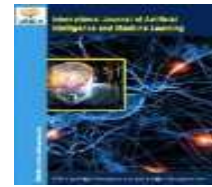




SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Confidence-Gated Model Invocation: A Decision-Quality Framework for Governing Machine Learning Outputs Within Enterprise Orchestration Pipelines

Dillibabu Arumugam

Independent Researcher.

Abstract

A model's internal confidence score is not, by itself, a decision about what an orchestration pipeline should do with the output that score accompanies. It is a signal. A pipeline that treats every signal identically has made an implicit governance decision without ever examining it.

This article develops a framework for treating the handling of that signal as an explicit architectural control at the orchestration layer, in place of the unexamined default in which any output crossing the pipeline boundary is executed as though it were certain. The research objective is to determine why production incidents attributed to model error are frequently better explained by an absent or poorly designed decision boundary between a model's output and the pipeline's action on it.

The research design synthesises calibration research, human-in-the-loop decision literature, and orchestration resilience patterns, read from the standpoint of an integration architect accountable for pipelines where automated decisions carry contractual or customer-facing consequences. The method derives three governance mechanisms - confidence-threshold routing, override logging with an enforced audit trail, and escalation-path design - filtered specifically to the orchestration-enforcement layer rather than the model-internal layer.

A controlled experiment reported by Sele and Chugunova supplies the framework's governing caution. Participants preferred an algorithmic recommender over an equally accurate human 66 per cent of the time, and that preference rose by a further seven percentage points once human-in-the-loop review became available - yet decisions made under the reviewed condition were significantly less accurate than those made under full delegation [5]. A review step must therefore be governed rather than assumed beneficial by default.

The outcome is a decision-quality framework in which every model output crossing the orchestration boundary is routed, logged, or escalated according to an explicit policy. The conclusion is that this boundary, not the underlying model's own accuracy, determines whether machine-learning-augmented automation is safe to deploy at production scale.

Keywords: machine learning governance; enterprise orchestration; confidence-gated automation; decision-quality framework; human-in-the-loop escalation; audit trail; real-time decisioning.

1. Introduction

A fraud-scoring model flags a transaction with a confidence score low enough that a careful reviewer would want a second opinion. The orchestration pipeline approves the transaction anyway. Nothing in its design distinguishes a confident output from an uncertain one once both cross the same boundary, so both are treated identically - and identically means autonomously.

The transaction is disputed days later. The incident review concludes that the model made a mistake, but the model's own evaluation metrics show nothing wrong. It reported its uncertainty accurately, exactly as a calibrated model is supposed to. What failed was a decision the pipeline never made explicitly: what to do with a low-confidence output once it arrives. That omission is invisible to any review process that asks only whether the model performed well.

Classical tail-latency research offers a useful analogy for why this matters more as pipelines grow more complex. A system composed of several individually well-behaved stages produces aggregate behaviour substantially worse than any single stage's number suggests, because the probability that at least one stage misbehaves rises with every stage added [6]. A confidence-gating decision is exactly this kind of stage. Skipping it does not make the underlying variability disappear; it makes the variability invisible until it surfaces as an incident.

This article treats that unmade decision as the central architectural problem in embedding machine learning within real-time enterprise orchestration. Model outputs vary in reliability by design, not by defect [1], [3]. Yet

most orchestration pipelines are built to consume every output identically regardless of the reliability signal attached to it. That gap - between a model's honestly reported uncertainty and a pipeline's blind consumption of its output - is where this article's contribution sits.

Three mechanisms compose the framework. Confidence-threshold routing determines whether an output proceeds autonomously, is degraded to a conservative fallback, or is escalated. Override logging creates an enforceable audit trail whenever a human or downstream system overrides the pipeline's default handling of an output. Escalation-path design specifies where an escalated decision goes and under what service-level expectation it must be resolved. None of the three lives inside the model. All three live at the orchestration boundary, which is precisely why they have been comparatively neglected: model teams optimise the model, and orchestration teams have historically treated a model's output as a black box to consume rather than a signal requiring its own governance.

