International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Flagged, Not Fixed: Human-in-the-Loop AI for Reconciliation Exceptions in the Financial Close

Srinivasarao Challagundla

Abstract

A reconciliation exception resolved silently by a model is not the same event as one flagged, reviewed, and documented by a controller, the two produce identical account balances but leave entirely different audit trails behind, and only one of them survives regulatory scrutiny. Evidence from finance and audit settings indicates that professionals do not extend algorithmic output the same benefit of the doubt they extend a human colleague: a single visible model error appears to suppress future reliance on that model disproportionately, and reviewers seem to demand more explanation from an AI-generated flag than from an equivalent human judgment. Multi-entity consolidation platforms already encode this caution structurally, through approval workflows, intercompany-match tolerance checks, and exception-routing rules that force a human decision point before a posting can finalize. Extending machine-learning-based anomaly detection into that architecture reads, on inspection, as a governance decision dressed in technical clothing rather than a pure technical upgrade, the value the AI layer adds lies in what it surfaces for review, not in what it quietly resolves. This article argues that reconciliation-exception handling should stay organized around verification workflows rather than automated correction, and it proposes a taxonomy of AI-flagging design elements alongside a comparative accountability framework, both intended to help close-cycle governance decisions keep pace with flagging technology that is maturing faster than the routing logic built around it.

Keywords: *AI-assisted reconciliation, close-cycle governance, exception routing, human-in-the-loop*

I. Introduction

Something is wrong with a reconciliation line, and a rules engine or a model has just said so. What happens next is the entire question this article is built around. Enterprise Performance Management platforms used for multi-entity consolidation, Oracle's Financial Consolidation and Close Cloud Service (FCCS), Account Reconciliation Cloud Service (ARCS), and related modules, have spent roughly a decade refining validation and control-check logic that routes unresolved items toward a human reviewer instead of letting them post unattended. Machine learning now promises to sharpen that detection layer further, catching anomalies a static rules engine would miss entirely. Whether AI can detect more exceptions is, at this point, barely in dispute. Whether the close-cycle architecture should let AI decide what happens to those exceptions is a separate question, and a much harder one.

Independent Researcher, USA

This article takes a position on that harder question: reconciliation-exception handling should remain a flag-and-route function, with verification, contextualization, and documentation performed by a human reviewer even as anomaly-detection models continue to improve. That position draws on validation-and-control-check design work carried out across large multi-entity FCCS/ARCS-class deployments, where approval workflows and intercompany-match tolerance checks were built specifically to interrupt automatic posting whenever a reconciliation item drifted outside a defined tolerance band. The underlying systems were not incapable of auto-resolving minor variances, they could have been configured that way. They were built to interrupt instead, because auto-resolution erases the audit trail a controller or external auditor later needs in order to reconstruct why a number moved.

What follows is a systems-architecture argument, not a machine-learning one, and that distinction matters for how the rest of the article should be read. Reconciliation exceptions sit at a junction between two literatures that rarely cite each other: anomaly-detection research, which optimizes for catching outliers, and audit-governance research, which asks whether a caught outlier received appropriate human review before it changed a reported figure. This article synthesizes both into a single design argument aimed at consolidation-platform architects. Its contribution to that synthesis is a governance framework, not a new detection method: a four-stage flag-contextualize-verify-document pipeline (Section III) that names where human judgment must sit inside an AI-augmented close cycle, paired with a maturity taxonomy for AI-flagging design (Table 1) that gives architects a vocabulary for the difference between a technically capable flag and a governable one. The taxonomy and pipeline are offered as a conceptual instrument for consolidation-platform design, in the same spirit as prior human-in-the-loop framing work [4], applied here specifically to the reconciliation-exception decision point that framing work does not itself address. It proposes a taxonomy of AI-flagging design elements (Table 1) and sets autonomous posting against human-in-the-loop flagging on accountability, auditability, and adoption durability (Table 2). A flowchart (Figure 1) formalizes the flag-contextualize-verify-document handoff the argument rests on. Short version: detection is easy. Disposition is where the risk lives.

