A Scalable and Privacy-Preserving Hybrid Blockchain Architecture for Secure Healthcare Data Management

Sanjida Sharmin[1], Mohammad Shamsul Arefin[2], Pranab Kumar Dhar[3], Zinnia Sultana[4], Sultana Akter[5]
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology, Chattogram-4349, Bangladesh[1,2,3]
Department of Computer Science and Engineering,
International Islamic University Chittagong, Chattogram-4318, Bangladesh[1,4]
Department of Business Administration,
International Islamic University Chittagong, Chattogram-4318, Bangladesh[5]

Abstract—**The protection of sensitive medical information has become a critical concern in modern digital healthcare. This study introduces a Hybrid Architecture that ensures secure and reliable healthcare data management through the integration of blockchain technology with off-chain and on-chain mechanisms. Patient records are encrypted using AES-256-GCM, stored in the InterPlanetary File System (IPFS), and verified using Merkle Tree structures, with only the root values anchored on Ethereum smart contracts. This design guarantees data security and integrity while achieving significant gas optimization by reducing on-chain storage costs. Experimental evaluation demonstrates that the proposed system achieves high scalability, efficient transaction processing, and strong resistance to tampering, ensuring confidentiality and auditability. By combining blockchain, cryptographic techniques, and distributed storage, the framework addresses pressing challenges of security, privacy, and trust in healthcare ecosystems. The results highlight the potential of Hybrid Architecture models to deliver a cost-effective, privacy-preserving, and scalable solution for next-generation Healthcare Data Security.**

Keywords*—*Blockchain; healthcare; AES-256-GCM; merkle tree; IPFS; data security; scalability; gas optimization; ethereum; hybrid architecture

I. Introduction

The digitization of healthcare data has led to increased dependency on electronic health record (EHR) systems and cloud-based platforms for storage and access. However, this growing reliance introduces serious challenges related to data privacy, integrity, interoperability, and unauthorized access. Traditional centralized healthcare data management architectures are susceptible to single points of failure, cyberattacks, and restricted transparency. These limitations necessitate novel architectural paradigms that can ensure secure, verifiable, and patient-centric data handling.

Blockchain technology has emerged as a transformative solution to address these challenges by enabling decentralized and tamper-resistant systems. With its core attributes—immutability, transparency, and consensus-driven validation—blockchain can reinforce data trustworthiness and facilitate secure data sharing across distributed healthcare providers. A growing body of research has explored the integration of blockchain in healthcare, recognizing its potential to decentralize control, enhance interoperability, and safeguard sensitive clinical data [1], [2] Several works provide a foundation for understanding the structural and operational principles of blockchain. In [3], the authors presents a comprehensive overview of blockchain architecture, consensus algorithms, and anticipated evolution. Hölbl et al. [1] systematically reviewed the applicability of blockchain in healthcare, identifying both opportunities and technical limitations. Ismail et al. [4] proposed a lightweight blockchain framework tailored for healthcare environments, focusing on scalability and performance. Kakei et al. [5] explored distributed authentication infrastructures based on Hyperledger Fabric, showing potential for cross-institutional trust establishment.

Moreover, performance analysis of blockchain frameworks such as Hyperledger Fabric have shown that private blockchains can offer high throughput and low latency for medical data exchanges [6]. Khan et al. [7] discussed blockchain's synergy with the Internet of Things (IoT) and smart grid systems in healthcare. Rathi et al. [8] demonstrated the value of blockchain-enabled orchestrators for multi-domain edge computing, enabling secure data transactions at the network edge. Edited volumes such as that by Namasudra and Deka [9] further provide insights into real-world deployments, regulatory constraints, and future trends in blockchain-driven healthcare ecosystems. Despite this progress, challenges persist in effectively linking on-chain and off-chain data management. Storing complete medical files on-chain is impractical due to gas fees and blockchain size constraints. Consequently, hybrid models integrating blockchain with distributed file systems like the InterPlanetary File System (IPFS) are gaining attention [10]. These architectures allow for off-chain storage of large medical files while maintaining their integrity through on-chain hashes.

However, existing blockchain-based healthcare data management systems continue to face critical challenges, including high gas costs from on-chain storage, limited scalability when handling large datasets, and insufficient privacy-preserving mechanisms. These limitations restrict their practical adoption in real-world healthcare environments. To address these gaps, this study proposes a hybrid blockchain architecture that combines AES-256-GCM encryption, IPFS off-chain storage, and Merkle root anchoring on Ethereum, thereby ensuring strong data confidentiality, verifiable integrity, and significant cost reduction while maintaining scalability.

This study proposes a hybrid blockchain-based framework that integrates Ethereum smart contracts and IPFS to create a secure, scalable, and tamper-resistant infrastructure for healthcare data management. Our main contributions are summarized as follows:

- A hybrid blockchain framework is proposed, where only Merkle roots of encrypted file batches are stored on-chain, achieving over 99% gas cost reduction compared to traditional methods.
- Patient records are encrypted using AES-256-GCM, ensuring data confidentiality, integrity, and secure off-chain storage.
- The system supports batch-wise Merkle tree construction and lightweight proof mechanisms, enabling scalable, high-throughput, and verifiable healthcare data management.

The remainder of this study is organized as follows: Section II reviews related work on blockchain-based healthcare data management and highlights existing challenges. Section III presents the proposed Hybrid Architecture, detailing the use of AES-256-GCM encryption, IPFS off-chain storage, Merkle Tree verification, and Ethereum smart contracts. Section IV discusses the experimental setup and performance evaluation, focusing on data security, scalability, and gas optimization. It provides a detailed discussion of the results, insights into parameter influence, and implications for practical healthcare deployment. Finally, Section V concludes the study by summarizing the contributions and outlining directions for future research.

II. Related Work

Blockchain technology has gained significant attention in healthcare for its potential to provide secure, tamper-proof data management. However, the inherent limitations in scalability pose ongoing challenges, particularly in high-volume environments, such as electronic health record (EHR) systems. Recent studies have proposed architectural enhancements and hybrid models to address these limitations.

Sadath et al. [11] introduced a hierarchical blockchain architecture tailored to healthcare, utilizing both local and global chains to minimize network congestion and improve transaction throughput. This model supports distributed data handling while preserving integrity, making it a viable solution for large-scale deployments. Similarly, Ali et al. [12] proposed a hybrid blockchain system augmented with deep learning techniques to prioritize transactions dynamically. Their model achieved improved responsiveness and reduced latency under high data loads without compromising security.

Mazlan et al. [13] provided a systematic review highlighting key scalability barriers, including throughput, latency, and interoperability. The authors discussed theoretical mitigation strategies such as Layer-2 solutions and sharding, but noted a lack of real-world implementations. In contrast, Jayabalan and Jeyanthi [14] presented a practical approach using IPFS for off-chain storage combined with on-chain metadata management, demonstrating measurable improvements in speed and scalability.