Contributions. This article makes four contributions. First, it reframes a recognised class of production incident - automation acting on an unreliable output - as an orchestration-layer design omission rather than a model-quality defect. Second, it specifies a confidence-tier decision matrix that makes tier assignment an auditable, pre-execution decision. Third, it derives three governance mechanisms, each tied to a distinct failure point and each paired with a monitoring signal. Fourth, it identifies the conditions under which each mechanism fails silently, including the threshold-drift and reviewer-capability dependencies that are the framework's principal operational risks.

Section 2 reviews calibration research, human-in-the-loop decision literature, and orchestration resilience patterns as converging concerns. Section 3 sets out the derivation method. Section 4 presents the framework. Section 5 addresses boundary conditions and limitations. Section 6 concludes.

2. Literature Review

2.1 Calibration improves the signal but not its handling

Calibration research establishes with considerable rigour that a model's stated confidence and its actual reliability frequently diverge as architectures grow more capable [2]. Foundational work found that modern neural networks are measurably worse calibrated than earlier, simpler architectures despite improved raw accuracy, and traced this to network depth, width, weight decay, and the normalisation techniques used during training [1]. Survey work across a broad range of model families confirms the divergence as a structural property rather than an artefact isolated to one architecture [2].

What calibration research does not address - because it sits one layer below where this article operates - is what a downstream consumer should do once it has an output and whatever confidence signal accompanies it. Calibration improves the signal. It says nothing about what happens next, and what happens next is the orchestration-layer question posed here.

2.2 Human review is not automatically protective

Human-in-the-loop literature supplies a finding that orchestration architects designing escalation paths should treat as a governing constraint. In a controlled online experiment ($N = 292$), participants preferred an algorithmic recommender over an equally accurate human recommender 66 per cent of the time, and that preference rose by a further seven percentage points when participants could monitor and adjust the recommendation. Decisions made in the human-in-the-loop condition were nonetheless significantly less accurate than those made under full delegation [5]. The mechanism the authors identify is instructive: human monitors were less likely to adjust recommendations containing larger errors, and when they did adjust them, corrected them less.

This inverts the intuition motivating most human-in-the-loop designs. It does not argue against escalation. It argues that escalation paths require explicit attention to what a reviewer actually does with an escalated decision, rather than an assumption that inserting a person into the loop is inherently protective.

Explainable and trust-oriented decision-support literature addresses a related question - why a decision-maker trusts or distrusts an automated output - and finds the tradeoff between accuracy and explainability persistent rather than awaiting a clean technical fix [3]. A recent review catalogues the visual, rule-based, case-based, and natural-language techniques organisations have applied to narrow this gap, while noting that highly accurate models still tend to sacrifice interpretability relative to simpler alternatives [4]. A pipeline that expects a human to trust conditionally, escalating only when trust is warranted, inherits this unresolved tension whether or not its designers acknowledge it.

2.3 Resilience patterns address the wrong failure mode

Orchestration resilience literature is well developed for infrastructure failures - network partitions, downstream unavailability, timeout cascades - and comparatively underdeveloped for the failure mode this article addresses: a downstream service that responds successfully and quickly, but whose response should not be trusted at face value. Circuit breaker and retry literature describes mechanisms for handling a service that fails to respond [7], [8]. It says little about a model-serving endpoint that responds every time, but with variable and sometimes low reliability.

Admission control research at the latency boundary offers a partial structural analogy. Work on response-time-objective-aware admission demonstrates that a system can make principled, threshold-based decisions about what to accept under load instead of relying on ad hoc rules, reporting 15 to 30 per cent fewer overall rejections than capacity-centric baseline policies at a mean admission-control decision overhead of 18 microseconds for millisecond-scale queries [9]. The confidence-threshold routing mechanism proposed here borrows that structure and applies it to output reliability instead of latency.

