International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Multiple Adaptive Storage Organization for Secure and Scalable Healthcare Systems Using Block-chain Technology: A Comprehensive Survey

Divyashree G[1], Dr. Basavanna M[2], Dr. Latha M[3]

[1]Research Scholar & Assistant Professor, Department of ECE & GSSSIETW, Mysuru. Karnataka
[2]Research Advisor & Assistant Professor, Department of ECE & GSSSIETW, Mysuru. Karnataka
[3]Research Co-Advisor & Associative Professor, Department of ECE & GSSSIETW, Mysuru. Karnataka

Abstract

The rapid growth of the Internet of Things (IoT) produces large volumes of heterogeneous data, creating challenges for conventional centralized storage in terms of latency, scalability, and fault tolerance. Adaptive storage addresses these issues by dynamically placing data across edge, fog, cloud, and local nodes according to network conditions and resource availability. However, IoT environments remain vulnerable to data tampering and unauthorized access. Block-chain can enhance security through decentralized, tamper-resistant data management, supporting integrity, transparency, and access control. This review examines IoT-based adaptive storage systems integrated with block-chain, focusing on storage architectures, dynamic data placement, security mechanisms, and performance metrics. Key challenges and trade-offs related to scalability, consensus latency, storage overhead, and energy consumption are analyzed. Emerging directions, including lightweight block-chain, AI-driven storage, and edge intelligence, are also discussed to support secure and efficient IoT data management.

Keywords: "Internet of Things" (IoT), "Block-chain", "Adaptive Storage Systems"," Multi-Tier Tiering", " Data Integrity"," Edge-Fog-Cloud Continuum".

I. INTRODUCTION

The rapid growth of the Internet of Things (IoT) generates massive volumes of heterogeneous data, posing critical challenges for traditional centralized storage in latency, scalability, and fault tolerance. Adaptive storage systems mitigate these limits by dynamically distributing data across edge, fog, cloud, and local nodes based on real-time network conditions and resource availability. However, securing resource-constrained IoT networks against data tampering and unauthorized access remains a major vulnerability. Integrating block-chain technology introduces a decentralized, tamper-resistant ledger that ensures data integrity, transparency, and access control. This review paper presents a comprehensive survey of adaptive storage systems combining IoT and block-chain technology. We systematically analyze existing storage architectures, dynamic data placement strategies, block-chain integration models, security mechanisms, and performance metrics. Additionally, core trade-offs—including scalability, consensus latency, storage overhead, and energy consumption—are critically evaluated.

Finally, we outline emerging research directions, such as AI-driven adaptive storage, lightweight block-chain frameworks, and edge intelligence, providing a roadmap for next-generation secure storage solutions.

A. Background

The Internet of Things (IoT) connects diverse devices, sensors, and intelligent systems that continuously generate large volumes of heterogeneous data. Applications such as healthcare, smart cities, agriculture, transportation, and industrial automation require storage systems that can provide secure, reliable, scalable, and low-latency data access.

Cloud storage offers high capacity and computing resources, but centralized architectures can increase latency, bandwidth usage, operating costs, and dependence on remote infrastructure. These limitations are critical for IoT applications requiring rapid data processing. Edge and fog computing reduce communication distance by providing resources closer to data sources; however, efficient data management across distributed and heterogeneous nodes remains challenging.

Adaptive storage addresses this challenge by dynamically placing data across local, edge, fog, and cloud resources according to data characteristics, application needs, network conditions, and available resources. This approach can improve resource utilization, response time, and fault tolerance.

Block-chain adds decentralized verification, tampers resistance, and traceability to distributed storage, supporting data integrity, authentication, and controlled access. However, its integration with IoT may introduce consensus delay, storage overhead, scalability issues, and additional energy consumption.

Recent advances in AI, ML, and edge intelligence enable predictive data placement, resource optimization, and automated fault management. Therefore, combining adaptive storage with block-chain provides a promising approach for secure and efficient IoT data management.