II. Literature Review: Three Tensions in AI-Assisted Reconciliation

A. Pattern Recognition Suits Detection, Not Disposition

Statistical anomaly-detection methods generalize well across domains, network intrusion, fraud, financial outliers, and their foundations are well established [2]. Financial-statement research builds on that foundation directly: machine-learning classifiers trained on financial-statement variables can predict material misstatements with meaningful accuracy [10], and models built from accounting ratios and textual disclosure signals demonstrate separation between clean and problematic filings [11]. Fraud-detection work pushes the case further into production-scale evidence, an XGBoost-based framework for mobile-payment fraud detection has been validated on more than 6 million transactions [15], a scale that argues for genuine operational maturity rather than a laboratory proof of concept. Imbalance-aware methods target the rare-event character of financial fraud specifically [14], while deep-learning approaches extend detection into unstructured disclosure text for financial-statement fraud [16]. A hybrid Random Forest-based AdaBoost model applied to going-concern opinion forecasting reportedly outperforms other machine-learning and hybrid approaches by 89% in accuracy terms [12], and full-population audit methods demonstrate that anomaly scoring can be applied across an entire transaction population instead of a sample [13].

None of this closes the gap that matters most for reconciliation work: detecting an anomaly is not the same task as deciding what that anomaly means for a specific ledger entry, entity relationship, or intercompany pairing. A model trained on historical misstatement patterns has no way of knowing why an entity restated an opening balance this quarter. It cannot know why a currency-translation adjustment landed outside its usual band this period. Business-process-monitoring research applying machine learning to event logs runs into the identical wall, models flag process deviations reliably enough, but distinguishing an error from a legitimate exception demands context the model simply does not hold [21]. Detection accuracy and disposition accuracy are different problems, and the literature answers the first one far more confidently than the second.

B. Verification Behavior Does Not Track Model Accuracy

A second tension concerns how professionals respond to AI output, somewhat independent of how good that output actually is. Trust in automation is not fixed; it appears to depend on the transparency of a system's reasoning and an operator's ability to calibrate reliance against demonstrated performance over time [1]. That calibration process turns out to be lopsided. Professionals who watch an algorithm err once tend to discount its future output more severely than they would discount a comparable human error, a pattern established in decision-making research and since replicated

in advisory settings [9]. Advisory-trust research complicates this further: trust in algorithmic advice does track advice accuracy, but not in a straight line, and users appear to penalize algorithmic error more harshly than raw accuracy figures would justify [18].

Audit-specific evidence sharpens the picture. Auditors evaluating complex management estimates propose smaller adjustments when contradictory evidence comes from an AI system than when the same evidence comes from a human specialist [17], a finding that cuts squarely against the assumption that professionals simply defer to machines. Exploratory research on AI adoption among auditors finds a comparable ambivalence, practitioners recognize efficiency gains but remain guarded about substituting algorithmic judgment for professional skepticism [8]. Clinical decision-support research, drawn from outside finance but structurally close, surveyed 119 users of an AI-based clinical decision-support system and found that willingness to act on a recommendation hinged on perceived explainability and control far more than on the system's stated accuracy [20]. Verification behavior, in short, does not scale down as machine accuracy scales up. It scales with explainability and control. Trust is earned, not inferred from a benchmark score.

C. Governance and Accountability Cannot Be Delegated to the Model

The third tension is institutional rather than behavioral. Explainability research separates models that are interpretable by design from models whose reasoning must be reconstructed after the fact, and the DARPA XAI program treated that distinction as a national research priority rather than an academic footnote [3]. Taxonomies of explainable-AI methods have multiplied since, cataloguing a wide range of technical approaches for making model reasoning legible to a reviewer [5], while experimental work shows that the mere presence of an explanation changes user behavior even when the explanation adds no new predictive content [19]. Explainability, then, does governance work as much as technical work, arguably more.

That governance dimension shows up directly in accounting and audit ethics research, which raises pointed accountability questions about AI-assisted audit judgment: who answers for a wrong flag, and can a firm discharge professional responsibility simply by pointing at a model's output [7]. Machine-readable accounting standards research extends the concern into disclosure architecture, contending that XBRL-tagged, AI-processed financial data is quietly reshaping what counts as an accounting judgment in the first place [6]. Risk-disclosure research grounds the concern empirically: a three-year panel of 10-K filings from 73 publicly listed S&P financial firms reveals how financial institutions construct accountability language around algorithmic decision systems in their public risk disclosures [22], evidence that boards are already treating AI governance as its own disclosure category, distinct from ordinary IT risk. Human-in-the-loop machine learning research frames the underlying design principle plainly: keeping a human decision point inside the loop is not a stopgap on the road to full automation. It is a deliberate architecture choice, with its own accountability profile [4].

