International Journal of Intelligent Systems

WILEY

SPECIAL ISSUE Zero Trust Security in Intelligent Systems for Smart Environments

RESEARCH ARTICLE OPEN ACCESS

Blockchain-Based AI Intelligent Systems in Healthcare: A Decentralized Federated Learning Framework With Zero Trust

Martin Wynn | Sepideh Mollajafari | Joe Cox | Mah Rukh | Md Hasibul Alam Ratul

School of Business, Computing and Social Sciences, University of Gloucestershire, Cheltenham, UK

Correspondence: Martin Wynn (mwynn@glos.ac.uk)

Received: 5 November 2025 | **Revised:** 30 June 2026 | **Accepted:** 7 July 2026

Guest Editor: Richard Murray

Keywords: AI | blockchain | Byzantine attack | federated learning | healthcare | IoT | poisoning attack | privacy | security | Sybil attack

ABSTRACT

Artificial intelligence (AI), blockchain (BC), and federated learning (FL) offer significant potential for strengthening data privacy, security, and decentralized decision-making in healthcare. However, healthcare AI systems remain vulnerable to adversarial threats, including data poisoning, Byzantine behavior, Sybil attacks, and unauthorized access. This article provides an integrative review of the pertinent literature from which a conceptual framework for exploring the relationship between BC, FL, and Zero Trust (ZT) is developed in the context of AI in healthcare. An experimental framework for a ZT approach based on BC and FL is then designed, tested, and evaluated. The framework combines strict identity verification, permissioned BC access control, immutable auditability, decentralized model training, and history-aware threat detection to enhance robustness while preserving data privacy. A series of simulations was conducted to evaluate the framework under poisoning, Byzantine, and Sybil attack scenarios, using $F1$-score as the primary model-performance metric and latency, gas cost, and throughput as BC-performance metrics. The results show that poisoning clients were consistently detected and excluded once adversarial behavior began, with only minor temporary performance degradation and subsequent recovery of global model performance. Under Byzantine attacks, the proposed history-aware trust filter improved the global $F1$-score across evaluated configurations, including an increase from 0.7762 under attack to 0.9117 with defense in one configuration. BC operations remained efficient, with latency generally between approximately 19 and 39 ms and throughput of 31.65 transactions per second, although upload operations incurred the highest gas cost. Sybil attack experiments further confirmed that cryptographic identity verification and permissioned access control prevented malicious identities from affecting the FL process, while attack cost increased linearly with no operational gain. The findings demonstrate that the proposed framework can effectively balance privacy preservation, adversarial robustness, auditability, and performance, making it a promising approach for secure healthcare systems.

1 | Introduction

In today's digital environment, organizations impacted by cyber-attacks face a major challenge in reconstructing their complex, interconnected technological systems. As cyber threats evolve rapidly, defense mechanisms often fail to keep pace, resulting in heightened vulnerability and a continually expanding attack surface. This poses significant risks, particularly for sectors handling sensitive data such as healthcare, where breaches can have far-reaching consequences. To mitigate these risks, it is essential for organizations to proactively assess current vulnerabilities and implement robust, adaptive cybersecurity measures that encompass a range of process categories—technological,

human, financial, organizational, and external [1]. Specifically as regards technology deployment, transitioning from outdated legacy systems to advanced technologies such as blockchain (BC) and artificial intelligence (AI) can enhance data security, transparency, and privacy. For example, federated learning (FL), a machine learning technique, can be used in conjunction with BC to train AI models collaboratively in decentralized devices to significantly enhance data security [2]. This is the focus of this article, which puts forward a decentralized FL model with Zero Trust (ZT) as a worked example to illustrate the potential benefits of this combination of these technologies in enhancing security and data privacy in the healthcare environment.

The development of new technologies is a continuous process, driven by the need to improve efficiency, accommodate innovation, and rectify security flaws inherent in previous systems [3]. However, despite these efforts, cyber adversaries frequently outpace security advancements, exploiting newly developed technologies soon after their deployment. This ongoing contest between technological advancement and malicious intrusion underscores the critical importance of resilient, secure architectures in safeguarding digital systems, notably in the healthcare sector [4, 5]. In this context, the ZT approach to security has emerged as of particular significance in sectors such as healthcare where patient data security and privacy is of paramount importance [6]. ZT is a network security strategy that assumes no one or anything can be trusted by default. It was first introduced by the United States Department of Defense in May 2021, as part of a system's architecture which aimed to prevent large-scale cyber-attacks [7, 8]. It requires verification for every request to access a network, even if the requestor is already connected to the network.

Smart sensing and intelligent systems are two rapidly evolving technologies that have revolutionized the way we collect and process data, and both rely on Internet of Things (IoT) devices for data collection. Smart sensing involves the use of IoT sensors to collect data, while intelligent systems involve using algorithms to analyze and interpret that data. However, a distinction is often made between intelligent systems and AI. As noted by Alton [9], "intelligent systems work with patterns just like AI, except intelligent systems don't have the deep neural networks that support self-learning" (para. 11). These technologies have the potential to transform industries and solve some of the most challenging societal problems. As regards healthcare, Kumar et al. [10] (p.1) concluded that deep generative AI "is driving revolutionary advancements across a wide range of medical facilities, specializations, and innovative research," but nevertheless cautioned that "these models face serious obstacles in their implementation in real-time applications, including challenges related to accuracy, cost-effectiveness, privacy, security, and authentication, as well as ethical concerns." In this context, this article addresses the following research questions (RQs):

RQ1. What conceptual framework can be developed to encapsulate the challenges, benefits, and potential of using BC and FL to implement ZT in healthcare digital data systems?

RQ2. Can an experimental framework be designed and tested that integrates FL within BC-based healthcare applications to operationalize ZT—ensuring robust authentication and authorization while enhancing the security and privacy of medical and healthcare data?

