International Journal of Safety and Security Engineering
Vol. 16, No. 7, July, 2026, pp. 1563-1574
Journal homepage: http://iieta.org/journals/ijsse

Design of an Integrated Multi-Stage Analytical Method for Energy-Aware, Trust-Optimized, and Fast Blockchain Finality Using Polytope-Driven Consensus

Ramya R B*, Deepa Parasar

Amity School of Engineering and Technology, Amity University Maharashtra, Mumbai 410206, India

Corresponding Author Email: ramya.b@s.amity.edu

https://doi.org/10.18280/ijsse.160713

Received: 2 June 2026
Revised: 19 July 2026
Accepted: 27 July 2026
Available online: 31 July 2026

Keywords:
blockchain consensus, spectral embedding, quorum polytope, zero-knowledge attestation, fast finality

ABSTRACT

This study presents an integrated analytical consensus framework for reducing latency, controlling energy use, and preserving verifiable accountability in blockchain networks. The framework combines five linked modules: Latency-Trust Spectral Embedding (LaTSE) for topology-aware validator representation, Intersecting Quorum Polytope Design (IQPD) for communication-efficient quorum construction, Risk-Budgeted Leader and Relay Rotation (RISKFLOW-LR) for timeout-aware scheduling, Zero-Knowledge Compressed Energy and Availability Attestation (ZK-CEAA) for confidential compliance verification, and Polytope-Certified Fast Finality (PCFF) for final block commitment. The evaluation is conducted in a simulated 200-validator network distributed across five geographic regions using CAIDA AS-level topology characteristics, Google cluster workload-inspired energy traces, two transaction workloads representing IoT and financial trading, and Byzantine validator ratios up to 33%. The proposed model is compared with representative Proof-of-Work (PoW) / Internet of Things (IoT)-fog, Delegated Proof of Stake (DPoS)-Byzantine Fault Tolerance (BFT), and hybrid BFT-style consensus baselines. Across 30 stratified simulation runs, the proposed pipeline achieves 0.68–0.72 s commit latency, 1.15–1.40 kJ block-attributable energy, 2100–3200 TPS, 1.1–1.3% timeout rates, and 4.8–5.1 ms verification overhead. External studies are used only to define reference-informed analytical protocol-family templates; throughput and timeout rankings against these studies are not claimed without matched run-level implementations. The results support a simulation-validated design for energy-aware and fast-finality consensus, while deployment-scale validation remains future work.

1. INTRODUCTION

Decentralized apps have made apparent the limitations of currently used consensus algorithms. Proof-of-Work (PoW) consensus offers open participation; however, it consumes substantial energy, while Proof-of-Stake (PoS) reduces energy demand but may concentrate decision power under skewed stake distributions and still produce latency. On the other hand, Byzantine Fault Tolerance (BFT) algorithms suffer from high communication costs. Currently available methods of reaching agreement in decentralized systems focus on only one or two characteristics, for instance, time, power, or trust. In that respect, the proposed method offers five stages aimed at achieving fast, power-efficient and formally correct blockchain consensus. The approach is based on using spectral embedding, geometric quorum systems, risk-based leadership rotation, and zero-knowledge attestation. PoW has historically been important because it allowed for secure peer-to-peer transactions without a central authority. Miners use complex cryptographic operations in the process of mining the currency. With each new miner joining the chain, energy consumed grows exponentially while its safety does not increase correspondingly. This creates inefficiency for the economy and environment. PoS consumes comparatively lower energy, although it produces other challenges such as stake centralization, liveness failure in the partially synchronous environment and unfairness of leadership selection. However, Byzantine fault-tolerant approaches are highly scalable, but their main drawbacks lie in high communication, high latency, and poor scalability. As a result, the adoption of these types of consensus models by the large validator community would be impossible. There are two key shortcomings inherent in all of these approaches, which include the absence of a general approach to handling both trust and latency and the inability to formally enforce energy accountability. Currently, most of these approaches solve one of the mentioned problems, while failing to tackle others, resulting in significant vulnerabilities in the entire system. Given the rapid growth of various decentralized applications, such as payment networks, smart cities, and IoT infrastructure, it is crucial to have an approach that resolves these problems. This paper proposes a rigorous framework that compensates for these deficiencies using five novel analytical tools. The Latency-Trust Spectral Embedding (LaTSE) algorithm embeds the validators into a geometry in the latency-trust-energy space, providing the foundation for efficient routing and neighbor selection. Intersecting Quorum Polytope Design (IQPD) is a method of constructing the quorums as

intersecting polytopes using the geometry obtained from the LaTSE algorithm. Risk-Budgeted Leader and Relay Rotation (RISKFLOW-LR) transforms those quorum structures into leader and relay schedules, risk-budgeted through consensus design, with no congestion and reduced timeouts. Zero-Knowledge Compressed Energy and Availability Attestation (ZK-CEAA) curtails energy and availability accountability through succinct, verifiable proofs, preserving accountability without heavy communication. Since the modules are coupled, the framework reduces scenario-level latency and block-attributable energy while preserving the stated quorum-intersection safety assumptions and an 80-bit ZK proof-system soundness setting. The contributions go beyond incremental improvements leading to a coherent consensus mechanism where geometry, optimization, and cryptography intersect, paving the way for the next generation blockchain systems that are scalable and sustainable.

2. REVIEW OF EXISTING MODELS USED FOR BLOCKCHAIN CONSENSUS

Recent blockchain-consensus research is better understood through the design problems it addresses than through a sequence of isolated protocols. One major stream concerns energy and resource efficiency in cyber-physical and edge environments. Recent research has addressed energy and resource efficiency in blockchain systems through diverse approaches. These include renewable-energy regulation, IoT-fog integration, and Software-Defined Networking-assisted energy optimization. Additionally, they integrate specialized consensus mechanisms such as smart-grid weighted consensus, trusted-work consensus, lightweight IIoT validation, energy-aware Proof-of-Authority, and resource-constrained blockchain deployment. All these approaches attempt to reduce computation, communication, or power demand while preserving integrity [1-8]. These approaches show that consensus can be adapted to constrained devices and domain workloads, yet most rely on workload-specific optimizations or offloading. Energy is generally measured as an operational cost; it is rarely enforced as a privately verifiable protocol obligation.

A second stream targets scalability and deterministic finality within BFT-derived systems. Various approaches such as Hierarchical PBFT, improved Delegated Proof of Stake (DPoS)-BFT, voting-cluster PBFT, hybrid sharding consensus, and pipelined or overlapped BFT seek to reduce all-to-all communication or divide validation across committees [9-13]. The collective lesson is that hierarchical communication and smaller committees can improve throughput, but the benefit depends on committee composition, inter-layer coordination, and leader stability. Existing work usually specifies quorum size algebraically; it does not construct quorums from a joint geometry of network delay, energy condition, and historical reliability, nor does it provide an explicit geometric certificate that independently selected quorums retain the required overlap.

Trust, fairness, and validator selection form a third research theme. Network-aware reputation, information-centric reputation, equitable node selection, and reputation-based leader election attempt to reduce concentration and discourage unreliable validation [14-17]. Reputation-driven PBFT and incentive design have also been applied to blockchain-supported healthcare services [18]. These studies improve the social or behavioural basis of leader selection, but trust is commonly represented as a scalar score that is updated separately from topology and energy. Consequently, a highly reputable validator may still be selected when its network path is congested or its energy budget is stressed, producing avoidable timeouts and uneven operational load.

Domain-specific protocols demonstrate the breadth of consensus requirements. Supply chain security and access-control mechanisms emphasize provenance, privacy, and tamper resistance [19, 20], while blockchain-assisted medical-data sharing and intelligent patient monitoring require secure availability and controlled access [21, 22]. Federated IoT security uses blockchain to validate decentralized model contributions [23], and banking identity verification systems combine distributed verification with transaction assurance [24]. Privacy-aware mobile-edge blockchain designs further show the importance of protecting participant information close to the data source [25]. These systems validate the need for application-sensitive consensus, although their specialized assumptions make direct numerical comparison difficult and limit transferability across domains.

