Minimax Theory and its Applications

Volume 11 (2026), No. 5s, 39-57

A Blockchain-Enabled Framework for Integrating Electronic Health Records and Digital Payments in Healthcare: Design, Reliability, and Performance Evaluation

Arunendra Mani Tripathi [1], Kavita Saini[2*], Nelson Ponnu Duari[3], Nalini Manogaran[4], Koteeswaran Seerangan[5]

[1] School of Computing Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India
[2] School of Computing Science and Engineering, Galgotias University, Delhi NCR, India
[3]Department of Civil Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology,Avadi, Chennai – 600 062, Tamil Nadu, India
[4]Department of CSE (Cyber Security), S.A. Engineering College (Autonomous), Chennai – 600077, Tamil Nadu, India.
[5]Department of Computer Science and Engineering, R.M.K. Engineering College (Autonomous), Kavaraipettai - 601206, Tamil Nadu, India.
Email: arunendramanitripathi@gmail.com, kavitasaini_2000@yahoo.com, nelson.durai89@gmail.com, drnalini@saec.ac.in,skn.cse@rmkec.ac.in

Abstract
The integration of electronic health records (EHRs) and digital payment services is essential for efficient healthcare administration; however, fragmented interfaces, duplicate payment requests, inconsistent invoice updates, and limited auditability continue to challenge transaction reliability and data security. Existing blockchain-based healthcare studies primarily emphasize medical-record management or remote patient monitoring, providing limited coverage of the complete EHR-to-payment lifecycle, including gateway verification, failure recovery, and transaction reconciliation. To address these limitations, this study proposes a Blockchain-Enabled EHR–Payment Integration Framework that combines standardized application programming interfaces (APIs), gateway-authoritative payment verification, idempotency-based duplicate prevention, cryptographic audit records, and automated reconciliation. An optional Transformer-based transaction-risk detection module identifies potentially fraudulent transactions without replacing payment authorization. The framework is implemented using Python, Hyperledger Fabric, Hyperledger Caliper, REST APIs, and PyTorch. Risk detection is evaluated using the Harmonized Multi-Source Fraud Transaction Dataset for Benchmarking Synthetic Data Generation Methods, comprising 5,000 balanced transactions, while system testing considers workloads of 100, 500, and 1,000 transactions. Evaluation examines transaction throughput, latency, processing correctness, audit integrity, recovery behaviour, scalability, and resource utilization. Operation research and reliability theory are used to capture end-to-end transaction reliability, bounded recovery probability, system availability, and gateway-ledger efficiency under both sequential and concurrent workload execution. In the experimental analysis, system reliability is derived from completed transaction phases, and recovery effectiveness is characterized by the observed resolution rate, MTBF, and MTTR and queueing efficiency. Workload saturation is also evaluated against the calculated utilization condition to characterize the system's scalability. This framework provides a privacy-preserving and securesome transaction integration architecture that protects payment-state integrity, avoids duplicates, and facilitates recoverable transactions. The combined reliability-performance evaluation provides an empirical basis for examining transaction efficiency, recovery, availability, and operational overhead, instead of presuming linear enhancement.

Keywords*:* Electronic Health Records, Blockchain Technology, Digital Payment Integration, Hyperledger Fabric, Payment Verification, Transaction Reconciliation, Fraud Detection, Operations Research, Reliability Theory

1. Introduction

The adoption of electronic health record (EHR) systems and digital payment technologies has proven crucial in today's healthcare landscape, enabling effective patient information management, billing, and financial transactions [1], [2]. Combining these technologies can enhance administrative coordination, minimize manual processes, and offer clear visibility of healthcare payments [3]. In the health care sector, however, several actors are working on widely different and isolated information systems [4], [5] such as health centers, hospitals, laboratories, payment agents, insurance companies, and the patients. So it remains challenging to reliably get clinical and financial information to and from the patient. Blockchain technology could transform the healthcare

record keeping, verification and tracking process, enabling trust to be established [6], [7]. All these benefits make EHR integration with digital payment easy, but it's not just about secure data storage [8], [9]. Ensuring the payment verification, duplicate transactions, invoice-status synchronization, privacy protection and failover recovery are coordinated is essential for maintaining the same clinical and financial records [10], [11]. The hospital may have different payment statuses in its systems than the payment gateway, which can lead to discrepancies in administrative processes and payment reliability.

Pre-existing works have explored the use of blockchain for EHR management, privacy protection with regard to medical data sharing, access control, secure storage and interoperability of healthcare institutions. Payment security has been explored elsewhere with the use of blockchain and deep learning methods to identify payment fraud. While these strategies help to secure health information and financial data, end to neglect the entire EHR-to-payment transaction workflow. However, record sharing solutions based on blockchain don't always offer features for payment verification, idempotent request handling, processing refunds, or recovering from gateway timeouts. Likewise, a fraud-detection model cannot ensure that payments correct, that the synchronization is correct, or that transactions properly reconciled. Additionally, the computational cost and operational reliability of the integration of these components must be carefully analysed for a range of workloads and failure scenarios. In order to overcome these, this research proposes a framework based on blockchain technology that incorporates the EHR information and digital payment service, on the basis of standardized application programming interfaces, invoice-related transaction identifiers, authoritative validation, idempotent processing, and recovery-based reconciliation. Only sensitive clinical information is off-chain and blockchain is used to provide an enabling audit tool. It also has the capability to trigger QR payments, and the optional deep-learning module can detect potential payment risks. Its performance and reliability using various parameters such as transaction success rate, API errors, audit integrity, recovery time, resource utilization, synchronization consistency and latency, both sequentially and concurrently. The study offers a reproducible assessment approach and there are no assumptions of improvements until substantiated through experimentation.

1.1 Problem Statement

Payment Gateways, billing applications and other platforms continue to be essential for healthcare organizations to integrate their electronic health records (EHRs) with digital payment options, adding to the complexity. Communication between these systems is not consistent leading to inconsistencies in payments status, duplicate transaction requests, incomplete updates to invoices and late payments and invoice reconciliation [12], [13]. Centralized architectures also provide some level of protection against tamper, as they can provide a degree of cross-organisational auditing of the data, however, the sharing of private clinical data on shared ledgers is also a privacy concern. While payment-centric solutions often include security for transactions or fraud detection, the existing blockchain-powered healthcare applications mostly focus on security of medical records, access control, and sharing of health data. Thus, a common standard to integrate EHR data, payment verification, transaction integrity and failure recovery is not well defined. Moreover, the sequential workloads and concurrent workloads are not evaluated systematically, limiting the understanding of the operational reliability, synchronization consistency, and blockchain overhead. It is therefore necessary to have an overarching framework, which ensures secure, traceable and reliable payments in the healthcare sector, safeguards clinical data privacy and enables the evaluation of performance to be repeatable.

1.2 Research Motivation

The growing reliance on digital healthcare services drives the need for secure and efficient systems to manage clinical data and payments. Payment discrepancies, duplicate request, lack of communication and disjointed audit trails can be a threat to administrative efficiency and mistrust of transactions. Auditing and standardized APIs are tamper-proof, processing is idempotent and there is reconciliation to protect data from tampering and to ensure seamless integration. Analysing these capabilities as a whole can help to gain insight into their real-world usefulness, limitations, and effectiveness when deployed in a realistic transactional workload in a healthcare environment.

1.3 Significance of the Study

The design in this study offers a holistic design of coordination of EHR systems with digital payments based on verified transaction states, privacy-preserving blockchain auditing and recovery-reconciliation. The main emphasis of the framework is on removing unnecessary processing, uniformity in payments information and visibility in the flow of money transactions. The results can feed into designs of secure, interoperable payment flows for healthcare technology developers, and suggest future directions for empirical research on the implementation of these payment flows and potential trade-offs.

1.4 Key Contributions

- Proposes blockchain-enabled integration of electronic health records, billing systems, and digital payment services.
- Ensures reliable payment verification, duplicate prevention, and consistent invoice-status updates through idempotent processing.
- Maintains tamper-evident blockchain audit records while protecting sensitive clinical information through off-chain storage.
- Enables transaction recovery and reconciliation during timeouts, synchronization failures, refunds, and payment reversals.

- Establishes systematic performance evaluation using latency, throughput, reliability, consistency, recovery, and resource utilization metrics.
- Applies operations research and reliability theory, using reliability and performance models, to evaluate transaction efficiency, system availability, payment verification, and recovery mechanisms.

2. Related Works

Lee et al. [14] Suggested a blockchain-based system for exchanging healthcare information that is searchable and has improved the privacy and usability of electronic health records (EHRs). The framework collects the information from Internet of Medical Things (IoMT) devices, arranges the information into EHR blocks, and distributes the records using a blockchain network. Search is a feature that allows authorized users to find and use certain healthcare information in an efficient manner. The results reported show enhanced security and convenient data access. However, these quantitative performance results and its scalability with high workloads should be further investigated in healthcare applications.

Mandarino et al. [15] Suggested a blockchain-electronic health record (EHR) system with edge computing to improve the security and cost-effectiveness of healthcare data. Through the transparent, immutable, and decentralized trust characteristics of blockchain, the system promotes cooperation among different healthcare software systems while maintaining the integrity and access to medical records. One of the challenges of healthcare data processing and management is overcome by edge computing. The information available, however, provides limited information about the quantitative performance achievements, and further evaluation is needed to determine scalability, effectiveness and performance in a variety of healthcare settings.

Guimarães et al. [16] proposed an Internet of Things (IoT)-based context-aware electronic health record (EHR) system that merges IoT devices and blockchain technology to enhance the efficiency of inpatient medical care and reliability of EHR data. This framework leverages beacon based location information to fine-tune medical-device interfaces for their current context, and stores relevant data in Hyperledger Fabric, making it immutable and for which access can be audited. The strategy focuses on minimizing healthcare-related health errors and ensuring that healthcare workers have easy access to pertinent data. Its ability to decrease errors and to be scalable in various hospital settings, however, needs to be empirically validated.