Prior research has demonstrated the effectiveness of BC technology in facilitating secure and transparent medical data exchange [11]. However, BC architecture possesses inherent limitations, notably its inability to provide dynamic threat detection. Furthermore, discrete AI and ZT solutions frequently lack robust mechanisms for verifiable trust anchors and transparent accountability. Moreover, these approaches do not comprehensively mitigate advanced threats such as Sybil and Byzantine attacks, and data poisoning. Consequently, despite the strong potential of integrating these complementary technologies, the exploration of integrated frameworks (applications) is relatively sparse in existing research.

The novelty of this study thus lies in the development of a decentralized FL model that is integrated with ZT architecture. This approach not only fortifies the security and privacy of healthcare systems but also provides proactive detection of malicious activities, notably major cyber threats including Sybil, Byzantine, and data poisoning attacks. Few, if any, studies have explored all of them using an integrated technological approach. Here, the proposed robust framework ensures resilient and reliable communication between IoT devices and FL servers within a trustable distributed network.

Following this introduction, Section 2 sets out the research methodology used in the study. Section 3 then assesses the literature of relevance to the research questions, from which a provisional conceptual framework (PCF) is put forward as the overarching context for the primary research phase, thus addressing RQ1. Section 4 then establishes the framework architecture and design, while Section 5 focuses on results and evaluation, addressing RQ2, presenting the findings from the experimental phase of the project. Finally, Section 6 provides a conclusion to the study, summarizing the main contribution, noting limitations, and outlining possible future research areas in the field of study.

2 | Research Methods

This study comprises three main phases involving a combination of research methods and philosophies (Figure 1). In Phase 1, an interpretivist philosophy is assumed, combined with an inductive qualitative approach, to assess the extant literature and develop a PCF for the subsequent phase of the research. Gill and Johnson [12] suggest that such an approach is most appropriate when the research aim is exploratory in nature, and the researchers seek to develop an explanation of the phenomenon being studied. Then, a positivist and quantitative approach is taken in first conducting and then evaluating the controlled experiment in Phases 2 and 3 of the research. The three phases are discussed in more detail below.

Phase 1 comprised an integrative review of the relevant literature to identify key themes and provide the basis for development of the PCF to guide Phase 2 of the research. A range of search strings was used to find appropriate material available on the Internet between January and April 2026, but a systematic literature review was not pursued. Search strings were combinations of the key themes evidenced in the RQs "ZT," "BC," "FL," "Healthcare," and variations on these main themes. Once some initial key sources were located, these provided references to other literature, which was then assessed, which in turn provided other

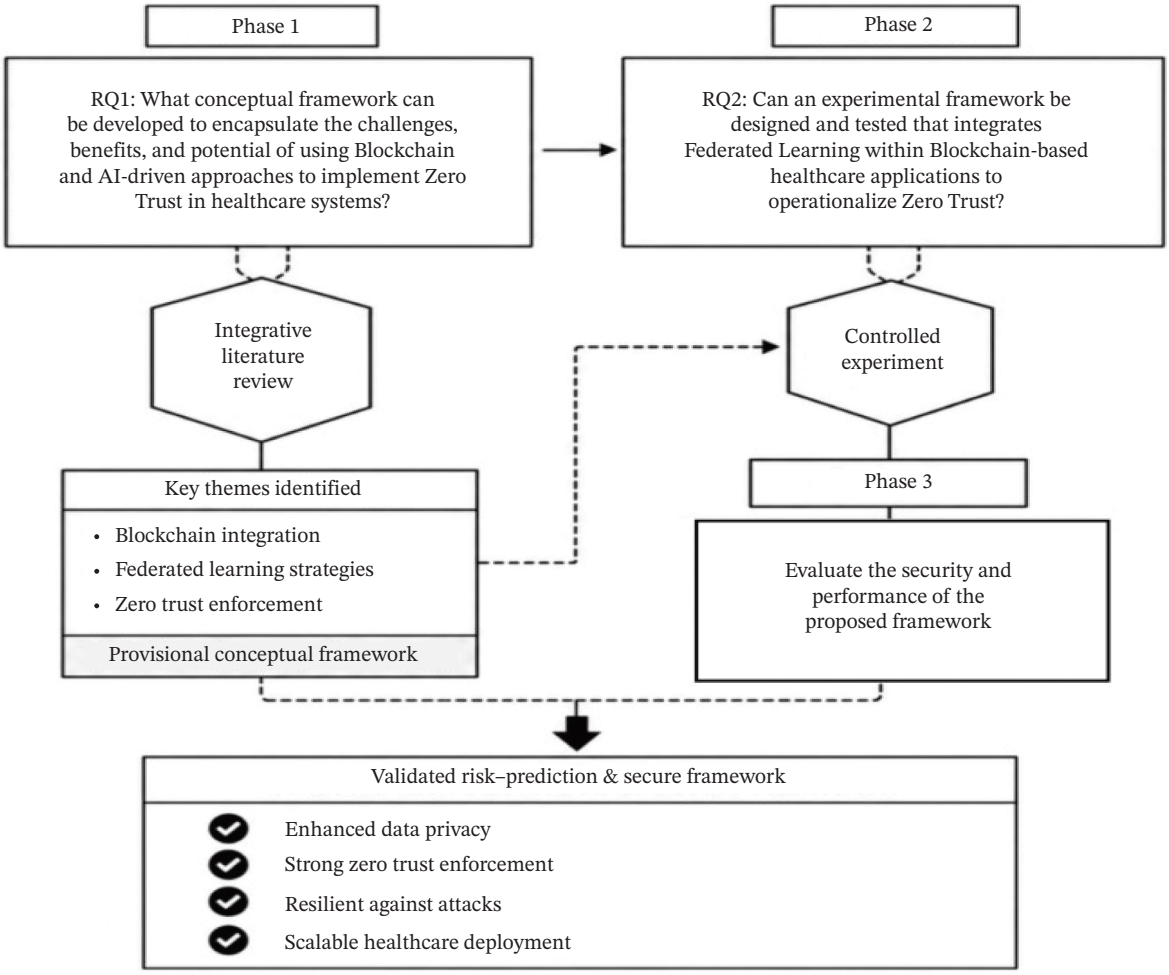

FIGURE 1 | The three-phase research process.

