

Article

A Federated Learning Framework for Privacy-Preserving Patient Monitoring with Lightweight Blockchain Anchoring

Thattapon Surasak ¹, Kou Yamada ^{2,*} and Jirayu Samkunta ^{1,*}
¹ Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand; thattapon.s@sci.kmutnb.ac.th

² Graduate School of Science and Technology, Gunma University, 1-5-1 Tenjin-cho, Kiryu 376-8515, Gunma, Japan

* Correspondence: yamada@gunma-u.ac.jp (K.Y.); jirayu.s@sci.kmutnb.ac.th (J.S.)

Abstract

This paper proposes a federated learning framework for privacy-preserving patient monitoring with lightweight blockchain anchoring. The framework keeps synthetic patient monitoring records local at each client and uses federated model aggregation to support collaborative learning without centralizing raw records. To improve traceability, the blockchain layer is specified as an anchoring mechanism that records compact evidence, including model hashes and participation metadata, rather than raw data or full model parameters. Experiments were conducted on synthetic patient monitoring data to evaluate framework behavior under non-IID client distributions, label noise, different client counts, partial client participation, and aggregation strategies. The centralized MLP baseline achieved approximately 0.89 overall accuracy but failed to detect alert cases, with 0% alert-class recall, showing that accuracy alone can be misleading in imbalanced monitoring scenarios. In the federated simulations, the model reached approximately 0.99 accuracy under clean labels, approximately 0.90 under 10% label noise, and approximately 0.70 under 30% label noise. Under a more difficult noisy, non-IID, dropout, and fixed skewed-client evaluation setting, the model stabilized at approximately 0.80 accuracy after 25 communication rounds. Client scaling from 5 to 20 clients remained stable, and FedAvg, weighted aggregation, and accuracy-trimmed robust aggregation produced similar final accuracy of approximately 0.98 in the 10-client setting. The results indicate that label quality strongly affects federated convergence, while blockchain anchoring should be interpreted as an auditability mechanism rather than a direct accuracy-improving component. This study provides a framework-level foundation for auditable federated patient monitoring in semi-trusted healthcare networks.

Keywords: blockchain anchoring; federated learning; healthcare privacy; patient monitoring; model provenance; auditability



Academic Editor: Nik Bessis

Received: 21 May 2026

Revised: 5 July 2026

Accepted: 10 July 2026

Published: 16 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Healthcare systems are increasingly moving toward continuous and distributed patient monitoring. Wearable sensors, home monitoring devices, mobile health platforms, and hospital information systems can now generate frequent physiological measurements that support early warning detection, chronic disease management, and timely clinical decision-making. These developments create an opportunity for data-driven patient monitoring models that can learn from diverse healthcare environments. However, they also create a

practical dilemma. The data needed to train reliable models are often distributed across institutions and devices, while the direct centralization of patient data raises concerns related to privacy, data ownership, institutional governance, and regulatory compliance.

Federated learning has become a promising response to this dilemma because it enables collaborative model training without requiring raw data to leave local sites. Each participant trains a model locally and shares model updates for aggregation, allowing a global model to benefit from distributed data while preserving data locality. This property has motivated its use in remote patient monitoring, electronic health record sharing, Internet of Medical Things data exchange, and medical image analysis [1–5]. In healthcare settings, however, keeping data local is only one part of the problem. A practical patient monitoring system must also handle heterogeneous data distributions, noisy labels, unreliable client participation, and the need to verify how the model evolves over time.

These issues are especially important in patient monitoring because healthcare clients rarely behave as clean and identical data sources. Different hospitals may serve different patient populations. Wearable devices may capture signals under different usage conditions. Local monitoring datasets may contain different proportions of normal and abnormal cases. Labels may also be noisy because clinical states can be uncertain, delayed, or inconsistently recorded. Under these conditions, a federated model may converge differently depending on the number of clients, the degree of label noise, and the aggregation strategy used. Therefore, federated learning for healthcare monitoring should be evaluated not only under clean settings, but also under controlled forms of non-IID data, noisy labels, client scaling, and aggregation variation.

Another unresolved issue is trust in the federated learning process. Conventional federated learning normally depends on a coordinator that receives local updates, performs aggregation, and redistributes the global model. This coordinator improves efficiency, but it also introduces a trust assumption. Participating institutions may have limited visibility into which clients participated, which model version was produced, and whether a particular update history can be verified later. In clinical and institutional environments, this lack of auditability can limit adoption even when raw data remain local.

Blockchain technology has therefore been investigated as a complementary mechanism for federated learning. Its immutability and tamper-evident record structure can support model provenance, participation logging, decentralized verification, and post hoc audit. Existing blockchain-based federated learning systems have explored smart contracts, secure aggregation, zero-knowledge proofs, self-sovereign identity, and post-quantum signatures to improve trust and security [6–8]. However, many of these mechanisms introduce substantial complexity. For lightweight patient monitoring, full on-chain aggregation or heavy cryptographic verification may not always be necessary. A more practical approach is to use blockchain as an anchoring layer that records compact evidence, such as model hashes and participation proofs, while keeping model training and aggregation off-chain.

Despite the growing literature on federated learning and blockchain-enabled healthcare systems, several gaps remain in the specific context of privacy-preserving patient monitoring. Many healthcare federated learning studies focus on data locality and predictive performance but provide limited mechanisms for lightweight auditability of model evolution and client participation. Many blockchain-based healthcare systems rely on complex security stacks, such as homomorphic encryption, secure multi-party computation, zero-knowledge proofs, or full smart-contract-based aggregation. These designs are valuable for high-assurance settings, but they may be difficult to justify in patient monitoring scenarios where latency, simplicity, and maintainability are also important. Existing works also do not consistently evaluate the combined effects of non-IID data, label noise, client scaling, and aggregation strategy selection within the same experimental workflow. This is

a critical limitation because these factors are likely to occur together in practical healthcare monitoring environments.

This study proposes a federated learning framework for privacy-preserving patient monitoring with lightweight blockchain anchoring. The framework is designed for semi-trusted healthcare environments where raw patient data remain local, model learning is performed collaboratively, and blockchain is used to record compact evidence of model evolution and client participation. The contribution of this work lies in connecting three system concerns that are often treated separately: federated learning for data locality, aggregation strategies for learning stability under heterogeneous and noisy client conditions, and blockchain anchoring for tamper-evident auditability. The proposed framework evaluates federated learning behavior under non-IID client distributions, controlled label noise, different client population sizes, and multiple aggregation strategies, including FedAvg, weighted aggregation, and robust aggregation. It then introduces a lightweight anchoring mechanism that records model hashes and participation proofs without storing raw patient data or full model parameters on-chain. This design does not claim to replace formal privacy mechanisms such as differential privacy, secure aggregation, or secure multi-party computation. Instead, it addresses a specific and practical trust gap in federated healthcare deployment: how to preserve local patient data while maintaining an auditable record of model training and client participation.

The remainder of this paper is organized as follows. Section 2 reviews related work on federated learning, privacy-preserving healthcare systems, robust aggregation, and blockchain-based auditability. Section 3 presents the proposed system architecture and explains the role of each layer. Section 4 describes the methodology, including synthetic patient data generation, non-IID simulation, label noise injection, federated learning procedures, aggregation strategies, and evaluation metrics. Section 5 reports the experimental results. Section 6 details the lightweight blockchain anchoring design. Section 7 discusses implications, limitations, and future work. Finally, Section 8 concludes the paper.

2. Related Work

The integration of federated learning and blockchain has become an important research direction for healthcare systems that must balance privacy, model utility, trust, and auditability. Healthcare data are naturally fragmented across hospitals, clinics, edge devices, wearable sensors, and remote monitoring platforms. These data are also sensitive and strongly regulated, making centralized model training difficult in practice. Federated learning addresses part of this challenge by allowing each participant to train locally and share model updates instead of raw data. However, federated learning alone does not provide a complete trust model. Model updates may still leak information, clients may submit unreliable or malicious updates, and the aggregation process may lack transparent evidence for later audit. Blockchain has therefore been explored as a complementary mechanism for immutable logging, decentralized verification, incentive management, identity control, and model provenance.

The literature can be understood as a progression from privacy-preserving distributed learning toward more accountable and verifiable collaborative intelligence. Early federated learning studies focused mainly on keeping data local. More recent studies have expanded the problem to include non-IID data, noisy labels, poisoning attacks, client dropout, communication cost, and verifiable computation. In healthcare, this progression is particularly important because technical performance alone is not enough. A clinically useful system must also provide evidence of participation, model version history, and protection against operational misuse. This section reviews the literature according to this progression and identifies the gap addressed by the proposed framework.

2.1. Federated Learning for Healthcare and Remote Patient Monitoring

Federated learning is well suited to healthcare because it supports collaborative model training while preserving institutional data ownership. In remote patient monitoring, this is especially relevant because physiological data are often collected continuously from wearable devices, home sensors, or edge gateways. These data streams can support early warning detection and personalized care, but direct centralization may conflict with privacy requirements and increase exposure to data breaches. Recent work has therefore moved toward federated designs that keep patient data local while enabling shared model improvement.

A strong example in the remote monitoring context is MedGuard-FL, which frames federated learning as a risk-aware and adaptive security system for e-healthcare environments. Rather than treating privacy as a fixed setting, the framework adjusts privacy and robustness mechanisms according to patient status and system risk. This is important because remote monitoring systems do not operate under constant clinical conditions. A patient in a stable state and a patient in a high-risk alert state may require different privacy, latency, and robustness trade-offs [1]. This perspective is useful for the present study because it confirms that patient monitoring should not be evaluated only as a static classification problem. The learning framework must also be understood as part of a healthcare control environment where data quality, responsiveness, and trust all matter.

Electronic health record sharing represents another healthcare setting where federated learning is relevant but insufficient by itself. In an AI-driven blockchain and federated learning framework for secure EHR sharing, federated distillation is combined with dynamic differential privacy, threshold key sharing, and Quorum blockchain audit trails [2]. This shows that cross-institutional healthcare learning requires more than local training. It also needs access control, auditability, and privacy adaptation. However, EHR sharing frameworks usually involve heavier infrastructure and stronger institutional assumptions than lightweight patient monitoring. The present work takes a narrower operational target by focusing on model training behavior and lightweight anchoring rather than a full EHR-sharing stack.

Medical image analysis has also been a major application area for healthcare federated learning. Explainable federated medical image analysis has used causal learning to reduce communication overhead while improving interpretability, and blockchain has been used to assess client data quality during aggregation [4]. This line of work is important because it shows that blockchain can be used not merely for storage, but also as a trust signal for model aggregation. Another medical imaging study applies blockchain-verified federated learning to brain tumor classification using colorized MRI scans and records model hashes, accuracy, and timestamps through smart contracts [5]. This is close to the auditability idea in the present work, although the application context differs. Medical imaging models are typically larger and task-specific, while patient monitoring often requires lighter models and lower operational overhead.

The healthcare literature therefore supports three broad conclusions. First, federated learning is a practical foundation for privacy-preserving healthcare AI because it reduces raw data movement. Second, healthcare deployments require mechanisms beyond privacy, including trust, traceability, and verification. Third, many existing healthcare systems either focus on high-complexity clinical data such as EHRs and medical images or introduce heavy security mechanisms that may not be necessary for lightweight patient monitoring. These observations motivate a framework that is simpler, auditable, and explicitly evaluated under noisy and heterogeneous monitoring conditions.

2.2. Heterogeneity, Label Noise, and Client Reliability in Federated Learning

A persistent challenge in federated learning is that client data are rarely independent and identically distributed. In healthcare, this issue is unavoidable. Hospitals serve different populations, wearable devices capture different user behaviors, and local monitoring datasets may contain different proportions of normal and abnormal cases. Under such heterogeneity, a global model trained through simple averaging may converge slowly or become biased toward dominant client distributions.

Recent studies have increasingly recognized that non-IID data must be considered together with robustness. A privacy-preserving poisoning-resistant blockchain-based federated learning scheme for IoMT data sharing directly addresses the tension between privacy and malicious-gradient detection. The framework uses threshold secret sharing and secure multi-party computation to protect privacy, while applying cosine similarity verification to detect harmful gradients [3]. This is relevant because it shows that privacy protection and poisoning resistance can conflict. If updates are strongly hidden, malicious updates become harder to verify. Therefore, robust healthcare FL requires a careful balance between confidentiality and model integrity.

Trustworthy federated learning schemes have also used encryption and robust client selection to handle both inference and poisoning threats. A blockchain-based trustworthy FL privacy protection scheme combines a threshold Paillier variant with a flexible Multi-Krum selection mechanism [9]. The significance of this work lies in its attempt to protect both local and global model updates while still filtering suspicious contributions. However, such approaches increase cryptographic and verification complexity. For a lightweight patient monitoring framework, the practical question is whether full encryption and robust selection are always necessary, or whether a simpler audit layer can still improve accountability when the deployment is semi-trusted.

Adaptive differential privacy has been proposed as another method for improving the privacy-utility balance in federated learning. DFLChain introduces adaptive Gaussian noise attenuation, with stronger noise in earlier training stages and reduced noise as the model converges [10]. This is a useful design principle because static privacy noise can unnecessarily damage model utility. However, adaptive DP also requires explicit privacy parameters, noise mechanisms, and accounting. This point is important for revising the current manuscript because any claim about differential privacy must be supported by specific values and mechanisms rather than mentioned only as a general concept.

Client dropout is another practical issue. Healthcare clients may fail to participate because of network instability, scheduling policies, device power limitations, or institutional constraints. BAHA-FL addresses dropout resilience by using blockchain-based adaptive historical averaging with exponential decay, allowing historical updates to compensate for missing current updates [11]. This work is useful because it treats dropout as an operational reliability problem rather than only a communication issue. Similarly, edge-aware federated learning architectures emphasize fault tolerance, asynchronous optimization, hierarchical aggregation, and blockchain-enhanced trust in resource-constrained edge environments [12]. These ideas are relevant to patient monitoring because edge healthcare nodes may not behave like stable data-center participants.

The implication for the present study is that experimental evaluation should not be limited to a clean federated setup. It should include non-IID partitioning, label noise, client scaling, and aggregation variation. These conditions do not fully cover all adversarial threats, but they represent a necessary baseline for showing whether the framework remains stable under realistic healthcare data conditions.

