Expert Systems With Applications 299 (2026) 129916

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

FLIT: Federated ledger and intelligence for trust-bootstrapping in the Internet of Medical Things

Debashis Das [a,*], Sourav Banerjee [b], Debashis De [c]

[a] Meharry Medical College, Nashville, 37203, USA
[b] Kalyani Government Engineering College, West Bengal, 741235, India
[c] Maulana Abul Kalam Azad University of Technology, West Bengal, 741249, India

ARTICLE INFO

Keywords:
Recommendation systems
Trust-weighted aggregation
Federated ledger
Privacy preservation
Smart contracts
Intelligent systems
Differential privacy

ABSTRACT

The rapid growth of the Internet of Medical Things (IoMT) has transformed how healthcare data is collected and analyzed to improve patient health through real-time monitoring and personalized treatment. However, there is a significant concern over data security, privacy, and trust among healthcare providers. To address these challenges, we present a federated ledger and intelligence framework named FLIT to provide secure and privacy-preserving data sharing across multiple healthcare stakeholders. The proposed method combines blockchain technology with federated learning (FL) to protect sensitive patient information. Herein, patient data is never shared in raw form and is encrypted and distributed among trusted stakeholders so that individual data points remain confidential and tamper-proof. Besides, access control is enforced using smart contracts that automatically grant or deny permissions based on predefined conditions. To enhance clinical decision-making, FL allows hospitals and healthcare providers to collaboratively train machine learning models without exchanging actual patient data. These models are then used to generate recommendations for patients' treatment plans, which can be shared across the network. We also implemented and tested the FLIT using smart contracts deployed on Ethereum environments and FL models trained on real-world healthcare datasets. Our experiment results evaluated several metrics, such as local and global model accuracy, training loss, trust score, and cost of smart contract operations across multiple Ethereum Virtual Machines. The proposed work achieves high model performance and preserves privacy by validating its robustness in real-world healthcare scenarios.

1. Introduction

There has been a massive buildup of confidential medical information in recent years due to the spread of electronic healthcare records and connected medical equipment. A major difficulty still lies in the secure and privacy-preserving use of this data, despite its enormous promise to enable predictive and personalized treatment (Barreau et al., 2008). Problems with compliance, legality, and single points of failure can affect patient records stored on centralized data-sharing platforms. Inefficient, slow, vulnerable, and providing little value to patients are the characteristics of outdated systems (Bhartiya & Mehrotra, 2014). According to new research, healthcare systems that are still in use today result in health data that is fragmented and isolated, making it difficult to interchange information because of different forms and standards. Basically, present stakeholders have real-time demands that are not being met by the current healthcare data since it is fragmented and unsuitable (Woodside, 2007). A digital revolution is happening in healthcare, though, as telemedicine, remote monitoring, and health technologies are becoming more popular (Stoumpos et al., 2023). Protecting patient data from cybersecurity risks has been a struggle for healthcare companies (Javaid et al., 2023). This is because cybersecurity measures have the potential to improve patient data's availability, integrity, and confidentiality (Kumar et al., 2023). The absence of openness in healthcare can impact many areas, including price, billing procedures, physician effectiveness, and patient choice (Sallam, 2023).

The recent methods of the Internet of Medical Things (IoMT) consist of a network of interconnected medical equipment and wearable sensors (Borozdina & Novkunskaya, 2022). Presently, electronic health records (EHR) are predominantly maintained in centralized systems, hence restricting their portability (Raj & Prakash, 2024). This centralized method heightens the danger of security violations and requires reliance on a singular authority (Bhushan et al., 2023). Conversely, peer-to-peer (P2P) data sharing (Das et al., 2021c) possesses significant value in healthcare and other sensitive data-sharing sectors. It provides numerous security advantages by eliminating single points of failure, as data is disseminated among trusted peers to enhance resilience against malicious

* Corresponding author.
E-mail addresses: debashis.das@mmc.edu (D. Das), sourav.banerjee@kgec.edu.in (S. Banerjee), debashis.de@makautwb.ac.in (D. De).

https://doi.org/10.1016/j.eswa.2025.129916
Received 5 August 2025; Received in revised form 10 September 2025; Accepted 28 September 2025
Available online 3 October 2025

attackers. These networks alleviate the hazards linked to centralized repositories susceptible to extensive breaches.

The Federated Learning (FL) architecture represents a crucial advancement in predictive modeling within healthcare (Wang et al., 2023), with important implications for enhancing patient care. Model sharing and collaborative training across decentralized data sources are constrained, and the management of patient consent is complex. The incorporation of blockchain (Das et al., 2020) into federated learning (FL) (Meenigea & Kolla, 2023) resolves the previously identified data-sharing challenges. Decentralizing data storage and employing encryption can improve data security in blockchain, thereby complicating the efforts of attackers to compromise patient records (Das et al., 2023). Through smart contracts and consent management mechanisms, patients can retain control over their health data and grant access to their data. We use FL for better data privacy and personalized healthcare recommendations (Das et al., 2025), which can be shared directly among peers to protect patient data (Ray et al., 2019). This mix of blockchain and FL can help to protect healthcare data, defend against cyberattacks, and create a safer and more privacy-preserving healthcare ecosystem (Singh et al., 2021).

We introduce FLIT, a federated ledger and intelligence approach for secure and privacy-preserving healthcare data sharing to boost trust among stakeholders. The main contribution of the work is given below:

- We present a blockchain-enabled, secure, and privacy-preserving framework named FLIT to protect healthcare communication, ensure patient privacy, and safeguard data security within the IoMT networks.
- We incorporate FL into the FLIT framework to facilitate the generation of personalized healthcare recommendations, which are securely disseminated across a peer-to-peer network.
- We create strong data encryption and secure distribution methods to maintain the utmost confidentiality and integrity of patient data in decentralized healthcare systems.
- We design a multi-threat defense mechanism to protect the proposed FLIT framework against inference attacks and rollback manipulation.
- We expand the capabilities of the FLIT with FL-enabled P2P data sharing for scalable and trustworthy collaborative intelligence.
- We provide experimental validation of smart contract mechanisms for securely sharing FL-enabled recommendation data and evaluating the performance of the underlying federated models.

In order to give a structured summary of our findings, this paper is divided into multiple important sections. Reviewing recent progress in blockchain-based healthcare and FL, the study starts with Section 2. The communication model and FLIT system architecture are described in Section 3. Details on the implementation, such as the logic of smart contracts and the roles of stakeholders, are addressed in Section 4. In Section 5, we look at how FL improves the privacy of data in FLIT. Formal mitigation strategies are introduced in Section 6 for attacker models. The experimental results that assess performance, security, and cost are detailed in Section 7. Lastly, the section ends with important points to remember and suggestions for where the research should go from here.

2. Literature review

2.1. Healthcare challenges of today

The healthcare sector is experiencing a digital transition characterized by the heightened utilization of telemedicine, remote monitoring, and digital health technologies. The epidemic expedited the adoption of virtual care modalities for consultations, monitoring, and follow-up care conducted remotely. Healthcare firms encounter the problem of protecting patient information from cybersecurity attacks while ensuring compliance with data protection rules (Javaid et al., 2023). Table 1 presents a comparison of several data-sharing strategies against other established techniques.

Security Risks to Sharing Patient Data. In the healthcare sector, protecting patient data during transmission is imperative. The stakes are exceedingly high, as this data includes an individual's most confidential medical history, treatments, and personal information. However, the healthcare sector confronts numerous security threats that jeopardize the integrity of this information (Kumar et al., 2023). Cyber dangers and unauthorized access jeopardize patient confidentiality, as well as data integrity and availability.

Data-driven Risk Assessment. Data-driven decision-making involves utilizing data and analytics to direct and inform the decision-making process (Shonchoy et al., 2023). Through the analysis and interpretation of pertinent data, companies may make educated decisions and enhance their plans for improved results. Although data-driven decision-making offers several advantages, it also entails possible hazards linked to exclusive reliance on data, which may jeopardize human life and safety, including the danger of fatalities (Mallick et al., 2024).

Fragmented Health Services. In fragmented healthcare systems, patients frequently encounter disjointed care, resulting in medical errors, delayed diagnoses, and superfluous expenses (Cerchione et al., 2023). The absence of cooperation among healthcare providers jeopardizes patient safety and leads to redundant testing. To improve patient outcomes, decrease healthcare costs, and promote patient satisfaction, tackling fragmentation in healthcare systems is critical. Healthcare systems can provide more efficient, patient-focused treatment and attain superior overall outcomes.

Lack of Transparency. In the absence of clear and accessible information, individuals frequently remain uninformed about essential elements of their healthcare, such as prices, quality, and provider performance (Sallam, 2023). This opacity may result in unexpected medical expenses, uneducated choices, and a feeling of helplessness in maneuvering through the healthcare system. Healthcare expenses might increase without justification.

Insurance Fraud. Insurance fraud perpetrated by individuals or healthcare professionals increases healthcare expenses for all. This fraudulent activity adversely affects insurance firms and imposes a financial burden on honest policyholders. Furthermore, insurance fraud can redirect resources from legitimate healthcare requirements, erode trust in the healthcare system, and jeopardize patient treatment. Through the identification and prevention of fraud, the healthcare sector may optimize resource allocation, manage expenses, and uphold the confidence of its stakeholders.

2.2. P2P Data sharing

P2P data sharing is a decentralized approach for the direct exchange of information between individual users or devices within a network (Cangir et al., 2021). This method deviates from the conventional client-server architecture, wherein a central server oversees data and connections. In P2P networks, data is disseminated across all connected peers (Das et al., 2021c). The decentralized structure improves resilience and fault tolerance by eliminating a single point of failure. P2P communication demonstrates secure, bidirectional data exchanges between users, characterized by efficient, autonomous, and distributed information flow. It fosters direct communication among peers in efficient data sharing, less latency, and enhanced data transmission speeds (Das et al., 2021b).

Requirements for P2P data sharing system. An effective and secure P2P data-sharing system (Tigelaar et al., 2012) is founded on stringent criteria that encompass numerous critical aspects of security and functionality. These standards guarantee that the system not only efficiently enables data sharing but also protects the confidentiality, integrity, and privacy of the shared information. The fundamental principle of every data-sharing system is the necessity of data protection. Employing encryption protocols for data both in transit and at rest guarantees the

Table 1
Comparison analysis of several existing methods with our proposed method.

Method	Year	Description	Pros	Cons
FL (Xu et al., 2021)	2021	Employing FL techniques for decentralized model training across healthcare data sources.	Preserves data privacy.	Complex model aggregation.
Distributed Ledger Technology (DLT) (Murugan et al., 2020)	2020	Exploring distributed ledger technologies like Hashgraph for secure healthcare data sharing.	High data integrity.	May require specialized expertise.
Decentralized Identity (Javed et al., 2021)	2021	Utilizing decentralized identity solutions to manage and control access to healthcare data securely.	Provides user-centric control.	Adoption and integration challenges.
Secure Multi-Party Computation (MPC) (Dong et al., 2021)	2022	Performing computations on encrypted data while preserving data privacy.	Preserves data privacy during analysis.	Computationally intensive.
Zero-Knowledge Proofs (Jedlicka & Grant, 2022)	2022	Proving data authenticity without revealing sensitive information.	Strong data privacy guarantees.	Computational overhead.
Homomorphic Encryption (Sendhil & Amuthan, 2021)	2000	Enabling computations on encrypted healthcare data while preserving privacy.	Privacy-preserving analysis.	Computational complexity.
Decentralized Applications (DApps) (Panigrahi et al., 2022)	2001	Developing DApps on blockchain platforms for peer-to-peer healthcare data sharing.	Transparency and security.	Limited scalability.
Consortium Blockchains (Du et al., 2020)	2000	Consortium blockchains where trusted entities collaborate on data management.	Enhanced trust among stakeholders.	Requires trust among members.
Privacy-Preserving AI (Khalid et al., 2023)	2001	AI models designed to operate on encrypted data for secure analysis.	Enables private analysis.	Complex model training.
Secure Data Sharing Protocols (Masud & Hossain, 2018)	2000	Protocols specifically designed for healthcare data sharing across stakeholders.	Tailored to healthcare requirements.	Adoption challenges.
Proposed FLIT	2023	Blockchain-enabled FL for secure and privacy-preserving recommended healthcare data sharing.	Provides data security; privacy-preserving recommended data sharing.	–

confidentiality of sensitive information (Draidi et al., 2011). Access controls provide meticulous regulation of data access, thwarting unlawful intrusions. Data integrity mechanisms ensure that the shared data remains unmodified and reliable (Venugopal et al., 2006). This enhances user agency and fosters confidence in the system. Decentralization is a fundamental principle of peer-to-peer systems. It diminishes dependence on central authorities and eliminates single points of failure (Bonifati et al., 2008).

Blockchain in P2P Data Sharing. Blockchain technology is transforming peer-to-peer data sharing by including trust, security, and transparency into the process (Kaveh et al., 2025). In conventional P2P data exchange, participants frequently encounter issues concerning trust and data integrity. Blockchain resolves these difficulties by offering a decentralized, immutable ledger in which data exchanges are securely documented. It obviates the necessity for intermediaries or central authorities. Smart contracts, which are self-executing agreements, can automate data-sharing arrangements, guaranteeing that data is accessed solely under predetermined conditions.

Smart Contracts in P2P Data Sharing. Smart contracts are essential for facilitating efficient and safe data sharing in peer-to-peer networks. Self-executing contracts are implemented on blockchain systems, functioning as automated middlemen that execute the stipulations of data-sharing agreements without requiring a central authority. In a P2P data-sharing framework, smart contracts may automate and enforce data-sharing agreements, so preserving data integrity and positioning themselves as essential elements in the advancement of secure and efficient P2P data-sharing ecosystems. In healthcare, innovative breakthroughs encompass the application of blockchain (Regueiro et al., 2021) and AI technology, particularly through FL integration (Dey et al., 2025). This section elucidates current methods utilizing blockchain and AI for data exchange, while also addressing the integration of both technologies to effectuate substantial improvements in healthcare.

Related Works The Internet of Things (IoT) is rapidly expanding in both benefits and challenges. For instance, the implementation of advanced machine learning in a centralized manner is unfeasible due to the substantial volume of data and privacy issues. To tackle this issue, FL presents a viable solution by maintaining data privacy through retention on end devices. The authors Ali et al. (2021) initiated new research by examining blockchain in IoT to tackle privacy issues, introducing FL, presenting use cases, and proposing blockchain-based traceability. Furthermore, model tampering continues to be a worry. To address these issues, the authors Chen et al. (2023) developed PPTFL, a framework for privacy-preserving and traceable FL. It integrates Hierarchical Aggregation FL (HAFL) for minimal overhead privacy safeguarding in industrial IoT (IIoT) environments, together with FL utilizing blockchain and IPFS to ensure parameter traceability and immutability.

In our progressively technology-oriented society, data security, particularly in the transmission of sensitive medical information, is of utmost significance. The authors Javed et al. (2023) presented a resilient architecture employing blockchain, local differential privacy, and FL learning for secure data exchange. This paradigm obviates the necessity for trust between data proprietors and controllers. The authors Lu et al. (2020) proposed a blockchain-based framework for secure data sharing across various entities in the IIoT. They employed FL, which safeguards privacy, to maintain data confidentiality while disseminating data models. They also developed FL into a permissioned blockchain consensus to facilitate rapid and secure sharing.