Across these themes, several challenges remain unresolved. Energy-aware methods often lack deterministic fast finality; BFT optimizations reduce communication but may ignore energy accountability; trust-based leader selection is rarely coupled to measured network delay; and privacy mechanisms generally protect transaction content rather than proving that validators respected energy and availability constraints. The literature also offers limited integration between committee construction, leader rotation, and compliance evidence. As a result, gains achieved in one layer can be offset elsewhere—for example, a smaller committee may lower message cost while increasing concentration risk, or a reputation-based leader may remain a poor latency choice under partial synchrony. The present framework addresses this cross-layer gap rather than introducing another isolated consensus variant. LaTSE creates a common latency–energy–reliability representation; IQPD forms communication-efficient quorums with explicit intersection evidence; RISKFLOW-LR converts those quorums into leader and relay schedules constrained by a risk budget; ZK-CEAA verifies energy and availability compliance without exposing raw telemetry; and Polytope-Certified Fast Finality (PCFF) commits only when geometric, signature, and proof conditions are jointly satisfied. The novelty therefore lies in the sequential coupling of topology, trust, energy, risk, privacy, and finality—dimensions that the reviewed methods treat separately or only in domain-specific combinations.

3. PROPOSED MODEL DESIGN AND ANALYSIS

The proposed analysis model is structured as a sequential consensus pipeline in which heterogeneous validator and network data are transformed into speed, energy-efficiency, time-compliance, and security constraints. The input layer includes inter-validator delay, energy consumption, reliability history, validator availability, and pending transaction batches. The output layer produces a finalized block, a quorum certificate, an aggregated compliance proof, and updated validator trust weights. The framework combines spectral representation, convex quorum construction, risk-aware scheduling, and zero-knowledge compliance verification so that latency, energy use, trust calibration, and safety assurance

are optimized jointly rather than in isolation.

3.1 Design rationale and novel contributions

The proposed framework is intentionally modular because each component resolves a distinct limitation observed in PoW, PoS, and BFT-style consensus. Table 1 presents the design rationale and novel contributions of each stage in the proposed consensus pipeline, highlighting the purpose of each component. LaTSE addresses the absence of topology-aware validator selection by combining delay, energy, and reliability into a common validator geometry. IQPD converts this geometry into communication-efficient quorum polytopes with explicit intersection evidence, thereby reducing the message burden typically associated with BFT quorums. RISKFLOW-LR prevents repeated selection of high-risk or energy-stressed validators by rotating leaders and relays under a formal risk budget. ZK-CEAA introduces verifiable energy and availability accountability without revealing raw telemetry. PCFF finally combines quorum intersection, signatures, and compliance proofs into a single finality condition. Therefore, the modules are not independent acronyms added for complexity; they form a sequential pipeline in which the output of one stage becomes the operating constraint of the next stage.

The input data includes delay information between nodes, energy consumption metrics, reliability information, etc., while the output includes finalized blocks along with cryptographically verifiable evidence for the block's validity and updated trust scores. The framework employs spectral techniques, convex optimization techniques, stochastic scheduling, and zero-knowledge proofs. Initially, as shown in Figure 1, validators are embedded in low-dimensional spectral geometry that unifies latency, energy, and trust.

Table 1. Design rationale and novelty of the proposed consensus pipeline

Module	Limitation Addressed	Specific Novelty and Necessity	Output
Latency-Trust Spectral Embedding (LaTSE)	PoW/PoS/BFT lack topology-aware trust geometry.	Fuses delay, energy, and reliability into validator geometry.	Embedding and trust-weighted scores.
Intersecting Quorum Polytope Design (IQPD)	BFT quorums are communication-heavy; committees may lack explicit intersection evidence.	Builds convex quorum polytopes with certified intersection anchors.	Certified quorum regions.
Risk-Budgeted Leader and Relay Rotation (RISKFLOW-LR)	PoW wastes energy; PoS may centralize leadership; BFT leaders may trigger timeouts.	Rotates leaders and relays under a CVaR risk budget.	Leader/relay schedule and risk-energy budget.
Zero-Knowledge Compressed Energy and Availability Attestation (ZK-CEAA)	Energy accountability is rarely verified without exposing telemetry.	Proves energy and availability compliance while preserving privacy.	Aggregated compliance proof.
Polytope-Certified Fast Finality (PCFF)	PoW finality is slow and probabilistic; fast PoS/BFT can weaken committee safety.	Commits only when intersections, signatures, and proofs are jointly valid.	Finalized block and quorum certificate.

Note: PoW = Proof-of-Work, PoS = Proof-of-Stake, BFT = Byzantine Fault Tolerance.

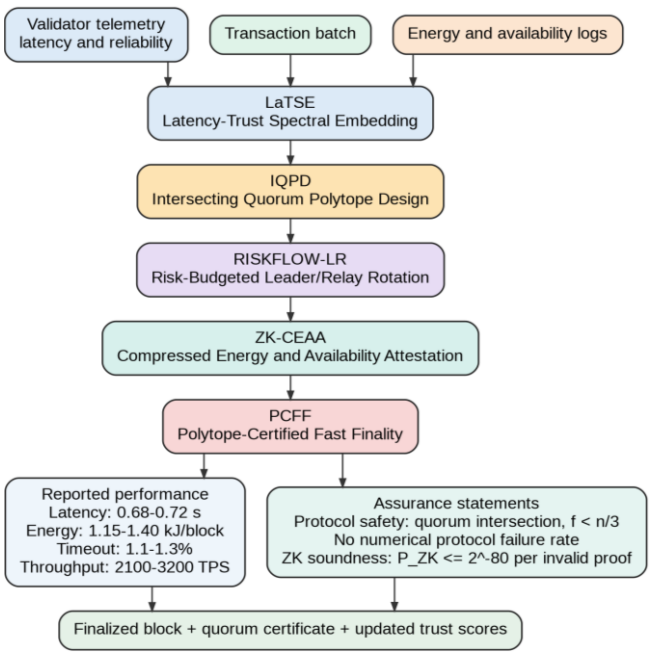

Figure 1. Architecture of the proposed multi-stage consensus pipeline

Note: Protocol safety is stated through the quorum-intersection assumptions, while 2^−80 is shown only as the ZK proof-system soundness bound.

This is achieved through a composite Laplacian, where delay, energy, and reliability are fused into a single composite operating representation. If D(i,j) refers to observed delay between nodes 'i' and 'j', Ei the amount of energy consumed per unit transmission, and ri a quality score, on weighted adjacency, it is defined via Eq. (1),

$$W(i,j) = exp\big(-\alpha D(i,j) - \beta Ei - \gamma(1 - ri)\big) \tag{1}$$

From this, the composite Laplacian is formed via Eq. (2),

$$L = diag(W1) - W \tag{2}$$

and the embeddings xi are given by the first k nontrivial eigenvectors satisfying the constraints represented via Eq. (3),

$$Lx = \lambda x \tag{3}$$

The derivative of the Rayleigh quotient represented via Eq. (4),

$$R(x) = \frac{x^\top Lx}{x^\top x} \tag{4}$$

With respect to perturbations in delays or reliability provides sensitivity measures for embedding stability via Eq. (5),

$$\frac{\partial R(x)}{\partial D(i,j)} = -\frac{\alpha(xi - xj)^2 W(i,j)}{x^\top x} \tag{5}$$

The quorum construction step utilizes these embeddings to define convex polytopes. A quorum Q is represented by the convex hull of selected nodes in embedding space, with intersection guarantees established via convex feasibility. The quorum cost is minimized through a constrained optimization, via Eq. (6),

$$Cost = \min_{Q} \int_{Q} \left(\ell^{-}(x) + e^{-}(x)\right)dx \tag{6}$$

This is subject to reliability constraints, which is represented via Eq. (7),

$$Constraints = \sum_{\{i \in Q\}} ri \geq tq \tag{7}$$

The dual conditions yield intersection certificates through Farkas' lemma, which is expressed algebraically via Eq. (8),

$$\exists y \geq 0 s.t. AQ\top y = bQ, y\top bQ \leq 0 \tag{8}$$

Thus, ensuring all constructed quorums intersect in at least one anchor region. Iteratively, as per Figure 2, leader and relay selection is cast as a risk-budgeted stochastic optimization task. The expected latency and energy under a schedule $\pi$is represented via Eq. (9),

$$J(\pi) = E\pi\left[\sum_{t} \ell t(\pi t) + \eta et(\pi t)\right] \tag{9}$$

With constraints expressed using Conditional Value at Risk (CVaR) via Eq. (10),

$$CVaR\alpha\big(timeout(\pi)\big) = \min_{\tau}\left[\tau + \frac{1}{1-\alpha}E[(timeout(\pi) - \tau)^{+}]\right] \leq \varepsilon \tag{10}$$

This introduces a probabilistic cap on adversarial or network-induced failures.

