JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-21, No.-08, August (2026) pp 36-55 ISSN (Print) 0973-8975

CROSS-LAYER LATENCY IN BLOCKCHAIN-ENABLED HIOT: A COMPREHENSIVE SURVEY

Sajal Chakraborty[1], Souritra Mandal[2], Subham Ghosh[3], Ankan Bhattacharya[4], Sankar Mukherjee[5], Biswa Mohan Acharya[6] Nabajyoti Medhi[7]

[1,6] Dept. of CSE, ITER, S 'O' A University, Bhubaneswar, Odisha-751030, India.

[2,3,4,5]Bengal College of Engineering and Technology, Durgapur, West Bengal-713212. India.

[7]Dept. of CSE, Tezpur University, Assam-784028, India.

Email: [1]sajal.cse2022@gmail.com, [2]souritramandal123@gmail.com
[3]sg5859563@gmail.com, [4]ankanbhattacharya08@gmail.com
[5]sanmukher@gmail.com, [6]biswaacharya@soa.ac.in, [7]nmedhi@tezu.ernet.in

Corresponding Author: **Souritra Mandal**

https://doi.org/10.26782/jmcms.2026.08.00003

(Received: June 04, 2026; Revised: August 03, 2026; Accepted: August 14, 2026)

Abstract

The Healthcare Internet of Things (HIoT) supports real-time monitoring and clinical decision-making through interconnected medical devices and sensors. Fog computing reduces communication delay by processing data near its source, while blockchain improves security, integrity, and decentralized trust. However, filtering, encryption, authentication, queueing, and consensus introduce additional latency, particularly in resource-constrained HIoT environments. Existing studies mainly address system-level delay, while micro-latency from local processing, hashing, memory access, and verification remains comparatively less explored. This study reviews latency-reduction approaches across the data filtering and encryption layers using a micro–macro perspective. A dependency-aware model links local processing with propagation, queueing, consensus and storage. Reported results are interpreted according to their architectural and evaluation conditions, while component-level analysis distinguishes isolated effects from coupled architectural improvements, showing that end-to-end latency arises from interactions across multiple processing stages.

Keywords: Blockchain, HIoT, Latency, Fog Computing, Micro and Macro latency.

I. Introduction

Healthcare Internet of Things (HIoT) systems enable real-time monitoring and personalized care through interconnected medical devices, sensors, and digital

infrastructures, generating large volumes of time-sensitive data that require timely processing [I]. Tertiary hospitals such as SKS Medical College & Hospital, Durgapur illustrate this growing digital ecosystem through diagnostic systems, intensive care facilities, laboratories, and EHR services. However, cloud-centric architectures introduce transmission delay, congestion, centralized processing overhead, and security risks. Edge and fog computing reduce these delays by processing data near IoMT devices, while blockchain improves integrity, decentralized trust, and access control [XV].

Blockchain-enabled HIoT still faces resource limitations, storage and energy constraints, cybersecurity threats, and transaction processing overhead [XXXV] [V] [VII] [I]. Increasing device density further affects congestion and responsiveness [I] [XXXV]. Hence, this study reviews latency challenges in the data filtering and encryption layers and examines relevant architectures and optimization strategies [III] [XXIV] [XXXIV]. The key contributions of this study are:

- Develops a dependency-aware micro–macro latency framework that links filtering, cryptographic processing, propagation, queueing, consensus, and storage across blockchain-enabled HIoT systems.
- Provides a condition-aware comparison of reported latency results by considering latency metric, blockchain architecture, consensus mechanism, workload, network scale, hardware, and evaluation environment where available.
- Distinguishes isolated component effects from coupled architectural improvements to avoid attributing system-level latency gains to a single optimization stage without supporting evidence.

II. Background Study

Understanding latency in blockchain-enabled HIoT requires examining how data is generated, processed, secured, and transmitted across IoT, fog, and blockchain layers. This architectural view provides the basis for identifying major delay sources, system constraints, and future research needs.

II.i. HIoT architecture supports continuous monitoring and timely clinical decisions through interconnected medical devices, communication networks, and distributed processing resources [XXIV] [XI]. It generally comprises sensing, communication, processing, and application layers. Wearable devices and biosensors collect physiological data, which may contain noise and redundancy [XXIV] [XXVII]. Data are transmitted through Bluetooth, Wi-Fi, and 5G, where communication delay occurs [XXXIV]. Edge, fog, and cloud resources and their limitations remain significant [XXXV]. The application layer delivers services such as remote monitoring and clinical decision support.

II.ii. Fog architecture extends cloud computing by placing distributed processing nodes closer to HIoT devices, where filtering, aggregation, and temporary storage are performed before relevant data are forwarded to the cloud [XXIV], thereby reducing network traffic and supporting latency-sensitive healthcare services [XXII]. However, the distributed and relatively exposed nature of fog nodes increases the risk of unauthorized access and data tampering. Since conventional fog systems don't

inherently provide immutability or decentralized trust, these limitations motivate the integration of secure and verifiable data management mechanisms [XXXVI].

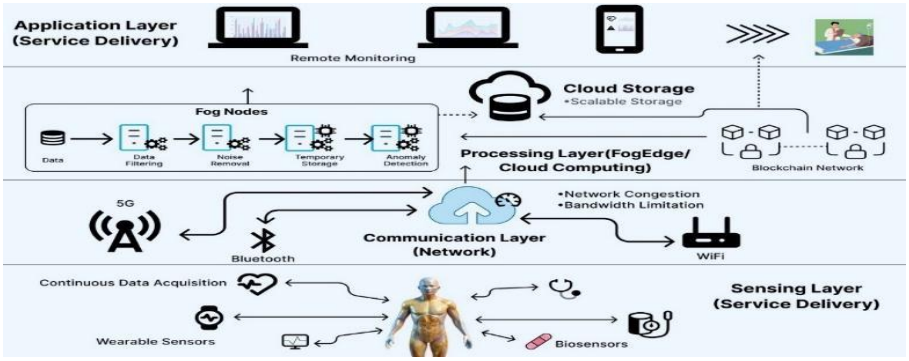

Fig. 1. Layer-wise Architecture of a Blockchain-Enabled HIoT System

II.iii. Blockchain architecture enables secure and decentralized data management through sequential processing stages that influence latency. Filtering and validation first remove redundant and noisy data before processing. The refined data then undergo encryption and hashing to ensure confidentiality and integrity, although these operations add computational overhead [III] [XI]. Digital signatures and authentication verify transactions, further increasing processing time [I]. Transactions are subsequently propagated across peers, where network topology and peer selection affect delay [XIV] [XIII] before consensus validates them to maintain trust and consistency. Confirmed data are then stored and retrieved from the distributed ledger, with performance depending on system design and indexing [V]. Finally, every data modification is recorded as a new transaction, preserving immutability and auditability at the cost of additional overhead [VIII].