III. Methodology and Analytical Framework

This article works as a conceptual and design-architecture analysis rather than an empirical study built on freshly collected data. Two inputs feed the analysis. The first is the literature synthesis above, organized around the three tensions just discussed. The second is a structured reflection on validation-and-control-check architecture built across two multi-entity FCCS/ARCS-class consolidation deployments, a rollout spanning more than 50 entities for the Al Fozan Group in Saudi Arabia, and a 13-entity rollout for the DTAC Group in Thailand. "Structured" here means a specific, bounded set of artifacts rather than unaided recollection: configuration change tickets and approval-workflow design documents from both rollouts, post-go-live incident notes covering the routing and false-positive episodes discussed below, and the author's own architecture-decision records written contemporaneously during each deployment to justify why a given tolerance-check or escalation rule was configured the way it was. The reflection was organized retrospectively around the four pipeline stages introduced later in this section, flag, contextualize, verify, document, coding each configuration decision, ticket, and incident note against the stage it

primarily affected. This is a single-author retrospective synthesis, not an interview-based or multi-rater qualitative study, and the analysis does not claim the generalizability of a larger-sample field study; instead, it claims the specificity of two complete, documented rollouts examined end to end. Both deployments required intercompany-match tolerance checks, approval-workflow configuration, and exception-routing logic engineered to force a human decision before any out-of-tolerance reconciliation item could post to the consolidated ledger (this pairing of literature and implementation experience is unusual for a paper in this space, which tends to sit on one side or the other).

The resulting framework treats reconciliation-exception handling as a four-stage pipeline: flag, contextualize, verify, document. A flag, whether rules-based or model-generated, marks a ledger item as needing review. Contextualization attaches entity-specific, period-specific, and relationship-specific information the model never natively holds: a restated opening balance, a known timing difference inside an intercompany pairing, a currency-translation quirk tied to one subsidiary's functional currency. Verification is the reviewer's judgment call, informed by the flag and its context but not dictated by either. Documentation captures the decision, its rationale, and a timestamp inside the platform's audit trail. Figure 1 renders this sequence as a flowchart, branching at the contextualization stage between auto-close items (low-risk, within tolerance) and escalation items (requiring controller or auditor sign-off).

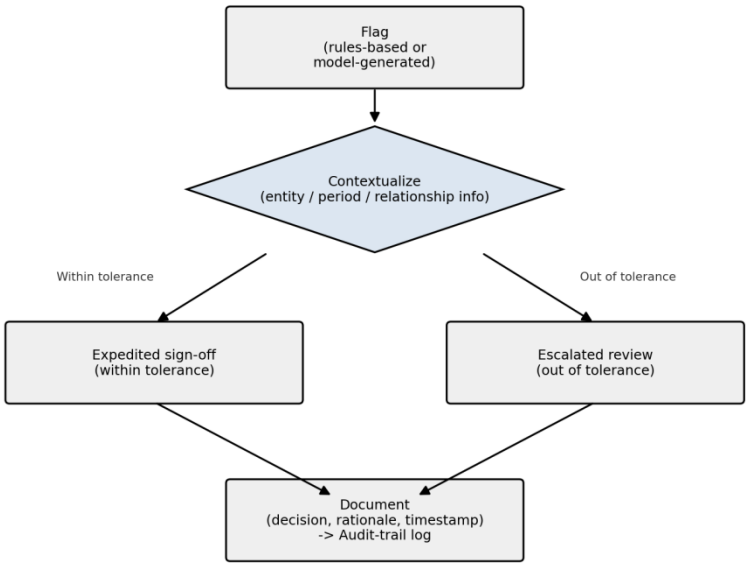

Figure 1. Flag → Contextualize → Verify → Document Handoff Sequence

Table 1 organizes the design elements separating a well-architected AI-flagging layer from a weak one, using three dimensions the literature synthesis points toward and that the deployment reflection independently converged on: context provided to the reviewer at the point of flagging (the gap identified in business-process-monitoring research, where a model flags a deviation but cannot supply the entity- or period-specific reason behind it [21]), reasoning transparency, whether the reviewer can see why the flag fired (the interpretability-versus-post-hoc-reconstruction distinction DARPA's XAI program treated as a national priority [3], and the subject of subsequent XAI-taxonomy work [5]), and escalation path, whether the routing logic reaches the right reviewer at the right authority level (the design variable human-in-the-loop research frames as a deliberate accountability architecture rather than a stopgap [4]). No published source proposes this three-row maturity ladder as such; Table 1 is the author's synthesis, built by mapping the Al Fozan and DTAC configuration history against those three literature-derived dimensions, and it is presented here as a design heuristic for architects, not as an externally validated instrument. The escalation-path dimension is not an abstraction pulled from the literature and then illustrated