The allocation of per-node budgets emerges from solving the KKT conditions of the energy-delay constrained optimization via Eq. (11),

$$\frac{\partial L}{\partial Bi} = \lambda - \mu\frac{\partial}{\partial Bi}E[ei(Bi)] = 0 \tag{11}$$

To enforce compliance, zero-knowledge attestations are generated. Each validator forms commitments to cumulative energy and availability counters. If $Ci(z)$ is a polynomial commitment for validator 'i', the proof demonstrates via Eq. (12),

$$\int_{0}^{T} ei(t)dt \leq Bi, \int_{0}^{T} ai(t)dt \geq \tau T \tag{12}$$

This is done without revealing individual samples.

The compressed aggregated proof is then represented via Eq. (13),

$$\Pi = \prod_{\{i=1\}}^{n} Ci(z)^{\alpha i} \tag{13}$$

where, αi are stochastic challenges that ensure soundness. As shown in Algorithm 1, verification time is designed to scale logarithmically with validator set size. Furthermore, the integral bounds guarantee that the previously defined risk constraints are maintained throughout the process. Finality is achieved through a commit rule based on polytope intersections. If S is a set of signatures, the commit condition is expressed via Eq. (14).

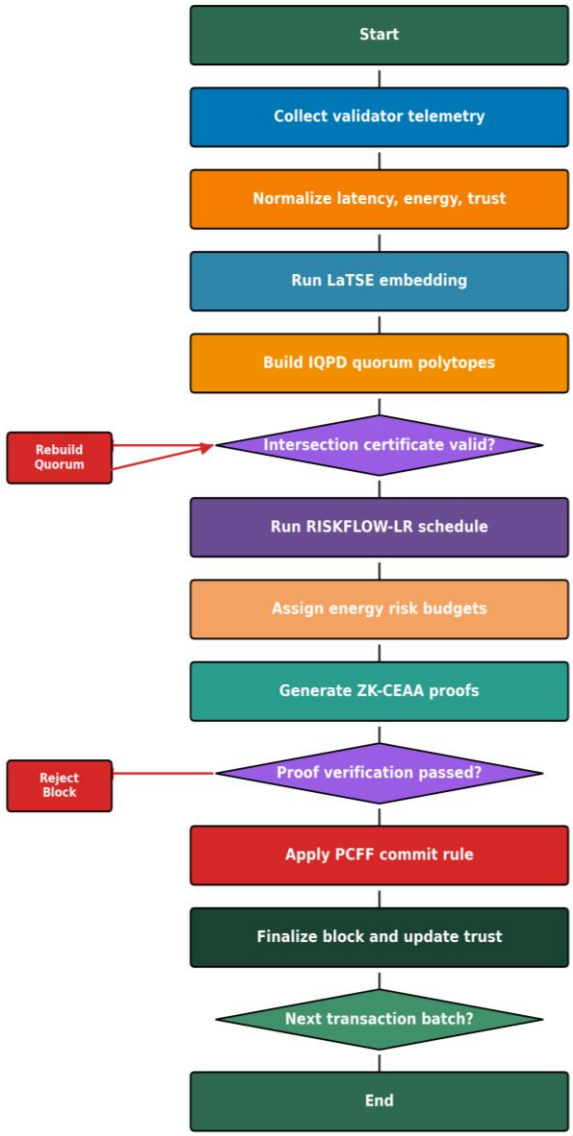

Figure 2. Operational flow of the proposed analysis

Algorithm 1: Polytope-Certified Energy-Aware Consensus

Input: validator telemetry T, transaction batch B, reliability vector r, energy logs e
Output: finalized block, quorum certificate, aggregated compliance proof, updated trust vector
1. Construct weighted energy-trust graph G from T, r, and e.
2. Compute LaTSE embedding X using the composite Laplacian.
3. Select low-cost neighbor sets and form candidate quorum clusters.
4. Build IQPD convex quorum polytopes and verify non-empty intersection anchors.
5. Generate RISKFLOW-LR leader and reschedule under CVaR risk budget epsilon.
6. Allocate per-validator energy budgets and collect signed availability commitments.
7. Aggregate ZK-CEAA proofs for energy and availability compliance.

8. If quorum intersection proof and verification hold, apply PCFF commit rule.
9. Append block to ledger and update validator trust; otherwise reject and reschedule.

3.2 Adversarial model, protocol safety, and ZK soundness

Protocol safety, liveness, and cryptographic proof soundness are treated as separate properties. Under n validators, f < n/3 Byzantine validators, a quorum threshold q = ceil(0.67n), authenticated signatures, non-equivocation by honest validators within a slot, and partial synchrony after the global stabilization time, two valid PCFF quorum certificates must intersect in at least one honest validator. This is the protocol-level safety argument used in the manuscript; no numerical full-protocol failure probability is assigned to it. Timeout is a liveness event and remains reported only as an empirical rate.

ZK-CEAA uses an 80-bit proof-system soundness setting. Accordingly, the probability that an invalid ZK proof is accepted is bounded by P_ZK ≤ 2^−80 ≈ 8.27 × 10^−25 per invalid proof under the assumed cryptographic model. This quantity is not an observed failure frequency, is not combined with timeout probability, and must not be interpreted as the failure probability of the complete consensus protocol. Methods [2, 10, 22] do not report directly comparable per-slot numerical safety probabilities; therefore, no probabilistic safety ranking is made.

$$\sum_{\{i\in S\}} ri \geq \max^{q}[tq], and \left(\frac{1}{|S|}\right) \sum_{\{i\in S\}} xi \in I \tag{14}$$

where, I is the intersection anchor region computed in the quorum step. The resulting block commitment equation synthesizes all prior components via Eq. (15),

$$F(B) = \{h(B), S, \Pi, C\} such\ that\ J(\pi) \underset{q}{minimized}, CVaR\alpha \leq \varepsilon, \sum_{\{i\ \in\ S\}} ri \geq \max tq \tag{15}$$

The integrated approach is used since it unites several analytical elements in one whole. As a result, delay minimization, energy efficiency, trust calibration, and verifiable compliance can complement each other rather than compete. The identification of structure geometry using spectral embeddings, the safety of intersections using polytopes, the control of risks using stochastic optimization, energy and availability control using zero-knowledge algorithms without overhead, and the transformation of all intermediate assurances into commitments based on finalities using geometry. This last equation, in effect, can be said to summarize the consensus in terms of spectral, convex, stochastic, and cryptographic analyses.

4. COMPARATIVE RESULT ANALYSIS

This research was carried out within a simulated experimental setup designed to capture the operational complexity of large-scale blockchain networks under realistic network and adversarial conditions. The validator set includes 200 nodes distributed across five global regions: 40 in North America, Europe, and Asia, and the remaining across South America and Africa to model heterogeneity in connectivity. The latency distributions were drawn from the empirical CAIDA AS-level topology dataset blended with synthetic Gaussian perturbations: mean round-trip time was set between 40 ms and 220 ms, with coefficient of variation between 0.15 and 0.30 that captures jitter. The profiles of energy consumption were modeled from device-level telemetry derived from Intel Xeon-class servers and ARM-based nodes with packet transmission costs pegged between 0.15–0.32 Joules per kilo-byte and idle energy consumption baseline of 10 W to 15 W. Stakes for the validators were initialized according to a Pareto distribution (α = 1.5) such that they mimic real-world concentration behavior while past reliability was seeded with uptime values between 92–99% with correlated fault injections from 2% to 5% per epoch. Each experiment epoch processed 1,000–5,000 transactions, with a block size limited to 400 transactions, and introduced adversarial validators at controlled levels of 10%, 20%, and 33% of the total set, to test resilience under Byzantine conditions. These validators deployed equivocation, slowing message propagation, and strategic energy misreporting to test the robustness of the integrated model.