II.iv. Cross-layer dependency model: Latency in blockchain-enabled HIoT emerges from a sequence of interdependent stages: data filtering → encryption/authentication → transaction propagation → queueing → consensus → storage. Although these delays are often evaluated separately, the reviewed architectures indicate that the output of one stage can alter the workload of the next. Fog/edge preprocessing reduces the volume of data forwarded upstream [XXII] [XXVI]. The cryptographic cost varies with payload size and available resources [XIX], and blockchain latency increases with transaction workload [XXVIII]. Accordingly, the end-to-end latency is expressed as:

$$L_{E2E} = L_f + L_e + L_a + L_p + L_q + L_c + L_s \quad (1)$$

Where L_f, L_e, L_a, L_p, L_q, L_c and L_s denote filtering, encryption, authentication, propagation, queueing, consensus and storage/retrieval latency, respectively. Equation (1) represents cumulative residence time; these components are dependent.

Let D_0 denote the raw HIoT data and D_f the volume after preprocessing. The filtering dependency is represented as:

$$D_f = g_f(D_0, \theta_f) \quad (2)$$

Where θ_f characterizes the filtering configuration, such as aggregation, compression, rule-based selection, or lightweight anomaly detection. In [XXII], fog nodes filter and

aggregate data before passing information, while [XXVI] applies selective edge processing to limit unnecessary transmission. Thus, filtering introduces local processing latency L_f while simultaneously reducing the workload presented to communication and blockchain stages. The resulting propagation latency can be represented, following the processing and transmission relationship considered in [XXXV], as:

$$L_p\,(D_f) = \frac{D_f}{B_{link}} + \frac{d}{v_{prop}} \qquad (3)$$

Where B_{link} is the available bandwidth, d is the transmission distance, and v_{prop} is the propagation speed. Hence, greater local filtering may increase L_f, yet the reduction in D_f can lower L_p, illustrating a direct micro-to-macro latency trade-off [XXXV].

A similar dependency occurs at the security layer. Let C_e denote the available cryptographic processing capacity and Ae the adopted encryption/authentication configuration. The associated latency is expressed as:

$$L_{ea} = L_e + L_a = f_{ea}\,(D_f, C_e, A_e) \qquad (4)$$

In [XIX][XXX] cryptographic operations are distributed between ESP32 devices and Raspberry Pi edge nodes, with AES-GCM processing increasing with message size. Likewise, [I] combines edge-assisted authentication and proxy re-encryption, showing that cryptographic workload and processing placement influence secure transaction latency. Therefore, security processing primarily contributes to micro latency, while its computational demand can also affect downstream blockchain processing. For blockchain processing, let λ_b denote the effective transaction arrival rate, μ_b the service rate, and Q_b the pending queue state. Queueing latency is represented as:

$$L_q = f_q\,(\lambda_b, \mu_b, Q_b),\ \mu_b = f_\mu\,(C_b, L_e, L_a\,) \qquad (5)$$

Where C_b denotes the processing capacity of the blockchain or associated edge/fog node. Increased cryptographic or authentication time can reduce the effective service rate and thereby increase transaction waiting. This workload sensitivity is evident in [XXVIII] where blockchain read/write latency generally increases as the number of transactions grows. Consensus latency is subsequently represented as:

$$L_c = f_c\,(N, P_c, \lambda_b, Q_b) \qquad (6)$$

Where N denotes the participating peers or validators, P_c represents the consensus mechanism and its configuration. The use of APBFT in the Hyperledger Indy-based framework [I] and the lightweight Proof-of-Authentication mechanism in [XXVIII] further indicates that validation latency depends on both workload and blockchain design.

The complete dependency chain is illustrated in Fig. 2 While Eq. (1) captures cumulative latency, Eqs. (2) - (6) describe how latency propagates across stages. Filtering modifies D_f and hence communication and transaction load; encryption and authentication affect the effective service rate μ_b , and the resulting arrival, service, and queue states influence consensus delay. Consequently, micro-level processing overhead can propagate macro-level transmission, queueing, and consensus latency. The formulation therefore provides a dependency-oriented analytical abstraction of the reviewed architectures rather than a numerically fitted model, since the underlying

studies employ heterogeneous workloads, hardware platforms, and blockchain configurations.

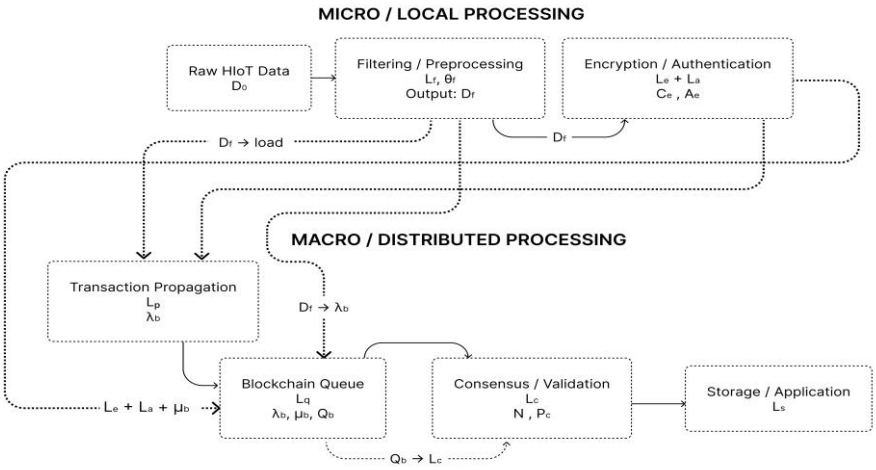

Fig. 2. Cross-layer Micro-Macro Latency Dependency in Blockchain HIoT
Solid arrows denote the processing sequence, while dashed arrows indicate dependencies that modify downstream workload or service conditions.

Equation (1) should be interpreted as the cumulative residence time of a record or transaction along the serial processing path. It does not imply statistical independence among the latency components. Parallel execution, retransmission, batching, and asynchronous processing are represented through the stage-dependent functions and system states rather than independent additive effects.

