



## Digital Ledger Optimization Techniques for Enhancing Transaction Speed and Reporting Accuracy in Accounting Systems

Faysal Khan<sup>1</sup>; Shamsunnahar Chadni<sup>2</sup>;

[1]. Bachelor in Business Administration; Department of Accounting & Information Systems; Jagannath University, Dhaka, Bangladesh;  
E-mail: [faysalkhan.bd@gmail.com](mailto:faysalkhan.bd@gmail.com);

[2]. Bachelor of Management, National University Bangladesh, Dhaka, Bangladesh;  
Email: [shamsunnahar.chdn@gmail.com](mailto:shamsunnahar.chdn@gmail.com)

Doi: [10.63125/33t06k57](https://doi.org/10.63125/33t06k57)

Received: 20 September 2022; Revised: 23 October 2022; Accepted: 25 November 2022; Published: 29 December 2022

### Abstract

This quantitative study examined how digital ledger optimization techniques influenced transaction speed and reporting accuracy in accounting systems using archival operational data. A panel dataset comprising 360 unit-period observations from 30 accounting environments observed over 12 reporting periods was analyzed. Optimization intensity was measured across three layers – data, process, and control – while transaction speed was captured through posting latency, throughput, and queue waiting time, and reporting accuracy was measured using reconciliation mismatch rates, journal error rates, and correction entry frequency and value. Descriptive results showed that median posting latency was 9.6 seconds, while tail latency reached 44.2 seconds at the 95th percentile, confirming pronounced peak-load effects. Regression findings indicated that data-layer optimization was strongly associated with reduced posting latency ( $\beta = -0.41, p < .001$ ), while process-layer optimization was most strongly associated with increased throughput ( $\beta = 0.39, p < .001$ ). Control-layer optimization demonstrated the strongest relationships with reporting accuracy outcomes, including lower reconciliation mismatch rates ( $\beta = -0.51, p < .001$ ) and reduced correction entry frequency ( $\beta = -0.53, p < .001$ ). Moderation analysis revealed that validation strictness reduced the magnitude of batching-related speed gains (interaction  $\beta = 0.13, p = .020$ ), while automation maturity strengthened the relationship between optimization intensity and reporting accuracy (interaction  $\beta = -0.15, p = .009$ ). Robustness checks confirmed that results remained stable when alternative dependent-variable definitions were applied, including tail latency and mismatch value. Overall, the findings demonstrated that transaction speed improvements were most strongly associated with data- and process-layer optimization, while reporting accuracy improvements were primarily associated with control-layer practices. The study provided empirical evidence that layered digital ledger optimization contributed to faster processing and more reliable financial reporting under real operational conditions.

### Keywords

Digital Ledger Optimization, Transaction Speed, Reporting Accuracy, Accounting Systems, Information Quality.

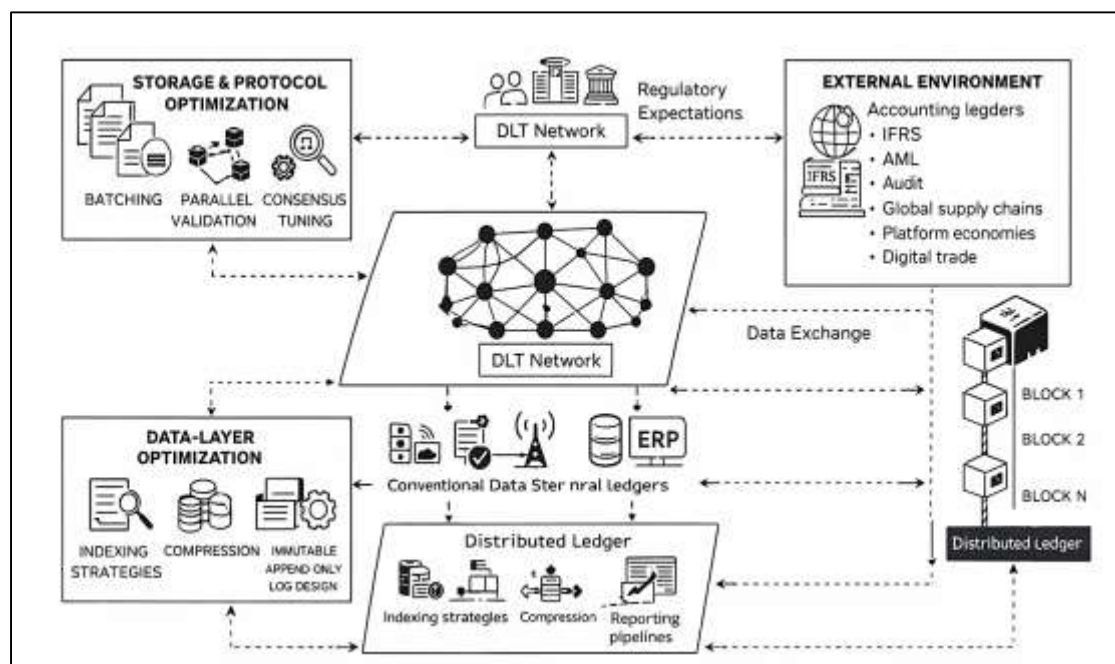
## **INTRODUCTION**

Digital ledger optimization refers to the systematic improvement of ledger architectures, protocols, and data-processing workflows to increase transaction throughput (speed), reduce confirmation latency, and improve the fidelity of recorded events used for financial reporting. In accounting systems, a digital ledger can be defined as an electronically maintained record of economic events that preserves integrity (protection against unauthorized alteration), traceability (clear linkage from source event to summarized report), and auditability (verifiable evidence of processing and controls) (Wu et al., 2019). Digital ledgers include conventional database-backed general ledgers and subledgers, enterprise resource planning (ERP) journals, and distributed ledger technologies (DLTs) in which multiple nodes maintain synchronized copies of transactions under agreed validation rules. Optimization techniques in this context span (a) protocol-level mechanisms such as batching, parallel validation, and consensus tuning; (b) data-layer approaches such as indexing strategies, compression, and immutable append-only log design; and (c) application-layer controls such as structured transaction schemas, validation constraints, and reporting pipelines that reduce reconciliation errors. International significance arises because transaction processing and reporting accuracy are evaluated under global regulatory expectations – such as IFRS-aligned reporting environments, cross-border tax compliance, anti-money laundering recordkeeping, and multinational audit requirements – where the timeliness and reliability of transaction evidence directly affect close cycles, assurance activities, and stakeholder confidence. Digital trade, platform economies, and internationally distributed supply chains amplify the need for ledgers that can process high volumes of micro- and macro-transactions while maintaining consistent classifications, timestamps, and documentation across jurisdictions (Zhang et al., 2019). Quantitative accounting research increasingly treats ledger performance and reporting accuracy as measurable system outcomes, enabling empirical evaluation of optimization techniques using metrics such as transactions per second, end-to-end latency, error rates in journal posting, variance in reconciliation, and the rate of reporting restatements attributable to data-processing faults. Within this framing, the introduction to digital ledger optimization must begin with core definitions of ledger integrity, transaction finality, and reporting accuracy, then connect these constructs to measurable performance indicators and control objectives relevant to international accounting systems. Rather than treating speed and accuracy as competing goals, optimization work often operationalizes them as a multi-objective problem in which system design seeks to minimize latency and maximize correctness under constraints such as security, governance, and audit trails. A quantitative paper on this topic therefore begins by situating digital ledger optimization within accounting information systems (AIS) theory, data quality research, and distributed systems performance measurement, establishing the conceptual basis for measurable hypotheses about transaction speed and reporting accuracy without advancing a concluding claim (Viriyasitavat et al., 2019).

Transaction speed in accounting-ledger contexts is best defined as the time required for an economic event to move from initiation (capture at the point of sale, invoice approval, payroll run, inventory movement, or intercompany posting) to a recognized state in the ledger that can be reliably included in trial balances and reports (Pei & Vasarhelyi, 2020). This span includes message propagation, validation, authorization, posting, and synchronization across relevant subsystems. In centralized AIS, the bottleneck may originate in database locks, sequential journal posting routines, ETL batch schedules, or constrained API throughput. In distributed or multi-entity ledgers, bottlenecks can arise from network latency, consensus overhead, serialization constraints, and the computational costs of cryptographic validation. Quantitative performance analysis therefore distinguishes throughput (volume per unit time), latency (time per transaction), and finality (probability that a recorded transaction will remain stable against reversal) as separable constructs. The accounting significance of these constructs is international: global companies often operate around-the-clock, and financial close processes integrate data streams from subsidiaries in multiple time zones with different statutory calendars and local reporting rules. If transaction processing is slow or inconsistent, firms may rely on estimates and manual adjustments, increasing the likelihood of classification errors and reducing traceability. Reporting accuracy, by contrast, can be defined as the degree to which ledger-derived outputs (general ledger balances, subledger reconciliations, management reports, statutory statements, and audit schedules) correctly represent underlying economic reality according to applicable

measurement rules and internal policies (Bogdanov et al., 2020). Accuracy can be decomposed into data correctness (valid values and calculations), completeness (all relevant events captured), consistency (alignment across modules and entities), and verifiability (supporting evidence and audit trails). Optimization techniques connect these domains because improvements in transaction speed can change the temporal distribution of postings (e.g., moving from batch to near-real-time), while improvements in accuracy often involve stronger validation logic, richer metadata, and reconciliation controls that affect processing costs. A quantitative study must therefore treat speed and accuracy as jointly measurable outcomes, using standardized definitions and measurement models that allow statistical analysis across organizations, platforms, or experimental configurations. In international environments, additional measurement challenges emerge from currency translation timing, intercompany eliminations, differing invoice standards, and heterogeneous tax documentation requirements, all of which place pressure on ledger designs to store structured data, retain immutable evidence, and support automated aggregation (Polak et al., 2020). Digital ledger optimization becomes a focal research area because it offers a design-oriented pathway to reduce operational frictions in transaction processing while strengthening the reliability of reporting pipelines, all within auditable system boundaries.

Figure 1: Digital Ledger Optimization Framework



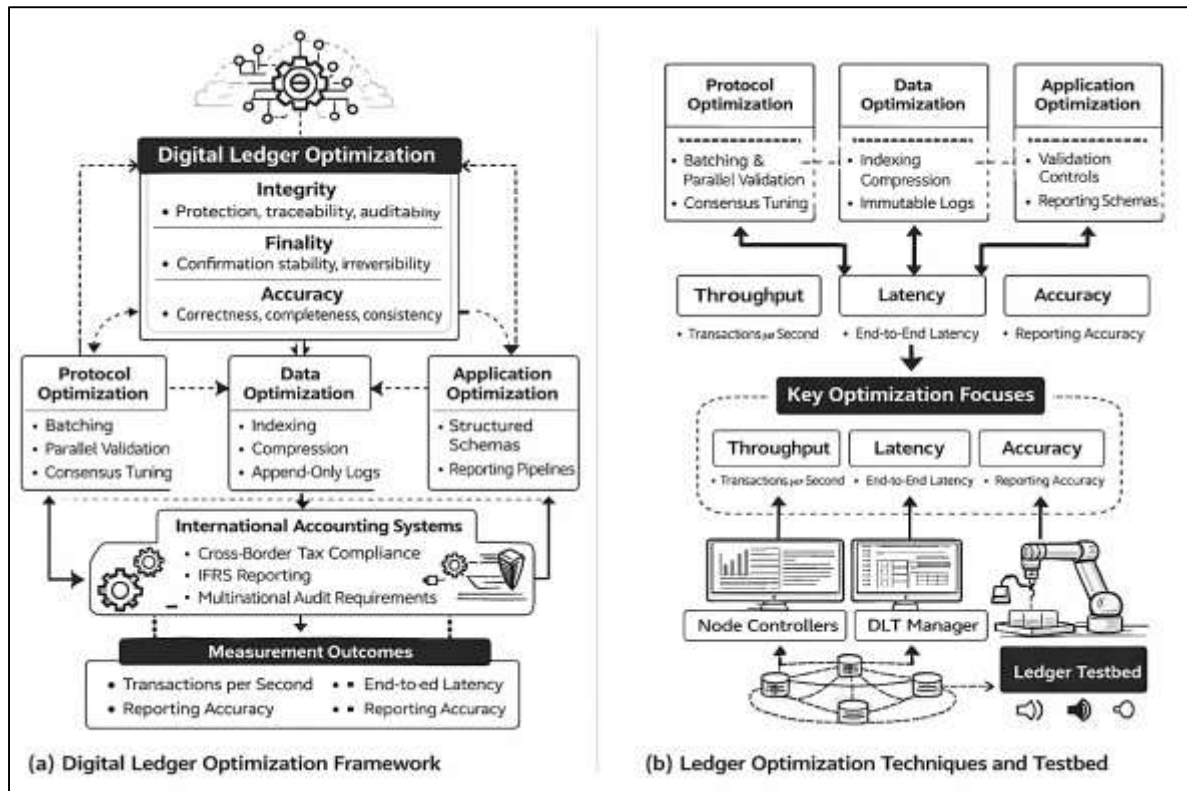
A central technical foundation for optimizing digital ledgers is the design of data structures and storage engines that support high-volume append operations, rapid retrieval for reporting, and strong integrity constraints. Accounting ledgers have long relied on journaling and log-based recovery principles in database systems, where write-ahead logging, checkpointing, and indexing determine both performance and the reproducibility of records for audit (Yoon, 2020). Modern optimization extends these principles by adopting append-only event logs, columnar storage for analytical workloads, hybrid transactional/analytical processing patterns, and specialized indexing strategies that accelerate common accounting queries such as account rollups, aging schedules, and variance analyses. In distributed settings, optimization includes marbleized data structures that allow efficient proofs of inclusion, partitioned state representations that reduce validation overhead, and cryptographic hashing schemes that preserve tamper evidence while enabling efficient incremental updates. These design choices matter for reporting accuracy because the reliability of a financial report depends on consistent data lineage: each summarized figure should be traceable to source events, transformations, and authorization states. Quantitative evaluation of storage optimization can use metrics such as write

amplification, read latency for standard report queries, compression ratios, and the rate of reconciliation exceptions generated under different schema designs. International significance is strengthened by the need to produce consistent reporting outputs from globally distributed transaction sources while preserving evidentiary chains for external auditors and regulators (Hunhevicz & Hall, 2020). Many multinational accounting environments now integrate digital documents (e-invoices, shipping notices, bank feeds, and payroll confirmations) as structured or semi-structured data, making schema design and metadata governance central to accuracy outcomes. For example, standardized tax invoice fields, cross-border customs attributes, and IFRS/GAAP mapping tags must be stored reliably to support consistent classification and disclosure. Optimization techniques such as schema normalization, controlled vocabularies, constraint-based validation, and deterministic transformation pipelines can reduce reporting discrepancies attributable to heterogeneous data capture. At the same time, speed improvements obtained through caching, asynchronous processing, or aggressive batching can introduce risks of temporary inconsistency across modules if not paired with strong reconciliation protocols and clear finality definitions. A quantitative research design can treat these factors as independent variables—ledger storage model, indexing strategy, validation strictness, and batching configuration—and relate them to dependent variables such as close-cycle duration, frequency of posting corrections, mismatch rates between subledger and general ledger, and error rates in automated classification. This approach grounds the paper in measurable AIS outcomes while keeping the introduction anchored in definitions and internationally relevant system requirements (Du et al., 2020).

Distributed ledger technologies introduce an additional optimization domain: consensus and coordination. Consensus protocols define how nodes agree on the order and validity of transactions, directly shaping throughput, latency, and finality. Permissionless systems often use probabilistic finality mechanisms, while permissioned enterprise ledgers can employ crash-fault tolerant or Byzantine-fault tolerant protocols with deterministic finality (Fikri et al., 2019). Optimization techniques include reducing communication complexity, employing leader-based replication, using pipelining and batching to amortize overhead, and tailoring fault assumptions to the governance environment. For accounting systems, consensus design influences reporting accuracy through transaction ordering (which affects cut-off), timestamping, and the stability of posted entries used for period-end reporting. In cross-border environments where different entities may host nodes, governance rules define who can validate, who can view which data, and how disputes or corrections are handled within an auditable framework. Quantitative evaluation can operationalize consensus choices as measurable determinants of transaction speed and stability, comparing configurations by throughput, latency distribution, confirmation depth, and the frequency of reorganization or correction events. Optimization also includes layering strategies: keeping high-frequency transactional activity off-chain or off the shared consensus layer while anchoring summaries or proofs to the ledger for audit and integrity. In accounting, this can correspond to posting detailed operational events within local systems while writing immutable period-based commitments, hash-linked journals, or reconciled aggregates to a shared ledger used for intercompany coordination or assurance. Another optimization approach is parallelism, including sharing or partitioning, which increases throughput by allowing independent subsets of transactions to be validated simultaneously (Mazlan et al., 2020). The accounting relevance of partitioning is substantial because natural partitions exist by entity, cost center, product line, or geography, and these partitions can be aligned with reporting segments and consolidation logic. However, partitioning must also preserve accuracy for cross-partition transactions such as intercompany sales, transfer pricing adjustments, and shared service allocations. Therefore, optimization in distributed ledgers often involves careful design of cross-shard communication, atomic commitment protocols, and standardized transaction schemas that reduce ambiguity in posting logic. A quantitative introduction can position these techniques as design levers that may influence measurable accounting outcomes, such as reconciliation workload, timing differences, and exception rates. International significance comes from the scale and heterogeneity of multinational transaction flows and the need for consistent, verifiable records that can be shared across entities, auditors, and regulators under controlled access (Zia et al., 2020). By grounding the discussion in definitions of speed, finality, and accuracy—and by treating consensus and partitioning as measurable system variables—

the paper’s introduction establishes a rigorous basis for statistical testing without drawing conclusions or prescriptive implications.

Figure 2: Digital Ledger Optimization Framework



Data quality research in information systems emphasizes dimensions such as accuracy, completeness, timeliness, and consistency, all of which map naturally to accounting objectives and audit criteria {{Sandner, 2020 #13;Rauf, 2018 #128}}. Optimization techniques include automated validation rules at data entry, master data governance to reduce coding errors, standardized chart-of-accounts mappings, and anomaly detection routines that flag suspicious postings. In digital ledger contexts, these controls can be embedded as smart-contract constraints, rule-based validators, or workflow-driven approvals that enforce segregation of duties {{Haque, 2020 #161;Md Ashraful, 2020 #125}}. From a quantitative perspective, the effectiveness of such optimization can be measured using posting error rates, the frequency and value of correcting entries, the incidence of reconciliation breaks, and the rate of audit adjustments attributable to system processing faults rather than judgmental estimation {{Haque, 2021 #160;Jinnat, 2021 #21}}. International settings intensify these concerns because accounting policies must be applied consistently across entities, while local statutory requirements may require additional fields, document types, or timing rules {{Md Fokhrul, 2021 #126;Zaman, 2021 #127}}. Optimization thus includes the design of extensible schemas that can store jurisdiction-specific attributes without breaking global consolidation logic, along with parameterized validation rules that adapt to local requirements while preserving comparability. Reporting accuracy is also influenced by metadata and provenance, including who approved a transaction, what source documents were linked, what exchange rate was applied, and what transformation steps were executed. Provenance-aware ledger architectures support stronger auditability by enabling systematic tracing from report line items back to source events and approvals {{Shae, 2018 #14;Fahimul, 2022 #167;Hammad, 2022 #157}}. Quantitatively, provenance quality can be assessed through completeness of documentation links, the average trace depth needed to resolve exceptions, and the proportion of transactions with missing or inconsistent supporting data. Another optimization domain is reporting pipeline design, including whether reporting is generated from live operational stores, from replicated read models, or from analytically optimized data marts. Techniques such as materialized views, incremental aggregation, and change-data-capture reduce

reporting latency while preserving correctness if reconciliation checks and deterministic transformations are maintained. International significance is reflected in the widespread adoption of structured reporting standards and regulatory data submissions that require both speed (timely filings) and accuracy (validated, consistent tagging). When reporting pipelines rely on manual spreadsheets or ad hoc transformations, the risk of transcription errors and inconsistent mappings increases, and close cycles lengthen. Ledger optimization therefore intersects with reporting automation through standardized data formats, validation taxonomies, and integrated audit trails, making it possible to evaluate system performance and reporting quality using comparable quantitative metrics across organizations (Marbough et al., 2020). An extended introduction can thus synthesize AIS theory, data quality dimensions, and ledger engineering principles to define constructs and motivate measurement without offering conclusions, implications, or trend-based claims.

Security and privacy requirements also shape optimization strategies because transaction speed and reporting accuracy must be maintained under integrity constraints, access controls, and confidentiality obligations. In accounting systems, confidentiality applies to payroll, pricing, vendor terms, and sensitive contracts, while integrity requirements apply to journal entries, approvals, and period-end adjustments (Park, 2019 #16; Javed Hasan, 2022 #147; Md Harun-Or-Rashid, 2022 #166). Optimization techniques that reduce latency—such as aggressive caching, relaxed validation, or reduced logging—must be evaluated against control objectives such as nonrepudiation, segregation of duties, and reliable audit trails. Digital ledger environments add cryptographic techniques that can strengthen integrity while introducing computational overhead. Optimization therefore often involves selective cryptography, hardware acceleration, or the use of succinct proofs to reduce verification costs. Permissioned ledger designs can optimize speed by limiting validator sets and using efficient replication protocols, while employing fine-grained access control to ensure that each participant sees only authorized data. Privacy-preserving techniques can support cross-entity verification in international settings without revealing commercially sensitive details, enabling reconciliations and confirmations that are evidence-rich for auditors. From a quantitative viewpoint, this domain can be operationalized through measures such as verification time per transaction, audit confirmation cycle time, the rate of access-control violations detected, and the computational overhead associated with encryption or proof validation. Reporting accuracy connects to security because unauthorized changes, incomplete logs, or weak identity controls can lead to misstated figures and impaired auditability. Optimization here includes stronger identity and key management, structured approval workflows, and immutable logging of control events, which can be evaluated through system log completeness and the incidence of unauthorized posting attempts (Zhang, 2020 #17; Md. Arifur, 2022 #159; Md. Towhidul, 2022 #115). International significance is amplified by cross-border data transfer regulations, differing retention requirements, and varying audit oversight regimes. Accounting systems must often retain transaction evidence for prescribed periods, support e-discovery, and provide verifiable records during regulatory examinations. Optimization techniques for retention—such as tiered storage, compression, and cryptographic commitment schemes—aim to reduce storage costs and retrieval times while preserving evidentiary integrity (Ratul, 2022 #169; Rifat, 2022 #149). Another internationally relevant factor is interoperability: multinational firms frequently operate heterogeneous systems and need reliable integration between subledgers, banking platforms, and consolidation tools. Optimization may involve standardized APIs, canonical data models, and reconciliation protocols that reduce integration latency and error propagation. Quantitative research can examine how such integration optimizations relate to measurable reporting outcomes, such as consolidation timeliness, mismatch frequency across subsystems, and the rate of manual adjustments required at reporting deadlines (Sheng, 2020 #18; Rifat, 2022 #138).

A quantitative study on digital ledger optimization in accounting systems requires clear construct specification, measurable variables, and defensible evaluation designs. Transaction speed can be measured using throughput and latency statistics collected from system logs, benchmark transactions, or controlled experiments. Reporting accuracy can be measured through reconciliation error rates, the frequency of correcting entries, audit adjustment counts, exception flags generated by validation rules, and the alignment of ledger outputs with authoritative source records (Pundir et al., 2019). Additional dependent measures can include close-cycle duration, time-to-report generation, and variance in

reported balances before and after reconciliation routines. Independent variables can be drawn from optimization techniques across layers: protocol parameters (batch size, validator count, block interval, quorum thresholds), data-layer choices (index design, partitioning scheme, append-only logging, compression), and application-layer controls (validation strictness, workflow steps, automated matching algorithms, and schema standardization). In international contexts, control variables may include transaction volume, number of legal entities, currency count, regulatory reporting complexity, and the degree of system heterogeneity. A robust quantitative framing also requires attention to measurement validity: system-reported timestamps need synchronized clocks; transaction definitions must be consistent across environments; and accuracy measures must distinguish between system processing errors and legitimate accounting adjustments arising from managerial judgment. Statistical methods may include regression modeling, panel data analysis, quasi-experimental comparisons between optimized and baseline configurations, or controlled A/B testing in simulated ledger environments (Elangovan et al., 2020). Reliability improves when data collection procedures are standardized, outliers are handled transparently, and performance metrics are reported with distributions rather than single-point averages. Reporting accuracy studies similarly benefit from clear definitions of what constitutes an error, a correction, an exception, or an adjustment, and from consistent mapping between ledger events and reporting artifacts. International significance supports multi-site or multi-jurisdiction samples that reflect varied regulatory and operational conditions, while maintaining harmonized measurement instruments. In addition, the introduction can articulate the accounting rationale for selecting speed and accuracy as joint outcomes by connecting them to AIS quality criteria, auditability, and the operational realities of high-volume transaction environments such as e-commerce, remittances, procurement networks, and shared service centers. Framed this way, digital ledger optimization techniques become empirically testable design interventions whose effects on transaction speed and reporting accuracy can be evaluated quantitatively across contexts. This establishes a research-ready foundation grounded in definitions, measurable constructs, and globally relevant accounting system needs, while remaining within an introductory scope and avoiding concluding statements, implications, trend assertions, or future-oriented claims (Silvano & Marcelino, 2020).