Security-focused frameworks have also shown indirect scalability benefits. Shamshad et al. [15] developed a blockchain-based EHR system with embedded access control via smart contracts. This reduced computational overhead on-chain and facilitated secure sharing. Further, Mazhar et al. [16] explored the convergence of blockchain, generative AI, and IoT in healthcare. Their study emphasized enhanced privacy, patient personalization, and interoperability, though challenges related to resource demands and decentralized storage remain.

Bathula et al. [17] conducted a comprehensive review of over 400 studies and concluded that blockchain–AI integration holds substantial promise for strengthening EHR reliability, supporting diagnostics, and managing pandemics. Collectively, the literature affirms that scalable blockchain solutions—especially those involving AI integration and off-chain data architectures—are central to building resilient, efficient, and secure healthcare infrastructures.

In the study by Taherdoost [18], the author explores the integration of blockchain technology within the healthcare sector, emphasizing its potential to enhance data privacy and security. The study provides a comprehensive overview of blockchain applications such as secure electronic health records (EHRs), patient consent management, and pharmaceutical supply chain tracking. It highlights key challenges including scalability, regulatory compliance, and interoperability with existing healthcare systems. Furthermore, the study discusses future research directions focused on improving the efficiency, adaptability, and ethical implementation of blockchain in healthcare. This work serves as a foundational reference for understanding both the opportunities and limitations of blockchain adoption in sensitive data environments like healthcare.

Kasyapa and Vanmathi [19] presents a thorough examination of blockchain integration in the healthcare domain, focusing on practical use cases, performance-related challenges, and corresponding mitigation strategies. Their investigation spans a variety of healthcare applications, including patient data management, clinical trials, and remote monitoring systems, where blockchain's decentralization and immutability offer clear benefits. The authors delve into performance bottlenecks such as latency, throughput, and energy consumption, and propose solutions involving lightweight consensus mechanisms and hybrid architectures. Their work not only showcases the transformative potential of blockchain in improving healthcare infrastructure but also provides critical insights into addressing technical and operational barriers for real-world deployment.

A comparative critique of notable blockchain-based healthcare frameworks is summarized in Table I. The table highlights the strengths and limitations of prior studies and clarifies how the proposed method advances beyond these works by integrating end-to-end encryption, scalable off-chain storage, and gas-efficient Merkle root anchoring.

III. Methodology

The proposed architecture for secure and scalable healthcare data management is structured into four sequential phases:

TABLE I. Comparative Analysis of Healthcare Blockchain Frameworks and Relation to the Proposed Method

Framework	Strengths	Weaknesses / Assumptions	Relation to Proposed Method
Sadath et al.[11]	Hierarchical chain reduces congestion; improves throughput.	Lacks end-to-end encryption; assumes scalability alone ensures privacy.	Employs AES-256-GCM encryption to ensure confidentiality while retaining scalability.
Matani et al.[20]	Batch-root commitments improve auditability and verification efficiency.	Limited scalability on large datasets; no off-chain optimization.	Uses batch-wise Merkle root anchoring with IPFS off-chain storage for scalable verification.
HealthBlock[21]	Strong decentralization and patient-centric control.	On-chain metadata storage increases gas cost and causes state bloat.	Achieves about 99.85% gas reduction by anchoring only Merkle roots on-chain.
MEDACCESSX[22]	Modular contracts improve flexibility for access control.	Complex contracts; confidentiality not guaranteed system-wide.	Maintains lightweight contract logic with strong encryption for end-to-end security.
AdaSec-Health[23]	Shard-level tuning enhances scalability.	Relies on tuning; encryption is partial or unclear.	Provides scalable hybrid storage with explicit AES-256-GCM protection.

Preprocessing, On-Chain storage, Off-Chain storage, and Verification & Decryption. Each component contributes to ensuring data confidentiality, integrity, and blockchain efficiency.The methodology involves the following key stages, as depicted in Fig. 1.

A. Dataset Preprocessing

The first stage of the proposed system involves preprocessing a structured healthcare dataset to prepare it for secure encryption, hashing, and metadata extraction. The dataset, typically stored in CSV format, contains fields such as patient_uid, PMID, file_path, title, patient, age, gender, and relevant_articles.

Table II presents representative entries from the raw dataset before preprocessing. The detailed preprocessing steps are outlined below.

1) Missing value handling: Records with null or incomplete values in critical fields (`patient_uid`, `age`, `gender`) are removed to ensure consistency and integrity throughout the encryption and verification pipeline.

2) Duplicate removal: Duplicate rows are identified and removed based on a composite key consisting of `patient_uid` and `PMID`, which uniquely represent each patient case and publication reference.

3) Age field normalization: The `age` field is often stored as a nested structure (e.g., `[[60.0, "year"]]`). A regular expression parser is used to extract the numeric value (e.g., 60) and convert it to an integer format for uniformity.

4) JSON conversion: Each row of the cleaned dataset is serialized into a standalone JSON object. These objects serve as the input to the encryption module and include metadata required for downstream file hashing and Merkle proof construction.

5) File organization: Each JSON record is saved as an individual file and assigned a filename based on its `patient_uid`. These files are stored in a staging directory for encryption and off-chain upload.

This preprocessing pipeline ensures the dataset is complete, uniquely identifiable, and structurally standardized before it enters the AES-256 encryption and Merkle tree computation stages.

B. On-Chain Processing

The on-chain component of the proposed system is responsible for storing cryptographic summaries (Merkle roots) of encrypted patient record batches to ensure tamperproof verification. The process is carried out using smart contracts deployed on the Ethereum Sepolia testnet and consists of the following steps:

1) Network connection: The system connects to the Ethereum Sepolia testnet using Infura as the remote Ethereum node provider. The Web3.py library is employed to facilitate interaction between the local Python environment and the blockchain network.

2) Smart contract integration: A pre-deployed smart contract, written in Solidity, is used to register and manage Merkle roots corresponding to batches of encrypted patient records. The contract contains methods for both storing and retrieving root values associated with batch identifiers. The smart contract behavior is formalized in Algorithm 1.

Algorithm 1 Merkle Root Commitment Smart Contract

```
1:  State Variables:
2:      $owner$ : Address of the contract deployer
3:      $roots$ : Mapping from batch ID to Merkle root

4:  Functions:
5:  function CONSTRUCTOR
6:      $owner \leftarrow$ msg.sender
7:  end function
8:  function STOREMERKLEROOT($bID$, $root$)
9:      if msg.sender $\neq owner$ then
10:         return "Unauthorized"
11:     end if
12:     if $roots[bID] \neq$ null then
13:         return "Already submitted"
14:     end if
15:     $roots[bID] \leftarrow root$
16:     Emit MerkleRootStored($bID$, $root$, block.timestamp)
17: end function
18: function GETMERKLEROOT($bID$)
19:     return $roots[bID]$
20: end function
```

3) Merkle root storage: For each batch of 1,000 encrypted files, a Merkle root is computed from their SHA-256 hashes.