In Yang et al. (2022), the authors delineated a method for the secure sharing of credit data and models within a distributed framework. They prioritized data privacy by disseminating models rather than raw data, employed a deletable Bloom filter to enhance data storage and consensus in FL, and implemented authority control and credit verification contracts to ensure the secure sharing of credit information. The experiments and security analysis of the paper validate that the system

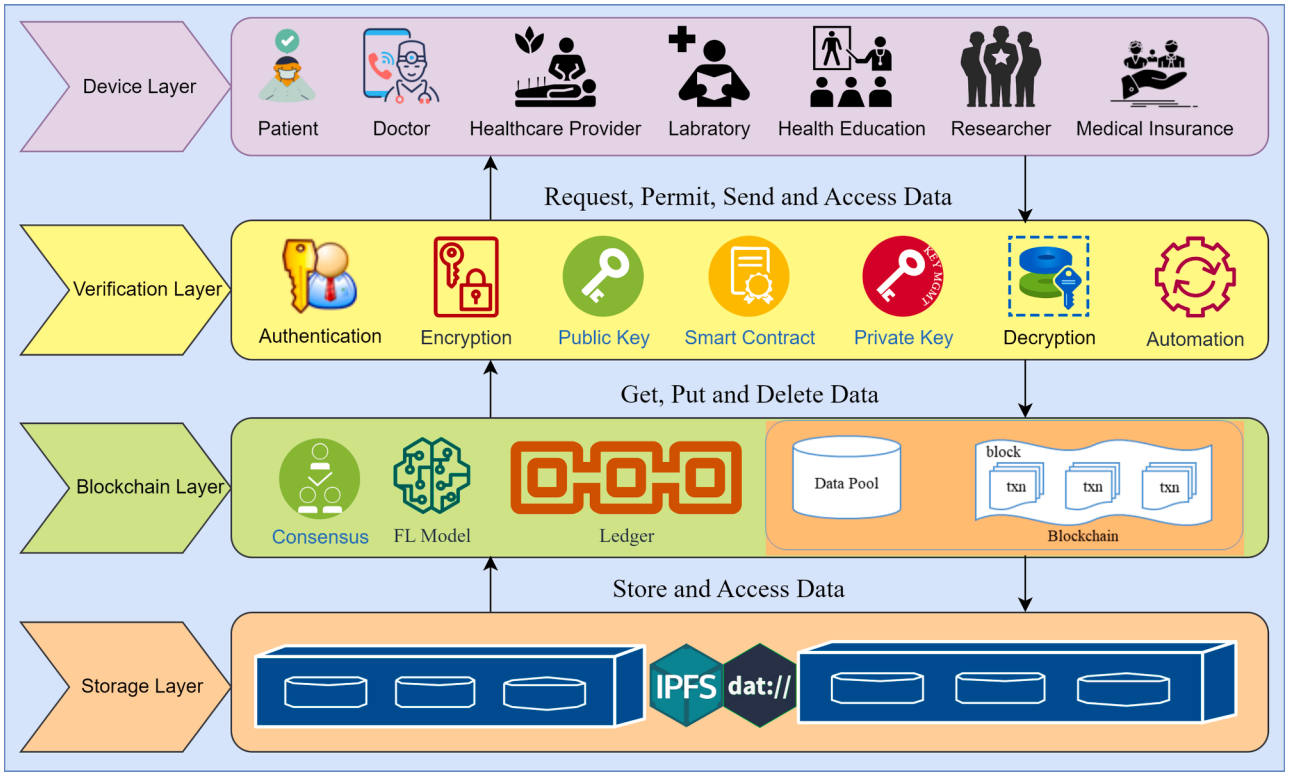


Fig. 1. Proposed FLIT system model.

is very accurate, efficient, and stable. Nonetheless, challenges related to scalability and potential adversarial assaults may require further investigation. Ahmed et al.Vyas et al. (2023) examined the digitization of healthcare and EHRs, which have proliferated over the past decade. Nonetheless, these systems are susceptible to data breaches. Blockchain technology possesses disruptive potential by improving data management and access control within the healthcare sector. This research examines the amalgamation of blockchain, artificial intelligence, and the Internet of Things in the healthcare sector. Mistry et al. (2021) presented an AI-enabled, blockchain-driven electronic health record system that employs neural networks to categorize individuals as possibly COVID-19 positive or negative. Patients' positive data is recorded on the blockchain with IPFS.

3. Proposed FLIT system model

This section defines the entities involved in the FLIT framework and lays out the core system assumptions under which our architecture operates. Fig. 1 shows the proposed system model. The proposed FLIT shares medical information and unites various stakeholders such as patients, hospitals, insurance companies, government authorities, practitioners, researchers, and medical stores. FLIT fosters a patient-centric, secure, and interconnected healthcare ecosystem. The notations and symbols used in this process are summarized in Table 2.

3.1. Stakeholder roles in the FLIT healthcare ecosystem

The proposed system can play a crucial role in enhancing communication and collaboration among various stakeholders in the healthcare ecosystem. The main role of each stakeholder is described below:

Patients can securely access and own their medical records, including test results, prescriptions, and treatment history, through the blockchain. They have greater control over their data and can grant or revoke access to healthcare providers as needed. Patient data is encrypted, and access is protected through cryptographic techniques. Beyond access control, patients also benefit from improved interoperability, where their data can be seamlessly transferred across institutions without duplication. In addition, FLIT empowers patients by enabling them to participate in federated training processes, contributing to global models while keeping their data private.

Hospitals can securely share patient data with other authorized healthcare providers. Blockchain ensures the immutability and transparency of medical records to reduce errors, fraud, or tampering. Hospitals can contribute to identifying data for research initiatives. Researchers can access aggregated and anonymized data for studies. Furthermore, hospitals benefit from reduced administrative burden since audit trails and compliance logs are automatically maintained on the blockchain. Hospitals also play a central role in hosting local training nodes so that models trained on institutional datasets contribute to improved healthcare outcomes without violating privacy regulations.

Insurance Companies can access verified medical records on the blockchain, reducing administrative burdens, fraud, and errors during claim processing. Blockchain-based AI systems can look for patterns and spot fraud, such as making duplicate claims or using fake documents. This makes the process of finding fraud more accurate and quicker. In addition, smart contracts can automate claim approvals and settlements as well as disputes and speed up reimbursements. Insurance providers can also use aggregated insights from federated models to better assess population-level health risks and design personalized insurance products.

Government Authorities can monitor public health trends and outbreaks more effectively by securely accessing aggregated and anonymized data from healthcare providers on the blockchain. Blockchain-based systems can enforce compliance with healthcare regulations, ensuring data privacy, security, and interoperability. Governments can also use blockchain audit trails to conduct real-time monitoring of healthcare delivery quality and detect anomalies in reporting. In addition, FLIT enables policymakers to support nationwide FL initiatives for equitable use of patient data and maintaining strict privacy and ethical boundaries.

Table 2
List of symbols and their definitions.

Symbol	Definition	Symbol	Definition
S_i	Stakeholder (e.g., hospital, clinic, lab)	R	Set of registered stakeholders
$V(S_i)$	Identity verification function for S_i	K_i	Symmetric encryption key for S_i
M_j	Medical record with ID j	γ_j	Content/data of medical record M_j
τ_j	Timestamp associated with M_j	S_{ij}	Owner of medical record M_j
$E(M_j, K)$	Encrypted version of M_j using key K	$T(\cdot)$	Blockchain transaction function
$\mathfrak{R}$	Blockchain storage or ledger reference	P_k	Access permission object
S_i^A	Address of the stakeholder granted access	$\wp_k$	Access level defined in P_k
$G(P_k, S_i^A)$	Access grant function	$AC(\cdot)$	Access control function
$R(P_k)$	Revocation function for permission P_k	AC_{update}	Updated access control after revocation
S_s	Stakeholder requesting record access	`RetrieveRecord`	Medical record retrieval function
`CheckPermission`(S_s)	Permission verification for stakeholder S_s	`Decrypt`$(\cdot)$	Decryption function for encrypted record
$\theta_t^{(i)}$	Local model parameters of S_i at round t	$\theta_{t+1}^{(i)}$	Updated local model after round t
$\Delta\theta^{(S_i)}$	Model update from stakeholder S_i	L_i	Local loss function on data D_i
$\nabla_\theta L_i$	Gradient of local loss function	η	Learning rate
D_i	Local dataset at stakeholder S_i	n_i	Number of samples in D_i
x_j	Input feature vector	y_j	Output label for x_j
$\tilde{\nabla}_\theta$	Perturbed gradient (DP-applied)	$\mathcal{N}(0, \sigma^2)$	Gaussian noise for differential privacy
ϵ, δ	Differential privacy budget parameters	$\delta^{(S_i)}$	Encrypted model update
M_i	Metadata log of model update	`Hash`$(\cdot)$	Cryptographic hash function
T_i	Blockchain transaction log	B	Blockchain ledger
M_{global}	Global model after aggregation	$A(\cdot)$	Model aggregation function
E_i	Evaluation result for S_i	D_i^{test}	Test dataset used for evaluation

Practitioners can access patient records and utilize FL-enabled recommendations for more accurate diagnoses, personalized treatment plans, and better patient outcomes. Blockchain facilitates secure communication and collaboration among healthcare providers. Practitioners also benefit from reduced duplication of diagnostic tests, since immutable records prevent redundant procedures across different facilities. Moreover, practitioners can rely on global federated models that are continuously updated with data from multiple institutions for better predictive accuracy in clinical decision-making.

Researchers can access anonymized patient data stored on the blockchain. Blockchain ensures the integrity and traceability of research data, promoting transparency, reproducibility, and trust in scientific findings. Blockchain-based systems can incentivize patients to contribute their health data for research purposes through tokenized rewards or participation in clinical trials. By leveraging FLIT, researchers gain access to diverse, cross-institutional datasets without violating privacy. This leads to faster discovery of treatment strategies, early disease detection models, and novel healthcare interventions grounded in secure, ethically sourced data.

3.2. FLIT System entities and their roles

Data Providers. Healthcare institutions, including hospitals, diagnostic labs, and clinics, serve as the primary data providers in the FLIT framework. They hold sensitive patient information such as medical records, diagnostic results, and treatment histories, which cannot be centralized due to strict privacy regulations like HIPAA and GDPR. To protect this data, providers train machine learning models locally on their own datasets and never transmit raw information outside their premises. Their local training not only safeguards privacy but also leverages diverse, real-world datasets across institutions, which enhances the robustness and generalizability of the global model. Instead, they contribute to the FL process by generating encrypted model updates or gradients, often enhanced with differential privacy to prevent leakage of individual patient details. These encrypted updates are then submitted to the blockchain through smart contracts, which validate and securely record each contribution. This ensures that all updates are transparent, tamper-proof, and auditable. In this way, FLIT transforms healthcare providers into active stakeholders in a decentralized ecosystem so that predictive models are both trustworthy and representative of heterogeneous IoMT environments.

Model Aggregator. In FLIT, the traditional central aggregator is replaced by a blockchain-coordinated process that eliminates single points of failure and central trust dependencies. The aggregator's primary task is to collect encrypted and privacy-preserving model updates from multiple data providers and combine them into a global model. Unlike conventional centralized schemes, aggregation in FLIT is performed off-chain, which reduces the computational burden on the blockchain and minimizes transaction latency. To preserve privacy and integrity, the aggregator employs cryptographic protocols such as Secure Multi-Party Computation (SMPC) and differential privacy during the aggregation process. These techniques ensure that no single entity can reconstruct sensitive local datasets from model updates. Once updates are decrypted by authorized entities, the model differences $\Delta\theta^{(i)}$ are combined using federated averaging (FedAvg) or weighted FedAvg, where weights n_i correspond to the size of each participant's dataset. This guarantees fairness and improves the representativeness of the global model in heterogeneous IoMT environments. After aggregation, a cryptographic hash of the global model is published back to the blockchain. In this way, the model aggregator not only consolidates distributed intelligence but also ensures that the resulting global model is secure, fair, and efficient under real-world IoMT constraints.

Smart contracts. Smart contracts in FLIT are unchangeable programs deployed on the Ethereum blockchain that serve as the backbone of system coordination and security. They ensure that only authenticated and authorized participants can submit model updates, automatically verifying the correctness and timeliness of each submission. Beyond record-keeping, smart contracts establish a transparent and tamper-proof audit trail by securely storing metadata and aggregation results on-chain. This guarantees accountability and prevents disputes over data contributions or model integrity. Smart contracts also regulate the timing of aggregation cycles so that updates from different participants are synchronized and processed fairly. This coordination is especially critical in heterogeneous IoMT environments, where devices may operate under varying connectivity and resource constraints. Smart contracts remove the need for centralized control and reduce the risk of manipulation or human error by enforcing predefined rules automatically. Overall, the inclusion of smart contracts strengthens FLIT by providing decentralization, automation, transparency, and trust. They reduce administrative overhead, prevent malicious behaviors, and ensure that every FL round is both verifiable and auditable for real-world healthcare applications.

Validators. Validators in FLIT are blockchain nodes that participate in consensus protocols. Their primary role is to confirm transactions

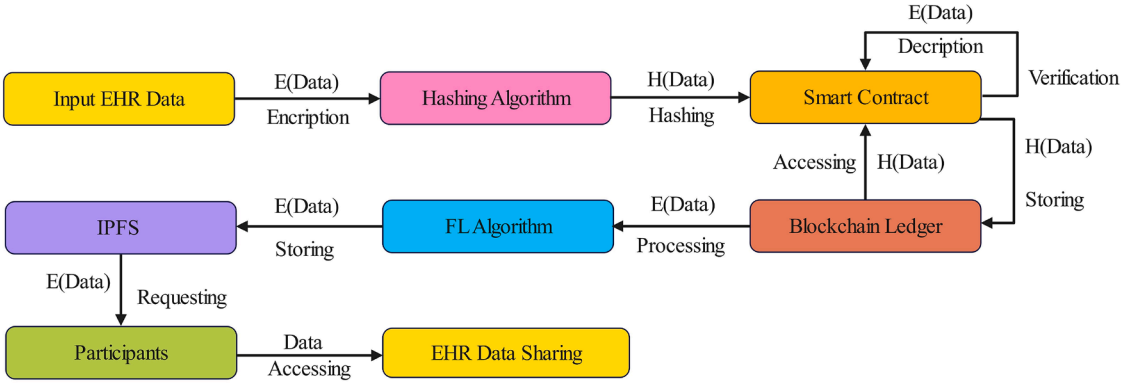


Fig. 2. Process diagram of securing medical data.

generated by smart contracts, finalize model update submissions, and ensure that no duplicate or malicious updates are accepted into the ledger. By distributing validation across multiple nodes, the system preserves decentralization while guaranteeing that each FL round is both auditable and tamper-resistant. To address scalability in real-world IoMT environments, lightweight consensus protocols can be employed to reduce confirmation latency and energy consumption so that the validation process does not become a bottleneck. This design choice enables FLIT to handle high volumes of transactions efficiently, even when numerous IoMT devices submit updates simultaneously.