The network term in Eq. (3) should distinguish serialization/transmission delay from physical propagation delay. A consistent expression is:

$$\mathbf{L_p\ (D_f)} = \frac{D_f}{B_{link}} + \frac{d}{v_{prop}}$$

For continuous HIoT streams, transactions awaiting validation remain in the pending state Q_b until finality is reached. A longer consensus interval can therefore increase buffer occupancy and the waiting time experienced by subsequent arrivals, providing the queueing consensus feedback path that complements Eqs. (5) and (6). Overall, filtering can reduce downstream payload and arrival pressure, whereas cryptographic, authentication, propagation, queueing, consensus and storage operations contribute to different forms of local and distributed service overhead. Their net contribution to end-to-end latency depends on workload, resources and architectural coupling; therefore, a system-level gain is not assigned to a single stage unless the reviewed study provides component-level isolation evidence.

Where controlled component measurements are available, the corresponding within-study latency change is used to support the identified dependency path; otherwise, no numerical interaction coefficient is inferred from heterogeneous experiments.

III. Methodology

This study focuses on the latency issues at data filtering and encryption layers in blockchain-enabled HIoT systems.

III.i. Planning and Research Strategy: The literature review indicates that cross-layer latency mitigation across data filtering and encryption remains insufficiently examined [IX] [XV]. This study focuses on latency sources and reduction strategies in blockchain-enabled HIoT systems using evidence from multiple established academic databases.

III.ii. Paper selection Criteria: The papers were selected between 2020 and 2026; the study aims to explore the latest advancements in blockchain and HIoT. Search strings were created separately to include all the literature present regarding latency in blockchain-based healthcare IoT systems. For review papers: ("blockchain" AND ("latency" OR "delay" OR "real-time" OR "real time") AND ("IoT" OR "HIoT" OR "medical" OR "healthcare") AND ("survey" OR "review" OR "systematic review")) string was used and for implementation papers: ("blockchain" AND ("latency" OR "delay" OR "real-time" OR "real time" OR "data filtering" OR "pre-processing") AND ("IoT" OR "HIoT" OR "medical" OR "healthcare") AND ("framework proposal" OR "algorithm proposal" OR "implementation")).

Exclusion Criteria: i. Studies outside healthcare/HIoT, **ii.** Not addressing latency or system performance, **iii.** Duplicate publications. The selected papers were systematically analyzed based on system architecture; sources of latency (micro- and macro-levels); and latency optimization techniques, particularly in data filtering and encryption layers.

IV. Existing Research

Existing studies mainly optimize individual stages of blockchain-enabled HIoT, while the propagation of local computational delay across the complete transaction path remains less explored. Micro latency arises from filtering, hashing, encryption, signature processing, CPU computation, and memory access [III] [XIII], whereas macro latency is linked to communication, transaction propagation, queueing, validation, and distributed storage [XII] [XXIV] [XI] [XXXIV]. This supports a cross-layer view in which local and distributed delays interact.

At the filtering layer, major latency sources include processing [XXXV] [XXII] [VII], transmission [XXXV] [XXII] [XXVI], queueing [V], storage [VII] [XXV], and security/validation [XXXV] [XXI] [XXXIII]. These components are connected through Eqs. (2) – (6): preprocessing adds local cost but can reduce transmitted data and transaction pressure, while limited edge/fog resources may shift delay downstream. Queueing increases when transaction arrivals exceed processing capacity [V] and storage placement further affects access overhead [VII] [XXV].

Recent solutions therefore combine fog edge filtering [XXXV], local analytics [XXII], on/off chain storage [XXV], adaptive consensus [VII], SDN-based fog selection [XXI], queue-aware blockchain models [V], edge preprocessing [XXVI], dual-ledger designs

[XXXIII], and lightweight authentication [I]. Their reported gains mainly reflect coordinated architectural optimization rather than a single isolated mechanism.

Table 1: Data Filtering Layer

Ref.	Year	Proc.	Trans.	Storage	Queue	Sec./Val.
[I]	2025	✓	✗	✗	✗	✓
[V]	2024	✗	✗	✗	✓	✗
[VII]	2026	✓	✗	✓	✓	✓
[XXI]	2025	✓	✓	✗	✗	✓
[XXII]	2025	✓	✓	✗	✗	✓
[XXV]	2025	✓	✗	✓	✗	✓
[XXVI]	2025	✓	✓	✗	✗	✗
[XXXIII]	2025	✗	✓	✗	✓	✓
[XXXV]	2025	✓	✓	✗	✗	✗

Ref.→ *References;* ***Proc.***→ *Processing delay;* ***Trans.***→ *Transmission delay;* ***Storage***→ *Storage latency;* ***Queue***→ *Queueing delay;* ***Sec./Val.***→ *Security / validation delay.*

Table 1 shows that filtering-layer latency spans processing, transmission, storage, queueing, and security/validation. Local filtering, anomaly detection, and feature extraction mainly contribute to micro latency [XXXV] [XXXIII] while transmission, queueing, and distributed storage mainly affect macro latency [V] [XIV]. Security influences both levels; hence, filtering should be viewed as a trade-off between added local preprocessing and reduced downstream communication and blockchain workload [VII] [XXIV] [XXXIV].

The encryption layer is shaped by payload size, cryptographic complexity, and processing capacity. RSA/ECDSA, hashing, and PoW increase computational and confirmation overhead [VIII], whereas FHE introduces substantial arithmetic cost [II]. Optimized Paillier encryption and Bloom-filter-assisted search reduce cryptographic and retrieval delay [XXIX] while edge-assisted authentication and server-supported decryption shift intensive security tasks away from constrained devices [I] [XXXI].

Micro and macro latency remain coupled during secure transaction processing. Proof-of-Authentication and permissioned validation reduce PoW-related overhead [XXVIII], but increasing payload and transaction volume can still increase blockchain delay [XIX]. Split encryption, hardware acceleration, and optimized broadcast protocols further reduce local cryptographic and communication latency [XIX] [XI] [XXXVIII] [XX].

Table 2: Encryption Layer

Ref.	Year	Con.	Enc./Dec.	Comp.	Key Gen.	Auth.
[I]	2025	✓	✓	✗	✗	✓
[III]	2023	✗	✗	✓	✓	✗

[VIII]	2025	✓	✗	✗	✓	✗
[XI]	2025	✗	✓	✓	✗	✗
[XIX]	2025	✗	✓	✓	✗	✗
[XXVIII]	2023	✓	✗	✗	✗	✓
[XXIX]	2025	✗	✓	✓	✗	✗
[XXXI]	2026	✗	✗	✗	✓	✓

Ref.→ *Reference;* ***Con.***→ *Consensus validation delay;* ***Enc./Dec.***→ *Encryption / Decryption overhead;* ***Comp.***→ *Computational overhead;* ***Key Gen.***→ *Key generation delay;* ***Auth.***→ *Authentication delay.*

Encryption/decryption, key generation, and authentication mainly contribute to micro latency because they depend on local device, edge, or fog resources [XXVIII], whereas distributed authentication and consensus validation contribute to macro latency through multi-node coordination [XXIV]. The encryption layer therefore links local cryptographic cost with downstream blockchain processing and validation [XIV] [XXI] [XXXI], supporting a cross-layer approach to latency optimization.