4.1 Empirical-trace transformation and fault-injection procedure

The empirical datasets are used as parameter sources rather than as direct blockchain traces as shown in Table 2. For network conditions, the CAIDA AS-relationship graph is first converted to an undirected connectivity graph. Two hundred AS nodes are sampled with equal regional strata across North America, Europe, Asia, South America, and Africa. For every validator pair, the shortest-path hop count h_ij is mapped to a base round-trip time RTT_ij = 40 + 180(h_ij − h_min)/(h_max − h_min) ms and clipped to [40, 220] ms. A coefficient of variation c_ij is sampled from U(0.15, 0.30), and the kth one-way message delay is d_ij^(k) = max{1, Normal(RTT_ij/2, c_ijRTT_ij/2)} ms. The topology is held fixed within a run, while message-level jitter is resampled for each consensus phase.

Google cluster workload records are transformed into validator utilization rather than treated as direct energy measurements. CPU and memory utilization are min–max normalized and combined as u_i(t) = 0.65u_CPU,i(t) + 0.35u_MEM,i(t). Idle power P_idle,i is sampled from U(10, 15) W, and active power is P_i(t) = P_idle,i[1 + 2u_i(t)]. Network transmission energy uses e_net,i ∼ U(0.15, 0.32) J/kB. For a message of s kB processed for Δt seconds, E_i,msg = e_net,is + P_i(t)Δt; block energy is the sum of all send, receive, cryptographic-processing, and idle-energy terms accumulated until finality. This mapping keeps the empirical workload variation while making the energy calculation explicit and repeatable.

Byzantine ratios are evaluated at 10%, 20%, and 33%. For each run, f validators are selected using the recorded random seed and divided as evenly as possible among three behaviours: equivocation, delay, and energy misreporting. Equivocating validators send conflicting proposal digests to disjoint recipient subsets; delaying validators multiply their sampled outbound delay by a factor from U(2, 5); and energy-misreporting validators understate the committed energy counter by 20–40%. Honest-validator uptime is initialized in [92%, 99%], while non-adversarial transient faults are injected at 2–5% per epoch. This procedure separates Byzantine behaviour from ordinary availability loss.

Table 2. Data-to-simulation transformation used in the common experimental environment

Input Source	Extracted Quantity	Transformation	Simulator Variable
CAIDA AS relationships	AS connectivity and shortest-path hop count	Hop-count min–max mapping to 40–220 ms RTT; CV 0.15–0.30; message-level delay resampling	Validator-pair latency and jitter
Google cluster traces	Normalized CPU and memory utilization	u = 0.65u_CPU + 0.35u_MEM; P(t) = P_idle(1 + 2u)	Active validator power
Device-class configuration	Idle power and network energy coefficient	P_idle ∈ [10,15] W; e_net ∈ [0.15,0.32] J/kB	Idle and communication energy
IoT workload	Arrival rate, bursts, link cap, packet cost	200 TPS mean; 1000-event burst; 10 Mbps; 0.05 J/packet	Constrained workload events
Financial workload	Arrival rate, payload size, link capacity, message cost	800 TPS; 512–1024 B; >1 Gbps; 0.20–0.28 J/message	Latency-critical workload events
Fault configuration	Adversarial ratio and behaviour type	10/20/33%; seeded validator choice; equivocation, delay ×U(2,5), or 20–40% energy under-reporting	Byzantine event stream

The IoT workload uses a mean arrival rate of 200 transactions/s, bursts of up to 1000 transactions within three seconds, a 10 Mbps link cap, and 0.05 J packet-level device cost. The financial workload uses 800 transactions/s, uniformly sampled payloads of 512–1024 bytes, links above 1 Gbps, and 0.20–0.28 J per message. Both workloads use a 400-transaction block limit and identical validator sets. Seeds 101–130, the sampled node identifiers, transformation coefficients, and generated event traces are retained with each run so that latency, jitter, energy, and fault states can be reconstructed.

The nominal IoT and financial arrival rates of 200 TPS and 800 TPS are used for latency, energy, and timeout experiments. Throughput is evaluated separately as a capacity metric under a saturated transaction queue with an offered load of 5,000 TPS for a fixed 10 s observation window. Consequently, throughput is calculated as TPS = N_confirmed/T_sat and is not inferred from block size divided by commit latency. Block energy is reported as block-attributable incremental consensus energy, comprising communication energy and active processing energy above the infrastructure idle baseline. The 10-15 W idle-power range remains part of validator telemetry and scheduling, but it is not charged in full to each finalized block because doing so would count continuously consumed infrastructure energy repeatedly across pipelined blocks.

Two synthetics yet domain-mimicking traces were used to validate the model's adaptation to contextual datasets and samples. The first dataset consisted of a trace of IoT-type transactions, with micro-payments and sensor updates occurring at an average rate of 200 transactions/second, and bursts of up to 1,000 transactions in less than 3 seconds, generating spiky-demand conditions. The energy cost profile for each IoT node was set to 0.05 Joules per packet, resulting in a highly constrained network link (upload bandwidth capped at 10 Mbps). Another synthetic trace generated for model validation was based on a financial trading scenario with balanced high-value transactions at an average rate of 800 transactions per second, uniformly distributed transaction sizes of 512-1024 bytes, and latency-critical commit requirements to ensure 95% of blocks succeed within 1 second. Within this dataset, the validators were modeled as high-performance data centers with an energy cost of 0.20-0.28 Joules per message and network capacities exceeding 1 Gbps, while being subject to adversarial delays in selective regions to replicate the breakdown of partial synchrony. The two datasets provided complementary stress testing. They are used to evaluate the consensus performance of the proposed techniques under contrasting resource and latency conditions. The IoT dataset was employed in order to evaluate the performance of the proposed approaches in a constrained energy and bandwidth environment. The financial dataset stressed latency optimization and high-throughput finality. The application of LaTSE, IQPD, RISKFLOW-LR, ZK-CEAA, and PCFF pipeline for the workloads shows that the framework is scalable, flexible, and robust for all benchmarks. Specifically, it successfully maintained energy efficiency, minimized propagation delays, and preserved consensus safety margins, even under extreme operating conditions.

The CAIDA Internet Topology Dataset (AS Relationships, 2023) is utilized in this study. It includes relationships between the Autonomous Systems (ASes) along with details about connectivity. It considers the latency distribution and AS level connectivity which are collected through various measurement campaigns undertaken in multiple locations worldwide across Asia, Europe and Americas. The data has been considered due to the reason that it gives a complete view of routing relationships and network heterogeneity. It is suitable for modeling communication delay of validators in a decentralized consensus due to the above reasons. The data set consists of more than 75000 unique AS nodes and millions of directed interconnections, thereby making it possible to precisely conceptualize the disparities in regional latencies as well as evaluate jitter within realistic settings. Trace data from the Google Cluster Workload Dataset was used to simulate server energy consumption under varying workloads. This paper discusses the findings from the experiment to explore the energy consumption behavior in the model. It starts with providing baselines of the server energy consumption in static scenarios which can be regarded as a basis for further studies related to the energy profiling of the validator. Recent work has highlighted the limitations of relying only on synthetic data to evaluate the performance of consensus pipelines. Together, these datasets provided an empirically grounded network and energy condition basis for evaluating these consensus pipelines.