B. Context and Motivation

The rapid growth of IoT generates large volumes of data from diverse devices and applications, creating challenges in storage, latency, security, and resource management. Centralized cloud storage can increase network traffic and delay while creating security and reliability concerns. In addition, IoT devices have different processing, memory, storage, and energy capabilities. These limitations require adaptive storage strategies that dynamically distribute data across local, edge, fog, and cloud resources according to system conditions and application needs.

[C]. Contributions of This Review

This review provides a structured analysis of block-chain-enabled adaptive storage for IoT environments. The major contributions are:

1. **Architecture Review:** Analyzes storage models across the edge–fog–cloud continuum and their integration with IoT.
2. **Adaptive Storage Analysis:** Reviews dynamic data placement and resource-allocation techniques, including AI and reinforcement-learning-based approaches.
3. **Block-chain Security:** Examines block-chain mechanisms for data integrity, authentication, access control, and secure data sharing, including on-chain and off-chain storage.
4. **Comparative Analysis:** Compares existing approaches using key metrics such as latency, scalability, storage overhead, energy consumption, security, and resource utilization.
5. **Taxonomy and Research Gaps:** Develops a structured taxonomy of adaptive storage approaches and identifies challenges and future research opportunities for secure, intelligent, and scalable IoT storage.

[D]. Related Works

1. The integration of the Internet of Things (IoT) with block-chain technology has attracted significant attention from researchers due to the increasing demand for secure, scalable, and adaptive data storage solutions. Numerous studies have focused on addressing the challenges of data management, security, privacy, and storage optimization in IoT environments. This section reviews the major contributions reported in the literature.
2. Early IoT storage systems primarily relied on centralized cloud computing infrastructures because of their high storage capacity and computational capabilities. Although cloud-based architectures provide scalability and ease of management, they suffer from several limitations, including high communication latency, bandwidth consumption, single points of failure, and vulnerability to cyber attacks. These limitations have motivated researchers to investigate distributed and adaptive storage frameworks.
3. With the emergence of edge and fog computing, several adaptive storage architectures have been proposed to process and store data closer to IoT devices. Edge-based storage reduces latency and network traffic by performing local computation, while fog computing extends these capabilities by providing intermediate storage and processing resources between edge devices and cloud servers. Hybrid cloud–fog–edge architectures have demonstrated improved Quality of Service (QoS), reduced response time, and efficient bandwidth utilization for delay-sensitive IoT applications.
4. Block-chain technology has been increasingly incorporated into IoT storage systems to enhance security, transparency, and trust. Distributed ledger technology eliminates the dependence on centralized authorities and provides immutable transaction records through cryptographic hashing and consensus mechanisms. Smart contracts further automate data access control, authentication, and integrity verification, enabling secure and decentralized data management. However, directly storing large volumes of IoT data on block-chain networks remains impractical due to storage limitations, transaction costs, and scalability concerns.
5. To address these challenges, researchers have proposed hybrid storage models that combine block-chain with off-chain storage mechanisms such as cloud databases, edge servers, distributed file systems (e.g., IPFS), and fog nodes. In these approaches, block-chain stores metadata, hashes, or access policies, while the actual IoT data are maintained in external storage systems. This strategy significantly reduces block-chain storage overhead while preserving data integrity and traceability.
6. Recent studies have introduced adaptive storage mechanisms that dynamically select storage locations based on multiple parameters, including data sensitivity, latency requirements, available storage capacity, network conditions, energy consumption, and security requirements. Machine learning and artificial intelligence techniques have also been explored to predict storage demands, optimize resource allocation, and improve decision-making in heterogeneous IoT environments. These intelligent approaches contribute to enhanced scalability and efficient utilization of distributed resources.
7. Security and privacy remain major research concerns in block-chain-enabled IoT storage systems. Existing literature proposes encryption techniques, attribute-based access control, identity management frameworks, lightweight authentication protocols, and privacy-preserving consensus mechanisms to protect sensitive IoT data. Nevertheless, resource-constrained IoT devices often face challenges in executing computationally intensive block-chain operations, prompting the development of lightweight block-chain protocols and energy-efficient consensus algorithms.