Medical Stores. Medical stores in the FLIT framework play a crucial role so that prescribed medicines are dispensed accurately and safely. They can verify the authenticity of digital prescriptions issued by authorized healthcare providers and cross-check them against the immutable patient medication history stored in the ledger. This reduces the risk of medication errors, such as over-dispensing, harmful drug interactions, or fraudulent prescriptions. In addition, blockchain integration enhances the transparency and traceability of pharmaceutical supply chains. Each transaction from manufacturer to distributor to retail pharmacy can be securely logged on-chain record that prevents counterfeit drugs from entering circulation. Smart contracts further automate verification by validating prescription signatures. Fig. 2 illustrates the workflow diagram of this secure data communication process and describes how medical stores leverage blockchain to access reliable information and contribute to the integrity and accountability of the broader healthcare ecosystem.

3.3. System assumptions in FLIT

To guarantee both practical feasibility and provable security, the FLIT framework adopts a set of carefully defined assumptions about its components and operating environment. These assumptions form the foundation for designing protocols that are secure under semi-honest or partially adversarial conditions.

Semi-Trusted Participants. Each participant $S_i \in \mathcal{HBC}$ (Honest-But-Curious) is assumed to execute the protocol correctly without deviation. However, such a participant may attempt to extract sensitive information from intermediate data, such as gradients δ_i to infer private contents of the local dataset D_i. To mitigate this inference risk, FLIT enforces the following measures: Each S_i must undergo identity verification via a smart contract-based registry: $\texttt{verify}(S_i) \Rightarrow \texttt{True}$. And model updates are transformed via secure mechanisms before being submitted:

$$\delta_i^{\text{secure}} = \text{Encrypt}(\text{DP}(\delta_i), K_i) \tag{1}$$

where $\text{DP}(\delta_i) = \delta_i + \mathcal{N}(0, \sigma^2)$ applies differential privacy and K_i is a client-specific symmetric key.

Immutable Ledger via Blockchain. We assume that the blockchain ledger $\mathcal{B}$ is tamper-proof and maintains integrity of all transactions, update hashes, and event logs. Specifically, for any transaction T_i: $\forall i,\ \text{Hash}(T_i) \in \mathcal{B} \Rightarrow$ Immutable. This immutability is maintained through a consensus mechanism, where a group of trusted validator nodes $\{V_1, \ldots, V_n\}$ co-sign blocks:

$$\sum_{k=1}^{K} \text{Sign}_k(B) \geq \text{Threshold} \tag{2}$$

The sum checks whether enough trusted parties have signed the block to achieve consensus (used in PoA, BFT, or multi-signature schemes). In Eq. (2), $\text{Sign}_k(B)$ denotes the digital signature of the k-th validator node V_k on the block B. The variable K represents the total number of authorized validator nodes that participate in the consensus process and sign blocks. The value Threshold specifies the minimum number of validator signatures required to accept the block B as valid typically defined as a majority or predefined quorum (e.g., 2 out of 3, or greater than 50 % of K).

Localized Federated Training. Federated model training occurs locally at each provider. Gradients are computed via local data as:

$$\delta_i = \nabla_\theta \mathcal{L}(\theta_t, D_i) \tag{3}$$

However, raw data D_i is never shared: $D_i \nrightarrow \mathcal{B}, \quad D_i \nrightarrow A$, where A represents the aggregator or any external observer. This design is compliant with regulations like HIPAA.

Secure Update Communication Channel. All communication of model updates is secured using encryption and TLS, and stored references are anchored on-chain or in IPFS. All model updates δ_i are transmitted over encrypted channels. The pipeline is $\delta_i \rightarrow$ TLS $\rightarrow$ Blockchain Layer $\rightarrow$ IPFS/Storage. Specifically, updates are encrypted using client-specific keys, sent over TLS-secured channels, and referenced on-chain using content-based hashes $\text{Hash}(\delta_i^{\text{enc}}) \in \mathcal{B}$

Bounded Adversarial Collusion. FLIT assumes a bounded Byzantine threat model where only a small fraction f of participants are adversarial. This is upper-bounded by $f < f_{\max} = \left\lfloor \frac{N-1}{3} \right\rfloor$. To tolerate this, FLIT uses Byzantine-resilient aggregation strategies (e.g., trust-weighted mean or coordinate-wise median), ensuring convergence and robustness even in the presence of faulty nodes $\theta_{t+1} = \theta_t + \text{Agg}(\delta_1, \delta_2, \ldots, \delta_N)$, where Agg is robust such that $\forall f < f_{\max},$ loss decrease $\Rightarrow$ True. Overall, these assumptions define the trust boundaries and security guarantees of the FLIT framework.

4. Proposed FLIT implementation methodology

This section provides a detailed understanding of the various processes involved in implementing the FLIT framework. The smart contract handles stakeholder registration, data storage, permission management, and interaction with the blockchain, which are described in detail:

4.1. Stakeholder registration and verification

Let R = {S_1, S_2, …, S_n} be the set of stakeholders registering in the system, where n is the total number of stakeholders. Each stakeholder S_i provides their registration information, including name, contact details, and organization affiliation. A verification function V(S_i) is defined to validate the identity and authenticity of each stakeholder S_i. This function may involve verifying documents, performing KYC checks, or integrating with external identity verification services. A smart contract is responsible for validating stakeholders by using their account addresses.

4.2. Medical record storage and encryption

Let M = {M_1, M_2, …, M_m} represent the set of medical records stored in the system, where m is the total number of records. Each medical record M_j contains attributes as shown below:

$$M_j = (j, \gamma_j, \tau_j, S_{ij}) \tag{4}$$

The encrypted record is stored on the blockchain in a transaction denoted as $T(E(M^j_{S_i}, K), \Re)$. The detailed procedure for secure addition of medical records by an authorized stakeholder is provided in Algorithm 1, which ensures only verified users can upload non-empty medical data. The overall process of adding medical records S_i can be represented by the equation:

$$\text{AddRecord}(M_j, S_i, K) = T(E(M^j_{S_i}, K), \Re) \tag{5}$$

Algorithm 1: Secure addition of medical records by authorized stakeholder S_i;.

```
Input: M^j_{S_i};
Output: True/False;
Restrict access to an account from any unauthorized user;
recordCount = =0;
if (S_i is registered and valid) then
    Check that the stored medical records are not empty;
    if (length of M^j_{S_i} > 0) then
        recordCount + +; /increment recordCount;
        M^j_{S_i}[recordCount].data = M^j_{S_i};
        create new MedicalRecord with M^j_{S_i} initially making
          sharing permission false;
        Set M^j_{S_i}[recordCount].isShared = false;
        store the MedicalRecord in medicalRecords;
        return MedicalRecordAdded(recordCount, S_i);
        return True;
    else
        Data should not be empty;
        return False;
else
    return Unauthorized S_i;
```

4.3. Access control and permissions

Granting Access to Stakeholders: Let P = P_1, P_2, …, P_p represent the set of permissions granted within the system, where p is the total number of permissions. Each permission P_k consists of the following attributes:

$$P_k = (S_i^A, j, \wp_k) \tag{6}$$

The permission-granting function is denoted as G(P_k, S_i^A), where P_k is the permission. First, an authorized stakeholder S_i grants selective access rights to another stakeholder S_s through the access-granting procedure described in Algorithm 2. This process ensures that permissions are only issued for valid medical record identifiers and that all authorization checks are satisfied before enabling access. This function verifies the permission request and updates the access control for the corresponding record. The access control function, AC, is represented as:

$$AC(M^j_{S_i}, S_i^A, P_k) = \begin{cases} 1 & \text{if } S_i \text{ has access to } M^j_{S_i} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

This function determines whether stakeholder S_s has access to the medical record $M^j_{S_i}$ based on permission P_k, where S_s is the permitted stakeholder.

Algorithm 2: Granting selective access to healthcare records for stakeholder S_s.

```
Input: j, S_i;
Output: P_k granted/denied, True/False;
Restrict access to an account from any unauthorized user;
Set S_s! = S_i;
if (registeredStakeholders[S_i] = true) then
    Check that the stored medical records of S_i are not empty;
    if (length of M^j_{S_i} > 0) then
        Check j is a valid record ID;
        if (j > 0 && j <= recordCount) then
            medicalRecords[j].P_k[S_s] = true;
            return PermissionGranted(j, S_s);
            Access granted with permission number P_k;
        else
            Invalid j;
    else
        Data should not be empty;
        return False ;
else
    return Unauthorized S_i;
```

Revocation Access from Stakeholders: The revocation of permissions can be represented by the function R(P_k), where P_k is the revoked permission. If at any point a granted permission must be withdrawn, the revocation procedure illustrated in Algorithm 3 securely removes the stakeholder's access to the corresponding record to maintain system integrity and auditability. The updated access control is $AC_{\text{update}}(M^j_{S_i}, S_i^A, P_k)$, revoke access S_s for record $M^j_{S_i}$ so that any unauthorized stakeholder's access to sensitive medical records is immediately revoked.

4.4. Medical record retrieval

The process of retrieving medical records by a stakeholder S_s involves a series of secure and validated steps. First, the stakeholder must be registered and verified to ensure they are an authorized entity within the system. Once validated, the system performs a permission check to confirm that S_s has been granted access to the requested record. If this check is successful, the encrypted medical record $E(M_{ij}, K)$ is retrieved. Finally, S_s uses the appropriate decryption key K to securely access the record's contents. This retrieval workflow is formally implemented in Algorithm 4, which ensures that only authenticated and authorized stakeholders can view medical records. It performs a sequential validation of registration, record availability, record identifier, and permission status before allowing decryption and data access. The overall process of secure record retrieval can be summarized as: RetrieveRecord(M_j, S_s) & → CheckPermission(S_s) & → Decrypt($E(M_j, K)$). Therefore, access to sensitive medical data is strictly limited to authorized stakeholders, thereby preserving patient privacy and the security guarantees provided by the blockchain infrastructure.

Algorithm 3: Secure revocation of access permissions to medical records for stakeholder S_s.

Input: j, S_s;
Output: Permission P_k revoked, True/False;
Restrict access to an account from any unauthorized user;
if *(registeredStakeholders[S_i] = true)* **then**
  Check that stored medical records are not empty;
  if *(length of $M^j_{S_i} > 0$)* **then**
    Check j is a valid record ID;
    if *(j > 0 && j <= recordCount)* **then**
      medicalRecords[j].$P_k[S_s]$ = false;
      return PermissionRevoked(j, S_s);
      Permission **revoked** with permission number P_k;
    else
      Invalid j;
  else
    Data should not be empty ;
    return False ;
else
  return Unauthorized S_i;

Algorithm 4: Controlled access of medical records by stakeholder S_i based on permission status.

Input: j;
Output: $M^j_{S_i}$, True/False;
Restrict access to an account from any unauthorized user;
if *(registeredStakeholders[S_i] = true)* **then**
  Check that stored medical records are not empty;
  if *(length of $M^j_{S_i} > 0$)* **then**
    Check j is a valid record ID;
    if *(j > 0 && j <= recordCount)* **then**
      Check that S_s has access to the specified record;
      if *(medicalRecords[j].$P_k[S_s]$ = true)* **then**
        return medicalRecords[j].$M^j_{S_i}$;
      else
        Access denied;
    else
      Invalid j;
  else
    Data should not be empty;
    return False ;
else
  return Unauthorized S_s;

5. FL Integration into proposed FLIT

This section outlines how FL is integrated into the FLIT framework to enable decentralized, privacy-preserving model training across multiple healthcare stakeholders, as shown in Fig. 3. It explains the local training process, parameter updates, and the role of differential privacy in protecting sensitive medical data during gradient sharing.

5.1. Local model training

In the FLIT framework, each stakeholder node $S_i \in S$ trains a personalized local model M_i on its own private medical dataset without sharing raw data. The local dataset is represented as:

$$D_i = \{(x_j, y_j)\}_{j=1}^{n_i} \tag{8}$$

Model training at each node is performed using a gradient-based optimization approach. Specifically, the model parameters $\theta_t^{(i)}$ at round t are updated using the following rule:

$$\theta_{t+1}^{(i)} = \theta_t^{(i)} - \eta \cdot \nabla_\theta L_i(M_i(x_j), y_j) \tag{9}$$

This formulation allows each stakeholder to independently optimize its model while preserving data locality. To ensure the privacy of sensitive healthcare information during training, FLIT employs differential privacy. Each node perturbs its gradient before sharing, using additive Gaussian noise:

$$\bar{\nabla}_\theta = \nabla_\theta + \mathcal{N}(0, \sigma^2 I) \tag{10}$$

This perturbation ensures that no single data point has a significant influence on the shared model update. The strength of privacy guarantees is controlled by the differential privacy parameters ϵ and δ, which define the privacy budget and the allowable probability of information leakage, respectively.

5.2. Model update and encryption process

In the FLIT framework, each participant S_i trains a local model using private medical data. After completing local training, the participant computes a model update $\Delta\theta^{(S_i)}$, which represents the change in model parameters between two successive training rounds. This is calculated as the difference between the updated parameters $\theta_{t+1}^{(S_i)}$, obtained after local training at round $t+1$, and the parameters before training $\theta_t^{(S_i)}$, that were in place at round t, such that

$$\Delta\theta^{(S_i)} = \theta_{t+1}^{(S_i)} - \theta_t^{(S_i)}. \tag{11}$$

Here, $\Delta\theta^{(S_i)}$ captures the knowledge learned from the participant's local data. To preserve the confidentiality of this update during transmission, each participant encrypts the update using a secure encryption function, denoted as `Encrypt`, which employs SMPC techniques. The encryption is performed using a unique key K_{S_i} assigned to the participantS_i for producing the encrypted update $\delta^{(S_i)}$ as follows:

$$\delta^{(S_i)} = \text{Encrypt}(\Delta\theta^{(S_i)}, K_{S_i}). \tag{12}$$

This mechanism ensures that only authorized aggregators possessing the correct decryption key can access the model updates for safeguarding the confidentiality of sensitive patient information and preventing any potential data leakage.

5.3. Smart contract coordination for update submission

In the FLIT framework, the encrypted model updates $\delta^{(S_i)}$, are submitted to the blockchain via a smart contract. This ensures that model updates are handled in a decentralized and secure manner, with critical checks to maintain the integrity, traceability, and trustworthiness of the data. The smart contract coordinates the submission of updates so that all actions are compliant with the system's rules and that no unauthorized access or modifications occur.

Verification of Participant Identity. The first step in the smart contract ensures that the identity of the participant submitting the model update is verified. Only authorized participants are allowed to submit updates to the system. The contract verifies the user's identity by checking a verified status. If the participant is not verified, the contract rejects the update submission.

Timestamping to Prevent Replay Attacks. To prevent replay attacks, where malicious actors might attempt to resend previous valid updates to disrupt the system, each model update is timestamped. The timestamp provides a clear record of when the update was submitted so that all transactions are processed in a chronological order. This prevents double-spending or resubmission of outdated updates, which could compromise the integrity of the FL process. The timestamp is recorded during the submission and emitted to the blockchain.