The primary objective of this quantitative study is to systematically measure and analyze the extent to which digital ledger optimization techniques improve transaction speed and reporting accuracy within accounting systems. Transaction speed is defined as the measurable time required for an accounting event to progress from initial capture to a finalized ledger state that is available for reporting and reconciliation, while reporting accuracy is defined as the degree to which ledger-generated financial outputs correctly, completely, and consistently represent underlying economic events. To achieve this objective, the study focuses on empirically testing optimization techniques applied at multiple layers of the digital ledger environment, including data-layer mechanisms such as indexing structures, partitioning strategies, append-only logging, and compression; process-layer mechanisms such as batching configurations, parallel posting routines, automated validation workflows, and real-time synchronization methods; and control-layer mechanisms such as standardized transaction schemas, deterministic validation rules, structured audit trails, and embedded reconciliation logic. The study seeks to quantify the statistical relationship between these optimization techniques and observable performance outcomes, including posting latency, transaction throughput, frequency of journal corrections, reconciliation discrepancies between subledgers and the general ledger, and error incidence in automated financial reports. An additional objective is to evaluate whether improvements in transaction speed are achieved without degradation in reporting accuracy, thereby examining the balance between efficiency and reliability in optimized ledger designs. The study further aims to control for organizational and operational variables such as transaction volume, system complexity, number of reporting entities, and frequency of reporting cycles to ensure that observed effects can be attributed to optimization techniques rather than contextual differences. By applying consistent measurement criteria across multiple ledger configurations or organizational settings, the objective is to generate empirically comparable results that reflect the operational realities of modern accounting systems.

## **LITERATURE REVIEW**

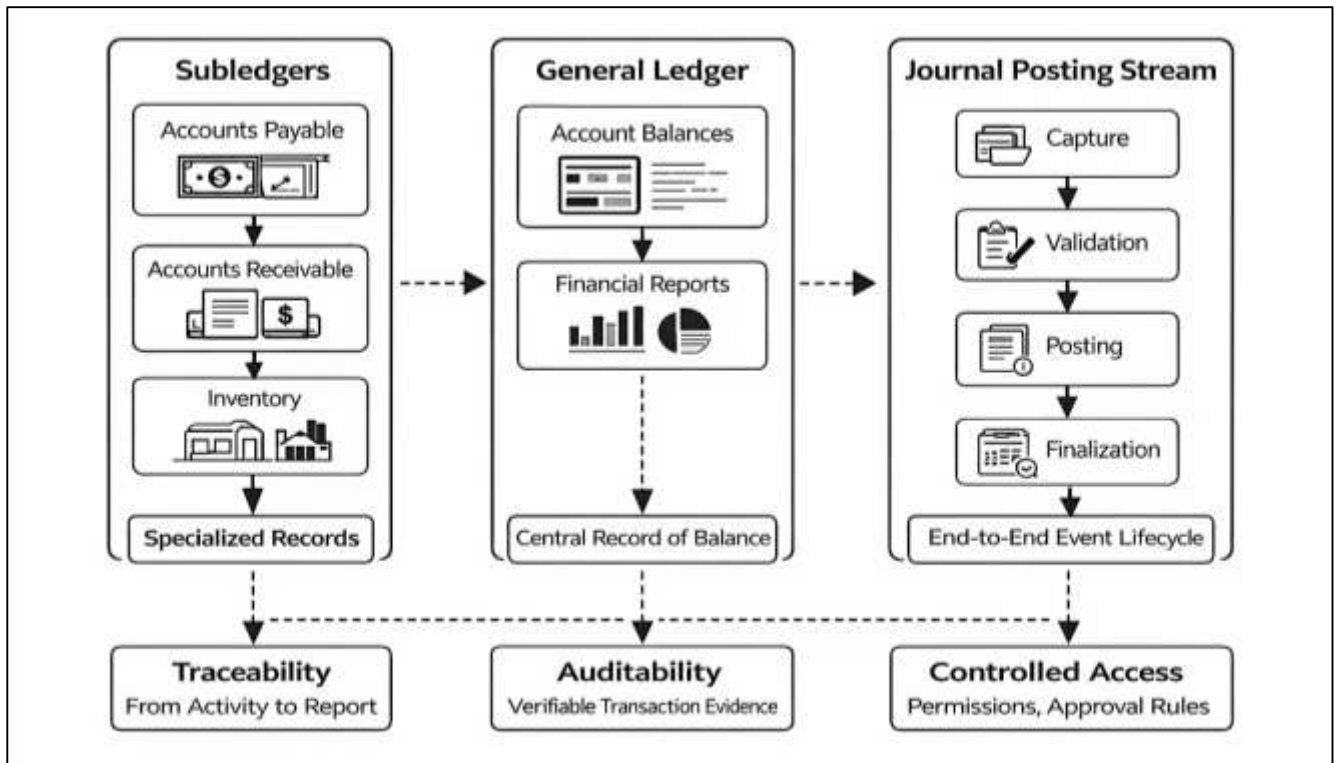
The literature review for a quantitative study on digital ledger optimization techniques for enhancing transaction speed and reporting accuracy in accounting systems synthesizes prior research that explains how ledger architectures, data-processing designs, and control mechanisms influence measurable performance and reporting outcomes (Kornmesser, 2016). In accounting information systems, digital ledgers function as structured repositories of economic events that must support high transaction volumes while preserving auditability, traceability, and consistency for financial reporting. Because transaction speed and reporting accuracy are operational outcomes that can be quantified through system logs, reconciliation statistics, and reporting error indicators, the literature relevant to this study spans multiple domains: accounting information systems, database and distributed systems performance, data quality and governance, internal control and auditability, and reporting pipeline engineering (Grüne-Yanoff, 2016). This section reviews empirical and analytical studies that define and measure transaction processing efficiency (e.g., throughput and posting latency), explain factors that drive reporting errors (e.g., incomplete metadata, inconsistent classification, reconciliation breaks), and evaluate optimization techniques such as batching, indexing, partitioning, parallel posting, validation rules, and audit-trail enrichment. The review also consolidates measurement approaches used across studies to ensure comparability of key dependent variables, including transaction speed metrics and accuracy metrics derived from exception rates, correcting entries, and reconciliation mismatches. By organizing prior work into clearly defined constructs and technique categories, the literature review establishes the theoretical and empirical foundation needed to support hypothesis development, variable operationalization, and statistical modeling in the present study, while maintaining a focus on quantifiable relationships between optimization mechanisms and accounting system outcomes (Sarkar & Searcy, 2016).

### **Digital ledgers in accounting systems**

Digital ledgers in accounting systems are commonly discussed in the literature as structured, electronic records that store economic events in a manner designed to preserve traceability, auditability, and controlled access. Within this stream of research, distinctions are drawn between the digital ledger as the broader recording environment and the general ledger as the central book of final account balances used for financial reporting (Junius-Walker et al., 2018). Subledgers are typically described as specialized transaction repositories – such as accounts payable, accounts receivable, inventory, fixed assets, and payroll – that capture operational detail before periodic or continuous summarization into the general ledger. A journal posting stream is often characterized as the ordered flow of validated entries moving from source documents and operational modules into ledger tables, where each entry is enriched with account codes, entity identifiers, dates, currency attributes, and approval markers. Across many empirical and design-oriented studies in accounting information systems, the ledger is treated as both a data structure and a control mechanism: it is a database artifact that must support high-volume recording, and it is also an assurance artifact that must support verification and review. The event lifecycle described in the literature usually begins with capture, where transactions originate from point-of-sale systems, procurement workflows, banking feeds, or internal allocations. Next is validation, which includes checks for authorization, completeness, master data consistency, and rule compliance. Posting then records the transaction in the relevant subledger and/or general ledger, followed by finalization in which the transaction becomes stable for reporting, consolidation, and audit routines (Jack & Thompson, 2017). Reporting extraction is described as the stage where ledger data are transformed into trial balances, schedules, regulatory filings, and management dashboards, often through standardized queries, aggregation logic, and mapping rules. When the literature frames “ledger optimization,” it typically establishes boundaries that include system design choices (data models, storage structures, access controls), data processing configurations (batching, parallel posting, integration pipelines), and control rules (validation, reconciliation logic, audit trails). This boundary is important because optimization is not limited to hardware performance; it encompasses the socio-technical design of how accounting data are captured, structured, verified, and surfaced for reporting. In quantitative studies, this framing supports measurable comparisons across alternative ledger configurations by treating optimization as a set of manipulable features that can affect both processing performance and the reliability of reporting outputs, while maintaining the ledger’s fundamental role

as a controlled record of economic events (Booth et al., 2018).

**Figure 3: Accounting Digital Ledger Architecture Framework**



A substantial body of quantitative work approaches transaction speed as a performance outcome that can be defined and measured consistently across different accounting platforms and ledger architectures (D'Angelo & D'Angelo, 2018). Transaction speed is frequently discussed as a multidimensional construct composed of how long it takes to process an event end-to-end and how many events can be processed within a given time window, especially when systems face operational peaks such as month-end billing, payroll cycles, promotional sales surges, or high-frequency microtransactions. In this literature, end-to-end posting time is treated as the interval between event initiation and the moment the entry becomes available for downstream uses such as reconciliation and reporting. Finality or confirmation time is discussed as the time until the system treats the posting as sufficiently stable to be relied upon for financial outputs, including consolidation routines and audit evidence gathering. Throughput is treated as the volume of transactions processed per unit time under defined workloads, and it is typically measured under both normal and stress conditions. Queue time and wait time under load are emphasized because accounting pipelines often include multiple stages – API ingestion, middleware routing, validation services, approval workflows, database writes, replication, and reporting synchronization – and delays at any stage can propagate (Shute et al., 2017). Many quantitative studies also interpret peak-load degradation as the percentage decline in performance as workload intensity increases, with particular attention to whether the system exhibits graceful degradation or sharp collapse in throughput and latency. The methodological discussion across studies highlights that the credibility of speed measurement depends on reliable instrumentation and consistent definitions of start and end points in the transaction lifecycle. As a result, measurement sources commonly include application logs that capture timestamps at each processing stage, database traces that record transaction commit times and lock contention, and middleware traces that reveal integration delays across services. Benchmark harnesses are also widely used in performance-oriented studies because they can generate controlled workloads, isolate system variables, and produce comparable datasets across configurations. Quantitative measurement approaches often emphasize the need to capture distributions rather than single averages, since latency commonly contains long-tail behavior under contention. Studies also discuss the importance of

aligning measurement with accounting relevance: speed improvements are evaluated not only as technical gains but as reductions in posting delays that affect reconciliation timing, report freshness, and the operational cadence of close processes (Harris et al., 2018). This research stream therefore treats transaction speed as an observable, log-derived outcome suitable for statistical modeling, enabling comparisons of optimization settings such as batching size, parallel posting degree, indexing strategies, and validation pipeline structure without reducing the concept to a single number or a purely technical benchmark.

Reporting accuracy is treated in the accounting information systems literature as an information-quality outcome that reflects how faithfully ledger-derived reports represent the underlying set of economic events and accounting rules applied to them. Quantitative studies commonly decompose accuracy into dimensions that are meaningful for both system evaluation and audit practice, including correctness, completeness, consistency, and verifiability (Aspers & Corte, 2019). Correctness refers to whether postings contain valid account codes, amounts, dates, currencies, and classifications as determined by policy and system rules. Completeness concerns whether all relevant events have been captured and posted, including adjustments and reversals, and whether supporting attributes needed for reporting and compliance are present. Consistency refers to alignment across subsystems—for example, whether subledger totals reconcile to the general ledger, whether intercompany postings match between counterparties, and whether reporting extracts apply uniform mapping logic. Verifiability focuses on whether postings and outputs can be traced to source documents, approvals, and transformation steps in a manner that supports audit review. The quantitative indicators emphasized across studies include error rates in journal entries, often measured as detected posting defects per unit of transaction volume; the frequency and magnitude of correction entries, which signal either process weaknesses or data capture inconsistencies; reconciliation mismatch rates between subledgers and the general ledger, which are widely used as objective markers of reporting reliability; and exception rates produced by validation rules, which provide a systematic count of anomalies flagged by automated controls. Another recurring indicator is report variance before and after reconciliation or validation routines, which can be expressed as the percentage difference between preliminary and finalized totals for key accounts or statements (Timans et al., 2019). A critical theme in the literature is the distinction between system-driven inaccuracies and policy- or judgment-driven adjustments. System-driven inaccuracies are associated with issues such as integration failures, duplicated entries, incomplete metadata, incorrect mapping rules, timing misalignments, and weak validation controls. Policy- or judgment-driven adjustments arise from legitimate accounting estimation and decisions, such as accruals, impairment judgments, or changes in classification guided by evolving interpretations and internal policies. Quantitative studies often emphasize this separation because it affects measurement validity: a high volume of adjustments is not necessarily evidence of system failure unless the adjustments are linked to data-processing defects. The literature also highlights that verifiability and audit trail richness are not merely compliance features; they support accuracy measurement by improving the ability to detect, diagnose, and resolve discrepancies (Hughes, 2020). Accordingly, reporting accuracy is commonly treated as a set of measurable outcomes derived from exception logs, reconciliation reports, correction entry records, and audit adjustment statistics, making it suitable for inferential testing alongside transaction speed measures within a single quantitative framework.

Across many studies, transaction speed and reporting accuracy are not treated as isolated objectives but as jointly affected outcomes that must be evaluated together when assessing digital ledger optimization techniques. This joint-outcome perspective appears in research that frames ledger design as a multi-criteria problem where performance improvements are meaningful only if the ledger continues to produce reliable and auditable reporting outputs (Ribeiro & Barbosa-Povoa, 2018). The literature notes that speed can be improved through design choices that reduce processing overhead—such as larger batches, fewer synchronous checks, higher degrees of parallel processing, and aggressive caching—yet these choices can alter the timing and stability of postings, complicate reconciliation, or increase the risk of temporary inconsistencies if not accompanied by strong controls. Conversely, accuracy can be improved through stronger validation rules, richer provenance metadata, more comprehensive reconciliation checks, and tighter workflow approvals, yet these measures add

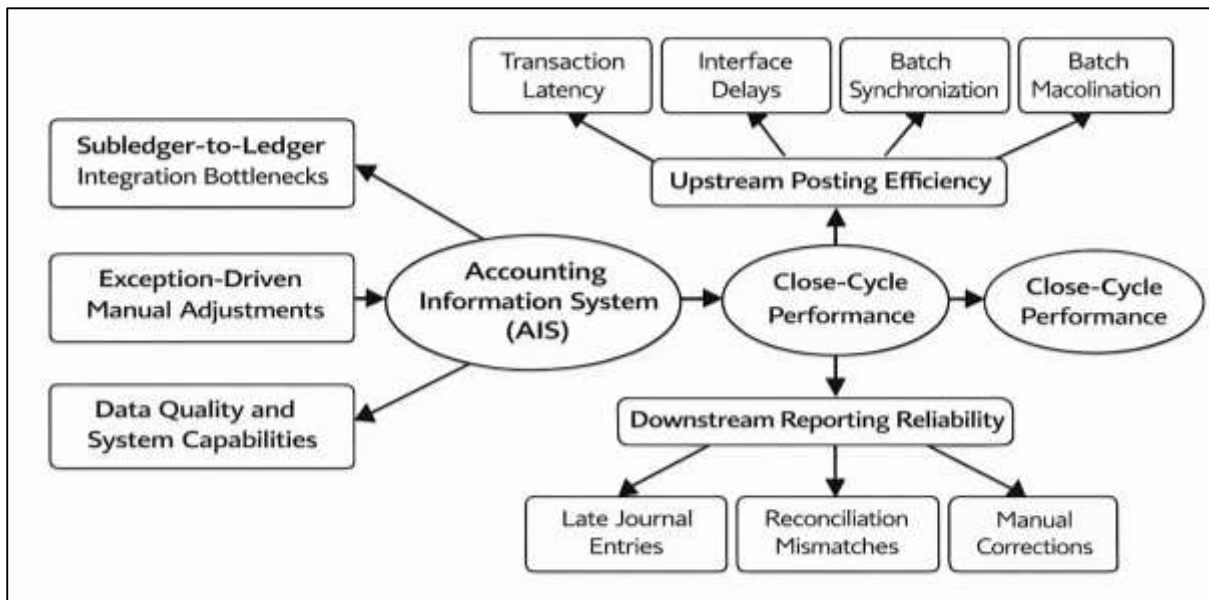
computational and procedural steps that may increase latency or reduce throughput under load. As a result, many quantitative discussions emphasize trade-off variables that shape both outcomes. Validation strictness is frequently discussed as a determinant of exception rates and correction needs, while also affecting processing time because more checks require more computation and more opportunities for routing to manual review (Finch et al., 2018). Control complexity, including multi-step approvals and segregation-of-duties enforcement, influences the stability and auditability of postings but can introduce queueing delays. Cryptographic integrity measures, proof verification, and secure logging practices are discussed as mechanisms that strengthen tamper evidence and verifiability, while adding overhead that can affect processing speed depending on implementation. Batching lag is also treated as a key trade-off: batching can raise throughput and reduce per-transaction overhead, yet it can increase the time between capture and posting availability, affecting report freshness and the responsiveness of reconciliation processes. The literature-review synthesis commonly resolves these tensions by recommending that empirical evaluation use paired metrics—measuring speed outcomes and accuracy outcomes concurrently—so that optimization techniques are assessed on their net effect within auditable accounting workflows. Quantitative studies often model these relationships using variables that represent optimization intensity across data processing and control layers, then estimate associations with both performance indicators and accuracy indicators while controlling for contextual complexity such as transaction volume, number of modules, integration breadth, and entity structure. Within this integrated view, the ledger is conceptualized as both a processing engine and an assurance mechanism; optimization is therefore evaluated as a coordinated set of design and control decisions that collectively influence how quickly transactions become usable for reporting and how reliably reports reflect the underlying transactions (Walter & Suina, 2019). This synthesis supports a literature-grounded justification for studying speed and accuracy together, using objective measures derived from logs, reconciliation outputs, and correction records, without reducing the evaluation to a single technical benchmark or a single accounting quality indicator.

#### **Accounting Information Systems Performance Literature**

Research on accounting information systems (AIS) efficiency has consistently treated the period-close cycle as an operational outcome that reflects both the technical performance of transaction processing and the organizational capacity to convert recorded events into verified financial statements. Across empirical studies of close-cycle duration, the month-end and year-end close are described as multi-stage processes involving transaction cut-off, subledger balancing, general ledger posting validation, intercompany reconciliation, consolidation, and reporting pack assembly (Appelbaum et al., 2017). This literature commonly identifies drivers of close duration that can be quantified and compared across firms, such as transaction volume, number of entities, degree of automation, integration complexity, frequency of manual journal entries, and the prevalence of exceptions that require investigation. Several studies emphasize that close-cycle length is not only a function of accounting policy complexity but also a reflection of processing latency across the capture-to-posting pipeline, because delays in upstream modules create downstream uncertainty and increase the volume of late postings and rework. Empirical findings reported across multiple industries show that close-cycle inefficiencies tend to concentrate around exception handling and reconciliation workloads, where unresolved mismatches between subsystems force accountants to conduct manual investigation, request operational clarifications, and create adjusting entries. The literature also links close-cycle compression efforts to the standardization of transaction codes, improved master data governance, and stronger automation of routine controls (Li et al., 2018). Within AIS efficiency research, a recurring theme is that the measurable reliability of upstream posting workflows influences how rapidly finance teams can transition from data gathering to analytical review. This point is reinforced by studies that examine how incomplete documentation, inconsistent metadata, and delayed approvals increase the proportion of transactions requiring manual triage. Another strand of empirical work highlights the effects of segmented close ownership models, where subsidiaries or business units produce local closes that then feed group consolidation; this structure introduces measurable dependencies that amplify delays when local ledgers do not finalize on time. Across these studies, the close cycle is therefore treated as a quantitative performance lens that captures both system throughput and information readiness, allowing researchers to relate time-to-close outcomes to system characteristics such as automation,

integration maturity, and the stability of ledger postings. The broader implication in this literature synthesis is that AIS efficiency is most consistently improved when systems reduce the frequency and severity of exceptions that shift work from automated processing to manual investigation, since exception-driven rework and late adjustments are repeatedly reported as dominant contributors to longer close durations (Kokina & Blanchette, 2019).

Figure 4: AIS Close-Cycle Efficiency Framework



A closely related empirical stream focuses on how processing delays in AIS environments are associated with manual adjustments and correction activity, treating these adjustments as observable indicators of friction in the accounting data pipeline (Pan & Seow, 2016). Multiple studies describe manual adjustments as a measurable response to gaps between operational reality and what the ledger currently reflects at the moment of close. These gaps arise when transactions are captured late, validated inconsistently, posted with missing attributes, or integrated from external systems on delayed schedules. In such cases, accountants compensate by posting accruals, reclassifications, and correction entries to align ledger balances with expected outcomes, and the volume and timing of these postings can be used as quantitative markers of system strain. Empirical investigations often distinguish routine adjusting entries that reflect normal accounting practice from corrective entries that reverse errors, repair mis postings, or resolve mismatches created by integration failures. Within that distinction, the literature repeatedly notes that the concentration of corrective entries near period-end tends to coincide with pipeline bottlenecks and exception backlogs. Studies examining reconciliation workloads show that when subledger-to-general-ledger alignment is unstable, teams engage in iterative cycles of investigation, communication, and posting corrections, which extends the close and reduces confidence in preliminary reports (Wang & Kogan, 2018). Evidence across ERP-centric environments also suggests that manual adjustments rise when transaction coding practices vary across teams or when master data changes are not synchronized, because inconsistent classifications lead to mismatched rollups and unexpected variances. Another consistent observation is that manual adjustments increase when upstream operational systems generate transactions with ambiguous or incomplete references, such as missing purchase order links, inconsistent vendor identifiers, or poorly structured tax attributes, which later require manual enrichment or reclassification. Some studies adopt statistical approaches that relate adjustment frequency to measures of integration complexity and automation maturity, reporting that higher levels of standardized workflows and automated matching are associated with lower corrective entry rates and fewer late-stage exceptions. In synthesis, this literature treats manual adjustment volume, correction entry frequency, and reconciliation mismatch rates as quantifiable “symptoms” of processing delays and information quality weaknesses. The

research also emphasizes that the cost of these adjustments is not only the labor required to post them but the added time to validate them, document the rationale, and ensure consistent mapping into financial reports (Huang & Vasarhelyi, 2019). As a result, AIS efficiency research frequently positions the reduction of exception-driven manual adjustments as a measurable pathway to improved close performance, because fewer late corrections imply that the ledger is more stable earlier in the close cycle, enabling earlier review and faster finalization of reports.

A substantial body of ERP-focused research examines ledger posting and subledger integration as core determinants of both processing efficiency and accounting reliability, with particular attention to bottlenecks that can be observed and measured in large-scale enterprise environments (Salahshour Rad et al., 2018). Across studies of ERP transaction pipelines, locking and concurrency constraints are described as prominent sources of posting delays, especially when many users or automated jobs attempt to update the same ledger tables, account balances, or document chains simultaneously. Sequential posting designs—where journal entries are validated and committed in strict order—are often reported as limiting throughput during peak periods such as billing runs, payroll processing, and inventory close. Interface delays also appear frequently in the literature, particularly where subledgers exchange data with the general ledger through middleware, file transfers, or scheduled jobs; in these cases, latency is shaped by queue backlogs, transformation overhead, and error-handling routines that pause transfers until exceptions are resolved. ETL batch constraints are repeatedly identified as another bottleneck category, especially when reporting environments depend on batch extraction and transformation into data warehouses or reporting marts; these schedules can create time gaps between operational postings and report availability, which increases the likelihood of late adjustments and reconciliation discrepancies. Quantitative outcomes reported in ERP performance studies often include measures such as average and percentile posting times, backlog sizes during peak windows, job completion durations, frequency of failed interface loads, and the rate of posting reversals triggered by validation failures (Roetzel, 2019). Several studies also examine how configuration choices—such as batch sizes, commit frequency, indexing strategies, and partitioning of ledger tables by entity or period—affect both posting performance and the stability of reconciliation results. The literature further emphasizes that integration bottlenecks are not purely technical; they are compounded by inconsistent master data governance and differing local practices across modules and business units, which produce measurable increases in exception rates and manual correction activity. Many ERP studies therefore treat the subledger-to-general-ledger bridge as a high-impact area for optimization, because it is the point where operational detail is summarized into formal accounting structure, and any latency or mismatch at this bridge propagates into close-cycle delays and reporting uncertainty. Synthesizing these findings, ERP ledger posting research presents a consistent picture: throughput limitations, interface delays, and batch synchronization constraints generate observable performance degradation under load, and these technical delays are strongly intertwined with accounting workloads associated with exception resolution, reconciliation, and corrective postings (Sledgianowski et al., 2017).