8. Although considerable progress has been achieved, several research gaps remain. Many existing solutions focus primarily on either storage optimization or security without jointly addressing scalability, interoperability, adaptive decision-making, and resource efficiency. Furthermore, comprehensive comparative analyses of multiple adaptive storage frameworks integrating IoT, block-chain, edge computing, fog computing, and artificial intelligence are limited. Therefore, this review systematically analyses existing adaptive storage architectures, classifies current approaches, compares their strengths and limitations, and identifies future research directions toward secure, scalable, and intelligent IoT storage systems.

II. REVIEW METHODOLOGY

A transparent review process is required so that the survey can be reproduced and its conclusions traced to the literature. The following protocol is proposed for execution before submission. The paper should not claim a completed systematic review until the searches, duplicate removal, screening, and full-text assessment have actually been performed.

A. Search Strategy and Sources

Recommended scholarly sources include IEEE Xplore, Scopus, Web of Science, ACM Digital Library, Springer-Link, and Science-Direct. Google Scholar may be used for supplementary discovery and forward/backward citation tracking. Record the date of each search, database, exact query, filters, number of records retrieved, and exported citation file.

Example search string: ("adaptive storage" OR "data placement" OR "storage optimization" OR "storage tiering" OR "resource allocation") AND ("Internet of Things" OR IoT) AND (block-chain OR "distributed ledger") AND (edge OR fog OR cloud).

B. Eligibility Criteria and Screening

Include peer-reviewed journal and conference papers that describe storage architecture, adaptive placement, IoT data management, edge/fog/cloud resource optimization, or block-chain-assisted data integrity and access control. Foundational works may be included when they explain a core mechanism. Exclude duplicate records, papers with insufficient technical detail, unrelated crypto-currency-only studies, and sources that do not support a research question. Document reasons for full-text exclusion. Use a PRISMA-style flow diagram only after real record counts are available; do not insert estimated counts.

C. Data Extraction and Quality Assessment

For each eligible study, extract: citation and year; application domain; architecture; storage tiers; data-placement method; decision variables; block-chain role; on-chain/off-chain design; security controls; experimental platform or dataset; baseline methods; reported metrics; limitations; and reproducibility information. Assess whether the research question, method, baselines, workload, metrics, and limitations are clearly reported. Because studies use different assumptions and experimental setups, avoid ranking methods by raw values unless the conditions are comparable.

III. BACKGROUND AND SYSTEM ARCHITECTURE

A. Storage Tiers

Local storage is directly attached to an IoT device, gateway, workstation, or institutional server. It can offer fast access and administrative control, but capacity, redundancy, and geographic resilience depend on the deployment. Edge storage places data near the source or user and can reduce network transfer for

time-sensitive tasks. Fog storage provides an intermediate layer that aggregates resources across nearby gateways or site servers. Cloud storage offers elastic capacity and centralized services, but access depends on connectivity, service configuration, and operational cost. A hybrid design can use existing institutional servers as one or more storage tiers rather than replacing them.

B. Adaptive Storage Lifecycle

A general adaptive storage lifecycle comprises data ingestion, classification, policy evaluation, placement, integrity registration, monitoring, migration, retention, and deletion. Classification may consider data size, sensitivity, access frequency, freshness, retention requirement, criticality, and expected growth. The placement engine evaluates available capacity, network conditions, access latency, and energy, reliability, and policy constraints. Monitoring then detects workload or resource changes and triggers migration or replication when the expected benefit exceeds the transfer and operational cost.

C. Block-chain and Off-Chain Storage

A hybrid architecture separates data storage from trust and audit functions. The data plane stores the payload in an appropriate storage tier. The block-chain plane records selected metadata and cryptographic commitments, and smart contracts may enforce agreed authorization or logging rules. A client can retrieve an object from the data plane and verify its hash against the registered commitment. Large payloads generally remain off-chain; storing hashes or pointers on-chain reduces ledger growth but introduces dependencies on pointer validity, storage availability, key management, and access policy. The ledger itself also requires governance, node management, and a suitable consensus mechanism.