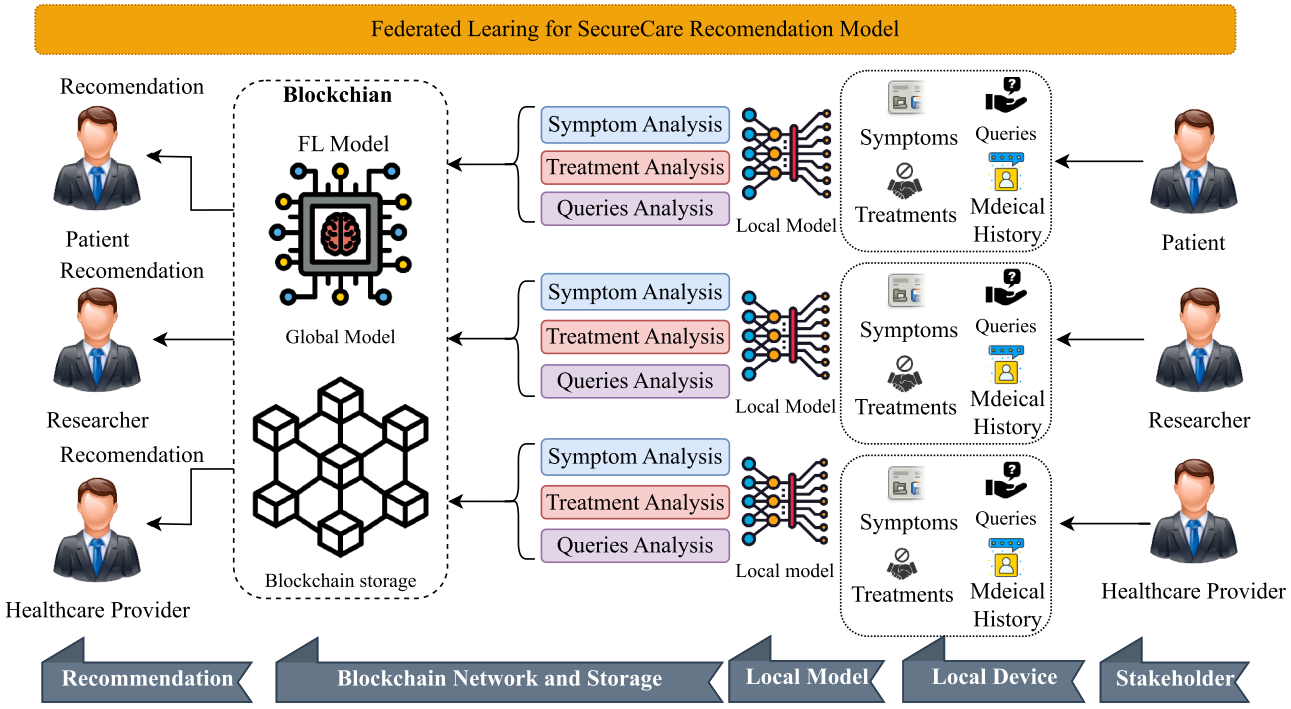


Fig. 3. The process of FL model integration to FLIT.

5.4. Metadata validation for structural integrity

The smart contract also performs metadata validation to ensure that each submitted model update complies with the system's structural and integrity requirements. This process guarantees that the updated data is properly formatted, complete, and ready for storage on the blockchain. The metadata associated with each update is denoted as $M_i = \{\text{ID}_i, \text{Hash}(\delta^{(S_i)}), \text{Timestamp}_i\}$ where ID_i is the unique identifier assigned to the participant S_i, $\text{Hash}(\delta^{(S_i)})$ represents the cryptographic hash of the encrypted model update $\delta^{(S_i)}$, and Timestamp_i records the exact submission time to ensure proper chronological sequencing of updates. This validation mechanism is essential for maintaining the consistency, integrity, and auditability of FL contributions within the FLIT system.

5.5. Blockchain logging and immutable audit trail

In the FLIT framework, all submissions of encrypted model updates are recorded on the blockchain ledger, denoted as B. This ensures that the entire process is transparent, traceable, and tamper-resistant, maintaining the integrity of the FL system. The blockchain serves as an immutable audit trail for a secure and decentralized record of every model update submission.

Blockchain Logging. Each model update submission is logged on the blockchain along with key information to maintain transparency and accountability in the system. The logged information includes:

- **Encrypted Model Update:** The encrypted model update $\delta^{(S_i)}$, which contains the changes to the model's parameters, is securely stored in the blockchain. This ensures that the update remains confidential and can only be decrypted by authorized participants or aggregators.
- **Node ID:** The unique identifier for the node or participant submitting the update, denoted as S_i. This helps trace the origin of the update and ensures that only verified participants are contributing to the FL process.
- **Timestamp:** The time at which the model update was submitted, denoted as τ_i. This timestamp is crucial for preventing replay attacks and ensures that the sequence of updates is correct and chronological.

The information is logged in the blockchain as $\text{Log}(S_i, \delta^{(S_i)}, \tau_i)$. This log entry serves as an immutable record of the update submission, allowing stakeholders to verify the integrity of the system and the model updates.

Replay Protection via Timestamping. The inclusion of a timestamp τ_i for each model update submission serves as a mechanism for preventing replay attacks. The blockchain ensures that updates can only be processed once and in the correct sequence by integrating timestamps. This feature prevents malicious actors from resubmitting outdated or duplicate updates, which could otherwise disrupt the FL process or lead to erroneous results.

Immutability via Distributed Ledger Properties. One of the core features of blockchain technology is its immutability, which ensures that once data is recorded, it cannot be altered or tampered with. The distributed nature of the blockchain means that the data is replicated across multiple nodes. This immutability guarantees that the log entries are permanent and secure for an indisputable record of every update and maintains the trustworthiness of the system.

5.6. Off-chain secure aggregation

In the FLIT framework, after all the encrypted model updates $\delta^{(i)}$ for $i = 1, 2, \ldots, N$ have been submitted to the blockchain and verified through smart contract interactions, the next step involves securely aggregating the updates to produce a global model. This aggregation process occurs off-chain to maintain privacy and computational efficiency to ensure that sensitive information remains protected throughout the process. Once the encrypted model updates have been received, the aggregator performs two critical tasks:

The aggregator decrypts the updates $\delta^{(i)}$ using the appropriate decryption keys. This ensures that the updates, which are originally encrypted to preserve confidentiality, can now be processed. Only authorized aggregators can perform the decryption due to their possession of the necessary decryption keys. After decryption, the model updates $\Delta\theta^{(i)}$ for each participant (i.e., the difference between the previous model

state $\theta_t^{(i)}$ and the new model state $\theta_{t+1}^{(i)}$) are aggregated to form the global model. The aggregation process uses a federated averaging algorithm (FedAvg), which computes the average of the model updates from all participants to generate a single, consolidated model. The aggregation can be expressed below:

$$M_{\text{global}} = A\left(\{\Delta\theta^{(i)}\}_{i=1}^{N}\right) \tag{13}$$

If the participants' updates need to be weighted based on factors such as the size of their local datasets, the aggregation formula can be adjusted to reflect these differences. Specifically, the global model parameters θ_{global} can be computed by normalizing each participant's contribution according to the number of data samples they possess. The weighted aggregation is given by:

$$\theta_{\text{global}} = \frac{\sum_{i=1}^{N} n_i \cdot \theta_{t+1}^{(i)}}{\sum_{i=1}^{N} n_i}, \tag{14}$$

where n_i denotes the number of data points available at participant i and θ_{global} is the final aggregated model. This weighted approach ensures that participants with larger datasets have proportionally greater influence on the global model in a more accurate and representative outcome across the distributed system.

5.7. Trust-weighted aggregation and complexity analysis

While classical and weighted FedAvg account for the size of local datasets, they assume that all participants are equally reliable. In real-world IoMT settings, however, some devices may be faulty, compromised, or malicious. To address this, FLIT introduces a dynamic trust score T_i for each participant S_i, which reflects the historical reliability and quality of their contributions.

Trust Score Construction. The trust score $T_i \in [0, 1]$ is updated at the end of each round based on three main factors: (i) *model contribution quality* measured by the alignment of the local update with the aggregated global direction; (ii) *behavioral consistency*, where frequent deviations or anomalous updates lower the trust score; and (iii) *participation history*, where nodes that contribute regularly and honestly are rewarded with higher trust values. This dynamic adjustment ensures that participants who attempt poisoning attacks or submit low-quality updates are gradually down-weighted.

Trust-Weighted Aggregation. The global update at round t is then computed using the trust-weighted formulation:

$$\theta_{t+1} = \frac{\sum_{i=1}^{N} T_i \cdot n_i \cdot \theta_{t+1}^{(i)}}{\sum_{i=1}^{N} T_i \cdot n_i}, \tag{15}$$

where n_i is the local dataset size of participant i. This formulation ensures that both data volume and participant reliability jointly determine influence in the global model.

Complexity Analysis. The computational complexity of the trust-weighted aggregation remains linear in the number of participants, i.e., $\mathcal{O}(N)$, since each update requires only a constant-time multiplication by the trust factor T_i. Compared to standard FedAvg, the additional overhead is negligible, as trust scores are updated using lightweight statistical checks or anomaly detection methods that operate in near real-time. The communication complexity is unchanged because the trust score is a scalar appended to each participant's update. Thus, the mechanism scales efficiently with the number of IoMT devices to provide stronger robustness against poisoning and unreliable contributions. This design balances theoretical simplicity with practical resilience. The trust-weighted extension enhances robustness and fairness for large-scale, adversarial IoMT deployments.

5.8. Global model update and on-chain logging

After the model updates from all participants have been securely aggregated, the next step is to generate the updated global model, denoted as M_{global}. This global model incorporates the contributions from each node and represents the collective learning of all participants. To ensure model integrity and facilitate model versioning, a cryptographic hash of the global model is computed $\text{Hash}(M_{\text{global}}) \rightarrow B$. The hash provides a secure anchor for the global model. The Solidity smart contract function handles the logging of the global model hash, which ensures that only authorized aggregators can log the global model's hash.

5.9. Feedback to participants

Once the global model has been updated and securely logged on the blockchain, each participant S_i receives feedback regarding the new model. This feedback can take one of two forms: either the complete updated global model M_{global}, or alternatively, a cryptographic hash of the global model accompanied by a reference to its download location. To assess the effectiveness of the global model, each participant evaluates it using their own local test dataset, denoted as D_i^{test}. The performance evaluation is expressed by the equation:

$$E_i = \text{Eval}(M_{\text{global}}, D_i^{\text{test}}), \tag{16}$$

This decentralized evaluation ensures that each stakeholder can independently verify the utility and relevance of the global model on their own data to support transparent and personalized validation. This evaluation process allows participants to gauge the performance of the global model in the context of their own data. Any performance feedback resulting from these evaluations can be logged for future learning rounds or for auditing purposes.

6. Attacker models and mitigation strategy

This section defines various adversarial threats within the FL environment and outlines formal mitigation strategies used in FLIT. It operates under a semi-honest threat model where participants follow protocol specifications but may engage in adversarial behavior such as privacy inference, poisoning attacks, or tampering with blockchain records. Our threat taxonomy spans four principal attacker types: (1) honest-but-curious aggregators, (2) poisoning attackers, (3) blockchain rollback attackers, and (4) inference attackers. Each class of threat is mitigated through a formally verifiable strategy.

6.1. Honest-but-curious aggregator

An honest-but-curious aggregator in FLIT follows the protocol as expected but may try to learn private information from the model updates it receives (Bonawitz et al., 2017). Even if the data is not raw, the aggregator could attempt techniques like gradient inversion or pattern analysis to infer sensitive patient details. This poses a serious privacy risk, as gradients might reveal identifiable information. Even when no raw data is shared, the transmission of gradient updates δ_j may pose a threat to data privacy if not properly protected.

Lemma 1 (Gradient Inversion Risk)**.** *The adversary observes the gradient update as follows:*

$$\delta_j = \nabla_\theta \mathcal{L}(\theta, D_j) \tag{17}$$

and attempts to reconstruct private data using an inference function:

$$\hat{D}_j = \mathcal{I}(\delta_j) \tag{18}$$

where $\mathcal{I}$ represents a gradient inversion method. This process poses a risk if gradients are shared in clear form.

Mitigation Strategy: FLIT adopts a multi-layered defense involving differential privacy, encryption, and blockchain-based auditing.

Theorem 1 (Differential Privacy Guarantee)**.** *Each participant perturbs their gradient update using Gaussian noise:*

$$\tilde{\delta}_j = \delta_j + \mathcal{N}(0, \sigma^2) \tag{19}$$

This satisfies (ϵ, δ)-differential privacy, ensuring that the presence or absence of any single record in D_j cannot be reliably inferred from the output.

Lemma 2 (Indistinguishability of DP Outputs). *Let M be the randomized mechanism applying differential privacy to model updates. For all neighboring datasets D, D' differing in at most one record, and for all subsets $S \subseteq Range(M)$, we have:*

$$\Pr[M(D) \in S] \leq e^{\epsilon} \cdot \Pr[M(D') \in S] + \delta \tag{20}$$

This bound ensures that DP outputs from similar datasets are statistically indistinguishable.

Theorem 2 (Encrypted Aggregation Security). *Once differential privacy is applied, each participant encrypts their noisy update:*

$$\delta_j^{enc} = Encrypt(\tilde{\delta}_j, K_j) \tag{21}$$

FLIT employs SMPC to perform aggregationins. The global aggregated update is obtained by decrypting the sum of all encrypted local contributions as expressed below:

$$\tilde{\Delta} = Decrypt\left(\sum_{j=1}^{N} \delta_j^{enc}\right) \tag{22}$$

6.2. Poisoning attacker

A poisoning attacker in FLIT is a malicious participant who tries to disrupt the learning process by submitting tampered or harmful model updates (Wang et al., 2022). These poisoning attacks may aim to degrade global model performance (availability attack) or inject targeted vulnerabilities (backdoor attack) that allow specific malicious inputs to trigger misclassifications. Such poisoned updates are often structured as:

$$\delta_j^{\text{malicious}} = \delta_j + \epsilon_{\text{poison}} \tag{23}$$

where ϵ_{poison} is adversarial noise crafted to bias the global model outcome.

Mitigation Strategy: FLIT counters poisoning attacks using a combination of anomaly scoring and trust-weighted aggregation. Each client update δ_j is scored using a statistical consistency function:

$$s_j = V(\delta_j), \quad V : \mathbb{R}^d \to [0, 1] \tag{24}$$

where low values of s_j indicate that the update is an outlier or likely poisoned. Updates are then aggregated using a trust-weighted scheme:

$$M_{\text{global}} = \frac{1}{\sum_{i=1}^{N} s_i} \sum_{i=1}^{N} s_i \cdot \delta_i \tag{25}$$

so that suspicious updates have reduced influence on the global model. All updates and their trust scores are logged via blockchain smart contracts for transparency and future accountability.

Lemma 3 (Adversarial Update Impact). *A poisoning attack modifies updates such that:*

$$\delta_j^{malicious} = \delta_j + \epsilon_{poison} \tag{26}$$

The anomaly scoring function V assigns a trust score s_j based on the consistency of δ_j with the distribution of benign updates.

Lemma 4 (Malicious Update Detection). *Let $S = \{s_1, s_2, \ldots, s_N\}$ be the trust scores of all clients. If the scoring function V satisfies the Lipschitz condition and exhibits bounded variance on benign updates, then:*

$$\mathbb{E}[s_j \mid \delta_j^{malicious}] < \tau \tag{27}$$

for a chosen threshold $\tau \in (0, 1)$. This allows the system to statistically flag poisoned updates.