relevant sources. Bell et al. [13] have observed that "the narrative review may be more suitable for qualitative or inductive researchers, whose research strategies are based on an interpretative epistemology" (p.97). Snyder [14] observes that integrative reviews have "the aim to assess, critique, and synthesize the literature on a research topic in a way that enables new theoretical frameworks and perspectives to emerge" (p. 335). Both Hart [15] and Fink [16] maintain that a literature review can assess previous research studies as a basis for subsequent stages in the research process.

From an analysis of the literature, the PCF not only identifies key concepts but can also integrate them into one single structure. The goal is to find factors, attributes, variables, and processes that provide an initial analytical frame for subsequent primary research. Levering [17] notes that a PCF is a good starting point to explain and examine a wider subject area. In similar vein, Jabareen [18] observes that conceptual frameworks connect several concepts in a network to investigate a phenomenon. Phase 1 thus allowed the mapping of the key concepts within the area of study, and identification of the types of evidence that are available [19]. In essence, it was a "broad scan of contextual literature" through which "topical relationships, research trends, and complementary capabilities can be discovered" [20] (p. 351).

Phase 2 focused on the experimental development and implementation of a BC-based framework, incorporating FL and ZT principles to enhance the security and privacy of healthcare records. This phase addressed the detection and mitigation of three major cyber threats: poisoning, Byzantine, and Sybil attacks. Such experiments can be seen as "a method of gathering information and data on a subject through observation in controlled settings" [21] (para. 1). Phase 3 then focused on testing and evaluating the proposed framework based on three main criteria: (1) the robustness of the FL aggregation for global models under varying attacks and their intensity, (2) the security and privacy of healthcare data through different attack detection mechanisms, and (3) the cost of the different operations in terms of gas cost, latency, and throughput metrics. The main tools, attack detection mechanisms, and platforms used in Phases 2 and 3 are detailed in Table 1.

The dataset taken for the proof-of-concept analysis is the Breast Cancer Wisconsin (Diagnostic) dataset [22], hosted on Kaggle, a widely recognized benchmark for evaluating machine learning algorithms in medical diagnosis. It contains 570 instances, each representing a tumor sample described by 30 numerical features that quantify the radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension of cell nuclei derived from digitized breast biopsy

TABLE 1 | Data collection tools and platforms used in Phases 2 and 3 of the research study.

Function	Tools, platforms and metrics
Development tools for smart contracts and FL	BC: Ethereum (Ganache, Local), Docker, Local private IPFS, Web3.py FL: Python programming including sklearn, numpy, pandas, collections, pickle libraries and decision tree as a classifier. ZT: Predefined rules in smart contracts
Testing platforms	Ganache (Local Ethereum), Web3.js Frontend (running on localhost) all ran on local machine
Dataset	Cancer diagnosis dataset
Metrics	Model Performance: Accuracy, *F*1-score, Precision, and Recall Threat Detection: (1) Median absolute deviation and mean–std rules for data poisoning, (2) History-aware filter on *F*1-scores for Byzantine attacks, (benchmarked against Krum and Trimmed Mean under identical attack conditions) and (3) Digital signature and access control for Sybil attack detection. Cost Analysis: Gas cost, latency, and throughput
Storage	Blockchain: IPFS, federated learning: user-server distributed model

images. These measurements include the mean, standard error, and worst (maximum) values for each attribute. The target variable, "diagnosis," classifies tumors as benign (B) or malignant (M), enabling supervised learning models to predict cancerous growths based on cellular characteristics. Because of its structured format, balanced class distribution, and high interpretability, the dataset has become a standard in computational pathology, supporting research in predictive healthcare analytics.

3 | Related Work and PCF

In this section, the first three sections (3.1–3.4) review the literature relevant to the key themes studied in this article: ZT, BC technology, and FL, and the relationships between these concepts as applied in practice, but in particular in the healthcare sector. Then, in Section 3.4, relevant frameworks from the literature of the past 2 years are examined and assessed. Building upon this review of pertinent literature, a PCF is developed to act as a basis for the experimental phase of the study.

3.1 | ZT in Healthcare

Lee et al. [23], in their study of ZT application models, observed "ZT is not a standalone security solution but rather a framework of concepts aimed at achieving a higher level of security. It provides a paradigm that outlines the fundamental philosophy, core principles, and operational guidelines for enhanced security," and "while the guiding principle of 'Never Trust, Always Verify' has gained widespread acceptance, many corporate security managers remain uncertain about how to implement ZT effectively" (p.1). In the context of healthcare, ZT "ensures that no user or device regardless of that user's status in the organization is trusted as authentic until verification. This is true even if the user has logged in many times before, and whether attempts to access the network come from inside or outside the organization. ZT provides a comprehensive set of security strategies that go beyond traditional, perimeter-based measures" [24] (para.9). Moreover, ZT prioritizes a proactive stance against cyber threats within the healthcare system by constantly monitoring and verifying access to critical applications [25].

Edo et al. [26], in making the case for an effective methodology primed on a ZT architecture philosophy, noted that "a critical subject of concern is system vulnerability, with the prevalence of insider attack and recurrent data breaches in the healthcare sector," and that "a more nuanced approach is required to close the insider vulnerability incidences" (p.1). Nevertheless, the challenges of implementing ZT in healthcare are considerable. They include the complexity involved in managing multiple layers of authentication and access controls across various healthcare applications. In addition, the impact of excessive security checks on the workforce may interrupt day-to-day operations and cause discontent among staff. Another aspect is dealing with the legacy of disparate technologies and standalone systems that are not well integrated—applying ZT principles in such technology environments can be problematic [27].