V. Discussion

Section V identifies where micro and macro latency arise across data filtering, security processing, communication, blockchain validation, and storage. Section II.iv complements that taxonomy by modelling these stages as a dependency chain rather than independent delay classes. Accordingly, the present discussion does not rank frameworks by their reported percentage improvement. Instead, it examines the conditions under which each latency result was obtained and then interprets the result against the state variables of the cross-layer model. In particular, workload, network size, consensus design, and processing resources instantiate the variables λ_b, N, P_c, C_b , and $C_{e,}$ while the reported latency metric determines which term or set of terms in E_q. (1) is being observed. The values in Table 3 are intentionally interpreted in a condition-aware manner. A percentage improvement describes a framework relative to the comparator used in that study; it is not a common performance scale across the literature. The listed studies differ in blockchain design, consensus, workload, network scale, hardware, deployment environment, and even in what is called latency. Thus, the 87.5% clinical response-time improvement in [XXVI], the 50% system latency reduction in [XXII], the 39% transaction latency reduction in [VIII], and the 34% cryptographic processing reduction in [XXIX] represent different quantities. Direct numerical ordering of these values would therefore be analytically misleading. Heterogeneous results are used below to identify recurring dependency paths and optimization trends, while quantitative comparison is confined to matched or study-controlled conditions.

Table 3: Comparative latency characteristics of selected reviewed frameworks

Ref.	Architecture / Blockchain	Consensus	Evaluation condition	Latency metric	Absolute latency	Improvement	Comparison status
[I]	Edge/Hyperledger_Indy(consortium)	APBFT	Simulation; hardware NR	Authentication latency	NR / NR	−2.93%latency; +98.33% auth.efficiency	(C3)

Ref.	Architecture / Blockchain	Consensus	Evaluation condition	Latency metric	Absolute latency	Improvement	Comparison status
[VIII]	NR/Private chain(SHA256,ECDSA, RSA)	PoW	Simulation; Raspberry Pi vs. unspecified higher-RAM node; up to 500 tx.	Transaction_latency;storage_footprint	15.3 ms (10 tx, single node)/scaling to 45.8 ms at 500tx	−39% vs. state-of-the-art; −90% storage	(C2)
[XI]	Edge / embedded / NR	NR	Mixed (simulation + case study); FPGA / microcontroller / ASIC	Enc/verification latency	NR / <1 ms (verification); ≤5 ms target	≤30(secure comm.); 20% (AES module)	(NC)
[XIX]	Edge–Fog /Hyperledger Fabric	NR	Testbed; ESP32 (AES128)+Raspberry Pi	Enc on/ auth latency	NR / NR	NR	(NC)
[XXII]	Fog vs. cloud-only / NR	NR	Simulation (JMeter workload)	System-level latency	NR / NR (ms scale)	−50%	(C2)
[XXV]	Fog/Private chain + Ethereum	PoA / DPoS	Simulation; hardware NR	Data processing latency	NR / NR	−58.4(comparator)−48.4 (on-chain baseline)	(C3)
[XXVI]	Edge_middlewarevs.Cloud_centric/NR	NR	Field deployment (4-site consortium)	Clinical alert response time	3.2 s / 0.4 s	−87.5% (avg.); up to −94% (arrhythmia)	(C1)
[XXVIII]	NR/Private chain (CP-ABE)	NR	Simulation; increasing transaction groups	Read/write latency	61 ms / 16ms avg.(range14–74 ms)	Not reported as % change	(C3)
[XXXI]	Edge–Fog /Hyperledger Fabric	NR	Simulation; 10–50 epochs	Access/crypto latency	7.2–8.8 ms (prior schemes)/3.3–6.6 ms	≈−40%	(C2)
[XXXIII]	Edge/Hyperledger_Fabric (pBFT/Raft)	pBFT / Raft	Testbed (Raspberry Pi nodes); ~3400 TPS (PQC-tuned)	E2E transaction latency	NR / ~180 ms	Resource use reduced to 13.11% of PoS baseline	(C2)
[XXXV]	Fog–Edge vs. cloud-only / NR	NR	Testbed (16-core Xeon server + simulated edge); Intel Xeon (16-core, 64 GB RAM)	E2E latency	NR / NR	−70(headline); −50(sub-analysis)	(C2)

C1.→ *Comparable Within family only;* ***C2.→*** *Conditional;* ***C3.→*** *Within-Study-Only;* ***NC.→*** *Not Comparable;* ***NR.→*** *None reported.*

The strongest reductions at system-level are generally reported by architectures that move computation closer to the data source and reduce the amount of information

forwarded upstream. However, these results must be interpreted through Eqs. (2) and (3), not as evidence that filtering alone determines end-to-end performance. In [XXVI], edge processing reduced clinical alert response time from 3.2 s to 0.4 s while also decreasing cloud-bound traffic. The evaluated framework combined filtering, aggregation, local anomaly processing, contextual processing, and resource prioritization; consequently, the 87.5% response time improvement is a property of the edge architecture as a whole rather than a controlled filtering ablation [XXVI]. The same distinction applies to the hybrid fog edge architecture of [XXXV], where selective transmission, fog/edge placement, security controls, and local analytics were jointly introduced while a 70% system latency reduction was reported [XXXV]. These studies are useful for establishing the $D_0 \rightarrow D_f \rightarrow L_p$ dependency, but they are supporting architectural evidence rather than directly comparable blockchain performance measurements. A similar coupling appears in blockchain-associated fog frameworks. The system in [XXII] reduced measured latency from 150 ms to 75 ms through local fog processing while simultaneously incorporating smart contract security and AI-assisted processing [XXII]. FogChainFlow [XXV] further separates time-sensitive local processing from on-chain storage and reports approximately 0.71 s per dataset, together with low off-chain fog device latency of 0.109 – 0.153 s [XXV]. The result supports the architectural principle that reducing the volume and distance of data movement can lower L_p and relieve downstream workload. It does not, however, establish a universal percentage gain because the comparator methods and experimental conditions differ. Likewise, the blockchain-SDN-zero-trust framework in [XXI] reduces average response time to 180.47 ms from 299.08 ms in one baseline configuration, but its SDN routing, zero-trust validation, fog scheduling, and blockchain security mechanisms are modified together [XXI]. The system-level evidence therefore supports a cross-layer trend locality and workload reduction can reduce macro latency without isolating a single causal stage.