D. Conceptual Architecture

The proposed conceptual architecture contains five cooperating modules: (1) IoT data acquisition and gateway interfacing; (2) data classification and policy management; (3) an adaptive decision engine; (4) a tiered storage pool comprising institutional/local servers, edge nodes, fog nodes, and cloud resources; and (5) a block-chain-based integrity and audit service. The decision engine selects a placement policy, while the block-chain service records verifiable events rather than serving as the primary repository for all raw data. Monitoring feeds current utilization, network delay, request frequency, and node health back to the decision engine.

IV. TAXONOMY OF ADAPTIVE STORAGE OPTIMIZATION

A. Rule-Based and Threshold-Based Placement

Rule-based methods apply predefined policies, such as keeping urgent records locally, moving cold data to cloud storage, or replicating critical objects when available capacity crosses a threshold. They are interpretable and comparatively straightforward to implement. Their weakness is that static thresholds may not adapt well to changing workloads or competing objectives, and policy conflicts can occur when security, cost, and latency requirements differ.

B. Optimization-Based Placement

Optimization methods formulate placement as a constrained decision problem. Decision variables can represent the storage tier, replication count, migration action, or retention policy for each object. Objectives may minimize a weighted combination of access latency, storage and transfer cost, energy use, and risk, subject to capacity, availability, security, and retention constraints. Multi-objective approaches are useful when no single solution minimizes every cost. Results depend strongly on the objective weights, workload model, constraints, and assumptions about migration overhead.

C. Machine Learning and Reinforcement Learning

Predictive methods can estimate access frequency, data growth, or workload peaks from historical observations. Reinforcement learning may learn placement actions from a changing environment and a reward that represents the chosen performance objectives. These methods introduce additional requirements: representative training data, safe exploration, inference overhead, explain ability, drift monitoring, and fallback behavior. An AI-based method should be compared with simple baselines and tested under workload shifts rather than evaluated only on a fixed simulation.

D. Replication, Caching, and Migration

Replication improves availability and can reduce read latency, but consumes extra capacity and increases consistency and update costs. Caching retains frequently accessed objects near users, while tier migration moves data between storage classes as access patterns change. A complete optimization policy must include the cost of moving data, temporary duplication, integrity verification, and cache invalidation. Decisions should avoid oscillation, where objects repeatedly move between tiers because a metric fluctuates around a threshold.

E. Taxonomy Summary

Approach	Decision basis	Potential benefit	Typical limitation
Rule/threshold based	Fixed policy and thresholds	Simple, explainable operation	Limited adaptation to changing workloads
Mathematical optimization	Objective function and constraints	Explicit trade-off management	Model complexity and uncertain inputs
Prediction/ML	Forecast access, size, or demand	Proactive placement	Needs representative data and retraining
Reinforcement learning	Reward from sequential decisions	Can adapt to dynamic conditions	Training cost, safety, and explain ability
Caching/replication	Access frequency and availability needs	Lower read latency and improved resilience	Extra storage and consistency overhead

TABLE I. TAXONOMY OF ADAPTIVE STORAGE APPROACHES

V. BLOCKCHAIN-ENABLED STORAGE MECHANISMS

A. On-Chain and Off-Chain Data

On-chain storage places data directly in the ledger and benefits from ledger-level replication and verification, but the cost and growth of the ledger make it unsuitable for many high-volume IoT payloads. Off-chain storage keeps payloads in conventional or distributed storage and registers a hash, identifier, timestamp, or policy reference on-chain. Hybrid storage is generally the more practical pattern for large IoT datasets, provided that data availability and pointer management are addressed.

B. Integrity, Identity, and Access Control

A cryptographic hash can help detect changes to a stored object by comparing the retrieved object's hash with a previously registered commitment. Digital signatures can associate transactions with an authorized identity, while smart contracts can record or enforce defined access rules. These mechanisms do not replace encryption or identity governance. If data are encrypted, the system must manage key distribution, revocation, recovery, and authorized sharing. Sensitive personal or healthcare information should not be exposed in immutable public records; metadata minimization and a permission design may

be appropriate depending on the threat model.