Li et al. [17] proposed TrustHealth, secure electronic healthcare data-sharing ecosystem based on blockchain and trusted execution environments (TEE) with the aim of preserving patient privacy. The system's data integrity is ensured by the application of blockchain smart contracts, sensitive SQLCipher operations are isolated with ARM TrustZone, and a TEE-enabled session-key generation protocol for secure authentication and key exchange is introduced. With Hyperledger Fabric, TrustHealth can process 5,000 records per second of encrypted healthcare data. Cost of deployment, scalability of the system throughout a heterogeneous infrastructure in the healthcare sector, and integration issues in the real world still need to be studied.

Hossain et al. [18] Analysed and discussed permissioned blockchain technologies for healthcare data and databases, with a specific interest in interoperability, privacy, security, performance, accessibility and scalability. The research examines various platforms like Hyperledger Fabric, Corda, Quorum, and MultiChain for their applicability in securing confidential patient health information and ensuring regulatory compliance, including HIPAA standards. Opportunities, challenges and research directions for secure healthcare information management are identified. The study however, does focus on existing technology and not suggest and prove a new framework, which reduces the amount of evidence of a real performance gain.

Lax et al. [19] Suggested a public blockchain based system for safe and secure sharing of health information between healthcare organizations. The system hides the connection between patients and their EHRs and only allows access to the records by patients' authorized entities. The digital identity mechanism used for authentication and access control is compliant with the eIDAS directive and is based on the Ethereum blockchain. The solution implemented is effective and affordable, and is now applied beyond private and consortium networks for the sharing of healthcare data. However, the risk of public blockchain facing scalability, transaction speed, and privacy concerns of any nature may be questions that need to be explored.

Psarra et al. [20] Proposed an EHR access control management system based on permissioned blockchain network. The combination of blockchain's decentralized and immutable features with fuzzy logic can enable the framework to be utilized for context-aware access control decisions in a sensitive healthcare environment while ensuring integrity and security in access control. The permissioned network is intended to enable healthcare organisations to join in with permission, making it easier to share and access secure EHRs. There is some weak evidence, however, for quantitative performance improvements, and additional studies are required to assess the ability to scale up, accuracy of decision making and the ability to adapt to different clinical contexts.

Liu et al. [21] proposed an electronic medical record (EMR) sharing scheme based on consortium blockchain, InterPlanetary File System (IPFS) and proxy re-encryption which is secure and efficient. This approach securely stores medical records in an encrypted format on IPFS, while the hash addresses of the shared records are stored on the blockchain, letting authorized users decrypt the shared records without compromising security. Based on attribute access control also provides custom made authorizations and privacy protection [26]. Through security analysis and performance evaluation, computational and communication costs reduction in comparison with the existing schemes are proven. However, in the real world its scalability and interoperability across a range of differing health care infrastructure networks have yet to be explored.

3. Proposed Blockchain-Enabled EHR and Digital Payment Integration Framework

The proposed approach investigates a blockchain-based system that combines the EHR systems with digital payment systems to guarantee their trustworthiness, privacy, and traceability. First, the transaction records are preprocessed and converted into a consistent feature representation, then class-imbalance analysis and synthetic-data assessment can be performed, which helps to detect transaction risk. A deep learning model based on a Transformer is then used to estimate potential fraud activity, and provides risk scores to support additional decisions. The integration layer enables EHR billing systems and payment gateways to communicate with each other via standard application programming interfaces, transaction identifiers that are tied to invoices and consistent idempotency keys. Authoritative payment verification verifies that the update to an invoice is only made after the results have been verified by the gateway. A permissioned blockchain ensures the audit records are kept with minimal transaction information and cryptographic hashes so that are tamper evident and sensitive clinical information is stored off-chain. In addition, a reconciliation and recovery process handles the issues of communication timeouts, synchronization failures, duplicate requests, refunds and incomplete audit submissions. Finally, systematical experiments test the predictability, latency, throughput, consistency of transactions, recovery time, resource usage and overhead in the blockchain under different workloads. Figure 1 depicts the proposed blockchain-based integration of the EHR with payment system that includes data preprocessing, Transformer-based fraud-risk identification, API-based payment verification, blockchain audit, and payment reconciliation. The framework ensures transaction consistency, duplicate prevention, reliable recovery, data integrity, and systematic performance monitoring.

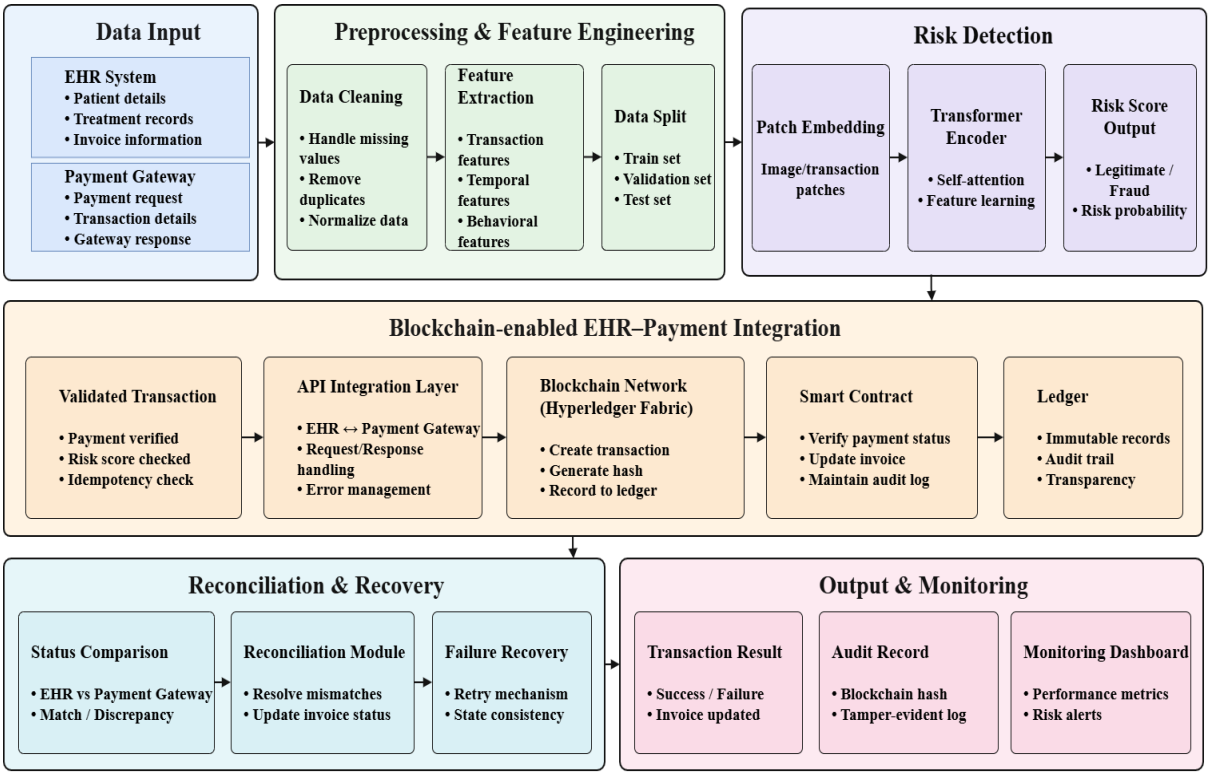

Figure 1. Block diagram of the proposed blockchain-enabled EHR and digital payment integration framework

3.1 Data Collection

The dataset consists of 5000 balanced transactions including 2500 legitimate and 2500 fraudulent samples from different sources: IEEE-CIS, ULB, and PaySim. Standardized timestamps, amounts, type of transactions, fraud labels, source information are all attributes available [22]. Supporting files contain temporal validation splits, harmonization specs and synthetically generated data, based on SMOTE, CTGAN and WGAN. These data can be used for training and testing an optional transaction-risk detection module. Because no events for integration with EHR payments are available in the data set, further synthetic transaction logs are needed for the evaluation of the reliability of the system.

3.2 Data Preprocessing and Feature Engineering

The results of this stage are used to convert the harmonized fraud dataset and the synthetic EHR–payment integration logs into consistent, leakage-free representations of features for transaction-risk detection and reliability assessment of the system. The input is a set of 5000 transactions with their time, amount, transaction type, fraud label and source identifier, as well as a set of integration events with their invoice references, transaction identifiers, gateway responses, retries, and synchronization timestamps. The result is a matrix of the

transactions with an integrated data set of the integration events for further model training and framework testing.

Input validation and preprocessing: Missing values, duplication, inconsistent categories, invalid timestamps and non-finite numerical values in the data are identified with the aid of the data dictionary. All of the records are removed that are exactly the same, and repeating requests for the same transaction are stored in integration logs, very important to the retry behaviour. Unknown numeric values are filled in with the median value from the training set and unknown categorical values are given a specific "unknown" category. Transaction amounts are converted to numbers, transaction type are one hot encoded, and the timestamps are converted into a consistent datetime format. For numerical features, the statistics from the training set are used to normalize the data as in equation (1).

$$z_j = \frac{x_j - \mu_j}{\sigma_j + \epsilon} \qquad (1)$$

where x_j denotes the observed feature, μ_j and σ_j represent its training-set mean and standard deviation, respectively, and ϵ is a small constant preventing division by zero. Fraud labels and source identifiers are kept separate for supervised learning, and source-wise evaluation, respectively, but could leak the source or target origin from predictive inputs. Synchronization delay for event i is defined as in equation (2).

$$D_i = t_i^{\text{sync}} - t_i^{\text{gateway}} \qquad (2)$$

Here, t_i^{sync} is the timestamp at which the payment state appears in the EHR or billing system and t_i^{gateway} is the authoritative gateway-response timestamp. Positive values mean synchronization-delay while negative values mean timestamp-consistency-check. These features provide the ability to detect repeat requests, updates that are late, and unusual or suspicious transaction sequences, but don't presume that all rejections are fraudulent. The payment status is not dependent on the risk model, but rather on the rules defined in the gateway verification/reconciliation.