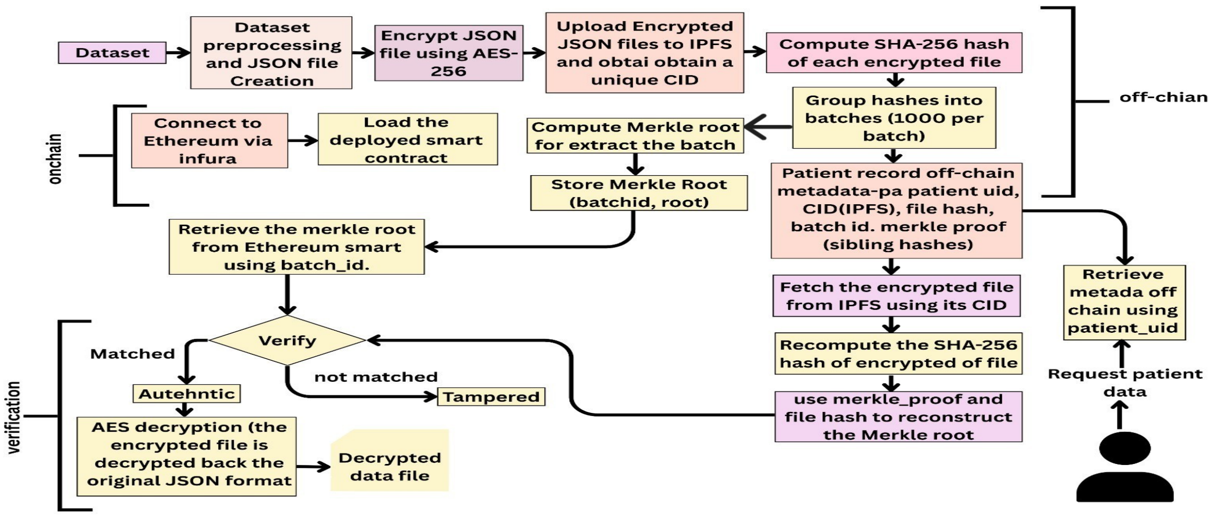
Dataset
Dataset preprocessing and JSON file Creation
Encrypt JSON file using AES-256
Upload Encrypted JSON files to IPFS and obtai obtain a unique CID
Compute SHA-256 hash of each encrypted file
Group hashes into batches (1000 per batch)
Compute Merkle root for extract the batch
Store Merkle Root (batchid, root)
Patient record off-chain metadata-pa patient uid, CID(IPFS), file hash, batch id. merkle proof (sibling hashes)
off-chian
onchain
Connect to Ethereum via infura
Load the deployed smart contract
Retrieve the merkle root from Ethereum smart using batch_id.
Fetch the encrypted file from IPFS using its CID
Recompute the SHA-256 hash of encrypted of file
use merkle_proof and file hash to reconstruct the Merkle root
Retrieve metada off chain using patient_uid
Request patient data
verification
Verify
Matched
not matched
Autehntic
Tampered
AES decryption (the encrypted file is decrypted back the original JSON format
Decrypted data file

Fig. 1. System architecture of the proposed hybrid blockchain system.

TABLE II. Sample Instances from the Raw PMC-Patients Dataset

id	patient_uid	PMID	file_path	title	patient	age	gender	relevant_articles
0	7665777-1	33492400	comm/PMC...	Early Physic...	This 60-year	[[60.0, 'year']]	M	{'32320506':1, ...}
1	7665777-2	33492400	comm/PMC...	Early Physic...	A 39-year-c	[[39.0, 'year']]	M	{'32320506':1, ...}
2	7665777-3	33492400	comm/PMC...	Early Physic...	One week a	[[57.0, 'year']]	M	{'32320506':1, ...}
3	7665777-4	33492400	comm/PMC...	Early Physic...	This 69-year	[[69.0, 'year']]	M	{'32320506':1, ...}
4	7665777-5	33492400	comm/PMC...	Early Physic...	This 57-year	[[57.0, 'year']]	M	{'32320506':1, ...}

The root, along with a unique batch identifier, is then submitted to the blockchain using the smart contract function `storeMerkleRoot(batchId, root)`. This ensures immutability, auditability, and decentralized verification of the batch integrity.

C. Off-Chain Storage and IPFS Upload

The off-chain component handles encryption, decentralized file storage, hash computation, and metadata management. This process is organized into six key steps:

1) Encrypt patient JSON files: Each patient's data is serialized into JSON format and encrypted using AES-256-GCM [24], [25]. This encryption ensures both data confidentiality and integrity. The full procedure of encryption is presented in Algorithm 2.

2) Upload encrypted files to IPFS: The encrypted files are uploaded to the InterPlanetary File System (IPFS), a distributed storage network. The returned content identifier (CID) for each file is saved for future reference and verification.

3) Compute SHA-256 hash: The SHA-256 hash of each encrypted file is computed. These hashes act as the leaf nodes of the Merkle tree and serve as unique digital fingerprints of the file content.

4) Batch file hashes: File hashes are grouped into batches of 1,000 files per batch. Each batch is prepared independently for Merkle tree construction.

5) Build Merkle tree: For each batch, a Merkle tree is constructed using the SHA-256 hashes. The root of each tree is computed and serves as a compact, verifiable summary of the batch contents. The batched Merkle construction is detailed in Algorithm 3.

6) Store off-chain metadata: For each file, corresponding metadata including patient UID, CID, file hash, batch ID, and Merkle proof is compiled and stored in a structured CSV file. This metadata supports efficient file retrieval and integrity verification. To verify the integrity of any file, its SHA-256 hash is recomputed and used along with the Merkle proof to reconstruct the root hash. This reconstructed root is compared with the on-chain Merkle root associated with the batch. If the verification succeeds, the file is decrypted using the AES-256 key. This ensures that only authenticated, untampered files are accessed and utilized, thereby preserving both security and data reliability.The procedure of decryption is presented in Algorithm 4.