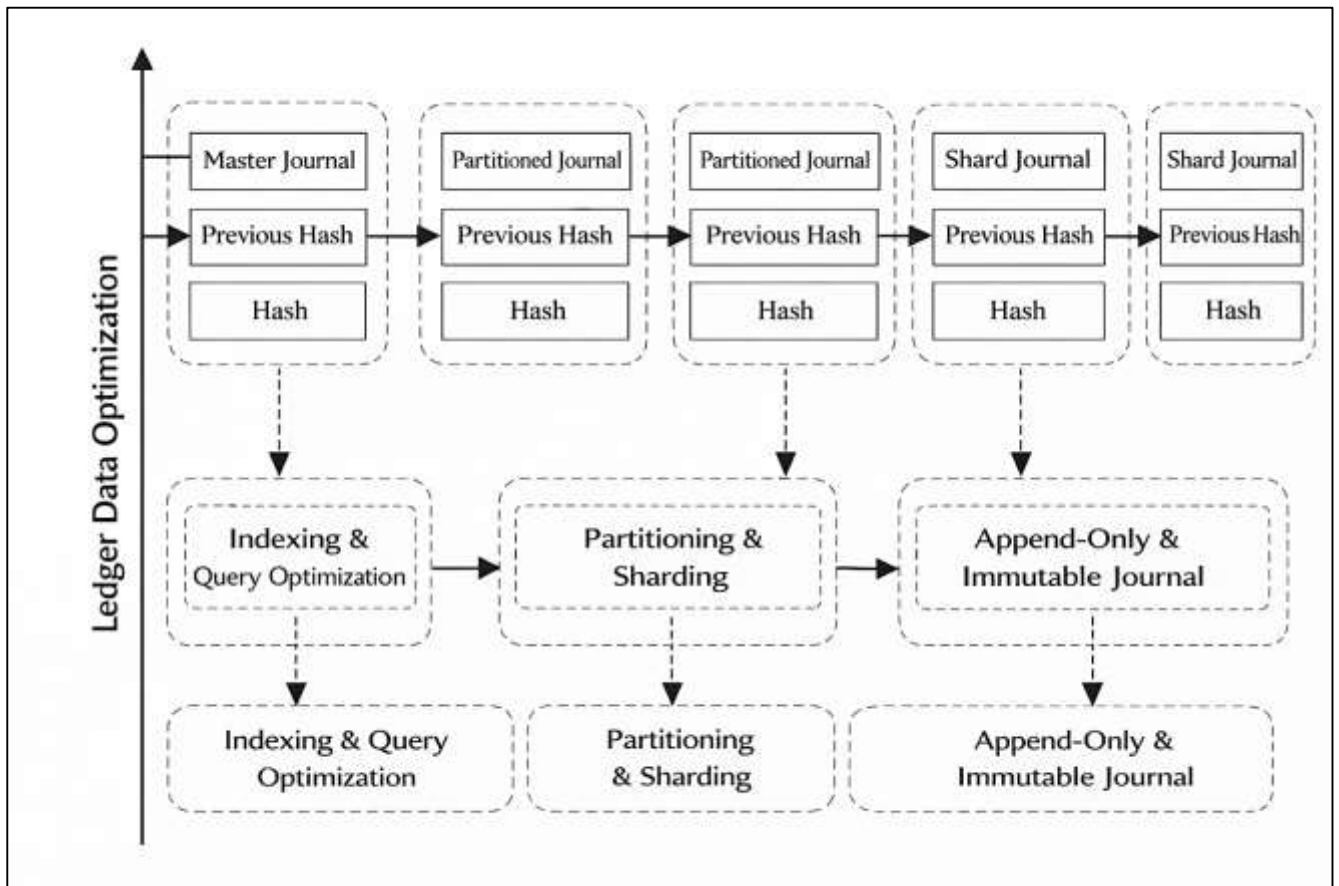
Information systems success research and data quality models provide widely used quantitative lenses for interpreting AIS performance outcomes, because they offer structured constructs for linking system characteristics to the quality of accounting outputs. Across many studies, data quality dimensions—such as accuracy, completeness, timeliness, consistency, and accessibility—are operationalized as measurable predictors of reporting reliability and user trust in financial information (Nguyen et al., 2017). In AIS contexts, these dimensions are frequently mapped to observable artifacts, including the frequency of validation exceptions, the proportion of transactions with missing attributes, the stability of subledger-to-general-ledger reconciliations, and the timeliness of report readiness relative to period-end. System quality constructs, which include reliability, responsiveness, integration capability, and usability, are similarly translated into measurable indicators such as system uptime during close, average response times for posting and queries, interface failure rates, and the number of manual workarounds required to complete routine accounting tasks. A recurring modeling approach in this literature links system quality to information quality and then to organizational outcomes such as decision usefulness, process efficiency, and compliance readiness. Within accounting systems, this chain is often evidenced by findings that faster, more stable posting pipelines and better integration are

associated with fewer exceptions and more consistent reporting outputs, while poor integration and weak data governance are associated with higher mismatch rates, more correcting entries, and longer close cycles (Chege & Wang, 2020). Studies using survey-based and archival approaches alike highlight that perceived reporting reliability often aligns with objective indicators of data quality and control effectiveness, such as reconciliation success rates and audit trail completeness. Importantly, data quality models allow AIS performance research to avoid treating speed as the only success metric; instead, they support multi-dimensional evaluation in which timeliness and correctness are assessed together through measurable proxies. This perspective aligns closely with ERP performance findings that emphasize the role of standardized data definitions, consistent coding structures, and automated validation in reducing downstream rework. Across the literature, reporting outcomes are framed as the culmination of system behavior and data quality practices, meaning that the same system improvements that reduce posting delays can also reduce error propagation if they enhance validation, consistency, and traceability. Synthesizing this body of work, information systems success and data quality research provide a coherent quantitative rationale for examining AIS efficiency, integration bottlenecks, and reporting reliability within a single model: system capabilities shape data quality characteristics, and data quality characteristics shape the measurable accuracy and readiness of accounting reports, alongside the operational performance reflected in close-cycle duration and the volume of manual adjustments (Tarafdar & Qrunfleh, 2017).

### **Ledger Data Architecture Optimization Techniques**

Indexing and query optimization for accounting workloads is widely discussed in the data-layer literature as a foundational technique for improving the responsiveness of digital ledgers without altering accounting semantics. Across studies of accounting information systems and enterprise databases, researchers consistently note that ledger data are queried in predictable patterns that differ from many general transactional applications (Shen et al., 2020). Common accounting workloads include account rollups across hierarchical charts of accounts, time-bounded searches aligned to fiscal periods, and entity-based filtering for subsidiaries, cost centers, projects, and business units. Because these workloads often combine aggregation, grouping, and sorting over large journal tables, studies emphasize that index selection and query plan design strongly influence how quickly reports can be generated and how efficiently users can drill down from summarized balances to line-level evidence. A recurring finding in this literature is that account rollup queries benefit from indexes aligned to account identifiers and posting dates, while period-driven reporting benefits from composite indexing strategies that support both temporal filtering and entity segmentation. Many studies also highlight the importance of indexing strategies for foreign keys that connect journals to source documents, subledger references, and dimension tables, because report generation and audit traceability often require joining across multiple tables at scale. Researchers additionally note that query optimization is closely tied to schema design choices, including whether accounting dimensions are stored in normalized tables or denormalized structures designed to reduce join cost (Zichichi et al., 2020). In empirical evaluations, improvements are typically demonstrated through reductions in query wait time and faster completion of standard report routines such as trial balances, aging reports, and variance analyses. The literature also describes how optimization can be evaluated using repeatable workloads that emulate real accounting usage, such as repeated period-close report runs and frequent ad hoc inquiries during reconciliation. Another common theme is that improvements are not limited to faster query execution; they also include reduced computational load on the system during peak close periods, which can indirectly stabilize posting performance by decreasing contention between reporting queries and transaction writes. Studies examining modern ledger environments often integrate indexing with mechanisms such as materialized aggregates, cached rollups, and optimized execution plans that reduce repetitive scanning of large journal tables. Overall, this body of work positions indexing and query optimization as a measurable lever that affects the timeliness of reporting extraction and the efficiency of investigative workflows, especially when accounting teams rely on rapid drill-down to resolve anomalies and confirm balances under deadline constraints (Paik et al., 2019).

Figure 5: Ledger Data Optimization Framework



Partitioning and sharing strategies appear throughout the data architecture literature as techniques for scaling ledger performance by distributing data across logical or physical segments, often aligned to accounting realities (Carrozzo et al., 2020). A consistent theme across studies is that accounting ledgers naturally support segmentation because transactions are commonly associated with discrete entities, business units, fiscal periods, and transaction categories. Researchers frequently describe partitioning by entity or business unit as a way to localize reads and writes, reduce contention, and accelerate routine reporting that is executed at the entity level before consolidation. Partitioning by fiscal period is also discussed as a common approach because financial reporting and audit procedures are period-based; separating journal data by time window can make period-close extraction and reconciliations more efficient by limiting the data scanned for each cycle. Transaction-type partitioning appears in studies that emphasize divergent access patterns, such as high-frequency operational postings versus less frequent adjusting entries, where isolating data classes can improve concurrency and maintenance operations. Empirical work in this area often reports that throughput and response time improve when partitions are designed to reflect real workload boundaries, particularly in high-volume environments where a single monolithic journal table becomes a bottleneck (Yang et al., 2020). At the same time, the literature underscores that partitioning introduces cross-partition challenges that are especially salient in accounting. Intercompany transactions, shared services allocations, and cross-entity settlements require coordinated posting logic, consistent references, and reliable reconciliation between partitions. Studies describe these cross-partition interactions as sources of overhead because the system must either coordinate transactions across segments or reconcile them after posting, and both approaches impose measurable costs in complexity and processing time. Researchers also highlight the accounting consequences of partition misalignment, such as mismatches in intercompany balances, timing differences in postings between counterparties, and increased exception handling workload. Quantitative evaluations often emphasize not only the improvement in processing capacity but also the additional coordination cost introduced by cross-partition workflows, which can be observed in increased integration steps, longer reconciliation routines, and higher rates of mismatch investigation

when the design is not harmonized with accounting processes. Another line of research describes sharing, where ledger data are distributed across nodes or servers, as an extension of partitioning that enables horizontal scalability. In ledger contexts, sharing is typically evaluated by examining how well the system sustains performance under increasing transaction volume and how reliably it supports consolidated reporting across shards (Rahman et al., 2020). Across these studies, partitioning and sharing are framed as optimization techniques that must be assessed through joint metrics: improvements in processing and reporting speed are meaningful when reconciliation stability and cross-entity consistency remain strong within the segmented architecture.

Append-only logs and immutable journal models are frequently discussed as data-layer designs that enhance auditability and record integrity while supporting performance characteristics suited to high-volume accounting environments. In many studies, append-only design is described as a natural fit for accounting because journal entry logic is fundamentally additive: transactions are recorded as entries that represent events, and corrections are typically handled through additional entries rather than destructive edits (Zhang et al., 2018). Researchers often emphasize that append-only structures align with audit expectations because they preserve historical states and make it easier to demonstrate how balances evolved over time. Immutable journal models are also discussed as mechanisms that reduce the risk of unauthorized alteration and improve traceability, since each new entry can be linked to prior states through logging sequences, hash-linked records, or structured provenance metadata. The literature highlights performance implications as well: append-only writes can be efficient because they reduce the complexity of in-place updates and can be optimized through sequential storage patterns, which are well-supported by modern storage engines. Empirical evaluations frequently examine write responsiveness, log ingestion stability under load, and the efficiency of retrieving a consistent view for reporting and reconciliation. A common theme is that immutability shifts the operational focus from editing records to managing corrections and adjustments as first-class events, which can improve transparency but requires robust logic for handling reversals, reclassifications, and late postings. Studies also note that integrity verification becomes an important measurable component in immutable designs, because audit confidence depends on the ability to confirm that records have not been altered and that the chain of custody is preserved (Gleim et al., 2020). In practice-oriented research, integrity verification is evaluated by the time required to validate record sequences, check transaction lineage, or confirm that summaries tie to underlying log entries. Another strand of literature focuses on how append-only models support reproducibility for audits and dispute resolution, as investigators can replay event sequences, reconstruct balances at specific timestamps, and verify approvals and supporting documentation embedded in the ledger. Researchers also connect immutable journal models to operational resilience, since append-only logs can support robust recovery patterns and reduce the risk of data corruption impacting finalized records. Quantitative outcomes in this literature often include measures of how quickly new entries can be persisted, how reliably correction workflows are executed, and how efficiently the system supports both investigative drill-down and high-level reporting extraction (Fan et al., 2020). Across these studies, append-only and immutable approaches are framed as optimizations that combine performance advantages in write patterns with strengthened auditability, with evaluation focused on observable timing, stability, and correction-handling behaviors rather than on conceptual claims alone.

#### **Transaction Processing Optimization Techniques**

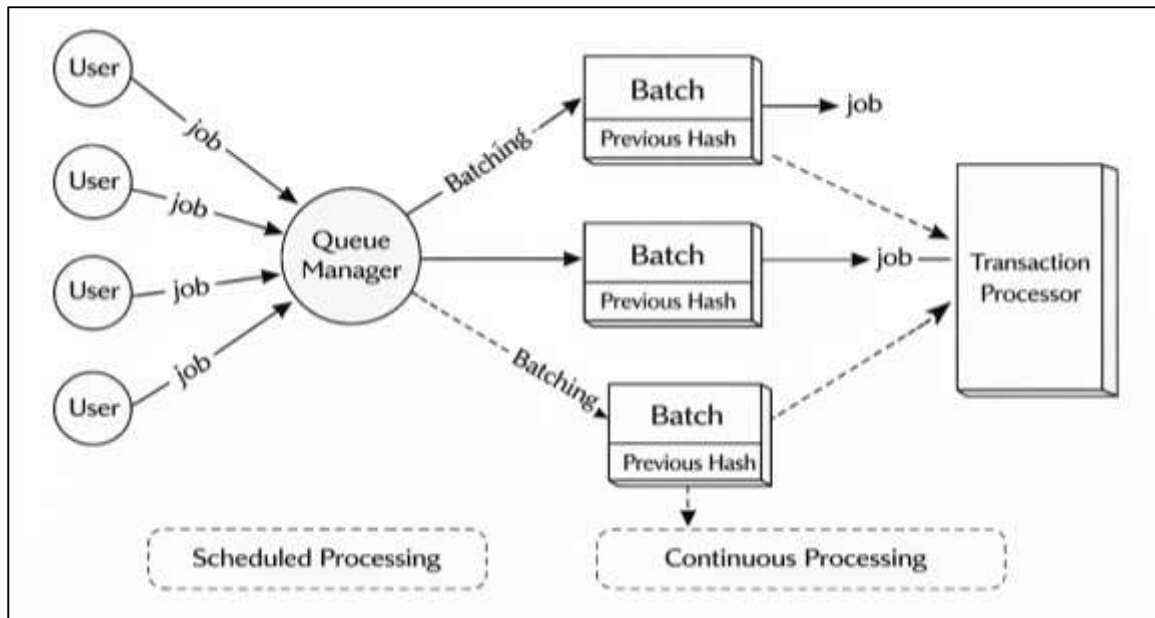
Batching versus real-time posting configurations are repeatedly discussed in the process-layer literature as a central design choice that shapes how quickly accounting events become ledger-ready and how predictably systems behave under peak loads. Studies describing batching emphasize that grouping transactions into scheduled posting runs can reduce per-transaction overhead by amortizing validation, authorization checks, and database commit costs across many entries (Luo et al., 2018). This approach is frequently linked to stable throughput during high-volume periods because queue scheduling can be tuned to prioritize critical posting streams, separate high-frequency operational entries from lower-frequency adjustments, and allocate resources based on workload class. At the same time, research comparing batch and near-real-time approaches highlights that batching introduces a timeliness gap between capture and ledger availability, which can influence operational decision-making, reconciliation pacing, and the cadence of close activities. In empirical investigations, batch size

and batch interval are typically treated as controllable configuration factors that influence posting capacity and the distribution of processing delays, especially when workloads exhibit spikes near cut-off times. Many studies report that larger batches tend to support higher sustained throughput under constrained compute resources, while smaller batches or more frequent runs reduce the time transactions wait before posting, producing fresher interim reports and earlier visibility of exceptions. The literature also notes that queue management practices shape outcomes as much as batch sizing: if scheduling policies prioritize certain transaction types, the system can maintain timeliness for material postings while deferring low-priority entries, though this can increase later reconciliation workload if deferred items affect rollups (Cai et al., 2016). Research further discusses hybrid approaches used in enterprise accounting, where routine transactions flow in near-real time during normal operations, and controlled batch runs execute for intensive activities such as allocations, depreciation, and consolidations. Across studies, the evaluation of batch versus real-time posting commonly uses log-based measures of waiting time, processing time, and the proportion of transactions posted within defined service windows, alongside accounting-facing indicators such as the timing of exception discovery and the number of late postings requiring additional review. A synthesized conclusion across this literature is that batching is not presented solely as a legacy constraint but as a deliberate optimization technique when system designers aim to reduce contention, manage peak-load variability, and coordinate posting with downstream reporting extracts. Real-time posting, by contrast, is described as operationally attractive for timeliness and early error detection, yet it requires robust validation pipelines and resource provisioning to prevent performance instability when transaction volumes surge (Fei et al., 2019). These studies collectively frame batching configuration as a quantitative lever that can be tuned to achieve predictable processing capacity while maintaining acceptable reporting readiness within the accounting calendar.

Parallel posting and concurrency control appear as another major process-layer theme because accounting systems must often handle simultaneous transaction streams across modules, business units, and integration endpoints without compromising ledger integrity (Liu et al., 2017). The literature describes parallel validation and posting designs as methods for increasing processing capacity by distributing workload across threads, services, or compute nodes, enabling multiple transactions to be validated and committed concurrently. In high-volume settings, studies frequently attribute posting slowdowns to serialized routines that force transactions to wait behind others, particularly when entries contend for the same ledger resources such as account balances, document chains, or shared reference tables. Parallelism is therefore discussed as an optimization that can reduce backlogs and improve responsiveness when implemented with careful coordination. However, empirical and technical research consistently emphasizes concurrency risks that can degrade data reliability if control mechanisms are weak. Double posting, race conditions, and inconsistent intermediate states are highlighted as potential failure modes when multiple processes attempt to apply updates to related records without adequate locking discipline or idempotency safeguards (Tolpygo et al., 2019). Deadlocks are discussed as a common operational symptom of poorly designed locking orders, where two or more posting processes hold mutually blocking locks and require intervention or timeout-based rollbacks. As a result, studies evaluating parallel posting often examine not only posting speed but also stability measures such as deadlock frequency, rollback rates, and the incidence of retried transactions, since frequent retries can erase throughput gains and introduce complexity for audit trails and reconciliation. Another recurring issue in the literature is the tension between strong isolation for correctness and relaxed isolation for performance, particularly in ledger environments where the correctness of account totals and the traceability of journal sequences are essential. Researchers describe design strategies such as partitioning posting streams by entity or account range, enforcing deterministic ordering for entries affecting the same accounts, and using idempotent transaction identifiers to prevent duplicates across retries. Workflow design is also treated as part of concurrency control: when approvals or validations occur asynchronously, systems must ensure that only authorized, validated transactions reach the commit stage, and that failed validations do not leave partial effects. Quantitative assessments reported across studies typically rely on system logs that track lock waits, commit times, conflict rates, and the proportion of transactions requiring rollback or manual remediation, along with accounting measures such as the number of corrections attributable to posting

conflicts. The synthesized view across this literature is that parallel posting is most effective when concurrency control is treated as an integrated control framework rather than a purely technical setting, since posting integrity must remain stable under load for reporting and audit reliance (Lim et al., 2020). Consequently, the process-layer research positions parallelism as a high-impact optimization technique whose benefits are evaluated through paired metrics capturing both increased processing capacity and reduced conflict-induced instability.

Figure 6: Batching vs Real-Time Posting



Event streaming and change data capture pipelines are widely discussed as process-layer mechanisms that shift ledger architectures toward continuous data movement, enabling faster synchronization between operational systems, the ledger core, and reporting environments. Studies in this area commonly describe event-driven designs as a means to improve timeliness by propagating transactional changes immediately as they occur, rather than waiting for scheduled batch extracts (Vahdati et al., 2018). Event streaming is presented as especially relevant for accounting ecosystems that integrate multiple source systems, such as sales platforms, procurement tools, banking interfaces, and inventory applications, because it provides a structured method to capture changes as an ordered stream of events. Change data capture, in particular, is described as a technique that monitors data modifications at the source and publishes them to downstream consumers, including ledger posting services and reporting replicas, thereby reducing the delay between operational updates and ledger visibility. The literature emphasizes that these pipelines can also improve observability by producing traceable event logs that capture what changed, when it changed, and how it was transformed, supporting both troubleshooting and audit inquiries. At the same time, studies report that event-driven pipelines introduce specific risks that must be quantified and managed. Propagation delay is treated as a key measure because delays can occur at multiple points: message brokers may queue events, validation services may throttle, and downstream writers may lag during peak periods (Saghiri et al., 2019). Duplication rate is another common metric, reflecting the reality that distributed message delivery can produce repeated events unless idempotent handling is enforced. Researchers often document how duplication can create reconciliation anomalies if duplicate postings are not prevented, or if deduplication logic is inconsistent across systems. Reconciliation mismatch rates are therefore used as an accounting-facing indicator of pipeline quality, capturing whether streamed updates remain consistent across subledgers, the general ledger, and reporting stores. Studies also highlight ordering challenges, noting that out-of-sequence event consumption can temporarily produce inconsistent balances or misaligned cut-offs if dependency rules are not maintained. In practice-oriented work, design responses include enforcing ordering keys aligned to accounting dimensions, implementing

exactly-once or effectively-once posting semantics through idempotency keys, and using checkpointing mechanisms to ensure that replays do not distort balances. Quantitative evaluations in the literature often combine technical measures such as lag time distributions and replay count with accounting measures such as exception rates and correction entry frequency associated with integration issues. Across these studies, event streaming and change data capture are framed as optimizations that can reduce the time between transaction capture and reporting availability while improving traceability, yet their success is judged by whether they maintain stable, reconcilable records across the system landscape (Zhang & Jiang, 2019). The literature synthesis therefore treats streaming pipelines as measurable process-layer interventions whose quality can be assessed by combining propagation metrics with reconciliation stability indicators.

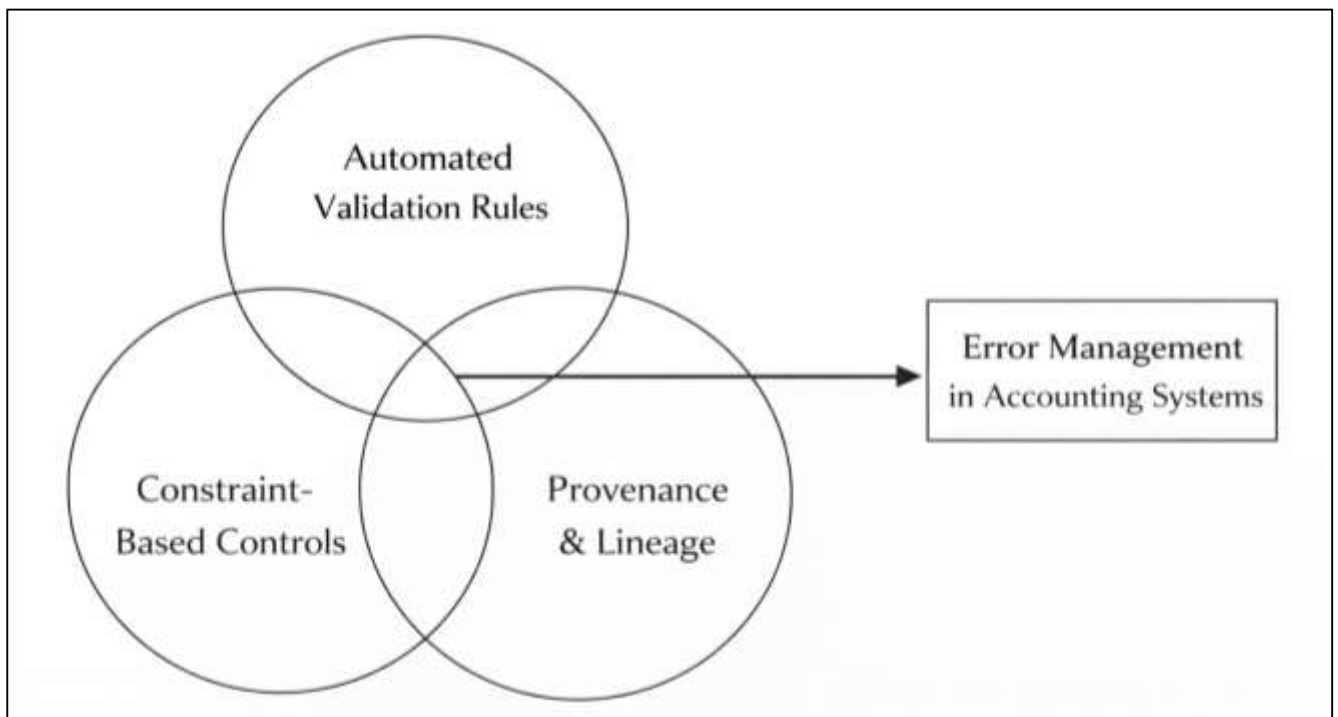
Caching and read-model replication for reporting are discussed throughout the process-layer literature as techniques that separate the needs of high-speed transaction posting from the needs of fast, repeatable reporting queries (Kamiński, 2019). Many studies describe a common tension in accounting systems: the same ledger store that must accept continuous writes is often expected to serve intensive read workloads during reconciliations and close, including repeated report runs, drill-down analysis, and ad hoc variance investigations. Caching is presented as a technique that reduces repeated computation and data retrieval by storing frequently requested results, such as account rollups, trial balance snapshots, and commonly filtered views used in dashboards. Read-model replication is similarly discussed as the creation of reporting-optimized copies of ledger data, sometimes described as read replicas, materialized views, or separate analytical stores populated from the ledger through replication pipelines. The literature notes that these techniques can significantly reduce report generation time, improve responsiveness for finance users during close, and reduce contention that would otherwise slow posting operations. However, studies also emphasize measurement of staleness, describing the time gap between the authoritative ledger state and what the reporting model shows. This staleness window is treated as an important governance concern because accounting decisions and reconciliations rely on the currency of information. Research frequently frames the evaluation problem as maintaining an acceptable balance between performance and freshness: more aggressive caching improves speed but can increase the risk that users act on outdated figures if invalidation is not timely (Chua et al., 2017). Consistency variance is also discussed as a measurable indicator, capturing whether replicated read models produce the same totals as the ledger for the same period and entity, particularly when replication is asynchronous and subject to delays or partial failures. Many studies document that inconsistency can arise from missing events during replication, schema mismatches between source and replica, or transformation rules that diverge over time, and they highlight that such inconsistencies increase reconciliation workload and diminish trust in reports. Design-focused research discusses mitigation approaches such as incremental refresh, change-based invalidation, checkpointed replication, and reconciliation checks that compare replica totals to ledger totals at defined intervals. Empirical evaluations often rely on reporting system logs, refresh statistics, and reconciliation records to quantify how quickly replicas update, how often caches are invalidated, and how frequently discrepancies occur between the reporting layer and the ledger. In synthesis, the literature frames caching and read-model replication as process-layer optimizations that improve reporting responsiveness and reduce operational friction during close, while requiring disciplined monitoring and control practices to keep staleness and inconsistency within tolerable bounds (Sengupta et al., 2018). The combined view across studies is that these techniques contribute to transaction speed indirectly by reducing read-write contention and to reporting accuracy indirectly by enabling repeatable, stable reporting extraction routines, provided that replication integrity and cache governance are maintained through measurable quality checks.