C. Consensus and Governance

Consensus determines how participating ledger nodes agree on accepted transactions. Permission deployments can limit participation to authorized institutions or departments, but they still require trust assumptions, node governance, recovery procedures, and protection against collusion or compromised credentials. The selected consensus protocol affects throughput, finality, energy consumption, and latency. A survey should report the specific protocol and configuration instead of treating block-chain as a single uniform technology.

D. Block-chain Overhead

The cost of block-chain integration includes transaction processing, ledger replication, consensus communication, cryptographic operations, storage of metadata, and administration. The integrity service should therefore be invoked at appropriate boundaries—for example, when a new object is committed, migrated, shared, or audited—rather than necessarily creating a ledger transaction for every sensor sample. Batching and periodic commitments may reduce overhead, but they also affect verification granularity and audit delay.

VI. COMPARATIVE ANALYSIS OF EXISTING RESEARCH

The literature spans several adjacent but distinct themes. Fog-placement surveys examine where services or application components should run [1], [2]; block-chain–edge surveys focus on the combined security and resource-management problem [3], [4]; and block-chain–IoT surveys organize applications and mechanisms for data management, trust, and traceability [5]. These themes overlap, but they do not necessarily solve the same optimization problem. In particular, application placement is not identical to data placement, and secure data sharing is not by itself adaptive storage optimization.

Research theme	Main focus	Useful contribution to this survey	Question that remains
Fog/edge placement	Placement of services or workloads	Decision variables and evaluation practices	How do placement decisions translate to persistent data placement and migration?
Block-chain + edge	Security, trust, resource management	Ledger roles and integration trade-offs	How can integrity overhead be included in a joint storage objective?
Block-chain + IoT	IoT security, traceability, applications	Threats and on/off-chain patterns	How should storage tiering adapt to workload and resource changes?
Cloud/fog resource allocation	Capacity, scheduling, bandwidth	Optimization constraints and resource metrics	How can cost, latency, energy, and security be jointly evaluated?

TABLE II. THEMATIC COMPARISON OF RELATED SURVEY AREAS

A. Evaluation Metrics

Metric	Definition / measurement consideration
Access latency	Time from request to successful retrieval; report average and tail latency where possible.
Storage utilization	Used capacity relative to available capacity, per tier and overall.
Total cost	Storage, transfer, transactions, compute, operations, and migration costs under stated assumptions.
Bandwidth / communication overhead	Bytes or messages transferred for ingestion, replication, migration, verification, and consensus.
Energy consumption	Energy used by devices, storage nodes, network communication, and ledger operations.
Availability / recovery	Successful access rate and recovery time under defined node or link failures.
Integrity verification	Detection of modified objects and the time/cost of verification.
Scalability	Performance as device count, data volume, transaction rate, and ledger size increase.
Security / privacy	Threat model, unauthorized access tests, key-management assumptions, and metadata exposure.

TABLE III. METRICS FOR COMPARATIVE EVALUATION

VII. RESEARCH GAPS AND CHALLENGES

A. Joint Optimization of Storage and Trust

A key research challenge is to optimize placement and migration while accounting for the cost of integrity registration, verification, access-control transactions, and ledger replication. If storage placement is optimized without these costs, the resulting design may underestimate latency, bandwidth, and resource consumption. Conversely, a block-chain-first design may impose unnecessary overhead on low-risk data. A joint policy should make the trust level and verification frequency explicit decision variables where appropriate.

B. Heterogeneous and Dynamic Resources

Local servers, edge gateways, fog nodes, and cloud resources differ in capacity, reliability, network proximity, administration, and energy profile. Their state changes as workloads arrive, nodes fail, and network links fluctuate. Adaptive systems need reliable monitoring, policy-aware placement, and migration decisions that account for the time and cost of moving data. Interoperability between storage systems and vendor-neutral policy descriptions is also important.

