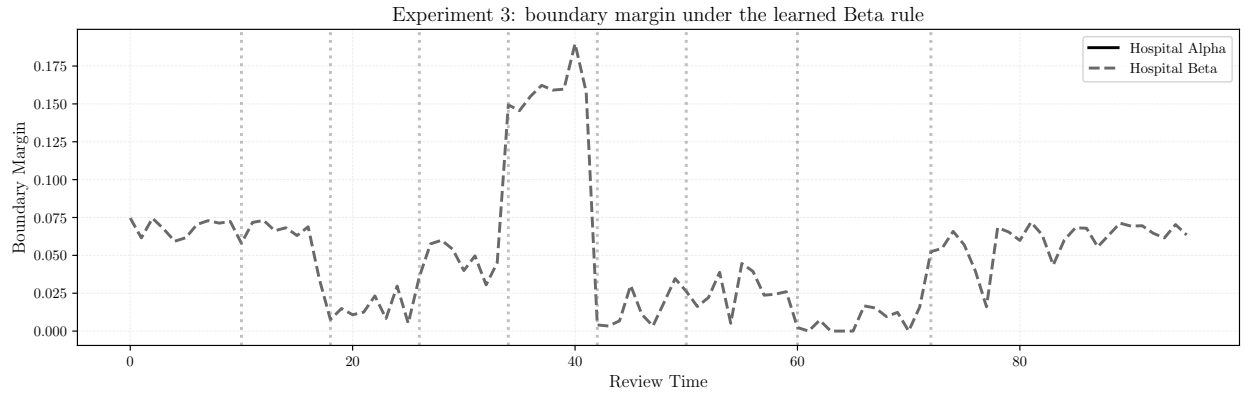


(a) Boundary margin over time

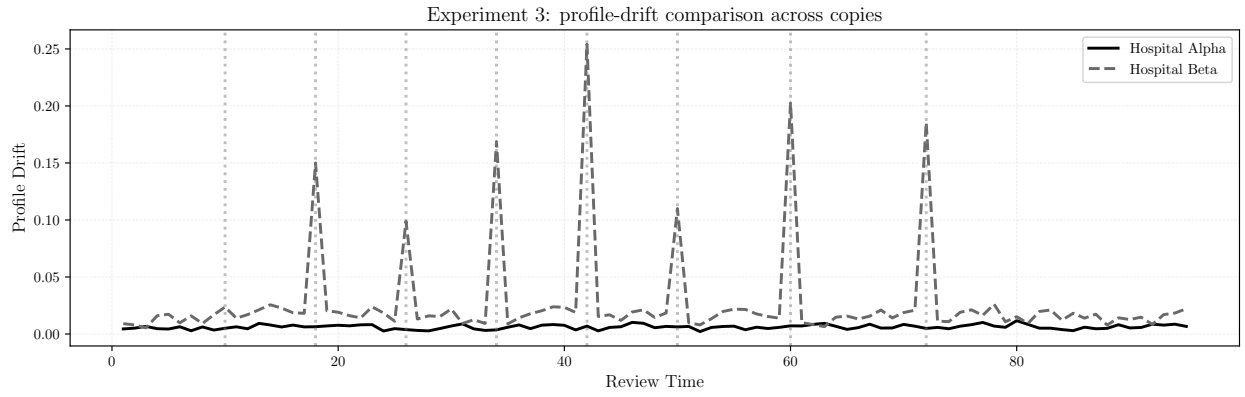


(b) Profile drift over time

Figure 6: Experiment 2. Reporting boundary margin and profile drift over time.



(a) Hospital Beta: Boundary margin over time



(b) Hospital Beta: Profile drift over time

Figure 7: Experiment 3. Reporting boundary margin and profile drift over time for Hospital Beta.