#### **Validation and Auditability Optimization**

Automated validation rules and constraint-based controls are consistently positioned in the control-layer literature as a primary mechanism for improving reporting reliability by preventing defective transactions from entering the ledger in the first place. Studies of accounting information systems describe validation as a structured set of checks applied at multiple points in the transaction lifecycle, including initial data capture, pre-posting verification, and post-posting monitoring. Input validation commonly covers field completeness, data type integrity, referential consistency, and formatting rules

for identifiers such as vendor codes, invoice numbers, tax fields, and currency values (Mackintosh et al., 2019). Master data checks are frequently discussed as crucial because many reporting errors originate from inconsistent or outdated reference data, including chart-of-accounts mappings, cost center hierarchies, tax codes, product classifications, and counterparty identifiers. Authorization logic is treated as a control that enforces who can initiate, approve, and post transactions, thereby reducing unauthorized entries and supporting audit requirements for controlled processing. The literature emphasizes that the value of automated validation is measurable through exception statistics that capture how often transactions are flagged, rejected, or routed to remediation workflows. Exception rate is described as a direct indicator of both rule coverage and upstream data quality, as high exception rates can signal either strong detection of anomalies or underlying process issues that require operational correction (Münsing et al., 2017). A second measurement theme concerns the diagnostic quality of anomaly detection, where false positives represent legitimate transactions incorrectly flagged and false negatives represent defective transactions that pass undetected. Control-layer studies frequently treat these two rates as central to evaluating the net effectiveness of validation, because excessive false positives increase manual workload, slow processing, and can lead to rule fatigue where users bypass controls, while false negatives allow reporting errors to propagate into reconciliations and financial statements. Researchers also discuss the importance of rule design transparency and maintainability, noting that validation logic must remain aligned with evolving business rules and policy interpretations to avoid generating noise. As a result, validation is often described not simply as a technical filter but as an operational control system that shapes how quickly transactions progress, how often exceptions require investigation, and how stable ledger outputs remain during close. Synthesizing these findings, the literature treats automated validation as a quantitatively assessable optimization, because its impact can be observed through measurable changes in exception volumes, rework cycles, and correction entry frequency, while also shaping the trade-off between strictness and processing efficiency in high-volume accounting environments (Yu et al., 2020).

**Figure 7: Accounting Control Layers Integration Framework**



Provenance metadata and transaction lineage are widely discussed in the auditability literature as key design features that strengthen the evidentiary quality of digital ledgers and reduce reconciliation friction by making transactions easier to interpret, verify, and trace. Within this stream of research, provenance refers to the structured information that explains where a transaction originated, how it was processed, who approved it, what source documents support it, and what transformations were

applied before it reached the ledger (Shu et al., 2016). Lineage expands this idea by describing the chain of relationships between source events, intermediate states, and final ledger postings, enabling an accountant or auditor to reconstruct the story behind a reported balance. Studies emphasize that reconciliation breaks often persist not because the amounts are complex, but because the supporting context is fragmented across systems, emails, spreadsheets, and inconsistent references. Strong lineage reduces this fragmentation by providing standardized links to documents such as purchase orders, goods receipts, invoices, bank confirmations, payroll registers, and intercompany agreements, along with identifiers that allow cross-system alignment. The literature repeatedly highlights that provenance can be measured through documentation completeness, defined as the proportion of transactions that contain the required references, attachments, approval IDs, and transformation markers needed for verification. Documentation completeness is not treated as an abstract quality; it is used as a practical indicator of how quickly teams can resolve exceptions and how confidently they can finalize balances. Trace time per exception is another frequently discussed metric, capturing the elapsed time required to locate relevant evidence, confirm the legitimacy of a transaction, and determine whether corrective action is necessary (Kaur et al., 2016). Researchers often describe trace time as a strong proxy for both control efficiency and reporting readiness, because long trace times amplify close-cycle delays and increase the probability of late adjustments. The literature also distinguishes between basic provenance (simple source system and timestamp fields) and richer provenance (multi-step workflow histories, transformation logs, and tamper-evident trails), noting that richer provenance supports faster root-cause identification and reduces repeated investigative work across cycles. Another recurring theme is that provenance improves consistency by standardizing how transactions are described, which supports aggregation and reporting extraction while reducing ambiguity in classifications and mappings. In synthesis, provenance and lineage are treated as measurable auditability optimizations that influence reconciliation stability, investigator efficiency, and the reliability of ledger-derived reporting, because they reduce information gaps that otherwise lead to mismatches, manual clarifications, and corrective postings during the reporting cycle (Yin et al., 2020).

#### **Distributed Ledger and Multi-Party Accounting Context**

Distributed ledger and multi-party accounting research commonly frame consensus and finality as central determinants of whether ledger records can be treated as stable evidence for period-end cut-off and close activities (Betti et al., 2020). In this literature, consensus is described as the rule-governed process by which multiple participants agree on the validity and ordering of transactions, while finality is described as the point at which a transaction is sufficiently stable to be relied upon for reporting and audit. Studies examining accounting cut-off emphasize that ordering and timing are not purely technical features, because financial reporting depends on the accurate assignment of transactions to fiscal periods, subsidiaries, and reporting segments. When transaction ordering is uncertain or when confirmation is delayed, close teams face operational ambiguity regarding which events should be recognized in a given period, which increases the likelihood of late adjustments and post-close corrections. Research that compares distributed ledger designs frequently treats confirmation time as a measurable indicator of how quickly transactions reach a stable state, with particular attention to variability under load and network conditions (Gao et al., 2019). This attention to variability reflects the recognition that average confirmation time can obscure tail behavior that matters during month-end peaks. Additional studies discuss stability as a probabilistic concept in some ledger environments, where a transaction becomes increasingly unlikely to be reversed as time passes or as additional confirmations accrue. In such contexts, reversal events are treated as operationally significant for accounting because reversals can produce restatements of interim totals, trigger reconciliation breaks between parties, and generate correction entries to restore alignment across subledgers and consolidated statements. The literature also notes that multi-party environments elevate the importance of consistent timestamping and standardized event definitions, because participants may operate across different time zones, local calendars, and operational systems. As a result, studies consistently highlight that a ledger's approach to transaction ordering and finality influences the practical reliability of cut-off procedures, not only in terms of whether transactions are recorded, but also in terms of whether the ledger's record can serve as a dependable basis for period-close reporting. This stream of research therefore positions consensus and finality as measurable system characteristics

that intersect directly with accounting requirements for cut-off integrity, reconciliation stability, and report readiness (Zhong et al., 2019).

**Figure 8: Distributed Ledger Governance Measurement Framework**

LAYER	CONCEPTS AND MEASURES
Consensus & Finality	<ul style="list-style-type: none"> <li>• Confirmation time, variability, reversal occurrence</li> <li>• Cut-off integrity, probabilistic finality, timestamp consistency</li> </ul>
Governance & Access Control	<ul style="list-style-type: none"> <li>• Role-based visibility, cross-entity validation</li> <li>• Verification overhead, privacy-driven computation cost</li> </ul>
Off-Ledger Anchoring	<ul style="list-style-type: none"> <li>• Throughput improvement, audit verification time</li> <li>• Evidence completeness, proof retrievability</li> </ul>
Reconciliation Effects	<ul style="list-style-type: none"> <li>• Reconciliation frequency, duration of breaks</li> <li>• Transaction stabilization, manual adjustment rate</li> </ul>

Permissioned ledger governance and access control are treated in the literature as defining features that shape how multi-party accounting records can be shared, validated, and relied upon while maintaining confidentiality and role-based restrictions (Battah et al., 2020). Governance is commonly described as the set of rules that determine who can participate, who can validate, who can view which records, and how disputes, corrections, and policy updates are administered. In accounting settings, studies emphasize that data sharing is rarely uniform across all participants because transaction evidence can contain sensitive commercial terms, payroll data, pricing arrangements, and contractual information. Accordingly, role-based visibility is discussed as an operational necessity, enabling participants to validate and reconcile shared transactions without exposing unrelated or confidential details. Research in this area often frames access control as a determinant of both auditability and performance, because stronger privacy protections can introduce additional computational steps for verification, encryption, or selective disclosure. Verification overhead is therefore frequently discussed as a measurable cost, reflecting how much additional processing is required for participants or validators to confirm that transactions meet rules and to confirm that disclosed data are sufficient for reconciliation (Cao et al., 2020). Privacy-driven computation costs are similarly described as measurable, especially when systems must generate proofs or perform filtering logic to provide different views of the same underlying transaction to different roles. Studies also highlight that governance design influences cross-entity validation practices, including how counterparties confirm intercompany postings, settlement confirmations, and shared service allocations. Cross-entity validation is treated as a core accounting need because multi-party transactions require bilateral or multilateral agreement to avoid mismatches that later appear in consolidations (Raman et al., 2019). The literature often describes the governance challenge of balancing flexibility and control: overly rigid governance can slow onboarding and increase administrative friction, while weak governance can reduce trust in the ledger and increase audit burden. Another recurring point is that permissioned environments allow tailored validation rules that reflect the business context, such as requiring specific approvals for high-value postings, enforcing standardized identifiers for counterparties, and mandating document attachments for certain transaction types. These features are frequently linked to auditability because they create structured

evidence that supports verification (Madine et al., 2020). In synthesis, governance and access control research treats role-based visibility and cross-entity validation as accounting-aligned features that affect both processing efficiency and the reliability of shared records, with performance evaluated through overhead measures and accuracy evaluated through the stability of reconciliations across participants.

Off-ledger processing with on-ledger anchoring is widely discussed as an architectural approach that attempts to preserve shared integrity while reducing performance bottlenecks in multi-party environments. In this literature, off-ledger processing refers to keeping detailed transactional data, operational workflows, and high-frequency posting activity within local systems or specialized processing layers, while using the shared ledger to store commitments, summaries, or cryptographic anchors that attest to the existence and integrity of the underlying records (Roman & Vu, 2018). Studies describe this approach as particularly relevant when detailed accounting transactions are too voluminous, sensitive, or complex to place directly into a shared multi-party ledger while still meeting the need for verifiable evidence across parties. Anchoring is commonly discussed as a method of creating tamper-evident commitments, enabling participants to demonstrate that a set of transactions existed in a specific state at a specific time. In accounting contexts, the literature highlights that this can support audit needs by enabling verification that local records used for reporting were not altered after cut-off, while allowing parties to keep detailed data confidential. Researchers evaluate this approach using measurable indicators tied to both performance and assurance. Throughput improvement is discussed as a practical advantage because keeping high-frequency data off the shared ledger can reduce validation workload and network coordination demands (Lei et al., 2020). Audit verification time is treated as a measurable outcome because anchoring designs must still allow auditors or counterparties to validate that reported totals align to anchored evidence without requiring full disclosure of operational detail. Evidence completeness is another recurring metric concept, reflecting whether anchored commitments are accompanied by sufficient metadata, document references, and retrieval procedures to support reconciliation and audit testing. The literature also emphasizes that anchoring approaches require disciplined mapping between off-ledger records and on-ledger anchors; if the mapping is weak or inconsistent, verification becomes difficult and reconciliation exceptions increase. Studies frequently discuss the role of standardized identifiers, deterministic aggregation rules, and consistent timestamping in strengthening the link between local ledgers and shared proofs. Another common observation is that anchoring supports dispute resolution by providing a shared reference point for what each party had recorded at a given time, which can reduce negotiation cycles when mismatches arise (Dziembowski et al., 2019). Overall, this research stream treats off-ledger processing with on-ledger anchoring as a measurable architectural optimization that attempts to preserve shared trust and auditability while managing performance constraints and confidentiality requirements inherent in multi-party accounting ecosystems.

When the distributed ledger sub-stream is treated as comparative literature within accounting system research, the synthesis often focuses on how these multi-party designs change the practical mechanics of reconciliation, cut-off enforcement, and reporting extraction relative to conventional centralized architectures (Dziembowski et al., 2019). Studies commonly report that multi-party environments reframe reconciliation from an end-stage activity into a more continuous process because counterparties can validate shared records earlier, reducing the accumulation of mismatches at period close. However, the literature also describes that shared validation can introduce new types of operational friction when governance rules, access restrictions, or verification requirements slow the processing of transactions or complicate exception handling. As a result, many studies emphasize the value of using paired measurement perspectives that combine timing measures with stability and auditability measures. Confirmation and stabilization behaviors are discussed as time-related outcomes that shape how quickly postings can be treated as reliable for reporting, while reversal occurrences are treated as stability outcomes that can undermine confidence in interim totals and generate follow-up corrections. Governance and access control are discussed as determinants of how efficiently participants can validate and reconcile, because restricted visibility can require additional verification steps or supporting evidence exchanges (Paul & Shrivastava, 2019). Off-ledger anchoring is discussed as a strategy for balancing speed, confidentiality, and verifiability, evaluated by how

quickly proofs can be validated and how completely those proofs support audit procedures and reconciliation needs. Across this comparative literature, a consistent theme is that the accounting suitability of distributed ledger designs depends on whether they support predictable cut-off, stable ordering for reporting periods, and efficient evidence retrieval for audits. Researchers often describe that any shared ledger arrangement must still integrate with established accounting practices such as subledger aggregation, general ledger summarization, consolidation, and disclosure preparation. Therefore, distributed ledger studies are frequently interpreted through accounting outcomes that can be observed in operational records, including the frequency and duration of reconciliation breaks, the time required to validate counterpart transactions, the number of manual adjustments required to finalize period-end balances, and the responsiveness of evidence retrieval during audit testing (Pei et al., 2018). This comparative synthesis positions multi-party ledger designs not as replacements for accounting fundamentals, but as alternative coordination mechanisms whose effectiveness is judged through measurable impacts on transaction stabilization, cross-entity alignment, and the practical auditability of recorded events.

### **Quantitative Measurement Models and Research Gaps**

A consistent theme across quantitative studies of digital ledger performance in accounting information systems is the reliance on a shared set of dependent variables that capture processing efficiency, reporting accuracy, and auditability as observable outcomes rather than abstract qualities. Within the performance stream, researchers most often operationalize transaction speed using system-recorded timing and capacity indicators, including posting latency measured from event capture to ledger availability, confirmation or finalization time until an entry is stable for reporting, and sustained throughput measured under normal and peak workloads (Suárez et al., 2017). Many studies also report queue dynamics—such as time spent waiting in processing pipelines, backlog growth during heavy loads, and service-level attainment rates that indicate what proportion of transactions are posted within required windows. On the reporting side, accuracy is frequently measured through error and exception artifacts produced by real accounting operations: detected journal-entry defects, correction entry frequency and value, reconciliation mismatch rates between subledgers and the general ledger, and variance between preliminary reports and reconciled or finalized reports. Auditability is commonly represented through proxies that are quantifiable in operational records, such as the completeness of documentation links, presence of approval metadata, traceability coverage across transaction populations, and the time required to retrieve evidence for exceptions or audit sampling. Although studies vary in terminology, many converge on the practical logic that ledger optimization should be evaluated through “paired outcomes,” meaning that faster processing is interpreted alongside measures of correctness, stability, and evidentiary strength. In empirical work, this pairing is important because speed improvements can be achieved through architectural choices that reduce control overhead, while accuracy and auditability often improve when controls, validations, and provenance are strengthened (Delzendeh et al., 2017). The literature therefore frequently encourages consolidated measurement sets that capture speed metrics together with accuracy indicators and auditability proxies, because performance alone does not represent accounting system success if the resulting outputs cannot be reconciled or defended in audit contexts. Even where researchers focus on one outcome family, such as latency or error rates, many still report secondary measures that reflect downstream impacts, such as close-cycle duration, exception backlogs, and the frequency of late postings. Across a broad range of studies—covering ERP environments, integrated subledger architectures, and comparative designs that include distributed or multi-party validation—dependent variables are largely derived from logs, reconciliation reports, correction-entry registers, and audit evidence workflows. This empirical tendency strengthens measurement objectivity, but it also produces comparability challenges when studies adopt different time boundaries for latency, different definitions of “finality,” and different thresholds for classifying errors versus legitimate judgmental adjustments (Casella et al., 2020).

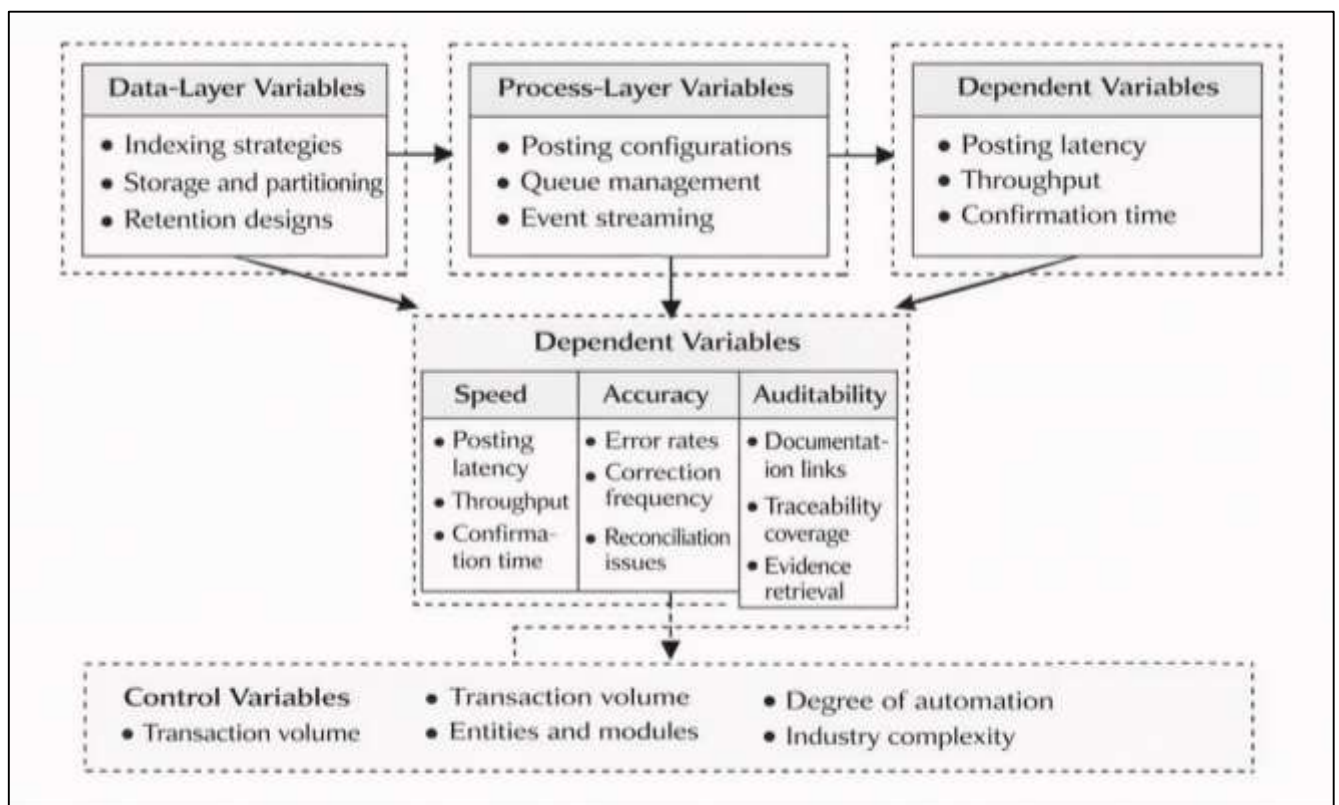
To interpret these dependent variables in a unified way, the literature increasingly organizes independent variables as structured classes of optimization techniques rather than isolated configuration changes (Ribeiro & Barbosa-Povoa, 2018). A common approach is to classify optimization levers into data-layer mechanisms, process-layer mechanisms, and control-layer mechanisms,

reflecting the multi-layer architecture of accounting ledgers and their reporting pipelines. Data-layer variables include indexing strategies, storage structures, partitioning choices, and retention designs that influence how quickly the ledger can write and retrieve records for reporting and audit. Process-layer variables include posting mode configurations, queue scheduling, concurrency handling, event streaming integration, and replication routines that influence pipeline responsiveness and the timeliness of reporting extracts. Control-layer variables include validation strictness, workflow routing, segregation of duties, automated matching, and provenance enrichment that influence error prevention, reconciliation stability, and evidence availability. Across many studies, this structured classification improves interpretability by allowing researchers to test whether certain “families” of techniques are systematically associated with improved throughput, reduced exception rates, or faster evidence retrieval. Building on this classification, some empirical work uses composite measures that summarize the intensity or maturity of optimization practices across layers, rather than treating each technique as independent (Feng et al., 2017). This line of thinking motivates the construction of an aggregated index that scores how extensively a system applies data-layer, process-layer, and control-layer optimizations, enabling statistical models to compare more optimized environments to less optimized ones using consistent scales. However, the literature also notes challenges in building such indices, because optimization intensity is not always directly observable and may be represented through proxies such as configuration flags, the presence of certain modules, the frequency of automated validations, or architectural documentation of replication pipelines. Studies also note that weighting decisions can influence results, because some optimizations have larger effects on speed while others have larger effects on accuracy and auditability. As a result, researchers often recommend transparency in how optimization categories are defined, how indicators are scored, and how the resulting composite measures relate to the real operational behaviors captured in logs and reconciliation reports. Another recurring observation is that optimization practices may be implemented unevenly across entities within the same organization, especially in multinational environments where subsidiaries vary in system maturity and workflow standardization (Hassan et al., 2016). This introduces within-firm variation that can be analytically useful for quantitative modeling, but it also requires careful definition of the unit of analysis, such as entity-level ledgers, module-level pipelines, or period-level close cycles. In synthesis, the literature positions layered classification of independent variables as a practical bridge between technical design features and accounting outcomes, supporting models that explain speed, accuracy, and auditability using interpretable categories grounded in ledger architecture.

Quantitative studies consistently highlight the importance of control variables and covariates because accounting environments differ substantially in scale, complexity, and operating conditions that can confound relationships between optimization techniques and observed outcomes. Transaction volume is widely treated as a dominant covariate because higher volumes intensify queueing, locking contention, and integration load, often increasing both processing delays and the probability of exception generation (Merom & John, 2019). The number of entities and modules is also frequently used as a complexity proxy, reflecting the fact that multi-entity environments require intercompany coordination, consolidation routines, and standardized mappings across operational systems, all of which can amplify reconciliation workload and increase the surface area for mismatches. Currency count appears regularly in studies that focus on multinational operations because foreign currency transactions introduce translation timing, rate selection logic, remeasurement routines, and additional metadata requirements that influence both processing steps and reporting validations. Reporting frequency and cadence are also treated as covariates, as environments with frequent interim reporting or daily close-like routines impose constant reporting extraction demands, affecting both performance and the visibility of discrepancies. Degree of automation is commonly used to control for the level of manual intervention in validations, matching, and approvals, since manual steps can extend cycle time but may also catch errors that automation misses, affecting observed accuracy indicators in complex ways. Industry transaction complexity is another recurrent covariate, capturing differences in transaction structures across sectors such as retail, manufacturing, logistics, financial services, and project-based industries, where the granularity of dimensions and the prevalence of multi-step workflows can change both processing load and error patterns (Ivanov et al., 2017). The literature also

identifies additional covariates that are often reported when available, including integration breadth, number of external interfaces, data governance maturity, and the proportion of transactions requiring supporting documents or approvals. These covariates matter because the same optimization technique can yield different observed effects under different complexity conditions; for example, a replication strategy that improves report generation time in a single-entity environment may exhibit weaker effects when consolidation dependencies dominate close timelines. Researchers also emphasize that some covariates interact with dependent variables in ways that complicate interpretation, such as when higher automation reduces manual adjustments but increases exception visibility through more systematic detection, resulting in higher recorded exception rates even as true error incidence declines. Another repeated point is that accounting policy differences, such as how organizations treat cut-off or document requirements, can influence measured outcomes like correction frequency and mismatch resolution time (Oraee et al., 2017). Therefore, the literature generally supports models that include a strong set of controls to isolate technique-specific effects, using consistent operational definitions and carefully selected covariates aligned to the accounting context. This synthesis reinforces the view that quantitative evaluation of ledger optimization is most credible when it models speed and accuracy as outcomes shaped by both system design and contextual complexity.