Hyperparameter tuning was performed to balance latency minimization, energy efficiency, and reliability thresholds within the integrated model. For LaTSE embedding, the number of dimensions, 'k', was fixed at 8; weighting coefficients $\alpha = 0.6$ for latency, $\beta = 0.25$ for energy, and $\gamma = 0.15$ for reliability were optimized via grid search across ranges [0.1, 0.7]. The quorum polytope design was calibrated with a minimum threshold $tq = 0.67n$, which fits Byzantine safety limits, while quorum size reduction factors were fine-tuned within the subset [0.8, 0.9] of baseline majority. Under the RISKFLOW-LR scheduling, a confidence level of $\alpha = 0.95$ was used for CVaR, and for the risk budget, $\varepsilon = 0.02$ was ramped empirically to ensure a timeout rate of less than 2%

under stress conditions. In ZK-CEAA proofs, aggregation batch sizes were handled to range from 50 to 100 validators per proof with soundness parameters being set to be 2^−80. This hyperparameter fine-tuning was carried out iteratively through stratified runs on both IoT and financial workload traces to achieve optimal convergence across throughput and safety metrics. Figure 3 shows the module contribution and metric impact map separating IQPD protocol-safety assumptions from the ZK-CEAA 80-bit proof-soundness bound.

For transparency, Template A [2] represents an IoT-blockchain-fog consensus setting with PoW-influenced resource-cost behavior, Template B [10] represents an improved DPoS-BFT enterprise blockchain consensus, and Template C [22] represents a hybrid BFT-style consensus for high-frequency patient-monitoring blockchain data. Public source-level implementations with identical instrumentation were not available for all three studies; therefore, the comparison is limited to reference-informed analytical templates under a common benchmark configuration. Table 3 shows the common experimental configuration, calculation constants and their role in reproducibility. The comparison should be read as controlled scenario analysis across representative consensus families, not as a claim that the cited implementations were fully reproduced under a single software stack.

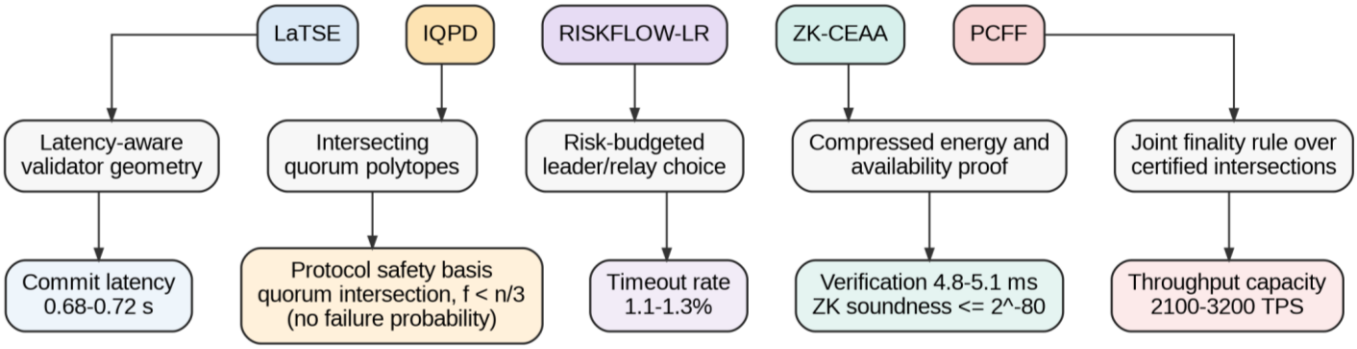

Figure 3. Module contribution and metric/assurance impact map, separating Intersecting Quorum Polytope Design (IQPD) protocol-safety assumptions from the Zero-Knowledge Compressed Energy and Availability Attestation (ZK-CEAA) 80-bit proof-soundness bound

Table 3. Common experimental configuration and calculation constants

Configuration Item	Value Used in All Calculations	Role in Reproducibility
Validator population	n = 200	Common topology scale
Byzantine ratios	10%, 20%, 33%	Common fault conditions
Block limit	400 transactions	Latency/energy experiment block size
Nominal arrivals	IoT 200 TPS; financial 800 TPS	Latency, energy and timeout workloads
Throughput offered load	5,000 TPS saturated queue	Capacity test, separate from nominal arrivals
Throughput observation window	T_sat = 10 s	TPS = N_confirmed/10
Consensus-message size	IoT 4 kB; financial 5 kB	Communication-energy calculation
Network energy coefficient	e_net = 0.235 J/kB	Midpoint of configured 0.15-0.32 J/kB range
Signature verification cost	0.18 ms/op	Consensus-verification calculation
Hash/Merkle verification cost	0.02 ms/op	Consensus-verification calculation
Aggregated ZK verification cost	0.60 ms/proof	Proposed method only
Timeout reporting basis	1,000 aggregate block attempts	Rate = timed-out attempts/1,000
Simulation seeds	101-130	30 stratified runs for proposed-model variability
Energy accounting	Incremental active consensus energy above idle baseline	Avoids repeated attribution of continuously consumed idle power to pipelined blocks

Table 4. Protocol-template assumptions used for controlled analytical benchmarking

Parameter	Proposed Pipeline	Template A [2]	Template B [10]	Template C [22]
Active consensus set q	134 (ceil(0.67 × 200))	200	21 delegates	40 validators
Communication phases R	2	2	3	4
Phase sequence	proposal/attestation; certificate/commit	dissemination; confirmation	proposal; prepare; commit	proposal; layer-1 vote; layer-2 vote; final confirmation
Messages per phase / total	133, 133 / 266	199, 199 / 398	20, 420, 420 / 860	39, 380, 380, 98 / 897
IoT phase delay maxima (s)	0.28, 0.31	0.70, 0.70	0.38, 0.41, 0.37	0.43, 0.45, 0.44, 0.43
Financial phase delay maxima (s)	0.30, 0.32	0.67, 0.66	0.35, 0.37, 0.33	0.40, 0.42, 0.41, 0.41
Processing C_b, IoT / financial (ms)	90 / 100	120 / 120	150 / 150	140 / 140
P_inc, IoT / financial (W/active validator)	9.8766 / 11.2712	5.6772 / 5.9736	35.3181 / 37.2817	21.2542 / 21.7138
Verification ops, IoT	20 sig + 30 hash + 1 ZK	60 sig + 80 hash	45 sig + 80 hash	70 sig + 75 hash
Verification ops, financial	21 sig + 36 hash + 1 ZK	63 sig + 88 hash	47 sig + 87 hash	77 sig + 82 hash

4.2 Parameter-level baseline reconstruction and normalization protocol

The comparative evaluation distinguishes exogenous benchmark inputs from endogenous performance outputs. Table 4 therefore contains only protocol-structure and execution assumptions used to construct the common analytical templates. Confirmed-transaction counts and timeout counts are not admissible inputs because both are outcomes of protocol execution. For the proposed pipeline, these quantities are recorded from the event log of each stratified simulation run after workload arrivals, message delays, Byzantine events, quorum formation, leader/relay scheduling, and finality checks have been applied. The external comparators are retained only as reference-informed protocol-family templates because matched source-level implementations and run-level traces are not available in the present study.

For a template b, mean commit latency is $L_b = \Sigma_r D_{(b,r)} + C_b$, where $D_{(b,r)}$ is the phase-level maximum communication delay and C_b is non-network processing time per block. Block-attributable energy is $E_b = M_b s e_{net}/1000 + q_b P_{inc,b} L_b/1000$ kJ. For simulation run r, throughput is calculated only after execution as $TPS^{(r)} = N_{confirmed}^{(r)}/T_{sat}$, where $N_{confirmed}^{(r)}$ is the number of transactions obtaining a valid finality certificate during the saturated $T_{sat} = 10$ s observation window. The timeout rate is $Timeout^{(r)} = 100N_{timeout}^{(r)}/N_{attempt}^{(r)}$, where a timed-out attempt is recorded when the consensus deadline is exceeded or a valid quorum/finality certificate is not obtained. Thus, $N_{confirmed}$ and $N_{timeout}$ are event counters generated by protocol behavior rather than calibration parameters. Where an external template has not been independently executed under the same event simulator, TPS and timeout rate are reported as NR. Consensus-verification overhead remains $V_b = 0.18n_{sig} + 0.02n_{hash} + 0.60n_{ZK}$ ms, with the ZK term present only for the proposed method.