Six main dimensions of ZT security are evident in the literature: verification—every device, app, and user is authenticated before accessing the network; encryption—everything on the network is encrypted; segmentation—the network is segmented to isolate sensitive information and contain attacks; monitoring—user behavior, device health, and service health are assessed; policies—to limit network access, appropriate policies need to be developed and applied; and threat identification—threats are identified in real time. Lee et al. [23] designed a framework that proposes and verifies a method to strengthen security using Secure Sockets Layer (SSL) and Digital Rights Management (DRM), which "effectively support centralized policy management and access control" (p.1). ZT security is a shift from a location-centric model to a more data-centric approach. Specifically in healthcare, the benefits include a reduced risk of data breaches—by continuously verifying access, the likelihood of unauthorized access to sensitive patient data is significantly minimized; enhanced patient privacy—strict access controls help protect sensitive medical information from unauthorized viewing; and improved cybersecurity posture—a proactive approach to security helps identify and mitigate potential threats early on.

The healthcare sector has increasingly adopted IoT devices to support patients through intelligent systems. However, despite the implementation of certain security measures, the current healthcare ecosystem continues to face significant security challenges, including data breaches, unauthorized access, and a lack of transparency. These vulnerabilities pose serious risks to patient data confidentiality, privacy, integrity, trust, and the overall efficiency of healthcare operations [4].

3.2 | BC Technology in Healthcare

BC technology emerges as a potential solution to the security issues in healthcare outlined above. BC offers a decentralized, immutable, and distributed ledger system that enhances data security, confidentiality, integrity, and transparency. By employing advanced cryptographic techniques such as digital signatures, hashing, and zero-knowledge proofs (ZKP), and enforcing access controls through smart contracts, BC enables secure data sharing with robust authentication mechanisms. This is particularly effective in protecting sensitive data collected via IoT devices, significantly reducing the risks of unauthorized access and data breaches [28]. Moreover, BC-based systems are inherently more resilient, providing real-time anomaly detection and tamper-evident audit trails. Integrating BC into the healthcare infrastructure is thus a strategic approach to building resilience against evolving digital threats and ensuring the long-term trustworthiness of health information systems. This integration not only enhances the security of medical information collected by IoT devices and protects patients' privacy but also provides an immutable and reliable shared ledger to maintain data availability, ensure data integrity, and build trust.

The concept of "interoperability" is often cited. IBM [29] note that "in the health field, interoperability refers to the timely and secure access, integration, and use of electronic health data to optimize health outcomes for individuals and populations" (para. 1) and often implies the ability of different information technology systems and software applications to communicate and exchange data and securely use the information that has been exchanged. Gordon and Catalini [30] note that "health data can live in multiple systems and sharing data requires numerous points of collaboration between entities," and that "as interoperability becomes more patient-centric, there is an opportunity to leverage BC technology to facilitate this exchange and give patients greater control over their data." Thus, "at a high level, BC technology can be thought of as a platform for digital exchange, where the platform functions without a traditional intermediary" (p. 227). Choquet [31] notes that "whilst patient-driving interoperability is an interesting style [trend] in healthcare," it is unclear "whether or not BC can facilitate the transition from institution-centric to patient-centric records sharing" (p.1).

In this context, decentralized data authentication and authorization are critical components of cryptography mechanisms which provide secure communication. Authentication is a process of validating user identities, whereas authorization determines what operations and functions users are permitted to perform once they are successfully authenticated [27]. Using public and private cryptographic keys with BC technology provides additional security within IoT-based healthcare systems to authenticate users and verify and validate data. Smart contracts on the BC can then manage key registration, verification, and revocation, ensuring that only authorized people can access or modify health data. This implies that each user has a decentralized identity based on asymmetric encryption which possesses a unique cryptographic key pair to control access to their medical data. Furthermore, each IoT device can be assigned unique cryptographic keys, enabling the generation of digital signatures for data authentication and validation [28]. All encrypted and hashed records are stored on BC permanently as blocks which are governed by a peer-to-peer network jointly adhering to a consensus protocol, dictating the rules for internodes to communicate and validate the newly generated blocks.

In healthcare, IoT devices generate massive, continuous streams of clinical data that must be stored and acted upon securely and in real time. Centralized databases struggle to provide global availability, robust access control, tamper-evident auditability, and strong privacy for electronic health records (EHRs). Recent studies indicate that disruptions and data breaches in healthcare have been linked to delays in care and increased in-hospital mortality during major cyber incidents [4, 32]. Having a decentralized distributed hashed ledger with privacy-preserving cryptography, and smart contracts to automate access control, can strengthen the security and efficiency of the healthcare system. As Daah et al. [33] note, "BC is one such technology that can provide a secure and transparent platform for information sharing in a ZT context" (p.2). In similar vein, Choquet [31] observes "the ability of different healthcare systems and providers to seamlessly exchange and use patient data are essential for efficient, high-quality care delivery. However, achieving interoperability has proven to be complex, hindered by disparate systems, data silos, and privacy concerns. BC technology has emerged as a promising solution to address these challenges, offering a decentralized and secure framework for data management" (p. 1). BC's immutability ensures that once data are recorded, they cannot be altered without consensus from the majority of the network, thereby safeguarding data integrity [34]. Core to the technology is its decentralization. Decentralized networks utilize spare computing power and storage resources from participants, reducing the need for centralized data centers and promoting resource sharing and energy efficiency.