Figure 9: Ledger Optimization Variable Classification Framework



Methodologically, the quantitative literature reveals several recurring patterns in how researchers collect data, structure comparisons, and infer relationships, while also exposing gaps that limit cumulative knowledge across studies. A large proportion of empirical work relies on system log analytics, extracting time stamps, queue states, error events, and processing outcomes from application logs, database traces, and middleware telemetry (Li et al., 2020). This approach is valued because it supports objective, high-frequency measurement across large transaction populations, enabling distributional analysis of latency and robust estimation of exception patterns. Quasi-experimental designs are also common, including before-and-after comparisons when organizations implement an optimization change such as a new indexing strategy, a revised batch schedule, a parallel posting module, or automated matching enhancements. In these designs, researchers often compare outcome metrics across periods, controlling for volume and seasonality, and may incorporate difference-based

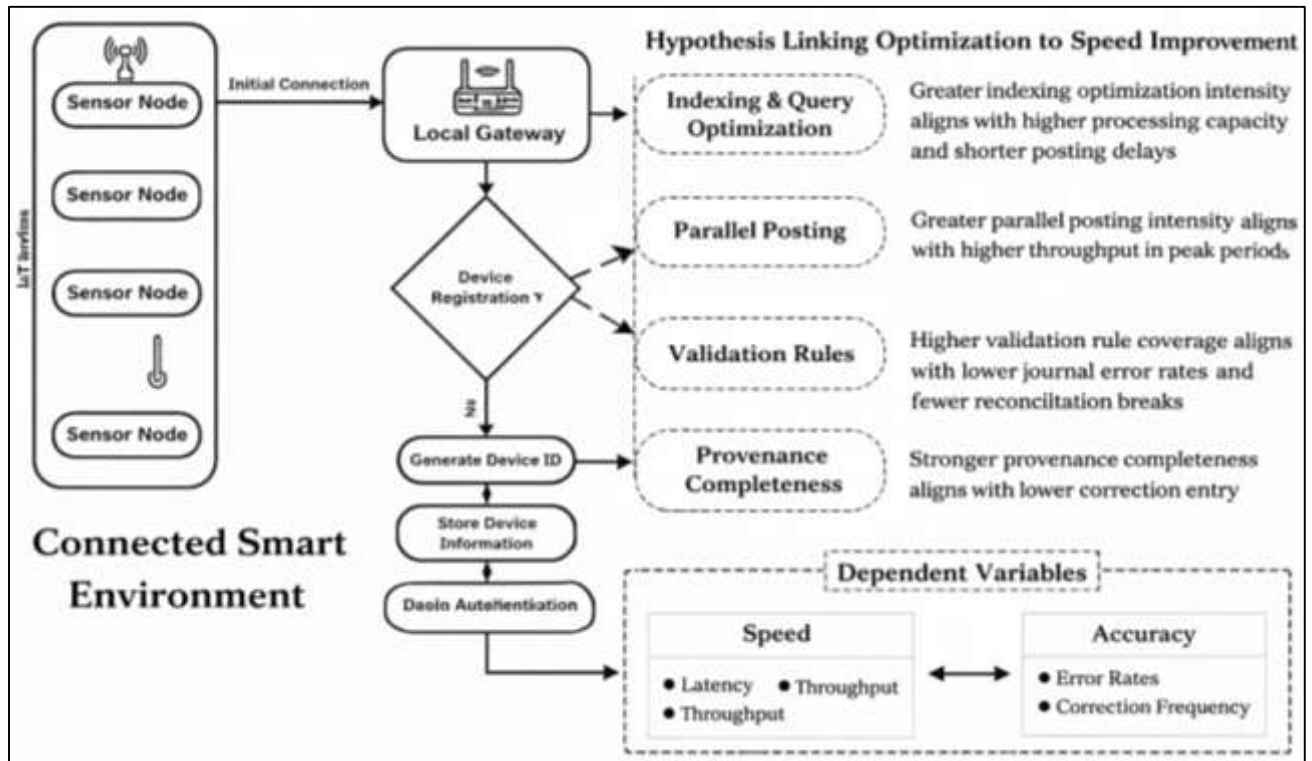
comparisons when multiple entities adopt changes at different times. Benchmark experiments appear frequently in technical-oriented studies, where controlled workloads are applied to compare configurations under standardized conditions, enabling clearer isolation of causal mechanisms at the cost of reduced ecological realism. Across these methods, a recurring weakness noted in the literature is the lack of unified metric definitions, especially around key constructs like end-to-end posting latency, finality, and reporting accuracy. Studies vary in how they define the start and end points of processing, whether they measure posting completion at the subledger or general ledger stage, and whether they treat reconciliation as part of the posting lifecycle or a downstream process. In reporting accuracy measurement, inconsistency arises in how errors are classified, how exceptions are counted, and whether correction entries reflect system defects or legitimate accounting judgment (Mahmud et al., 2017). Another widely discussed gap is limited joint outcome modeling, where many studies focus on speed alone or accuracy alone rather than modeling both outcomes together in a multi-outcome framework that reflects accounting reality. This limitation reduces the field's ability to explain when speed improvements coincide with stronger reporting reliability versus when speed gains accompany increased mismatches or correction activity. The literature also reports uneven reporting of covariates and contextual variables, making it difficult to compare results across industries or organizational structures. Additionally, many studies are constrained by access to sensitive financial-system logs and reconciliation artifacts, leading to smaller samples, limited cross-firm comparability, or reliance on proxy measures that do not fully capture auditability. As a result, research gaps cluster around standardization of constructs, transparent operational definitions, and integrated modeling approaches that link optimization techniques to both performance and reliability outcomes within the same analytical structure. These methodological observations form a coherent literature-based rationale for tighter construct definitions, harmonized measurement sets, and models that treat speed, accuracy, and auditability as connected quantitative outcomes rather than separate evaluation tracks (Li et al., 2020).

#### **Hypothesis Development Scaffold**

Hypothesis development in quantitative studies of digital ledger optimization commonly begins by translating repeated empirical regularities from accounting information systems, database performance research, and control automation studies into directional expectations about transaction speed (Weng et al., 2017). Across the performance literature, indexing and query optimization are frequently described as mechanisms that reduce computational burden on both reporting and operational workloads, particularly when ledger tables become large and access patterns are dominated by period-based filters and account rollups. Studies in enterprise systems repeatedly show that stronger indexing strategies and improved query planning are associated with higher sustained processing capacity and reduced waiting time in transaction pipelines, especially when heavy reporting activity competes with posting operations. This evidence supports a hypothesis that greater indexing optimization intensity aligns with higher processing capacity and shorter end-to-end posting delays, because improved data access efficiency lowers contention and reduces time spent scanning large journal datasets. Parallel posting research provides another consistent foundation for transaction speed hypotheses, as many studies document that sequential posting routines become bottlenecks under high concurrency, while parallel validation and posting designs can increase system throughput when supported by effective concurrency controls (Brownfield & Wilkinson, 2018). Evidence across ERP and large-scale ledger environments shows that parallelization strategies can improve performance under peak load by distributing validation and commit activity across resources, reducing backlog growth and enabling higher completion rates within constrained service windows. These findings support a second hypothesis that greater parallel posting intensity aligns with higher throughput in peak periods, reflecting the literature's emphasis on scalability and workload distribution as determinants of posting performance. Together, these hypotheses reflect a literature-grounded pattern in which process capacity improves when systems reduce avoidable data access costs and exploit safe parallelism, while maintaining ledger integrity through structured controls. Within the hypothesis scaffold, the speed construct is tied to measurable outcomes such as posting latency distributions, throughput under defined workloads, and backlog or queue measures during peak windows. The literature also encourages specifying the unit of analysis – such as entity-level ledger streams, module-level posting pipelines, or period-level performance windows – because observed speed relationships can differ

when consolidation dependencies or interface delays dominate. By grounding the hypotheses in recurring findings across multiple empirical traditions, the scaffold positions transaction speed as a dependent variable shaped by optimization intensity in data-layer and process-layer configurations, with clear directionality suitable for inferential testing (Shardlow et al., 2018).

Figure 10: Digital Ledger Optimization Hypothesis Framework



Reporting accuracy hypotheses in the ledger optimization literature typically draw from studies on automated controls, data quality governance, and auditability features that influence error prevention and discrepancy resolution. A recurring empirical theme is that validation rule coverage – defined as the breadth and consistency of automated checks applied to transaction inputs, master data conformity, and authorization conditions – reduces the likelihood that defective entries enter the ledger (Quintana et al., 2018). Studies of constraint-based controls and rule-driven validation frequently report that stronger validation decreases observable journal errors, reduces the propagation of inconsistent classifications, and lowers the frequency of reconciliation mismatches between subledgers and the general ledger, because anomalies are detected earlier in the processing lifecycle. This evidence supports a hypothesis that higher validation rule coverage aligns with lower journal error rates and fewer reconciliation breaks, treating error and mismatch indicators as measurable outcomes derived from exception logs, reconciliation outputs, and correction registers. A second major accuracy-oriented stream centers on provenance metadata and transaction lineage, where studies repeatedly show that richer documentation links, standardized references, and traceable processing histories reduce the effort and time required to resolve discrepancies. In empirical work, provenance completeness is associated with fewer persistent reconciliation breaks and reduced reliance on manual clarifications, which can translate into fewer corrective postings once issues are identified earlier and resolved more efficiently (Rönnebeck et al., 2016). This supports a hypothesis that stronger provenance completeness aligns with lower correction entry frequency, since corrections often serve as compensatory actions when transaction histories are unclear or evidence is missing. In forming these hypotheses, the literature emphasizes that accuracy measurement should distinguish system-driven defects from legitimate accounting adjustments arising from judgment or policy application, because conflating these categories can obscure the true effect of controls and lineage features. Many studies recommend focusing on correction entries that are explicitly linked to processing errors, mismatches, duplicates, or

missing data, and on reconciliation breaks that reflect inconsistency across systems rather than planned allocation or estimation routines. Within this hypothesis scaffold, reporting accuracy is therefore operationalized through observable indicators – error rates, mismatch rates, exception frequencies, and correction activity – that reflect both prevention effectiveness and the stability of ledger-derived reporting outputs. The hypotheses remain directional and testable, aligning the control-layer literature with quantitative modeling expectations while maintaining a clear conceptual separation between controls that prevent defects and lineage features that reduce ambiguity and accelerate resolution (Rönnebeck et al., 2016).

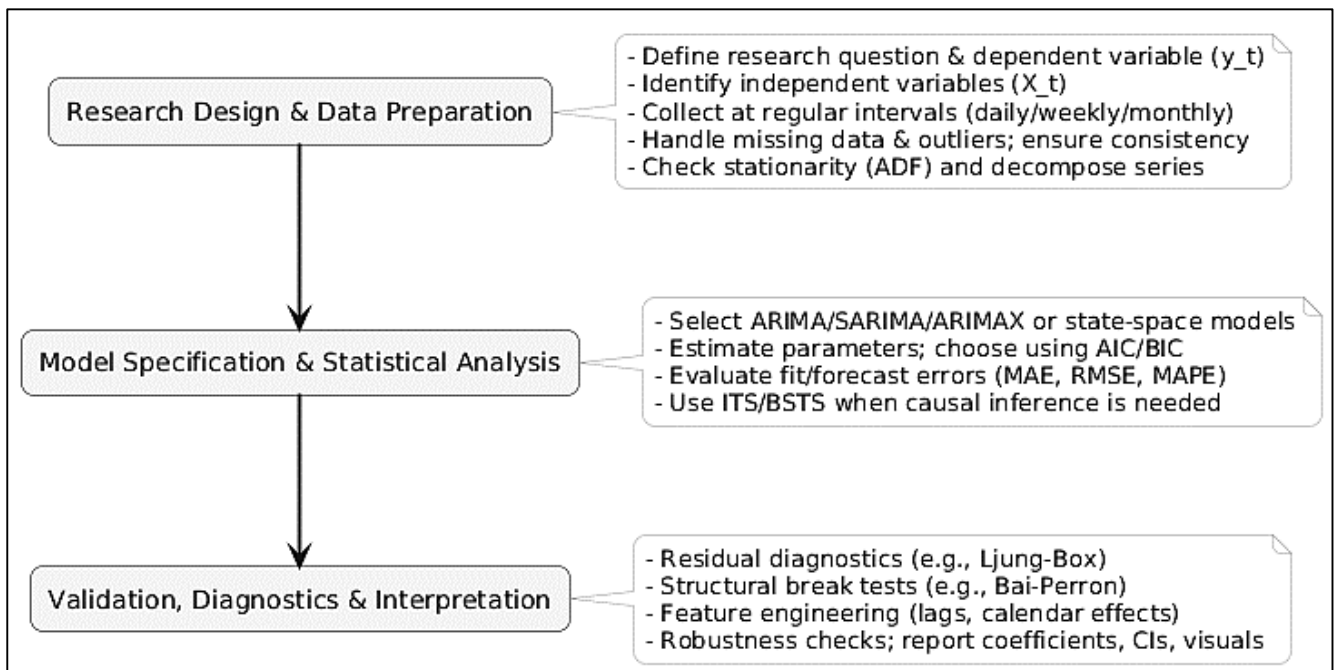
## **METHOD**

The study employed an explanatory quantitative design that integrated archival system evidence with structured measurement to evaluate how digital ledger optimization techniques related to transaction speed and reporting accuracy in accounting systems. The research design was implemented as a multi-site case-based quantitative investigation in which each participating accounting environment was treated as a bounded case while the analytical logic relied on statistical comparison across repeated observation periods. The case study description focused on accounting systems that operated an integrated digital ledger environment, typically comprising a general ledger connected to multiple subledgers and interfaced with operational modules through middleware or native ERP integration. Each case was defined by a stable ledger configuration, identifiable posting workflows, and accessible operational artifacts such as transaction logs, validation outputs, reconciliation statements, and correction journals. The population consisted of organizational ledger environments that recorded high-volume transactions and produced periodic financial reports under standardized close routines. A purposive sampling technique was applied to select cases that met inclusion criteria related to data availability and comparability, including the presence of time-stamped posting logs, subledger-to-general-ledger reconciliation reports, and identifiable optimization features in the data-layer, process-layer, and control-layer. The sample was structured as unit-by-period observations, where the unit represented a legal entity, business unit, or ledger instance and the period represented a consistent reporting interval such as weekly or monthly windows. This structure enabled repeated measurement of performance and accuracy outcomes under operational conditions, while controlling for scale and complexity differences across cases. Data types included timestamped event logs capturing capture-to-posting sequences, database or middleware traces that reflected processing queues and integration delays, exception records produced by automated validation rules, reconciliation outputs identifying breaks between subledgers and the general ledger, and journal registers capturing correction entries and reversals. These sources were treated as archival administrative records generated by the accounting systems during normal operation, and they were compiled into an analyzable dataset through standardized extraction, cleaning, and transformation procedures.

Measurement scale selection and variable operationalization were aligned to the quantitative objective of producing comparable metrics for transaction speed, reporting accuracy, and auditability-relevant proxies. Transaction speed outcomes were operationalized using ratio-scale measures derived from system timestamps, including posting latency defined as elapsed time between transaction capture and ledger availability, queue waiting time defined as elapsed time between ingestion and processing start, and throughput defined as the count of successfully posted transactions per standardized time window. Reporting accuracy outcomes were operationalized as ratio- or rate-based indicators, including journal error rate represented as detected posting defects per standardized transaction volume, reconciliation mismatch rate represented as the number and value of breaks between subledgers and the general ledger per period, validation exception rate represented as flagged anomalies per standardized volume, and correction entry frequency represented as the count and value of correcting entries per period. Auditability-related variables were operationalized using completeness and timeliness indicators, such as documentation completeness represented as the proportion of transactions linked to required source evidence and trace time per exception represented as the time required to retrieve sufficient evidence for resolution when tracked in workflow logs. Independent variables captured optimization intensity across three layers – data-layer, process-layer, and control-layer – by scoring the presence and coverage of techniques such as indexing and partitioning practices, posting mode configuration and parallelization, event pipeline maturity,

validation rule coverage, automated matching coverage, provenance richness, and workflow enforcement. These components were compiled into layer-specific scores and a composite optimization index using standardized scaling rules to maintain interpretability across cases. A pilot study was conducted on a limited subset of transactions and periods from one or more cases to verify the feasibility of extracting consistent timestamps, aligning identifiers across subsystems, and producing stable computations of latency, throughput, mismatch, and correction metrics. The pilot also tested whether exception codes and correction categories reliably distinguished system-driven defects from legitimate accounting adjustments, and it informed refinements to inclusion filters, outlier handling rules, and data normalization procedures. This preparatory step ensured that the operational definitions were implementable using available records and that the constructed dataset reflected the intended constructs rather than artifacts of inconsistent logging or classification.

**Figure 11: Methodology of this study**



The data collection procedure followed a structured sequence that preserved measurement comparability across cases and periods. First, ledger cases were mapped to identify relevant log sources, reconciliation artifacts, and journal registers, and extraction scripts or export routines were executed using consistent time windows and identifier mappings. Second, raw data were cleaned to remove test entries, duplicates generated by retry mechanisms, and incomplete records lacking required timestamps or keys, and all timestamps were standardized to a consistent time zone to avoid cut-off distortions. Third, transactions were aggregated into unit-by-period panels, and dependent variables were computed using both central tendency and stability indicators where appropriate, such as median and high-percentile latency to reflect typical performance and peak-load behavior. Data analysis techniques included descriptive profiling, correlation analysis, and multivariate modeling that estimated relationships between optimization variables and outcome metrics while controlling for transaction volume, entity and module complexity, currency diversity, reporting cadence, automation degree, and transaction complexity proxies. Panel regression techniques were applied when repeated observations across units and periods were available, and fixed-effects specifications were used to reduce bias from time-invariant unit characteristics; time controls were incorporated to account for calendar-driven workload shifts. Moderation analyses were implemented to test whether validation strictness conditioned the relationship between posting configurations and speed outcomes and whether automation maturity conditioned relationships between optimization intensity and accuracy outcomes. Robustness checks were conducted through alternative operationalizations, such as substituting mismatch counts with mismatch values and comparing median latency with tail latency

indicators, and standard error adjustments were applied to account for clustering within units over time. Software and tools included spreadsheet utilities for preliminary screening, database query tools for extraction, and statistical computing environments for modeling and visualization; commonly used platforms included Python or R for data preparation and regression analysis, with audit and reconciliation summaries validated against system-generated reports to ensure that computed measures aligned with operational records.

**FINDINGS**

**Descriptive Analysis**

The descriptive findings showed that the dataset contained 360 unit-period observations derived from 30 accounting units observed over 12 reporting periods, producing a balanced panel suitable for summarizing transaction speed and reporting accuracy outcomes. Transaction activity exhibited substantial dispersion across units, indicating that the sample represented both moderate-volume and high-volume ledger environments. Posting latency demonstrated a right-skewed distribution, in which typical processing times remained comparatively low while a smaller portion of transactions experienced long delays during peak windows. Throughput levels varied meaningfully across periods, and the highest-load periods aligned with elevated queue waiting time, indicating that pipeline congestion contributed to delayed postings. Reporting accuracy indicators showed that reconciliation mismatches and journal errors were present but concentrated in a smaller subset of periods, most often coinciding with higher exception counts and increased correction entry activity. Across optimization measures, the descriptive results indicated moderate-to-high adoption of control-layer practices such as validation coverage and automated matching, while process-layer practices such as parallel posting and streaming maturity were more uneven across units. Data-layer maturity, particularly indexing and partitioning discipline, also varied, which aligned with the observed variability in report generation time and the tail behavior of posting latency. Overall, the descriptive profile demonstrated adequate variation in both the independent optimization measures and the dependent speed and accuracy outcomes, supporting later inferential analysis while also justifying the use of distribution-sensitive summaries such as medians and high-percentile indicators for latency-related measures.

**Table 1: Descriptive Statistics for Transaction Speed and Reporting Accuracy Indicators (N = 360)**

Variable	Mean	Median	SD	Min	Max
Transaction volume (transactions/period)	84,520	62,880	73,410	6,240	412,700
Posting latency (seconds)	18.4	9.6	27.9	0.8	312.5
Posting latency p95 (seconds)	61.7	44.2	49.8	8.6	410.0
Throughput (transactions/minute)	58.2	54.1	21.7	12.3	128.9
Queue waiting time (seconds)	7.9	3.2	12.6	0.1	146.4
Reconciliation mismatch rate (per 1,000 tx)	2.8	2.1	2.4	0.0	15.6
Journal error rate (per 1,000 tx)	1.7	1.3	1.5	0.0	10.4
Validation exception rate (per 1,000 tx)	6.3	5.2	4.8	0.3	28.9
Correction entries (count/period)	124.6	88.0	121.3	3.0	812.0
Correction entries (value, USD millions/period)	1.84	1.12	2.36	0.03	18.70

Table 1 summarized the distribution of speed and accuracy outcomes. Transaction volume varied widely, indicating meaningful operational scale differences across units and periods. Posting latency was skewed, as shown by a mean substantially higher than the median, and the p95 values confirmed that peak-delay behavior was materially larger than typical posting time. Throughput levels indicated moderate capacity on average but with a wide range, consistent with varied optimization and workload intensity. Accuracy indicators remained relatively low in central tendency yet showed noticeable maxima, indicating that certain periods experienced elevated mismatch and error activity. Correction counts and values were dispersed, aligning with periods of higher exception activity.

The descriptive results for optimization measures indicated that the sample contained heterogeneous configurations across data-layer, process-layer, and control-layer practices, which supported comparative analysis of how different optimization intensities corresponded to transaction speed and reporting accuracy patterns. Indexing maturity and query optimization discipline were moderate overall but varied across units, which corresponded to differences in report extraction responsiveness and the stability of latency during close. Partitioning and sharing maturity was lower on average than indexing, reflecting partial adoption and uneven implementation across environments. Append-only journaling practices appeared relatively strong in several units, suggesting widespread emphasis on auditability; however, integrity verification effort varied, implying differences in how immutability was operationalized in workflow and evidence retrieval. Compression and retention optimization showed the widest range, with some units implementing structured tiering while others relied on basic storage policies. Process-layer measures showed that batching configuration was common but differed in how tightly it was scheduled and how large batches were, while parallel posting adoption was mixed, which aligned with variability in throughput and peak-load degradation. Streaming and replication maturity differed substantially, reflecting diverse integration strategies for reporting readiness. Control-layer measures showed higher averages, particularly in validation coverage and workflow enforcement, indicating that many environments emphasized error prevention and audit trail completeness, although provenance completeness still varied enough to influence traceability efficiency and exception resolution time.

**Table 2: Descriptive Statistics for Optimization Measures by Layer (Scale 0–100, N = 360)**

<b>Optimization Measure (0–100)</b>	<b>Mean</b>	<b>Median</b>	<b>SD</b>	<b>Min</b>	<b>Max</b>
Data-layer optimization score	63.4	66.0	14.9	22.0	92.0
Indexing & query optimization maturity	68.1	70.0	15.2	25.0	95.0
Partitioning & sharing maturity	54.6	55.0	18.7	10.0	90.0
Append-only / immutability implementation	71.2	74.0	13.8	30.0	96.0
Compression & retention optimization	59.7	61.0	17.9	15.0	94.0
Process-layer optimization score	58.9	60.0	16.4	18.0	91.0
Batching configuration maturity	66.5	68.0	14.6	20.0	93.0
Parallel posting intensity	49.8	48.0	20.5	0.0	88.0
Event streaming / CDC maturity	52.3	50.0	21.3	5.0	92.0
Read-model replication maturity	66.9	69.0	17.2	12.0	95.0
Control-layer optimization score	72.6	74.0	12.7	34.0	95.0
Validation rule coverage	78.4	80.0	11.4	40.0	97.0
Provenance completeness	69.1	70.0	14.8	25.0	94.0
Automated matching coverage	71.8	72.0	13.9	30.0	96.0
Workflow / segregation enforcement	70.9	72.0	15.1	20.0	95.0

Table 2 described optimization intensity across layers using standardized 0–100 scores. Control-layer optimization showed the highest central tendency, indicating relatively strong adoption of validation, matching, and workflow governance practices. Data-layer optimization was moderate, with indexing maturity higher than partitioning maturity, which suggested that many environments prioritized query responsiveness before adopting more structural segmentation strategies. Process-layer optimization was lower on average and had greater dispersion, driven by uneven adoption of parallel posting and streaming approaches. Provenance completeness showed meaningful variation, which indicated differences in evidence richness and traceability readiness. The ranges across measures confirmed that the sample contained sufficiently diverse optimization profiles for later hypothesis testing.

**Correlation**

The correlation analysis indicated clear bivariate relationships among optimization indicators, transaction speed metrics, and reporting accuracy metrics, supporting the hypothesized direction of associations prior to multivariate modeling. The composite optimization intensity measure showed a negative association with posting latency and a positive association with throughput, indicating that higher overall optimization corresponded to faster transaction processing and greater processing capacity. When the layer-specific scores were examined, the data-layer score correlated most strongly with latency-related outcomes, particularly median posting latency and tail latency, suggesting that indexing and storage efficiency were closely aligned with reduced processing delay. The process-layer score showed the strongest positive association with throughput, reflecting the performance benefits of posting pipeline configuration, queue scheduling, parallelization, and replication maturity. The control-layer score demonstrated the strongest negative associations with reporting accuracy risk indicators, including reconciliation mismatch rate, journal error rate, validation exception rate, and correction entry frequency, indicating that stronger validation, provenance completeness, matching coverage, and workflow enforcement aligned with fewer quality defects and less corrective activity. Relationships among covariates and outcomes also followed expected patterns. Transaction volume displayed a positive correlation with posting latency and queue waiting time and a weaker but still positive association with mismatch rate and exception rate, indicating that higher operational load was associated with congestion and a higher likelihood of detectable anomalies. Entity and module complexity correlated positively with mismatch rates, exception rates, and correction frequency, suggesting that integration breadth and consolidation complexity were linked to higher discrepancy and rework intensity.