Algorithm 2 AES-256-GCM Encryption of Patient Records

```
Require: Patient record $D$; 256-bit AES key $K$; 12-byte nonce $N$
Ensure: Encrypted blob $E$ containing $\{N,\ ciphertext,\ tag\}$
1: $plaintext \leftarrow \textsc{Serialize}(D)$
2: $cipher \leftarrow \textsc{AesGcmInit}(K, N)$
3: $(ciphertext, tag) \leftarrow \textsc{Encrypt}(cipher, plaintext)$
4: $E \leftarrow N \,\|\, ciphertext \,\|\, tag$
5: return $E$
```

Algorithm 3 Merkle Root Construction from Encrypted Files

```
Require: Encrypted files $F = \{f_1, \dots, f_n\}$; batch size $B$
Ensure: Merkle roots $R = \{r_1, \dots, r_k\}$
 1: $H \leftarrow []$                                    ▷ List of SHA-256 hashes
 2: for each $f_i \in F$ do
 3:     $h_i \leftarrow \textsc{Sha256}(f_i)$
 4:     APPEND $h_i$ to $H$
 5: end for
 6: Partition $H$ into batches $\{B_j\}$ of size $B$;   $R \leftarrow []$
 7: for each batch $B_j$ do
 8:     while $|B_j|$ is not a power of 2 do
 9:         APPEND last$(B_j)$ to $B_j$                  ▷ Pad with last hash
10:     end while
11:     while $|B_j| > 1$ do
12:         $L \leftarrow []$
13:         for $k \leftarrow 1$ to $|B_j|$ step 2 do
14:             $p \leftarrow \textsc{Sha256}(B_j[k] \,\|\, B_j[k{+}1])$
15:             APPEND $p$ to $L$
16:         end for
17:         $B_j \leftarrow L$
18:     end while
19:     APPEND $B_j[1]$ to $R$                           ▷ Merkle root for this batch
20: end forreturn $R$
```

Algorithm 4 AES-256-GCM Decryption of Patient Records

```
Require: Encrypted blob $E = N \,\|\, C \,\|\, T$; 256-bit key $K$; (optional)
    AAD $A$
Ensure: Decrypted patient record $D$
 1: if $|E| < 28$ then
 2:     return Error: malformed input
 3: end if
 4: $N \leftarrow E[0:12]$                                ▷ 12-byte nonce
 5: $T \leftarrow E[-16:]$                                ▷ 16-byte tag
 6: $C \leftarrow E[12:-16]$                              ▷ ciphertext
 7: $cipher \leftarrow \textsc{AesGcmInit}(K, N)$
 8: if $A$ provided then
 9:     $\textsc{AesGcmUpdateAAD}(cipher, A)$
10: end if
11: $plaintext \leftarrow \textsc{DecryptAndVerify}(cipher, C, T)$
12: if verification fails then
13:     return Error: authentication failed
14: end if
15: $D \leftarrow \textsc{ParseJson}(plaintext)$
16: return $D$
```

IV. Results and Evaluation

This section presents the experimental setup, evaluation metrics, and performance analysis of the proposed hybrid blockchain architecture. The system is evaluated for gas efficiency, latency, throughput, scalability, and data security using real-world datasets.

A. Experimental Setup

The proposed framework was implemented using Python 3.x and deployed in Google Colab for off-chain processing. SHA-256 hashing and AES-256-GCM encryption were performed using the `hashlib` and `cryptography` libraries, respectively. IPFS Desktop and `go-ipfs` were used for decentralized file storage, while Web3.py and Infura were employed to interact with smart contracts on the Ethereum Sepolia testnet. Smart contracts were developed using Solidity and deployed via the Hardhat framework.

Experiments were conducted on a computing system equipped with an Intel(R) Core(TM) i7-4510U CPU operating at 2.00 GHz (with a turbo frequency of 2.60 GHz) and 6 GB of RAM, running on Windows 10 Pro (64-bit). The proposed hybrid blockchain architecture was implemented and evaluated using the Ethereum Sepolia testnet, accessed via the Infura gateway. Encryption of patient records was performed using the AES-256-GCM algorithm to ensure both data confidentiality and integrity. The experimental dataset consisted of 39,936 encrypted patient records in JSON format, which were processed and managed across the off-chain and on-chain components of the system.

B. Evaluation Metrics

The performance of the proposed system is evaluated using five core metrics: gas cost, latency, throughput, scalability, and data security. Two of these metrics—latency and gas consumption—are further analyzed mathematically to validate the framework's efficiency.

1) Latency modeling: The total system latency (L_{total}) is the cumulative delay incurred during data preprocessing, storage, and verification phases. The overall system latency (L_{total}) can be modeled as the sum of initiation, confirmation, and off-chain delays, as formulated in Eq. (1):

$$\begin{aligned} L_{\text{total}} = {} & (t_{\text{initiation}} + t_{\text{confirmation}}) \\ & + (t_{\text{IPFS\_upload}} + t_{\text{IPFS\_retrieval}}) \\ & + (t_{\text{hash\_compute}} + t_{\text{hash\_compare}}) \end{aligned} \tag{1}$$

where, $t_{\text{initiation}}$ and $t_{\text{confirmation}}$ represent the time taken for transaction submission and blockchain block confirmation, respectively. $t_{\text{IPFS\_upload}}$ and $t_{\text{IPFS\_retrieval}}$ correspond to the delays associated with decentralized storage operations in IPFS. Finally, $t_{\text{hash\_compute}}$ and $t_{\text{hash\_compare}}$ denote the time required for computing file hashes and verifying file integrity.

Empirical results show that blockchain transaction latency ($L_{\text{on-chain}}$) averages around 14.3 seconds, while IPFS upload and retrieval ($L_{\text{off-chain}}$) contribute an additional 2 to 4 seconds. As shown in Eq. (2), the overall system latency is modeled as the sum of on-chain and off-chain delays.

$$L_{\text{total}} \approx L_{\text{on-chain}} + L_{\text{off-chain}} \tag{2}$$

Thus, the total latency for the hybrid blockchain–IPFS framework is approximately in the range of 16 to 18 seconds, validating the framework's suitability for non-real-time healthcare applications.

2) Throughput modeling: Throughput (T_{system}) is defined as the rate at which encrypted patient records are processed, verified, and committed within the system. It is a critical performance metric that reflects the system's efficiency in handling large-scale healthcare datasets. As expressed in Eq. (3), the system throughput is defined as the ratio of the total number of records to the overall system latency.

$$T_{\text{system}} = \frac{N_{\text{records}}}{L_{\text{total}}} \tag{3}$$

where, N_{records} is the number of encrypted records in a given batch and L_{total} is the total latency (as defined in Equation 1) required to process and verify the batch.