C. Lack of Comparable Evaluation

Reported improvements can be difficult to compare when studies use different workloads, network assumptions, baselines, hardware, block-chain configurations, or cost models. Future evaluations should define workload characteristics, data sizes, access distributions, failure scenarios, and resource limits. They should compare at least a static baseline, a non-block-chain adaptive baseline, and the proposed block-chain-assisted method to isolate the effect of adaptation and the effect of block-chain.

D. Security, Privacy, and Data Lifecycle

Immutable audit records can conflict with data minimization and deletion requirements if personal data are written directly to the ledger. Secure systems must define which metadata are recorded, how identifiers can reveal information, how encryption keys are governed, and how off-chain deletion or retention is evidenced. Threat analysis should include compromised gateways, malicious storage nodes, unauthorized users, ledger-node collusion, denial of service, and loss of keys.

E. Cost and Sustainability

Adaptive placement can reduce some storage or communication costs, but the net result depends on migration, replication, monitoring, and block-chain overhead. Claims of cost reduction must therefore be supported by a stated cost model and measured or defensible assumptions. Energy-aware optimization should include network communication and consensus operations rather than only storage-device energy.

VIII. FUTURE RESEARCH DIRECTIONS

A. AI-Assisted Predictive Placement

Forecast access frequency, growth, and workload peaks to move data before performance degrades. Use interpretable baselines and test generalization across workload changes.

B. Lightweight and Permission Ledger Designs

Investigate batching, compact commitments, efficient consensus, and permission membership where the deployment requires institutional governance. Quantify the latency and security trade-offs.

C. Self-Adaptive and Fault-Aware Storage

Use health monitoring to detect node or link failures and trigger recovery, replication, or re-placement under explicit reliability constraints.

D. Privacy-Preserving Data Sharing

Combine encryption, least-privilege authorization, minimal on-chain metadata, and controlled key management for multi-department or multi-organization deployments.

E. Reproducible Benchmarks

Develop workload profiles and benchmark scenarios that vary data size, read/write ratio, network delay, node capacity, failures, and transaction rate. Publish configurations and scripts where permitted.

F. Sustainability-Aware Optimization

Include storage, communication, migration, and consensus energy in a multi-objective formulation, and report the trade-off between efficiency and resilience.

IX. CONCEPTUAL FRAMEWORK FOR THE PH.D. RESEARCH

A. Proposed Workflow

- Collect heterogeneous IoT or institutional data through sensors, gateways, and interfacing modules.
- Classify each data object using size, sensitivity, access frequency, priority, retention, and freshness.
- Monitor available capacity, node health, network latency, transfer cost, and storage policy constraints.
- Select a target tier—local/institutional server, edge, fog, or cloud—using a rule-based or optimization-driven decision engine.
- Store the payload in the selected data store and register a hash, object identifier, timestamp, and relevant access event on a block-chain ledger.

- Verify integrity on retrieval or migration and enforce authorization before data are shared.
- Reassess placement as conditions change; migrate or replicate only when the expected benefit exceeds the movement and verification cost.
- Evaluate against baselines for latency, utilization, cost, communication overhead, energy, availability, and integrity.

B. Example Optimization Formulation

A potential formulation is to minimize a weighted objective $J = \alpha L + \beta C + \gamma E + \delta M + \varepsilon R$, where L denotes data-access latency, C denotes monetary or operational cost, E denotes energy consumption, M denotes migration and communication overhead, and R denotes a defined risk or reliability penalty. The coefficients α, β, γ, δ, and ε express the application's priorities. The decision variables may include object-to-tier assignment, replication count, migration timing, and verification policy. Constraints should represent tier capacity, access-control rules, retention, availability, and maximum latency. This is a conceptual formulation, not a validated algorithm; the final research must define each metric precisely and justify normalization and weights.