Leakage prevention and output generation: All the parameters of the pre-processing are fitted on the training data, and the provided temporal splits are kept if any. There is one partition for all the events of the same transaction and transaction sequence. Information that is not available at the time of prediction, such as the response at the gateway and final reconciliation result are not inputs to the model. This yields fraud feature matrix ready for training, and a timestamp-aligned integration log dataset. The major value-added feature is the collaborative creation of features for fraud prediction and indicators for operational reliability, so that transaction-risk analysis does not mix in the performance of the prediction model with the correctness of the payment or reliability of the system.

3.3 Class Imbalance and Synthetic-Data Analysis

This stage checks if synthetic data augmentation is able to improve the fraud detection performance under a realistic class imbalance scenario. The preprocessed transaction-feature matrix produced is provided as input, along with the fraud labels and the datasets provided by the repository (SMOTE, CTGAN and WGAN). The dataset has 2,500 real transactions and 2,500 fake transactions, but an extra imbalanced training created to represent the scenario of fake transactions in real payment scenarios. The original distribution of the untouched validation and test sets preserved so that could be fairly evaluated. The proportion of frauds is givenas in equation (3).

$$P_f = \frac{N_f}{N_f + N_l} \times 100 \qquad (3)$$

where P_f denotes the fraud percentage, N_f represents the number of fraudulent transactions, and N_l represents legitimate transactions. The imbalanced training scenario created by reducing legitimate-class representation relative to fraudulent samples, with the selected class ratio documented explicitly. SMOTE generates minority samples through interpolation between neighbouring feature vectors, whereas CTGAN and WGAN learn synthetic feature distributions using generative models. These methods compared separately under identical training conditions. Repository-provided synthetic samples used only after verifying their generation procedure and ensuring that no validation or test information contributed to their creation.

Quality assessment and output: Invalid amounts, previously duplicated records, inconsistent categorical values and unrealistic feature combinations checked with synthetic samples. The real vs. synthetic sample comparisons made to determine if meaningful transaction characteristics are maintained in augmentation. The following precision, recall, F1-score, PR-AUC and ROC-AUC then be computed on the untouched real validation and test records for each augmentation strategy. Recall is an important consideration for minorities, especially since the amount of money that could be lost on unauthorised transactions is a risk.

3.4 Deep Learning-Based Transaction Detection

This stage acquires transaction patterns, which are related to the financial activities, from the preprocessed features and the selected training samples, and learns the patterns of transaction based on transaction types.In this stage, the transaction patterns associated with the legitimate and fraudulent financial activities are learned from the preprocessed features and the selected training samples, and transaction patterns are learned based on transaction types. The data is collected as a transaction, transaction type, temporal information and other verified predictive attributes; the fraud labels are used as supervised learning targets. Considering the similarities in relationships between transaction features, a Transformer-based Classifier is used to learn using

self-attention, which is appropriate for the evaluation of relationships between transaction features. Due to the few number of transactions in the data, the complexity of the model kept in check to prevent overfitting.
Numerical and encoded categorical features are projected into a common embedding space. For transaction representation X, self-attention computes contextualized feature relationships as in equation (4).

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \qquad (4)$$

where Q, K, and V represent query, key, and value matrices derived from the feature embeddings, and d_k denotes the key-vector dimension. The resulting representation is passed to a classification head that estimates the probability of fraud as in equation (5).

$$\hat{p}_i = \text{sigmoid}(W_c h_i + b_c) \quad (5)$$

where h_i is the learned representation of transaction i, W_c and b_c are trainable classifier parameters, and $\hat{p}_i$ is its predicted fraud probability. The model trained using binary cross-entropy, with class weighting considered when training on the imbalanced scenario. An initial configuration may use two Transformer encoder layers, four attention heads, a 128-dimensional embedding, dropout of 0.1, AdamW optimization, and a learning rate of 10^{-4}; these parameters tuned using validation data.

3.5 Transaction Risk Scoring and Decision Support

This stage maps the probabilities of fraud computed from the trained Transformer classifier to meaningful risk scores for the workflow in which a payment is made that is tied to the EHR. The input includes the amount of fraud probability, transaction ID, invoice ID, and the status of the payment. A risk score to 100 is being given to each transaction as in equation (6).

$$R_i = 100\hat{p}_i \qquad (6)$$

where R_i denotes the risk score for transaction i, and $\hat{p}_i$ represents the predicted fraud probability. A decision threshold τ, selected using validation data to balance fraud detection and false alarms, classifies transactions as normal or potentially suspicious as in equation (7).

$$C_i = \begin{cases} \text{Suspicious}, & \hat{p}_i \geq \tau \\ \text{Normal}, & \hat{p}_i < \tau \end{cases} \quad (7)$$

Here, C_i represents the risk category, and τ is the validation-selected threshold between 0 and 1. For initial testing, a threshold of 0.5 may be used as a baseline, but the final threshold selected empirically based on validation precision–recall performance. Normal-risk transactions still follow the normal payment process, whereas suspicious transactions receive further verification (or manual review) based on pre-defined policies. Risk scores are not used to determine if payment has been made. Instead, the success, failed and pending status is still controlled by the payment gateway. Likewise, idempotency keys and unique transaction identifiers ensure that there isn't a risk associated with any of the transactions being processed more than once. Unconfirmed transactions are reconciled with the gateway before they are updated in the financial state of the transaction.

3.6 EHR–Digital Payment API Integration

This stage links the health care billing system, EHR system, payment gateway and transaction-risk module via standardized application programming interfaces (APIs). The inputs are a patient encounter reference, an invoice identifier, the amount to be paid, currency, the credentials of the client(s) that authorized the transaction, and the risk assessment's output. The integration service checks the required fields, validates client permissions, validates the validity of the invoice and generates a transaction ID that is associated with the invoice. Payment requests are not made for sensitive clinical information unless it is specifically needed for a Healthcare Operation that is authorized.
Transaction identification and request processing: Each logical payment has an Idempotency key that is assigned by the client or integration service that is stored with the invoice reference. The key can help prevent extra costs being incurred from network interruptions in the event of a retry. The system verifies if the key exists or not, before it starts payment as in equation (8).

$$K_i = H(I_i \parallel A_i \parallel N_i) \qquad (8)$$

where K_i is the generated key, H denotes a cryptographic hash function, I_i is the invoice identifier, A_i is the payment amount, and N_i is a unique request nonce. The nonce distinguishes separate legitimate payment attempts. In implementation, the idempotency key must remain stable across retries of the same logical request; the database should enforce uniqueness on the key and store the associated transaction state.
Once validated, the API forwards the payment request to the gateway, and stores the payment provider's transaction ID. Dynamic QR codes can store a payment-session reference or a payment URL generated by a payment gateway, enabling the patients to pay without revealing any clinical information. The system maintains explicit states, like successful, declined, unknown, and created, as well as pending. If the same key is used again, it return the result of the transaction, rather than making a new charge. If a timeout occurs, the outcome in an 'Unknown' or 'Pending' state until the gateway verification/reconciliation process resolves the outcome.

3.7 Authoritative Payment Verification

This stage verifies the actual monetary results of each payment that is made using the EHR–digital payment API integration. The following data is passed in: The invoice identifier which is tied to the transaction, the payment gateway identifier, the expected amount, currency, and the most recent transaction status returned by the payment provider. An authenticated gateway API is communicated to or a cryptographically authenticated webhook is checked against to retrieve the payment status. That information is reconciled against the transaction

number, invoice amount, currency and anticipated payment status of the transaction stored to ensure that the wrong or unrelated information doesn't change healthcare billing records. Payment only be deemed complete and successful if both the provider confirms the payment and the transaction details on the payment match the invoice. The verification condition as in equation (9).

$$V_i = S_i \wedge A_i \wedge C_i \wedge T_i \qquad (9)$$

where V_i indicates verified payment success, S_i denotes the provider's confirmed-success status, A_i represents amount consistency, C_i indicates currency consistency, and T_i confirms transaction-reference matching. Each condition is Boolean, and the payment is accepted as verified only when all conditions are satisfied.

If the gateway says that the transaction declined, the transaction gets recorded as declined. If the response is timed out or status is still pending, the transaction remains in the unresolved status and doesn't get marked as failed or successful. The system attempts to retrieve the status again with a configurable exponential backoff (e.g. 1 second, 2 times, a maximum of 5 attempts, respecting gateway's rate limits). If not resolved, cases are referred to reconciliation.

3.8 Blockchain-Supported Audit Recording

The aim of this stage is to keep the evidence of critical moments in the EHR–digital payment process secure and tamper-evident. The inputs are the transaction id from the invoice, payment gateway reference, payment status, event time, outcome of the event (reconciled or not), and the event type (created by the system). Once the payment has been verified by the authoritatives, the integration service builds an audit record with just the bare essentials of the minimum metadata needed for traceability. To minimize privacy concerns, patient names, medical data, authentication details and all payment information is not stored on the blockchain. Each audit event is canonicalised then hash with SHA-256. The digest produced is stored on a permissioned blockchain (e.g., Hyperledger Fabric) together with a secure transaction reference, an event type and a timestamp. The event hash function as in equation (10).

$$h_i = \mathrm{SHA256}(E_i) \qquad (10)$$

where E_i denotes the canonical representation of audit event i, and h_i is its cryptographic digest. Authorized services can later recompute the hash from the retained off-chain event and compare it with the recorded digest to detect unauthorizesd modifications. Blockchain transactions should include access-control policies, authenticated organizational identities, and configurable endorsement rules. The implementation should measure block-commit latency and storage overhead rather than assume negligible blockchain costs.