By replacing the traditional model of a centralized system with a decentralized, distributed ledger, BC enables a trust framework that enhances data availability and transparency in healthcare systems, and data can thus be shared among hospitals, laboratories, pharmacies, and other stakeholders securely. This distributed decentralized data management architecture mitigates the risks associated with single points of failure, unauthorized access and enhances the integrity, auditability, and immutability of medical records. Access rights can be defined and enforced automatically when specified conditions are met, ensuring that only authenticated and authorized people and entities can read or write specific data segments. For example, a smart contract may allow temporary, patient-authorized access to EHRs, enforcing predefined temporal and data-scope constraints and revoking permissions upon expiry.

Liu et al. [35] observe that "given the increasing emphasis on ZT networked environments (e.g., all network traffic is considered untrustworthy, irrespective of the source), a number of security

solutions, including those based on BC, have been designed to operate in such settings." For example, in their study of secure medical image sharing systems, Sultana et al. [36] put forward a model that integrated ZT principles with BC. BC was used "to keep an audit trail of medical/health data transmissions for future examination," while "ZT principles were employed in keeping medical data safe during transmission and enhancing security on the user's side" (p.8). However, in noting the limitations of the study, the authors pointed out "in the future, the plan involves implementation of the total framework and its deployment on the Ethereum BC to test out its scalability and efficiency in the real world. The plan also includes incorporation of all the proposed security layers, testing and analyzing their effectiveness quantitatively by deploying it in an actual industry" (p.8).

The rise of wearable technology and IoT devices introduced significant challenges related to data security and device integrity. A particular concern lies in the embedded firmware within many medical IoT devices, which require regular updates and patches to defend against malware infections and zero-day exploits. Although existing research has addressed some of these security concerns, additional measures are needed to enhance the resilience of IoT-enabled healthcare infrastructures. In this context, Akkaoui [4] proposed a BC-based authentication scheme to ensure the authenticity of IoT devices and prevent the deployment of counterfeit hardware. This decentralized solution leverages smart contracts to manage and verify firmware updates in a secure and tamper-resistant manner. By distributing the authentication and verification processes across a BC network, the approach mitigates reliance on a single centralized authority, thereby enhancing system robustness and fault tolerance.

In summary, BC can be used in conjunction with IoT devices in healthcare to deliver major benefits in terms of security, data integrity, and privacy. Digital signatures authenticate users and IoT devices, while link blocks using encryption algorithms and hash functions ensure secure transactions by providing an immutable ledger, forming the backbone of security, confidentiality, and integrity of data collected from IoT within the decentralized BC system [37]. As regards data ownership, BC technology can be the catalyst for a paradigm shift in the management of healthcare information by enabling patients to retain verifiable ownership of their medical data. Unlike centralized systems, where records are stored and controlled by healthcare providers or third parties that are vulnerable to data misuse or data breach, BC allows patients to claim ownership of their healthcare data. Through cryptographic keys and smart contracts, individuals can grant, monitor, and revoke access to their records, ensuring that data usage aligns with their consent preferences [38].

3.3 | FL in Healthcare

FL is a decentralized machine learning approach that enables multiple users (such as hospitals, wearable devices, or mobile health applications) to collaboratively train a shared global model without sharing their raw data. This is particularly critical in healthcare settings, where data privacy and compliance with regulations such as HIPAA and GDPR are of paramount importance. In a typical FL setup, a central aggregator (often cloud-based) initializes a global model and distributes it to participating users. Each user trains the model locally on its private health data, such as electrocardiogram (ECG) readings, medical images, or symptom logs, using on-device computation. After local training, the users send only the model updates (e.g., gradients or weights) back to the server. The server then aggregates these updates, commonly using techniques such as federated averaging, to form an improved global model. This cycle is repeated over several rounds until the model converges [39]. This process is ideal for health IoT environments, where data is generated continuously and distributed across various devices, institutions, or geographic locations. FL enables the collective intelligence of diverse medical datasets to be harnessed while keeping patient data localized, thus reducing the risk of data breaches due to centralization [40].

However, despite its decentralized nature, FL is not immune to privacy and security risks. In the context of health IoT, these risks become even more critical because the data involved, such as biometric readings, diagnostic images, and treatment histories, are highly sensitive and can be life-impacting. Several privacy issues arise in such settings. Li et al. [39] observe that although raw data are never directly shared in FL, the gradients or weights transmitted to the central aggregator can still leak information about the underlying local data. Techniques such as membership inference or gradient inversion can reconstruct sensitive patient data from shared updates. The authors also raise the issue of statistical heterogeneity–health data collected from different hospitals, devices, or patient populations is often non-independent and identically distributed (non-IID). This not only complicates the learning process but also increases the susceptibility of updates to being deanonymized or inferred, especially when data from outlier patients are used. Ali et al. [40] point out that malicious users may engage in model poisoning by sending manipulated updates, aiming to skew the model's performance. This could have disastrous effects on healthcare applications, where misdiagnoses or incorrect predictions could endanger lives. Among the most serious threats are zero-day attacks which leverage previously unknown vulnerabilities in the FL system. Examples of zero-day threats include exploiting insecure model aggregation functions to inject backdoors into the global model, manipulating update schedules to introduce targeted model drift, and leveraging outdated firmware in edge IoT devices to gain unauthorized access to local model updates or data. These are particularly dangerous in health IoT because of the continuous data exchange, reliance on real-time feedback, and high stakes decision-making involved.

The complexity of FL systems and the diversity of IoT hardware mean that the attack surface is large, and traditional defenses such as firewalls or antivirus software are inadequate in addressing real-time, previously unseen threats [40, 41]. Nevertheless, as pointed out by Kairouz et al. [42], BC facilitates consensus-driven update validation whereby, instead of relying on a central aggregator, BC enables distributed consensus. This can be used to validate each round of model updates by having multiple validators (e.g., hospital nodes) verify that the update complies with privacy, statistical, and cryptographic constraints. This reduces the chance of a successful zero-day attack slipping through the system unnoticed [41]. In addition, BC can complement privacy-preserving mechanisms such as secure multi-party computation (SMC) or differential privacy (DP). By recording only anonymized cryptographic proofs instead of full

model updates, BC can help enforce distributed DP while keeping update provenance verifiable [42, 43].