Recent work on production-grade agentic workflows reaches a parallel conclusion from a different direction: systems chaining multiple automated decisions require explicit, engineered checkpoints rather than end-to-end automation with no governed intervention point [13]. Regulatory guidance converges on the same position. Article 14 of Regulation (EU) 2024/1689 requires that high-risk AI systems be designed so that natural persons can effectively oversee them, including the ability to disregard, override, or reverse the output [15]. Risk-management guidance treats continuous reassessment as a structural component rather than a post-deployment addition [16]. Neither body of guidance is written for the orchestration architect specifically. Both arrive independently at the position this article formalises for that architect.

3. Method

This article presents a structured framework derivation, following the design-science tradition in which the artefact produced is the contribution and its justification rests on systematic derivation from prior evidence plus explicit statement of applicability conditions. The standpoint is that of a practising integration architect accountable for pipelines in which automated decisions carry direct consequences for customers and counterparties. It is not a controlled experiment comparing governed against ungoverned pipelines under identical conditions.

The derivation proceeded in three stages.

First, candidate governance mechanisms were drawn from the calibration, human-in-the-loop, and orchestration-resilience literature reviewed in Section 2, spanning threshold-based admission, escalation design, and resilience-pattern precedent, including recovery patterns such as bulkheads and adaptive backpressure documented in recent systematic reviews [14].

Second, each candidate was filtered against a scope requirement: operation at the orchestration-enforcement layer, governing what happens to an output after it is produced, rather than at the model-internal layer, improving the output's own accuracy. This filter excluded calibration-improvement techniques - temperature scaling, ensemble methods, recalibration curves - from the framework itself. Not because they are unimportant, but because a well-calibrated model still needs an orchestration-layer policy for what to do with its output, and a poorly calibrated model does not become safe to consume merely because the orchestration layer declines to notice.

Third, the surviving mechanisms were organised into the three-part framework of Section 4, each tied explicitly to the evidence motivating its inclusion.

Scope and evidence boundaries. This article does not evaluate specific calibration techniques, model architectures, or explainability methods in depth, since those operate on the model side of the boundary addressed here. The claim is that orchestration-layer governance matters independently of model-side accuracy improvement, and in the incident scenarios motivating this framework often matters more. A second boundary concerns the evidence base: the three mechanisms were derived against fraud-detection and payment-authorisation pipelines, and their transfer to domains with different consequence structures - a recommendation engine, where a wrong output degrades experience rather than causing financial harm - has not been tested here.

4. The Framework

4.1 A confidence-tier decision matrix

The framework requires every model output crossing the orchestration boundary to be classified into one of a small number of confidence tiers, each mapped explicitly to a pipeline action. This is the framework's foundational move. It refuses the default in which every output proceeds identically regardless of its reliability signal, and makes tier assignment itself an auditable, first-class decision.

Table 1: Confidence-tier decision matrix

Confidence tier	Example trigger condition	Pipeline action	Audit requirement
High	Score above upper threshold, consistent with historical reliability for this decision type	Autonomous execution	Standard logging
Moderate	Score between thresholds, or model flags input as atypical	Degrade to conservative fallback or lower-fidelity path	Fallback invocation logged
Low	Score below lower threshold, or confidence signal absent or unreliable	Escalate to human or secondary system review	Full override and escalation audit trail
Anomalous	Output inconsistent with input distribution or business rules	Reject and flag for investigation	Full audit trail; incident review triggered

Threshold placement deserves specific attention. Thresholds set against a model's validation-set confidence distribution routinely drift out of calibration within weeks of production deployment, because the production input distribution is never quite the validation distribution. The discrepancy surfaces first as a rising anomalous-tier rate, long before it appears in any accuracy metric the model team is watching, as illustrated in Figure 1. The gap between validation and production behaviour is itself well documented: one anti-fraud model reported a validation-sample recall of 0.506 alongside a false-positive rate in the 0.15 to 0.20 range on held-out test data, figures that illustrate how much headroom a static threshold has to be wrong about [10]. Treating threshold values as a fixed constant rather than a monitored, periodically recalibrated parameter is one of the more common ways this framework fails silently.