The transaction-level studies provide direct evidence for the workload-sensitive part of Eqs. (5) and (6). In [VIII], latency rises as the transaction and storage load increases: the study reports 15.3 ms for ten transactions on one node, while the storage-growth experiment increases from 22.1 ms to 45.8 ms as the transaction batch grows from 50 to 500 [VIII]. Similarly, [XXVIII] evaluates transaction groups from 100 to 1000 and reports average read and write latencies of 61 ms and 16 ms, respectively, with latency increasing as the transaction count grows [XXVIII]. These observations are consistent with the role of λ_b and Q_b in E_q. (5), but neither study provides a term-by-term measurement of queueing and consensus delay. The evidence therefore supports workload sensitivity of the blockchain service path rather than a claim that consensus alone caused the observed increase. If validation, storage and ledger placement are all changed at the same time, the role of architecture becomes clearer. The multi-ledger design combines Hyperledger Fabric, Holochain, IPFS, post-quantum cryptography and edge intelligence [XXXIII]; its high-scale extension reports approximately 3400 TPS at about 180 ms, while a permissioned PBFT + IPFS configuration is reported below 200 ms [XXXIII]. These figures demonstrate that transaction finality can remain sub-second under a carefully engineered permissioned architecture, but the contribution of consensus cannot be separated from the Holochain/Fabric division of labour, off-chain storage and edge processing. In [XIX], the average blockchain

transaction latency is approximately 0.13 s in a split-cryptography edge-Fabric implementation [XIX]. Because encryption placement, digital signatures, secure transport, and blockchain commitment coexist in the same execution path, the reported transaction latency similarly represents an integrated service chain. Based on the studies, it is concluded that Eq. (6) is more suitable as an analysis tool for the key variables N, P_c, λ_b, and Q_b, rather than a ranking criterion for different consensus algorithms.

The cryptographic computation is the source of the best micro-latency data since several research papers provide direct measurements of the local processes. For example, the AES-128-GCM encryption of the ESP32 grows from 11.32ms when using 10 KB to 56.49ms with 50 KB, indicating the influence of the payload size on the local security processing defined by Eq. (4) [XIX]. This local increase can also reduce the effective service capacity available to subsequent blockchain processing when secure transactions are produced continuously. The result therefore connects a micro-level cryptographic cost to the service-rate term μ_b in Eq. (5), although the source does not separately measure the resulting queueing increment. The privacy-preserving scheme in [XXIX] provides a different form of component evidence. Fast exponentiation reduces Paillier encryption/decryption time by an average of 34% across the tested key sizes, while Bloom-filter-assisted searchable encryption keeps ciphertext retrieval approximately constant as the stored data grows [XXIX]. These results are directly relevant to local cryptographic and retrieval latency, but they should not be translated into an equivalent end-to-end percentage because downstream propagation, queueing, and consensus were not isolated under the same experiment. The authentication framework in [I] combines Hyperledger Indy, APBFT, edge processing, and NuCypher threshold proxy re-encryption and reports a 2.93% latency reduction together with a 98.33% improvement in authentication efficiency [I]. Since the security, computation placement, and consensus mechanisms change together, the overall latency improvement is evidence for the integrated authentication architecture rather than for one cryptographic primitive. The ablation reported in [VIII] is particularly important for causal interpretation. At the same T10/node-1 configuration, the full SHA-256+RSA+ECDSA system records 15.3 ms latency; removing ECDSA reduces latency to 13.8 ms, removing RSA to 12.9 ms, and removing SHA-256 to 11.5 ms [VIII]. This controlled comparison demonstrates that individual security operations have measurable processing costs. It does not imply that these mechanisms should be removed: the source explicitly reports loss of signature security, access-control strength, or data integrity when the corresponding component is omitted. The relevant design problem is therefore not minimum latency in isolation, but minimum latency subject to the required security properties.

The only time in the communication layer where the literature on this topic has an intervention is during propagation. The Dual Perigee study evaluated peer selection in a matched 50-node private Ethereum PoA setup and reported decreases of 54.7% in transaction latency and 48.5% in block latency, respectively, when compared to default peering [XIV]. Because the blockchain, consensus family, and emulator environment remain unchanged but the peer selection strategy is varied, this research has higher component-level validation compared to the other frameworks. The outcome of broadcasting in [XXXIII] offers a complementary trade-off for the system in terms of

latency: the encrypted MQTT approach can achieve a broadcast latency of approximately 50 – 120 ms, while the blockchain-based covert channel is approximately 150 – 400 ms, but offers distributed auditability and more decentralized trust [XXXIII]. The numbers presented above cannot be seen as the comparison of two independent systems since they show that the latency constraint depends on the level of security and trust assumed. Thus, the propagation factor of Eq. (3) depends on the payload and link characteristics, as well as topology and replication and communication semantics selected for the blockchain architecture. Combined, it seems there is no universal ordering in which the same layer in the HIoT hierarchy results in the greatest delay or the same optimization results in the greatest improvement. However, there are recurring dependency patterns: filtering and edge/fog processing impact D_f and arrival pressure in the upstream path; topologies and peer selection impact propagation; transaction volume and resource availability impact queueing; and consensus/storage design impact finality and buffering in the downstream path. The dependencies are directly represented by the different paths shown in Section II.iv. The result of this is that the effects are workload- and architecture-dependent, and therefore Table 3 does not rank the effects as percentages.

VI. Analysis

The quantitative discussion in Section V establishes what each study measured and under which conditions. The remaining analytical question is whether a reported latency gain can be assigned to a particular component. This distinction is necessary because the dependency model in Section II.iv predicts interaction: a change to filtering can modify propagation and arrival pressure, a change to encryption/authentication can modify service time, and a change to communication or consensus can modify queue occupancy and finality.