3.4 | Relevant Frameworks

A number of frameworks combining ZT, BC, and FL in the context of healthcare and other sectors have surfaced in recent years, and these provide a benchmark against which the framework put forward here can be assessed. A comparative summary across key dimensions is provided in Table 2, identifying gaps in prior work that the proposed framework fills. These are discussed below.

Pokhrel et al. [44] proposed a zero-trust architecture (ZTA) for remote working and collaboration within IoT networks, with the overall aim of "ensuring robust security in a decentralized ZTA framework without relying on vulnerable central servers" (p. 8). Their framework couples BC-based FL with anomaly detection: It "includes a robust aggregation mechanism designed to counteract malicious updates from compromised clients, enhancing the security of the global learning process" (p. 7), while an unsupervised, incremental anomaly-detection component is used to identify previously unseen, zero-day behaviors. The authors concluded that their algorithm demonstrated "superior performance... compared to traditional FL approaches, particularly in scenarios involving device failures and poisoning attacks" (p. 12). This work is the closest in spirit to the framework presented here, as both integrate BC-based FL with anomaly detection under a ZT paradigm. However, the research of Pokhrel et al. [44] was not applied to the healthcare sector and was not tested against the full range of adversarial scenarios addressed in this study—most notably Sybil attacks—and does not provide the immutable auditability or the history-aware threat detection mechanism developed here, relying instead on clustering-based detection of novel anomalies.

Salim et al. [45], recognizing the exponential growth in "smart cities, factories, and marine industries" (p. 1), proposed a ZT, BC-enabled framework for scalable and secure IoT networks. The framework is grounded in AI-based optimization and employs "dynamic trusted gateways for continuous device authentication and integrates a deep reinforcement learning (DRL)–driven sharding mechanism to enhance BC scalability" (p. 1). The authors note that "despite advances in IoT security, existing frameworks struggle to address the dynamic nature of modern threats" and that "this exposes systems to insider threats, where compromised devices bypass security checks, resulting in data breaches and operational disruption" (p. 1). Their approach combines hierarchical sharding, DRL for adaptive trust assessment, and decentralized risk-aware access control. Evaluated in a simulated IoT environment, the results confirm that the framework can efficiently handle an increasing number of devices without significant performance degradation. The authors concluded that their proposal "addresses the limitations of static security policies, centralized learning, and high-overhead cryptographic methods, enabling dynamic authentication, optimized resource allocation, and continuous security enforcement" (p.12), while acknowledging that certain vulnerabilities remain, including targeted adversarial attacks on the DRL agents. However, the emphasis in the study by Salim et al. [45] is on BC scalability and device trust rather than federated model security; it does not employ FL for collaborative model training; is not tested against the poisoning, Byzantine, and Sybil scenarios examined here; and does not address immutable auditability.

Focusing more directly on the security benefits of BC, Bernabé Murcia et al. [46] designed and implemented "a novel solution to provide decentralized authentication/authorization across the continuum on distributed architectures, covering different objectives" (p. 14), providing "decentralized enrollment agreement, node enrollment and authorization during resource acquisition" (p. 14). Key aspects such as decentralization, interoperability, trust management, and privacy-enhancing capabilities were assessed, with distributed ledger technologies used to establish a decentralized identity ecosystem. The solution prioritizes interoperability, enabling nodes to seamlessly access and share their identities across different domains and environments, and incorporates privacy-preserving techniques that allow individuals to selectively disclose identity attributes while safeguarding sensitive information. The authors concluded that "including a decentralized authorization such as the one presented in this work does not represent a critical or appreciable overhead to the frameworks, while providing interesting security and privacy features" (p. 14). Although not specifically applied to the healthcare sector, the study offers a useful framework illustrating the value and feasibility of deploying decentralized authentication and authorization based on BC technology. This focus of this study is distinct from the research reported in this article: It addresses decentralized identity management rather than FL or model-level adversarial robustness, conducts no attack simulation, and does not incorporate a history-aware threat detection mechanism.

Several authors [47, 51] identify postquantum cryptographic algorithms as a key area for future research to safeguard BC transactions and communications against quantum attacks. In this context, Aleisa [48] puts forward a framework that "integrates BC technology with ZTA and postquantum cryptography" (p. 18660) in his study to enable privacy-preserving authentication, access control, and secure communication. The proposed systems architecture is based on a unified quantum-resilient BC and a ZKP privacy authentication framework. Experimental results indicated the potential quantum attack risk outcomes revealed "important weaknesses that affect the security, privacy, and efficiency of IoT networks." Specifically, "conventional public key infrastructures (PKI), which are susceptible to quantum computer algorithms, indicate a high level of risk," and that "developing postquantum cryptography alongside ZKP provides a solution to overcoming those hurdles" (p.18674). This framework integrates BC, ZTA, and postquantum cryptography for robust, privacy-preserving authentication and access control in IoT environments while utilizing quantum-resistant ZKP verification. However, it does not encompass FL, history-aware Byzantine detection benchmarked against Krum and Trimmed Mean, or smart contract-enforced role-separated access control under real attack conditions.

Specifically as regards healthcare, Zhu et al. [49] studied an automatic medical diagnosis service based on deep learning which had been introduced in e-healthcare. They noted that because of privacy regulations, "insufficient data sharing among medical centers has led to many severe challenges for automated medical diagnostic services, including diagnostic accuracy" (p. 1). To address these issues, a new privacy-preserving Byzantine-

TABLE 2 | Comparative summary of recent literature across key dimensions.