2.3. Blockchain for Trust, Auditability, and Model Provenance

Blockchain is frequently introduced into federated learning to address trust gaps left by conventional server-based aggregation. In traditional FL, the coordinator receives local updates, computes the global model, and redistributes it. Although efficient, this creates dependence on the coordinator. Participants may have limited visibility into which updates were submitted, whether the aggregation process was altered, or how model versions evolved. Blockchain can help by recording tamper-evident evidence of the training process.

A key design choice is whether blockchain should perform aggregation or only record evidence. Some blockchain-based FL systems place a larger part of the learning workflow on-chain. ScaleBCFL, implemented with Hyperledger Fabric, decentralizes the conventional aggregation server and introduces sharding to reduce communication cost [13]. FedBN similarly uses blockchain in the federated training process and improves communication efficiency by reducing the frequency of blockchain authentication and packaging operations [14]. These works show that blockchain can support FL coordination, but they also reveal the cost of deeper blockchain involvement. More on-chain operations generally mean higher latency, storage demand, and implementation complexity.

Other studies support a lighter interpretation of blockchain as an audit layer. Blockchain-based approaches for secure federated learning have shown that registering model updates on-chain can provide auditability without necessarily harming FL performance [15]. This idea is central to the present paper. For patient monitoring, the blockchain does not need to store raw patient data or full model parameters. It can instead store compact evidence such as model hashes, timestamps, participation records, and verification metadata. This design preserves the main benefit of blockchain, namely tamper-evident traceability, while avoiding unnecessary computation on-chain.

Web3 healthcare verification studies further support compact blockchain evidence. A trustworthiness verification framework for Web3.0 healthcare communities uses feature extraction, similarity matrices, encrypted verification, and blockchain records to verify multi-modal healthcare federated learning while reducing on-chain storage and computational cost [16]. This is particularly relevant because healthcare systems may generate large models and multi-modal data. If every update or model artifact is placed directly on-chain, scalability becomes unrealistic. The present framework therefore follows a lightweight anchoring logic: blockchain records proof of model evolution and participation, while learning and aggregation remain off-chain.

This distinction helps clarify the contribution of the proposed system. It does not claim that blockchain improves model accuracy directly. Instead, blockchain improves the accountability of the training process. The model is trained through federated learning, while blockchain anchoring provides a verifiable record that can support later audit, compliance review, or forensic analysis if model behavior is questioned.

2.4. Verifiable Training, Identity Control, and Advanced Privacy Mechanisms

As blockchain-based federated learning has matured, researchers have started to address more advanced verification and identity problems. A basic blockchain log can show that a transaction occurred, but it may not prove that local training or aggregation was computed correctly. Zero-knowledge proofs have therefore been proposed to verify training and aggregation without exposing the underlying model contents. A ZKP-based blockchain FL framework verifies both local and global computations off-chain while storing verification evidence on-chain [6]. This is an important development because it avoids the high cost and privacy exposure of full on-chain model storage. However, ZKP integration still introduces complexity that may be unnecessary for all healthcare scenarios.

Identity management is another trust issue in federated learning. If participants cannot be authenticated reliably, malicious or unauthorized clients may join the training process. Blockchain-based self-sovereign identity provides one solution by allowing users or organizations to manage verifiable credentials through decentralized identifiers [7]. This approach is useful for federated settings because it separates identity verification from centralized account management. For healthcare, identity control is relevant when multiple hospitals, clinics, or edge gateways participate in model training. However, the present study focuses on anchoring model and participation evidence rather than implementing a full decentralized identity system.

Long-term cryptographic resilience is also emerging as a concern. A post-quantum secure blockchain-based FL framework for healthcare analytics evaluates Falcon, Dilithium, and SPHINCS+ signatures and shows that signature verification can become a major contributor to blockchain cost [8]. This finding is important because it reminds system designers that stronger cryptographic guarantees are not free. In healthcare, data and audit records may remain sensitive for long periods, but the choice of cryptographic mechanisms must be balanced against latency, cost, and deployment feasibility.

Privacy-preserving blockchain FL has also been studied in mobile crowdsourcing, where participants are resource-constrained and may require verifiable rewards. A blockchain-based privacy-preserving FL framework for mobile crowdsourcing combines masking techniques, the Chinese Remainder Theorem, and blockchain-based verification to support privacy and trust simultaneously [17]. Although mobile crowdsourcing is not healthcare, the setting is useful because it resembles remote monitoring in terms of distributed devices, limited resources, and participant trust concerns.

Together, these studies show that verifiability, identity, and advanced privacy mechanisms are available, but they should be selected according to deployment goals. A system designed for high-risk EHR sharing may justify threshold cryptography, ZKPs, or post-quantum signatures. A patient monitoring framework that primarily needs model traceability may reasonably adopt a lighter anchoring design. The key is to state clearly what the system protects against and what it does not.

2.5. Blockchain-Based Federated Learning in Healthcare Systems

Healthcare-specific blockchain FL studies provide the closest foundation for the proposed framework. These studies generally agree that federated learning protects data locality, while blockchain improves trust, transparency, or auditability. However, they differ substantially in system complexity, threat assumptions, and application targets.

In IoMT data sharing, PPBFL directly addresses privacy leakage and poisoning resistance through threshold secret sharing, secure multi-party computation, cosine similarity verification, and smart-contract-based aggregation [3]. This is one of the strongest healthcare-related works because it treats privacy and poisoning as simultaneous problems. Nevertheless, its protection level comes with a more complex protocol stack. In contrast, the present framework does not attempt to solve secure aggregation and poisoning resistance through cryptographic verification. It focuses on a lighter auditability layer and evaluates model behavior under noisy and heterogeneous settings.

In remote patient monitoring, MedGuard-FL introduces adaptive defense mechanisms based on patient status and system risk [1]. This clinical-risk perspective is valuable because it connects FL security to the healthcare context. The present study is more limited in clinical adaptation, but it contributes by emphasizing lightweight blockchain anchoring and empirical comparison across client counts, label noise, and aggregation strategies.

For EHR sharing, AI-driven blockchain and federated learning frameworks provide stronger privacy and access-control designs, including dynamic differential privacy, thresh-

old key sharing, and blockchain audit trails [2]. These systems are broader than patient monitoring and are often designed for cross-institutional data sharing. The present study is narrower but more focused on the operational question of how a lightweight FL monitoring framework behaves under common data-quality and client-distribution challenges.

Medical imaging studies provide another comparison point. Explainable FL with blockchain-based client valuation improves interpretability and data-quality-aware aggregation in medical image analysis [4]. Blockchain-verified brain tumor classification records hashes, accuracy, and timestamps to support auditability in a federated MRI classification pipeline [5]. These systems demonstrate that blockchain records can improve confidence in multi-site clinical AI workflows. However, medical imaging models often require larger neural architectures, higher computation, and task-specific preprocessing. Patient monitoring can use lighter models and therefore benefits from an anchoring design that avoids unnecessary blockchain burden.

Healthcare review work also confirms the growing relevance of blockchain, smart contracts, and federated learning for secure AI systems. A systematic review of blockchain-based smart contract frameworks for secure AI healthcare systems identifies privacy preservation, secure information exchange, decentralized access management, and AI-blockchain integration as key directions [18]. Another healthcare-focused blockchain FL study emphasizes secure communication, decentralized learning, patient privacy, and trust management in IoT-based healthcare systems [19]. These review-oriented works support the general direction of the present study, but they also highlight unresolved issues related to scalability, interoperability, and energy efficiency.

The healthcare literature therefore leaves a clear design space. Existing works either provide strong but complex security mechanisms or focus on task-specific clinical applications. The present study targets a middle position: a practical federated learning framework for patient monitoring that keeps data local, evaluates learning stability under realistic simulation conditions, and adds blockchain anchoring for tamper-evident traceability without moving full learning computation on-chain.

2.6. Lessons from Cross-Domain Blockchain Federated Learning

Although this paper focuses on healthcare, cross-domain blockchain FL studies provide useful lessons for system design. The same basic problem appears across many domains: multiple parties need to collaborate without sharing raw data and without fully trusting one another. Construction safety services use blockchain-based FL to support collaborative governance and secure model sharing among organizations [20]. Smart-grid and peer-to-peer energy studies use blockchain and federated learning to support decentralized control, transaction trust, and privacy-preserving market coordination [21,22]. Accounting fraud detection uses blockchain consensus and federated Transformer networks to support cross-organization fraud modeling while preserving institutional data sovereignty [23].

These cross-domain systems are not directly comparable to patient monitoring, but they reinforce two important design principles. First, blockchain is most useful when it records or verifies interactions among parties that do not fully trust one another. Second, blockchain should not be added blindly. If the system places too much computation or data on-chain, scalability and latency become limiting factors. These lessons support the decision to use blockchain as an anchoring layer rather than as a full aggregation engine.

Other cross-domain work highlights specialized mechanisms that may inspire future healthcare extensions. BloBS-FL combines split learning with blockchain-based decentralized federated learning and PBFT consensus to improve robustness in adversarial settings [24]. Blockchain-based federated feature unlearning addresses the right to be forgotten by recording update operations and improving traceability of unlearning

processes [25]. FLOWER applies federated learning and blockchain-based consensus to real-time trajectory endorsement in connected autonomous vehicles [26]. These works show that blockchain FL can support resilience, unlearning, and real-time verification. However, each mechanism should be adopted only when it matches the healthcare use case and operational constraints.

For the present study, the cross-domain literature mainly supports the architectural logic rather than the clinical motivation. It confirms that lightweight evidence recording, decentralized verification, and tamper-evident logs are useful across distributed systems. The healthcare contribution of this paper is to apply that logic to patient monitoring and to evaluate the federated learning component under noisy, heterogeneous, and scalable client conditions.

2.7. Research Gap and Positioning of This Study

The literature shows that federated learning, blockchain, and privacy-preserving computation are increasingly connected, but the specific problem of lightweight auditable patient monitoring remains insufficiently addressed. Existing healthcare federated learning studies show the value of local training, but they often do not provide a simple and auditable record of model evolution. Blockchain healthcare systems provide traceability, but many rely on heavy cryptographic mechanisms, full smart-contract aggregation, or application-specific designs that may be difficult to deploy in lightweight monitoring settings. Robust federated learning studies address poisoning, dropout, or non-IID data, but these issues are not always evaluated together with blockchain-based auditability.

A second gap concerns the separation of security properties. Privacy, robustness, and auditability are sometimes discussed as if they were the same objective. They are not. Federated learning protects data locality. Differential privacy reduces information leakage from updates. Robust aggregation mitigates unreliable or malicious contributions. Blockchain anchoring provides tamper-evident evidence of training history. These mechanisms complement one another, but no single mechanism solves the entire healthcare trust problem. A clear system design must therefore state which layer addresses which risk.

A third gap concerns deployment practicality. Several advanced systems use homomorphic encryption, secure multi-party computation, threshold cryptography, zero-knowledge proofs, post-quantum signatures, or full on-chain aggregation. These mechanisms are valuable for high-assurance settings, but they may introduce overhead that is not always justified for semi-trusted patient monitoring networks. A lighter design can be more appropriate when the main objective is to preserve local data, maintain stable model training, and provide auditable evidence of participation and model versioning.

This study is positioned within this practical middle ground. The proposed framework uses federated learning to train patient monitoring models without transferring raw patient data. It evaluates model behavior under non-IID data, label noise, client scaling, and different aggregation strategies. It then adds lightweight blockchain anchoring to record model hashes and participation proofs. The blockchain layer is not used to store raw data, execute full aggregation, or claim complete protection against inference attacks. Instead, it provides traceability and accountability for the federated learning process.

This positioning directly addresses the gap between technically strong but complex blockchain FL systems and lightweight healthcare monitoring needs. The intended contribution is not another heavyweight cryptographic protocol. Rather, it is a practical framework that connects privacy-preserving learning, robustness evaluation, and audit-oriented blockchain anchoring in a form suitable for semi-trusted healthcare environments.

To further clarify the positioning of this study, Table 1 provides a qualitative comparison with representative blockchain-based federated learning studies. The table is not

intended as a head-to-head performance benchmark because the compared studies use different datasets, threat models, learning tasks, and blockchain deployment assumptions. Instead, it summarizes how the proposed framework differs in terms of application focus, blockchain role, security complexity, and evaluation scope.

Table 1. Qualitative positioning comparison with representative blockchain-based federated learning studies.

Study	Application Focus	Blockchain and Security Role	Positioning Relative to This Study
MedGuard-FL [1]	Remote patient monitoring	Uses a risk-aware adaptive security framework with privacy and robustness mechanisms adjusted according to patient and system risk	Closest in clinical monitoring motivation, but it emphasizes adaptive security defenses, while this study emphasizes lightweight blockchain anchoring and controlled convergence behavior under synthetic non-IID and noisy settings
PPBFL for IoMT data sharing [3]	Internet of Medical Things data sharing	Combines blockchain with threshold secret sharing, secure multi-party computation, cosine similarity verification, and smart-contract-based aggregation	Provides stronger privacy and poisoning-resistance mechanisms, while this study targets a lighter semi-trusted monitoring setting with blockchain used mainly for model-hash and participation-record anchoring
AI-driven blockchain and FL for EHR sharing [2]	Secure electronic health record sharing	Combines federated learning with blockchain audit trails and additional privacy or key-management mechanisms	Addresses broader EHR sharing and access-control concerns, while this study focuses on patient monitoring simulation, aggregation behavior, and lightweight auditability
Explainable federated medical image analysis with blockchain [4]	Medical image analysis	Uses blockchain in a federated medical image analysis setting, including support for trust or valuation in the learning process	Focuses on medical imaging and explainability, while this study uses lightweight tabular monitoring simulation and evaluates label noise, client scaling, and aggregation strategies
Blockchain-verified FL for brain tumour classification [5]	Brain tumour classification using MRI scans	Uses blockchain verification in a federated medical imaging pipeline, including evidence such as model-related records and smart-contract support	Demonstrates blockchain-supported auditability in a task-specific imaging pipeline, while this study targets lightweight patient monitoring and does not place full learning computation on-chain
ZKP-based blockchain FL [6]	Verifiable federated training and aggregation	Uses zero-knowledge proofs to verify training and aggregation while storing evidence on-chain	Provides stronger verifiability guarantees, while this study deliberately avoids ZKP complexity and uses hash-based anchoring as a lightweight auditability mechanism

Table 1. Cont.