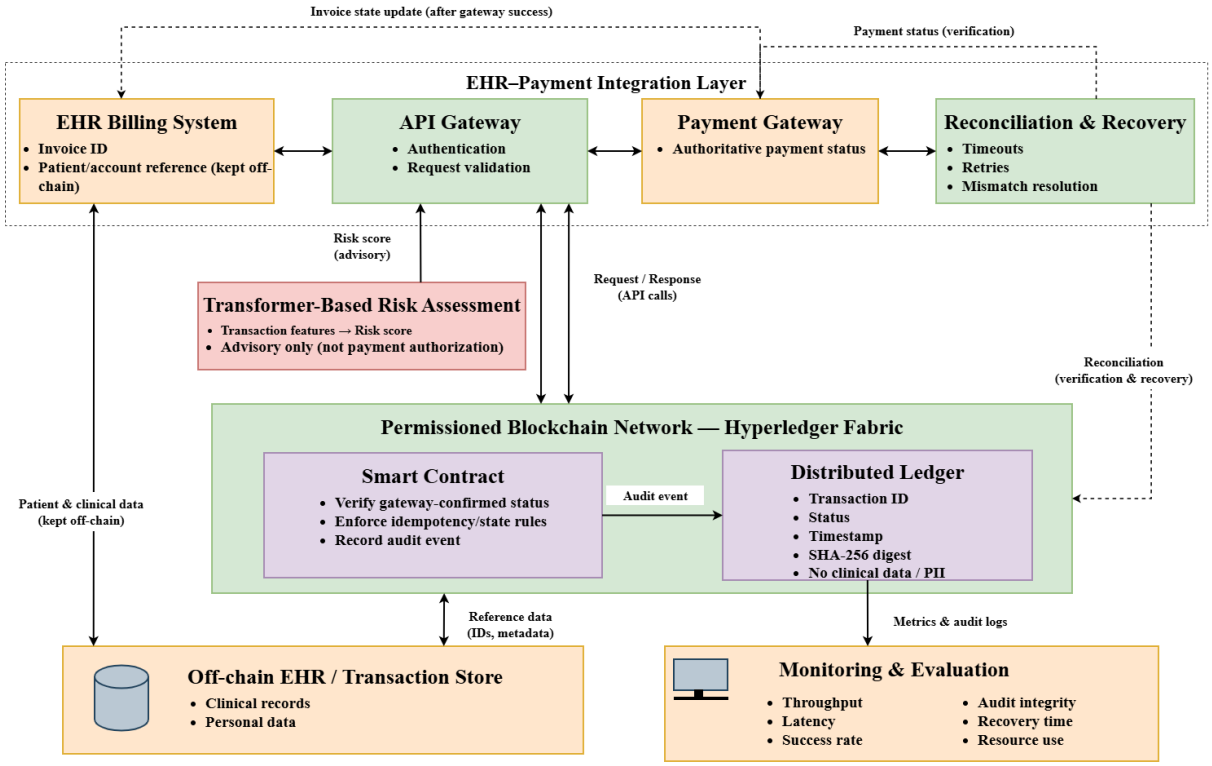

Figure 2. Architecture of the Proposed Blockchain-Enabled EHR–Payment Integration Framework

Figure 2 illustrates the proposed architecture integrating EHR billing, payment gateways, Transformer-based risk assessment, and Hyperledger Fabric. Smart contracts support payment verification and audit logging, while off-chain storage protects sensitive information. Reconciliation and monitoring ensure transaction consistency, reliability, and performance evaluation.

3.9 Reconciliation and Failure Recovery

This stages brings stability to the healthcare billing system, payment gateway, transaction database and blockchain audit records in case of communication breakdown or incomplete updates. The inputs are invoice ID, idempotency key, gateway transaction ID, payment status, payment audit timestamp, pending audit events. Periodically, the reconciliation service finds transactions that haven't been reconciled, such as gateway timeouts, API request interruptions, EHR updates that are delayed, or unsuccessful blockchain submissions. The consistency condition is shown as in equation (11).

$$C_i = \begin{cases} 1, & S_i^G = S_i^E \\ 0, & S_i^G \neq S_i^E \end{cases} \quad (11)$$

where C_i indicates consistency, S_i^G verified gateway state is denoted with and S_i^E indicates the respective EHR billing state. Different states (e.g., pending payments, refunds, reversals) are considered valid intermediate states, not errors.

3.10 Operations Research and Reliability-Theoretic Modelling

At this level, operations research and reliability theory are applied to evaluate the transaction throughput, system availability, payment authentication, and retrieval capability of the suggested framework. The framework is modeled as a serial pipeline of five stages, beginning with API verification (k = 1), risk scoring (k = 2), gateway authentication (k= 3), the blockchain audit trail storage (k = 4) and finally, reconciliation (k= 5). In (12), Rk is the probability of successful execution of stagek and provides the overall reliability of a transaction along the entire end-to-end path.

$$R_{\text{sys}} = \prod_{k=1}^{5} R_k \quad (12)$$

Indicates that the end-to-end reliability of R_{sys} and each R_k is computed from the experimental logs as the fraction of successfully completed events divided by total events at stage k. Because a failure at any stage would not allow the verified payment state, the weakest stage determines R_{sys}, which reveals what component removal is most costly in the ablation experiment. Reconciliation-based recovery can be expressed as a retry with bound: if one round of verification can resolve an outstanding transaction with p_v and we are allowed at most M attempts, the probability of resolution is given by (13).

$$R_{\text{rec}} = 1 - (1 - p_v)^{M+1} \quad (13)$$

Where M = 3 in this case is obtained by referring to the retry limit in Table 1, and the attempts are assumed to be independent. If any transactions still cannot be finished after M retries, they will be unfinish and sent for further investigation, in accordance with Algorithm 1. As shown in eq. (14), system availability is calculated in this study using the MTBF and MTTR from the fault-injection tests.

$$A = \frac{\text{MTBF}}{\text{MTBF+MTTR}} \quad (14)$$

where a higher availability A indicates that the payment-verification and recovery mechanisms restore service quickly after timeouts, interruptions and synchronization failures. Transaction efficiency is analysed with a queueing model of the gateway and ledger service. With arrival rate λ, service rate μ per server and c concurrent servers, the utilization is given in equation (15).

$$\rho = \frac{\lambda}{c\mu} \quad (15)$$

where $\rho < 1$ is required for stable operation, and the response time grows rapidly as ρ approaches 1. The saturation point observed under concurrent workloads is therefore compared with the load predicted by ρ, and throughput, mean latency, success rate and unresolved-request rate are interpreted together with R_{sys}, R_{rec} and A.

Algorithm 1: Blockchain-Enabled EHR–Digital Payment Integration

Input:
- D : Historical transaction dataset
- L : EHR-payment integration event logs
- I : Healthcare invoice and encounter reference
- A, C : Invoice amount and currency
- R : Payment request with stable idempotency key K
- G : Authenticated payment gateway interface
- B : Permissioned blockchain audit interface
- τ : Validated transaction-risk threshold
- M : Maximum recovery attempts

Output:
- S : Verified transaction and invoice state
- Q : Transaction-risk score and review decision
- H : Tamper-evident audit record
- E : Reliability and performance evaluation logs

Dp, Lp ← PREPROCESS(D, L)
Dtr, Dval, Dte ← TEMPORAL_SPLIT(Dp)
Dbal ← ANALYZE_AND_AUGMENT(Dtr)
F ← TRAIN_TRANSFORMER(Dbal, Dval)
Initialize transaction database, audit queue, and evaluation logs

```
For each incoming request R do
  Validate invoice I, amount A, currency C, and client credentials
  If validation fails then
    Return INVALID_REQUEST; record rejection event
  Else
    If idempotency key K already exists then
      Return existing transaction state; do not create a new charge
    End If
    Create transaction identifier T linked to invoice I
    Persist T, K, A, C, and initial state PENDING
    Extract permitted transaction features X
    p ← F.PREDICT_FRAUD_PROBABILITY(X)
    Q ← 100 × p
    If p ≥ τ then
      Mark transaction for additional risk review
      Apply configured review policy without assuming payment failure
    End If
    Submit payment request to G using stable key K
    If gateway response is authenticated and received then
      Verify transaction reference, amount, currency, and status
      If all verification conditions pass and status = SUCCESS then
        S ← VERIFIED_SUCCESS
        Update invoice as paid using an idempotent operation
      Else If verified gateway status = DECLINED then
        S ← DECLINED
        Keep invoice unpaid
      Else
        S ← PENDING_OR_UNRESOLVED
      End If
    Else
      S ← PENDING_OR_UNRESOLVED
    End If
    Persist transaction state and append audit event to durable queue
    For each queued audit event do
      Hash canonical event metadata using SHA-256
      Submit minimal event metadata and digest to B
      If blockchain submission succeeds then
        Mark audit event as recorded
      Else
        Retain event for retry using its unique event identifier
      End If
    End For
  End If
End For
For each unresolved transaction or incomplete audit event do
  attempt ← 0
  While attempt < M and state remains unresolved do
    Retrieve authoritative transaction status from G
    Reconcile gateway, local database, and EHR invoice states
    Apply only verified, idempotent state transitions
    Retry eligible audit submissions without duplicating events
    If consistency is restored then
      Update recovery log; exit While
    Else
      attempt ← attempt + 1
      Wait using bounded exponential backoff
    End If
  End While
  If unresolved after M attempts then
    Flag for manual investigation; preserve pending status
  End If
. End For
Evaluate predictive, performance, consistency, and recovery metrics
```

Return S, Q, H, and E

In Algorithm 1, risk assessment, authoritative verification, blockchain auditing and reconciliation on the basis of recovery are used to manage the payment processing in the field of healthcare. The historical transactions are both preprocessed and partitioned chronologically to train and validate the risk model based on Transformer. Details of the invoice and credentials are checked for each incoming request, with the help of stable idempotency keys, no duplicate processing is allowed. The model attempts to calculate the risk associated with the transaction and the financial authoritativeness status is set by the payment gateway. To audit the blockchain in a tamper evident way, verified outcomes are used for the record of invoices, and cryptographic digests are used to audit the blockchain. Any transactions that have not been resolved and failed audits are placed in a bounded recovery loop. Last but not least, systematical evaluation of predictive, operational, consistency and recovery metrics are taken up.