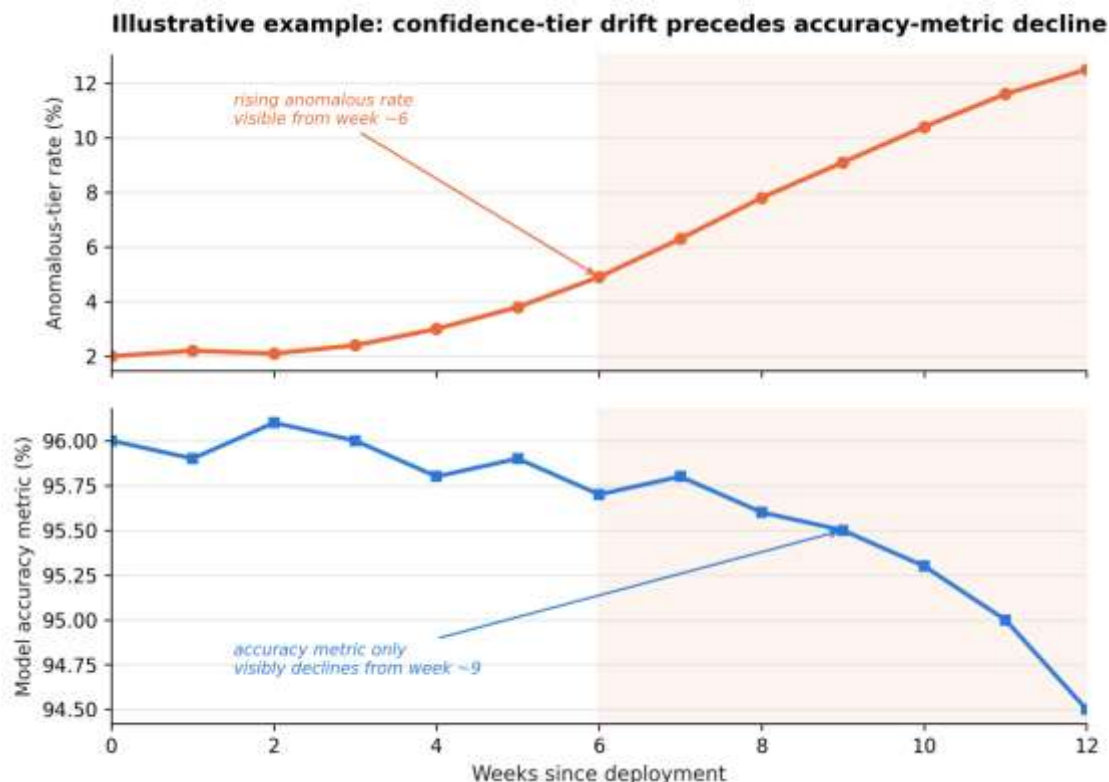


Figure 1: Confidence-tier distribution drifting over time. The anomalous-tier rate becomes visible from roughly week six; the accuracy metric a model team is more likely to be watching does not visibly move until roughly week nine. Illustrative example.

4.2 Three governance mechanisms

Confidence-threshold routing operationalises the matrix at request time. It evaluates a model output's confidence tier and routes accordingly, instead of treating threshold placement as a one-time design decision made without reference to the pipeline's actual production distribution of confidence scores. The admission-control precedent is instructive: objective-aware threshold policies achieved 15 to 30 per cent fewer rejections than capacity-centric baselines at a mean decision overhead of 18 microseconds, evidence that principled threshold-based routing can outperform uniform handling of heterogeneous inputs at negligible operational cost [9].