Study	Application Focus	Blockchain and Security Role	Positioning Relative to This Study
FedBN [14]	Communication-efficient blockchain-based federated learning	Uses blockchain within the federated learning process to improve communication or coordination efficiency	Represents deeper blockchain involvement in FL coordination, while this study keeps learning and aggregation off-chain and uses blockchain only for compact evidence anchoring
Blockchain-based approaches for secure FL [15]	Secure federated learning architecture	Discusses blockchain-based mechanisms for improving security and auditability in federated learning	Supports the general feasibility of blockchain-assisted FL, while this study narrows the design to patient monitoring, controlled non-IID simulation, and lightweight model-provenance anchoring
Proposed framework	Privacy-preserving patient monitoring simulation	Uses blockchain as a lightweight anchoring layer for model hashes, round metadata, timestamps, and client participation records	Connects data locality, aggregation stability, and auditability in one controlled workflow without claiming full privacy protection, clinical validation, or blockchain-driven accuracy improvement

This comparison highlights the intended middle-ground position of the proposed framework. Prior systems often provide stronger security guarantees through mechanisms such as secure multi-party computation, zero-knowledge proofs, adaptive privacy, or deeper blockchain-based coordination. These mechanisms are valuable for high-assurance deployments, but they also increase implementation and operational complexity. In contrast, the proposed framework focuses on a lighter semi-trusted patient monitoring scenario in which federated learning preserves data locality, aggregation strategies support learning stability, and blockchain anchoring provides tamper-evident evidence for later audit.

3. System Architecture

The proposed framework is designed as a lightweight federated learning architecture with blockchain-based anchoring for patient monitoring. Its main objective is not to move the entire learning process onto a blockchain, but to combine efficient off-chain federated model training with tamper-evident on-chain evidence. This design follows a practical assumption for semi-trusted healthcare environments: participating institutions or monitoring nodes are registered and allowed to join the learning process, but their participation history and model update records should still be verifiable after training.

The architecture contains four logical components: the client layer, the coordinator layer, the blockchain anchoring layer, and the privacy and governance control plane. The client layer preserves data locality by keeping patient monitoring data at the source. The coordinator layer manages model distribution and aggregation. The blockchain anchoring layer records compact evidence, such as model hashes and participation proofs, without storing patient records or full model parameters on-chain. The privacy and governance control plane defines the operational boundary of the system, including authorization, consent awareness, and audit access. Figure 1 illustrates the proposed system architecture.

Federated Learning System Architecture with Lightweight Blockchain Anchoring

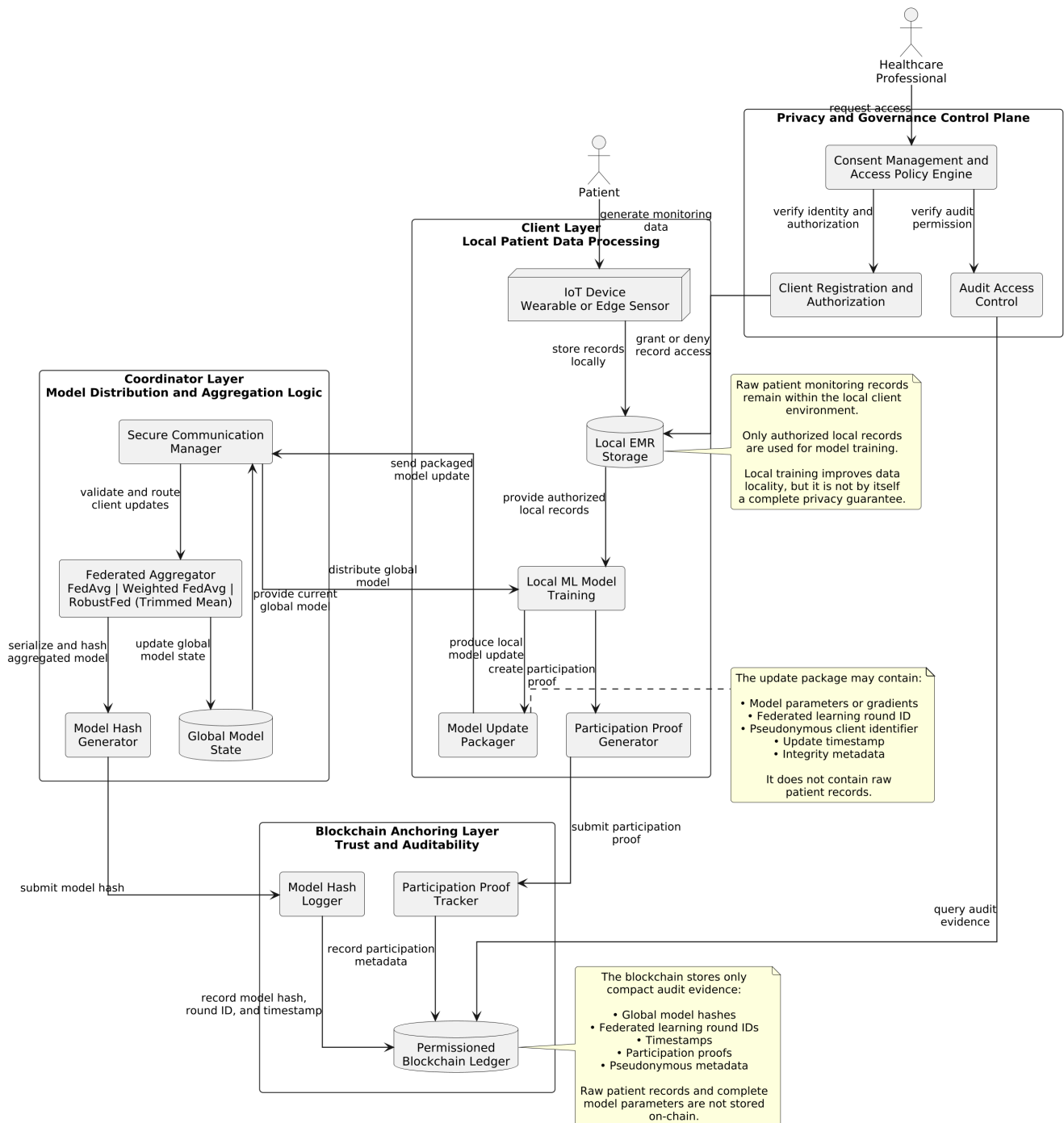


Figure 1. Federated learning system architecture with lightweight blockchain anchoring.

3.1. Design Rationale and Layered Overview

The layered design is adopted to separate learning, coordination, auditability, and governance responsibilities. This separation is important because privacy, robustness, and auditability are different system properties. Federated learning supports data locality, but it does not automatically provide a verifiable history of model updates. Robust aggregation can improve training stability, but it does not by itself provide an immutable audit trail. Blockchain can support tamper-evident traceability, but it should not be used as a replacement for all learning and aggregation operations when lightweight deployment is required.

For this reason, the framework keeps computationally intensive learning operations off-chain. Local model training is performed by clients, and global aggregation is handled by the coordinator. The blockchain layer is used only to anchor selected metadata. This avoids unnecessary blockchain storage growth and reduces the cost associated with full on-chain aggregation. Similar design concerns have been discussed in blockchain-based federated learning systems, where storing complete model information or executing aggregation directly on-chain can increase latency and computational overhead [6,15,16]. The proposed architecture therefore uses blockchain only where it adds clear value: model provenance, client participation evidence, and post hoc auditability.

3.2. Client Layer: Local Patient Data Processing

The client layer represents healthcare sites, edge gateways, or patient monitoring nodes that hold local data. Each client keeps its patient monitoring records within its own environment and does not transmit raw data to the coordinator or the blockchain. In the experimental implementation, each client trains a local multilayer perceptron model using its local synthetic patient monitoring dataset. The architecture can be extended to other model types, but the reported experiments focus on the MLP configuration described in Section 4.

Each client performs four main functions. The first is local data preparation, where patient monitoring features are normalized and arranged for model training. The second is local model training, where the client updates its model parameters using only local data. The third is update packaging, where the trained model parameters or weight updates are prepared for submission to the coordinator. The fourth is evidence generation, where the client creates a participation record and contributes to the hash-based evidence used by the blockchain anchoring layer.

This layer is included to preserve institutional and patient data sovereignty. In health-care settings, local control is important because raw patient data may be subject to legal, ethical, and institutional restrictions. By keeping data local, the framework reduces the exposure created by centralized data collection. However, this does not mean that all privacy risks are eliminated. Model updates may still leak information under strong inference attacks. Therefore, the client layer should be understood as a data-locality mechanism, not as a complete privacy guarantee.

3.3. Coordinator Layer: Model Distribution and Aggregation

The coordinator layer manages the federated learning workflow. At the beginning of each communication round, the coordinator distributes the current global model to selected clients. After local training, the coordinator receives model updates from participating clients and applies an aggregation strategy to produce the next global model. The updated global model is then redistributed for the following round.

The coordinator is retained for efficiency. Fully decentralized aggregation can improve trust decentralization, but it may also introduce additional latency, communication cost, and implementation complexity. For lightweight patient monitoring, a coordinator-based design is more practical, provided that its actions can be audited. The proposed framework therefore does not remove the coordinator. Instead, it reduces blind trust in the coordinator by anchoring selected evidence from each training round.

Three aggregation strategies are considered in the framework: FedAvg, weighted aggregation, and robust aggregation. FedAvg provides the standard baseline by averaging participating client updates. Weighted aggregation adjusts the contribution of clients according to the amount of local data or other predefined weighting criteria. Robust aggregation reduces the influence of suspicious or unstable updates and is useful when

client data are noisy or heterogeneous. The mathematical formulation of these strategies is provided in Section 4.

The coordinator also prepares the global model evidence used by the blockchain anchoring layer. After each aggregation round, the resulting global model is serialized and hashed. The hash is not intended to reveal the model itself. Instead, it acts as a compact fingerprint of the model version. If the model is later modified, the hash will change, allowing inconsistency to be detected during audit.

3.4. Blockchain Anchoring Layer: Trust and Auditability

The blockchain anchoring layer provides tamper-evident recordkeeping for the federated learning process. It records compact metadata rather than raw patient data or complete model parameters. The main records include the global model hash, communication round identifier, participating client identifiers or pseudonyms, timestamp, and participation proof. These records allow auditors or system administrators to verify which clients participated in a given round and whether a later model version matches the anchored hash.

This layer is intentionally lightweight. It does not perform full model aggregation, does not store local datasets, and does not store complete model parameters. This design reduces storage demand and avoids exposing model contents through transparent ledgers. The blockchain is used as an audit and provenance mechanism rather than as the primary computation engine. This choice is aligned with the objective of supporting patient monitoring systems, where latency, simplicity, and maintainability are important.

The anchoring layer also improves accountability. If a model version is disputed, the stored hash can be compared with the model artifact held by the coordinator or institution. If a client denies participation, the participation proof can be checked against the corresponding round record. If model history is audited, the sequence of anchored hashes provides a tamper-evident trace of model evolution. These properties do not prevent every attack, but they strengthen post hoc verification and reduce the risk of unverifiable model handling.

3.5. Privacy and Governance Control Plane

The privacy and governance control plane defines how system participation, access, and audit activities are controlled. In a practical healthcare deployment, this plane would include client registration, role-based access, institutional authorization, consent-aware participation policies, and audit permissions. Its purpose is to ensure that only authorized healthcare institutions or monitoring nodes participate in federated training and that audit records can be accessed by appropriate parties.

This component is separated from the learning pipeline because governance rules may differ across institutions and jurisdictions. For example, one hospital consortium may require strict institutional approval before each training task, while another may allow periodic automated participation under a standing agreement. Separating governance from model training allows the federated learning workflow to remain technically consistent while supporting different operational policies.

The proposed framework does not implement a full self-sovereign identity or consent-management infrastructure. Instead, it defines a governance interface that can be extended with such mechanisms in future deployment. Blockchain-based self-sovereign identity and verifiable credential systems can strengthen participant authentication, while smart-contract-based consent systems can support more granular access policies [7]. These are treated as compatible extensions rather than required components of the present lightweight architecture.

3.6. Trust Boundaries and Design Assumptions

The proposed architecture assumes a semi-trusted healthcare network. Participating clients are registered entities, such as hospitals, clinics, or approved monitoring gateways, but they may still experience unreliable participation, noisy local data, or operational failures. The coordinator is assumed to execute the aggregation protocol, but its output should be auditable through anchored model hashes. The blockchain network is assumed to provide tamper-evident storage for submitted metadata.

The framework is designed to resist or mitigate specific risks. It reduces raw data exposure by keeping patient records local. It improves learning stability by allowing different aggregation strategies to be evaluated under non-IID and noisy settings. It improves auditability by anchoring model hashes and participation proofs. However, the framework does not claim to prevent all privacy or security attacks. In particular, blockchain anchoring alone does not prevent gradient inversion, membership inference, or model poisoning. Defenses against such attacks require additional mechanisms such as differential privacy, secure aggregation, secure multi-party computation, robust anomaly detection, or zero-knowledge verification.

This boundary is important. The proposed blockchain layer is an auditability mechanism, not a universal security solution. Its role is to make the federated learning process more traceable and accountable while preserving the lightweight nature of patient monitoring. The architecture therefore provides a practical foundation that can be strengthened with heavier cryptographic protections when the deployment context requires stronger guarantees.

4. Methodology

This section describes the methodology used to evaluate the proposed federated learning framework. The experiments were designed as controlled simulations rather than clinical deployment trials. The objective was to examine the behavior of federated learning under synthetic patient monitoring conditions with heterogeneous client distributions, controlled label noise, different federation sizes, partial client participation, and multiple aggregation strategies. The simulation was also used to demonstrate how lightweight blockchain anchoring can be integrated as an auditability layer without moving raw patient data or complete model parameters on-chain.

The methodology consists of synthetic patient data generation, client-level data partitioning, label noise injection, centralized and local baseline training, federated model training, aggregation strategy comparison, client scaling analysis, blockchain anchoring specification, and performance evaluation. Because the dataset is synthetic, all results should be interpreted as framework-level evidence under controlled simulation conditions rather than evidence of clinical effectiveness.

4.1. Synthetic Patient Dataset Design

A synthetic patient monitoring dataset was generated to support controlled and ethically safe experimentation. This design avoids the use of real patient records while allowing the data distribution, label generation process, label noise, and number of clients to be controlled. Each sample represents a simplified patient monitoring record generated from physiological indicators and auxiliary noise features.