3) Gas cost matrices: In blockchain-based systems, particularly those utilizing Ethereum smart contracts, the transaction execution cost is quantified through gas consumption. As shown in Eq. (4), the gas cost in Ethereum is determined by multiplying the gas used by the gas price.

$$\text{Gas Cost (ETH)} = \text{Gas Used} \times \text{Gas Price (ETH)} \tag{4}$$

where, Gas Used denotes the actual number of gas units consumed during transaction execution, and Gas Price (ETH) is the market price of one unit of gas, typically denominated in Gwei (where $1\,\text{Gwei} = 10^{-9}\,ETH$).As formulated in Eq. (5), the gas cost in USD is obtained by multiplying the gas cost in ETH with the current ETH price in USD.

$$\begin{aligned}\text{Gas Cost (USD)} = &(\text{Gas Used} \times \text{Gas Price (ETH)}) \\ &\times \text{ETH Price (USD)}\end{aligned} \tag{5}$$

C. Gas Cost Efficiency

In Ethereum-based smart contract systems, the cost of executing a transaction is determined by gas consumption. The proposed method significantly reduces gas costs by batching encrypted file hashes into Merkle trees and storing only the Merkle root on-chain. As defined in Eq. (6), the total gas consumption is obtained by multiplying the number of batches with the gas required per root.

$$G_{\text{total}} = N_{\text{batches}} \times G_{\text{per\_root}} \tag{6}$$

where,

- $N_{\text{batches}} = 40$ is the number of file batches.
- $G_{\text{per\_root}} = 30,000$ gas is the average gas consumed to store a single Merkle root on-chain.

Substituting into Eq. (6), the total gas cost becomes Eq. (7):

$$G_{\text{total}} = 40 \times 30{,}000 = 1{,}200{,}000 \text{ gas} \tag{7}$$

TAs shown in Eq. (8), the total gas consumption for processing all files individually is computed as:

$$\begin{aligned}G_{\text{individual}} &= N_{\text{files}} \times G_{\text{per\_hash}} \\ &= 39,936 \times 20,000 \\ &= 798,720,000\,\text{gas}\end{aligned} \tag{8}$$

Using this value, the gas savings ratio is then derived in Eq. (9) as:

$$\eta_{\text{gas}} = 1 - \frac{G_{\text{total}}}{G_{\text{individual}}} = 1 - \frac{1,200,000}{798,720,000} \approx 0.9985 \tag{9}$$

This demonstrates that the proposed system achieves a remarkable gas savings ratio of approximately 0.9985, which corresponds to a reduction of nearly 99.85% in gas cost compared to executing all transactions, individually.

D. Gas Cost Analysis in Ethereum

The following parameters:

- Total gas used to store 40 Merkle roots: G_{total} = 1,200,000 gas
- Gas price: 30 Gwei = 30×10^{-9} ETH
- ETH/USD exchange rate: $3,000

The total ETH cost is computed as:

$$\begin{aligned}\text{Cost}_{\text{ETH}} &= 1{,}200{,}000 \times 30 \times 10^{-9} \\ &= 0.036\,\text{ETH}\end{aligned} \tag{10}$$

Then, converting to USD:

$$\text{Cost}_{\text{USD}} = 0.036 \times 3000 = \$108 \tag{11}$$

The total ETH cost is computed, as shown in Eq. (10), and then converted into USD using Eq. (11).

E. Latency Analysis

Fig. 2 illustrates the latency distribution across 40 batches of encrypted patient records during off-chain processing and on-chain Merkle root submission. Each batch consists of approximately 1,000 encrypted files. The measured latency includes file encryption, IPFS upload, SHA-256 hash computation, Merkle tree construction, and Ethereum transaction confirmation.

The latency across batches ranges from approximately 11.4 seconds to 28.1 seconds, with most batches falling within the 18 to 26 second interval. Variations are primarily attributed to network fluctuation during IPFS storage and Ethereum block confirmation delays. Notably, the final batch exhibits the lowest latency due to a reduced number of files.

This consistent and bounded latency behavior demonstrates the system's ability to maintain predictable performance while processing large-scale healthcare datasets, making it suitable for near real-time and secure health data workflows.

F. Throughput Performance

Throughput is defined as the number of encrypted patient records processed per second, including encryption, IPFS upload, Merkle tree construction, and Merkle root submission. Fig. 3 illustrates the throughput achieved across 40 batches of encrypted files, each consisting of approximately 1,000 records.

As shown, throughput ranges from approximately 36 to 74 files per second, depending on batch size, system load, and network conditions. Batches with fewer files (e.g., Batch 40) naturally exhibit higher throughput due to reduced processing

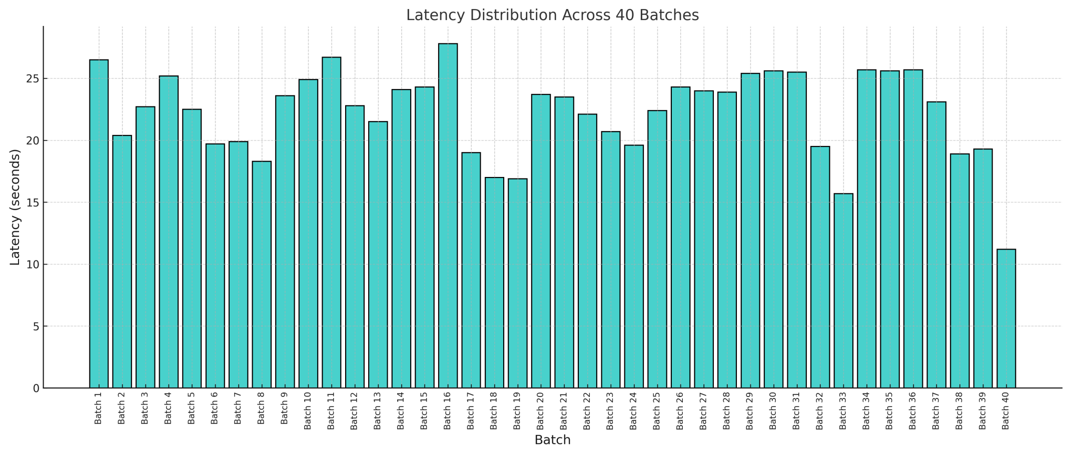

Fig. 2. Latency distribution for 40 batches of encrypted patient records.

time. The consistent throughput across most batches reflects the stability of the proposed system under typical workloads and validates its ability to handle large volumes of healthcare data efficiently.

G. Off-Chain Scalability Analysis

To evaluate the system's scalability in terms of storage distribution, the off-chain encrypted files were grouped into 40 batches of approximately 1,000 files each. Fig. 4 illustrates the estimated storage size per batch in megabytes (MB).

As shown, most batches maintain a consistent size of approximately 194 to 195 MB, demonstrating balanced storage allocation and predictable performance. The final batch, containing fewer files, exhibits a slightly lower size, validating the linear scaling behavior of the proposed system. This confirms that batch-based off-chain file management is both scalable and storage-efficient for large-scale healthcare datasets.

H. Security Assurance

AES-256-GCM encryption provided confidentiality and integrity through its built-in authentication tag. In conjunction with Merkle proof-based integrity verification, the system offers a secure mechanism for handling sensitive healthcare data in decentralized environments.

I. Comparative Analysis with Prior Works

Table III presents a comparative analysis of the proposed hybrid blockchain system against three recent healthcare blockchain frameworks: HealthBlock [21], Sadath et al. [11], and Matani et al. [20]. The comparison focuses on ten key criteria, including storage architecture, on-chain data strategy, encryption, scalability, and verification methodology.