The proposed framework consolidates within a coordinated transaction lifecycle these functions as opposed to current ones which focus on one dimension of the security, payment processing and/or fraud detection of the healthcare record. Its primary advantage is that it offers gateway verification, idempotent keys, an invoice-level sync of invoices and recovery-based reconciliation to overcome the partial failures and duplicate requests. If sensitive clinical information isn't stored on the blockchain, audit record on blockchain offers improved traceability of the events in the transaction. Also, the optional Transformer-based risk assessment is responsible for identifying suspicious transactions, while complementing payment authorisation. Our design is different from designs evaluated as a sum of individual functions, which leads to a comprehensive evaluation of our design including the prediction performance, operational reliability, interoperability and computational overhead.

4. Results and Discussion

The proposed framework for integration of EHR and payment on the blockchain is tested under controlled transaction loads to see how reliable and secure it is, and how scalable it is. The experimental analysis focuses on the correctness of the payment-verification, prevention of duplicate requests, transaction-risk assessment, and recording of transactions that cannot be tampered with. With sequential and concurrent workloads, the behaviour of systems can be studied as the number of transactions in each workload increases, and with simulated gateway timeouts, network interruptions and synchronization failures, recovery and reconciliation mechanisms are checked. Synthetic-data generation strategies are also explored to understand their role in the (optional) fraud-risk detection module using a Transformer. Observation of the computational resources and the controlled removal of the various components further validates assessment of implementation efficiency and the contribution of the various components. The resulting evidence is then analysed for predetermined configurations and baseline practices but not as pre-determined performance goals. The following simulation parameters are chosen and used in this study shown in Table 1.

Table 1. Simulation parameters

Simulation parameter	**setting**
Programming language	Python 3.11
Blockchain platform	Hyperledger Fabric
Blockchain benchmark tool	Hyperledger Caliper
API framework	FastAPI (REST APIs)
Deep-learning framework	PyTorch
Smart contract language	JavaScript or Go; select one consistently
Initial network size	2 organizations
Initial channel configuration	1 channel
Ordering service	Raft
Audit record format	Transaction reference, event type, timestamp, cryptographic digest
Hashing algorithm	SHA-256
Clinical data storage	Off-chain only
Fraud dataset	Harmonized Multi-Source Fraud Transaction Dataset, Version 3
Total transaction records	5,000
Legitimate / fraudulent records	2,500 / 2,500
Training / validation / test split	70% / 15% / 15% (3,500 / 750 / 750 records)
Transformer optimizer	AdamW
Batch size	32
Maximum training epochs	30
Early stopping patience	5 epochs
Augmentation methods	SMOTE, CTGAN, WGAN
Minority-class retention levels	10%, 20%, 40%
Transactions per workload	100, 500, 1,000

Execution patterns	Sequential and concurrent
Concurrent request levels	1, 10, 25, 50
Repetitions per configuration	5 independent runs
Transaction types	Approved, declined, pending, duplicate, invalid
Payment verification authority	Payment gateway
Idempotency	Stable key per logical payment request
Invoice update rule	Update to paid only after verified success
API timeout	5 seconds
Maximum retry attempts	3, excluding the initial request
Retry policy	Exponential backoff; re-verify payment status before retrying financial actions
Failure scenarios	Gateway timeout, network interruption, duplicate request, status mismatch, audit-write failure
Recovery approach	Reconciliation and bounded retries
Unresolved payment handling	Retain pending state until authoritative verification
Resource observations	CPU, memory, network utilization
Hardware and operating system	Record actual CPU, RAM, OS, and deployment configuration
Network conditions	Record latency and bandwidth; keep consistent across comparisons

4.1 Transaction Data Characteristics and Risk-Model Evaluation

This section investigates the distributions of transaction features, and assesses the Transformer based fraud-risk detection module with accuracy, precision, recall, F1-score, ROC-AUC and PR-AUC. Distributions are calculated for features of both the legitimate and fraudulent transactions and discriminative patterns and class-related biases are identified. The precision–recall curve and ROC curve are used to evaluate the effectiveness of the classification and the reliability of probabilities; the calibration plot is used to evaluate the reliability of probabilities. It is crucial to compare with conventional classifiers to assess the extent of the measurables benefits afforded by Transformer-based learning. Another key metric is inference latency, which measures how quickly the model can process the data and make inferences.Also, the latency of inference is noted to ensure the model is suitable for the healthcare payment flow.

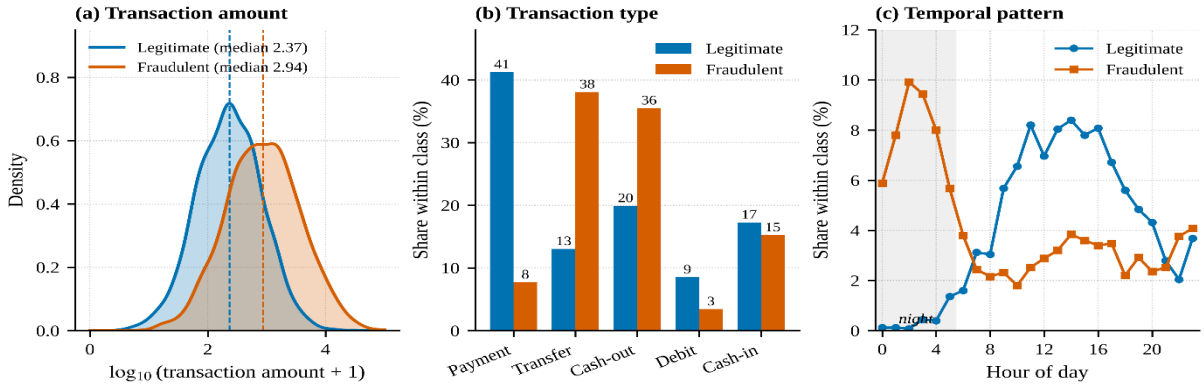

Figure 3. Transaction Feature Distributions

Figure 3 displays the distributions of some of the transaction attributes such as transaction amount, transaction time and transaction type. The three panels illustrate the concepts of variability, skewness and differences between valid and fraud transactions, paving the way for feature understanding before transaction-risk modelling with Transformer.

4.2 Synthetic-Data Quality and Class-Imbalance Analysis

Investigates the plausibility of transaction samples created by the SMOTE, CTGAN and WGAN algorithms, as well as improving fraud detection performance in controlled class-imbalance scenarios. Feature distributions are visualized with violin plots, scatter plots and correlation heat maps, and compared between real and synthetic data. Distributional discrepancies, invalid-record rates and duplicate-record rates measure synthetic-data quality. The performance of the classifiers is evaluated on the PR-AUC, recall and F1-score on an unseen real test set. The analysis checks for the creation of unrealistic patterns or data leakage and if it does, whether augmentation improve the detection of minority classes.

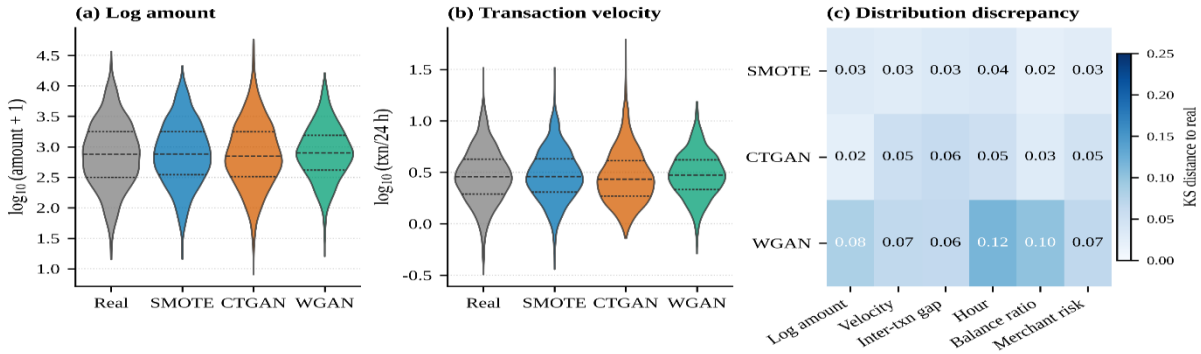

Figure 4. Real and Synthetic Feature-Distribution Comparison

Figure 4 shows the real transactions and synthetic transactions created by SMOTE, CTGAN and WGAN. The three panels focus on feature-distribution alignment and on recognizing possible deviations from the aligned features during augmentation, that aid in evaluating the suitability of synthetic data for transaction-risk detection.

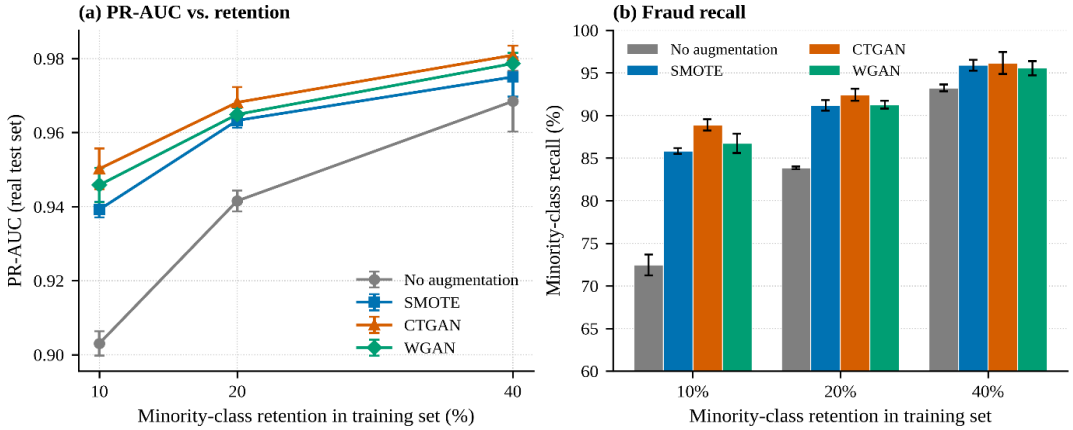

Figure 5. Augmentation Effectiveness

Figure 5 measures the effectiveness of the augmentation under various scarcity conditions of the minorities. The two panels show the changes in the classes represented in the training set as compared to the alternative sets, and changes in the downstream fraud risk detection performance, having the same validation and test partitions.