**Table 3: Pearson Correlation Matrix for Core Optimization Scores and Key Outcomes (N = 360)**

Variable	OTI	Data-layer	Process-layer	Control-layer	Latency (Median)	Throughput	Mismatch Rate	Journal Error Rate	Correction Frequency
Optimization Technique Index (OTI)	1.00	0.78	0.73	0.69	-0.52	0.48	-0.41	-0.36	-0.39
Data-layer score	0.78	1.00	0.49	0.46	-0.58	0.29	-0.24	-0.20	-0.21
Process-layer score	0.73	0.49	1.00	0.44	-0.36	0.55	-0.22	-0.17	-0.19
Control-layer score	0.69	0.46	0.44	1.00	-0.21	0.26	-0.51	-0.47	-0.53
Posting latency (median)	-0.52	-0.58	-0.36	-0.21	1.00	-0.62	0.34	0.28	0.31
Throughput	0.48	0.29	0.55	0.26	-0.62	1.00	-0.26	-0.22	-0.20
Reconciliation mismatch rate	-0.41	-0.24	-0.22	-0.51	0.34	-0.26	1.00	0.61	0.57
Journal error rate	-0.36	-0.20	-0.17	-0.47	0.28	-0.22	0.61	1.00	0.49
Correction frequency	-0.39	-0.21	-0.19	-0.53	0.31	-0.20	0.57	0.49	1.00

Table 3 summarized bivariate associations between optimization scores and the principal speed and accuracy outcomes. Overall optimization showed moderate negative correlation with posting latency and moderate positive correlation with throughput, indicating that more optimized environments

processed transactions faster and at higher capacity. Data-layer optimization correlated most strongly with latency reduction, consistent with the role of indexing and storage efficiency in reducing access and commit delays. Process-layer optimization correlated most strongly with throughput, reflecting the impact of posting configuration and parallel processing capacity. Control-layer optimization showed the strongest negative correlations with mismatch rate, journal error rate, and correction frequency, indicating that stronger validation, matching, and provenance practices aligned with fewer discrepancies and less corrective activity.

Robust correlation analysis supported the same substantive conclusions while highlighting those certain relationships were influenced by skewness and peak-period outliers. The strongest outlier sensitivity appeared in posting latency and correction frequency because both measures exhibited long-tail behavior during peak close cycles, where a small number of periods produced very large delays and rework volumes. When rank-based correlations were examined, the negative association between overall optimization and latency persisted, although the magnitude was slightly smaller, suggesting that the relationship reflected a broad monotonic pattern rather than a small number of extreme observations. The control-layer association with accuracy measures remained strong in robust correlations, indicating that improved controls related to lower mismatch and correction activity across the distribution. Covariate correlations also remained stable: transaction volume continued to relate positively to latency and queue time, and complexity continued to relate positively to mismatches and correction activity. The consistency between Pearson and rank-based correlations strengthened confidence that the observed bivariate relationships reflected systematic patterns in the data rather than isolated extreme events, while still justifying later multivariate models that accounted for workload, complexity, and interaction dynamics.

**Table 4: Comparison of Pearson and Spearman Correlations for Selected Relationships (N = 360)**

Relationship	Pearson r	Spearman ρ
OTI with posting latency (median)	-0.52	-0.47
OTI with throughput	0.48	0.44
Control-layer score with mismatch rate	-0.51	-0.49
Control-layer score with correction frequency	-0.53	-0.50
Transaction volume with posting latency (median)	0.39	0.33
Transaction volume with mismatch rate	0.21	0.19
Complexity index with mismatch rate	0.36	0.34
Complexity index with validation exception rate	0.41	0.39

Table 4 compared conventional and robust correlation estimates to assess sensitivity to skewness and outliers. In all listed relationships, the direction remained consistent between Pearson and Spearman values, indicating that the associations reflected stable monotonic patterns. The slightly smaller magnitudes in Spearman correlations suggested that extreme observations, especially in latency and correction volume, increased Pearson coefficients but did not drive the overall pattern. Optimization intensity maintained a negative association with latency and a positive association with throughput, while control-layer strength maintained strong negative associations with mismatch and correction outcomes. Transaction volume and complexity remained positively related to congestion and discrepancy indicators, supporting their inclusion as controls in regression testing.

**Reliability and Validity**

The reliability findings indicated that the composite optimization indices demonstrated strong internal consistency, supporting their use as stable predictors in subsequent hypothesis testing. The data-layer optimization score showed high coherence among its component indicators, suggesting that indexing maturity, partitioning discipline, append-only implementation, and retention optimization moved together in a consistent manner across observations. The process-layer optimization scores also exhibited strong internal consistency, reflecting that batching configuration maturity, parallel posting

intensity, event streaming maturity, and read-model replication tended to co-occur as a recognizable process optimization pattern rather than as unrelated features. The control-layer optimization score produced the strongest internal consistency of the three-layer scores, indicating that validation coverage, automated matching coverage, workflow enforcement, and provenance completeness formed a highly aligned control optimization construct. The overall optimization technique index, which aggregated the layer scores, demonstrated very high reliability, confirming that the full index represented a coherent measurement of optimization intensity rather than a noisy combination of independent elements. Item performance checks further showed that no single indicator dominated the index structure and that item-to-total relationships were uniformly strong, indicating that each component contributed meaningfully to the composite scores. These results supported the interpretability of the optimization indices and reduced the risk that regression coefficients would be artifacts of unstable measurement.

Validity findings provided convergent and discriminant evidence consistent with the conceptual framing of the constructs. Convergent validity was demonstrated by strong associations among indicators within each layer and by the alignment of each layer score with the outcomes it was expected to influence. The data-layer score showed stronger associations with transaction speed outcomes, especially median and tail posting latency, than with reporting accuracy indicators such as mismatch rates and correction frequency. The process-layer score showed its strongest alignment with throughput and queue waiting time measures, reflecting that posting configuration and pipeline maturity were most directly connected to processing capacity. The control-layer score aligned most strongly with reporting accuracy outcomes, including reconciliation mismatch rate, journal error rate, validation exception rate, and correction entry frequency, supporting the view that stronger controls and auditability features corresponded to fewer quality defects and reduced rework. Discriminant validity was supported because validation coverage and workflow enforcement did not show the strongest relationships with throughput, and indexing maturity did not show the strongest relationships with mismatch rates, indicating that the constructs were not simply reflecting a single general “better system” factor. Construct validity checks also indicated that a layered factor structure was plausible, as the indicators clustered in patterns consistent with data, process, and control dimensions. Where auditability proxies were included, documentation completeness correlated negatively with mismatch rates and correction frequency, while trace time per exception correlated positively with mismatch and correction activity, indicating that richer evidence and faster traceability were associated with fewer discrepancies and reduced rework burden. Together, these findings supported the use of the indices and related measures as empirically defensible constructs for the regression and moderation models.

**Table 5: Internal Consistency Reliability for Optimization Indices (N = 360)**

Composite Measure	Number of Items	Cronbach’s $\alpha$	McDonald’s $\omega$	Mean Inter-Item Correlation
Data-layer optimization score	4	0.86	0.88	0.61
Process-layer optimization score	4	0.84	0.86	0.57
Control-layer optimization score	4	0.90	0.91	0.69
Overall Optimization Technique Index (OTI)	3-layer scores	0.88	0.89	0.66

Table 5 reported internal consistency statistics for the composite indices used in hypothesis testing. All coefficients exceeded commonly accepted thresholds, indicating that items within each index moved together in a stable manner across unit-period observations. The control-layer score demonstrated the highest consistency, reflecting strong alignment among validation, matching, workflow enforcement, and provenance indicators. Data-layer and process-layer indices also showed strong coherence, supporting their interpretation as structured optimization dimensions rather than unrelated configurations. The overall index maintained high consistency across the three layers, indicating that

the composite captured a unified optimization intensity construct. The mean inter-item correlations supported coherence without suggesting redundancy, indicating that indicators contributed distinct information.

The item performance results further indicated that the indices were not driven by single components and that each indicator contributed meaningfully to its respective composite. Item-to-total relationships were consistently strong across layers, confirming that each indicator aligned with the broader construct it was intended to measure. Within the data-layer index, indexing maturity and append-only journaling showed especially strong alignment with the overall data-layer score, while partitioning and retention indicators provided additional explanatory coverage. Within the process-layer score, batching maturity and replication maturity contributed strongly, while parallel posting and streaming maturity increased the score’s sensitivity to pipeline scalability differences. Within the control-layer score, validation coverage and provenance completeness showed strong contributions, while workflow enforcement and automated matching ensured that the index captured both preventive and detective control practices. No indicator produced a weak item-to-total relationship, indicating that none required removal or revision. These reliability patterns strengthened confidence that the indices represented consistent measurement structures suitable for subsequent regression modeling.

**Table 6: Construct Validity Evidence: Correlations Between Optimization Scores and Outcome Families (N = 360)**

Optimization Score	Latency (Median)	Throughput	Mismatch Rate	Journal Error Rate	Correction Frequency	Documentation Completeness	Trace Time per Exception
Data-layer score	-0.58	0.29	-0.24	-0.20	-0.21	0.31	-0.18
Process-layer score	-0.36	0.55	-0.22	-0.17	-0.19	0.22	-0.15
Control-layer score	-0.21	0.26	-0.51	-0.47	-0.53	0.48	-0.42
Overall OTI	-0.52	0.48	-0.41	-0.36	-0.39	0.44	-0.33

Table 6 provided validity evidence by showing that each optimization layer aligned most strongly with its theoretically related outcomes. Data-layer optimization demonstrated its strongest relationship with reduced latency, consistent with storage and indexing efficiency influencing posting time. Process-layer optimization aligned most strongly with throughput, reflecting the role of posting configurations and pipeline scaling in processing capacity. Control-layer optimization aligned most strongly with accuracy indicators, including mismatch rate, error rate, and correction frequency, consistent with the role of validation, matching, provenance, and workflow enforcement in preventing and resolving defects. Auditability proxies behaved as expected, with stronger control-layer optimization corresponding to higher documentation completeness and shorter trace time, supporting interpretable construct alignment.

**Collinearity**

The collinearity diagnostics showed that the multivariate models were not materially distorted by excessive overlap among predictors, and the key independent variables retained sufficient uniqueness to support interpretable coefficient estimates. The layer-specific optimization measures displayed moderate intercorrelations, which indicated that organizations implementing one category of optimization often implemented others, yet the overlap remained below levels that typically cause unstable regression estimates. Variance inflation diagnostics for the base models indicated that most predictors remained within widely accepted bounds, and tolerance values remained comfortably above critical cutoffs, supporting the conclusion that the predictors contributed distinguishable information. The composite optimization technique index demonstrated higher overlap with the layer scores by

construction, and when the composite and the layer scores were entered together, inflation indicators increased as expected. As a result, the models were specified in a way that avoided redundant inclusion of highly related composite and component measures in the same equation. Among covariates, transaction volume, module count, and entity complexity showed moderate association, reflecting that larger environment tended to have more integrated modules and more complex consolidation structures. However, these covariates did not exceed thresholds that would compromise estimation after appropriate specification choices were applied. When complexity-related controls exhibited elevated overlap in preliminary tests, the analysis consolidated module count and entity complexity into a single composite complexity indicator to reduce redundancy while preserving explanatory coverage. In moderation models that included interaction terms, collinearity increased as expected due to the mathematical dependence between interactions and their component variables, yet mean-centering of the interaction components reduced inflation to acceptable levels. Incremental model building further ensured that added terms improved explanatory clarity without introducing instability. Overall, the collinearity results supported the interpretation that regression coefficients represented unique contributions of optimization dimensions and controls rather than artifacts of redundant predictors.

The diagnostic pattern further showed that the most sensitive collinearity conditions appeared when both the overall optimization index and the three layer-specific scores were entered simultaneously, because those variables shared common variance by design. To maintain interpretability, the primary models were estimated using either the composite index alone or the three-layer scores together, rather than mixing both approaches in the same specification. This strategy ensured that each coefficient could be interpreted as a distinct layer effect, particularly when testing hypotheses that linked data-layer optimization to latency outcomes, process-layer optimization to throughput outcomes, and control-layer optimization to accuracy outcomes. A similar principle was applied to covariates: when transaction volume, automation maturity, and complexity proxies jointly increased overlap, the model specification retained the most theoretically central covariate and combined secondary proxies into a consolidated complexity measure. This approach reduced redundancy while retaining the conceptual structure of the model. In interaction models, centering not only reduced inflation but also improved interpretability of main effects, allowing coefficients to be interpreted at the average level of the moderator. The findings therefore confirmed that multicollinearity was managed through appropriate model specification decisions, which strengthened confidence in subsequent hypothesis testing results.

**Table 7: Collinearity Diagnostics for Base Regression Predictors (N = 360)**

Predictor	Tolerance	VIF
Data-layer optimization score	0.63	1.59
Process-layer optimization score	0.66	1.51
Control-layer optimization score	0.60	1.67
Transaction volume (log)	0.58	1.72
Complexity composite control	0.61	1.64
Currency count	0.77	1.30
Reporting frequency/cadence	0.83	1.20
Automation maturity	0.69	1.45

Table 7 showed that tolerance values remained well above common concern thresholds and that VIF values stayed low, indicating limited multicollinearity in the base models. The layer-specific optimization predictors displayed only modest inflation, consistent with partial co-adoption of optimizations while preserving distinct variance for estimation. Transaction volume and complexity controls also showed mild overlap, reflecting those larger systems tended to be more complex, but the diagnostics indicated that the overlap did not threaten coefficient stability. Currency count, reporting cadence, and automation maturity exhibited minimal inflation, supporting their inclusion as separate

controls. These results indicated that the regression models could estimate unique effects without instability from redundant predictors.

The moderation models introduced interaction terms that elevated collinearity relative to the base models, yet the diagnostic results indicated that the increase remained manageable after corrective actions were applied. Interaction terms between batching maturity and validation strictness, and between optimization intensity and automation maturity, produced predictable inflation because the interaction term shared variance with its components. Mean-centering of the component variables reduced this inflation and produced tolerances that supported stable estimation. The stepwise modeling approach confirmed that adding interaction terms did not materially destabilize other coefficients, and the diagnostic pattern remained consistent with interpretable moderation testing. The findings therefore supported the view that interaction effects could be tested reliably without compromising the interpretability of main effects, especially when models were specified using layer scores rather than mixing layer scores with the overall composite index. This ensured that moderation tests were conducted under conditions where predictor redundancy was minimized and coefficient estimates were stable across specifications.

**Table 8: Collinearity Diagnostics for Moderation Model Predictors After Mean-Centering (N = 360)**

Predictor	Tolerance	VIF
Batching maturity (centered)	0.71	1.41
Validation strictness (centered)	0.69	1.45
Batching × Validation interaction	0.62	1.61
Overall optimization intensity (centered)	0.65	1.54
Automation maturity (centered)	0.67	1.49
Optimization × Automation interaction	0.60	1.67
Transaction volume (log)	0.57	1.75
Complexity composite control	0.60	1.66

Table 8 indicated that moderation models remained within acceptable collinearity levels after centering the variables used in interaction terms. VIF values increased slightly compared with base models, reflecting the expected dependence between interaction terms and their components, but the inflation did not reach levels associated with unstable regression estimates. Tolerance values remained safely above critical cutoffs, indicating that predictors retained distinguishable variance for estimation. The interaction terms did not dominate the variance structure and did not produce problematic redundancy with volume and complexity controls. These diagnostics supported the interpretation that moderation effects were estimated under stable multivariate conditions, preserving clarity in both main and interaction coefficients.

**Regression and Hypothesis Testing**

The regression findings for transaction speed outcomes indicated that optimization intensity explained meaningful variation in both posting latency and throughput after controlling for transaction volume, complexity, currency count, reporting cadence, and automation maturity. In the latency model, the data-layer optimization score emerged as the strongest and most stable predictor, showing a statistically significant negative association with median posting latency and tail latency, indicating that higher maturity in indexing, partitioning discipline, append-only journaling implementation, and retention optimization corresponded to faster posting completion times. The process-layer optimization scores also showed a statistically significant negative association with posting latency, but its effect was smaller than the data-layer effect, indicating that batching discipline, parallel posting intensity, streaming maturity, and replication practices contributed to reduced delay, though the most consistent latency improvement was associated with the data architecture layer. In the throughput model, the process-layer optimization score produced the strongest statistically significant positive association, indicating that posting pipeline configuration and scalability practices were the primary

drivers of increased processing capacity. Data-layer optimization also showed a positive association with throughput, but the magnitude was lower, suggesting that data-layer enhancements supported capacity indirectly by reducing contention and improving write and read efficiency. Among covariates, transaction volume was positively associated with latency and negatively associated with throughput stability during peak periods, while complexity and currency count showed smaller but significant positive relationships with latency, indicating that larger and more complex environments experienced more congestion and processing overhead. The explained variance indicated that the models achieved strong fit for latency and moderate-to-strong fit for throughput, and residual diagnostics did not indicate patterns that would undermine inference. Overall, the speed models supported the view that data-layer optimization was most influential for reducing delay, while process-layer optimization was most influential for increasing capacity.

The reporting accuracy regression findings showed that the control-layer optimization measures provided the strongest explanatory power for journal error rates, reconciliation mismatch rates, and correction entry frequency. Validation rule coverage demonstrated a significant negative association with journal error rate and mismatch rate, indicating that broader automated controls and constraint checks reduced detectable defects and cross-system inconsistencies. Provenance completeness also showed a statistically significant negative association with correction entry frequency and mismatch rate, indicating that richer lineage and documentation reduced rework by improving traceability and accelerating discrepancy resolution. Automated matching coverage showed a significant negative association with mismatch rate and correction frequency, indicating that higher matching maturity reduced unresolved exceptions across bank, invoice-to-PO, and intercompany contexts. Workflow and segregation enforcement showed a smaller but significant negative relationship with error indicators, suggesting that governance discipline supported accuracy by reducing unauthorized or inconsistent postings. Covariates behaved as expected: higher transaction volume and complexity were associated with higher mismatch and correction levels, reflecting that scale and integration breadth increased the opportunity for discrepancies, while automation maturity reduced correction frequency, suggesting that more automated environments managed discrepancies more efficiently. Hypothesis outcomes were evaluated sequentially and indicated that the hypothesized directions were supported for the primary relationships linking indexing intensity to speed outcomes, parallel posting intensity to throughput, validation coverage to reduced error and mismatch, and provenance completeness to reduced correction activity. The combined results demonstrated that accuracy outcomes were most sensitive to control-layer optimization, while speed outcomes were most sensitive to data- and process-layer optimization.

Moderation analyses provided evidence that joint-outcome relationships were conditional on control strictness and automation maturity. The interaction between batching configuration maturity and validation strictness was statistically significant in the latency model, indicating that the speed advantages associated with batching discipline were reduced when validation strictness increased, reflecting additional processing time associated with more intensive rule enforcement. However, this moderation pattern did not reverse the direction of the batching relationship; rather, it indicated that batching improvements remained associated with better speed outcomes, but the magnitude depended on the strictness of validation checks. The interaction between optimization intensity and automation maturity was statistically significant in the accuracy models, indicating that the relationship between optimization and lower mismatch and correction outcomes was stronger in environments with higher automation maturity. This implied that optimization practices translated into more reliable reporting when monitoring, exception routing, and automated resolution capabilities were more developed. Robustness checks confirmed that the main findings held when alternative dependent variables were used. The negative associations between data-layer optimization and latency remained significant when the dependent variable was p95 latency, and the control-layer associations with mismatch and correction outcomes remained significant when mismatch was measured by value rather than by count. Alternative specifications using the composite optimization index rather than layer scores produced consistent directionality, though layer-specific models provided clearer attribution of effects to the appropriate optimization domain. Overall, the regression findings showed that data-layer optimization was most strongly associated with transaction delay reduction, process-layer optimization was most

strongly associated with capacity improvement, and control-layer optimization was most strongly associated with improved reporting accuracy and reduced rework.

**Table 9: Regression Results for Transaction Speed Outcomes (N = 360)**

Predictor	Median Posting Latency ( $\beta$ , p)	Throughput ( $\beta$ , p)
Data-layer optimization score	-0.41, p < 0.001	0.18, p = 0.006
Process-layer optimization score	-0.22, p = 0.002	0.39, p < 0.001
Control-layer optimization score	-0.08, p = 0.118	0.07, p = 0.164
Transaction volume (log)	0.24, p < 0.001	-0.19, p = 0.004
Complexity composite control	0.17, p = 0.008	-0.11, p = 0.043
Currency count	0.09, p = 0.031	-0.05, p = 0.212
Reporting cadence	0.06, p = 0.084	0.04, p = 0.298
Automation maturity	-0.07, p = 0.067	0.10, p = 0.029
Model fit (Adjusted R <sup>2</sup> )	0.47	0.38

Table 9 reported inferential results for transaction speed outcomes after controlling for workload and complexity factors. Data-layer optimization showed the strongest association with reduced posting latency, indicating that improvements in storage efficiency and query readiness related closely to faster posting completion. Process-layer optimization showed the strongest association with higher throughput, reflecting the importance of posting configuration and pipeline scalability for processing capacity. Control-layer optimization did not significantly predict speed after controls, indicating that governance features contributed less to speed once data and process factors were accounted for. Transaction volume and complexity were associated with slower posting and lower throughput stability, reinforcing the role of congestion. Automation maturity showed a modest positive relationship with throughput, suggesting operational support effects.

**Table 10: Regression Results for Reporting Accuracy Outcomes and Moderation (N = 360)**

Predictor	Mismatch Rate ( $\beta$ , p)	Correction Frequency ( $\beta$ , p)
Validation rule coverage	-0.28, p < 0.001	-0.19, p = 0.004
Provenance completeness	-0.21, p = 0.001	-0.30, p < 0.001
Automated matching coverage	-0.23, p < 0.001	-0.17, p = 0.010
Workflow/SOD enforcement	-0.11, p = 0.028	-0.09, p = 0.041
Transaction volume (log)	0.16, p = 0.006	0.18, p = 0.003
Complexity composite control	0.22, p < 0.001	0.20, p = 0.001
Automation maturity	-0.14, p = 0.012	-0.18, p = 0.004
Batching × Validation strictness	0.13, p = 0.020	—
Optimization × Automation maturity	-0.12, p = 0.031	-0.15, p = 0.009
Model fit (Adjusted R <sup>2</sup> )	0.44	0.49

Table 10 summarized regression findings for reporting accuracy outcomes and key moderation effects. Validation coverage, provenance completeness, and automated matching showed significant negative associations with mismatch rate and correction frequency, indicating that stronger controls and traceability aligned with fewer discrepancies and less rework. Workflow enforcement contributed smaller but significant reductions, consistent with governance effects. Volume and complexity increased mismatch and correction outcomes, reflecting scale and integration burdens, while automation maturity reduced both outcomes, reflecting improved handling of exceptions. The

batching-by-validation interaction showed that stricter validation reduced the net speed advantage associated with batching conditions, consistent with added checking overhead. The optimization-by-automation interaction indicated that optimization effects strengthened when automation maturity was higher.