Unlike existing models that either store full metadata on-chain or rely on non-scalable storage mechanisms, the proposed system leverages off-chain IPFS storage and stores only 40 Merkle roots on-chain—one for each batch of 1,000 encrypted patient records. This significantly reduces on-chain data bloat while preserving verifiability through Merkle proofs. The use of AES-256 encryption further ensures end-to-end confidentiality and integrity, which is either partially implemented or entirely missing in the compared systems.

In terms of scalability, the proposed framework supports high-volume batch uploads and demonstrates high data scalability. It adopts a public Ethereum testnet (Sepolia), offering better decentralization and accessibility compared to permissioned or hybrid networks like Fabric and sharded blockchains. Furthermore, it incorporates a robust security model combining AES encryption, Merkle proof, and access control lists (ACLs), making it well-suited for large-scale, privacy-sensitive healthcare environments.

J. Comparative Results Across Datasets

The differences observed in comparative results across datasets can be attributed to the architectural limitations of existing frameworks. Approaches such as Sadath et al.[11] and Matani et al.[20] perform adequately on smaller datasets but exhibit reduced scalability at higher volumes due to limited batching and lack of off-chain storage. HealthBlock emphasizes decentralization but incurs significant gas costs from on-chain metadata storage, which becomes inefficient as data size grows. In contrast, the proposed method consistently maintains scalability and efficiency across datasets by integrating AES-256-GCM encryption, IPFS off-chain storage, and batch-wise Merkle root anchoring. These design choices make the algorithms particularly well-suited for structured and semi-structured healthcare data, including electronic health records, demographic information, prescriptions, and transaction logs. For unstructured or multimedia data such as medical imaging or continuous IoT sensor streams, additional preprocessing into metadata or compact representations is necessary, as direct storage may increase retrieval latency.

1) Latency comparison with existing methods: Fig. 5 presents a comparative latency analysis between the proposed method and three existing blockchain-based healthcare frameworks: Matani et al. [20], Sadath et al. [11], and HealthBlock [21]. The x-axis represents the number of transactions, while the y-axis shows latency in seconds.

The proposed method consistently demonstrates the lowest latency across all transaction scales (1,000 to 6,000), ranging

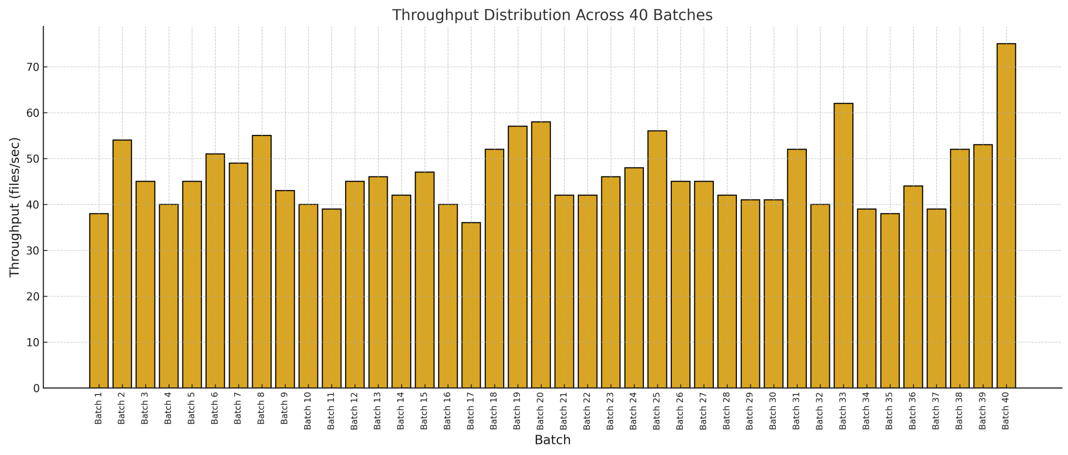

Fig. 3. Throughput distribution across 40 batches, expressed in files per second.

TABLE III. Comparative Analysis of the Proposed Hybrid Blockchain System Against Prior Works

Criteria	**Proposed System**	**HealthBlock [21]**	**Sadath et al. [11]**	**Matani et al. [20]**
Storage	IPFS Remote (∼180 MB/batch)	OrbitDB + IPFS	CouchDB	Shard-distributed
On-Chain Data	40 Merkle Roots	Per-file metadata	Batch root	Per-level TX hashes
Hashing	SHA-256	SHA-256	SHA-256	SHA-256
Merkle Tree	Yes (40 roots)	No	No	Used in scoring phase
Encryption	AES-256	Partial (OrbitDB/IPFS)	No encryption	No encryption
Batch Uploads	1,000 per batch	No	Yes	Implied via sharding
Blockchain Type	Public (Ethereum Sepolia)	Private (Fabric)	Private (Fabric)	Hybrid Sharded
Verification Type	Merkle proof	Log traceability	Proof of computation	Hash-linked consensus
Data Scalability	High	Low	Medium	Medium
Security Model	AES + Merkle + ACL	N/A	Metadata only	Hierarchical + scoring

TABLE IV. Comparative Analysis of Gas Cost and Architecture: Proposed System Against Prior Work

Criteria	**Proposed System**	**MEDACCESSX [22]**	**AdaSec-Health [23]**
Blockchain Type	Public (Ethereum Sepolia)	Private Ethereum Testnet	Public (Ethereum EIP-1559)
Storage Strategy	Off-chain IPFS + Merkle Root anchoring	On-chain smart contracts for access + data metadata	IPFS + ECOA-optimized shard blockchain
On-Chain Data	40 Merkle Roots for 39,936 files	User, access, data, and policy contracts (UMC, ACC, etc.)	Dynamic block/meta parameters via ECOA
Gas Optimization Method	Merkle batching, minimal on-chain state	Modular contracts but fixed logic	Shard tuning: validation degree, block size, interval
Total Gas Used	1.2M gas for 39,936 files	3.1M gas for deployment (all contracts)	Avg. marginal cost = 0.47 gas units per transaction
Gas Cost in USD	$108 @ 30 Gwei, $3K/ETH	$154.65 (same rate)	Not directly priced, implied savings
Gas Savings (%)	99.85% vs. per-file storage	Not explicitly reported	Reported via performance gains under ECOA
Smart Contract Complexity	Minimal (root commit/retrieve only)	High (RBAC, ABAC logic, events)	Medium to high (governance, consensus, voting)
Batch Handling	1,000 per batch	Per access/role	Per 4-block shard optimization
Security Model	AES-256-GCM + Merkle + ACL	Access logs on-chain + ABAC/RBAC	ECOA-based scoring + CP-ABE + reputation
Scalability Focus	Gas-efficient IPFS and Merkle design	Functional control granularity	Adaptive shard configuration for forks/delay

from 1.8 to 3.2 seconds. This efficiency is attributed to the use of Merkle root batching, AES-256-GCM encryption, and lightweight on-chain commitment. In contrast, the methods by Matani et al.[20] and Sadath et al.[11] exhibit significantly higher and more variable latency, primarily due to per-transaction or per-batch on-chain data handling without