Override logging creates an enforced record whenever a human reviewer, or a downstream automated system, overrides the pipeline's default handling of an output - whether that override accepts a low-confidence output routed for review, or rejects a high-confidence output a reviewer distrusts for reasons outside the model's evidence. This responds directly to the finding that human oversight does not automatically improve decision quality [5]. Without logged, aggregated overrides, an organisation cannot detect a pattern in which its reviewers are systematically degrading otherwise-reliable automated decisions, or conversely rubber-stamping outputs a well-designed threshold flagged for good reason.

Escalation-path design specifies where an escalated decision goes, under what service-level expectation, and with what authority to act. This closes a gap the human-in-the-loop literature identifies but does not resolve [5], and it operationalises the effective-oversight requirement that Article 14 places on high-risk systems [15]. An escalation path with no defined service-level expectation degrades, quietly and almost immediately, into a queue nobody is accountable for clearing - functionally equivalent to no escalation path at all, but one that looks governed on an architecture diagram. Any retried or re-escalated inference call within this path must preserve idempotent processing semantics, so that a governance-driven retry does not silently duplicate a decision already acted upon [17].

Table 2: Governance mechanisms mapped to trigger conditions, audit requirements, and monitoring signals

Mechanism	Trigger condition	Audit requirement	Monitoring signal
Confidence-threshold routing	Output confidence tier evaluated at request time	Tier classification logged per decision	Distribution of decisions by confidence tier over time
Override logging	Human or downstream system overrides default handling	Full override record: original tier, override action, rationale	Override rate and direction (accept-despite-flag versus reject-despite-clear)
Escalation-path design	Output routed to moderate, low, or anomalous tier	Escalation SLA attainment logged	Escalation queue age and resolution rate

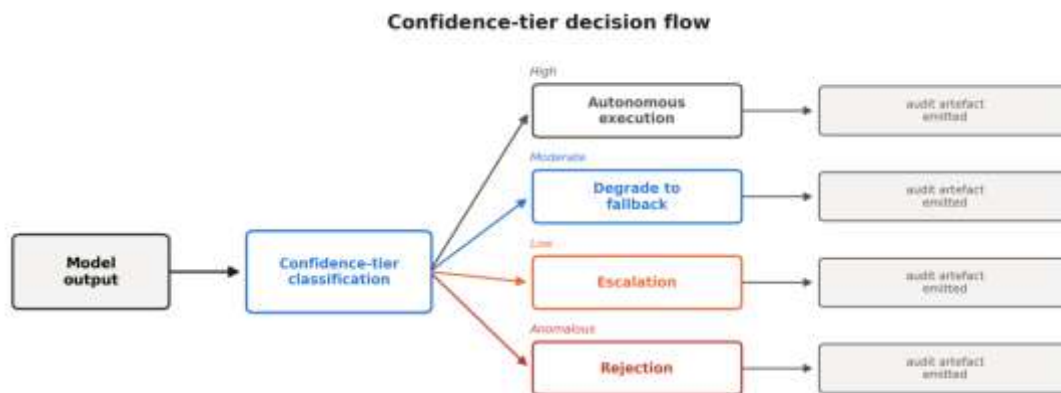


Figure 2: Confidence-tier decision flow. A model output entering the orchestration boundary is classified into a confidence tier and branches to autonomous execution, degrade-to-fallback, escalation, or rejection, with an audit artefact emitted on every branch.

5. Discussion and Limitations

The framework's value concentrates in pipelines where an automated decision carries direct financial, regulatory, or customer-facing consequence, and where the cost of occasionally escalating a decision that turns out to have been safe is materially lower than the cost of autonomously executing one that turns out to have been wrong. Tiered decisioning logic of this general kind has been deployed in production: one Chinese commercial bank reported that its share of provincially implicated debit cards fell from over 50 per cent to 20 per cent relative to three peer institutions in the seven months following deployment of a dynamic-quantification anti-fraud model [10]. That figure is a peer-relative share rather than an absolute fraud rate, and it should be read as an illustration of the kind of outcome tiered decisioning can contribute to, not as a transferable effect size.