Theorem 3 (Byzantine Resilience). *Let $s_j \in [0, 1]$ denote the trust score for client j. The global model update is computed using (25). If fewer than $f < \frac{N}{3}$ clients are adversarial, and the scoring function correctly assigns lower weights to poisoned updates, then M_{global} remains close to the aggregation over benign participants. This provides robustness against Byzantine failures.*

6.3. Blockchain rollback attacker

A blockchain rollback attacker tries to take advantage of blockchain forks or reorganization events to alter or erase previously submitted model updates or audit logs (Saad et al., 2019). This type of attack undermines the trust in the system's history and accountability. In the context of FLIT, such an attacker may try to delete, reorder, or alter records of model update transactions T_i, which could lead to the loss of auditability, misalignment in federated training iterations, or unjustified acceptance/rejection of updates. This attack often uses chain forks or deliberate stalling of consensus to rewrite previously accepted transactions. Specifically, the attacker seeks to remove a valid block $B_t \in \mathcal{B}$ from the chain $\mathcal{B}$, such that:

$$\exists B_t : \text{Hash}(B_t) \in \mathcal{B} \to \text{Hash}(B_t) \notin \mathcal{B}' \tag{28}$$

where $\mathcal{B}'$ is the altered chain. This breaks the system's trust assumptions and may lead to incorrect acceptance or rejection of model updates.

Lemma 5 (Rollback Probability Bound). *Let $\mathcal{V} = \{V_1, V_2, \ldots, V_K\}$ be the set of trusted validators. If only validators in $\mathcal{V}$ can propose blocks, then for a rollback attack involving f malicious validators and r confirmations required for finality, the success probability is bounded as:*

$$P_{rollback} \leq \left(\frac{f}{K}\right)^r \tag{29}$$

As long as $f < \frac{K}{2}$, rollback becomes statistically improbable.

Lemma 6 (Validator Quorum for Transaction Finality). *A transaction T is only accepted into block B if a sufficient number of validators sign it:*

$$\sum_{k=1}^{K} Sign_k(T) \geq Threshold \tag{30}$$

This ensures Byzantine fault tolerance and prevents any single compromised validator from unilaterally altering or committing a transaction.

Theorem 4 (Tamper Detection). *Let $Hash(B_t) = H_t$ be the cryptographic fingerprint of block B_t. Suppose FLIT periodically commits H_t to a tamper-resistant external ledger $\mathcal{E}$ (e.g., IPFS or public blockchain). Then for any rollback attempt producing modified block B_t', the following holds $Hash(B_t') \neq H_t \Rightarrow$ Violation Detected. This ensures rollback attempts are detectable and traceable by cross-verifying against the external anchor.*

6.4. Inference attacker

An inference attacker attempts to reconstruct sensitive training data by analyzing received model gradients or updates, using techniques like membership inference or model inversion (Hu et al., 2022). This poses a risk of exposing private patient information, including re-identifying individuals or revealing sensitive attributes. An inference attacker analyzes model updates (e.g., gradients) during federated training to recover sensitive information about training data. This includes techniques like *Model Inversion:* Reconstructing input data $\hat{D}_i = \mathcal{I}(\delta_i)$ from gradients δ_i and *Membership Inference:* Determining whether a specific data record was used in training. Such attacks are especially dangerous in healthcare settings, where gradients may encode fine-grained, identifiable features of patients.

Lemma 7 (Gradient Inversion Risk). *Let $\delta_i = \nabla L(\theta_t, D_i)$ denote the gradient computed by client i. An inference attacker attempts to reconstruct training data as:*

$$\hat{D}_i = \mathcal{I}(\delta_i) \tag{31}$$

where $\mathcal{I}$ is a reconstruction function. Such reconstructions are feasible in the absence of protection mechanisms.

Lemma 8 (Sensitivity Bounding via Clipping). *To reduce privacy leakage, gradients are clipped using:*

$$\delta_i \leftarrow \frac{\delta_i}{\max\left(1, \frac{\|\delta_i\|_2}{C}\right)} \tag{32}$$

Table 3
Tools used in experimenting with FLIT.

Tool	Version	Use case(s)
TensorFlow	2.11.0	Training deep learning models (e.g., LSTM autoencoders) on client data in isolation.
PySyft	0.7.1	Privacy-preserving FL via secure computation and model update sharing.
Flower (FLwr)	1.5.0	Lightweight framework to simulate FL servers and multiple edge clients.
TensorFlow Federated	0.53.0	Composing and simulating FL algorithms using TensorFlow.
OpenDP Library	0.2.0	Applies differential privacy mechanisms (e.g., Gaussian noise) to protect client updates.
Remix IDE	0.33.2	Web IDE to develop, test, and deploy Solidity smart contracts.
Solidity Compiler	0.8.18	Used to compile smart contracts with defined syntax and logic.
Remix VM	Built-in	Local EVM for rapid contract testing without deploying to a live blockchain.
Ganache CLI	6.12.2	Simulates local Ethereum blockchain for debugging and transaction testing.
Web3.py	5.31.1	Python library to interact with Ethereum nodes and deployed contracts.
SPDX License	GPL-3.0	Ensures legal compliance and openness of deployed smart contracts.
IPFS	0.14.0	Distributed storage for off-chain backup and audit anchoring of model update logs.
Python	3.10+	Main scripting language for FLIT's federated orchestration and blockchain control.
PyCryptodome	3.18.0	Implements local encryption of model updates and random key generation for clients.

This ensures bounded contribution per participant before applying privacy-preserving noise.

Theorem 5 ((ε, δ)-DP via Gaussian noise)**.** *Noisy gradients are computed as:*

$$\delta_i^{DP} = \delta_i + \mathcal{N}(0, \sigma^2 I) \tag{33}$$

This mechanism satisfies (ϵ, δ)-differential privacy. As σ increases, privacy guarantees become tighter to prevent effective inversion or membership inference.

Lemma 9 (Averaging Improves Privacy)**.** *If the server observes only the sum of noisy updates:*

$$\mathcal{M} = \sum_{i=1}^{N} \delta_i + \mathcal{N}(0, N\sigma^2) \tag{34}$$

Then, due to averaging and the central limit theorem, the overall privacy bound improves to a tighter (ϵ', δ')-DP compared to individual transmissions.

Theorem 6 (Secure Aggregation Privacy)**.** *Cryptographic protocols allow the server to compute $\sum_{i=1}^{N} \delta_i^{DP}$ without accessing any individual gradient. This prevents leakage even if the server is semi-honest.*

7. Experiment results and performance analysis

7.1. Environment setup

We experiment with the proposed algorithms in the proposed framework using Ethereum Remix IDE (Das et al., 2021a). We design and implement a smart contract using Solidity (Das et al., 2021a) language. The Remix IDE is a popular integrated development environment for programming and testing smart contracts on various blockchain platforms. It provides several Ethereum Virtual Machine (EVM) options that can be used for cost analysis when deploying and executing smart contracts (Das et al., 2022). We first connected to the Ethereum Sepolia testnet using Infura, a widely used Ethereum infrastructure provider. Infura allowed us to interact with the Ethereum network without running our own nodes, as shown in Fig. 4. The Web3 library was used to create this connection. However, Table 3 shows the useful tools applied in this experiment.

For FL, the experimental setup uses Python 3.8, with TensorFlow 2.x for model training and scikit-learn for evaluation. Simulated federated clients train local models on their respective partitions of the diabetes dataset. A trust-weighted aggregation mechanism is employed at the central aggregator using gradient clipping, differential privacy, and secure aggregation to ensure robustness against poisoning and inference attacks.

7.2. Dataset selection

To evaluate the FL capabilities of the proposed FLIT framework, we utilize the Diabetes 130-US hospitals for the years 1999–2008 dataset, available from the UCI Machine Learning Repository (Clore et al., 2014). This dataset contains over 100,000 patient records collected from 130 U.S. hospitals, including demographic information, diagnosis codes, treatment history, lab results, and readmission status. For the federated setting, the dataset is horizontally partitioned by hospital ID to simulate data distribution across multiple healthcare institutions, each acting as a federated client. The data is preprocessed by handling missing values, encoding categorical variables, and standardizing numerical features. This enables privacy-preserving training for models focused on predicting readmission risk and recommending treatment strategies.

Selection Criteria. From the raw dataset, we selected attributes relevant to readmission prediction and treatment optimization, as these are critical indicators for clinical decision support. Attributes with high proportions of missing or inconsistent entries were excluded. Only records with complete demographic and outcome data were retained to ensure robust model training and reproducibility.

Preprocessing. The dataset underwent a structured preprocessing pipeline. Missing numerical values were imputed with hospital-specific medians, while categorical values were imputed with the mode. Categorical features such as race, admission type, and discharge disposition were one-hot encoded, while ordinal features such as age brackets were integer encoded. Continuous features, including lab results and visit counts, were normalized to zero mean and unit variance. Although the dataset is tabular rather than multimodal in the sense of time-series and images, we applied *synchronization* of temporal variables (e.g., aligning visit dates) and *noise reduction* to exclude inconsistent or duplicate entries. These steps stabilize model training and make results more reproducible in heterogeneous federated environments.

Parameter Tuning. Model hyperparameters were optimized using a grid search strategy at the client level and validated through the central aggregator. The search space included learning rate (0.001–0.01), batch size (32, 64, 128), and local training epochs (1, 3, 5). Dropout rates between 0.2 and 0.5 were explored to prevent overfitting. Aggregation weights were dynamically adjusted based on client trust scores, reflecting accuracy and reliability in prior rounds. The final configuration integrated gradient clipping (threshold = 1.0), differential privacy (ϵ = 1.0), and secure aggregation, providing resilience against adversarial attacks while preserving patient privacy.

7.3. Smart contract-based record management

This section details the implementation of core smart contract functionalities in the FLIT framework, including stakeholder registration, medical record addition, and permission control. Each function enforces

```python
from web3 import Web3

# Function to connect to Ethereum blockchain (via Infura)
def connect_to_blockchain():
    infura_url = "https://sepolia.infura.io/v3/917db09e94724857af5fa6141b467294"
    w3 = Web3(Web3.HTTPProvider(infura_url))

    if w3.is_connected():
        print("Connected to Ethereum Testnet via Infura")
    else:
        print("Failed to connect to Ethereum")

# Call the function to check connection
if __name__ == "__main__":
    connect_to_blockchain()
```

```
Connected to Ethereum Testnet via Infura
```

Fig. 4. Establish connection to Ethereum testnet via Infura.

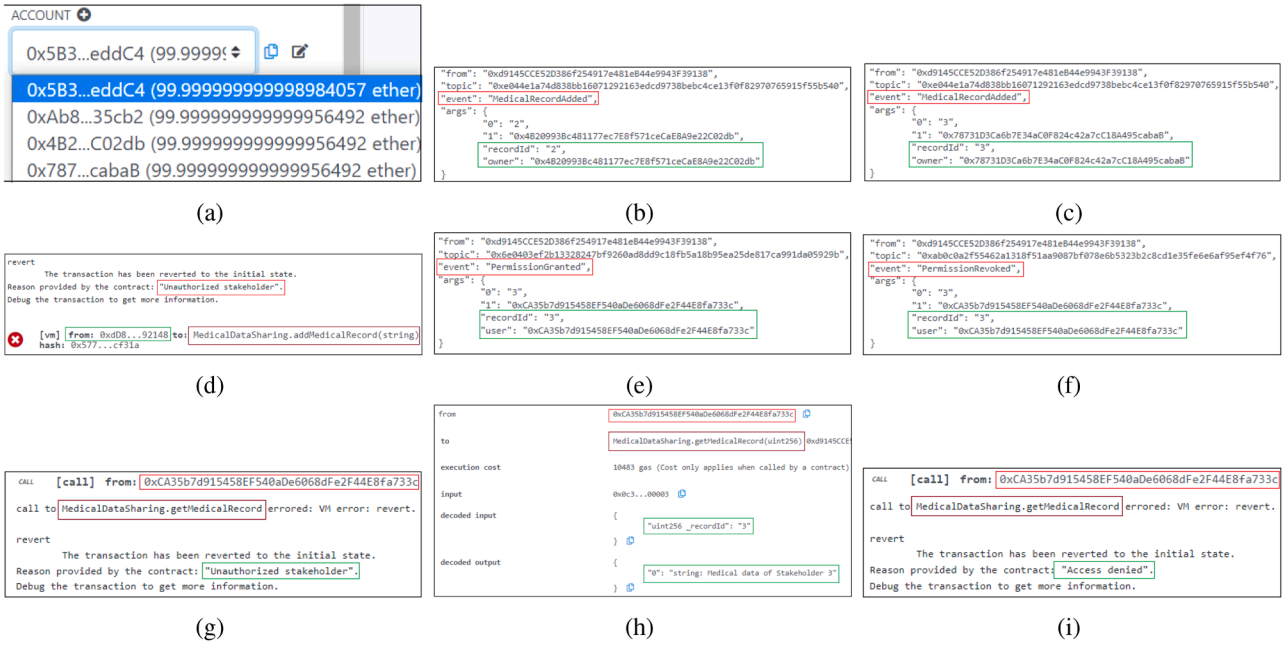


Fig. 5. Registration, authentication, and data sharing between stakeholders. (a) Registered stakeholders' accounts. (b) S_i: 0x4B2…C02db adds medical data with record ID j = 2. (c) S_i: 0x787…cabaB adds medical data with record ID j = 3. (d) An unregistered stakeholder cannot add data. (e) Permission granted to a stakeholder. (f) Permission revoked from a stakeholder. (g) An unregistered stakeholder cannot retrieve data. (h) Data retrieved by a permissioned registered stakeholder. (i) A permissionless registered stakeholder cannot retrieve data.

secure access for only authorized entities can interact with sensitive healthcare data.

Stakeholder Registration and Verification. registeredStakeholders function maintains a record of registered S_i in the system. By assigning the value "true" to the "registeredStakeholders", it indicates that (S_i^A) is the corresponding address of the registered S_i. Fig. 5(a) shows the list of registered S_i assigned for the experiment.

Adding Medical Records. We created a function called "addMedicalRecord" that adds a new medical record to the system. It validates that there is actual data present to be added as a medical record. If M_j is empty, the function execution will stop and throw an exception. registeredStakeholders[S_i] = true, which means S_i is a registered stakeholder in the system. This verification ensures that only authorized participants can add medical records. Fig. 5(b) and (c) show that the medical data is stored by two different S_i. However, Fig. 5(d) shows that no unauthorized stakeholders can add data.