Table 4: Component-level attribution of latency optimization

Ref.	Primary optimization	Other simultaneous changes	Reported latency outcome	Component isolated?	Interpretation
[I]	Authentication (hybrid)+Consensus modification (APBFT)	Edge/Fog offload (Y)	−2.93% latency; +98.33% auth. efficiency	No	Coupled (associative outcome; multiple concurrent changes)
[VIII]	Cryptography (SHA-256, ECDSA, RSA)	Auth./Access (Y); Storage (off-chain /dual)	−39% latency; −90% storage	Partial	Coupled (associative outcome; multiple concurrent changes)
[XI]	Cryptography (hardware-accelerated primitives)	Comm./Routing (Y); Edge/Fog offload (Y)	<1 ms verification; −20 to −30%	Partial	Isolated (locally measured, single-component effect)

Ref.	Primary optimization	Other simultaneous changes	Reported latency outcome	Component isolated?	Interpretation
[XIX]	Filtering	Cryptography (AES128); Auth./Access (JWT); Storage (on-chain); Edge/Fog offload (Y)	Not quantified in source	NR	Cannot isolate (no quantitative causal claim supported)
[XXII]	Edge/Fog offload	None reported (NR)	−50% latency; −30% energy	No	Coupled (associative outcome; multiple concurrent changes)
[XXV]	Consensus modification (PoA/DPoS) + Storage (off-chain)	Edge/Fog offload (Y)	−48.4 to −58.4%	No	Coupled (associative outcome; multiple concurrent changes)
[XXVI]	Filtering (smart filtering)	Comm./Routing (Y); Edge/Fog offload (Y)	−87.5% (avg.); −94% (arrhythmia)	No	Coupled (associative outcome; multiple concurrent changes)
[XXVIII]	Cryptography / access control (CP-ABE)	Storage (on-chain)	61 ms / 16 ms avg. read/write	No	Cannot isolate (no quantitative causal claim supported)
[XXXI]	Cryptography (ECC)	Auth./Access (Y); Storage (on-chain); Edge/Fog offload (Y)	≈−40% vs. prior schemes	Partial	Isolated (locally measured, single-component effect)
[XXXIII]	Cryptography (PQC) + Consensus tier (pluggable, untouched)	Comm./Routing (Y); Storage (on-chain); Edge/Fog offload (Y)	~180 ms E2E at 3400 TPS; resource use −13.11%	Partial	Coupled (associative outcome; multiple concurrent changes)
[XXXV]	Edge/Fog offload	Comm./Routing (Y)	−70% (headline); −50% (sub-analysis)	Partial	Coupled (associative outcome; multiple concurrent changes)

Table 4 therefore maps each principal implementation result to its optimization mechanism and explicitly records whether the component was isolated. “Partial” denotes direct component timing or partial ablation without a complete system-level isolation. “No” denotes an integrated architecture in which multiple mechanisms change concurrently without component-level control. “NR” is used only when the source does not provide enough information to determine isolation.

Table 4 changes the interpretation of the literature importantly. Most large end-to-end or system-level improvements are reported by integrated architectures. The fog, edge, blockchain, routing, storage, and security mechanisms in [XXII] [XXV] [XXXIII] [XXXV] are introduced in combination; consequently, their system-level latency improvements must be attributed to the combined architecture. In contrast, [VIII] offers a controlled peer-selection comparison within a common PoA environment and includes a cryptographic ablation at a fixed transaction/node condition. This set of work provides further evidence in favor of latency effects at the level of components. What differentiates this case from the one described by the reviewer is that the association of some architectural feature with lower end-to-end latency does not mean that the feature caused this effect.

The evidence has been interpreted as a dependency, and not as a universal percentage gain, as all the reviewed frameworks encourage the early removal of redundant or non-critical data at the data filtering stage. The pre-processing changes D_f as per Eq. (2) in the notation of Section II.iv If D_f decreases, the serialization portion of L_p can be decreased, and the number of transactions or the size of transactions can be decreased, which can lead to a decrease in λ_b before the blockchain queue. This is why edge/fog frameworks, such as [XXII] [XXVI] [XXXV], claim to have lower system or clinical response latency and lower upstream traffic [XXII] [XXVI] [XXXV]. The same goes for Fog-Chain-Flow [XXV], which demonstrates the ability to move time-sensitive processing off-chain and nearer to the data source. However, these frameworks also modify computation placement, scheduling, routing, security processing, or storage. The filtering itself adds L_f, and more selective or intelligent preprocessing can add more local computation, but at the same time reduce the communication and blockchain workload downstream. The evidence in the system-level studies is not adequate to support the stronger claim that the end-to-end reduction is solely due to filtering, but it does support the pathway filtering → lower payload/arrival pressure → lower downstream latency.

The encryption layer shows more transparency of local cryptographic processing and downstream system latency. The local dependence is made explicit in equation (4) by the payload D_f, the cryptographic capacity C_e, and the chosen security configuration. As seen in [XIX], the processing time of local AES-GCM increases with the size of the message. If the local security-processing interval is long, it will lower the transaction delivery rate to the blockchain service path under continuous transaction generation. This establishes a dependency chain involving cryptographic workload, service time, μ_b, queueing, and consensus, as represented by Eqs. (4) – (6). The security-latency trade-off is also reflected in the component evidence. For each removal in [VIII], there is measurable latency for ECDSA, RSA, and SHA-256, and each removal reported in [VIII] corresponds to the weakening of a specific security property. Similarly, [XXIX] optimizes the cost of the Paillier encryption/decryption algorithm, but does not propose that security operations can be neglected [XXIX]. These findings argue for the removal of cryptographic controls, but not in a blanket fashion and for lightweight implementation, acceleration and appropriate offloading. The integrated authentication result in [I] should be interpreted on the same basis: its improvement reflects the combined effect of edge placement, permissioned validation and proxy re-encryption rather than a separately measured contribution from any one mechanism [I].

Communication and blockchain validation are the most obvious examples of macro latency that cannot be considered as a set of independent categories. The authors in [XIV] show that the topology and peer selection can make a big difference in transaction and block propagation in a controlled peer-selection experiment. Redundant propagation can cause more traffic and queue pressure, and delayed delivery can cause transactions or blocks to take longer to be available to validators. Both exhibit increased latency with increasing workload at the transaction-processing stage [VIII] [XXVIII] as predicted by the λ_b–μ_b–Q_b dependency in E_q. (5). Consensus and storage then alter the time period of pending state. The integrated designs in [XXXIII] [XXV] reduce ledger pressure either by multi-ledger or on/off chain placement, but they also change the storage, propagation, and validation behavior at the same time [XXV] [XXXIII]. Therefore, the overall end-to-end enhancement should be considered as a combined architectural effect. This interpretation also avoids double counting: If a reduction occurs after a move off-chain, it is not a filtering gain, a storage gain, and a consensus gain unless the study has separate controlled measurements.