Each record contains seven input features. The physiological features are heart rate, systolic blood pressure, diastolic blood pressure, oxygen saturation, and body temperature. Two unrelated noise features were also included to make the classification problem less separable and to simulate irrelevant variation that may appear in practical monitoring

data. The target variable is a binary status label, where 0 denotes a normal condition and 1 denotes an alert condition.

Let the complete synthetic dataset be denoted by

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N, \quad (1)$$

where N is the total number of samples, $\mathbf{x}_i \in \mathbb{R}^7$ is the feature vector of sample i , and $y_i \in \{0, 1\}$ is the binary monitoring status. The feature vector is defined as

$$\mathbf{x}_i = [HR_i, SBP_i, DBP_i, SpO2_i, Temp_i, Noise1_i, Noise2_i], \quad (2)$$

where HR_i denotes heart rate, SBP_i denotes systolic blood pressure, DBP_i denotes diastolic blood pressure, $SpO2_i$ denotes oxygen saturation, and $Temp_i$ denotes body temperature.

For the default 10-client dataset and the label noise experiments, each client contained 200 samples. For client k , where $k \in \{0, \dots, 9\}$, the synthetic features were generated according to the distributions in Table 2.

Table 2. Synthetic data generation parameters for the default 10-client setting.

Feature	Generation Rule for Client k
Heart rate	Normal distribution with mean $75 + k$ and standard deviation $8 + (k \bmod 3)$
Systolic blood pressure	Normal distribution with mean $115 + 2k$ and standard deviation 10
Diastolic blood pressure	Normal distribution with mean $70 + k$ and standard deviation 8
Oxygen saturation	Normal distribution with mean 98 and standard deviation 1
Body temperature	Normal distribution with mean $36.5 + 0.3(k \bmod 3)$ and standard deviation 0.4
Noise feature 1	Normal distribution with mean 0 and standard deviation 1
Noise feature 2	Uniform distribution over the interval $[0, 100]$
Samples per client	200 samples
Random seed	42

The alert label was assigned using a threshold-based physiological rule. A sample was labeled as an alert case only when body temperature was high and oxygen saturation was low. The rule is defined as

$$y_i = \begin{cases} 1, & \text{if } Temp_i > 37.5 \text{ and } SpO2_i < 96.5, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

This rule was selected to preserve a meaningful relationship between the physiological features and the alert label while keeping the experiment controlled and reproducible. If a generated client dataset contained only one class, a small number of labels was flipped to ensure that both normal and alert classes were present in the local dataset.

4.2. Client Partitioning and Non-IID Simulation

The federated learning setting contains K clients. Each client represents a healthcare institution, edge gateway, or patient monitoring node that stores its own local dataset. The global training data are distributed across clients as

$$\mathcal{D}_{train} = \bigcup_{k=1}^K \mathcal{D}_k, \quad \mathcal{D}_k \cap \mathcal{D}_j = \emptyset \quad \text{for } k \neq j, \quad (4)$$

where $\mathcal{D}_k$ denotes the local dataset of client k , and $n_k = |\mathcal{D}_k|$ denotes the number of local samples.

The data were non-independent and non-identically distributed because each client was generated using different physiological parameter settings. In the default 10-client dataset, additional client-level skew was introduced through class resampling when sufficient class samples were available. Clients 0, 1, and 2 were biased toward normal samples, while clients 3, 4, and 5 were biased toward alert samples. This design was used to simulate monitoring environments in which institutions or devices observe different alert prevalence.

For each client k , the local alert ratio is defined as

$$\rho_k = \frac{|\{(\mathbf{x}_i, y_i) \in \mathcal{D}_k : y_i = 1\}|}{|\mathcal{D}_k|}. \quad (5)$$

A non-IID federation is obtained when ρ_k varies across clients. This variation creates different local learning conditions and allows the experiment to observe whether the global model remains stable under heterogeneous client distributions.

To examine client scaling, separate datasets were generated for $K = 5$, $K = 10$, and $K = 20$ clients. Each client still contained 200 samples. For the scaling experiment, cyclic client-specific parameters were used to maintain heterogeneous but comparable client distributions across different federation sizes. The generation rules for the scaling datasets are summarized in Table 3.

Table 3. Synthetic data generation parameters for client scaling experiments.

Component	Setting
Client counts	$K \in \{5, 10, 20\}$
Samples per client	200 samples
Heart rate distribution	Normal distribution with mean $70 + 3(k \bmod 5)$ and standard deviation $8 + (k \bmod 3)$
Systolic blood pressure distribution	Normal distribution with mean $115 + 2(k \bmod 4)$ and standard deviation 10
Diastolic blood pressure distribution	Normal distribution with mean $70 + (k \bmod 4)$ and standard deviation 8
Oxygen saturation distribution	Normal distribution with mean 98 and standard deviation 1
Body temperature distribution	Normal distribution with mean $36.5 + 0.3(k \bmod 3)$ and standard deviation 0.4
Noise feature 1	Normal distribution with mean 0 and standard deviation 1
Noise feature 2	Uniform distribution over the interval $[0, 100]$
Alert rule	$Temp > 37.5$ and $SpO2 < 96.5$
Class correction	A small number of labels was flipped if a client contained only one class
Random seed	42

4.3. Label Noise Injection

Label noise was introduced to simulate uncertainty in patient monitoring labels. In practical healthcare environments, monitoring labels may be affected by delayed diagnosis,

inconsistent annotation, device error, or ambiguity in the definition of alert status. To model this issue, labels were randomly flipped before training.

Let η denote the label noise rate. The noisy label $\tilde{y}_i$ is defined as

$$\tilde{y}_i = \begin{cases} 1 - y_i, & \text{with probability } \eta, \\ y_i, & \text{with probability } 1 - \eta. \end{cases} \quad (6)$$

Two noise configurations were used. First, in the default non-IID dataset, 20% label flipping was applied to selected clients, namely clients 1, 3, 5, 7, and 9. This setting created a difficult federated learning scenario in which only part of the federation contained corrupted labels. Second, for the label noise sensitivity experiment, three separate datasets were generated with $\eta = 0$, $\eta = 0.10$, and $\eta = 0.30$. These settings are referred to as clean, mild-noise, and heavy-noise conditions, respectively.

4.4. Baseline Models

Two baseline settings were used before the federated learning experiments. The first baseline was centralized learning, where all client datasets were combined into a single dataset. The combined dataset was shuffled using a fixed random seed and split into 80% training data and 20% test data. The centralized model used a scikit-learn `MLPClassifier` with two hidden layers, `hidden_layer_sizes=(32,16)`, and `max_iter=200`. The centralized baseline was used to provide a conventional non-federated reference point.

The second baseline was local-only learning. In this setting, each client trained and evaluated its own model using only its local dataset. Each local dataset was shuffled and split into 80% training data and 20% test data. The local-only model used a scikit-learn `MLPClassifier` with one hidden layer, `hidden_layer_sizes=(16,)`, and `max_iter=200`. Local-only learning was used to show how models behave when no cross-client collaboration is performed.

4.5. Federated Learning Protocol

The federated learning process follows a round-based protocol. Let t denote the communication round and T denote the total number of communication rounds. At the beginning of round t , the coordinator sends the current global model parameters $\mathbf{w}^t$ to selected clients. Each selected client trains a local model using only its own dataset and returns the updated model parameters to the coordinator. Raw patient records remain local and are not transmitted.

The local training objective of client k is defined as

$$F_k(\mathbf{w}) = \frac{1}{n_k} \sum_{(\mathbf{x}_i, \tilde{y}_i) \in \mathcal{D}_k} \ell(f(\mathbf{x}_i; \mathbf{w}), \tilde{y}_i), \quad (7)$$

where $F_k(\mathbf{w})$ is the local empirical loss, $f(\mathbf{x}_i; \mathbf{w})$ is the prediction produced by the model with parameters $\mathbf{w}$, and $\ell(\cdot)$ is the binary classification loss.

The corresponding global federated objective is

$$F(\mathbf{w}) = \sum_{k=1}^K \frac{n_k}{N_{train}} F_k(\mathbf{w}), \quad (8)$$

where $N_{train} = \sum_{k=1}^K n_k$ is the total number of training samples across all clients.

The federated model was implemented using scikit-learn `MLPClassifier`. The model used one hidden layer with 16 neurons, `hidden_layer_sizes=(16,)`, `max_iter=1`, `warm_start=True`, and `random_state=42`. Each client standardized its local features using

StandardScaler. The model parameters consisted of coefficient matrices and intercept vectors, corresponding to `coefs_` and `intercepts_` in the scikit-learn implementation.

The global model was initialized once before the federated training loop. At each communication round, a subset of clients was selected through random partial participation. If d denotes the number of dropped clients in a round, then the participating client set is

$$S_t \subseteq \{1, \dots, K\}, \quad |S_t| = K - d. \quad (9)$$

The dropout value was treated as an experimental parameter and was adjusted according to the federation size to ensure that at least one client remained active for aggregation. The default 10-client federated simulation used 25 communication rounds and random partial participation.

For the federated convergence experiments, Client 3 was used as a fixed evaluation client. This choice was made to observe model behavior under a consistent skewed client condition across communication rounds. Therefore, the federated convergence results should be interpreted as a fixed-client non-IID evaluation rather than as a global clinical test set evaluation.

The model and training settings used in the centralized, local-only, and federated experiments are summarized in Table 4.

Table 4. Model and training settings used in the experiments.

Component	Setting
Centralized baseline model	MLPClassifier, hidden_layer_sizes=(32,16), max_iter=200
Local-only baseline model	MLPClassifier, hidden_layer_sizes=(16,), max_iter=200
Federated local model	MLPClassifier, hidden_layer_sizes=(16,), max_iter=1, warm_start=True
Feature scaling	StandardScaler fitted locally for each training setting
Random state	42 for data generation and scikit-learn models
Federated communication rounds	25 rounds
Federated evaluation setting	Fixed Client 3 evaluation for convergence tracking
Primary metric	Accuracy

4.6. Aggregation Strategies

Three aggregation strategies were evaluated in the federated learning simulation, namely FedAvg, weighted aggregation, and robust aggregation. Let $\mathbf{w}_k^{t+1}$ denote the model parameters returned by client k after local training in round t . In the MLP implementation, $\mathbf{w}_k^{t+1}$ represents both coefficient matrices and intercept vectors.

FedAvg was used as the standard aggregation baseline. It computes the next global model by averaging the parameters submitted by participating clients:

$$\mathbf{w}^{t+1} = \frac{1}{|S_t|} \sum_{k \in S_t} \mathbf{w}_k^{t+1}. \quad (10)$$

This strategy gives equal influence to all participating clients.

Weighted aggregation accounts for differences in local dataset size. The contribution of each participating client is weighted according to the number of local samples:

$$\mathbf{w}^{t+1} = \sum_{k \in S_t} \frac{n_k}{\sum_{j \in S_t} n_j} \mathbf{w}_k^{t+1}. \quad (11)$$

This strategy gives greater influence to clients with larger local datasets. In the generated datasets used in this study, the number of samples per client was generally fixed at 200. Therefore, weighted aggregation is mathematically close to FedAvg in equal-size settings, but it is included because it supports extension to unequal healthcare client sizes.

Robust aggregation was implemented as an accuracy-trimmed aggregation strategy. Each participating client model was evaluated on its own local dataset after local training. Let a_k denote the local accuracy of client k . The participating client models were sorted according to a_k , and the lowest and highest 10% were removed before averaging the remaining model parameters. Let R_t denote the retained client set after trimming. The robust aggregation rule is defined as

$$\mathbf{w}^{t+1} = \frac{1}{|R_t|} \sum_{k \in R_t} \mathbf{w}_k^{t+1}. \quad (12)$$

The retained set is defined as

$$R_t = \text{Trim}_{10\%}(\{(k, a_k) : k \in S_t\}), \quad (13)$$

where $\text{Trim}_{10\%}(\cdot)$ removes the lowest- and highest-ranked client models based on local accuracy. This approach is intended to reduce the influence of unusually poor or unusually dominant local models. It should be interpreted as a practical trimming heuristic rather than a complete defense against adaptive poisoning attacks.

4.7. Blockchain Anchoring Procedure

The blockchain component was specified as a lightweight anchoring layer for auditability. The blockchain layer was not used to store raw patient data, complete model parameters, or full training datasets. Instead, it was designed to store compact evidence of model evolution and client participation.

After each aggregation round, the resulting global model parameters can be serialized and converted into a cryptographic hash:

$$h^{t+1} = H(\text{Serialize}(\mathbf{w}^{t+1})), \quad (14)$$

where $H(\cdot)$ denotes a cryptographic hash function and h^{t+1} is the hash of the global model produced after round t .

The anchoring record for each round is represented as

$$r^{t+1} = (t + 1, h^{t+1}, S_t, \tau^{t+1}), \quad (15)$$

where $t + 1$ is the round identifier, h^{t+1} is the global model hash, S_t is the set of participating clients, and τ^{t+1} is the timestamp of the anchoring event. This record provides a compact representation of model provenance and client participation.

This anchoring procedure is included as an auditability mechanism rather than as a performance-improving learning component. It allows a model artifact to be checked later by recomputing its hash and comparing it with the anchored value. It can also support post hoc review of which clients participated in each communication round. The current simulation does not report blockchain transaction latency, throughput, gas cost, or deployment-level storage overhead. Those measurements require a separate blockchain deployment benchmark and are treated as future work.

4.8. Evaluation Metrics and Experimental Design

The primary evaluation metric was classification accuracy. Accuracy was used because the current federated implementation records convergence behavior by measuring accuracy after each communication round. Accuracy is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (16)$$

where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives, respectively.

The centralized baseline also produced a confusion matrix to examine class-level behavior. The local-only baseline generated per-client classification reports, including precision, recall, and F1-score. However, the federated convergence experiments reported in this simulation primarily use accuracy. Therefore, the federated results should be interpreted as convergence and stability evidence rather than as a complete class-balanced clinical evaluation.

The experimental design is summarized in Table 5.

Table 5. Summary of experimental settings.