after the fact; it is the dimension the Al Fozan deployment surfaced directly, and the episode is worth reconstructing in more detail than a passing mention allows. An early version of the tolerance-check logic on that engagement routed out-of-tolerance intercompany mismatches to a single, undifferentiated reconciliation queue shared across all fifty-plus entities, rather than to the controller responsible for the specific entity pair involved. The routing rule keyed on transaction type (intercompany elimination, currency-translation adjustment, and so on) rather than on entity hierarchy, on the assumption that transaction type would correlate closely enough with the right reviewer. It did not. Items sat in the shared queue for review cycles at a time because no single owner had clear responsibility for triaging it, and multiple controllers assumed a mismatch belonging to a peer entity would be picked up by someone else. The failure mode was diagnosed only after a controller escalated a stale item during a close-cycle status review and asked, pointedly, who actually owned queue triage. The remediation rebuilt the escalation path around entity hierarchy: each flagged item now routes first to the controller of record for the entity that generated it, with transaction type used only as a secondary filter for prioritization within that controller's queue, not as the primary routing key. That fix was not planned into the original design; it emerged from a documented production failure. It was carried forward as a standing design rule into the thirteen-entity DTAC rollout, where entity-hierarchy-first routing was configured from the initial build rather than retrofitted, which is itself evidence for the taxonomy's claim that escalation-path design deserves the same deliberate attention as detection-model selection.

Table 1. Taxonomy of AI-Flagging Design Elements by Maturity Level

Flagging Maturity	Context Provided	Reasoning Transparency	Escalation Path
Rules-based static threshold	Minimal - threshold breach only	High - rule fully known in advance	Often generic queue routing
Anomaly-detection statistical flag	Moderate - statistical deviation noted	Low - model internals opaque	Variable - depends on platform design
Explainable-AI flag with feature attribution	High - contributing factors surfaced	Moderate-to-high - feature attribution visible	Can be tuned to entity/authority level

IV. Results and Discussion

The synthesis points toward a specific design conclusion, and it holds regardless of whether the detection layer is a simple statistical threshold or a more sophisticated anomaly-detection model [2]: AI output belongs in the architecture as an input to human verification, not as a stand-in for it. This is not a cautionary flourish tacked onto an otherwise technical argument. It follows directly from how an audit trail gets built. An autonomous posting event that resolves a flagged discrepancy without a documented human decision leaves the platform unable to answer a question an auditor will eventually ask: why was this discrepancy treated as immaterial? Approval-workflow architecture in FCCS/ARCS-class platforms answers that question by design, because every exception above a configured tolerance generates a reviewer action record. No equivalent record exists when a model resolves the same exception on its own.

Table 2 sets autonomous posting against human-in-the-loop flagging across three dimensions: accountability (does the decision trace to a named reviewer), auditability (can an external auditor reconstruct the decision path without relying on model internals), and adoption durability (does the control survive staff turnover, system migration, and regulatory scrutiny). Human-in-the-loop flagging appears to outperform autonomous posting on all three dimensions in the architecture examined here, though this comparison is qualitative, not a scored benchmark, since no dataset directly measures adoption-durability outcomes across the two architectures at scale. That gap in the evidence base is itself worth flagging, no pun intended. Table 2 should be read as a structured restatement of the argument's logic, not as an independent evidentiary result: it organizes what the

argument implies about the two architectures rather than reporting a measurement of them. The evidentiary weight in this article sits instead in the two deployment episodes reconstructed in Sections III and IV, the Al Fozan escalation-path failure and the recurring false-positive patterns in both rollouts, which are the closest thing this paper offers to observed data, and readers should weight the table accordingly.