C. Experimental Plan

A staged evaluation can begin with a trace-driven simulation or controlled prototype and proceed to a small deployment using available servers and edge devices. Test cases should vary object size, access frequency, read/write ratio, storage capacity, network delay, and node failures. Compare: (i) static centralized storage; (ii) adaptive tiered storage without block-chain; and (iii) adaptive tiered storage with block-chain-based integrity and audit functions. This three-way comparison helps distinguish the value of adaptive placement from the overhead and benefits introduced by block-chain. Report repeated-run variability and avoid presenting unmeasured performance improvements as findings.

D. Expected Research Outcomes

Expected outcomes include a reproducible taxonomy and comparison of prior work; a clearly specified adaptive storage decision model; a prototype or simulation that integrates tiered data placement with verifiable integrity records; and an empirical assessment of trade-offs. Whether the approach reduces cost or latency must be established through evaluation against appropriate baselines rather than assumed in advance.

X. CONCLUSION

Adaptive storage optimization for IoT requires coordinated decisions across heterogeneous local, edge, fog, and cloud resources. Block-chain can provide a verifiable record of integrity commitments and access events, but its overhead and governance requirements must be considered as part of the system design. The survey framework presented here organizes the literature around placement strategies, hybrid on-chain/off-chain mechanisms, evaluation metrics, and open challenges. The most important research opportunity is to evaluate storage placement, migration, security, and resource efficiency together under reproducible workloads. Before submission, the proposed review protocol must be executed, the comparative tables must be populated from the screened studies, and the final claims must be tied to verified sources.

ACKNOWLEDGMENT

The authors may acknowledge institutional support, facilities, or funding here, where applicable. Confirm the exact wording with the institution before submission.

REFERENCES

1. M. M. Islam, F. Ramezani, H. Y. Lu, and M. Naderpour, "Optimal placement of applications in the fog environment: A systematic literature review," Journal of Parallel and Distributed Computing, vol. 174, pp. 46–69, 2023, doi: 10.1016/j.jpdc.2022.12.001.
2. S. Smolka and Z. Á. Mann, "Evaluation of fog application placement algorithms: A survey," Computing, vol. 104, pp. 1397–1423, 2022, doi: 10.1007/s00607-021-01031-8.
3. H. Xue, D. Chen, N. Zhang, H.-N. Dai, and K. Yu, "Integration of blockchain and edge computing in internet of things: A survey," Future Generation Computer Systems, vol. 144, pp. 307–326, 2023, doi: 10.1016/j.future.2022.10.029.
4. G. Baranwal, D. Kumar, and D. P. Vidyarthi, "Blockchain based resource allocation in cloud and distributed edge computing: A survey," Computer Communications, vol. 209, pp. 469–498, 2023, doi: 10.1016/j.comcom.2023.07.023.
5. S. K. Dwivedi et al., "Blockchain-Based Internet of Things and Industrial IoT: A Comprehensive Survey," Security and Communication Networks, vol. 2021, Article 7142048, 2021, doi: 10.1155/2021/7142048.
6. "Blockchain Applications for Internet of Things—A Survey," Internet of Things, vol. 27, Article 101254, 2024, doi: 10.1016/j.iot.2024.101254.
7. M. Zolghadri, P. Asghari, S. E. Dashti, and A. Hedayati, "Resource allocation in Fog–Cloud Environments: State of the art," Journal of Network and Computer Applications, vol. 227, Article 103891, 2024, doi: 10.1016/j.jnca.2024.103891.
8. "A systematic review of the purposes of Blockchain and fog computing integration: classification and open issues," Journal of Cloud Computing, vol. 11, Article 80, 2022, doi: 10.1186/s13677-022-00353-y.
9. R. Kumar and N. Agrawal, "Analysis of multi-dimensional Industrial IoT (IIoT) data in Edge–Fog–Cloud based architectural frameworks: A survey on current state and research challenges," Journal of Industrial Information Integration, 2023, doi: 10.1016/j.jii.2023.100504.
10. A. Hemmati, N. Khaledian, and A. M. Rahmani, "Toward a unified computing paradigm: a survey and roadmap for data management in integrated IoT, fog, and cloud services," Peer-to-Peer Networking and Applications, 2026, doi: 10.1007/s12083-025-02183-6.