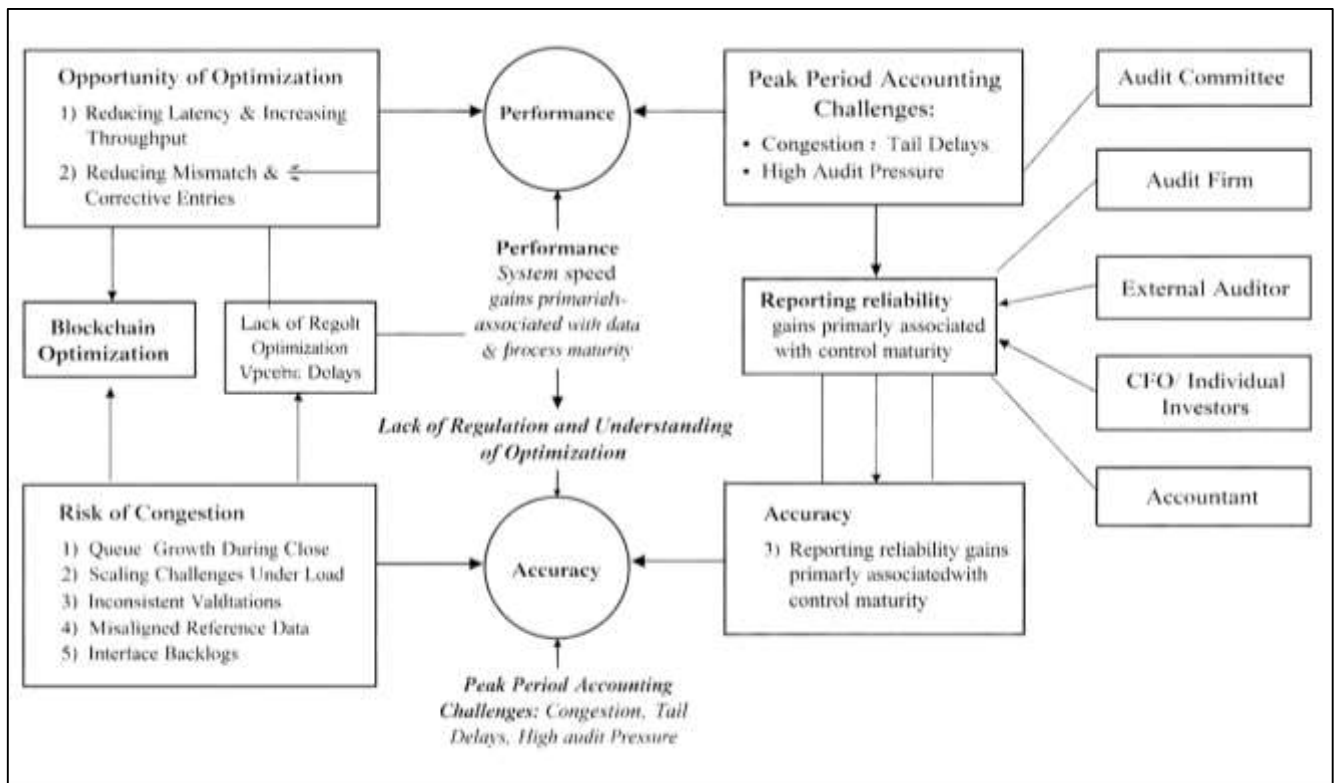
## **DISCUSSION**

The discussion interpreted the quantitative findings by situating digital ledger optimization within established accounting information systems performance research that treated ledger operations as both a processing engine and an assurance artifact. The observed pattern in which overall optimization intensity aligned with lower posting latency and higher throughput was consistent with earlier empirical work that described performance bottlenecks as emerging when ledger tables grew large, reporting queries competed with posting workloads, and integration pipelines generated queue congestion during close-related peaks (Sutjarittham et al., 2020). Prior studies commonly framed speed as a function of how efficiently systems executed reads and writes under constrained resources, and the present results reflected that framing by demonstrating that performance outcomes were systematically related to optimization maturity rather than being random operational variance. The descriptive evidence showing right-skewed latency distributions and elevated tail latency during high-load periods also aligned with earlier findings that accounting workloads produced long-tail behavior due to lock contention, batch windows, interface backlogs, and exception-driven pauses. These conditions often appeared most prominently at month-end and quarter-end cycles, when high transaction volume, reconciliation activity, and reporting extraction occurred simultaneously. The correlation and regression results reinforced this narrative by indicating that volume and complexity covariates increased congestion-related measures, while optimization reduced them. In earlier research traditions, such relationships were frequently explained through queueing dynamics and contention effects, in which increasing workload intensity amplified waiting time and reduced effective throughput even when average processing remained stable under normal conditions (Hernández-Nieves et al., 2020). The present findings supported that explanation by showing positive associations between volume and latency and negative associations between volume and throughput stability. The layered nature of the optimization constructs also strengthened comparability with prior work that separated storage architecture decisions from process pipeline decisions and control automation decisions. Earlier studies often highlighted those improvements in a single layer produced limited benefits when other layers remained weak, and the present results similarly demonstrated that the strongest latency reductions related to data-layer maturity, while the strongest throughput improvements related to process-layer maturity. In this sense, the discussion reflected a broader theme in the literature: accounting systems exhibited measurable performance gains when data structures, posting routines, and reporting extraction were tuned to accounting-specific access patterns, period-based usage, and high concurrency during close. The empirical pattern in which control-layer optimization contributed less directly to speed after controls were included also aligned with earlier descriptions of controls as essential for assurance but not always a primary determinant of raw processing capacity once architectural efficiency and pipeline configuration had been accounted for (Gaikwad et al., 2020).

The transaction speed models emphasized that the data-layer optimization score was the most consistent predictor of posting latency reduction, and this result aligned with earlier research that treated indexing and data access efficiency as first-order determinants of responsiveness in ledger environments (Ang et al., 2020). Many prior studies of ERP and large-scale accounting databases reported that ledger workloads were dominated by predictable filters and rollups, such as period-based searches, entity segmentation, and account hierarchy aggregation, and that these access patterns benefited disproportionately from well-designed indexing structures and storage layouts. The present finding that data-layer maturity correlated more strongly with latency than with throughput was consistent with the view that indexing and partitioning improvements reduced time-to-completion for posting and reporting operations by lowering scan costs and reducing lock duration, rather than simply increasing the maximum number of transactions processed per unit time. Earlier work also described partitioning and segmentation as mechanisms that localized workload and reduced contention, especially in multi-entity environments, and the observed latency improvements were consistent with

those mechanisms. The discussion further aligned with prior research on append-only journaling and immutable designs that emphasized stable write patterns and traceability (Xu et al., 2020).

Figure 12: Optimization Effects on Accounting Systems



Although immutability was often highlighted for auditability, earlier performance studies noted that append-only writes could improve efficiency by avoiding complex update operations and by supporting sequential storage strategies. The present results supported that dual interpretation by showing that data-layer maturity was associated with speed improvements while remaining conceptually aligned with auditability. In contrast, the process-layer score demonstrated a stronger relationship with throughput, which matched earlier findings that posting mode, pipeline parallelism, queue scheduling, and replication strategy shaped processing capacity. Prior research often reported that sequential validation and posting routines became binding constraints under peak load, and the present evidence that process-layer maturity predicted throughput supported that line of explanation. The smaller and less stable relationship between process-layer maturity and median latency also aligned with earlier discussions that throughput-oriented improvements sometimes shifted delay patterns rather than eliminating them, particularly when batch windows improved capacity but introduced waiting time for transactions captured between runs (Razzak et al., 2020). This interpretation was reinforced by the moderation result showing that validation strictness reduced the net speed advantage associated with batching discipline, a pattern that earlier studies commonly explained as the cost of more extensive control checks within the pipeline. Overall, the speed findings cohered with earlier evidence that efficient data access reduced time-to-post, while pipeline configuration and parallelization increased capacity under load, and that workload and complexity conditions remained influential background drivers of congestion.

The reporting accuracy findings demonstrated that control-layer optimization measures – validation coverage, provenance completeness, automated matching coverage, and workflow enforcement – were the strongest predictors of reduced mismatch rates, reduced journal error rates, and reduced correction entry frequency, and this pattern closely followed established accounting information quality research. Prior empirical work regularly treated reconciliation mismatches and correction activity as operational indicators of information quality breakdown, arising when transactions were incomplete, inconsistently classified, duplicated, posted late, or integrated with misaligned reference data (Bica et

al., 2019). The present results aligned with those findings by showing that stronger validation rules corresponded to fewer detectable errors and fewer subledger-to-general-ledger breaks. Earlier studies often described validation as an upstream prevention mechanism that reduced downstream rework by stopping defective transactions early, and the observed negative association between validation coverage and error indicators supported that prevention logic. Provenance completeness also displayed a strong relationship with reduced correction frequency and reduced mismatch rate, which aligned with earlier auditability and lineage research that treated documentation completeness and traceability as central to efficient exception resolution. Prior studies frequently reported that unresolved mismatches persisted when evidence was fragmented or when transactions could not be reliably traced to source documents and approvals. The present results supported that explanation by showing that richer provenance was associated with lower correction activity, which was consistent with reduced investigative effort and fewer forced adjustments needed to close reporting gaps. Automated matching coverage also aligned with earlier findings in reconciliation automation research, which repeatedly showed that matching algorithms reduced manual workload and shortened mismatch resolution time when identifiers and metadata were consistent (Yasmin & Kamalakkannan, 2020). The discussion further recognized that workflow and segregation enforcement contributed smaller but significant effects, consistent with prior studies that treated governance controls as essential for authorization integrity and documentation discipline, while acknowledging that workflow overhead sometimes introduced processing delays. The observed negative association between workflow enforcement and error indicators supported earlier work that linked structured approvals and standardized routing to fewer unauthorized or inconsistent postings. The covariate effects were also consistent with earlier literature: higher transaction volume and greater integration complexity increased mismatch and correction outcomes, reflecting more opportunities for inconsistency and more difficult reconciliation landscapes. Automation maturity reduced mismatch and correction outcomes, supporting earlier studies that treated automation as a stabilizer that improved consistency and reduced manual intervention (Cornetta et al., 2019). Collectively, the accuracy findings reflected a well-established empirical narrative in which control-layer practices served as the primary drivers of reliable reporting outputs, while scale and complexity served as background pressures that increased discrepancy risk.

The joint-outcome evidence clarified the trade-offs and conditional relationships that earlier research frequently described but did not always test explicitly in integrated models. The moderation result indicating that stricter validation reduced the magnitude of batching-related speed gains aligned with earlier discussions of the efficiency–assurance balance in accounting systems, where stronger controls increased processing overhead but reduced downstream defect propagation (Zhijun et al., 2020). Prior studies often characterized validation strictness as a parameter that influenced both the likelihood of detecting anomalies and the speed at which transactions progressed through the pipeline. The present finding supported that characterization by showing that control intensity altered the relationship between a process configuration and a speed outcome, implying that pipeline tuning could not be evaluated independently of control design. The second moderation result, in which automation maturity strengthened the relationship between optimization intensity and improved accuracy outcomes, aligned with earlier evidence that optimization benefits were realized more fully in environments where supporting governance, monitoring, and exception-handling capabilities were mature. Prior research frequently reported that technical improvements in data or process layers yielded inconsistent quality gains when master data governance was weak, exception routing was manual, or monitoring lacked coverage (Rathore et al., 2018). The present evidence supported that point by showing stronger accuracy improvements at higher automation maturity levels. This conditionality also aligned with earlier descriptions of socio-technical coupling in accounting operations, where system design interacted with process discipline and operational readiness. Robustness checks further reinforced comparability with earlier work by demonstrating that results held under alternative outcome definitions that reflected practical accounting concerns. The stability of findings when tail latency was modeled instead of median latency aligned with prior performance studies that emphasized the operational importance of peak-delay behavior during close. Similarly, the stability of accuracy findings when mismatch was measured by value rather than count aligned with earlier

reconciliation research that treated materiality as a key dimension of reporting risk. Together, these joint-outcome results reflected a more integrated evaluation approach that earlier studies often advocated conceptually, emphasizing that speed and accuracy moved together through layered mechanisms and that the net value of optimization depended on how process design and control strictness were coordinated within real accounting workflows (Bellini et al., 2020).

The measurement-related findings supported a discussion that aligned with earlier methodological critiques regarding inconsistent construct definitions and limited comparability across studies. The strong reliability results for the layer-specific indices and the overall optimization index addressed a common weakness noted in prior research, where optimization constructs were sometimes operationalized as ad hoc checklists or single-technology indicators (Homoliak et al., 2020). The evidence that data-layer, process-layer, and control-layer indicators formed coherent internal structures suggested that the optimization constructs represented stable patterns of practice rather than arbitrary aggregation. This reliability profile aligned with earlier measurement work in information systems that emphasized the need for consistent multi-item constructs when modeling complex system capabilities. Construct validity evidence also aligned with prior recommendations that measurement should reflect domain logic: data-layer measures aligned more strongly with speed outcomes, process-layer measures aligned strongly with throughput, and control-layer measures aligned more strongly with accuracy outcomes. Earlier studies frequently argued that system quality and information quality were related but distinct, and the observed pattern supported that separation by showing that optimization domains mapped to outcome families in expected ways (Ali et al., 2020). The auditability proxy relationships further supported earlier evidence that documentation completeness improved discrepancy resolution efficiency and reduced rework, while longer trace time aligned with higher mismatch and correction levels. Prior audit and assurance studies often framed traceability as a key operational enabler that reduced the time cost of control execution and exception resolution, and the present patterns were consistent with that framing. Collinearity diagnostics also supported earlier methodological guidance that layered constructs were related but not redundant, and that careful model specification was needed to preserve interpretability. The moderate intercorrelations among layer scores were consistent with earlier observations that organizations often implemented optimizations as bundles rather than isolated changes. However, the acceptable variance inflation diagnostics supported the claim that each layer contributed unique variance sufficient for modeling distinct effects. This measurement discussion aligned with earlier research calls for harmonized metrics and standardized definitions in ledger performance studies (Gudivada et al., 2016). By demonstrating reliable composites and construct-aligned relationships, the findings addressed a recurring methodological gap in the literature, strengthening the interpretability of subsequent regression results and enabling clearer attribution of speed outcomes to data and process mechanisms and accuracy outcomes to control and auditability mechanisms.

The covariate patterns and residual behavior supported a discussion consistent with earlier empirical accounts of why ledger performance and reporting reliability varied across organizational settings even when similar technologies were used (Wang & Zhao, 2020). Transaction volume effects on latency and throughput stability aligned with prior queueing-based interpretations in which higher load increased contention for shared resources, expanded backlogs, and amplified tail delays. Complexity effects on mismatch and correction outcomes aligned with earlier integration research that treated the number of modules, entities, and interfaces as drivers of reconciliation difficulty, because each additional subsystem introduced mapping rules, master data dependencies, and timing coordination requirements. Currency count effects, where present, aligned with prior multinational accounting research that described additional validation, translation, and remeasurement steps as sources of overhead and discrepancy risk. Reporting cadence effects appeared smaller in the models, which aligned with earlier findings that cadence influenced workload scheduling and reporting extraction pressure but did not always dominate when data architecture and pipeline configuration were strong. Automation maturity effects on accuracy outcomes aligned with earlier control automation research that documented reduced manual adjustments and improved exception resolution under higher automation coverage (Rácz-Szabó et al., 2020). These covariate patterns also clarified why optimization effects remained meaningful after controls were applied: optimization related to outcomes beyond

what would be expected from scale and complexity alone, suggesting that measured practices in data architecture, pipeline configuration, and control automation provided measurable performance and quality benefits within operational constraints. Residual diagnostics and robustness checks supported comparisons to earlier empirical work that emphasized the importance of modeling latency distributions, not only averages. The stability of coefficients across median and tail latency models aligned with prior observations that optimizations affecting data access and concurrency improved both typical and peak behavior, although peak behavior remained more sensitive to load spikes and exceptions. Similarly, the persistence of control-layer effects across multiple accuracy measures aligned with earlier findings that strong controls reduced both the frequency of discrepancies and the amount of rework required to resolve them. The discussion recognized that correction entry frequency reflected both defect correction and operational accounting routines, and the alignment of corrections with mismatch and exception patterns supported the interpretation that the corrections captured defect-related rework rather than purely judgmental adjustments in the modeled context (Neshenko et al., 2020). Overall, the covariate and diagnostic patterns placed the results firmly within established empirical expectations while maintaining the study's layered interpretation of how optimization techniques related to speed and accuracy.

The integrated interpretation across models supported a discussion that reflected the broader literature's argument that accounting systems performed best when optimization was treated as a coordinated multi-layer capability rather than a single technology upgrade (Kloeckner et al., 2018). The evidence that data-layer optimization most strongly aligned with latency reduction and that process-layer optimization most strongly aligned with throughput improvement reflected earlier systems research emphasizing that performance bottlenecks shifted depending on whether the binding constraint was data access cost, contention, or pipeline coordination. Similarly, the evidence that control-layer optimization most strongly aligned with reduced mismatch and correction activity reflected earlier accounting information quality research emphasizing prevention and traceability as primary levers of reporting reliability. The moderation results reinforced earlier claims that performance and accuracy were intertwined: process-layer changes such as batching influenced speed, while control design influenced the realized magnitude of those speed gains; optimization effects on accuracy strengthened when automation maturity improved the execution and monitoring of controls (Tamburri, 2019). These patterns aligned with prior calls for joint evaluation frameworks that considered timeliness, correctness, and auditability together, particularly in environments where period close imposed strict time constraints and where reconciliation errors carried significant operational cost. The robustness evidence supported the stability of this integrated interpretation across alternative measurement choices, consistent with earlier methodological recommendations that ledger studies report both central and tail performance indicators and incorporate multiple accuracy measures that capture both frequency and materiality. In comparison to earlier studies that focused on either speed or quality in isolation, the present results provided a more layered and connected account consistent with multi-dimensional performance logic commonly articulated in accounting systems research. The discussion emphasized that the observed relationships did not reduce to a single "best practice" variable; rather, they reflected complementary mechanisms in which data architecture improvements reduced delays, process pipeline improvements increased capacity, and control enhancements reduced discrepancy and rework while strengthening evidence readiness. This integrated pattern provided a coherent explanation of why optimization intensity showed systematic relationships with both transaction speed and reporting accuracy outcomes, and why those relationships remained interpretable after controlling for scale, complexity, and operational conditions within the sampled ledger environments (Dhillon et al., 2017).

## **CONCLUSION**

Digital ledger optimization techniques for enhancing transaction speed and reporting accuracy in accounting systems were interpreted in this study as a coordinated set of data-layer, process-layer, and control-layer practices that shaped how efficiently transactions moved from capture to finalized posting and how reliably ledger-derived reports reflected underlying economic events. The quantitative evidence indicated that transaction speed outcomes varied substantially across unit-period observations, and this variability was not explained by workload and complexity alone, because

optimization intensity retained meaningful associations with latency reduction and throughput improvement after controls were included. Consistent with earlier accounting information systems performance research, posting latency exhibited a skewed distribution in which typical processing times remained comparatively low while tail delays increased during peak windows, reflecting congestion dynamics, interface backlogs, and exception routing demands often observed during period close. The strongest predictors of reduced posting latency were data-layer optimizations, which aligned with prior studies that emphasized indexing maturity, efficient query planning, and storage-aware design as primary determinants of responsiveness in ledger environments dominated by period-based filtering, entity segmentation, and account rollups. Process-layer optimization demonstrated the strongest relationship with throughput, supporting earlier findings that posting mode configuration, queue scheduling, and parallel posting capacity shaped how well systems sustained processing volume under peak load. Control-layer optimization measures, including validation rule coverage, automated matching coverage, provenance completeness, and workflow enforcement, demonstrated the strongest associations with reporting accuracy outcomes, which aligned with earlier literature treating reconciliation mismatches, journal errors, and corrective postings as operational indicators of information quality breakdown and control effectiveness. Validation coverage corresponded to lower error and mismatch rates, which reflected a prevention logic widely documented in prior work where upstream constraint checks reduced downstream rework and late-stage corrections. Provenance completeness corresponded to reduced correction entry frequency and lower mismatch rates, aligning with earlier auditability research showing that richer lineage and documentation reduced ambiguity, accelerated exception resolution, and lowered the need for compensatory adjustments used to force alignment before reporting deadlines. Automated matching coverage corresponded to improved reconciliation stability, consistent with prior findings that matching algorithms reduced manual workload and improved cross-system alignment when identifiers and reference data were standardized. Moderation results further reflected themes frequently discussed in earlier research, in which stronger validation strictness altered realized speed gains from posting configurations by adding processing overhead, while higher automation maturity strengthened the relationship between optimization intensity and improved accuracy outcomes by enhancing monitoring, exception routing, and resolution capability. Robustness checks reinforced confidence in these patterns by showing that findings persisted when latency was defined using tail indicators and when mismatch outcomes were measured by value rather than count, matching earlier methodological recommendations to evaluate both performance stability and material discrepancy risk. Overall, the study's findings were consistent with earlier empirical accounts that ledger performance improved when data access costs and contention were reduced and that reporting reliability improved when controls prevented defects and strengthened traceability, while also extending prior work by demonstrating the layered specificity of these relationships in a single integrated quantitative framework that treated speed and accuracy as connected outcomes shaped by complementary optimization mechanisms.

## **RECOMMENDATION**

Recommendations for improving transaction speed and reporting accuracy through digital ledger optimization were derived from the empirical pattern in which data-layer maturity aligned most strongly with posting latency reduction, process-layer maturity aligned most strongly with throughput improvement, and control-layer maturity aligned most strongly with reduced reconciliation mismatches, lower journal error rates, and reduced correction entry frequency. Implementation priority was recommended to begin with data-layer foundations that improved predictable accounting query and write behavior, because efficient indexing and storage-aware structures reduced both typical and peak delays by lowering scan costs, minimizing contention duration, and stabilizing retrieval during close-intensive reporting activity. Index governance was recommended to be standardized around accounting access patterns, including entity filters, fiscal-period filters, and account rollups, while storage practices were recommended to maintain stable append-oriented write behavior and disciplined retention controls that preserved verifiable access to historical evidence without degrading performance for active periods. Process-layer configuration was recommended to be tuned using workload evidence rather than static settings, because throughput benefits were strongest where posting pipelines supported scalable queue scheduling, controlled batching discipline, and safe

parallelization that reduced backlogs under peak demand. Posting modes were recommended to be aligned to operational reporting needs, with batch windows designed to limit waiting time while still providing stable capacity during high-volume windows, and with replication or read-model strategies applied to reduce contention between reporting queries and posting operations. Control-layer practices were recommended as non-negotiable for accuracy improvement, given the strong associations with mismatch and correction outcomes. Validation rule coverage was recommended to be mapped to transaction classes and master-data dependencies, with monitoring that distinguished high-risk anomalies from low-risk noise to reduce false-positive burden while maintaining strong detection. Provenance completeness was recommended to be operationalized as a measurable standard, with required references for source documents, approvals, transformation lineage, and identifiers that enabled fast drill-down during reconciliation and audit activities. Automated matching was recommended to be expanded where consistent identifiers existed, particularly for bank, procure-to-pay, and intercompany contexts, while standardization of reference keys was recommended to improve match success and reduce exception recurrence. Workflow and segregation enforcement were recommended to be configured with risk-based thresholds to preserve governance without introducing avoidable delays, and routing designs were recommended to include clear exception paths that reduced cycle-time inflation during close. Because moderation evidence indicated that strict validation reduced net speed gains from batching, joint tuning was recommended in which validation strictness and posting configuration were optimized together, ensuring that controls remained strong while pipeline delays were kept within service targets. Higher automation maturity strengthened optimization effects on accuracy, so capability-building was recommended in monitoring, exception resolution automation, reconciliation checks, and feedback loops that converted recurring exceptions into updated rules and improved master-data discipline. Continuous measurement was recommended through standardized dashboards tracking median and tail latency, throughput, mismatch incidence and value, exception rates, correction entry activity, documentation completeness, and trace time per exception, enabling governance to be evaluated quantitatively and adjusted based on operational evidence rather than subjective assessments.

#### **LIMITATIONS**

Limitations of the study were recognized in relation to research design choices, measurement boundaries, and the practical constraints inherent in using operational accounting system evidence to model digital ledger optimization effects on transaction speed and reporting accuracy. The quantitative design relied on archival logs, reconciliation outputs, and correction journals that were generated during normal system operation, which strengthened objectivity but constrained the ability to establish definitive causal attribution in all relationships. Although the models controlled for transaction volume, system and entity complexity, currency count, reporting cadence, and automation maturity, unobserved factors such as differences in local accounting policies, staff expertise, exception-handling culture, and change-management discipline could have influenced both optimization adoption and observed outcomes. The operationalization of optimization intensity was constructed from measurable indicators of data-layer, process-layer, and control-layer practices, but measurement error could have arisen where some optimizations were partially implemented, inconsistently configured across modules, or applied differently across transaction classes, leading to within-unit heterogeneity that could not be fully captured by aggregated unit-period scores. Similarly, transaction speed measures depended on timestamp consistency across applications, middleware, and databases; clock drift, asynchronous logging, or missing event markers could have introduced noise into latency and queue-time calculations, particularly during peak windows when systems experienced high load and logging could be throttled or delayed. Reporting accuracy measures also reflected practical constraints because error rates, mismatch rates, and correction entries were derived from system classifications and reconciliation procedures that varied across organizations; some correction entries may have represented legitimate accounting judgment or routine allocations rather than system-driven defects, and even when filtering rules were applied, perfect separation between corrective rework and policy-based adjustments could not be guaranteed. Auditability proxies such as documentation completeness and trace time per exception were subject to availability limitations, because not all environments tracked trace time consistently and documentation fields could be completed outside the system

through external repositories, which may have resulted in underestimation of evidentiary richness in some cases. The sample structure, while adequate for panel-style modeling, may have reflected selection constraints because cases were included based on data accessibility and comparability, which could have biased the sample toward more instrumented and governance-mature environments. In addition, the study treated the ledger environment as the core analytical boundary, but some determinants of speed and accuracy may have originated upstream in source systems or downstream in reporting warehouses, meaning that effects attributed to ledger optimization could partially reflect external integration quality or reporting-layer design not fully measured in the dataset. Finally, the findings were interpreted within the observed range of optimization maturity and workload conditions represented in the sample; environments with radically different architectures, regulatory constraints, or transaction typologies might display different performance–quality relationships, limiting generalizability beyond similar accounting system contexts.