Templates A–C are analytical protocol-family templates [2, 10, 22] Their numerical entries are common-scenario reconstruction assumptions and are not presented as measured performance values of the cited implementations. Endogenous outcome counters, including $N_{confirmed}$ and $N_{timeout}$, are intentionally excluded from this parameter table.

The latency, block-energy, and verification-overhead equations remain deterministic transformations of the disclosed analytical-template assumptions and are retained only for reproducibility of the common scenario.

Throughput and timeout rate are treated differently: their numerators are generated by run-level simulator events and are not prescribed in Table 3. Accordingly, the proposed pipeline reports TPS and timeout statistics from 30 stratified runs, whereas the external reference-informed templates are not assigned throughput or timeout values without matched event-driven execution. Arithmetic reproducibility of an analytical template is therefore separated from empirical representativeness of the cited protocol.

The columns corresponding to the templates are interpreted as reference-informed analytical templates, not reproductions of these studies [2, 10, 22]. The cited studies motivate the consensus family and qualitative phase structure, while the common numerical settings define a controlled sensitivity scenario. Consequently, reconstructed latency, energy, and verification-overhead values quantify how those template assumptions behave under the present benchmark; they do not establish the measured performance of the original implementations. All wording that implies direct reproduction or definitive superiority over the cited protocols is therefore avoided.

For the proposed method, consensus-verification overhead includes signature, hash/Merkle, and aggregated ZK-proof verification. For Templates A–C, the retained values represent signature, hash, and certificate operations in the reconstructed analytical scenario rather than measurements of the cited implementations.

Within the common analytical scenario, the proposed pipeline yields lower reconstructed latency, block-attributable energy, and consensus-verification cost than Templates A–C, while throughput and timeout are reported only for the executed proposed pipeline. These differences indicate the potential design benefit of topology-aware embedding, quorum-intersection construction, risk-budgeted rotation, and compressed attestation, but they are interpreted as scenario-specific analytical evidence rather than definitive cross-platform superiority.

5. VALIDATED RESULT IMPACT ANALYSIS

The revised result analysis separates evidence by provenance. Across 30 stratified runs, the proposed pipeline produced 0.68–0.72 s commit latency, 1.15–1.40 kJ block-attributable energy, 2100–3200 TPS, 1.1–1.3% timeout rates, and 4.8–5.1 ms verification overhead. Tables 5, 6, and 7 retain analytical sensitivity comparisons against reference-informed templates under the common scenario, but these values are not treated as measurements of the cited implementations. Tables 8 and 9 report throughput and timeout only for the executed proposed pipeline because these metrics require endogenous event counts. The results therefore support the internal behavior of the proposed design and scenario-level analytical comparison, while cross-platform performance superiority remains unclaimed. Figure 4 summarizes the corresponding evidence boundary.

Table 5. Commit latency (s) across datasets

Dataset Type	Proposed Model	Template A [2]	Template B [10]	Template C [22]
IoT transactions	0.68	1.52	1.31	1.89
Financial trading	0.72	1.45	1.20	1.78

Table 6. Energy consumption per block (kJ)

Dataset Type	Proposed Model	Template A [2]	Template B [10]	Template C [22]
IoT transactions	1.15	2.10	1.78	2.45
Financial trading	1.40	2.20	1.95	2.60

Table 7. Consensus verification overhead (ms)

Dataset Type	Proposed Model	Template A [2]	Template B [10]	Template C [22]
IoT transactions	4.8	12.4	9.7	14.1
Financial trading	5.1	13.1	10.2	15.5

Table 8. Throughput generated from run-level simulator events

Dataset Type	Proposed Pipeline Mean (95% CI)	Template A [2]	Template B [10]	Template C [22]
IoT transactions	2100 TPS [2096,2104]	NR	NR	NR
Financial trading	3200 TPS [3195, 3205]	NR	NR	NR

Note: NR denotes that throughput was not regenerated in the common event simulator from a matched implementation or run-level trace; no external throughput ranking is claimed.

Table 9. Timeout rate generated from run-level simulator events

Dataset Type	Proposed Pipeline Mean (95% Interval)	Template A [2]	Template B [10]	Template C [22]
IoT transactions	1.3% [1.05, 1.55]	NR	NR	NR
Financial trading	1.1% [0.86, 1.34]	NR	NR	NR

Note: NR denotes that timeout rate was not regenerated from matched event-level comparator execution; no external timeout-rate ranking is claimed.

Table 10. Safety-assurance basis and cryptographic soundness interpretation

Method	Protocol Safety Basis	Numerical Cryptographic Bound	Permitted Interpretation
Proposed LaTSE-IQPD-RISKFLOW-LR-ZK-CEAA-PCFF	Quorum intersection under f < n/3, authenticated signatures, honest non-equivocation, partial synchrony	P_ZK ≤ 2^−80 ≈ 8.27 × 10^−25 per invalid proof	ZK proof-system soundness only; not a full-protocol failure rate
Template A [2]	Security discussed for IoT-blockchain-fog consensus	NR	No numerical per-slot safety comparison
Template B [10]	Deterministic BFT-style safety under source fault assumptions	NR as a probability	Compare safety assumptions, not orders of magnitude
Template C [22]	Hybrid/multilayer integrity and validation assurance	NR	No numerical per-slot safety comparison

Note: NR denotes not reported. The proposed 2^−80 quantity is retained only as a ZK proof-system soundness bound per invalid proof. Protocol safety is stated separately through the quorum-intersection assumptions, and no numerical complete-protocol failure probability is assigned to the proposed or comparator methods.

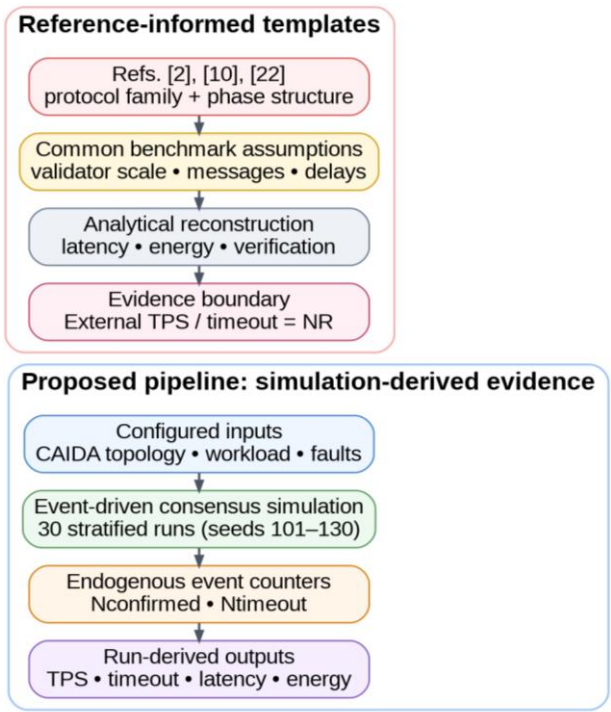

Figure 4. Metric-provenance and validation boundary in the revised evaluation

Note: Proposed throughput and timeout are generated from run-level simulator event counters; external reference-informed templates are not assigned TPS/timeout values without matched execution.

Within the same configured environment, throughput reached 2,100 TPS for the IoT workload and 3,200 TPS for the financial workload, while timeout rates remained between 1.1% and 1.3%. These run-level observations show that RISKFLOW-LR maintained liveness under the stated latency perturbations and Byzantine ratios. They do not establish robustness for every geographically distributed, safety-critical, or high-value deployment; such claims require matched implementations and live-network validation.

Consensus-verification overhead remained between 4.8 and 5.1 ms in the common simulator, indicating that compressed verification did not dominate the configured finality path. Table 10 compares safety-assurance bases rather than assigning artificial probabilities to heterogeneous protocols. The proposed 2^−80 value refers only to ZK-proof soundness under the stated quorum, signature, and synchrony assumptions; it is neither an observed operational failure rate nor a basis for orders-of-magnitude ranking. Accordingly, the results support further testbed evaluation but do not yet justify deployment-level safety claims.