Escalation is not a safety guarantee. This is the framework's most important limitation, and it follows directly from the evidence in Section 2.2 [5]. An escalation path that routes low-confidence outputs to reviewers who are not equipped, incentivised, or given sufficient context to evaluate them can produce worse outcomes than no escalation path at all. The three mechanisms govern where a decision goes. They do not, by themselves, guarantee that wherever it goes is equipped to handle it well. An architecture diagram showing an escalation path is not evidence that the path functions.

Escalation design misses what has no trigger. Work on escalation criteria for AI incident response identifies a structural weakness in event-driven escalation: where incidents are assessed individually, systemic harms that emerge only through accumulation risk going under-detected [12]. A confidence-gating architecture inherits this weakness directly. Every mechanism proposed here fires on a discrete output, so a slow degradation visible only across thousands of individually acceptable decisions will not trigger any of them. Aggregate monitoring of tier distribution over time is the partial answer, which is why it appears as a monitoring signal in Table 2 rather than as an optional extra.

Instrumentation is the binding constraint. The monitoring signals this framework requires - confidence-tier distribution, override rate and direction, escalation SLA attainment - demand observability infrastructure many production machine learning pipelines do not carry. Focus-group research across banking, financial technology, industrial technology, and e-commerce organisations finds that ML observability practice concentrates on data quality and infrastructure monitoring rather than decision-level attribution, and that coverage gaps are most pronounced at the organisational boundaries between the teams that build models and the teams that operate the pipelines consuming their outputs [11]. Retrofitting this instrumentation is a nontrivial undertaking, and organisations should expect the instrumentation effort to precede, not follow, any measurable governance benefit.

Override logging is diagnostic, not preventive. Logging an override after the fact does not prevent a bad override; it makes the pattern visible in aggregate, typically well after any individual override has produced its consequence. Organisations expecting real-time prevention from this mechanism will be disappointed. Its value is corrective over time. Realising that value depends on resilience patterns already well established in distributed systems practice - bulkheading a runaway escalation queue from otherwise-healthy pipeline capacity, and applying backpressure once queue depth exceeds a defined threshold [14], [7].

Evidential limitation. This article reports no experiment of its own. The framework is derived from published evidence and architectural practice, and every quantitative result cited comes from a study of some other system. The claim made here is that the mechanisms are directionally supported by that evidence, not that any specific effect size transfers to another deployment.

Future work. The clearest next step is empirical. Instrument a pipeline consuming a calibrated classifier, replay a labelled decision stream through it twice - once with uniform handling and once with tier-based routing - and report decision accuracy, escalation volume, and cost per avoided error at several threshold placements. That design also yields the threshold-drift measurement Section 4.1 asserts but does not demonstrate. A second direction is the reviewer-capability question: given the finding in [5], what properties of an escalation destination determine whether human review improves or degrades the decision?

6. Conclusion

The boundary between a model's output and the pipeline's action on that output is an architectural control surface, not an incidental byproduct of model quality. A model that reports its uncertainty accurately provides no guarantee that the pipeline consuming its output will act on that uncertainty appropriately, because that action is a separate design decision the model cannot make on the orchestration layer's behalf.

Enterprises that make confidence-threshold routing, override logging, and escalation-path design explicit, governed components of their orchestration layer convert an implicit and inconsistently applied assumption into an architectural property that can be measured, audited, and revised. Enterprises that continue treating

every model output as equally actionable will keep discovering, in production and after the fact, that the gap between a model's stated confidence and a pipeline's blind execution of its output is where consequential automation failures originate. That gap is not a modelling problem. It is an orchestration design decision, made by default whenever it is not made deliberately.

Declarations

Funding. This research received no external funding.

Conflicts of interest. The author declares no conflict of interest.

Data availability. No new data was generated. All evidence cited is available from the referenced sources.