Experimental Component	Setting
Data type	Synthetic patient monitoring data
Input features	Heart rate, systolic blood pressure, diastolic blood pressure, oxygen saturation, body temperature, and two noise features
Target variable	Binary Status label, where 0 denotes normal and 1 denotes alert
Samples per client	200 samples
Default client count	10 clients
Client scaling settings	5, 10, and 20 clients
Label noise settings	Clean data with 0%, mild noise with 10%, and heavy noise with 30% random label flipping
Additional noisy non-IID setting	20% label flipping for selected clients in the default 10-client dataset
Communication rounds	25 rounds
Client participation	Random partial participation with dropout adjusted according to federation size
Aggregation strategies	FedAvg, weighted aggregation, and accuracy-trimmed robust aggregation
Federated evaluation set	Fixed Client 3 evaluation for convergence tracking
Primary reported metric	Accuracy
Blockchain role	Model-hash and participation-record anchoring design

The experiments were structured into four groups. The baseline experiment compares centralized learning, local-only learning, and federated learning. The label noise experiment compares clean, mild-noise, and heavy-noise synthetic datasets. The client scaling experiment compares 5-client, 10-client, and 20-client federations. The aggregation experiment compares FedAvg, weighted aggregation, and accuracy-trimmed robust aggregation under the same federated protocol.

All reported simulations use controlled synthetic data and fixed random seeds. The current revision does not report mean and standard deviation across multiple independent

seeds. Consequently, the results should not be interpreted as statistical performance estimates across repeated random trials. Instead, they provide a controlled proof-of-concept evaluation of the proposed framework under reproducible simulated conditions. Future work should extend the evaluation using multiple random seeds, public patient monitoring datasets, additional class-balanced metrics, and deployment-level blockchain overhead measurements.

5. Experimental Results

This section presents the experimental results of the proposed federated learning framework under controlled synthetic simulation settings. The evaluation focuses on four aspects: baseline learning behavior, sensitivity to label noise, sensitivity to the number of clients, and the effect of the aggregation strategy. The reported results should be interpreted as framework-level evidence obtained from synthetic patient monitoring data rather than clinical validation on real patient records.

The experiments were conducted using fixed random seeds and controlled synthetic datasets. Therefore, the reported values represent observed results from reproducible simulation runs rather than mean performance across repeated random trials. No unsupported mean or standard deviation values are reported in this section.

5.1. Baseline Comparison

The first experiment compares three learning settings: centralized training, local-only training, and federated learning. The centralized model provides a conventional reference point when all client data are assumed to be available in one location. The local-only setting evaluates how individual clients perform when each client trains independently using only its own local data. The federated setting evaluates whether collaborative learning can produce stable global model behavior while keeping raw client data local.

Figure 2 shows the confusion matrix of the centralized MLP model. The centralized model correctly classifies most normal samples, but it fails to detect alert cases. Specifically, the model correctly classifies 356 normal samples and misclassifies 42 alert samples as normal. It also produces 2 false alert predictions and no true alert predictions. This produces an overall accuracy of approximately 0.89, but the alert-class recall is 0.00. This finding shows that overall accuracy alone can be misleading in imbalanced patient monitoring scenarios. A model may appear accurate while failing to detect clinically important alert cases.

Figure 3 presents the accuracy of local-only MLP models across individual clients. The results show noticeable variation among clients. Some clients achieve high local accuracy, while others achieve lower performance. This variation reflects the effect of non-IID client distributions, client-specific feature distributions, and client-level class imbalance. Although local-only training can perform well for some clients, it does not produce a shared global model and does not support collaborative learning across the federation.

Figure 4 shows the federated learning accuracy under a difficult simulation setting that includes noisy client data, non-IID client distributions, random client dropout, and fixed skewed-client evaluation. In this experiment, Client 3 was used as the fixed evaluation client to provide a consistent non-IID evaluation condition across communication rounds. The model starts at approximately 0.74 accuracy and gradually improves before stabilizing at approximately 0.80 after 25 communication rounds. Although this performance is lower than the clean-data setting, it indicates that the federated model can maintain stable convergence under a more difficult distributed setting.

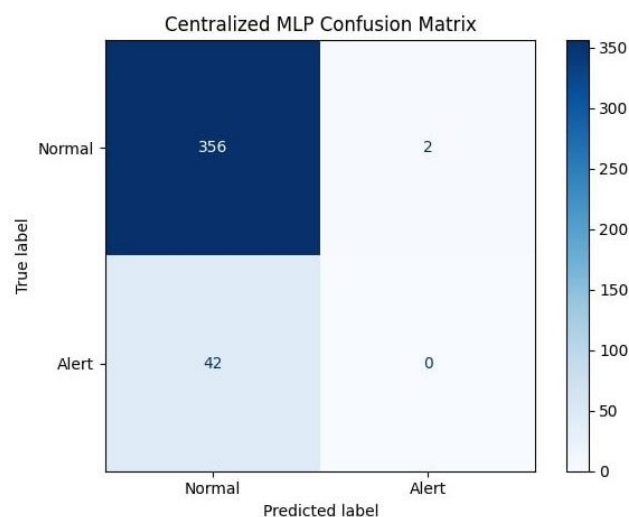


Figure 2. Confusion matrix of the centralized MLP model.

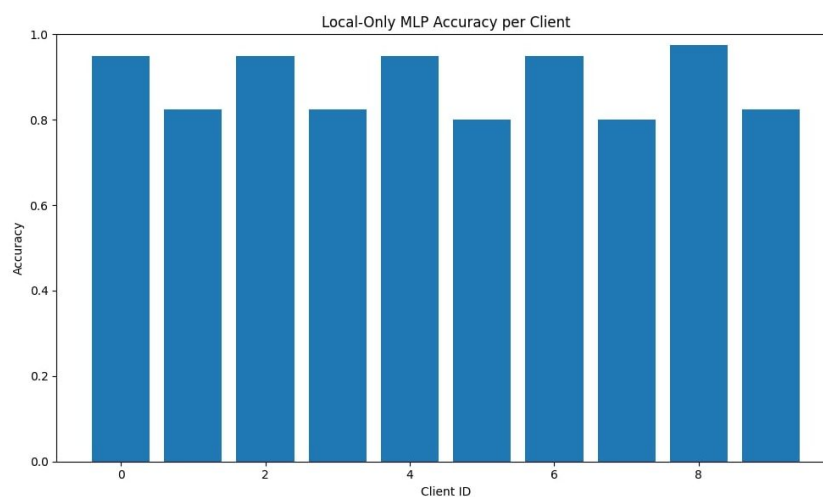


Figure 3. Accuracy of local-only MLP models across individual clients.

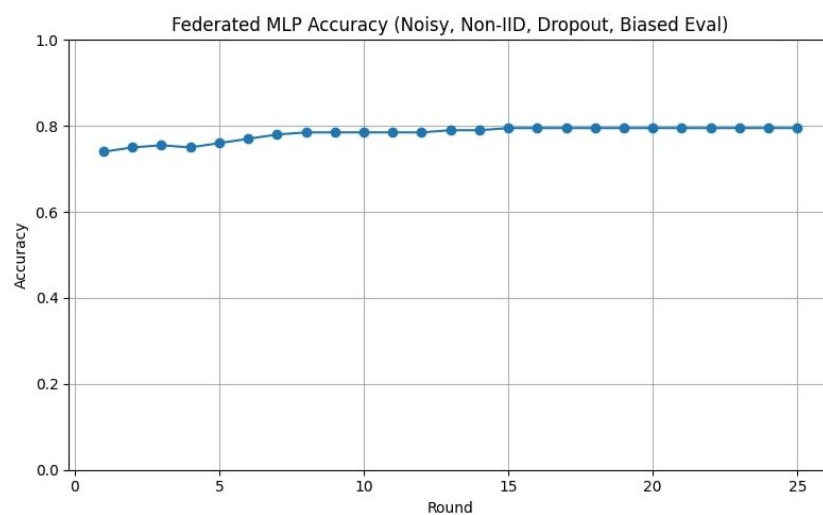


Figure 4. Federated MLP accuracy under noisy, non-IID, dropout, and fixed skewed-client evaluation settings.

5.2. Effect of Label Noise

The second experiment evaluates the effect of label noise on federated learning convergence. Three data conditions were considered: clean data, mild label noise, and heavy

label noise. The clean setting uses no random label flipping. The mild-noise setting flips 10% of labels. The heavy-noise setting flips 30% of labels.

Figure 5 shows that the clean setting converges rapidly and reaches approximately 0.99 accuracy. Under mild label noise, the model still converges consistently but stabilizes at a lower level of approximately 0.90. Under heavy label noise, the model remains substantially lower and stabilizes around 0.70. These results indicate that label quality has a strong effect on federated convergence. The degradation is especially visible when moving from mild to heavy label corruption, suggesting that repeated aggregation of corrupted local updates can limit global model performance.

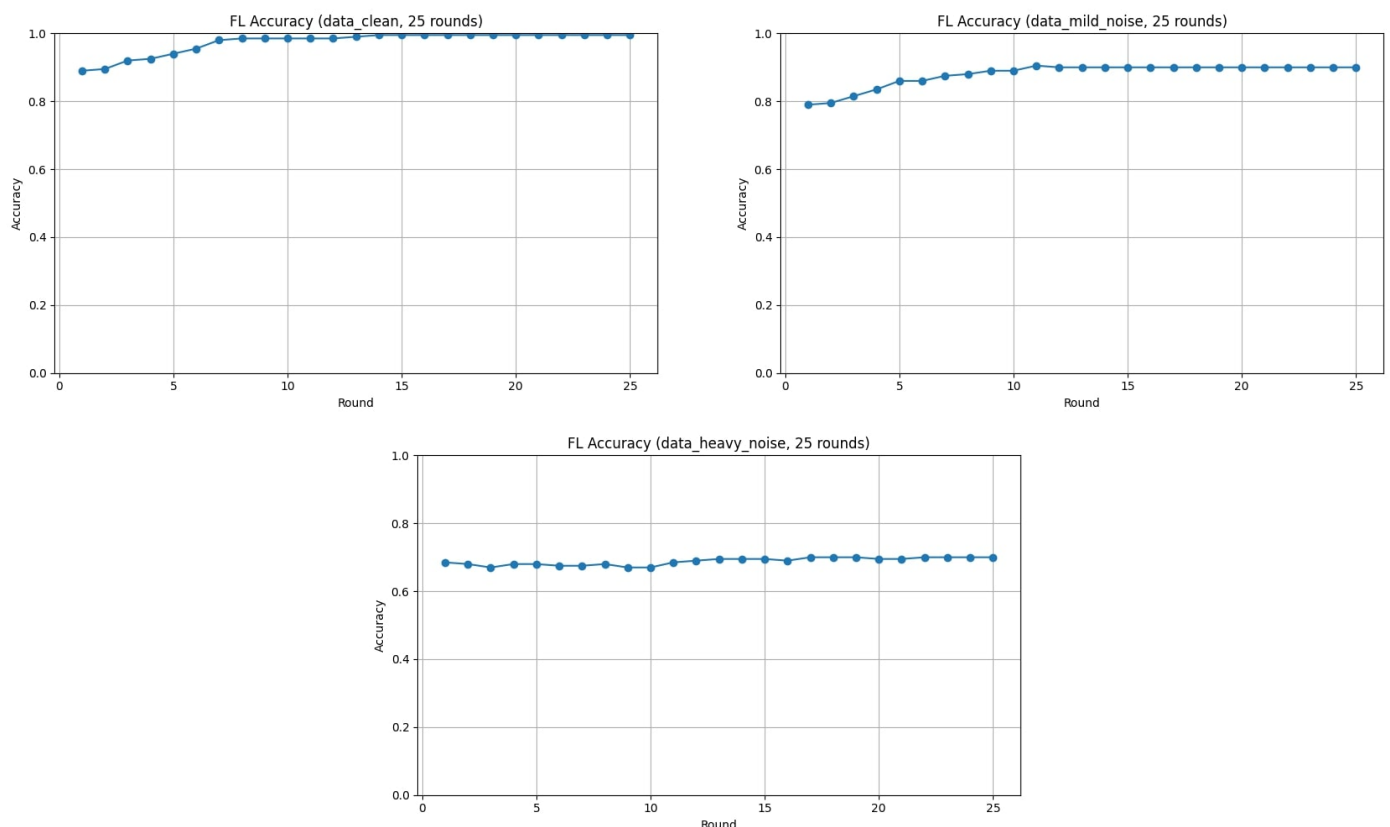


Figure 5. Federated learning accuracy under clean, mild-noise, and heavy-noise data conditions.

5.3. Client Scaling Sensitivity

The third experiment examines whether the federated learning framework remains stable as the number of clients increases. The client population was varied across 5, 10, and 20 clients while maintaining 25 communication rounds. Each client contained 200 synthetic monitoring samples.

As shown in Figure 6, all client-scale settings show stable convergence. With 5 clients, the model starts at approximately 0.87 and reaches approximately 0.99 by the final round. With 10 clients, the model starts at approximately 0.82 and stabilizes around 0.98. With 20 clients, the model starts at approximately 0.90 and reaches approximately 0.98 within the early communication rounds. These results suggest that the framework can maintain stable learning behavior as the number of clients increases in the synthetic simulation. The faster early convergence observed with 20 clients may be explained by broader client participation and more diverse aggregated updates. However, this interpretation should be treated cautiously because the experiment is based on synthetic data and fixed simulation settings.