The condition-aware and attribution-aware synthesis reveals that the major limitation of the current evidence base is not a shortage of reported latency improvements, but the limited ability to compare and decompose them. First, latency definitions remain inconsistent: clinical response time, blockchain transaction latency, cryptographic execution time, access latency and broadcast latency are often discussed together despite measuring different segments of E_q. (1). Second, workload is incompletely reported in many studies; transaction rate, payload size, concurrent devices, validator/peer count and queue occupancy are necessary to interpret λ_b, N and Q_b. Third, hardware and computation placement are frequently omitted even though C_e and C_b directly determine service capacity. Fourth, very few studies report stage-by-stage measurements of L_f, L_e, L_a, L_p, L_q, L_c and L_s under the same workload. A stronger experimental design for future blockchain-enabled HIoT systems should therefore combine absolute latency reporting with controlled ablation. At minimum, a study should preserve the same workload, network scale and hardware while changing one optimization mechanism at a time. Where mechanisms are expected to interact, factorial or staged experiments can quantify interaction terms rather than attributing the full end-to-end gain to a single layer. Queue occupancy, arrival rate, effective service rate, and finality time should be logged together so that the micro-to-macro paths in Eqs. (4) – (6) can be tested empirically. Such measures would transform the current dependence theory from an abstract analysis on survey data to a quantifiable cross-layer model.

It is not possible to categorize the analyzed frameworks in terms of their performance ranking based on the reduction percentages since they employ different latency measures and testing conditions. However, based on the evidence presented, the following four optimization strategies can be recognized: edge/fog processing minimizing the amount of incoming data and reducing the load of arriving data; cryptography acceleration/ offloading, decreasing the time of local services; topology-aware communication, minimizing transaction block propagation; and permissioned, lightweight/distributed ledgers, minimizing the validation and storage load.

In the context of Section II.iv, the synthesis above helps make the main cross-layer point of the survey without overstating causal relationships. Processing decisions locally impact the state passed further, which may be transmitted via communication, queuing, consensus, and storage. In this sense, it is the most convincing evidence that can be used for drawing dependency paths and making trade-offs rather than pointing at one dominating layer of latency. When a source gives isolated measurement, e.g., peer selection isolation in [XIV] or cryptographic isolation in [VIII], component-level analysis is justified; otherwise, the observed effect is due to the whole system architecture.

VII. Conclusion

This study demonstrates a cross-layer analysis of latency in blockchain-supported HIoT systems with a dependency-aware micro-macro framework. The analysis reveals that end-to-end latency is dependent between stages. Data volume and transaction load downstream of the filtering stage affect subsequent processing, while cryptographic and authentication operations impact local service time, propagation, queueing, consensus, and storage, which impact distributed processing delay. The comparative analysis also reveals that the latency reduction reported in the studies cannot be considered as a common performance scale due to the differences in the architecture, consensus, workload, hardware, and network size of the studies reviewed, as well as the different definitions of latency. Likewise, numerous system-level enhancements come from concurrent changes in filtering, communication, storage, security, and blockchain processing; thus, the impact of these changes can't always be attributed to a single component.

These findings support the dependency-based model rather than a universal order of latency sources. To make sure that latency is minimized to its best possible level, all these factors need to be balanced and optimized in the right manner. This means that future research will have to provide concrete numbers for the absolute latency at each of these layers, with some workloads and hardware, but still keeping the desired security levels.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

I. A. A. Khan, A. A. Laghari, R. Alroobaea, A. M. Baqasah, M. Alsafyani, H. Alsufyani, and S. Ullah, “A lightweight scalable hybrid authentication framework for Internet of Medical Things (IoMT) using blockchain hyperledger consortium network with edge computing,” Scientific Reports, vol. 15, no. 1, p. 19856, 2025.

II. A. Ali, B. A. S. Al-Rimy, F. S. Alsubaei, A. A. Almazroi, and A. A. Almazroi, "HealthLock: Blockchain-based privacy preservation using homomorphic encryption in Internet of Things healthcare applications," Sensors, vol. 23, no. 15, p. 6762, 2023.

III. A. Hashim, R. Rulaningtyas, and K. Chellappan, "Optimizing security and latency in blockchain-based health data management: A Hyperledger Fabric approach with multi-objective optimization," Jurnal Kejuruteraan, vol. 37, no. 5, pp. 2469–2485, 2025.

IV. A. Shahidinejad and J. Abawajy, "An all-inclusive taxonomy and critical review of blockchain-assisted authentication and session key generation protocols for IoT," ACM Computing Surveys, vol. 56, no. 7, pp. 1–38, 2024.

V. A. U. Rehman, N. Tariq, M. A. Jan, F. Khan, H. Song, and M. Ibrahim, "A blockchain-based hybrid model for IoMT-enabled intelligent healthcare system," IEEE Transactions on Network Science and Engineering, vol. 11, no. 4, pp. 3512–3521, 2024.

VI. C. Bollineni, M. Sharma, A. Hazra, P. Kumari, S. Manipriya, and A. Tomar, "IoT for next-generation smart healthcare: A comprehensive survey," IEEE Internet of Things Journal, vol. 12, no. 16, pp. 32616–32639, 2025.

VII. C. H. V. N. U. B. Murthy and M. Lawanya Shri, "An adaptive blockchain framework for federated IoMT with reinforcement learning-based consensus and resource forecasting," Scientific Reports, 2026.

VIII. G. Niu, "A blockchain-based secure and privacy-preserving healthcare data management framework with SHA-256 and PoW consensus," Informatica, vol. 49, no. 20, 2025.

IX. H. Gupta and P. Verma, "A survey on data security and latency frameworks in healthcare using blockchain," in 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 1745–1750, 2024.

X. H. Taherdoost, "Blockchain-based Internet of Medical Things," Applied Sciences, vol. 13, no. 3, p. 1287, 2023.

XI. J. Babatope, "Designing energy-efficient cryptographic modules for securing real-time data communication in embedded Internet of Things (IoT) devices," International Research Journal of Modernization in Engineering, Technology and Science, vol. 7, no. 8, p. 1995, 2025.

XII. J. E. Abang, H. Takruri, R. Al-Zaidi, and M. Al-Khalidi, "Latency performance modelling in Hyperledger Fabric blockchain: Challenges and directions with an IoT perspective," Internet of Things, vol. 26, p. 101217, 2024.

XIII. K. Koshikawa, J.-D. Kim, W.-J. Hwang, K. Nguyen, H. Sekiya, "Dual Perigee: Reducing Latency in IoT Blockchain through Efficient Peer Selection", Proc. IEEE 99th Vehicular Technology Conference (VTC2024-Spring), pp. 1–5, 2024.

XIV. K. Koshikawa, Y. Su, J.-D. Kim, W.-J. Hwang, Z. Li, K. Nguyen, and H. Sekiya, "Impacts of overlay topologies and peer selection on latencies in IoT blockchain," IEEE Transactions on Network and Service Management, vol. 23, pp. 1630–1646, 2025.