Table 2. Autonomous Posting vs. Human-in-the-Loop Flagging

Dimension	Autonomous Posting	Human-in-the-Loop Flagging
Accountability	No named reviewer; decision attributed to model	Decision traces to a named, accountable reviewer
Auditability	Requires reconstructing model internals	Reconstructable from documented reviewer action record
Adoption durability	Vulnerable to trust collapse after visible errors [9]	Survives turnover/migration via documented process, not model dependency

Adoption research complicates the purely architectural case, however. Auditors report ambivalence toward AI tools even where efficiency gains are visible [8], and algorithm-aversion research suggests a single visible model error can suppress future reliance disproportionately to its actual severity [9]. A human-in-the-loop architecture, read this way, is not only a safeguard against model error. It is also a hedge against the adoption failure that follows once reviewers lose trust in a system faster than the system's accuracy would justify. Flag-and-route design gives reviewers repeated, low-stakes chances to build calibrated trust, precisely because every flag gets verified rather than acted on unilaterally.

The explainability findings reinforce this from a different angle. An explanation changes reviewer behavior even when it adds no new predictive content [19], which suggests the value of an explainable-AI layer in reconciliation work is partly psychological and partly procedural. Procedurally, an explained flag gives the reviewer a starting point for the contextualization stage. Psychologically, it makes the reviewer more willing to actually engage with the flag rather than dismiss or rubber-stamp it. Both effects matter for close-cycle governance. A rubber-stamped flag is functionally no review at all, even with a human name sitting on the approval record.

One friction point deserves direct acknowledgment rather than a footnote. Tolerance-band logic in both the Al Fozan and DTAC deployments occasionally generated false-positive escalations: items flagged as out-of-tolerance that a reviewer, on inspection, judged immaterial and explainable by a known timing difference. The pattern recurred in a small number of identifiable shapes rather than at random. Month-end cutoff differences between two entities closing on slightly different calendars produced timing mismatches that cleared automatically the following period; these accounted for a recurring share of escalations in the Al Fozan rollout until controllers began tagging them at the point of review so the pattern was visible across cycles rather than re-diagnosed each month. A second recurring shape involved intercompany loan interest accruals that posted a day apart across two entities' local ledgers, tripping the tolerance band on same-day comparison even though both entities' books tied out correctly once the accrual date lag was accounted for. In the DTAC deployment, a comparable pattern surfaced around foreign-exchange revaluation timing on intercompany balances denominated in a currency other than either entity's functional currency. None of these were adjudicated by loosening the tolerance band, which would have let genuine exceptions through along with the recurring false positives; instead, each documented pattern was fed back into the contextualization stage as a standing annotation, so a reviewer encountering the same shape again could clear it in seconds rather than re-investigating from scratch. This was not an architecture failure. It was the architecture doing exactly what it was designed to do, at the cost of reviewer time, and the cost was reduced not by suppressing flags but by making the human review step faster to execute. A lower false-positive rate from a more refined anomaly-detection layer would reduce

that cost further [2], provided its output stays subject to the same verification discipline. A more accurate model bolted onto an autonomous-posting architecture would not reduce risk at all. It would automate more errors with more confidence.

V. Broader Implications

The argument extends past reconciliation exceptions to any close-cycle control point where machine-learning output could plausibly trigger an automated action on its own: intercompany elimination adjustments, currency-translation overrides, journal-entry approval routing, disclosure-tagging validation under machine-readable reporting standards [6]. The governing question stays constant across all of these: does the control architecture generate a documented, attributable human decision before system state changes, or does a model's confidence score quietly substitute for that decision? Risk-disclosure evidence from S&P financial firms suggests boards already treat this as a distinct governance category, separate from ordinary system-availability risk [22]. Consolidation-platform architects should expect that separation to sharpen rather than soften as AI-flagging layers move from novel add-on to standard feature across EPM platforms.

There is a regulatory dimension the consolidation-platform community has not given enough attention, and it may be the more consequential gap. Audit-ethics research raises accountability questions that regulators have only begun translating into binding disclosure or control requirements [7], and machine-readable accounting research suggests the infrastructure for AI-processed financial data is arriving well ahead of the governance frameworks meant to constrain it [6]. Enterprise architects building exception-routing logic today are, in effect, pre-empting requirements regulators will likely formalize later anyway. None of this argues for slowing AI adoption in reconciliation tooling. It argues for architects treating documentation-layer design as a regulatory dependency now, before a supervisor names it as one, rather than as a retrofit once a disclosure requirement forces the issue.