Permission Granting to Stakeholders. We create a function called "grantAccess" that grants permission to a specified S_s for accessing a particular medical record $M^j_{S_i}$. j is the identifier of the medical record for which S_i wants to grant access. S_i is the stakeholder whom S_i wants to grant access to the medical record $M^j_{S_i}$. The created function sets the value of medicalRecords[j]. $P_k[S_s]$ to "true" in the permissions mapping associated with the medical record. This action allows the specified S_s to access the record. Fig. 5(e) The event "PermissionGranted" indicates that access has been granted to the specified S_s (here, 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c) for the

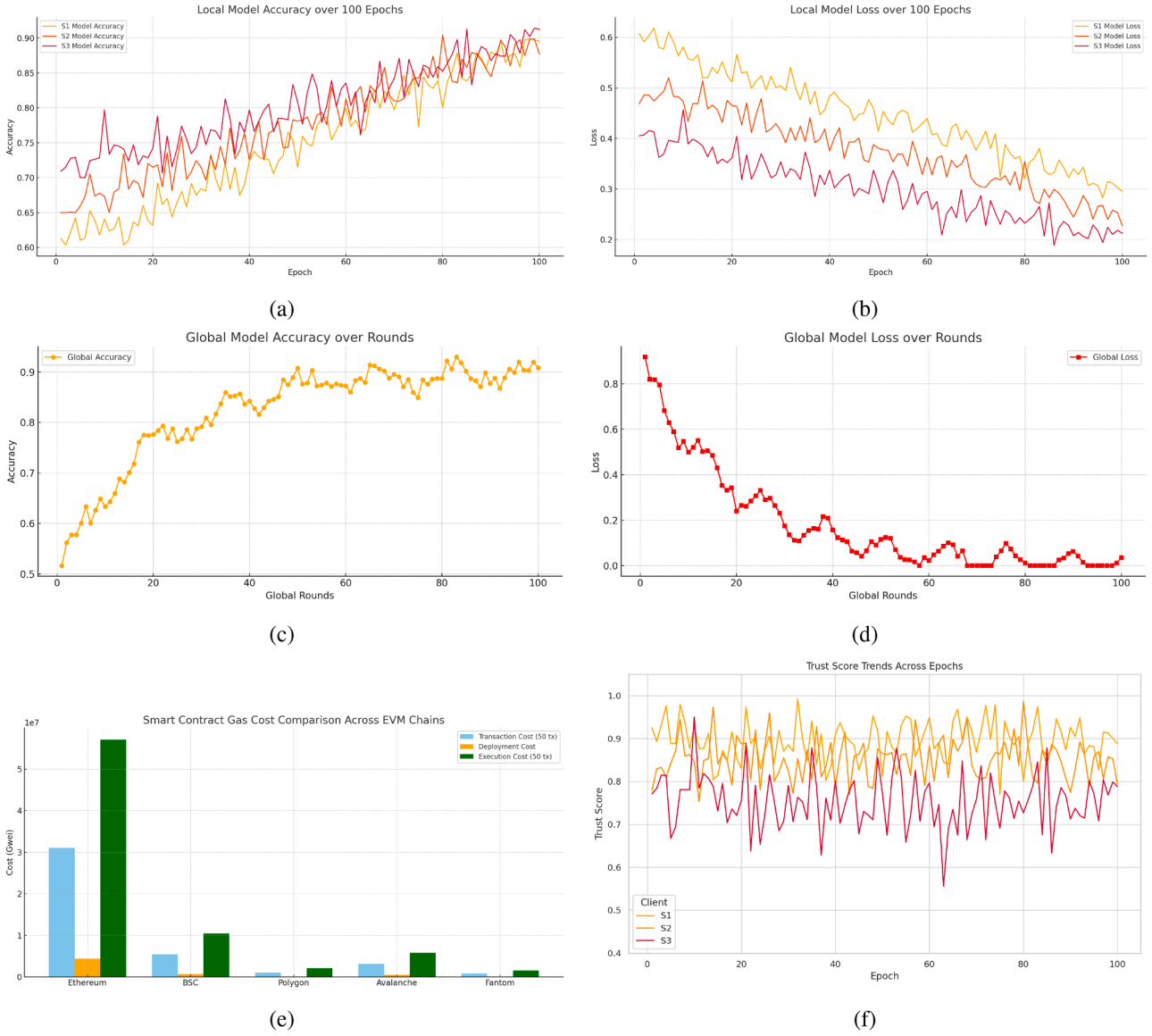


Fig. 6. Performance evaluation of the proposed FLIT framework. (a) Accuracy progression of local models across 100 training epochs for individual clients. (b) Loss trends of local models over 100 epochs. (c) Federated global model accuracy across aggregation rounds. (d) Global model loss across federated training rounds. (e) Smart contract gas cost analysis for key operations (deployment, execution, and transaction) on Ethereum Virtual Machine (EVM) chains. (f) Trust score evolution for clients S1, S2, and S3 trust-based filtering throughout FL.

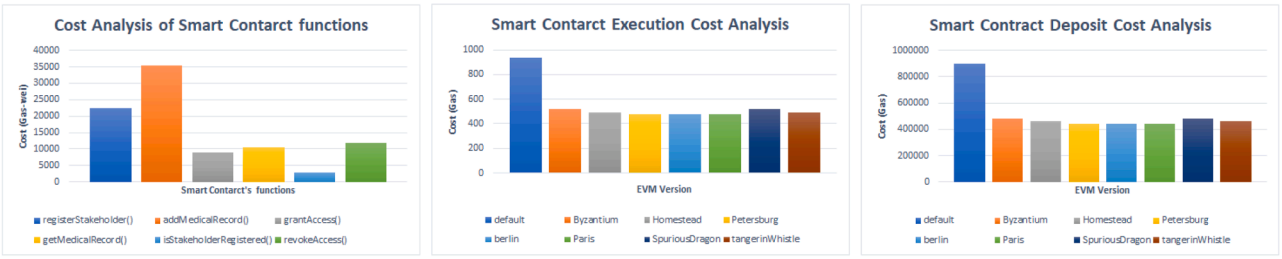
medical record with the given record ID j (here, 3) associated with S_i (here, 0x4B20993Bc481177ec7E8 f571ceCaE8A9e22C02db).

Permission Revocation from Stakeholders. We create a function called "revokeAccess" (shown in Fig. 9) that revokes permission for a specified S_s to access a particular medical record. S_s is the stakeholder from whom S_i wants to revoke access to medical data $M^j_{S_i}$. The created function sets the value of medicalRecords[j]. $P_k[S_s]$ to "false" in the permissions mapping associated with the medical record. This action removes access rights for the specified stakeholder. The event "PermissionRevoked" indicates that access permission has been revoked from S_s for the medical record with the given record ID j.

Data Retrieval by Stakeholders. We create a function called "getMedicalRecord" that retrieves the data of a medical record based on the given record ID j. Record ID j is validated for checking the existence of medical records in the system. If the record ID is not valid, the function execution will stop. A function is called to check that S_s has permission to access the specified medical record. Access control ensures that only authorized S_s can retrieve sensitive medical information, as shown in Fig. 8. If all the required verification and authentication are complete, the function medicalRecords[j]. $M^j_{S_i}$ retrieves the data of the specified medical record with the given j. However, unauthorized stakeholders cannot access the specified data, as shown in Fig. 5(f). As shown in Fig. 5(i), a registered stakeholder cannot retrieve data without the data owner's permission.

7.4. Performance analysis of FLIT

Performance analysis of the proposed framework is crucial to identify areas for improvement, optimize system performance, and ensure



(a) (b) (c)

Fig. 7. Gas-cost analysis of medical-data-sharing operations on various Ethereum Virtual Machine (EVM) platforms under different compiler optimization levels: (a) gas consumption for core smart-contract functions; (b) execution gas cost per transaction across EVMs; (c) deposit gas cost per transaction across EVMs.

the delivery of enhanced smart healthcare solutions. It involves benchmarking, stress testing, simulation, and evaluation of key performance indicators to efficiently and effectively utilize the technology in a healthcare context.

Local Model Performance. Fig. 6(a) and (b) show the accuracy and loss of local models trained over 100 epochs. The steady improvement in accuracy and the consistent reduction in loss demonstrate that individual IoMT clients can extract useful knowledge from their private data without depending on centralized storage. This is critical for healthcare applications, since medical data often resides in different hospitals or devices and cannot be freely shared due to privacy and regulatory concerns. The results validate that FLIT can support heterogeneous, decentralized learning, where each client contributes to knowledge building while retaining full control of its sensitive data.

Global Model Performance. Fig. 6(c) and (d) track the accuracy and loss of the global model during multiple federated rounds. Unlike single-site models, the aggregated global model benefits from the diverse data distributions across participants, achieving higher generalization and robustness. The use of trust-weighted aggregation ensures that updates from reliable nodes have greater influence, while differential privacy and secure aggregation guard against data leakage and malicious inference. The results confirm that FLIT maintains strong predictive performance while meeting strict privacy requirements—an essential balance for medical IoMT deployments where accuracy must coexist with confidentiality.

Smart Contract Cost Analysis. Fig. 6(e) examines the cost of executing FLIT's smart contracts across different Ethereum Virtual Machine (EVM) networks. Results clearly show that running on optimized EVM versions or on Layer-2 solutions (e.g., Polygon, Optimism) significantly reduces gas consumption compared to the Ethereum mainnet. This finding underscores a key design choice: in real-world IoMT settings, deployment platform selection directly affects affordability and scalability. For instance, in hospital networks or wearable ecosystems with thousands of daily transactions, high gas costs could become prohibitive. By showing that Layer-2 chains cut execution costs, the results confirm that FLIT can be realistically adopted without overwhelming operational expenses.

Trust Analysis. Fig. 6(f) plots the trust score evolution of three participants (S1, S2, and S3) across federated training. The results reveal that FLIT's dynamic trust mechanism is effective at distinguishing between reliable and unreliable participants. Trust scores for honest contributors remain stable or increase, while anomalous or malicious contributors are down-weighted over time. This adaptive adjustment ensures that the global model is not corrupted by poisoned updates or low-quality data. For IoMT environments, where device heterogeneity and potential adversarial behavior are common, trust analysis is vital. It provides resilience against data poisoning, Sybil attacks, and unreliable sensors, making the framework robust in practice.

Execution Cost Analysis. Fig. 7 condenses the on-chain expense profile of FLIT and highlights how different blockchain operations contribute to overall cost. As shown in Fig. 7(a), state-changing functions, such as adding, granting, or revoking access records, consume the majority of gas because they alter the blockchain ledger and therefore require consensus and storage updates across all nodes. In contrast, read-only queries are negligible in cost, since they do not change the ledger state and can be served locally without transaction fees. The execution cost analysis further compares gas consumption across Ethereum Virtual Machine (EVM) versions. Results show that the default EVM configuration incurs the highest execution cost (close to 900 gas units), while optimized versions such as Byzantium, Homestead, Petersburg, Berlin, Paris, SpuriousDragon, and TangerineWhistle reduce execution cost by almost half. This demonstrates the importance of selecting an appropriate EVM version: newer EVMs can significantly reduce runtime cost for resource-constrained healthcare environments.

Deposit and Contract Creation Cost Analysis. Fig. 7(b) shows the one-time costs of deploying smart contracts on different EVM versions, including the initial deposit and contract creation fees that depend on the size and complexity of the bytecode. The default EVM has the highest cost (about 9,00,000 gas units) due to limited optimization, while versions like Byzantium, Petersburg, and Berlin cut this overhead by 40–50 % through more efficient execution. For IoMT deployments, this means that large or poorly optimized contracts can create significant upfront expenses, especially when many institutions deploy the same framework. Keeping contracts modular and compiler-optimized helps reduce these costs and makes them easier to maintain. Although deposit and creation costs occur only once, they can still be a barrier to large-scale adoption. FLIT addresses this challenge by using lightweight, modular code and supporting deployment on sidechains and Layer-2 networks, where creation costs are much lower than on the Ethereum mainnet.

Function-Level Gas Cost Analysis. Fig. 7(c) breaks down the gas consumption of individual smart contract functions and the relative costs of different operations. Among them, the addMedicalRecord() function is the most expensive (≈35,000 gas units), since it requires writing new medical information to the blockchain, which involves creating or updating multiple state variables. Other operations, such as registerStakeholder() and revokeAccess(), also consume significant gas, as they modify access control records and permissions within the contract's state. These write-heavy functions account for most of the recurring operational cost because every state change must be validated by the blockchain network and permanently stored across all nodes. Functions like getMedicalRecord() and isStakeholderRegistered() incur very little cost because they are read-only queries. Such operations simply retrieve information without altering the blockchain's state so they do not require consensus or incur transaction fees. This distinction emphasizes that in IoMT contexts, where data must be frequently updated and access permissions regularly managed, state-changing functions represent the main driver of long-term operating costs. From a deployment perspective, these findings suggest that minimizing unnecessary writes, batching updates where possible, and designing lightweight data structures are critical strategies to keep recurring costs manageable. FLIT ad-

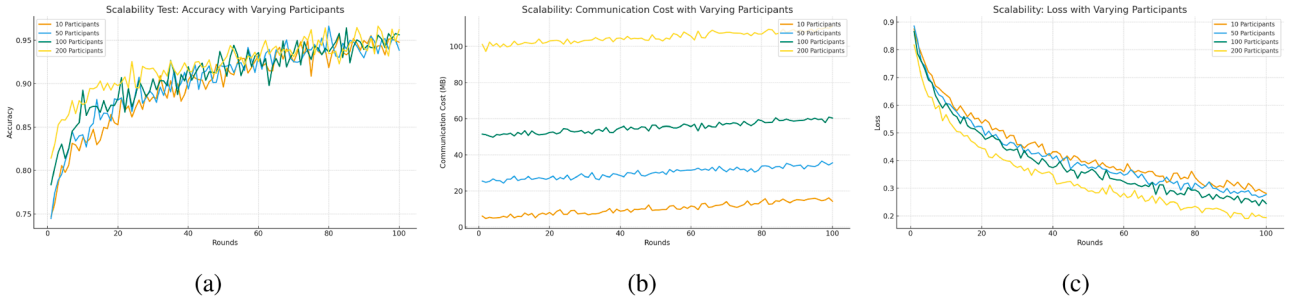


Fig. 8. Scalability evaluation of the FLIT framework under varying numbers of participants. (a) Accuracy trends across different participant sizes over 100 rounds. (b) Communication and execution costs grow with increasing participants. (c) Loss of convergence behavior under different participant configurations.