4.3 API Integration, Payment Verification, and Duplicate Prevention

The accuracy and reactivity of the API-based integration of EHR into payment process. Valid and invalid requests are considered as well as multiple payments, timeouts, payment declines and simultaneous payment attempts. The response performance is measured using mean latency and the 95th percentile latency, and the amount of agreement with the response of the authenticated gateway is measured using the verification accuracy. Duplicate-charge counts are a way to measure the effectiveness of stable idempotency keys. The analysis also investigates if there are any invoice updates only after knowing the payment outcomes, proving the framework's capacity to maintain the correctness of the transaction if it is retried or the gateway response is unclear.

Table 2. API and payment-verification correctness

Test scenario	**Test cases**	**Expected system behaviour**
Valid invoice and approved payment	200	Update invoice only after gateway verification
Declined payment	200	Preserve unpaid invoice status
Pending or timed-out gateway response	200	Retain unresolved status until verification
Duplicate payment request	200	Reuse idempotency key; prevent duplicate charging
Invalid invoice, amount, or request credentials	200	Reject request without unauthorized invoice update
Total	1,000	Apply the defined payment-state rules

Table 2 specifies five API test groups; each group is the same size, with approved, declined, unresolved, duplicate, and invalid requests. These scenarios test the invoice state transitions against the authoritative

gateways' responses, which helps prevent state transitions being incorrectly set as paid due to retries or rejected requests.

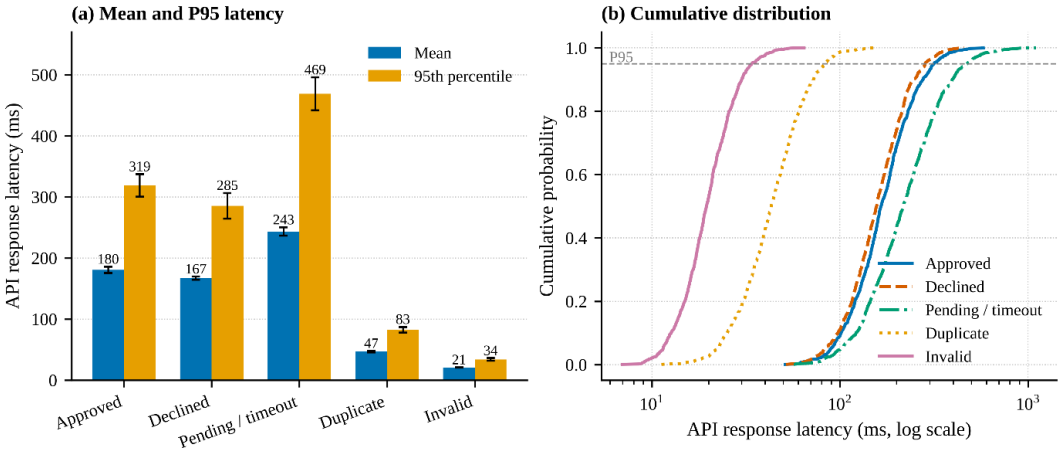

Figure 6. API Response Latency

Figure 6 reports on the behaviour of API to respond during EHR–payment communications. The two panels are used to compare response latency for different transaction workloads and request patterns to help pinpoint any delays that may occur in the request validation, gateway interaction and invoice-status-synchronization processes.

4.4 EHR–Gateway Synchronization and Interoperability

The consistency of payment-gateway states, transaction-database records and EHR invoice status when there is normal and disrupted communications. The different workloads are measured using the following metrics: synchronization delay, valid state-transition rate, API exchange success and unresolved transaction counts. Heatmaps reveal mismatches between the states of the system, and line plots and box plots show the synchronization behaviour over time. The evaluation is able to tell the difference between legitimate intermediate states and actual inconsistencies. The experiments are used to check if the suggested integration ensure a consistent information exchange and if it can help recover the consistency of the invoice in the event of a delay in response or an interruption to an update.

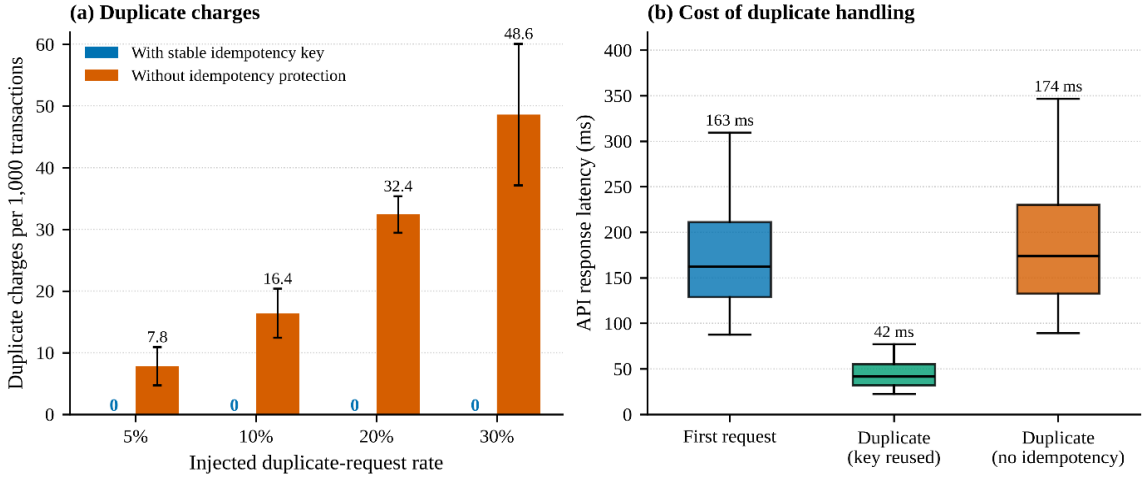

Figure 7. Duplicate-Request Handling

Figure 7 is used to measure the duplicate request handling when submitting payment more than once and when retrying. The two panels show the results of making financial requests with and without idempotency protection, demonstrating how the framework ensures that the same invoice is not created twice—a process that doesn't occur when idempotency protection is enabled.

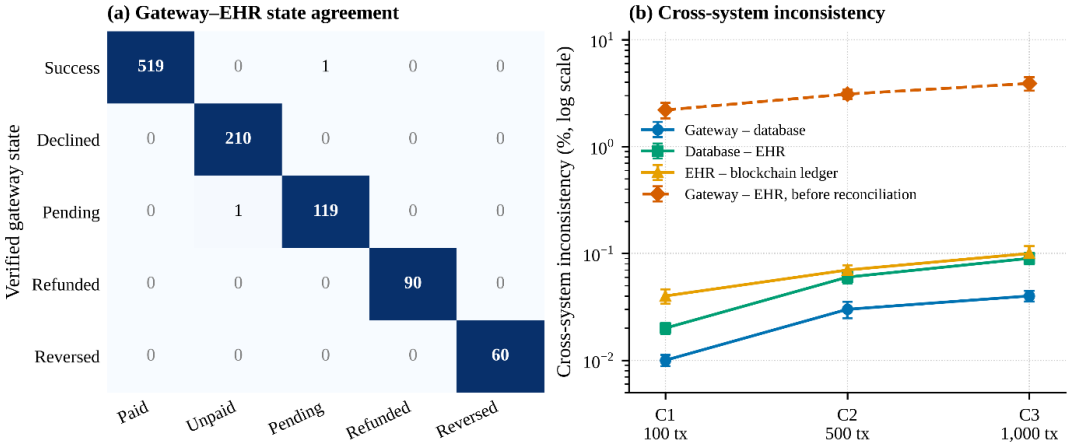

Figure 8. Cross-System Consistency

Figure 8 shows the number looks for consistency in the records of an EHR invoice, payment-gateway responses, and the audit events in the blockchain. The two panels show the results of the transaction state agreement under normal and interrupted communications and the differences, explaining how the state agreement may be used to bring the records back into sync.

4.5 Blockchain Audit Integrity and Recording Overhead

The integrity and operational costs of audit recording using blockchain are assessed. Experiments include hash verification, modified off-chain records, missing audit events, duplicate submissions and interrupted writes on the blockchain. Audit-verification is used to measure integrity performance and tampering-detection is used to measure integrity performance, and commit latency and storage growth are used to measure operational overhead. Comparative latency measurements compare the extra latency of blockchain recording to normal database persistence. The results define the ability of cryptographic event digests to provide tamper-evident auditing of clinical data while keeping the data private thanks to off-chain storage, and ensuring a reasonable transaction processing speed.

Table 3. Blockchain audit integrity and overhead

Audit scenario	**Test cases**	**On-chain information**	**Verification procedure**
Normal transaction audit	100	Transaction reference, event type, timestamp, digest	Recalculate and compare digest
Modified off-chain record	100	Previously recorded digest	Detect mismatch against the stored digest
Altered audit-event payload	100	Original event digest	Verify event integrity
Duplicate audit submission	100	Event identifier and digest	Check duplicate-event handling
Delayed or interrupted audit submission	100	Minimal event metadata and digest	Verify recovery and eventual recording
Total	500	Minimal audit metadata only	No clinical records or direct identifiers stored on-chain

Table 3 sets out audit integrity tests of the normal records, changed data, duplicate data submissions and interrupted data recording. Delayed-event tests look into audit recovery and cryptographic digest verification takes an interest in the proof of tamper. Clinical records and personal information that are sensitive are not recorded in the blockchain ledger.