A practical note for platform vendors and in-house architecture teams follows from this. Flagging maturity (Table 1) deserves treatment as a governance variable configured on purpose, not a technical setting left at a vendor default. A model with high reasoning transparency that still routes to a generic queue delivers weak governance value despite strong technical performance, the Al Fozan routing friction described in Section III is a small-scale version of exactly this problem. Escalation-path design probably deserves as much architectural attention as detection-model selection, maybe more. In practice this remains an under-resourced discipline: implementation budgets for AI-augmented close tooling still weight model selection and data-pipeline engineering far more heavily than routing-logic design and reviewer-facing explanation interfaces, a misallocation given how much of the accountability and auditability benefit seems to derive from the latter (Author's primary implementation data; broadly corroborated by adoption-ambivalence findings in the audit literature) [8]. Budgets, in other words, are pointed at the wrong half of the problem.

VI. Conclusion

A reconciliation exception is a governance event before it is a technical one. Treating it as a problem for an increasingly accurate model to resolve quietly misreads what the close cycle actually needs, not fewer flags, but flags routed with enough context to accountable reviewers who leave a documented decision behind them. The architecture already sitting inside multi-entity consolidation platforms, approval workflows, intercompany-match tolerance checks, exception-routing logic, was not built in anticipation of machine learning specifically, yet it anticipates almost exactly the governance problem machine learning now introduces. Folding AI-generated flags into that architecture looks like a natural next step. Swapping the architecture out for autonomous posting is not a next step. It is a regression wearing efficiency as a disguise. The reviewer verifies a flag and records the reasoning behind a decision, doing work that no model can perform on its own. Flag it. Do not fix it unattended.

Acknowledgment

The author acknowledges the enterprise consolidation-platform teams whose deployment experience informed the architectural observations in this article.

Author Contributions

Srinivasarao Challagundla is the sole author and is responsible for the conceptualization, analysis, and writing of this article in its entirety.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] John D. Lee, Katrina A. See, "Trust in Automation: Designing for Appropriate Reliance," Human Factors, Vol. 46, No. 1, 2004, pp. 50-80. https://journals.sagepub.com/doi/10.1518/hfes.46.1.50_30392

[2] Varun Chandola, Arindam Banerjee, Vipin Kumar, "Anomaly Detection: A Survey," ACM Computing Surveys, Vol. 41, No. 3, Article 15, 2009, pp. 1-58. https://dl.acm.org/doi/10.1145/1541880.1541882

[3] David Gunning, David W. Aha, "DARPA's Explainable Artificial Intelligence (XAI) Program," AI Magazine, Vol. 40, No. 2, 2019, pp. 44-58. https://dl.acm.org/doi/10.1145/3301275.3308446

[4] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, Ángel Fernández-Leal, "Human-in-the-Loop Machine Learning: A State of the Art," Artificial Intelligence Review, Vol. 56, No. 4, 2023, pp. 3005-3054. https://link.springer.com/article/10.1007/s10462-022-10246-w

[5] Timo Speith, "A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods," Proc. ACM FAccT '22, 2022, pp. 2239-2250. https://dl.acm.org/doi/10.1145/3531146.3534639

[6] Alessio Faccia, "Machine-Readable Accountability: eXtensible Business Reporting Language, Artificial Intelligence, and the Institutional Rewriting of Accounting Judgement," Journal of Risk and Financial Management, Vol. 19, No. 7, Article 547, 2026. https://www.mdpi.com/1911-8074/19/7/547

[7] Ivy Munoko, Helen L. Brown-Liburd, Miklos Vasarhelyi, "The Ethical Implications of Using Artificial Intelligence in Auditing," Journal of Business Ethics, Vol. 167, No. 2, 2020, pp. 209-234. https://link.springer.com/article/10.1007/s10551-019-04407-1

[8] Ravi Seethamraju, Angela Hecimovic, "Adoption of Artificial Intelligence in Auditing: An Exploratory Study," Australian Journal of Management, Vol. 48, No. 4, 2023, pp. 780-800. https://www.researchgate.net/publication/361834015_Adoption_of_artificial_intelligence_in_auditing_An_exploratory_study