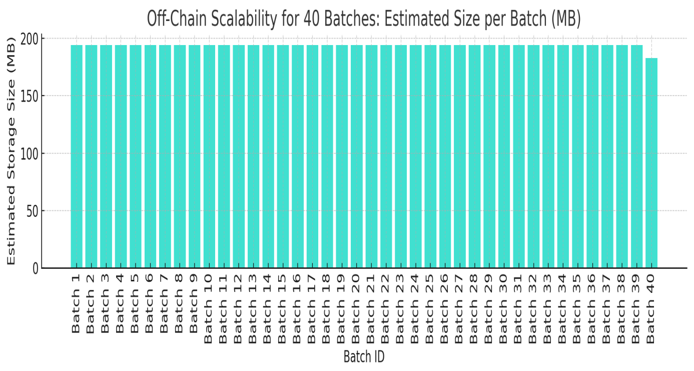

Fig. 4. Off-chain scalability analysis: estimated file size distribution across 40 batches.

optimized batching. HealthBlock performs moderately but lacks Merkle-based verification and full encryption, leading to comparatively higher overhead.

This result confirms that the proposed system is well-suited for secure healthcare applications where low-latency processing is essential for real-time or near-real-time interactions.

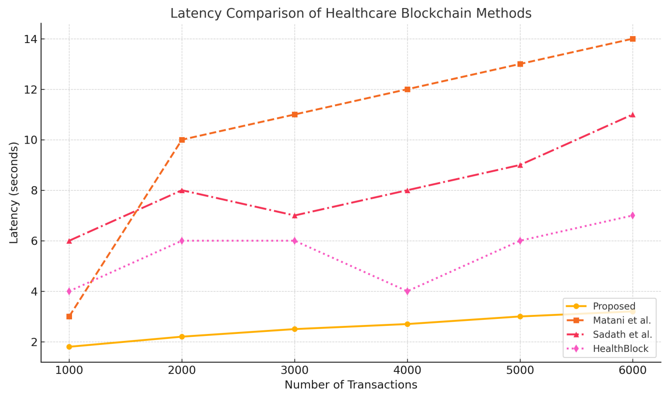

Fig. 5. Latency comparison of the proposed hybrid blockchain system with existing methods across increasing transaction volumes.

2) Throughput comparison with prior methods: Fig. 6 illustrates the throughput performance of the proposed method compared to existing blockchain-based healthcare systems developed by Matani et al. [20], Sadath et al. [11], and HealthBlock [21]. Throughput is measured as the number of transactions successfully processed per second across increasing transaction volumes.

The proposed method demonstrates superior scalability, with throughput values rising linearly from approximately 560 to over 1,850 transactions per second, as the number of transactions increases from 1,000 to 6,000. This is attributed to the efficient batch processing design, lightweight Merkle root anchoring, and minimal on-chain data overhead. In contrast, existing approaches such as Matani et al. and Sadath et al. show relatively static or non-linear throughput behavior due to more extensive on-chain storage and lack of optimization in data handling.

HealthBlock performs moderately well but exhibits inconsistent throughput beyond 4,000 transactions. The consistent and scalable performance of the proposed system confirms its suitability for high-throughput healthcare data environments.

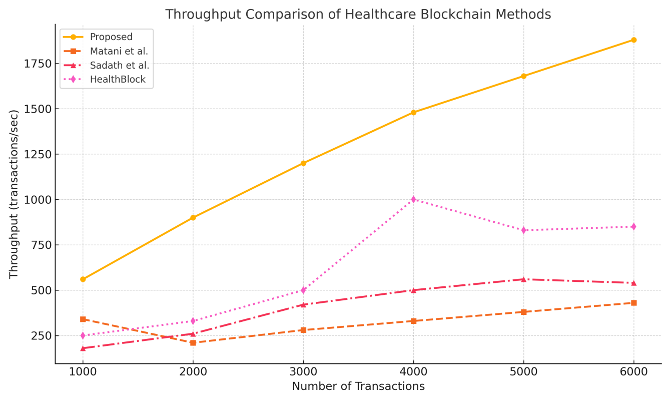

Fig. 6. Throughput comparison of the proposed system with existing methods under varying transaction volumes.

3) Gas cost comparison with prior healthcare blockchain frameworks: Table IV provides a detailed comparative analysis of the proposed hybrid blockchain system against two notable healthcare blockchain frameworks—MEDACCESSX and AdaSec-Health. The evaluation focuses on gas optimization techniques, contract architecture, batch processing strategy, and scalability models.

K. Influence of Parameters

The influence of algorithmic parameters has been systematically examined to highlight their role in the performance of the proposed architecture. The batch size has a direct impact on gas optimization: larger batches significantly reduce transaction costs by minimizing the number of on-chain commitments, although they may introduce a slight increase in latency during Merkle root computation. The AES-256-GCM encryption parameters ensure robust data security while imposing only negligible overhead on processing time, making them suitable for large-scale deployments. In contrast, IPFS upload and retrieval times contribute substantially to off-chain latency, as formalized in Eq. (1) and Eq. (2). The gas price in the Ethereum network strongly influences the overall deployment cost; however, the batching mechanism employed in this study demonstrates resilience against network congestion and fluctuating gas conditions. Finally, the Merkle tree depth, inherently determined by batch size, affects proof generation and verification complexity. Collectively, these observations underscore the necessity of careful parameter tuning to achieve a balanced trade-off among scalability, security, and efficiency in real-world healthcare data management scenarios.

V. Conclusion

The findings demonstrate that the proposed hybrid blockchain framework achieves a practical balance between privacy preservation, scalability, and cost efficiency in healthcare data management. By integrating AES-256-GCM encryption, IPFS-based off-chain storage, and Merkle root anchoring on Ethereum, the system ensures end-to-end confidentiality and verifiable integrity while reducing gas consumption by nearly 99.85% compared to conventional on-chain storage.

An important observation is that off-chain operations—particularly IPFS upload and retrieval—dominate overall system latency. This suggests that optimizing distributed storage through improved node configurations, caching strategies, or integration with Layer-2 storage solutions may yield greater performance benefits than adjusting blockchain parameters alone.

In comparison with existing healthcare blockchain frameworks, the proposed design introduces novelty by simultaneously ensuring confidentiality, integrity, and scalability in a cost-effective manner. This positions the framework as a practical and viable solution for strengthening trust, security, and reliability in digital healthcare ecosystems.