Key dimension/source	Technologies used	Attack types	Organization focus	Metrics reported	Experiment environment	Comment
Pokhrel et al. [44] A new framework, based on a robust decentralized zero-trust architecture (ZTA) and underpinned by a blockchain-enabled federated learning (BFL) approach.	The framework combines blockchain-based FL with a zero trust architecture (ZTA) and an unsupervised clustering module for anomaly detection. It incorporates a robust aggregation mechanism to resist Byzantine model updates, an incremental anomaly detection component to handle novel zero-day threats, and an adaptive algorithm to adjust trust thresholds based on user context. Trust computation is decentralized across FL nodes.	The framework targets malicious model updates submitted by compromised FL clients, analogous to Byzantine attacks. It additionally addresses zero-day attacks, defined as novel anomalous behaviors not seen during training. Insider threats within decentralized IoT networks are also considered, where trusted participants may act adversarially.	The framework targets general-purpose IoT networks and decentralized remote-work and collaboration environments in the context of next-generation communications. It addresses the need for robust ZTA in scenarios where FL nodes operate across heterogeneous, untrusted network segments.	Detection accuracy and communication overhead.	Performance Evaluation was done using the MNIST classification dataset. There are 100 devices and 2 miners that cooperate to perform the trust computation.	Closest in spirit to the research presented in this article, the key differentiator is the use of unsupervised incremental clustering for zero-day attack detection in this research [44]. Here, also, general IoT is targeted and there are less experimental details and poisoning and Sybil attacks are not covered.
Salim et al. [45] Put forward a zero-trust blockchain-enabled framework for scalable and secure IoT networks.	The framework employs a hierarchical sharding blockchain architecture enhanced by deep reinforcement learning (DRL) to dynamically optimize shard allocation and cross-shard transactions. Zero-trust authentication is integrated with dynamic trusted gateways and a multidimensional trust evaluation engine using a sliding-window model. Smart contracts manage shard membership and node trust. The system was deployed on Raspberry Pi 4 devices, reflecting edge IoT hardware constraints.	The framework addresses insider attacks involving pre-authenticated devices that subsequently behave maliciously, as well as adaptive collusion attacks where multiple nodes coordinate to subvert the sharding mechanism. Dishonest node behavior within shards is detected through the multidimensional trust evaluation engine.	Not sector specific. Applies to IoT ecosystems.	Performance metrics: the framework achieves 2500 transactions per second (TPS) at 700 nodes, a 14% increase over competing frameworks. Latency was significantly lower than alternatives. Cross-shard transactions are reduced, and node trust evaluation is improved by 41%, maintaining 85% throughput even with 20% malicious nodes.	The framework is evaluated on Raspberry Pi 4 hardware acting as IoT-class edge devices, with a simulated IoT network scaled up to 700 nodes. The blockchain employs DRL-driven dynamic sharding, tested under varying network loads and with up to 20% of nodes behaving maliciously.	The study employs ZT with blockchain for IoT scalability but lacks FL and adversarial simulation. It does not address healthcare or poisoning/Byzantine/ Sybil explicitly.

(Continues)

TABLE 2 | (Continued)

Key dimension/source	Technologies used	Attack types	Organization focus	Metrics reported	Experiment environment	Comment
Bernabé Murcia et al. [46] The article focuses on Decentralized Identity Management (DIM) when diverse computing environments and devices are in evidence. Interoperability is prioritized, enabling nodes to seamlessly access and share their identities across different computing environments.	The solution is built on Self-Sovereign Identity (SSI) principles using Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) secured by Zero-Knowledge Proofs (ZKPs). It leverages Distributed Ledger Technology (DLT) for credential anchoring and the FLUIDOS/Liqo orchestration framework for resource sharing across multidomain computing continuum environments. Decentralized Identity Management (DIM) replaces traditional certificate authorities for cross-domain identity verification.	The work addresses unauthorized access and identity spoofing across multidomain environments, where entities from one domain may attempt to impersonate or exceed their authorization in another. Trust exploitation arising from implicit cross-domain federation assumptions is mitigated through DIDs and VCs.	The focus is on enabling secure resource sharing and identity federation across heterogeneous administrative domains—such as when an organization wishes to offload workloads to a partner's edge nodes, without relying on a central identity authority. It does not target a specific industry vertical such as healthcare.	The framework reports latency for identity operations. No throughput, gas cost, or security attack simulation metrics are reported, as the evaluation focuses on demonstrating the operational feasibility and performance overhead of the DIM solution relative to centralized baselines.	The framework is deployed within the FLUIDOS European research project testbed, spanning a multidomain computing continuum environment with cloud and edge components managed via the Liqo resource-sharing framework.	The study addresses decentralized identity management and privacy-preserving authentication across multidomain environments rather than FL model security or adversarial robustness. It could complement the framework presented in this research as an identity layer for authenticating FL participants across organizational domains, rather than serving as a direct alternative. No explicit adversarial attack simulation is conducted; the security guarantees are derived from the cryptographic properties of SSI and ZKPs rather than empirical attack-defense experiments.