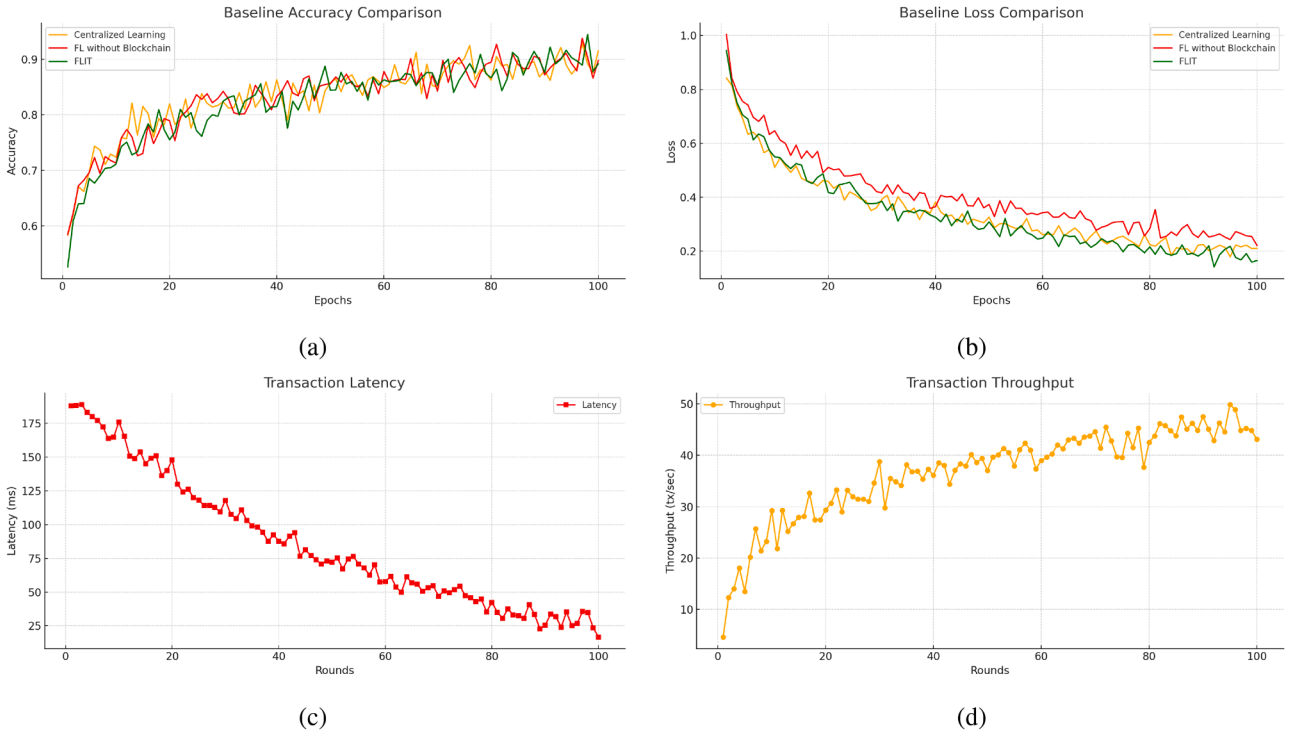


Fig. 9. Performance evaluation of FLIT in comparison to baseline approaches and blockchain-level operations. (a) Accuracy trends of centralized learning, FL without blockchain, and FLIT. (b) Loss convergence for the same approaches. (c) Latency behavior of smart contract operations across training rounds. (d) Throughput scalability of blockchain transactions with increasing rounds.

dresses this by anchoring only hashes of aggregated updates on-chain so that the number of costly state-changing transactions is reduced, while still maintaining data integrity, auditability, and security guarantees.

Scalability Analysis. Fig. 8(a) illustrates the effect of scaling participants on global model accuracy. As the number of participants increases, accuracy remains stable and improves steadily across training rounds over 90 % in large-scale settings. This indicates that FLIT can effectively integrate updates from a diverse set of contributors without performance degradation, which is critical for heterogeneous IoMT environments. Fig. 8(b) presents the communication and execution costs as the number of participants grows. While costs increase linearly with participants due to more frequent update submissions and smart contract validations, the slope remains manageable. This demonstrates that FLIT achieves favorable scalability by anchoring only metadata and hashes on-chain, thereby preventing exponential growth in gas consumption. Fig. 8(c) shows loss convergence trends across different participant counts. Despite increased heterogeneity with more clients, the loss decreases consistently and converges to a stable minimum. This demonstrates that FLIT's trust-weighted aggregation mechanism is effective in balancing contributions for robustness even when participant numbers scale significantly.

Baseline Performance. Fig. 9(a) and (b) present the accuracy and loss comparison between centralized learning, FL without blockchain, and the proposed FLIT framework. The results indicate that FLIT achieves accuracy levels comparable to centralized learning while outperforming FL without blockchain. Similarly, FLIT demonstrates faster and more stable loss convergence so that the addition of blockchain and trust mechanisms does not degrade model quality. Instead, it strengthens accountability and auditability while preserving learning efficiency.

Transaction Latency. Fig. 9(c) shows the latency trends of blockchain transactions across training rounds. Latency starts high during initial rounds due to contract initialization and consensus overhead, but decreases steadily as rounds progress. This trend highlights the adaptability of the system so that FLIT can support near-real-time medical IoT applications where responsiveness is critical.

Transaction Throughput. Fig. 9(d) demonstrates the throughput scalability of blockchain-based transactions. The throughput improves consistently with training rounds nearly 50 transactions per second by the

Table 4
Comparison of IoMT-specific security frameworks with the proposed FLIT framework.

Method	Technique Used	Description	Pros	Cons
Blockchain-IoMT Security Framework (Akkal et al., 2025)	Blockchain for data integrity	Secures IoMT medical records with blockchain to ensure immutability and auditability.	Provides tamper-proof audit trail.	High storage and energy overhead on IoMT devices.
Lightweight Cryptographic Protocols (Al Hayajneh et al., 2021)	ECC-based lightweight encryption	Uses elliptic-curve cryptography for secure communication in wearables and sensors.	Low computational cost; energy-efficient.	Protects only communication, not analytics or trust.
Edge-Assisted IoMT Security (Khan et al., 2020)	Edge/Fog computing	Offloads IoMT computation to edge servers for secure real-time analytics.	Reduces latency; enhances responsiveness.	Edge nodes may become central trust points.
Federated IoMT Framework (Ghosh & Ghosh, 2023)	FL	Collaborative model training across IoMT devices without raw data exchange.	Preserves privacy; enables cross-client learning.	Device heterogeneity reduces convergence efficiency.
Zero-Trust IoMT Security Model (Almuseelem, 2025)	Zero-Trust Architecture	Applies continuous authentication and micro-segmentation in IoMT networks.	Fine-grained access control; minimizes insider threats.	High management complexity; may impact latency.
Hybrid Edge-Cloud IoMT Security (Khan et al., 2025)	Hybrid edge-cloud with ML	Uses a hybrid edge-cloud setup to process and secure IoMT data streams with ML classifiers.	Balances resource use between edge and cloud.	Potential single point of failure at cloud; privacy risks.
Blockchain-FL Healthcare Framework (Singh et al., 2022)	Blockchain + FL	Integrates FL and blockchain for secure healthcare analytics across IoMT hospitals.	Combines privacy preservation with trust management.	Limited analysis of device-level constraints and scalability.
AI-Enhanced IoMT Security (Srivastava et al., 2025)	AI anomaly detection	Uses deep learning models to detect intrusions in IoMT device traffic.	Strong detection accuracy; adaptive learning.	Requires high computational resources; vulnerable to adversarial attacks.
Proposed FLIT	Blockchain + FL + Privacy Enhancements	Combines blockchain-based trust with FL, differential privacy, gradient clipping, and secure aggregation for robust IoMT healthcare data sharing.	Ensures end-to-end security, scalability, and trust management across heterogeneous IoMT environments.	Increased computation at aggregators, higher communication cost, and energy limits on IoMT devices.

end of the process. This growth reflects efficient handling of aggregated updates and optimized contract execution so that FLIT maintains high scalability even under frequent communication loads in federated IoMT environments.

Evaluation of Key Performance Dimensions. In assessing the overall performance of the FLIT framework, five key aspects are considered: security, privacy, FL efficiency, data sharing, and real-time access control. Security is provided through blockchain-based encryption, tamper-proof smart contracts, and decentralized data exchange, which protect patient records from unauthorized access and single points of failure. Privacy is preserved using strong encryption, data fragmentation, and role-based access rules enforced by smart contracts to limit unnecessary exposure. FL efficiency is shown through accurate and timely predictive modeling for disease risk and treatment recommendations, with distributed training across healthcare providers enabling fast model convergence. Data sharing is evaluated by how well the system delivers accurate insights and allows seamless, trustworthy exchange of healthcare information. Real-time access control is achieved by using smart contracts to instantly apply access rules and enable secure, timely retrieval of data in line with clinical needs. Together, these features demonstrate that FLIT can deliver a secure, privacy-preserving, efficient, and patient-focused platform for healthcare data sharing and analytics.

7.5. Justification of assumptions in iomt settings

The assumptions adopted in FLIT are not only theoretically motivated but also reflect realistic conditions in IoM environments. Below, we justify their applicability in practical deployments.

Semi-Trusted Participants. Healthcare institutions, diagnostic laboratories, and pharmacies participating in IoMT ecosystems are regulated entities bound by frameworks such as HIPAA and GDPR. While these organizations generally comply with protocols, they may still attempt to extract sensitive knowledge from shared updates. Hence, the honest-but-curious model is realistic. FLIT strengthens this assumption by enforcing blockchain-based identity verification and embedding differential privacy in every update to prevent data leakage.

Immutable Ledger. Blockchain immutability is already widely leveraged in healthcare pilots and consortia for EHR sharing, pharmaceutical supply chains, and insurance auditing. In these contexts, a permissioned blockchain with validator nodes operated by hospitals, insurers, and regulators ensures tamper-proof record keeping. Thus, the assumption of an immutable ledger is not merely theoretical but aligns with real-world compliance requirements for auditability and transparency.

Localized Federated Training. Hospitals and clinics are prohibited from sharing raw patient data across institutions due to strict privacy regulations. FL has therefore been adopted in medical research (e.g., for COVID-19 diagnosis and multi-hospital EHR prediction studies), where local-only training is the only feasible solution. FLIT builds on this existing practice, making the assumption of localized training both practical and legally compliant.

Secure Communication Channels. TLS/SSL is already the industry standard for securing health data transfers across networks, and many hospitals enforce VPNs at the device level. FLIT extends this widely adopted practice by adding client-specific encryption keys and anchoring references in IPFS and blockchain. This ensures data confidentiality and authenticity throughout the update pipeline, consistent with current IoMT security deployments.

Bounded Adversarial Collusion. In regulated healthcare environments, large-scale adversarial collusion is highly unlikely since stakeholders are trusted institutions with reputational and legal accountability. The assumption of bounded collusion ($f < \lfloor (N-1)/3 \rfloor$) is standard in blockchain protocols such as PBFT and PoA, and it is equally applicable to FL environments. FLIT further mitigates this threat by incorporating trust-weighted aggregation to down-weight anomalous or malicious participants dynamically.

7.6. Limitation of this work and future direction

Our experiment primarily concentrated on the utilization of blockchain for sharing data and models within the context of FL. Our prototype still runs in a controlled testbed, not a nationwide IoMT deployment, so scalability, network churn, and cross-hospital interoperability remain partly untested. Privacy relies on DP and encryption hyperparameters that trade utility for protection and may need careful tuning per task. Trust scores and anomaly filters assume mostly honest behavior and can mis-rank clever attackers. Finally, secure off-chain aggregation and key management introduce operational complexity and a residual trust surface that future fully decentralized MPC/TEE deployments need to shrink.

7.7. Deployment challenges in heterogeneous IoMT environments

Although the experimental evaluation confirms the feasibility of FLIT. The deployment of it in real-world IoMT settings introduces several challenges that must be considered.

Latency. Medical IoT devices often operate under strict time constraints, particularly for applications such as continuous monitoring or emergency response. Blockchain consensus and smart contract execution can introduce additional delays. FLIT mitigates this by adopting lightweight sidechains and batching updates off-chain, but real-world deployments must still balance security with time-sensitive responsiveness.

Device Constraints. IoMT nodes, including wearables and implantable sensors, are resource-constrained in terms of computation, memory, and battery life. Running FL updates and cryptographic operations directly on such devices may be infeasible. In practice, lightweight edge gateways or hospital servers can offload these operations, while devices focus on data capture and minimal preprocessing.

Interoperability. Healthcare IoMT networks are highly heterogeneous, spanning devices from multiple vendors, protocols, and data standards (Regalado & Han, 2025). This diversity poses challenges for seamless integration into a federated framework. FLIT addresses interoperability by separating local training logic from blockchain coordination and by anchoring only metadata and model hashes on-chain. However, widespread adoption will require adherence to common healthcare data exchange standards and continued integration efforts. While FLIT provides architectural features that alleviate latency, device limitations, and interoperability gaps, these challenges highlight the need for gradual deployment strategies and collaboration with healthcare infrastructure providers.

8. Conclusion

The need for the proposed FLIT becomes evident in the face of critical challenges surrounding data security, privacy, and seamless communication in the healthcare sector. The core tenets of this framework involve encryption, data fragmentation, distribution among trusted peers, and safeguarding the confidentiality and integrity of sensitive patient information. Moreover, it efficiently manages access control through smart contracts, granting authorized entities real-time access to specific data based on predefined conditions, facilitating a balance between privacy and collaboration among healthcare providers and patients. The integration of FL into FLIT amplifies the quality of healthcare recommendations and insights. FL can further strengthen FLIT's commitment to patient privacy and data security by enabling a decentralized network of healthcare providers to collaboratively train models. The experiment results provide empirical evidence that FLIT is a robust and effective framework for sharing healthcare data and models securely. As for future work, we aim to enhance the creation of predictive models that can significantly enhance the quality of healthcare recommendations and insights. Furthermore, this avenue of research presents an exciting prospect for advancing the capabilities of FLIT and solidifying its position as a secure and patient-centered platform for healthcare data sharing.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the auhtor(s) used Grammarly and QuillBot tools in order to improve the readability and language of the manuscript. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Akkal, M., Cherbal, S., Annane, B., & Lakhlef, H. (2025). Btmh: A blockchain-powered trust management system for iomt in healthcare. *Computer Networks*, (p. 111589).

Ali, M., Karimipour, H., & Tariq, M. (2021). Integration of blockchain and federated learning for internet of things: Recent advances and future challenges. *Computers & Security*, *108*, 102355.

Almuseelem, W. (2025). Secure latency-aware task offloading using federated learning and zero trust in edge computing for ioMT. *IEEE Access*, *13*, 117808–117830. https://doi.org/10.1109/ACCESS.2025.3586730

Barreau, D., Capra, R., Dumais, S., Jones, W., & Pérez-Quiñones, M. (2008). Introduction to keeping, refinding and sharing personal information. *ACM Transactions on Information Systems*, *26*(4). https://doi.org/10.1145/1402256.1402257

Bhartiya, S., & Mehrotra, D. (2014). Challenges and recommendations to healthcare data exchange in an interoperable environment. *Electronic Journal of Health Informatics*, *8*(2), 16.

Bhushan, B., Kumar, A., Agarwal, A.K., Kumar, A., Bhattacharya, P., & Kumar, A. (2023). Towards a secure and sustainable internet of medical things (ioMT): Requirements, design challenges, security techniques, and future trends. *Sustainability*, *15*(7), 6177.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 1175–1191).

Bonifati, A., Chrysanthis, P.K., Ouksel, A.M., & Sattler, K.U. (2008). Distributed databases and peer-to-peer databases: Past and present. *ACM SIGMOD Record*, *37*(1), 5–11.

Borozdina, E., & Novkunskaya, A. (2022). Patient-centered care in russian maternity hospitals: Introducing a new approach through professionals' agency. *Health*, *26*(2), 200–220.

Cangir, O.F., Cankur, O., & Ozsoy, A. (2021). A taxonomy for blockchain based distributed storage technologies. *Information processing & management*, *58*(5), 102627.

Cerchione, R., Centobelli, P., Riccio, E., Abbate, S., & Oropallo, E. (2023). Blockchain's coming to hospital to digitalize healthcare services: Designing a distributed electronic health record ecosystem. *Technovation*, *120*, 102480.

Chen, J., Xue, J., Wang, Y., Huang, L., Baker, T., & Zhou, Z. (2023). Privacy-preserving and traceable federated learning for data sharing in industrial iot applications. *Expert Systems with Applications*, *213*, 119036.

Clore, John, C.K.D.J, & Strack, B., (2014). Diabetes 130-US hospitals for years 1999–2008. UCI Machine Learning Repository. https://doi.org/10.24432/C5230J.