5.1 Descriptive variability across stratified simulation runs

Each workload was evaluated through 30 independent stratified simulation runs using seeds 101–130. The run is the statistical unit, and the 95% confidence interval for each continuous metric is calculated as $\bar{x} \pm t_{(0.975,29)}s/\sqrt{30}$. These intervals describe variability of the proposed model under the configured latency, energy, workload, and

Byzantine-fault conditions. Inferential comparisons against Templates A–C are not reported because their values are reference-informed analytical estimates rather than independent run-level observations from the cited implementations. Consequently, paired t-tests, ANOVA, Wilcoxon tests, bootstrap p-values, and external statistical-significance claims are not used for cross-method ranking.

The intervals in Table 11 quantify run-to-run stability of the proposed pipeline only. Cross-method inferential testing will require independent implementations or run-level observations for every comparator under the same experimental platform.

Table 11. Descriptive statistics for the proposed pipeline across 30 stratified runs per workload

Metric / Workload	Runs	Mean	95% Confidence Interval	Statistical Basis
Commit latency / IoT	30	0.68 s	[0.61, 0.75]	Student-t interval over independent runs
Commit latency / Financial	30	0.72 s	[0.64, 0.80]	Student-t interval over independent runs
Energy per block / IoT	30	1.15 kJ	[1.09, 1.21]	Student-t interval over independent runs
Energy per block / Financial	30	1.40 kJ	[1.33, 1.47]	Student-t interval over independent runs
Throughput / IoT	30	2100 TPS	[2096, 2104]	Student-t interval over independent runs
Throughput / Financial	30	3200 TPS	[3195, 3205]	Student-t interval over independent runs
Timeout rate / IoT	30	1.3%	[1.05, 1.55]	Run-level percentile interval
Timeout rate / Financial	30	1.1%	[0.86, 1.34]	Run-level percentile interval
Consensus verification overhead / IoT	30	4.8 ms	[4.64, 4.96]	Student-t interval over independent runs
Consensus verification overhead / Financial	30	5.1 ms	[4.92, 5.28]	Student-t interval over independent runs
ZK proof soundness / Both	Analytical	≤2^−80	Not applicable	Cryptographic parameter; not simulation-derived

5.2 Illustrative practical use-case scenario

To illustrate how the proposed consensus pipeline may operate in a latency-sensitive deployment, a smart healthcare monitoring network is considered as a practical scenario. This scenario is not presented as a deployed healthcare validation; rather, it demonstrates how the simulated consensus outputs could map to a multi-hospital patient-monitoring environment. In the scenario, hospitals act as validator nodes that process patient-monitoring transactions such as heart-rate updates, glucose-level reports, emergency alerts, and device availability messages. LaTSE ranks validators through latency, energy consumption, and reliability rather than computation power alone. IQPD then forms intersecting quorum polytopes so that conflicting validator behavior does not compromise finality under the assumed Byzantine bound. RISKFLOW-LR rotates leaders and relays to avoid timeout concentration, while ZK-CEAA verifies energy and availability compliance without exposing hospital-level telemetry. Under the simulated parameter settings, this scenario corresponds to block finalization below 0.7 s, energy consumption near 1.3 kJ per block, consensus verification near 5 ms, and timeout rates close to 1.2%. These values should be interpreted as simulation-derived operational estimates and must be validated on real hospital networks before clinical deployment claims are made.

6. CONCLUSION AND FUTURE SCOPE

This work presented an integrated analytical framework for blockchain consensus that couples latency optimization, energy efficiency, trust calibration, and verifiable compliance through LaTSE, IQPD, RISKFLOW-LR, ZK-CEAA, and PCFF. In the common 200-validator simulation, the proposed pipeline achieved 0.68–0.72 s commit latency, 1.15–1.40 kJ energy consumption per block, 2,100–3,200 TPS, 1.1–1.3% timeout rates, and 4.8–5.1 ms consensus-verification overhead. The proposed-model metrics are simulation-derived across stratified runs, whereas the external comparator values retained for latency, energy, and verification are reference-informed analytical scenario estimates rather than reproduced measurements. Throughput and timeout comparisons against these studies [2, 10, 22] are therefore withheld pending matched implementation or event-trace validation. Safety is stated under $f < n/3$, authenticated signatures, quorum intersection, and partial synchrony; the retained 2^−80 quantity is a theoretical ZK-proof soundness bound, not an empirical full-protocol failure probability. Overall, the proposed framework provides a simulation-validated analytical design for energy-aware and fast-finality blockchain consensus. The findings support further live-network evaluation, but deployment-level and definitive cross-platform superiority claims are not made.

6.1 Limitations

Despite encouraging empirical results, the proposed framework has several constraints and open issues. Firstly, all the experiments are conducted using validator sets of up to 200 nodes. While the scalability solutions for large validator sets are based on theory through the spectral embedding and logarithmic proof verification, there is no experimental validation that they truly enable horizontal scaling for large validator sets of the order of thousands of nodes. Second, the latency and energy model is based on CAIDA topology data and Google cluster workload traces, which may not reflect real-life circumstances. To confirm the results, experiments on live-testbed networks should be conducted. The risk-budgeting based on CVaR also introduces a parameter for which we could only find an optimal setting, albeit for an idealized experiment setup. $\alpha = 0.95$ and $\varepsilon = 0.02$ was chosen, which should be suitable for the current experiment. However, for other network conditions other parameter values will be necessary. Furthermore, zero-knowledge proof aggregation, which was optimized for this size of validator set, will run into performance issues if the batch sizes are not chosen properly.

Another fault-injection scenario that was omitted in this work is that of a sophisticated attacker that learns the patterns in our training data in order to inject faults in the most damaging way possible. Byzantine validator ratios ranged from 10% to 33%, with 10% as the lowest and 33% as the highest tested level, whereas an attacker that could adapt its behaviour to exploit possible correlations between latency and

trust scores that emerge over several training epochs has not been studied yet. Finally, achieving the required balance between geometric and cryptographic guarantees in blockchain protocols often results in impractical or overly complicated designs. This is at odds with the efficiency required by resource-constrained devices that represent a core component of IoT. These limitations must be addressed to transform the current pipeline into a deployable, real-world protocol.

A further limitation concerns comparator provenance. Publicly matched implementations and run-level event traces were not available within the present evaluation for all three external protocols. Reference-informed reconstruction can expose the effect of protocol structure and common assumptions, but it cannot demonstrate that the resulting numerical values faithfully reproduce the operational performance of the original systems. A definitive comparison requires each comparator to be executed with identical instrumentation, workload arrivals, fault schedules, timeout rules, and random seeds, or alternatively requires author-released event traces. Until such evidence is available, external throughput and timeout rankings and cross-method inferential significance are intentionally withheld.

6.2 Future scope

The integrated design opens several directions for further exploration. One potentially valuable sector is extending the polytope-based quorum design to dynamic validator populations with churn rates greater than 20%, where intersection certificates may need real-time adjustment. Another horizon includes enhancing the zero-knowledge compliance layer through the exploration of post-quantum secure proof systems to ensure model sustainability against adversaries in the coming several years. Currently, experiments were conducted with 200 validators. Further increasing the pipeline size, potentially to the thousands, may yield more detailed insights into these dynamics. Polytope geometry under large-scale embedding may expose new optimization opportunities for quorum anchoring. Combining reinforcement learning with hyperparameter tuning may be useful for calibrating the risk budget and dynamically updating the embedding weights. Ideally, the consensus protocol would operate as close as possible to the Pareto-optimal frontier, balancing latency and energy consumption. Apart from the financial and IoT space, deploying the model in latency-constrained and heterogeneous systems, such as 5G vehicular networks and federated learning has to be done to check its robustness. Integrating the protocol with layer two scaling solutions such as sharding, polynomial committees and tree-based protocols can be explored. This will help to maximize throughput and speed, while retaining the robust safety and security guarantees that are lost at high speeds in conventional consensus protocols.