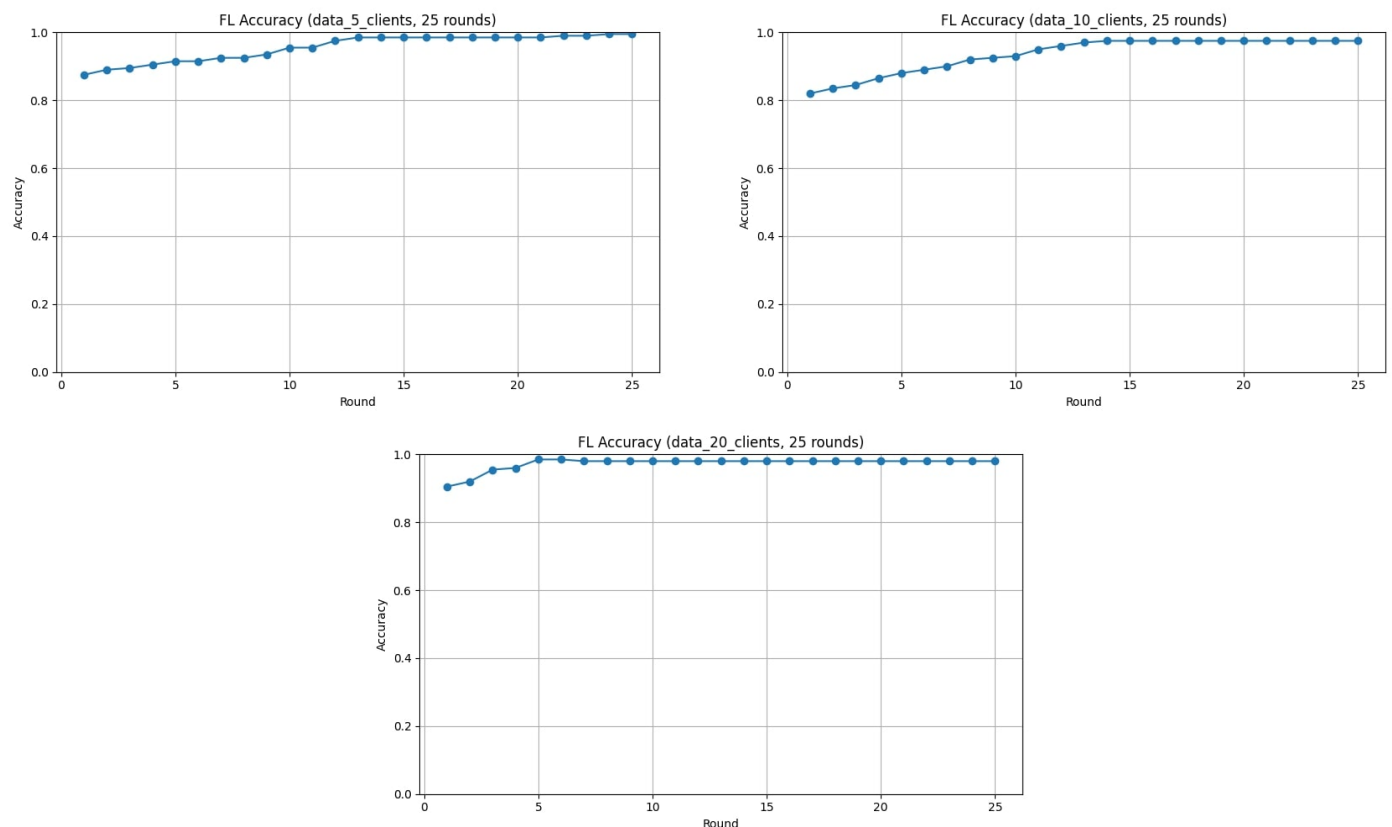


Figure 6. Federated learning accuracy with 5, 10, and 20 clients.

5.4. Aggregation Strategy Evaluation

The fourth experiment compares three aggregation strategies: FedAvg, weighted aggregation, and accuracy-trimmed robust aggregation. The objective is to evaluate whether the choice of aggregation rule affects convergence behavior under the same 10-client federated setting.

Figure 7 shows that all three aggregation strategies produce stable convergence and reach similar final accuracy. FedAvg provides a strong baseline and increases from approximately 0.82 to approximately 0.98. Weighted aggregation follows a similar trajectory and reaches nearly the same final performance. This similarity is expected because the synthetic datasets used in this experiment contain the same number of samples per client. Under equal client sizes, weighted aggregation becomes mathematically close to FedAvg.

The accuracy-trimmed robust aggregation strategy also converges smoothly and produces comparable final accuracy. In this experiment, its advantage is not a large final-accuracy gain. Rather, its role is to reduce the influence of unusually low-performing or unusually dominant local models by trimming the lowest- and highest-ranked client models before averaging. The present simulation is not highly adversarial, so the benefit of robust aggregation should be interpreted as a robustness-oriented design property rather than as a demonstrated superiority in final accuracy.

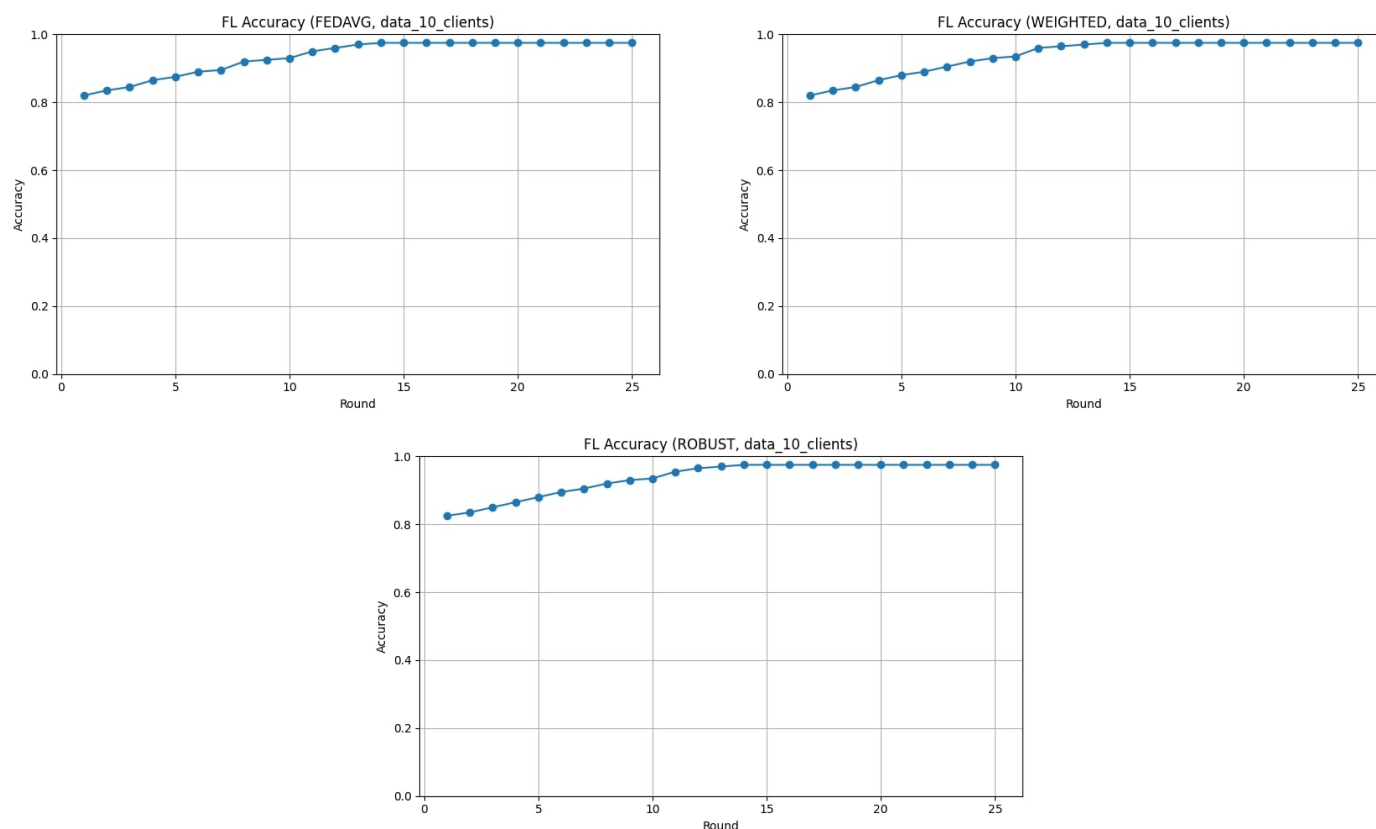


Figure 7. Performance comparison of FedAvg, weighted aggregation, and accuracy-trimmed robust aggregation under the 10-client setting.

5.5. Summary of Experimental Findings

Table 6 summarizes the observed experimental results. The reported values are approximate final accuracies read from the corresponding simulation outputs and convergence plots. They are not reported as mean values across multiple independent random seeds.

Table 6. Summary of observed experimental results under controlled synthetic simulation settings.

Experiment	Observed Final Accuracy	Main Observation
Centralized MLP	≈ 0.89	High overall accuracy but 0.00 alert-class recall
Local-only MLP	0.80–0.97	Strong client-level variation under non-IID data
Federated noisy non-IID setting	≈ 0.80	Stable convergence under noisy data, non-IID distribution, dropout, and fixed skewed-client evaluation
Clean labels	≈ 0.99	Rapid and stable convergence
Mild label noise, 10%	≈ 0.90	Moderate degradation but stable convergence
Heavy label noise, 30%	≈ 0.70	Substantial degradation under stronger label corruption
5 clients	≈ 0.99	Stable convergence with small federation size
10 clients	≈ 0.98	Stable convergence with moderate federation size
20 clients	≈ 0.98	Fast early convergence with broader client participation
FedAvg	≈ 0.98	Strong baseline aggregation performance
Weighted aggregation	≈ 0.98	Similar final performance to FedAvg under equal client sizes
Accuracy-trimmed robust aggregation	≈ 0.98	Similar final performance with robustness-oriented trimming

The results show that the proposed federated learning framework can maintain stable learning behavior under controlled non-IID and noisy synthetic conditions. The label noise experiment demonstrates that data quality is a major determinant of convergence. The client scaling experiment suggests that the framework remains stable as the number of clients increases from 5 to 20 in the simulation. The aggregation comparison shows that FedAvg, weighted aggregation, and accuracy-trimmed robust aggregation perform similarly in the current setting.

These findings should be interpreted within the limits of the experimental design. The current results do not establish clinical effectiveness, do not report multi-seed statistical confidence, and do not demonstrate that blockchain anchoring improves classification accuracy. Rather, the federated learning results show convergence behavior under controlled synthetic conditions, while the blockchain layer contributes auditability through model-hash and participation-record anchoring.

6. Blockchain Anchoring Design

This section describes the lightweight blockchain anchoring design used to support auditability in the proposed federated learning framework. The anchoring layer is not designed to perform model training, execute aggregation, or store patient-level data. Instead, it records compact metadata that can be used to verify model provenance and client participation after each federated learning round. This design follows the principle that blockchain should be used only where immutability and tamper-evident recordkeeping provide clear value, while computationally intensive learning operations remain off-chain.

6.1. Anchoring Objectives and Scope

The main objective of the blockchain anchoring layer is to make the federated learning process more traceable. In a conventional coordinator-based federated learning system, participating clients may not have a persistent and tamper-evident record of which model version was produced, which clients participated, and when each training round occurred. This may create difficulties during audit, compliance review, or post hoc investigation of model behavior.

The proposed anchoring layer addresses this gap by recording model hashes and participation metadata on a permissioned blockchain. The blockchain record does not contain raw patient data, local training records, gradients, or complete model parameters. It contains only compact evidence that can be used to verify whether a model artifact corresponds to a specific training round. Therefore, the blockchain layer supports auditability and provenance, but it should not be interpreted as a complete privacy or security mechanism.

This scope is important because blockchain anchoring alone does not prevent gradient inversion, membership inference, model poisoning, or collusion among clients. These risks require additional defenses such as differential privacy, secure aggregation, secure multi-party computation, robust anomaly detection, or zero-knowledge verification. In the present framework, blockchain anchoring is used specifically to strengthen traceability and accountability in a lightweight manner.

6.2. Anchoring Record Structure

At the end of each communication round, the coordinator serializes the resulting global model and computes a cryptographic hash. The hash acts as a compact fingerprint of the model version. If the model artifact is later modified, the recomputed hash will differ from the anchored value.

Let $\mathbf{w}^{t+1}$ denote the global model parameters produced after communication round t . The model hash is computed as

$$h^{t+1} = H(\text{Serialize}(\mathbf{w}^{t+1})), \quad (17)$$

where $H(\cdot)$ denotes a cryptographic hash function, $\text{Serialize}(\cdot)$ denotes deterministic model serialization, and h^{t+1} denotes the hash of the global model version.

The anchoring record for round $t + 1$ is defined as

$$r^{t+1} = (t + 1, h^{t+1}, A^{t+1}, S_t, \tau^{t+1}, p^{t+1}), \quad (18)$$

where $t + 1$ is the round identifier, h^{t+1} is the global model hash, A^{t+1} is the aggregation strategy used in the round, S_t is the set of participating clients, τ^{t+1} is the timestamp, and p^{t+1} is the participation proof associated with the round.

Table 7 summarizes the fields stored in the anchoring record.

Table 7. Blockchain anchoring record schema.

Field	Description	Stored Content
Round identifier	Communication round index	Integer value
Model hash	Hash of the serialized global model	Cryptographic hash
Aggregation strategy	Aggregation rule used in the round	FedAvg, weighted, or robust
Participant set	Clients participating in the round	Client identifiers or pseudonyms
Timestamp	Time of anchoring event	System timestamp
Participation proof	Evidence of client participation	Hash or signed metadata

The participant set may be stored using institutional identifiers, pseudonyms, or hashed client identifiers depending on the governance policy of the healthcare consortium. In privacy-sensitive settings, pseudonymous identifiers are preferable because they reduce unnecessary exposure of institutional participation patterns.

6.3. Round-Level Anchoring Workflow

The anchoring workflow is executed after the coordinator completes aggregation in each communication round. The process is intentionally separated from model training so that blockchain operations do not interrupt local learning or global aggregation.

The workflow consists of five steps. First, clients train local models and submit their model updates to the coordinator. Second, the coordinator applies the selected aggregation strategy and produces the next global model. Third, the global model is serialized and hashed. Fourth, the coordinator prepares the anchoring record containing the round identifier, model hash, aggregation strategy, participant set, timestamp, and participation proof. Finally, the anchoring record is submitted to the permissioned blockchain.

The round-level anchoring procedure is summarized as follows.

1. The coordinator receives local model updates from participating clients.
2. The coordinator aggregates the submitted updates and produces $\mathbf{w}^{t+1}$.
3. The global model is serialized using a deterministic serialization process.
4. The model hash h^{t+1} is computed using Equation (17).
5. The anchoring record r^{t+1} is constructed using Equation (18).
6. The anchoring record is submitted to the permissioned blockchain.
7. The blockchain returns a transaction identifier that can be used for later audit.

This design keeps patient data and model parameters off-chain. The blockchain stores only compact metadata, while the model artifact remains in the federated learning

environment. This separation reduces storage overhead and avoids exposing complete model parameters through the ledger.