[9] Berkeley J. Dietvorst, Joseph P. Simmons, Cade Massey, "Algorithm Aversion: People Erroneously Avoid Algorithms After Seeing Them Err," Journal of Experimental Psychology: General, Vol. 144, No. 1, 2015, pp. 114-126. https://www.researchgate.net/publication/268449803_Algorithm_Aversion_People_Erroneously_Avoid_Algorithms_After_Seeing_Them_Err

[10] Chanyuan Abigail Zhang Parker, Lanxin Jiang, Soohyun Cho, Miklos A. Vasarhelyi, "Predicting Material Misstatements Using Machine Learning," The Accounting Review, Vol. 100, No. 6, 2025, pp. 225-262. https://www.researchgate.net/publication/394285297_Predicting_Material_Misstatements_Using_Machine_Learning

[11] Jeremy Bertomeu, Edwige Cheynel, Eric Floyd, Wenqiang Pan, "Using Machine Learning to Detect Misstatements," Review of Accounting Studies, Vol. 26, No. 2, 2021, pp. 468-519. https://link.springer.com/article/10.1007/s11142-020-09563-8

[12] Alpaslan Yaşar, "Forecasting Auditor's Going Concern Opinion Using with Hybrid Robust Machine Learning Model," PLOS ONE, Vol. 21, No. 3, 2026, Article e0345071. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0345071

[13] Hui Yan, "A Full Population Auditing Method Based on Machine Learning," Sustainability, Vol. 14, No. 24, Article 17008, 2022. https://www.mdpi.com/2071-1050/14/24/17008

[14] Ezaz Mohammed Al-dahasi, Rama Khaled Alsheikh, Fakhri Alam Khan, Gwanggil Jeon, "Optimizing Fraud Detection in Financial Transactions with Machine Learning and Imbalance Mitigation," Expert Systems, Vol. 42, No. 2, Article e13682, 2025. https://onlinelibrary.wiley.com/doi/10.1111/exsy.13682

[15] Petr Hajek, Mohammad Z. Abedin, Uthayasankar Sivarajah, "Fraud Detection in Mobile Payment Systems Using an XGBoost-based Framework," Information Systems Frontiers, Vol. 25, No. 5, 2023, pp. 1985-2003. https://link.springer.com/article/10.1007/s10796-022-10346-6

[16] Patricia Craja, Alisa Kim, Stefan Lessmann, "Deep Learning for Detecting Financial Statement Fraud," Decision Support Systems, Vol. 139, Article 113421, 2020. https://www.sciencedirect.com/science/article/abs/pii/S0167923620301767

[17] Benjamin P. Commerford, Sean A. Dennis, Jennifer R. Joe, Jenny W. Ulla, "Man Versus Machine: Complex Estimates and Auditor Reliance on Artificial Intelligence," Journal of Accounting Research, Vol. 60, No. 1, 2022, pp. 171-201. https://onlinelibrary.wiley.com/doi/abs/10.1111/1475-679X.12407

[18] Stefan Daschner, Robert Obermaier, "Algorithm Aversion? On the Influence of Advice Accuracy on Trust in Algorithmic Advice," Journal of Decision Systems, Vol. 31, No. sup1, 2022, pp. 77-97.

https://www.tandfonline.com/doi/full/10.1080/12460125.2022.2070951

[19] H. Felix Wittmann, "Explanation Matters: An Experimental Study on Explainable AI," Electronic Markets, Vol. 33, No. 1, Article 17, 2023. https://www.researchgate.net/publication/370655066_Explanation_matters_An_experimental_study_on_explainable_AI

[20] Avishek Choudhury, "Factors Influencing Clinicians' Willingness to Use an AI-Based Clinical Decision Support System," Frontiers in Digital Health, Vol. 4, Article 920662, 2022. https://www.frontiersin.org/journals/digital-health/articles/10.3389/fdgth.2022.920662/full

[21] Wolfgang Kratsch, Jonas Manderscheid, Maximilian Röglinger, Johannes Seyfried, "Machine Learning in Business Process Monitoring," Business & Information Systems Engineering, Vol. 63, No. 3, 2021, pp. 261-276. https://aisel.aisnet.org/bise/vol63/iss3/5/

[22] Eleni Lioliou, M. May Seitanidi, Lea Stadtler, "Governance under Algorithmic Opacity: How Financial Firms Construct Accountability and Control around AI in Risk Disclosures," Information Systems Frontiers, 2026, pp. 1-17. https://link.springer.com/article/10.1007/s10796-026-10762-y