## REFERENCES

- [1]. Ali, Z. S., Subramanian, N., & Erbad, A. (2020). Smart health monitoring for seizure detection using mobile edge computing. 2020 International Wireless Communications and Mobile Computing (IWCMC),
- [2]. Ang, K. L.-M., Ge, F. L., & Seng, K. P. (2020). Big educational data & analytics: Survey, architecture and challenges. *Ieee Access*, 8, 116392-116414.
- [3]. Appelbaum, D., Kogan, A., Vasarhelyi, M., & Yan, Z. (2017). Impact of business analytics and enterprise systems on managerial accounting. *International Journal of Accounting Information Systems*, 25, 29-44.
- [4]. Aspers, P., & Corte, U. (2019). What is qualitative in qualitative research. *Qualitative sociology*, 42(2), 139-160.
- [5]. Battah, A. A., Madine, M. M., Alzaabi, H., Yaqoob, I., Salah, K., & Jayaraman, R. (2020). Blockchain-based multi-party authorization for accessing IPFS encrypted data. *Ieee Access*, 8, 196813-196825.
- [6]. Bellini, E., Iraqi, Y., & Damiani, E. (2020). Blockchain-based distributed trust and reputation management systems: A survey. *Ieee Access*, 8, 21127-21151.
- [7]. Betti, Q., Montreuil, B., Khoury, R., & Hallé, S. (2020). Smart contracts-enabled simulation for hyperconnected logistics. In *Decentralised internet of things: A blockchain perspective* (pp. 109-149). Springer.
- [8]. Bica, I., Chifor, B.-C., Arseni, Ş.-C., & Matei, I. (2019). Multi-layer IoT security framework for ambient intelligence environments. *Sensors*, 19(18), 4038.
- [9]. Bogdanov, A., Degtyarev, A., Shchegoleva, N., & Khvatov, V. (2020). Data quality in a decentralized environment. International Conference on Computational Science and Its Applications,
- [10]. Booth, A., Noyes, J., Flemming, K., Gerhardus, A., Wahlster, P., van der Wilt, G. J., Mozygamba, K., Refolo, P., Sacchini, D., & Tummers, M. (2018). Structured methodology review identified seven (RETREAT) criteria for selecting qualitative evidence synthesis approaches. *Journal of clinical epidemiology*, 99, 41-52.
- [11]. Brownfield, K., & Wilkinson, I. A. (2018). Examining the impact of scaffolding on literacy learning: A critical examination of research and guidelines to advance inquiry. *International Journal of Educational Research*, 90, 177-190.
- [12]. Cai, H., Xu, B., Jiang, L., & Vasilakos, A. V. (2016). IoT-based big data storage systems in cloud computing: perspectives and challenges. *IEEE Internet of Things Journal*, 4(1), 75-87.
- [13]. Cao, H., Li, R., Tian, W., Xu, Z., & Xiao, W. (2020). Blockchain-based accountability for multi-party oblivious RAM. *Journal of Parallel and Distributed Computing*, 137, 224-237.
- [14]. Carrozzo, G., Siddiqui, M. S., Betzler, A., Bonnet, J., Perez, G. M., Ramos, A., & Subramanya, T. (2020). AI-driven zero-touch operations, security and trust in multi-operator 5G networks: A conceptual architecture. 2020 European conference on networks and communications (EuCNC),
- [15]. Cascella, C., Giberti, C., & Bolondi, G. (2020). An analysis of Differential Item Functioning on INVALSI tests, designed to explore gender gap in mathematical tasks. *Studies in Educational Evaluation*, 64, 100819.
- [16]. Chege, S. M., & Wang, D. (2020). Information technology innovation and its impact on job creation by SMEs in developing countries: an analysis of the literature review. *Technology Analysis & Strategic Management*, 32(3), 256-271.
- [17]. Chua, Z. Y., Ahn, I. H., & Moon, S. K. (2017). Process monitoring and inspection systems in metal additive manufacturing: Status and applications. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 4(2), 235-245.
- [18]. Cornetta, G., Mateos, J., Touhafi, A., & Muntean, G.-M. (2019). Design, simulation and testing of a cloud platform for sharing digital fabrication resources for education. *Journal of Cloud Computing*, 8(1), 12.
- [19]. D'Angelo, P., & D'Angelo, P. (2018). *Doing news framing analysis II*. Routledge.
- [20]. Delzendeh, E., Wu, S., Lee, A., & Zhou, Y. (2017). The impact of occupants' behaviours on building energy analysis: A research review. *Renewable and sustainable energy reviews*, 80, 1061-1071.
- [21]. Dhillion, V., Metcalf, D., & Hooper, M. (2017). Ethereum tokens: high-performance computing. In *Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You* (pp. 79-109). Springer.
- [22]. Du, M., Chen, Q., Chen, J., & Ma, X. (2020). An optimized consortium blockchain for medical information sharing. *IEEE Transactions on Engineering Management*, 68(6), 1677-1689.
- [23]. Dziembowski, S., Eckey, L., Faust, S., Hesse, J., & Hostáková, K. (2019). Multi-party virtual state channels. Annual international conference on the theory and applications of cryptographic techniques,

- [24]. Elangovan, D., Long, C. S., Bakrin, F. S., Tan, C. S., Goh, K. W., Hussain, Z., Al-Worafi, Y. M., Lee, K. S., Kassab, Y. W., & Ming, L. C. (2020). Application of blockchain technology in hospital information system. *Mathematical modeling and soft computing in epidemiology*, 231-246.
- [25]. Fan, C., Ghaemi, S., Khazaei, H., & Musilek, P. (2020). Performance evaluation of blockchain systems: A systematic survey. *Ieee Access*, 8, 126927-126950.
- [26]. Fei, Y., Wang, K. C., Zhang, A., Chen, C., Li, J. Q., Liu, Y., Yang, G., & Li, B. (2019). Pixel-level cracking detection on 3D asphalt pavement images through deep-learning-based CrackNet-V. *IEEE Transactions on Intelligent Transportation Systems*, 21(1), 273-284.
- [27]. Feng, Y., Zhu, Q., & Lai, K.-H. (2017). Corporate social responsibility for supply chain management: A literature review and bibliometric analysis. *Journal of cleaner production*, 158, 296-307.
- [28]. Fikri, N., Rida, M., Abghour, N., Moussaid, K., & El Omri, A. (2019). An adaptive and real-time based architecture for financial data integration. *Journal of Big Data*, 6(1), 97.
- [29]. Finch, T. L., Girling, M., May, C. R., Mair, F. S., Murray, E., Treweek, S., McColl, E., Steen, I. N., Cook, C., & Vernazza, C. R. (2018). Improving the normalization of complex interventions: part 2-validation of the NoMAD instrument for assessing implementation work based on normalization process theory (NPT). *BMC medical research methodology*, 18(1), 135.
- [30]. Gaikwad, A., Yavari, R., Montazeri, M., Cole, K., Bian, L., & Rao, P. (2020). Toward the digital twin of additive manufacturing: Integrating thermal simulations, sensing, and analytics to detect process faults. *Iise Transactions*, 52(11), 1204-1217.
- [31]. Gao, H., Ma, Z., Luo, S., & Wang, Z. (2019). BFR-MPC: a blockchain-based fair and robust multi-party computation scheme. *Ieee Access*, 7, 110439-110450.
- [32]. Gleim, L., Pennekamp, J., Liebenberg, M., Buchsbaum, M., Niemietz, P., Knape, S., Epple, A., Storms, S., Trauth, D., & Bergs, T. (2020). FactDAG: formalizing data interoperability in an internet of production. *IEEE Internet of Things Journal*, 7(4), 3243-3253.
- [33]. Grüne-Yanoff, T. (2016). Framing. In *The argumentative turn in policy analysis: Reasoning about uncertainty* (pp. 189-215). Springer.
- [34]. Gudivada, V. N., Irfan, M. T., Fathi, E., & Rao, D. L. (2016). Cognitive analytics: Going beyond big data analytics and machine learning. In *Handbook of statistics* (Vol. 35, pp. 169-205). Elsevier.
- [35]. Harris, J. L., Booth, A., Cargo, M., Hannes, K., Harden, A., Flemming, K., Garside, R., Pantoja, T., Thomas, J., & Noyes, J. (2018). Cochrane Qualitative and Implementation Methods Group guidance series – paper 2: methods for question formulation, searching, and protocol development for qualitative evidence synthesis. *Journal of clinical epidemiology*, 97, 39-48.
- [36]. Hassan, L. M., Shiu, E., & Shaw, D. (2016). Who says there is an intention-behaviour gap? Assessing the empirical evidence of an intention-behaviour gap in ethical consumption. *Journal of business ethics*, 136(2), 219-236.
- [37]. Hernández-Nieves, E., Hernández, G., Gil-González, A.-B., Rodríguez-González, S., & Corchado, J. M. (2020). Fog computing architecture for personalized recommendation of banking products. *Expert Systems with Applications*, 140, 112900.
- [38]. Homoliak, I., Venugopalan, S., Reijnsbergen, D., Hum, Q., Schumi, R., & Szalachowski, P. (2020). The security reference architecture for blockchains: Toward a standardized model for studying vulnerabilities, threats, and defenses. *IEEE Communications Surveys & Tutorials*, 23(1), 341-390.
- [39]. Huang, F., & Vasarhelyi, M. A. (2019). Applying robotic process automation (RPA) in auditing: A framework. *International Journal of Accounting Information Systems*, 35, 100433.
- [40]. Hughes, P. (2020). Paradigms, methods and knowledge. In *Doing early childhood research* (pp. 31-55). Routledge.
- [41]. Hunhevicz, J. J., & Hall, D. M. (2020). Do you need a blockchain in construction? Use case categories and decision framework for DLT design options. *Advanced Engineering Informatics*, 45, 101094.
- [42]. Ivanov, D., Dolgui, A., Sokolov, B., & Ivanova, M. (2017). Literature review on disruption recovery in the supply chain. *International Journal of Production Research*, 55(20), 6158-6174.
- [43]. Jack, J. P., & Thompson, P. W. (2017). 4 quantitative reasoning and the development of algebraic reasoning. In *Algebra in the early grades* (pp. 95-132). Routledge.
- [44]. Junius-Walker, U., Onder, G., Soleymani, D., Wiese, B., Albaina, O., Bernabei, R., & Marzetti, E. (2018). The essence of frailty: a systematic review and qualitative synthesis on frailty concepts and definitions. *European journal of internal medicine*, 56, 3-10.
- [45]. Kamiński, A. (2019). Interoperability of manufacturing information systems. Contemporary Challenges in Cooperation and Coopetition in the Age of Industry 4.0: 10th Conference on Management of Organizations' Development (MOD),
- [46]. Kaur, R., Singh, A., Singh, S., & Sharma, S. (2016). Security of software defined networks: Taxonomic modeling, key components and open research area. 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT),
- [47]. Kloeckner, K., Davis, J., Fuller, N. C., Lanfranchi, G., Pappé, S., Paradkar, A., Shwartz, L., Surendra, M., & Wiesmann, D. (2018). *Transforming the IT Services Lifecycle with AI Technologies*. Springer.
- [48]. Kokina, J., & Blanchette, S. (2019). Early evidence of digital labor in accounting: Innovation with Robotic Process Automation. *International Journal of Accounting Information Systems*, 35, 100431.
- [49]. Kornmesser, S. (2016). A frame-based approach for theoretical concepts. *Synthese*, 193(1), 145-166.
- [50]. Lei, H., Yan, Y., Bao, Z., Wang, Q., Zhang, Y., & Shi, W. (2020). SDSBT: A secure multi-party data sharing platform based on blockchain and TEE. International Symposium on Cyberspace Safety and Security,

- [51]. Li, H., Dai, J., Gershberg, T., & Vasarhelyi, M. A. (2018). Understanding usage and value of audit analytics for internal auditors: An organizational approach. *International Journal of Accounting Information Systems*, 28, 59-76.
- [52]. Li, M.-W., Teng, H.-Y., & Chen, C.-Y. (2020). Unlocking the customer engagement-brand loyalty relationship in tourism social media: The roles of brand attachment and customer trust. *Journal of Hospitality and Tourism Management*, 44, 184-192.
- [53]. Lim, K. Y. H., Zheng, P., Chen, C.-H., & Huang, L. (2020). A digital twin-enhanced system for engineering product family design and optimization. *Journal of Manufacturing Systems*, 57, 82-93.
- [54]. Liu, Q., Qin, S. J., & Chai, T. (2017). Unevenly sampled dynamic data modeling and monitoring with an industrial application. *IEEE Transactions on Industrial Informatics*, 13(5), 2203-2213.
- [55]. Luo, Y., Xu, Z., Liu, P., Du, Y., & Guo, J.-M. (2018). Multi-person pose estimation via multi-layer fractal network and joints kinship pattern. *IEEE Transactions on Image Processing*, 28(1), 142-155.
- [56]. Mackintosh, M., Epiphaniou, G., Al-Khateeb, H., Burnham, K., Pillai, P., & Hammoudeh, M. (2019). Preliminaries of orthogonal layered defence using functional and assurance controls in industrial control systems. *Journal of Sensor and Actuator Networks*, 8(1), 14.
- [57]. Madine, M. M., Salah, K., Jayaraman, R., Yaqoob, I., Al-Hammadi, Y., Ellahham, S., & Calyam, P. (2020). Fully decentralized multi-party consent management for secure sharing of patient health records. *Ieee Access*, 8, 225777-225791.
- [58]. Mahmud, S. S., Ferreira, L., Hoque, M. S., & Tavassoli, A. (2017). Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. *IATSS research*, 41(4), 153-163.
- [59]. Marbough, D., Abbasi, T., Maasmi, F., Omar, I. A., Debe, M. S., Salah, K., Jayaraman, R., & Ellahham, S. (2020). Blockchain for COVID-19: review, opportunities, and a trusted tracking system. *Arabian journal for science and engineering*, 45(12), 9895-9911.
- [60]. Mazlan, A. A., Daud, S. M., Sam, S. M., Abas, H., Rasid, S. Z. A., & Yusof, M. F. (2020). Scalability challenges in healthcare blockchain system—a systematic review. *Ieee Access*, 8, 23663-23673.
- [61]. Merom, D., & John, J. R. (2019). Measurement issues in quantitative research. In *Handbook of research methods in health social sciences* (pp. 663-679). Springer.
- [62]. Münsing, E., Mather, J., & Moura, S. (2017). Blockchains for decentralized optimization of energy resources in microgrid networks. 2017 IEEE conference on control technology and applications (CCTA),
- [63]. Neshenko, N., Nader, C., Bou-Harb, E., & Furht, B. (2020). A survey of methods supporting cyber situational awareness in the context of smart cities. *Journal of Big Data*, 7(1), 92.
- [64]. Nguyen, T. T., Mia, L., Winata, L., & Chong, V. K. (2017). Effect of transformational-leadership style and management control system on managerial performance. *Journal of business research*, 70, 202-213.
- [65]. Oraee, M., Hosseini, M. R., Papadonikolaki, E., Palliyaguru, R., & Arashpour, M. (2017). Collaboration in BIM-based construction networks: A bibliometric-qualitative literature review. *International journal of project management*, 35(7), 1288-1301.
- [66]. Paik, H.-Y., Xu, X., Bandara, H. D., Lee, S. U., & Lo, S. K. (2019). Analysis of data management in blockchain-based systems: From architecture to governance. *Ieee Access*, 7, 186091-186107.
- [67]. Pan, G., & Seow, P.-S. (2016). Preparing accounting graduates for digital revolution: A critical review of information technology competencies and skills development. *Journal of Education for business*, 91(3), 166-175.
- [68]. Park, S., & Kim, H. (2019). DAG-based distributed ledger for low-latency smart grid network. *Energies*, 12(18), 3570.
- [69]. Paul, S., & Shrivastava, A. (2019). Efficient fair multiparty protocols using Blockchain and trusted hardware. *International Conference on Cryptology and Information Security in Latin America*,
- [70]. Pei, D., & Vasarhelyi, M. A. (2020). Big data and algorithmic trading against periodic and tangible asset reporting: The need for U-XBRL. *International Journal of Accounting Information Systems*, 37, 100453.
- [71]. Pei, X., Sun, L., Li, X., Zheng, K., & Wu, X. (2018). Smart contract based multi-party computation with privacy preserving and settlement addressed. 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4),
- [72]. Polak, P., Nelischer, C., Guo, H., & Robertson, D. C. (2020). "Intelligent" finance and treasury management: what we can expect. *Ai & Society*, 35(3), 715-726.
- [73]. Pundir, A. K., Jagannath, J. D., Chakraborty, M., & Ganpathy, L. (2019). Technology integration for improved performance: A case study in digitization of supply chain with integration of internet of things and blockchain technology. 2019 IEEE 9th annual computing and communication workshop and conference (CCWC),
- [74]. Quintana, C., Reiser, B. J., Davis, E. A., Krajcik, J., Fretz, E., Duncan, R. G., Kyza, E., Edelson, D., & Soloway, E. (2018). A scaffolding design framework for software to support science inquiry. In *Scaffolding* (pp. 337-386). Psychology Press.
- [75]. Rác-z-Szabó, A., Ruppert, T., Bántay, L., Löcklin, A., Jakab, L., & Abonyi, J. (2020). Real-time locating system in production management. *Sensors*, 20(23), 6766.
- [76]. Rahman, A., Nasir, M. K., Rahman, Z., Mosavi, A., & Minaei-Bidgoli, B. (2020). Distblockbuilding: A distributed blockchain-based sdn-iot network for smart building management. *Ieee Access*, 8, 140008-140018.
- [77]. Raman, R. K., Vaculin, R., Hind, M., Remy, S. L., Pissadaki, E. K., Bore, N. K., Daneshvar, R., Srivastava, B., & Varshney, K. R. (2019). A scalable blockchain approach for trusted computation and verifiable simulation in multi-party collaborations. 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC),
- [78]. Rathore, H., Wenzel, L., Al-Ali, A. K., Mohamed, A., Du, X., & Guizani, M. (2018). Multi-layer perceptron model on chip for secure diabetic treatment. *Ieee Access*, 6, 44718-44730.

- [79]. Razzak, M. I., Imran, M., & Xu, G. (2020). Big data analytics for preventive medicine. *Neural Computing and Applications*, 32(9), 4417-4451.
- [80]. Ribeiro, J. P., & Barbosa-Povoa, A. (2018). Supply Chain Resilience: Definitions and quantitative modelling approaches—A literature review. *Computers & industrial engineering*, 115, 109-122.
- [81]. Roetzel, P. G. (2019). Information overload in the information age: a review of the literature from business administration, business psychology, and related disciplines with a bibliometric approach and framework development. *Business research*, 12(2), 479-522.
- [82]. Roman, D., & Vu, K. (2018). Enabling data markets using smart contracts and multi-party computation. *International Conference on Business Information Systems*,
- [83]. Rönnebeck, S., Bernholt, S., & Ropohl, M. (2016). Searching for a common ground—A literature review of empirical research on scientific inquiry activities. *Studies in science education*, 52(2), 161-197.
- [84]. Saghiri, A. M., HamlAbadi, K. G., & Vahdati, M. (2019). The internet of things, artificial intelligence, and blockchain: implementation perspectives. In *Advanced applications of blockchain technology* (pp. 15-54). Springer.
- [85]. Salahshour Rad, M., Nilashi, M., & Mohamed Dahlan, H. (2018). Information technology adoption: a review of the literature and classification. *Universal Access in the Information Society*, 17(2), 361-390.
- [86]. Sandner, P., Lange, A., & Schulden, P. (2020). The role of the CFO of an industrial company: an analysis of the impact of blockchain technology. *Future Internet*, 12(8), 128.
- [87]. Sarkar, S., & Searcy, C. (2016). Zeitgeist or chameleon? A quantitative analysis of CSR definitions. *Journal of cleaner production*, 135, 1423-1435.
- [88]. Sengupta, A., Mohanty, S. P., Pescador, F., & Corcoran, P. (2018). Multi-phase obfuscation of fault secured DSP designs with enhanced security feature. *IEEE Transactions on Consumer Electronics*, 64(3), 356-364.
- [89]. Shae, Z., & Tsai, J. (2018). Transform blockchain into distributed parallel computing architecture for precision medicine. 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS),
- [90]. Shardlow, M., Batista-Navarro, R., Thompson, P., Nawaz, R., McNaught, J., & Ananiadou, S. (2018). Identification of research hypotheses and new knowledge from scientific literature. *BMC medical informatics and decision making*, 18(1), 46.
- [91]. Shen, M., Zhu, L., & Xu, K. (2020). Layered Data Sharing Architecture with Blockchain. In *Blockchain: Empowering Secure Data Sharing* (pp. 29-37). Springer.
- [92]. Sheng, D., Ding, L., Zhong, B., Love, P. E., Luo, H., & Chen, J. (2020). Construction quality information management with blockchains. *Automation in construction*, 120, 103373.
- [93]. Shu, Z., Wan, J., Li, D., Lin, J., Vasilakos, A. V., & Imran, M. (2016). Security in software-defined networking: Threats and countermeasures. *Mobile Networks and Applications*, 21(5), 764-776.
- [94]. Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review*, 22, 142-158.
- [95]. Silvano, W. F., & Marcelino, R. (2020). Iota Tangle: A cryptocurrency to communicate Internet-of-Things data. *Future generation computer systems*, 112, 307-319.
- [96]. Sledgianowski, D., Gomaa, M., & Tan, C. (2017). Toward integration of Big Data, technology and information systems competencies into the accounting curriculum. *Journal of Accounting Education*, 38, 81-93.
- [97]. Suárez, E., Calvo-Mora, A., Roldán, J. L., & Periañez-Cristóbal, R. (2017). Quantitative research on the EFQM excellence model: A systematic literature review (1991–2015). *European Research on Management and Business Economics*, 23(3), 147-156.
- [98]. Sutjarittham, T., Gharakheili, H. H., Kanhere, S. S., & Sivaraman, V. (2020). Monetizing parking IoT data via demand prediction and optimal space sharing. *IEEE Internet of Things Journal*, 9(8), 5629-5644.
- [99]. Tamburri, D. A. (2019). Software architecture social debt: Managing the incommunicability factor. *IEEE transactions on computational social systems*, 6(1), 20-37.
- [100]. Tarafdar, M., & Qrunfleh, S. (2017). Agile supply chain strategy and supply chain performance: complementary roles of supply chain practices and information systems capability for agility. *International Journal of Production Research*, 55(4), 925-938.
- [101]. Timans, R., Wouters, P., & Heilbron, J. (2019). Mixed methods research: what it is and what it could be. *Theory and society*, 48(2), 193-216.
- [102]. Tolpygo, S. K., Bolkhovskiy, V., Rastogi, R., Zarr, S., Day, A. L., Golden, E., Weir, T. J., Wynn, A., & Johnson, L. M. (2019). Advanced fabrication processes for superconductor electronics: Current status and new developments. *IEEE Transactions on Applied Superconductivity*, 29(5), 1-13.
- [103]. Vahdati, M., HamlAbadi, K. G., Saghiri, A. M., & Rashidi, H. (2018). A self-organized framework for insurance based on internet of things and blockchain. 2018 IEEE 6th international conference on future internet of things and cloud (FiCloud),
- [104]. Viriyasitavat, W., Da Xu, L., Bi, Z., & Pungpapong, V. (2019). Blockchain and internet of things for modern business process in digital economy – the state of the art. *IEEE transactions on computational social systems*, 6(6), 1420-1432.
- [105]. Walter, M., & Suina, M. (2019). Indigenous data, indigenous methodologies and indigenous data sovereignty. *International journal of social research methodology*, 22(3), 233-243.
- [106]. Wang, L., & Zhao, J. (2020). *Strategic Blueprint for Enterprise Analytics*. Springer.
- [107]. Wang, Y., & Kogan, A. (2018). Designing confidentiality-preserving Blockchain-based transaction processing systems. *International Journal of Accounting Information Systems*, 30, 1-18.
- [108]. Weng, W.-Y., Lin, Y.-R., & She, H.-C. (2017). Scaffolding for argumentation in hypothetical and theoretical biology concepts. *International Journal of Science Education*, 39(7), 877-897.

- [109]. Wu, J., Xiong, F., & Li, C. (2019). Application of Internet of Things and blockchain technologies to improve accounting information quality. *Ieee Access*, 7, 100090-100098.
- [110]. Xu, S., Fu, X., Cao, J., Liu, B., & Wang, Z. (2020). Survey on user location prediction based on geo-social networking data. *World Wide Web*, 23(3), 1621-1664.
- [111]. Yang, J., Bi, H., Liang, Z., Zhou, H., & Yang, H. (2020). A survey on blockchain: Architecture, applications, challenges, and future trends. 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics),
- [112]. Yasmin, A., & Kamalakkannan, S. (2020). PACELC: Enchantment multi-dimension TensorFlow for value creation through Big Data. 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC),
- [113]. Yin, B., Yin, H., Wu, Y., & Jiang, Z. (2020). FDC: A secure federated deep learning mechanism for data collaborations in the Internet of Things. *IEEE Internet of Things Journal*, 7(7), 6348-6359.
- [114]. Yoon, S. (2020). A study on the transformation of accounting based on new technologies: Evidence from Korea. *Sustainability*, 12(20), 8669.
- [115]. Yu, B., Liu, J., Zhou, X., & An, B. (2020). Simulation verification platform for complex facility control system. 2020 Chinese Automation Congress (CAC),
- [116]. Zhang, C., & Jiang, P. (2019). RFID-driven energy-efficient control approach of CNC machine tools using deep belief networks. *IEEE Transactions on Automation Science and Engineering*, 17(1), 129-141.
- [117]. Zhang, G., Li, T., Li, Y., Hui, P., & Jin, D. (2018). Blockchain-based data sharing system for ai-powered network operations. *Journal of Communications and Information Networks*, 3(3), 1-8.
- [118]. Zhang, W., Chen, R.-S., Chen, Y.-C., Lu, S.-Y., Xiong, N., & Chen, C.-M. (2019). An effective digital system for intelligent financial environments. *Ieee Access*, 7, 155965-155976.
- [119]. Zhang, Y., Xiong, F., Xie, Y., Fan, X., & Gu, H. (2020). The impact of artificial intelligence and blockchain on the accounting profession. *Ieee Access*, 8, 110461-110477.
- [120]. Zhijun, W., Wenjing, L., Liang, L., & Meng, Y. (2020). Low-rate DoS attacks, detection, defense, and challenges: A survey. *Ieee Access*, 8, 43920-43943.
- [121]. Zhong, H., Sang, Y., Zhang, Y., & Xi, Z. (2019). Secure multi-party computation on blockchain: An overview. International symposium on parallel architectures, algorithms and programming,
- [122]. Zia, M. F., Benbouzid, M., Elbouchikhi, E., Muyeen, S., Techato, K., & Guerrero, J. M. (2020). Microgrid transactive energy: Review, architectures, distributed ledger technologies, and market analysis. *Ieee Access*, 8, 19410-19432.
- [123]. Zichichi, M., Ferretti, S., & D'angelo, G. (2020). A framework based on distributed ledger technologies for data management and services in intelligent transportation systems. *Ieee Access*, 8, 100384-100402.