6.4. Verification and Audit Process

The anchored records support two forms of verification: model integrity verification and participation verification. Model integrity verification checks whether a model artifact corresponds to a previously anchored model version. Given a stored model artifact $\mathbf{w}^{t+1}$, an auditor can recompute its hash and compare it with the blockchain record:

$$H(\text{Serialize}(\mathbf{w}^{t+1})) \stackrel{?}{=} h^{t+1}. \quad (19)$$

If the two hashes match, the model artifact is consistent with the anchored record. If they do not match, the model artifact may have been modified, incorrectly stored, or replaced.

Participation verification checks whether a client was recorded as participating in a specific communication round. The auditor retrieves the anchoring record r^{t+1} and examines the participant set S_t or the corresponding participation proof p^{t+1} . This supports accountability when institutions need to confirm whether a client contributed to a given round.

The audit process does not require access to raw patient records. It only requires the model artifact, the corresponding blockchain record, and the verification procedure. This is useful in healthcare settings because auditability can be supported without centralizing sensitive patient data.

6.5. Lightweight Storage and Overhead Considerations

The anchoring layer is designed to minimize storage overhead by avoiding full model storage on-chain. If B_r denotes the approximate size of one anchoring record and T denotes the number of communication rounds, the total blockchain storage required for a training task can be expressed as

$$B_{total} = T \times B_r. \quad (20)$$

Because B_r contains only metadata such as hashes, identifiers, timestamps, and proof values, it is substantially smaller than storing full model parameters. This is the main reason that the proposed framework uses blockchain anchoring rather than on-chain model storage or on-chain aggregation.

The computational overhead of anchoring mainly comes from model serialization, hash computation, transaction submission, and transaction confirmation. These operations are performed after aggregation and can be executed asynchronously. As a result, the anchoring mechanism can support auditability without directly affecting local training. However, actual latency and throughput depend on the selected blockchain platform, consensus mechanism, network configuration, and transaction policy.

In this study, the blockchain component is treated as a lightweight design layer for auditability. A full deployment-level benchmark of blockchain latency, throughput, and storage growth is outside the present experimental scope. Future work should evaluate these operational metrics using a concrete permissioned blockchain implementation such as Hyperledger Fabric or a comparable consortium blockchain platform.

6.6. Deployment Scenario in Healthcare Networks

In a practical healthcare deployment, the proposed anchoring layer can be implemented within a consortium of hospitals, clinics, or approved patient monitoring providers. Each institution operates a federated client that trains locally on its own monitoring data.

The coordinator manages model aggregation, while the permissioned blockchain records evidence of model versions and client participation.

This deployment model is suitable for semi-trusted healthcare networks. Participants are known and authorized, but the system still benefits from tamper-evident records because institutions may require verifiable evidence of model history, training participation, and audit trails. For example, if a global patient monitoring model is later reviewed by a regulatory body or institutional committee, the anchored model hashes can help verify which version was used and when it was produced.

The anchoring design also supports future integration with stronger governance mechanisms. Client authentication can be extended using decentralized identity or verifiable credentials. Consent policies can be connected to smart-contract-based access rules. Stronger privacy mechanisms, such as secure aggregation or differential privacy, can be added to protect model updates. These extensions are compatible with the proposed architecture, but they are not required for the lightweight anchoring design evaluated in this study.

In summary, the blockchain anchoring layer provides an audit-oriented complement to federated learning. It does not improve classification accuracy directly and does not replace formal privacy-preserving computation. Its value lies in making the model training process more transparent, traceable, and accountable while preserving the off-chain efficiency of federated learning.

7. Discussion

The experimental results provide several insights into the design of privacy-preserving patient monitoring systems based on federated learning and lightweight blockchain anchoring. The findings should be interpreted at the framework level because the evaluation was conducted using controlled synthetic patient monitoring data. The results do not establish clinical effectiveness. Rather, they clarify how federated learning behaves under heterogeneous and noisy client conditions and how blockchain anchoring can complement the learning process through traceability and auditability.

The main interpretation of this study is that federated learning, aggregation strategy, and blockchain anchoring address different parts of the healthcare trust problem. Federated learning supports data locality by allowing clients to train without transferring raw patient records. Aggregation strategies influence convergence behavior under heterogeneous client conditions. Blockchain anchoring provides tamper-evident evidence of model versions and client participation. These mechanisms are complementary, but they should not be treated as interchangeable security guarantees.

7.1. Federated Learning Behavior Under Heterogeneous Client Conditions

The baseline results show that centralized learning can produce misleadingly high overall accuracy when the dataset is imbalanced. Although the centralized MLP model correctly classified most normal samples, it failed to identify alert cases in the confusion matrix. This finding is important for patient monitoring because alert cases are usually less frequent but clinically more important. A model that performs well mainly on normal cases may still be unsuitable for monitoring tasks if it misses abnormal or alert conditions.

The local-only results also show that individual clients can achieve different accuracy levels depending on their local data distributions. This reflects the non-IID nature of distributed patient monitoring environments. In practical healthcare networks, each institution or monitoring node may observe different patient populations, device conditions, annotation practices, and alert frequencies. Therefore, local-only training may produce models that are useful within individual sites but less reliable as shared monitoring intelligence.

The federated learning results indicate that collaborative training can provide stable global behavior under distributed conditions. In the noisy, non-IID, dropout, and fixed skewed-client evaluation setting, the global model converged steadily and stabilized at approximately 0.80 accuracy. This result is lower than the clean-data experiments, but it is more informative because it reflects a more difficult simulation environment. The result suggests that federated learning can maintain usable convergence behavior even when client data are imperfect and unevenly distributed.

However, this finding should be interpreted carefully. The federated convergence experiment used a fixed skewed client for evaluation rather than a broad external clinical test set. Therefore, the result should be understood as evidence of convergence behavior under a controlled non-IID evaluation condition. It should not be interpreted as proof that the model would generalize to real patient populations without further validation.

7.2. Effect of Label Noise and Data Quality

The label noise experiment shows that data quality is a major determinant of federated learning performance. The clean setting reached near-perfect accuracy, while the mild-noise setting with 10% random label flipping stabilized at a lower but still consistent level. In contrast, the heavy-noise setting with 30% random label flipping produced a much lower convergence curve. This pattern indicates that repeated aggregation cannot fully compensate for corrupted local labels.

This result has practical implications for patient monitoring. In real healthcare environments, labels may be uncertain due to delayed confirmation, ambiguous clinical interpretation, sensor error, or inconsistent institutional annotation. If noisy labels are distributed across clients, they may affect the global model through repeated aggregation. Therefore, practical deployment should include data-quality screening, label validation, anomaly detection, or client reliability assessment before or during federated learning.

The result also supports the need for robust aggregation, but with an important qualification. Robust aggregation can reduce the influence of unstable or unusual client updates, yet it cannot fully repair systematic label corruption. If many clients train on corrupted labels, the global model may still degrade. Therefore, robustness should be combined with local data governance rather than treated as a replacement for label quality control.

7.3. Client Scaling and Aggregation Strategy

The client scaling experiment shows that the framework remains stable when the number of clients increases from 5 to 10 and 20 under the controlled synthetic setting. All three settings converged smoothly and reached high final accuracy. This suggests that the proposed framework can support different federation sizes without obvious convergence collapse in the present simulation.

The 20-client setting showed faster early convergence than the smaller-client settings. A possible explanation is that broader participation provides more diverse updates during aggregation. However, this interpretation should be treated cautiously because the results are based on controlled synthetic data. In real healthcare deployments, increasing the number of clients may also increase communication variability, institutional heterogeneity, client dropout, governance complexity, and operational cost.

The aggregation strategy comparison shows that FedAvg, weighted aggregation, and accuracy-trimmed robust aggregation produced similar final accuracy in the current simulation. FedAvg remains a strong baseline. Weighted aggregation is useful when client data sizes are unequal, although its behavior becomes similar to FedAvg when all clients contain the same number of samples. Accuracy-trimmed robust aggregation is useful as

a simple robustness-oriented heuristic, but its advantage may become visible only under stronger client corruption, outlier updates, or adversarial behavior.

Therefore, the main lesson from the aggregation experiment is not that one aggregation strategy clearly dominates the others. Rather, the result shows that the aggregation strategy should be selected according to the expected deployment condition. FedAvg is appropriate as a simple baseline. Weighted aggregation is appropriate when client data sizes differ substantially. Robust aggregation is more relevant when unreliable, noisy, or suspicious client updates are expected.

7.4. Role of Blockchain Anchoring

The results should not be interpreted as evidence that blockchain improves classification accuracy. The learning performance is determined mainly by the data distribution, label quality, model architecture, and aggregation strategy. The role of blockchain anchoring is different. It supports traceability, model provenance, and post hoc auditability.

In the proposed framework, the blockchain layer records compact evidence such as model hashes, round identifiers, timestamps, aggregation metadata, and participation records. These records allow later verification of whether a model artifact corresponds to a specific training round and which clients participated in that round. This is useful in semi-trusted healthcare networks where institutions may collaborate without fully delegating trust to a single coordinator.

This auditability function is important because federated learning still depends on coordination. Even when raw patient data remain local, participants may need evidence that the training process was conducted as expected. Blockchain anchoring provides tamper-evident records for this purpose without storing raw data or full model parameters on-chain. Therefore, the proposed blockchain layer should be understood as an accountability mechanism rather than a universal security mechanism.

This design choice also reflects a practical trade-off. Full on-chain aggregation, complete model storage, or heavy smart-contract execution may increase transparency, but they can also increase latency, storage cost, implementation complexity, and operational burden. Lightweight anchoring takes a narrower approach by storing only compact evidence. This is more suitable for a framework-level patient monitoring design where auditability is needed but full blockchain-based computation is not required.

7.5. Privacy and Security Boundaries

The proposed framework preserves data locality because raw patient records are not transmitted during training. However, data locality should not be equated with complete privacy. Model updates may still expose information under strong inference attacks. Blockchain anchoring also does not prevent gradient inversion, membership inference, model poisoning, or collusion among clients.

This distinction is central to the interpretation of the framework. Federated learning reduces the need for raw data centralization. Aggregation strategies improve learning stability under heterogeneous client conditions. Blockchain anchoring improves auditability and model provenance. These mechanisms are complementary, but none of them alone solves the entire privacy and security problem.

Several stronger privacy and security alternatives could be integrated into future versions of the framework. Differential privacy can reduce information leakage from model updates, but it requires explicit privacy parameters, noise mechanisms, clipping rules, and privacy accounting. Secure multi-party computation and secure aggregation can protect individual client updates during aggregation, but they increase communication and computation overhead. Homomorphic encryption can enable computation over encrypted

updates, but it is often expensive for iterative model training. Trusted execution environments can protect computation inside isolated hardware-backed environments, but they introduce hardware dependency and trust assumptions. Zero-knowledge verification can support stronger proof of correct computation, but it adds protocol complexity.

These alternatives are not direct replacements for blockchain anchoring. They address different risks. Privacy-preserving computation protects data or model updates during training. Robust aggregation mitigates unreliable or malicious contributions. Blockchain anchoring records evidence for audit and accountability. A practical healthcare deployment may combine several of these mechanisms, but the selection should depend on the risk level, latency tolerance, governance model, and available infrastructure.

7.6. Operational and Regulatory Considerations

Healthcare deployment requires more than model accuracy. A practical system must support accountability, access control, audit trails, data governance, and responsibility assignment. These requirements are increasingly important because healthcare AI systems are expected to be explainable, traceable, and governed throughout their life cycle. In this context, blockchain anchoring can support evidence retention by recording model version hashes and client participation history.

However, blockchain also introduces operational costs. Even a permissioned blockchain requires node operation, access management, identity management, storage planning, backup policies, monitoring, and maintenance. If every communication round produces an anchoring transaction, the ledger grows with the number of rounds, models, and participating clients. The storage burden may be modest for hash-only anchoring, but it is not zero. Energy consumption is also relevant, especially if the network uses inefficient consensus mechanisms or runs across multiple institutions with redundant infrastructure.

For clinical deployment, these costs must be justified by the governance value of the audit trail. A lightweight anchoring design is easier to justify when it supports model provenance, dispute resolution, institutional accountability, or compliance review. It is harder to justify if the blockchain layer is added without a clear audit purpose. Therefore, the proposed framework uses blockchain only as an evidence layer and avoids placing raw data, full model parameters, or full aggregation logic on-chain.

Regulatory expectations also vary across jurisdictions and healthcare systems. Practical implementation would need to align with local requirements for patient privacy, medical data governance, AI risk management, cybersecurity, auditability, and institutional accountability. The present study does not claim regulatory compliance. Instead, it provides a technical foundation that can be extended with jurisdiction-specific governance controls, consent policies, audit access rules, and model approval workflows.

7.7. Limitations

The findings of this study should be interpreted within several methodological and operational boundaries. The most important limitation is the use of synthetic patient monitoring data. Although synthetic data allows the experimental conditions to be controlled, including non-IID client distributions, label noise, and client scaling, it cannot fully capture the complexity of real clinical data. Real patient monitoring data may contain missing values, device artifacts, temporal dependencies, inconsistent sampling rates, and clinically ambiguous labels that are not fully represented in the present simulation. Therefore, the results should be understood as framework-level evidence rather than clinical validation.

Another limitation concerns the scope of the experimental evaluation. The current results focus mainly on accuracy and convergence behavior because the objective was to examine whether the federated learning process remains stable under heterogeneous

and noisy client settings. However, patient monitoring requires more detailed evaluation than overall accuracy alone. The centralized confusion matrix shows that a model can achieve high overall accuracy while failing to detect alert cases, which is clinically undesirable. Future evaluation should therefore report precision, recall, F1-score, ROC-AUC, alert-class sensitivity, and confusion matrices in a systematic manner across all major experimental settings.

The current experiments also do not report mean and standard deviation across multiple random seeds. The results were produced under fixed random seeds and should therefore be interpreted as reproducible controlled simulation outcomes rather than statistical estimates across repeated trials. Multi-seed evaluation would require rerunning the complete data generation, federated training, aggregation comparison, and client scaling pipeline. This is an important next step for strengthening the empirical evidence.

The blockchain component is evaluated as an anchoring design rather than as a fully benchmarked deployment. The proposed design explains how model hashes and participation records can support auditability, but the current study does not measure transaction confirmation time, throughput, storage growth, energy cost, or anchoring overhead in a real permissioned blockchain network. These operational metrics are necessary before the framework can be considered ready for deployment in a hospital consortium or healthcare monitoring network.