Future work will extend this study by deploying the framework in real-world blockchain environments, integrating decentralized identity mechanisms for stronger patient-centric control, and implementing adaptive access control for secure multi-stakeholder sharing. Additional directions include exploring Layer-2 scalability enhancements, supporting unstructured healthcare data such as medical imaging and IoT streams, and enabling cross-chain interoperability to facilitate secure record exchange across healthcare institutions.

References

[1] M. Hölbl, M. Kompara, A. Kamišalić, and L. N. Zlatolas, "A systematic review of the use of blockchain in healthcare," *Symmetry*, vol. 10, no. 10, p. 470, 2018.

[2] S.-K. Kim and H. C. Vong, "Secured network architectures based on blockchain technologies: A systematic review," *ACM Computing Surveys*, 2025.

[3] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*. Ieee, 2017, pp. 557–564.

[4] L. Ismail, H. Materwala, and S. Zeadally, "Lightweight blockchain for healthcare," *IEEE access*, vol. 7, pp. 149 935–149 951, 2019.

[5] S. Kakei, Y. Shiraishi, M. Mohri, T. Nakamura, M. Hashimoto, and S. Saito, "Cross-certification towards distributed authentication infrastructure: A case of hyperledger fabric," *IEEE Access*, vol. 8, pp. 135 742–135 757, 2020.

[6] G. Al-Sumaidaee, R. Alkhudary, Z. Zilic, and A. Swidan, "Performance analysis of a private blockchain network built on hyperledger fabric for healthcare," *Information Processing & Management*, vol. 60, no. 2, p. 103160, 2023.

[7] F. A. Khan, M. Asif, A. Ahmad, M. Alharbi, and H. Aljuaid, "Blockchain technology, improvement suggestions, security challenges on smart grid and its application in healthcare for sustainable development," *Sustainable Cities and Society*, vol. 55, p. 102018, 2020.

[8] V. K. Rathi, V. Chaudhary, N. K. Rajput, B. Ahuja, A. K. Jaiswal, D. Gupta, M. Elhoseny, and M. Hammoudeh, "A blockchain-enabled multi domain edge computing orchestrator," *IEEE Internet of Things Magazine*, vol. 3, no. 2, pp. 30–36, 2020.

[9] S. Namasudra and G. C. Deka, *Applications of blockchain in healthcare*. Springer, 2021, vol. 83.

[10] S. Sharmin, I. H. Sarker, M. Shamim Kaiser, and M. S. Arefin, "Interplanetary file system-based decentralized and secured electronic health record system using lightweight algorithm," in *Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021*. Springer, 2021, pp. 691–702.

[11] L. Sadath, D. Mehrotra, and A. Kumar, "Scalability performance analysis of blockchain using hierarchical model in healthcare," *Blockchain in Healthcare Today*, vol. 7, pp. 10–30 953, 2024.

[12] A. Ali, H. Ali, A. Saeed, A. Ahmed Khan, T. T. Tin, M. Assam, Y. Y. Ghadi, and H. G. Mohamed, "Blockchain-powered healthcare systems: enhancing scalability and security with hybrid deep learning," *Sensors*, vol. 23, no. 18, p. 7740, 2023.

[13] A. A. Mazlan, S. M. Daud, S. M. Sam, H. Abas, S. Z. A. Rasid, and M. F. Yusof, "Scalability challenges in healthcare blockchain system—a systematic review," *IEEE access*, vol. 8, pp. 23 663–23 673, 2020.

[14] J. Jayabalan and N. Jeyanthi, "Scalable blockchain model using off-chain ipfs storage for healthcare data security and privacy," *Journal of Parallel and distributed computing*, vol. 164, pp. 152–167, 2022.

[15] S. Shamshad, K. Mahmood, S. Kumari, C.-M. Chen *et al.*, "A secure blockchain-based e-health records storage and sharing scheme," *Journal of Information Security and Applications*, vol. 55, p. 102590, 2020.

[16] T. Mazhar, S. khan, T. Shahzad, M. A. khan, M. M. Saeed, J. B. Awotunde, and H. Hamam, "Generative ai, iot, and blockchain in healthcare: application, issues, and solutions," *Discover Internet of Things*, vol. 5, no. 1, p. 5, 2025.

[17] A. Bathula, S. K. Gupta, S. Merugu, L. Saba, N. N. Khanna, J. R. Laird, S. S. Sanagala, R. Singh, D. Garg, M. M. Fouda *et al.*, "Blockchain, artificial intelligence, and healthcare: the tripod of future—a narrative review," *Artificial Intelligence Review*, vol. 57, no. 9, p. 238, 2024.

[18] H. Taherdoost, "Privacy and security of blockchain in healthcare: Applications, challenges, and future perspectives," *Sci*, vol. 5, no. 4, p. 41, 2023. [Online]. Available: https://www.mdpi.com/2413-4155/5/4/41

[19] M. S. B. Kasyapa and C. Vanmathi, "Blockchain integration in healthcare: A comprehensive investigation of use cases, performance issues, and mitigation strategies," *Frontiers in Digital Health*, vol. 6, p. 1359858, 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fdgth.2024.1359858/full

[20] A. Matani, A. Sahafi, and A. Broumandnia, "Improving scalability in blockchain systems using multi-level sharding based on heterogeneity of network nodes," *Computing*, vol. 107, no. 2, pp. 1–34, 2025.

[21] B. Zaabar, O. Cheikhrouhou, F. Jamil, M. Ammi, and M. Abid, "Healthblock: A secure blockchain-based healthcare data management system," *Computer Networks*, vol. 200, p. 108500, 2021.

[22] G. Shi, M. Qi, Q. Zhong, N. Li, W. Gao, L. Zhang, and L. Gao, "Medaccessx: A blockchain-enabled dynamic access control framework for iomt networks," *Sensors*, vol. 25, no. 6, p. 1857, 2025. [Online]. Available: https://doi.org/10.3390/s25061857

[23] R. Alghamdi, S. Qamar, F. K. Shaikh, and M. A. Alqarni, "Improving data transmission through optimizing blockchain sharding in cloud–iot based healthcare applications," *Egyptian Informatics Journal*, vol. 26, no. 2, pp. 123–135, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110866525000544

[24] K. P. N. Rao and S. Chinnaiyan, "Blockchain-powered patient-centric access control with midc aes-256 encryption for enhanced healthcare data security," *Acta Informatica Pragensia*, vol. 13, no. 3, pp. 374–394, 2024. [Online]. Available: https://aip.vse.cz/pdfs/aip/2024/03/03.pdf

[25] Y. Wang *et al.*, "Aes-256 encryption for secure file storage in blockchain systems," *International Journal of Innovative Research in Technology*, vol. 11, no. 10, pp. 4488–4490, 2020. [Online]. Available: https://ijirt.org/publishedpaper/IJIRT174494_PAPER.pdf