Use of AI tools. Generative AI tools were used to assist with literature synthesis, drafting, and editing during the preparation of this manuscript. All analysis, interpretation, and conclusions are the author's own, and the final content was reviewed and approved by the author.

References

- [1] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in Proc. 34th Int. Conf. on Machine Learning (ICML), PMLR vol. 70, 2017, pp. 1321–1330.
- [2] J. Geng, F. Cai, Y. Wang, H. Koepl, P. Nakov, and I. Gurevych, "A survey of confidence estimation and calibration in large language models," in Proc. 2024 Conf. North American Chapter of the ACL: Human Language Technologies (NAACL), vol. 1, 2024, pp. 6577–6595.
- [3] A. Kovari, "AI for decision support: Balancing accuracy, transparency, and trust across sectors," *Information*, vol. 15, no. 11, art. 725, 2024. doi: 10.3390/info15110725
- [4] G. Kostopoulos, G. Davrazos, and S. Kotsiantis, "Explainable artificial intelligence-based decision support systems: A recent review," *Electronics*, vol. 13, no. 14, art. 2842, 2024. doi: 10.3390/electronics13142842
- [5] D. Sele and M. Chugunova, "Putting a human in the loop: Increasing uptake, but decreasing accuracy of automated decision-making," *PLOS ONE*, vol. 19, no. 2, art. e0298037, 2024. doi: 10.1371/journal.pone.0298037
- [6] J. Dean and L. A. Barroso, "The tail at scale," *Communications of the ACM*, vol. 56, no. 2, pp. 74–80, Feb. 2013. doi: 10.1145/2408776.2408794
- [7] M. T. Nygard, *Release It! Design and Deploy Production-Ready Software*, 2nd ed. Raleigh, NC, USA: Pragmatic Bookshelf, 2018.
- [8] M. R. Saleh Sedghpour, C. Klein, and J. Tordsson, "An empirical study of service mesh traffic management policies for microservices," in Proc. 2022 ACM/SPEC Int. Conf. on Performance Engineering (ICPE '22), 2022, pp. 17–27. doi: 10.1145/3489525.3511686
- [9] H. Xu and J. A. Colmenares, "Bouncer: Admission control with response time objectives for low-latency online data systems," in *Companion of the 2024 ACM SIGMOD Int. Conf. on Management of Data*, 2024, pp. 400–413. doi: 10.1145/3626246.3653384
- [10] F. Li and Z. Chen, "Dynamic quantification anti-fraud machine learning model for real-time transaction fraud detection in banking," *Discover Computing*, vol. 28, art. 59, 2025. doi: 10.1007/s10791-025-09549-7
- [11] J. Leest, I. Gerostathopoulos, P. Lago, and C. Raibulet, "Monitoring and observability of machine learning systems: Current practices and gaps," arXiv:2510.24142, 2025. (Preprint; not peer reviewed.)
- [12] F. Gomez, M. Ball, M. Harre, L. Preston, J. Schwab, and C. Machado, "Designing escalation criteria for international AI incident response: Criteria, triggers, and thresholds," in Proc. 2nd Workshop on Technical AI Governance Research (TAIGR) at ICML, 2026. arXiv:2604.23183
- [13] E. Bandara, R. Gore, P. Foytik, S. Shetty, R. Mukkamala, et al., "A practical guide for designing, developing, and deploying production-grade agentic AI workflows," arXiv:2512.08769, 2025. (Preprint; not peer reviewed.)
- [14] M. Mohammad, "Resilient microservices: A systematic review of recovery patterns, strategies, and evaluation frameworks," arXiv:2512.16959, 2025. (Preprint; not peer reviewed.)
- [15] European Parliament and Council of the European Union, Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), Article 14: Human oversight, Official Journal of the European Union, 2024.

- [16] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1. Gaithersburg, MD, USA: NIST, Jan. 2023. doi: 10.6028/NIST.AI.100-1
- [17] M. Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol, CA, USA: O'Reilly Media, 2017.