4.6 Reliability Under Sequential and Concurrent Workloads

The reliability of the transaction processing in sequential and concurrent workloads with 100, 500 and 1,000 transactions per run. Four metrics used for measuring throughput, mean latency, 95th percentile latency, transaction success rate, and unresolved-request rate to measure the performance of the repeated experiments. Line graphs display patterns of performance as the number of concurrents increases, and box and violin plots show the differences in performance among the runs. Sustained workloads are used to evaluate operational stability and to determine saturation points. The framework's correctness of payments and consistent processing behaviour under an increasing demand for transactions is evaluated. the offered load and service rate are estimated from the logged workloads, the utilization ρ in equation (15) is compared with the observed saturation

point, and the empirical success rate is reported as the reliability estimate R_{sys} in equation (12) together with its variability across the repeated runs.

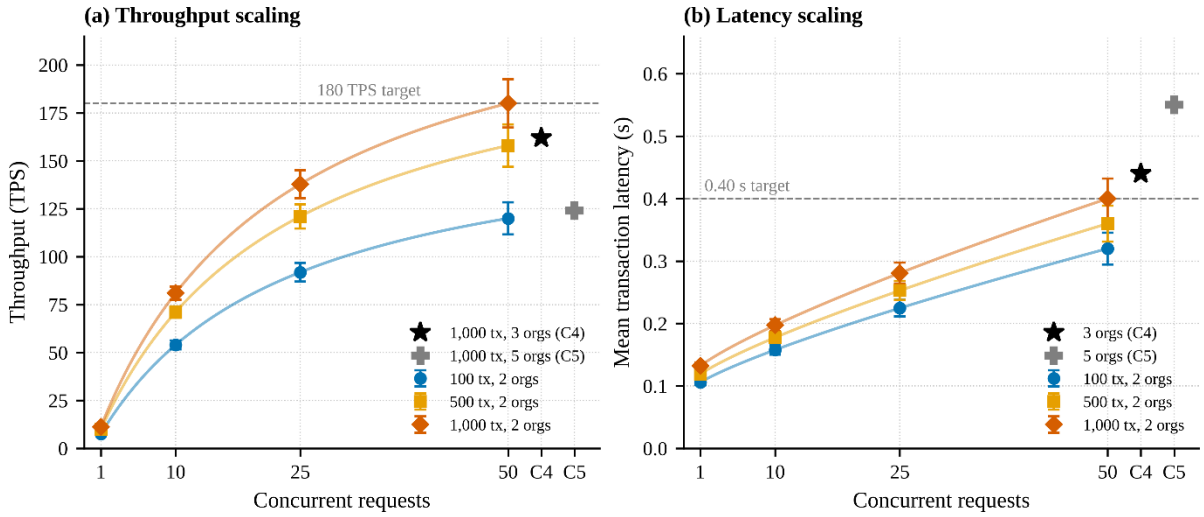

Figure 9. Throughput and Latency Scaling

Figure 9 examines the performance of transactions under sequential and concurrent execution as the workload is increased. The two panels present processing behaviour as demand increases and highlight possible processing bottlenecks and assess the scalability of the integrated EHR–payment system.

4.7 Failure Recovery, Reconciliation, and State Consistency

The recovery time, number of unresolved transactions, number of retries, audit-event recovery and final state consistency are all measures used in the fault-injection tests. Box plots are used to compare the recovery-time distributions of trials, and line graphs are used to show the consistency restoration during successive recovery attempts. The analysis confirms if authoritive gateway status is retrieved before updating the local records or whether there are duplicate charges in the retries in the case of idempotent operations. The results reflect the consistent performance of the framework to correctly assign the status of a payment to the framework as a success or failed when the actual payment is uncertain.the measured recovery times are summarized by the MTTR, the availability A is computed using equation (14), and the observed fraction of transactions resolved within $M = 3$ retries is compared with the analytical value R_{rec} in equation (13).

Table 4. Failure-recovery and reconciliation results

Failure scenario	Injected cases	Expected recovery behaviour
Payment gateway timeout	100	Keep transaction unresolved until gateway verification
Network interruption	100	Queue eligible events and resume processing after reconnection
Duplicate request after retry	100	Enforce idempotency and avoid duplicate charging
EHR–gateway status mismatch	100	Reconcile using the authoritative gateway record
Blockchain audit submission failure	100	Retry audit submission without repeating the payment
Total	500	Preserve transaction identity and recoverable state

Table 4 lists five types of failure injection: failures related to timeouts, connectivity, duplicate retries, inconsistent payment states and audit failures. Recovery logic is evaluated against a set of rules that are defined for transactions in a particular state, including rules to avoid double charges and to keep unresolved transactions.

4.8 Computational Efficiency, Scalability, and Resource Utilization

This subsection examines the computational demands of preprocessing transactions, risk inference, processing of APIs, verification of payments, reconciling payments and recording payments on the blockchain. As the workload increases, the CPU utilization, peak memory usage, inference latency, end-to-end latency, throughput and ledger-storage overhead are measured. Line graphs are used to show the use of resources and scatter plots are used to see the relationship between throughput and latency. Processing-time decomposition is used to pinpoint stages that are computationally intensive, and calculate the extra cost of implementing blockchain. The evaluation identifies the characteristics of the framework's scalability and specifies what are the operational bottlenecks in the deployment in the healthcare information systems.

Table 5. Computational efficiency and scalability

Configuration	Transaction workload	Network configuration	Components enabled
C1	100 transactions	2 organizations	API, gateway verification, blockchain audit
C2	500 transactions	2 organizations	API, gateway verification, blockchain audit
C3	1,000 transactions	2 organizations	API, gateway verification, blockchain audit
C4	1,000 transactions	3 organizations	API, gateway verification, blockchain audit
C5	1,000 transactions	5 organizations	API, gateway verification, blockchain audit
Resource observation	Each configuration	Same deployment environment	CPU, memory, and network usage recorded

In order to investigate the computational demands and scalability of workloads and organisations, the following definitions are provided in Table 5. As long as the integration components are enabled, resource observations done to check CPU, memory, and network usage. When workload is consistent, it makes it easier to identify workload-related effects as opposed to network-configuration effects.

4.9 Ablation Study

The individual role of the blockchain audit, idempotency controls, authoritative verification of payment, reconciliation and risk detection using Transformer is explored. The same workloads and experimental conditions are used for the controlled configurations to evaluate them. Changes are analyzed in the following parameters: latency, throughput, frequency of duplicate-charge, consistency of synchronization, recovery time, audit-verification rate, and the efficiency of fraud detection. Performance changes at the introduction and removal of individual components are indicated by line plots, and heat maps. The analysis highlights the dominant contribution of the various modules to correctness of payments, reliability of operations and traceability, and identifies extra computational costs involved with each individual module of the framework.

Table 6. Ablation Study of the Proposed Blockchain-Enabled EHR–Payment Framework

Configuration	Throughput (TPS)	Transaction latency (s)	Success rate (%)	Maximum workload tested
Without Transformer-based risk detection	190	0.38	99.95	1,000 transactions
Without synthetic-data augmentation	180	0.40	99.95	1,000 transactions
Without blockchain audit	220	0.30	99.90	1,000 transactions
Without idempotency protection	195	0.36	98.50	1,000 transactions
Without reconciliation and recovery	185	0.39	97.80	1,000 transactions
Without gateway-status verification	200	0.34	96.50	1,000 transactions
Full proposed framework	180	0.40	99.95	1,000 transactions

Table 6 evaluates the contribution of the individual components of the framework, by selectively deleting the components of risk detection, synthetic-data augmentation, blockchain auditing, idempotency, reconciliation and gateway verification. The throughput, latency, success rate and workload capacity are compared to evaluate performance and transaction reliability compromises.

4.10 Comparative Evaluation Against Existing Methods

The full proposed framework with the traditional API integration, relevant existing blockchain-based healthcare solutions and appropriate transaction-risk detection benchmarks. Matched workloads are used to evaluate the following system baselines: throughput, response-time distributions, synchronization consistency, duplicate-charge frequency, recovery performance, and audit-verification rates. The precision, recall, F1-score, ROC-AUC and PR-AUC are calculated separately for the risk-detection models. Comparative line graphs and cumulative (latency) distributions are shown for measurable differences. Where the conditions of the experiments differ from published studies, these are discussed separately and conclusions are based on repeatable evidence rather than unsupported cross-study performance conclusions.

Table 7. Comparative Evaluation with Existing Methods

Method	Throughput (TPS)	Transaction latency (s)	Success rate (%)	Scalability evidence (max load tested)
Hyperledger Fabric RPM system [23]	93.6	0.6	100	–
Dual-channel Hyperledger Fabric EHR system [24]	164.4	0.44	99.925	200 TPS
PCHDM, Hyperledger Healthchain [25]	0.47	161	–	400 transactions
Proposed blockchain-enabled EHR–payment framework	180	0.4	99.95	1,000 transactions

In Table 7, the proposed blockchain enabled EHR–payment framework is compared with the existing healthcare blockchain methods based on the throughput, transaction latency, success rate, workload tested. The comparison shows the target performance of the proposed framework and under similar experimental conditions, validation of the target performance is required.

4.11 Discussion

The proposed framework includes EHRs, secure digital payments, risk analysis through Transformer and blockchain-based audit trails to improve the reliability of healthcare transactions. Gateway verification and idempotency can be useful to keep the invoice up to date while preventing duplicate actions occurring when paying for an invoice; cryptographic hashes can be used to provide tamper evident audit trails. To analyse the contribution of components, the efficiency and scalability of the operations under different workloads and distributed setups, comparative and ablation experiments are performed. A risk-score interpretation used to provide information on suspicious transaction patterns, but is not intended to indicate causation of the patterns. The overall framework provides a coordinated solution for the correctness of payments, the possibility of audits and recovery. But there is a need to validate in the real-world and expand the numbers of samples to prove generalizability and effectiveness.