The robustness evaluation is limited to label noise, non-IID data, client scaling, dropout, and aggregation strategy comparison. These settings are useful for controlled framework evaluation, but they do not cover stronger adversarial conditions such as model poisoning, gradient inversion, membership inference, or client collusion. The framework also assumes a semi-trusted healthcare environment in which participating clients are registered and authorized. This assumption is reasonable for hospital consortia or approved monitoring networks, but it may not be sufficient for open or highly adversarial systems.

Finally, the current study does not implement differential privacy, secure aggregation, secure multi-party computation, homomorphic encryption, trusted execution environments, or zero-knowledge verification. These mechanisms are discussed as compatible extensions, not as implemented components. This boundary is important because blockchain anchoring alone does not provide formal privacy protection against inference attacks.

7.8. Future Work

Future work should first extend the evaluation to real-world patient monitoring datasets or publicly available healthcare datasets. This would allow the framework to be tested under more realistic feature distributions, missing data patterns, clinical labels, temporal dependencies, and alert frequencies. Such validation is necessary to determine whether the convergence behavior observed in the synthetic simulation can be reproduced under practical healthcare conditions.

Further experiments should also broaden the performance evaluation beyond accuracy. In patient monitoring, false negatives and false positives have different operational consequences. A missed alert may delay intervention, while excessive false alerts may increase alarm fatigue. Therefore, future studies should report precision, recall, F1-score, ROC-AUC, alert-class sensitivity, and confusion matrices across all major experimental settings.

Future work should also include multi-seed evaluation. Running the complete experimental pipeline across multiple random seeds would allow mean and standard deviation values to be reported for each major configuration. This would make the comparison of label noise levels, client counts, and aggregation strategies more statistically reliable.

The blockchain anchoring layer should be implemented and benchmarked on a concrete permissioned blockchain platform. Future work should measure transaction con-

firmation time, throughput, ledger storage growth, energy cost, and anchoring overhead compared with a no-anchoring baseline. These measurements would clarify the practical cost of auditability and help determine whether lightweight anchoring remains feasible as the number of clients and training rounds increases.

Future versions of the framework should also integrate stronger privacy and robustness mechanisms. Differential privacy, secure aggregation, secure multi-party computation, homomorphic encryption, trusted execution environments, and zero-knowledge verification can provide stronger protection against inference attacks, malicious updates, and unverifiable computation. However, these mechanisms should be evaluated not only by their security benefits but also by their effects on accuracy, latency, communication cost, implementation complexity, and deployment feasibility.

Finally, the framework should be extended toward deployment-oriented governance. Practical healthcare adoption requires more than model training and blockchain logging. Future work should examine client registration, institutional authorization, consent-aware participation, audit access, model version approval, incident response, and responsibility assignment among participating institutions. These governance mechanisms are essential for moving the framework from technical feasibility toward real-world healthcare adoption.

8. Conclusions

This study presented a federated learning framework for privacy-preserving patient monitoring with lightweight blockchain anchoring. The framework was designed for semi-trusted healthcare environments in which raw patient monitoring data should remain local, while the model training process still requires traceability and accountability. Federated learning was used to support collaborative model training across distributed clients, and blockchain anchoring was specified as a lightweight mechanism for recording compact evidence of model evolution and client participation.

The experimental results show that federated learning can maintain stable convergence under controlled synthetic non-IID and noisy patient monitoring conditions. The centralized baseline achieved high overall accuracy but failed to detect alert cases, demonstrating that accuracy alone can be misleading in imbalanced monitoring scenarios. The label noise experiment showed that data quality strongly affects convergence, with clean data producing the strongest performance and heavier label corruption causing substantial degradation. The client scaling experiment showed stable convergence across 5, 10, and 20 clients under the synthetic simulation settings. The aggregation comparison showed that FedAvg, weighted aggregation, and accuracy-trimmed robust aggregation reached similar final accuracy in the current non-adversarial setting.

The main contribution of this work is not a claim that blockchain improves classification accuracy. Rather, the contribution lies in connecting three complementary system concerns: federated learning for data locality, aggregation strategies for learning stability, and blockchain anchoring for auditability. The proposed anchoring layer records model hashes and participation metadata without storing raw patient records or full model parameters on-chain. This design provides a lightweight basis for model provenance and post hoc verification while preserving the efficiency of off-chain learning.

The study also clarifies the boundaries of the proposed framework. Federated learning reduces raw data centralization, but it does not guarantee complete privacy. Blockchain anchoring improves traceability, but it does not prevent inference attacks, poisoning attacks, or client collusion. Robust aggregation can improve stability under selected unreliable-client conditions, but it does not replace data quality control or formal adversarial defense. These distinctions are important for designing realistic healthcare AI systems.

The current findings should therefore be interpreted as framework-level evidence from controlled synthetic simulation rather than clinical validation. Future work should extend the evaluation using real-world or public healthcare datasets, report class-balanced metrics across all major experimental settings, conduct multi-seed evaluation with mean and standard deviation, benchmark blockchain anchoring overhead on a concrete permissioned blockchain platform, and integrate stronger privacy-preserving mechanisms such as differential privacy, secure aggregation, secure multi-party computation, homomorphic encryption, trusted execution environments, and zero-knowledge verification. These extensions are necessary for moving the framework from controlled simulation toward practical deployment in healthcare monitoring networks.

Author Contributions: Conceptualization, T.S. and K.Y.; methodology, T.S.; software, T.S.; validation, T.S. and J.S.; formal analysis, T.S.; investigation, T.S.; resources, T.S. and J.S.; data curation, T.S.; visualization, T.S.; writing—original draft preparation, T.S.; writing—review and editing, T.S., K.Y. and J.S.; supervision, K.Y. and J.S.; project administration, T.S. All authors have read and agreed to the published version of the manuscript.

Funding: The article processing charge was funded by Thattapon Surasak.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study were synthetically generated according to the procedures described in Section 4. The generated datasets and scripts can be made available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Benabderrahmane, F.; Kerkouche, E.; Bouchemal, N. Risk-Aware Privacy-Preserving Federated Learning for Remote Patient Monitoring: A Multi-Layer Adaptive Security Framework. *Appl. Sci.* **2026**, *16*, 29. [\[CrossRef\]](#)
2. Javed, M.S.; Hennache, A.; Imran, M.; Khan, M.K. AI-Driven Blockchain and Federated Learning for Secure Electronic Health Records Sharing. *Electronics* **2025**, *14*, 4774. [\[CrossRef\]](#)
3. Zhu, X.; Li, H. Privacy-Preserving Poisoning-Resistant Blockchain-Based Federated Learning for Data Sharing in the Internet of Medical Things. *Appl. Sci.* **2025**, *15*, 5472. [\[CrossRef\]](#)
4. Mu, J.; Kadoch, M.; Yuan, T.; Lv, W.; Liu, Q.; Li, B. Explainable Federated Medical Image Analysis Through Causal Learning and Blockchain. *IEEE J. Biomed. Health Inform.* **2024**, *28*, 3206–3218. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Chittem, M.B.; Ghanta, S.; Cherukuri, V.A.; Lingincni, P.S.; Jalluri, R.; Pradhan, A.K. Blockchain Verified Federated Learning for Brain Tumour Classification using Colorized MRI Scans. In Proceedings of the 2025 IEEE 4th International Conference on Data, Decision and Systems (ICDDS), Dharwad, India, 4–6 December 2025; pp. 135–140. [\[CrossRef\]](#)
6. Ebrahimi, E.; Sober, M.; Hoang, A.T.; Ileri, C.U.; Sanders, W.; Schulte, S. Blockchain-based Federated Learning Utilizing Zero-Knowledge Proofs for Verifiable Training and Aggregation. In Proceedings of the 2024 IEEE International Conference on Blockchain (Blockchain), Copenhagen, Denmark, 19–22 August 2024; pp. 54–63. [\[CrossRef\]](#)
7. Zeydan, E.; Blanco, L.; Mangues-Bafalluy, J.; Arslan, S.S.; Turk, Y.; Kumar Yadav, A.; Liyanage, M. Blockchain-Based Self-Sovereign Identity: Taking Control of Identity in Federated Learning. *IEEE Open J. Commun. Soc.* **2024**, *5*, 5764–5781. [\[CrossRef\]](#)
8. Commey, D.; Hounsinnou, S.G.; Crosby, G.V. Post-Quantum Secure Blockchain-Based Federated Learning Framework for Healthcare Analytics. *IEEE Netw. Lett.* **2025**, *7*, 126–129. [\[CrossRef\]](#)
9. Liu, X.; Li, X.; Gu, A.; Lv, S.; Su, J. A Blockchain-Based Trustworthy Federated Learning Privacy Protection Scheme. In Proceedings of the 2024 IEEE Wireless Communications and Networking Conference (WCNC), Dubai, United Arab Emirates, 21–24 April 2024; pp. 1–6. [\[CrossRef\]](#)
10. Qin, J.; Yang, B.; Wang, H. A Blockchain-Based Framework for Enhancing Adaptive Differential Privacy in Federated Learning. In Proceedings of the 2025 5th International Conference on Neural Networks, Information and Communication Engineering (NNICE), Guangzhou, China, 10–12 January 2025; pp. 1166–1169. [\[CrossRef\]](#)

11. Liao, B.S.; Li, J.S.; Liu, I.H.; Liu, C.K. Blockchain-Based Adaptive Historical Averaging for Client Dropout Resilience in Federated Learning. In Proceedings of the 2025 Seventh International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, 27–30 June 2025; pp. 1–4. [\[CrossRef\]](#)
12. Madupati, B.; Somayajula, R.; Vegesna, R.V.; Dasi, U. Edge-Aware Federated Learning: A Scalable and Fault-Tolerant System Architecture. In Proceedings of the 2025 Global Conference in Emerging Technology (GINOTECH), Pune, India, 9–11 May 2025; pp. 1–7. [\[CrossRef\]](#)
13. Hung, P.H.; Wang, B.S.; Chang, Y.W.; Kuo, T.T.; Wang, L.J.; Lin, C.Y. An Implementation of Blockchain-based Federated Learning Framework. In Proceedings of the 2024 International Conference on Consumer Electronics—Taiwan (ICCE-Taiwan), Taichung, Taiwan, 9–11 July 2024; pp. 165–166. [\[CrossRef\]](#)
14. Peng, Z.; Song, Y.; Wang, Q.; Xiao, X.; Tang, Z. FedBN: A Communication-Efficient Federated Learning Strategy Based on Blockchain. In Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Tianjin, China, 8–10 May 2024; pp. 754–759. [\[CrossRef\]](#)
15. De Souza, L.A.C.; Camilo, G.F.; Rebello, G.A.F.; Guimarães, L.C.B.; Campista, M.E.M.; Costa, L.H.M.K. Blockchain-based Approaches for Secure Federated Learning. In Proceedings of the 2024 6th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Berlin, Germany, 9–11 October 2024; pp. 1–4. [\[CrossRef\]](#)
16. Wang, T. Trustworthiness Verification for Federated Learning in Web3.0 Healthcare Communities. In Proceedings of the 2025 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Wuhan, China, 15–18 December 2025; pp. 2953–2959. [\[CrossRef\]](#)
17. Ma, H.; Huang, S.; Guo, J.; Lam, K.Y.; Yang, T. Blockchain-Based Privacy-Preserving Federated Learning for Mobile Crowdsourcing. *IEEE Internet Things J.* **2024**, *11*, 13884–13899. [\[CrossRef\]](#)
18. Prasad, V.J.; Prabhu, S.N. Blockchain-based Smart Contract Frameworks for Secure AI Healthcare Systems: A Systematic Review of Privacy-Preserving Methodologies. In Proceedings of the 2026 7th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 20–22 May 2026; pp. 202–208. [\[CrossRef\]](#)
19. Singh, K.N.; Singh, A.K. Blockchain-Based Federated Learning Approach for Privacy Preservation in IoT-Based Healthcare. In Proceedings of the 2025 International Conference on Engineering Innovations and Technologies (ICoEIT), Bhopal, India, 4–5 July 2025; pp. 427–431. [\[CrossRef\]](#)
20. Pan, X.; Ding, L.; Zhong, B.; Shen, L.; Wang, Y. Collaborative Governance: Blockchain-Based Federated Learning for Construction Safety Service. *IEEE Trans. Eng. Manag.* **2025**, *72*, 1163–1177. [\[CrossRef\]](#)
21. Veerasamy, V.; Sampath, L.P.M.I.; Singh, S.; Nguyen, H.D.; Gooi, H.B. Blockchain-Based Decentralized Frequency Control of Microgrids Using Federated Learning Fractional-Order Recurrent Neural Network. *IEEE Trans. Smart Grid* **2024**, *15*, 1089–1102. [\[CrossRef\]](#)
22. Kumar, V.; Niazi, M.A.; Sajid, Q.; Koondhar, M.A.; Kim, Y.S.; Ammirul Atiqi Mohd Zainuri, M.; Alkoradees, A.F. A Federated Learning Framework for Proactive Congestion Management in Blockchain-Based P2P Energy Markets. *IEEE Access* **2026**, *14*, 37509–37524. [\[CrossRef\]](#)
23. Lin, J. BFTChain-Net: A Blockchain-based Federated Transformer Network for Cross-Organization Accounting Fraud Detection. In Proceedings of the 2025 4th International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 10–12 December 2025; pp. 204–210. [\[CrossRef\]](#)
24. Mishra, S.; Prajapati, Y.; Aggarwal, K.; Pal, S. BloBS-FL: A Robust Architecture of Blockchain Based Decentralized Split Federated Learning. In Proceedings of the 2026 18th International Conference on COMMunication Systems and NETWORKS (COMSNETS), Bengaluru, India, 6–10 January 2026; pp. 1349–1354. [\[CrossRef\]](#)
25. Guo, S.; Li, L.; Du, J. Joint-FU: Blockchain-Based Federated Feature Unlearning Method. In Proceedings of the 2025 IEEE 31th International Conference on Parallel and Distributed Systems (ICPADS), Hefei, China, 14–18 December 2025; pp. 1–9. [\[CrossRef\]](#)
26. Sullivan, B.; Svendsen, S.; Khan, J.A. Flower: Federated Learning Based Zero-Trust Consensus Protocol for Real-Time Trajectory Endorsement in Cavs. In Proceedings of the 2025 IEEE 101st Vehicular Technology Conference (VTC2025-Spring), Oslo, Norway, 17–20 June 2025; pp. 1–7. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.