XV. M. Anjum, N. Kraiem, H. Min, A. K. Dutta, Y. I. Daradkeh, and S. Shahab, "Opportunistic access control scheme for enhancing IoT-enabled healthcare security using blockchain and machine learning," Scientific Reports, vol. 15, no. 1, p. 7589, 2025.

XVI. M. A. Obaidat, M. Rawashdeh, M. Alja'afreh, M. Abouali, K. Thakur, and A. Karime, "Exploring IoT and blockchain: A comprehensive survey on security, integration strategies, applications and future research directions," Big Data and Cognitive Computing, vol. 8, no. 12, p. 174, 2024.

XVII. M. Ejaz, T. Kumar, I. Kovacevic, M. Ylianttila, and E. Harjula, "Health-BlockEdge: Blockchain-edge framework for reliable low-latency digital healthcare applications," Sensors, vol. 21, no. 7, p. 2502, 2021.

XVIII. M. Kumar, A. Kumar, S. Verma, P. Bhattacharya, D. Ghimire, S.-H. Kim, and A. S. S. Hosen, "Healthcare Internet of Things (H-IoT): Current trends, future prospects, applications, challenges and security issues," Electronics, vol. 12, no. 9, p. 2050, 2023.

XIX. M. Q. Al-Tuma and M. K. Ibrahem, "Enhanced secure blockchain-edge-Internet of Things system for real-time healthcare monitoring," International Journal of Intelligent Engineering and Systems, vol. 18, no. 10, pp. 494–514, 2025.

XX. M. Zhou, L. Zeng, Y. Han, P. Li, F. Long, D. Zhou, I. Beschastnikh, and M. Wu, "Mercury: Fast transaction broadcast in high performance blockchain systems," in IEEE INFOCOM 2023 - IEEE Conference on Computer Communications, pp. 1–10, 2023.

XXI. N. Kaur, A. Mittal, U. K. Lilhore, S. Simaiya, S. Dalal, K. Saleem, and E. S. Ghith, "Securing fog computing in healthcare with a zero-trust approach and blockchain," EURASIP Journal on Wireless Communications and Networking, vol. 2025, no. 1, p. 5, 2025.

XXII. N. M. Ali, Y. A. El Ghafar, and G. A. El Khayat, "Reducing latency and enhancing data security in healthcare IoT systems using fog computing," in 2025 International Conference on Machine Intelligence and Smart Innovation (ICMISI), pp. 7–12, 2025.

XXIII. N. S. Al-Blihed, N. F. Al-Mufadi, N. T. Al-Harbi, I. A. Al-Omari, and M. A. Al-Hagery, "Blockchain and machine learning in the Internet of Things: A review of smart healthcare," IAES International Journal of Artificial Intelligence, vol. 12, no. 3, pp. 995–1006, 2023.

XXIV. O. Cheikhrouhou, K. Mershad, F. Jamil, R. Mahmud, A. Koubaa, and S. R. Moosavi, "A lightweight blockchain and fog-enabled secure remote patient monitoring system," Internet of Things, vol. 22, p. 100691, 2023.

XXV. P. Karthikeyan and K. Brindha, "FogChainFlow: On-off blockchain data management for optimizing IoT healthcare," Cluster Computing, vol. 28, no. 16, p. 1011, 2025.

XXVI. R. R. Pai and B. Pandey, "Edge computing for healthcare IoT: Designing distributed data processing architectures for clinical devices," in Proc. IEEE Int. Conf. on Compute, Control, Network & Photonics (ICCCNP), pp. 1–6, 2025.

XXVII. S. Akter, M. Tabassum, R. Sultana, and M. S. H. Bhuiyan, "Blockchain for real-time healthcare data acquisition: A systematic review of sensor network applications and challenges," American Journal of Interdisciplinary Studies, vol. 6, no. 1, pp. 208–235, 2025.

XXVIII. S. Badri, S. U. Jan, D. M. Alghazzawi, S. Aldhaheri, and N. Pitropakis, "BIoMT: A blockchain-enabled healthcare architecture for information security in the Internet of Medical Things," Computer Systems Science and Engineering, vol. 46, no. 3, pp. 3667–3684, 2023.

XXIX. S. Guan, Y. Cao, and Y. Zhang, "Blockchain-enhanced data privacy preservation and secure sharing scheme for healthcare IoT," IEEE Internet of Things Journal, vol. 12, no. 5, pp. 5600–5614, 2025.

XXX. S. Lee, M. Kim, J. Lee, R.-H. Hsu, M.-S. Kim, and T. Q. S. Quek, "Facing latency of Hyperledger Fabric for blockchain-enabled IoT: Modeling and analyses," IEEE Network, vol. 37, no. 6, pp. 232–239, 2023.

XXXI. S. Muthuvel, S. Priya, and K. Sampath Kumar, "Design of a multi-layered privacy-preserving architecture for secure medical data exchange in cloud environments," Scientific Reports, vol. 16, no. 1, 2026.

XXXII. S. M. and V. K. Chattu, "A review of artificial intelligence, big data, and blockchain technology applications in medicine and global health," Big Data and Cognitive Computing, vol. 5, no. 3, p. 41, 2021.

XXXIII. S. Ravisankar and R. Maheswar, "SecureEdge-MedChain: A post-quantum blockchain and federated learning framework for real-time predictive diagnostics in IoMT," Sensors, vol. 25, no. 19, p. 5988, 2025.

XXXIV. T. E. Ali, F. I. Ali, P. Dakić, and A. D. Zoltan, "Trends, prospects, challenges and security in the healthcare Internet of Things," Computing, vol. 107, no. 1, p. 28, 2025.

XXXV. U. Islam, M. N. Alatawi, A. Alqazzaz, S. Alamro, B. Shah, and F. Moreira, "A hybrid fog-edge computing architecture for real-time health monitoring in IoMT systems with optimized latency and threat resilience," Scientific Reports, vol. 15, no. 1, p. 25655, 2025.

XXXVI. W. V. Solis, J. M. Parra-Ullauri, and A. Kertesz, "Exploring the synergy of fog computing, blockchain and federated learning for IoT applications: A systematic literature review," IEEE Access, vol. 12, pp. 68015–68060, 2024.

XXXVII. Y. Singh, M. A. Jabbar, S. K. Shandilya, O. Vovk, and Y. Hnatiuk, "Exploring applications of blockchain in healthcare: Road map and future directions," Frontiers in Public Health, vol. 11, p. 1229386, 2023.

XXXVIII. Y. Zhu, C. Hua, D. Zhong, and W. Xu, "Design of low-latency overlay protocol for blockchain delivery networks," IEEE Wireless Communications and Networking Conference (WCNC), pp. 1182–1187, 2022.