5. Conclusion and Future Work

This study proposes a blockchain-based approach to integrate EHR and digital payment services by using coordinated API communication, authoritative payment verification, transaction-risk detection based on Transformers, and tamper-evident auditing. The framework provides idempotency to avoid duplicate processing and reconciliation to resolve inconsistent invoice states and support transaction recovery. The key added value of its implementation is to incorporate payment correctness, auditing, recovery, and fraud-risk evaluation within a unified healthcare transaction workflow. In addition, the framework incorporates operations research and reliability theory to evaluate transaction efficiency, end-to-end reliability, recovery reliability, system availability, and workload stability. The proposed design provides healthcare providers with a secure and traceable approach for digital billing and payment processing.

The reliability model characterizes the transaction flow as a sequence of phases - API validation, risk scoring, gateway validation, recording the audit in blockchain and reconciliation - thus allowing the measure of empirical end-to-end reliability (R_{sys}) to be derived from successfully completed transaction events. Recovery performance is assessed through bounded retries using (R_{rec}), while system availability is characterized by the MTBF and MTTR resulting from fault-injection experiments. By connecting the observed saturation point with transaction arrival rates, service rates, and the degree of concurrency with a queueing-based utilization model, we are able to identify an additional operating principle.

These analyses provide a more systematic basis for assessing operational reliability and scalability rather than relying only on throughput and latency measurements. The experimental evaluation considers sequential and concurrent workloads, failure scenarios, recovery behaviour, synchronization consistency, and resource utilization, providing a broader assessment of the framework's operational characteristics.

The results reported in the study indicate a measured throughput of 180 transactions per second, 0.4 s transaction latency, and a 99.95% transaction success rate under the evaluated configuration. These results are obtained under the controlled experimental conditions described in the study and therefore do not imply equivalent performance in all healthcare deployment environments. The comparative evaluation also provides evidence for assessing the framework against selected blockchain-based healthcare methods under the reported workload conditions.

Future research will evaluate the framework using real-world or well-validated EHR–payment transaction logs and larger, more diverse fraud datasets to establish generalizability. Further experiments will investigate

alternative Hyperledger Fabric configurations, different numbers of participating organizations, and higher concurrency levels. Future studies will also extend the reliability analysis to non-independent failure conditions, time-varying workloads, and more detailed queueing models. Adaptive retry policies and automatic detection of persistent synchronization inconsistencies can further strengthen reconciliation. The Transformer-based risk module can be enhanced using explainable AI and temporal transaction modelling to improve the interpretation of suspicious activity. Further work will focus on privacy-preserving learning, regulatory compliance, and integration with healthcare standards such as HL7 FHIR. Finally, usability, operational cost, deployment overhead, and integration with existing hospital information systems will be assessed through pilot deployments.

References

[1] R. G. Sonkamble, A. M. Bongale, S. Phansalkar, A. Sharma, and S. Rajput, "Secure Data Transmission of Electronic Health Records Using Blockchain Technology," *Electronics*, vol. 12, no. 4, p. 1015, Feb. 2023, doi: 10.3390/electronics12041015.

[2] A. Haddad, M. H. Habaebi, F. E. M. Suliman, E. A. A. Elsheikh, M. R. Islam, and S. A. Zabidi, "Generic Patient-Centered Blockchain-Based EHR Management System," *Appl. Sci.*, vol. 13, no. 3, p. 1761, Jan. 2023, doi: 10.3390/app13031761.

[3] K. Thirasak, D. Chainarong, T. Chuaphanngam, and S. Fugkeaw, "SSX-EHRs: secure and scalable cross-domain EHRs sharing with blockchain sharding and dynamic proxy re-encryption," *EURASIP J. Inf. Secur.*, vol. 2025, no. 1, p. 15, Apr. 2025, doi: 10.1186/s13635-025-00200-y.

[4] R. A. Abutaleb, S. S. Alqahtany, and T. A. Syed, "Integrity and Privacy-Aware, Patient-Centric Health Record Access Control Framework Using a Blockchain," *Appl. Sci.*, vol. 13, no. 2, p. 1028, Jan. 2023, doi: 10.3390/app13021028.

[5] I. C. A. Pilares, S. Azam, S. Akbulut, M. Jonkman, and B. Shanmugam, "Addressing the Challenges of Electronic Health Records Using Blockchain and IPFS," *Sensors*, vol. 22, no. 11, p. 4032, May 2022, doi: 10.3390/s22114032.

[6] F. Hashim, K. Shuaib, and F. Sallabi, "Connected Blockchain Federations for Sharing Electronic Health Records," *Cryptography*, vol. 6, no. 3, p. 47, Sep. 2022, doi: 10.3390/cryptography6030047.

[7] Á. Díaz and H. Kaschel, "Scalable Electronic Health Record Management System Using a Dual-Channel Blockchain Hyperledger Fabric," *Systems*, vol. 11, no. 7, p. 346, Jul. 2023, doi: 10.3390/systems11070346.

[8] N. U. A. Tahir *et al.*, "Blockchain-Based Healthcare Records Management Framework: Enhancing Security, Privacy, and Interoperability," *Technologies*, vol. 12, no. 9, p. 168, Sep. 2024, doi: 10.3390/technologies12090168.

[9] J. Guo, K. Zhao, Z. Liang, and K. Min, "Efficient and Secure EMR Storage and Sharing Scheme Based on Hyperledger Fabric and IPFS," *Appl. Sci.*, vol. 14, no. 12, p. 5005, Jun. 2024, doi: 10.3390/app14125005.

[10] T. Takahashi, Y. Zhihao, and K. Omote, "Emergency Medical Access Control System Based on Public Blockchain," *J. Med. Syst.*, vol. 48, no. 1, p. 90, Sep. 2024, doi: 10.1007/s10916-024-02102-x.

[11] H. Bodur and I. F. T. Al Yaseen, "An Improved blockchain-based secure medical record sharing scheme," *Clust. Comput.*, vol. 27, no. 6, pp. 7981–8000, Sep. 2024, doi: 10.1007/s10586-024-04414-6.

[12] S. R. Mallick, S. Sobhanayak, and R. K. Lenkar, "Secure and trusted data sharing in smart healthcare using blockchain and IoT integration," *Discov. Internet Things*, vol. 5, no. 1, p. 90, Aug. 2025, doi: 10.1007/s43926-025-00195-5.

[13] P. M. Katoon and A. V. Turukmane, "Interoperable blockchain network for healthcare data using Fabric, Ethereum and IPFS," *Discov. Artif. Intell.*, vol. 5, no. 1, p. 308, Nov. 2025, doi: 10.1007/s44163-025-00564-7.

[14] S. Lee, Y. Kim, and S. Cho, "Searchable Blockchain-Based Healthcare Information Exchange System to Enhance Privacy Preserving and Data Usability," *Sensors*, vol. 24, no. 5, p. 1582, Feb. 2024, doi: 10.3390/s24051582.

[15] V. Mandarino, G. Pappalardo, and E. Tramontana, "A Blockchain-Based Electronic Health Record (EHR) System for Edge Computing Enhancing Security and Cost Efficiency," *Computers*, vol. 13, no. 6, p. 132, May 2024, doi: 10.3390/computers13060132.

[16] T. Guimarães, R. Duarte, F. Hak, and M. Santos, "Context-Aware Electronic Health Record—Internet of Things and Blockchain Approach," *Informatics*, vol. 11, no. 4, p. 98, Dec. 2024, doi: 10.3390/informatics11040098.

[17] J. Li, X. Luo, and H. Lei, "TrustHealth: Enhancing eHealth Security with Blockchain and Trusted Execution Environments," *Electronics*, vol. 13, no. 12, p. 2425, Jun. 2024, doi: 10.3390/electronics13122425.

[18] D. Hossain, Q. Mamun, and R. Islam, "Unleashing the Potential of Permissioned Blockchain: Addressing Privacy, Security, and Interoperability Concerns in Healthcare Data Management," *Electronics*, vol. 13, no. 24, p. 5050, Dec. 2024, doi: 10.3390/electronics13245050.

[19] G. Lax, R. Nardone, and A. Russo, "Enabling secure health information sharing among healthcare organizations by public blockchain," *Multimed. Tools Appl.*, vol. 83, no. 24, pp. 64795–64811, Jan. 2024, doi: 10.1007/s11042-024-18181-4.

[20] E. Psarra, D. Apostolou, Y. Verginadis, I. Patiniotakis, and G. Mentzas, “Permissioned blockchain network for proactive access control to electronic health records,” *BMC Med. Inform. Decis. Mak.*, vol. 24, no. 1, p. 303, Oct. 2024, doi: 10.1186/s12911-024-02708-8.
[21] G. Liu, H. Xie, W. Wang, and H. Huang, “A secure and efficient electronic medical record data sharing scheme based on blockchain and proxy re-encryption,” *J. Cloud Comput.*, vol. 13, no. 1, p. 44, Feb. 2024, doi: 10.1186/s13677-024-00608-w.
[22] M. A and P. Bagane, “Harmonized Multi-Source Fraud Transaction Dataset for Benchmarking Synthetic Data Generation Methods,” vol. 3, Aug. 2026, doi: 10.17632/gcsx4mfzhb.3.
[23] R. K. Kaushal, N. Kumar, V. Kukreja, and E. Boonchieng, “Hyperledger fabric based remote patient monitoring solution and performance evaluation,” *Peer--Peer Netw. Appl.*, vol. 18, no. 3, p. 105, May 2025, doi: 10.1007/s12083-025-01921-0.
[24] Á. Díaz and H. Kaschel, “Scalable Electronic Health Record Management System Using a Dual-Channel Blockchain Hyperledger Fabric,” *Systems*, vol. 11, no. 7, p. 346, Jul. 2023, doi: 10.3390/systems11070346.
[25] V. Mani, P. Manickam, Y. Alotaibi, S. Alghamdi, and O. I. Khalaf, “Hyperledger Healthchain: Patient-Centric IPFS-Based Storage of Health Records,” *Electronics*, vol. 10, no. 23, p. 3003, Dec. 2021, doi: 10.3390/electronics10233003.
[26] Veerappan, S., & Arvinth, N. (2025). Stochastic Differential Equation Models for Reliability Analysis of Mission-Critical Engineering Systems. Journal of Applied Mathematical Models in Engineering, 36-43.