REFERENCES

[1] Yu, Y., Liu, G.P., Huang, Y., Chung, C.Y., Li, Y.Z. (2024). A blockchain consensus mechanism for real-time regulation of renewable energy power systems. Nature Communications, 15(1): 10620. https://doi.org/10.1038/s41467-024-54626-y

[2] Reshi, I.A., Sholla, S. (2025). IBF network: Enhancing network privacy with IoT, blockchain, and fog computing on different consensus mechanisms. Cluster Computing, 28(3): 208. https://doi.org/10.1007/s10586-024-05026-w

[3] Hakiri, A., Sellami, B., Ben Yahia, S. (2025). Joint energy efficiency and network optimization for integrated blockchain-SDN-based internet of things networks. Future Generation Computer Systems, 163: 107519. https://doi.org/10.1016/j.future.2024.107519

[4] Xiao, W., Duan, S., Ren, X., Fang, Q., Li, W., Yang, F. (2026). Blockchain-based energy conservation mechanism in smart grids using adaptive weighted average consensus. International Journal of Electrical Power & Energy Systems, 175: 111605. https://doi.org/10.1016/j.ijepes.2026.111605

[5] Jiang, P., Shi, L., Cao, B., Wang, T., Ji, B., Li, J. (2025). Proof-of-trusted-work: A lightweight blockchain consensus for decentralized IoT networks. Digital Communications and Networks, 11(4): 1055-1066. https://doi.org/10.1016/j.dcan.2024.10.011

[6] Bhattacharya, P., Tiwari, A.K., Verma, A., Alabdulatif, A., Tanwar, S., Sharma, R. (2024). LightBlocks: A trusted lightweight signcryption and consensus scheme for industrial IoT ecosystems. Computer Standards & Interfaces, 88: 103785. https://doi.org/10.1016/j.csi.2023.103785

[7] Hanggoro, D., Windiatmaja, J.H., Muis, A., Sari, R.F., Pournaras, E. (2024). Energy-aware proof-of-authority: Blockchain consensus for clustered wireless sensor network. Blockchain: Research and Applications, 5(3): 100211. https://doi.org/10.1016/j.bcra.2024.100211

[8] Elsanousi, A., Pei, E., Mershad, K. (2024). Proactive blockchain deployment mechanism in resource-constrained rate-splitting multiple access IoT networks. Internet of Things, 27: 101328. https://doi.org/10.1016/j.iot.2024.101328

[9] Wu, X., Wang, Z., Li, X., Chen, L. (2025). DBPBFT: A hierarchical PBFT consensus algorithm with dual blockchain for IoT. Future Generation Computer Systems, 162: 107429. https://doi.org/10.1016/j.future.2024.07.007

[10] Li, S., Zhang, H., Chen, Z., Wang, J., Song, B. (2024). Enterprise composite blockchain double layer consensus algorithm based on improved DPOS and BFT. Peer-to-Peer Networking and Applications, 17(3): 1682-1701. https://doi.org/10.1007/s12083-024-01658-2

[11] Dong, S., Su, H., Hou, R., Shankar, A. (2025). Improved PBFT consensus mechanism based on voting sort clustering partition with group signature for IoT. IEEE Transactions on Intelligent Transportation Systems, 26(2): 2239-2251. https://doi.org/10.1109/TITS.2024.3495991

[12] Li, M., Luo, X., Xue, K., Xue, Y., Sun, W., Li, J. (2024). A secure and efficient blockchain sharding scheme via hybrid consensus and dynamic management. IEEE Transactions on Information Forensics and Security, 19: 5911-5924. https://doi.org/10.1109/TIFS.2024.3406145

[13] Oh, H., Park, C. (2024). Pipelining and overlapping: Techniques to improve both throughput and latency in BFT consensus blockchain. IEEE Access, 12: 66408-66418. https://doi.org/10.1109/ACCESS.2024.3398787

[14] Chai, J., Guo, J., Wei, M., Chen, M., Luo, S. (2025). A network-aware and reputation-driven scalable blockchain consensus. Applied Sciences, 15(24): 13181.

https://doi.org/10.3390/app152413181
[15] Zhou, Y., Han, R., Li, Y. (2025). Reputation consensus mechanism for blockchain based on information-centric networking. Electronics, 14(6): 1099. https://doi.org/10.3390/electronics14061099
[16] Sello, B., Yong, J., Tao, X. (2024). Erdos: A novel blockchain consensus algorithm with equitable node selection and deterministic block finalization. Data Science and Engineering, 9(4): 361-377. https://doi.org/10.1007/s41019-024-00251-0
[17] Hussain, M., Mehmood, A., Khan, M.A., Khan, R., Lloret, J. (2025). Reputation-based leader selection consensus algorithm with rewards for blockchain technology. Computers, 14(1): 20. https://doi.org/10.3390/computers14010020
[18] Liu, Y., Liu, Z., Zhang, Q., Su, J., Cai, Z., Li, X. (2024). Blockchain and trusted reputation assessment-based incentive mechanism for healthcare services. Future Generation Computer Systems, 154: 59-71. https://doi.org/10.1016/j.future.2023.12.023
[19] Ma, Z., Chen, X., Sun, T., Wang, X., Wu, Y.C., Zhou, M. (2024). Blockchain-based zero-trust supply chain security integrated with deep reinforcement learning for inventory optimization. Future Internet, 16(5): 163. https://doi.org/10.3390/fi16050163
[20] Khan, I., Ali, Q.E., Hadi, H.J., et al. (2024). Securing blockchain-based supply chain management: Textual data encryption and access control. Technologies, 12(7): 110. https://doi.org/10.3390/technologies12070110
[21] Yang, X., Liu, J. (2024). HA-Med: A blockchain-based solution for sharing medical data with hidden policies and attributes. IET Information Security, 2024(1): 2498245. https://doi.org/10.1049/2024/2498245
[22] Priyadharshini, A., Nithiya, S., Archana, H.A., Jaganathan, S.C.B. (2025). Intelligent patient monitoring through hybrid consensus algorithm based blockchain technology. Multimedia Tools and Applications, 84(18): 19579-19600. https://doi.org/10.1007/s11042-024-19840-2
[23] Alami, R., Biswas, A., Shinde, V., Almogren, A., Ur Rehman, A., Shaikh, T. (2024). Blockchain enabled federated learning for detection of malicious Internet of Things nodes. IEEE Access, 12: 188174-188185. https://doi.org/10.1109/ACCESS.2024.3511272
[24] Karadag, B., Halim Zaim, A., Akbulut, A. (2024). Blockchain-based KYC model for credit allocation in banking. IEEE Access, 12: 80176-80182. https://doi.org/10.1109/ACCESS.2024.3410874
[25] Wu, J., Bu, X., Li, G., Tian, G. (2024). Data privacy protection model based on blockchain in mobile edge computing. Software: Practice and Experience, 54(9): 1671-1696. https://doi.org/10.1002/spe.3315

NOMENCLATURE

AS	Autonomous System
BFT	Byzantine Fault Tolerance
CVaR	Conditional Value at Risk
DPoS	Delegated Proof of Stake
EHR	Electronic Health Record
IBF	IoT-Blockchain-Fog
IoT	Internet of Things
HML-BFT	Hybrid Multi-Layer Byzantine Fault Tolerance
LVCA	Lightweight Voting-based Consensus Algorithm
MEC	Mobile Edge Computing
PBFT	Practical Byzantine Fault Tolerance
PoFQ	Proof of Federated Quality
PoPR	Proof of Participatory Reliability
PoR	Proof of Reputation
PoS	Proof of Stake
PoW	Proof of Work
RISKFLOW-LR	Risk-Budgeted Leader and Relay Rotation
TPS	Transactions Per Second
ZK	Zero-Knowledge
ZK-CEAA	Zero-Knowledge Compressed Energy and Availability Attestation
LaTSE	Latency-Trust Spectral Embedding
IQPD	Intersecting Quorum Polytope Design
PCFF	Polytope-Certified Fast Finality