Das, D., Banerjee, S., & Biswas, U. (2021a). A secure vehicle theft detection framework using blockchain and smart contract. *Peer-to-Peer Networking and Applications*, *14*, 672–686.

Das, D., Banerjee, S., Chatterjee, P., Biswas, M., Biswas, U., & Alnumay, W. (2022). Design and development of an intelligent transportation management system using blockchain and smart contracts. *Cluster computing*, *25*(3), 1899–1913.

Das, D., Banerjee, S., Chatterjee, P., Ghosh, U., & Biswas, U. (2023). A secure blockchain enabled v2v communication system using smart contracts. *IEEE Transactions on Intelligent Transportation Systems*, *24*(4), 4651–4660. https://doi.org/10.1109/TITS.2022.3226626

Das, D., Banerjee, S., Chatterjee, P., Ghosh, U., Mansoor, W., & Biswas, U. (2021b). Design of a blockchain enabled secure vehicle-to-vehicle communication system. In *2021 4th International conference on signal processing and information security (ICSPIS)* (pp. 29–32). IEEE.

Das, D., Banerjee, S., Ghosh, U., Biswas, U., & Bashir, A.K. (2021c). A decentralized vehicle anti-theft system using blockchain and smart contracts. *Peer-to-Peer Networking and Applications*, *14*, 2775–2788.

Das, D., Banerjee, S., Mansoor, W., Biswas, U., Chatterjee, P., & Ghosh, U. (2020). Design of a secure blockchain-based smart iov architecture. In *2020 3rd International conference on signal processing and information security (ICSPIS)* (pp. 1–4). https://doi.org/10.1109/ICSPIS51252.2020.9340142

Das, D., Chatterjee, P., Banerjee, S., Ghosh, U., & Al-Numay, M.S. (2025). Blockchain-enabled federated learning for security and privacy in consumer electronics devices. *IEEE Transactions on Consumer Electronics*, *71*(1), 2262–2270. https://doi.org/10.1109/TCE.2025.3528934

Dey, T., Bera, S., Mukherjee, A., De, D., & Buyya, R. (2025). Flyer: Federated learning-based crop yield prediction for agriculture 5.0. *IEEE Transactions on Artificial Intelligence*, *6*(7), 1943–1952. https://doi.org/10.1109/TAI.2025.3534149

Dong, X., Randolph, D.A., Weng, C., Kho, A. N., Rogers, J.M., & Wang, X. (2021). Developing high performance secure multi-party computation protocols in healthcare: A case study of patient risk stratification. *AMIA Summits on Translational Science Proceedings*, *2021*, 200.

Draidi, F., Pacitti, E., & Kemme, B. (2011). P2Prec: A P2P Recommendation System for Large-Scale Data Sharing. In A. Hameurlain, J. Küng, & R.Wagner, (Eds.), *Transactions on Large-Scale Data- and Knowledge-Centered Systems III: Special Issue on Data and Knowledge Management in Grid and P2P Systems*, (pp. 87–116). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-23074-5_4

Du, M., Chen, Q., Chen, J., & Ma, X. (2020). An optimized consortium blockchain for medical information sharing. *IEEE Transactions on Engineering Management*, *68*(6), 1677–1689.

Ghosh, S., & Ghosh, S.K. (2023). Feel: Federated learning framework for elderly healthcare using edge-iomt. *IEEE Transactions on Computational Social Systems*, *10*(4), 1800–1809.

Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S., & Zhang, X. (2022). Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, *54*(11s), 1–37.

Javaid, M., Haleem, A., Singh, R.P., & Suman, R. (2023). Towards insighting cybersecurity for healthcare domains: A comprehensive review of recent practices and trends. *Cyber Security and Applications*, *1*, (p. 100016). https://doi.org/10.1016/j.csa.2023.100016

Javed, I.T., Alharbi, F., Bellaj, B., Margaria, T., Crespi, N., & Qureshi, K.N. (2021). Health-ID: A blockchain-based decentralized identity management for remote healthcare. In *Healthcare* (p. 712). MDPI (*9*).

Javed, L., Anjum, A., Yakubu, B.M., Iqbal, M., Moqurrab, S.A., & Srivastava, G. (2023). Sharechain: Blockchain-enabled model for sharing patient data using federated learning and differential privacy. *Expert Systems*, *40*(5), e13131.

Jedlicka, J., & Grant, E.S. (2022). Data privacy through zero-knowledge proofs. In *2022 Fourth international conference on emerging research in electronics, computer science and technology (ICERECT)* (pp. 1–7). IEEE.

Kaveh, S., Ebrahimzadeh, F., & Safa, R. (2025). Leveraging blockchain and iot for secure and scalable healthcare innovations. *Annals of Healthcare Systems Engineering*, *2*(1), 16–27. https://doi.org/10.22105/ahse.v2i1.27

Khalid, N., Qayyum, A., Bilal, M., Al-Fuqaha, A., & Qadir, J. (2023). Privacy-preserving artificial intelligence in healthcare: Techniques and applications. *Computers in Biology and Medicine*, *158*, (p. 106848). https://doi.org/10.1016/j.compbiomed.2023.106848

Khan, A.A., Laghari, A.A., Alroobaea, R., Baqasah, A.M., Alsafyani, M., Alsufyani, H., & Ullah, S. (2025). A lightweight scalable hybrid authentication framework for internet of medical things (ioMT) using blockchain hyperledger consortium network with edge computing. *Scientific Reports*, *15*(1), 19856.

Khan, M.N., Rao, A., & Camtepe, S. (2020). Lightweight cryptographic protocols for iot-constrained devices: A survey. *IEEE Internet of Things Journal*, *8*(6), 4132–4156.

Kumar, M., Kumar, A., Verma, S., Bhattacharya, P., Ghimire, D., Kim, S.h., & Hosen, A.S. (2023). Healthcare internet of things (h-iot): Current trends, future prospects, applications, challenges, and security issues. *Electronics*, *12*(9), 2050.

Lu, Y., Huang, X., Dai, Y., Maharjan, S., & Zhang, Y. (2020). Blockchain and federated learning for privacy-preserved data sharing in industrial iot. *IEEE Transactions on Industrial Informatics*, *16*(6), 4177–4186. https://doi.org/10.1109/TII.2019.2942190

Mallick, S.R., Sobhanayak, S., & Lenka, R.K. (2024). Blockchain-enhanced iot ecosystem for healthcare: Transformative potentials, applications, challenges, solutions, and future perspectives. *Computers & Industrial Engineering*, *197*, 110538. https://doi.org/10.1016/j.cie.2024.110538

Masud, M., & Hossain, M.S. (2018). Secure data-exchange protocol in a cloud-based collaborative health care environment. *Multimedia Tools and Applications*, *77*, 11121–11135.

Meenigea, N., & Kolla, V.R.K. (2023). Exploring the current landscape of artificial intelligence in healthcare. *International Journal of Sustainable Development in Computing Science*, *5*(1), 1–10.

Mistry, C., Thakker, U., Gupta, R., Obaidat, M.S., Tanwar, S., Kumar, N., & Rodrigues, J.J.P.C. (2021). Medblock: An AI-enabled and blockchain-driven medical healthcare system for COVID-19. In *ICC 2021 - IEEE international conference on communications* (pp. 1–6). https://doi.org/10.1109/ICC42927.2021.9500397

Murugan, A., Chechare, T., Muruganantham, B., & Kumar, S.G. (2020). Healthcare information exchange using blockchain technology. *International Journal of Electrical and Computer Engineering*, *10*(1), 421.

Panigrahi, A., Nayak, A.K., & Paul, R. (2022). Healthcare EHR: A blockchain-based decentralized application. *International Journal of Information Systems and Supply Chain Management (IJISSCM)*, *15*(3), 1–15.

Raj, A., & Prakash, S. (2024). Privacy preservation of the internet of medical things using blockchain. *Health Services and Outcomes Research Methodology*, *24*(1), 112–139 (pp. 1–28).

Ray, P.P., Dash, D., & De, D. (2019). Edge computing for internet of things: A survey, e-healthcare case study and future direction. *Journal of Network and Computer Applications*, *140*, 1–22. https://doi.org/10.1016/j.jnca.2019.05.005

Regalado, P.H., & Han, T. (2025). Mediverse: A secure and scalable iot-MR framework for real-time health and performance monitoring. *IEEE Access*, *13*, 97511–97528. https://doi.org/10.1109/ACCESS.2025.3570731

Regueiro, C., Seco, I., de Diego, S., Lage, O., & Etxebarria, L. (2021). Privacy-enhancing distributed protocol for data aggregation based on blockchain and homomorphic encryption. *Information Processing & Management*, *58*(6), 102745.

Saad, M., Spaulding, J., Njilla, L., Kamhoua, C., Shetty, S., Nyang, D., & Mohaisen, A. (2019). Exploring the attack surface of blockchain: A systematic overview. arXiv:1904.03487.

Sallam, M. (2023). ChatGPT utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns. In *Healthcare* (p. 887). MDPI (*11*).

Sendhil, R., & Amuthan, A. (2021). Contextual fully homomorphic encryption schemes-based privacy preserving framework for securing fog-assisted healthcare data exchanging applications. *International Journal of Information Technology*, *13*(4), 1545–1553.

Shonchoy, A.S., Mahzab, M.M., Mahmood, T.I., & Ali, M. (2023). Data driven contagion risk management in low-income countries using machine learning applications with COVID-19 in south asia. *Scientific Reports*, *13*(1), 3732.

Singh, A.K., Anand, A., Lv, Z., Ko, H., & Mohan, A. (2021). A survey on healthcare data: A security perspective. *ACM Transactions on Multimidia Computing Communications and Applications*, *17*(2s), 1–26.

Singh, S., Rathore, S., Alfarraj, O., Tolba, A., & Yoon, B. (2022). A framework for privacy-preservation of iot healthcare data using federated learning and blockchain technology. *Future Generation Computer Systems*, *129*, 380–388.

Srivastava, S., Kansal, K., Sai, S., & Chamola, V. (2025). Secure cognitive health monitoring using a directed acyclic graph-based and ai-enhanced iomt framework. *Digital Communications and Networks*, *11*(3), 594–602.

Stoumpos, A.I., Kitsios, F., & Talias, M.A. (2023). Digital transformation in healthcare: Technology acceptance and its applications. *International Journal of Environmental Research and Public Health*, *20*(4), 3407.

Tigelaar, A.S., Hiemstra, D., & Trieschnigg, D. (2012). Peer-to-peer information retrieval: An overview. *ACM Transactions on Information Systems (TOIS)*, *30*(2), 1–34.

Venugopal, S., Buyya, R., & Ramamohanarao, K. (2006). A taxonomy of data grids for distributed data sharing, management, and processing. *ACM Computing Surveys (CSUR)*, *38*(1), 3–es.

Ahmed, I., Chehri, A., & Jeon, G. (2023). Artificial intelligence and blockchain enabled smart healthcare system for monitoring and detection of covid-19 in biomedical images. *IEEE/ACM transactions on computational biology and bioinformatics*, *21*(4), 814–822.

Wang, W., Li, X., Qiu, X., Zhang, X., Brusic, V., & Zhao, J. (2023). A privacy preserving framework for federated learning in smart healthcare systems. *Information Processing & Management*, *60*(1), 103167. https://doi.org/10.1016/j.ipm.2022.103167

Wang, Z., Ma, J., Wang, X., Hu, J., Qin, Z., & Ren, K. (2022). Threats to training: A survey of poisoning attacks and defenses on machine learning systems. *ACM Computing Surveys*, *55*(7), 1–36.

Woodside, J.M. (2007). Edi and erp: A real-time framework for healthcare data exchange. *Journal of Medical Systems*, *31*, 178–184.

Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., & Wang, F. (2021). Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, *5*, 1–19.

Yang, F., Qiao, Y., Abedin, M.Z., & Huang, C. (2022). Privacy-preserved credit data sharing integrating blockchain and federated learning for industrial 4.0. *IEEE Transactions on Industrial Informatics*, *18*(12), 8755–8764. https://doi.org/10.1109/TII.2022.3151917

Dr. Debashis Das is currently working as a postdoctoral fellow in the Department of Computer Science and Data Science in the School of Applied Computational Sciences at Meharry Medical College. He was recognized among the world's top 2 percent of Scientists in 2025 by Stanford University and Elsevier. He received his Ph.D. in Computer Science and Engineering from the University of Kalyani, India, in 2023. He is a member of IEEE. Dr. Das has more than 50 publications in various peer-reviewed journals and conferences and has over 950 citations, with an h-index of 19 and an i10-index of 30 by Google Scholar. He is a reviewer for various peer-reviewed journals of the IEEE Transactions, Elsevier, and Springer. His research interests include cybersecurity, blockchain technology, and artificial intelligence. He has served as an invited TPC member or chair for numer-

ous international conferences, including CICBA, BCCA, CCGRID, ICDCN, IEEE STP-CPS, ICSPIS, ISORC, and IoST.

Dr. Sourav Banerjee is a senior member of IEEE. He received a Ph.D. degree in Computer Science and Engineering from the University of Kalyani in 2018. He completed his B. E in Computer Science and Engineering and M.Tech in Computer Science and Engineering. He is a permanent faculty member in the Department of Computer Science and Engineering of Kalyani Government Engineering College at Kalyani, West Bengal, India. He has authored a good number of reputed Journal articles, Book, Book chapters, and international conferences. His research interests include Big Data, IoT, IoMT, Zero-trust architecture, Sustainable engineering, Blockchain, Cloud Computing, Green Cloud Computing, Green AI, 6G, Cloud Robotics, Distributed Computing, and Mobile Communications, Smart City, Global Warming. He is a member of ACM, IAE. He is a SIG member of MIR Lab, USA. He is an Editorial board member of Sustainability Journal, Wireless Communication Technology. He has performed as invited TPC member, Coordinator for so many international conferences, like CICBA , Globecom, IEEE STP-CPS, ICSPIS, ICRCICN, etc.

Dr. Debashis De is a Professor in the Department of Computer Science and Engineering at Maulana Abul Kalam Azad University of Technology, West Bengal, India. He is a Senior Member of IEEE, Fellow of IETE, Fellow of NBSP and Life Member of CSI, ACM, IEI, and Chartered Engineer. He has received prestigious awards such as the Boyscast Fellowship (India) to work at Heriot Watt University Scotland, Edinburgh, Endeavour Fellowship (Australia) from deputy Prime minister of Australia for his Postdoc at University of Western Australia, Perth, and URSI Young Scientist Award (2005) from president of India APJ KALAM at New Delhi and 2011 at Istanbul, TURKEY. He was honored with the JC Bose Research Award IETE New Delhi (2016). Shiksha-Ratna Award (2019) by Govt of West Bengal and the IETE Chartered Engineer for his contributions. He has led research projects funded by AICTE, UGC, DST, DST-FIST, World Bank, TEQIP I and TEQIP II coordinator, AMAZON WEB Services, and MeitY with total budget Twenty (20) crore. His h-index is 49 with 11,800 citations, and he ranks among the top 2 % of global scientists, Standford, USA and Ranked 122 in India by Research.com. His research focuses on Edge Intelligence, Federated learning , Sustainable computing, Internet of Things, and Quantum Computing.