(Continues)

TABLE 2 | (Continued)

Key dimension/source	Technologies used	Attack types	Organization focus	Metrics reported	Experiment environment	Comment
Vusumuzi and Godwin [47] An AI-driven Zero Trust framework integrated with blockchain Technologies. Design Science Research approach, combining architectural modeling with simulation-based evaluation.	The framework integrates machine learning for anomaly detection, adaptive authentication, and real-time policy enforcement, alongside a blockchain layer for immutable logging and distributed trust management.	The model adapts dynamically to evolving threats while maintaining verifiable audit trails through blockchain.	For secure and privacy-preserving healthcare data sharing.	Metrics included performance latency, throughput (TPS), and resource utilization. Security intrusion, detection rate, false positives, resilience against replay, and insider attacks. Compliance alignment with GDPR/HIPAA indicators.	Simulated mHealth scenarios were used for performance and security evaluation. This involved wearable devices and electronic health records.	The article explores how AI can enhance the core principles of Zero Trust, continuous verification, least-privilege access, and adaptive risk assessment, while leveraging blockchain's immutability and decentralization to ensure auditability and compliance. However, it does not include decentralized federated model training, history-aware Byzantine client detection, or empirical evaluation under poisoning, Byzantine, and Sybil attack scenarios.
Aleisa [48] Introduces a "Unified Quantum-Resilient Blockchain-Zero-Knowledge Proofs Privacy Authentication Framework (QBC-ZKPAF)" to provide IoT environments with greater security.	Integrates blockchain technology with zero trust architecture (ZTA) and postquantum cryptography. Technologies used include a hybrid Reinforcement-Lattice Blockchain KeyGen for quantum-resilient key generation, Deep Q-Network Multifactor Secure Key (DQN-MFSK) for dynamic selection of keys, and Zero-Knowledge Proof for privacy-preserving signatures.	Focuses on security and privacy in IoT environments, where emerging quantum threats have arrived. Quantum resilience measures a system's ability to resist quantum computing attacks, particularly on cryptographic protocols.	For IoT networks as well as privacy-sensitive blockchain applications. Potential for sectors such as healthcare and smart cities, to address unique device and network constraints but is not specific to the healthcare sector.	Experimental results demonstrate 98% privacy preservation, 700 transactions per second throughput, 0.7 J energy consumption, 0.98 quantum resilience, and 96% access control.	The QBC-ZKPAF was implemented in Python and validated using the Edge-IIoTset Cyber Security Dataset of IoT and IIoT from the Kaggle Database.	The proposed framework integrates blockchain, ZTA, and postquantum cryptography to ensure robust, privacy-preserving authentication, and access control in IoT environments. However, it does not include FL, history-aware Byzantine detection benchmarked against Krum and Trimmed Mean, or smart contract-enforced role-separated access control under real attack conditions.

(Continues)

TABLE 2 | (Continued)

Key dimension/source	Technologies used	Attack types	Organization focus	Metrics reported	Experiment environment	Comment
Zhu et al. [49] The article puts forward a privacy-preserving Byzantine-resilient swarm learning (PBSL) model.	The proposal centers on swarm learning, a form of blockchain-based federated learning. Threshold fully homomorphic encryption (TFHE) is adopted, to protect data privacy and provide secure aggregation. Deep learning is united with blockchain-based smart contract platforms. Smart contracts are developed in the Solidity programming language.	The following Byzantine attacks were considered: Label Flipping Attack, Backdoor Attack, Arbitrary Model Attack, and Untargeted Model Poisoning. Malicious gradient uploaded by malicious medical centers is assessed using cosine similarity.	Healthcare: automated medical diagnostic services.	Test accuracy and test error rate are used as evaluation measures on different training datasets. For malicious Medical Centers, the results were verified by using the no-attack FedSGD scheme as the baseline, and the Krum scheme, Bulyan, Trim-mean as the control.	The model is implemented on a private Ethereum blockchain setup with the MNIST and FashionMNIST medical datasets. Smart contract performs a secure global gradient aggregation algorithm instead of the centralized server. CKKS Scheme is deployed using the TenSEAL library.	The focus on swarm learning differentiates the article from others reviewed here. The PBSL is able to defend against a variety of known security attacks, being resistant to poisoning attacks while protecting data privacy. However, the PBSL does not encompass history-aware threat detection nor does it assess Byzantine and Sybil attack scenarios.
Chinnasamy et al. [50] The article sets out an access control system, using smart contracts to enhance security while sharing electronic health records among patients and healthcare providers.	The research develops a scalable Electronic Health Records (EHR)–sharing mechanism, using blockchain technology and distributed storage via interplanetary file systems (IPFSs). The study analyzes access control over data uploads and distribution, and proposes a mobile cloud management technique based on smart contracts and blockchains. Contracts were created and implemented using a Solidity scripting language. Blockchain communication was executed by web3.js. contracts.	Security analysis was done using different attack situations.	The focus is on the secure sharing of health data between patients and healthcare providers in e-health applications using cloud-based Internet of Things (IoT).	Security is assessed using different evaluation criteria to illustrate the effectiveness of the proposed system compared with existing e-health methods. A compatibility test using a cloud-based IoT configuration is used to analyze performance. Effectiveness was evaluated in various real-world scenarios, including IoTs, and vehicular network virtualization.	An Ethereum blockchain framework is used to construct an e-health system. This maintains data blocks and executes smart contracts, allowing developers to create various blockchain-based solutions such as e-healthcare. The study used a platform to exchange EHRs on a mobile cloud. Two AmazonWeb Computing node VMs were used as miners, and two Ubuntu 16.04 LTS VMs acted as administrators and EHR managers.	The research provides a potential solution for secure data sharing in mobility computing while protecting personal health information from potential risks. However, the adoption of a hybrid public blockchain exposes patient identifiers and access patterns to any network participant, raising privacy concerns. Strict identity verification, history-aware threat detection are absent, and a centralized admin controls all smart contract rules, introducing a single point of trust failure. Poisoning, Byzantine, and Sybil attack scenarios are not specified.

resilient swarm learning (PBSL) was proposed that is resistant to poisoning attacks while protecting data privacy. The authors noted "specifically, after the BC-based point-to point network is established, each medical center calculates its local gradients according to the local training data" (p. 2). Their experimental methodology was based on a private Ethereum BC setup with real medical datasets, and a number of Byzantine attacks were considered (label flipping attack, backdoor attack, arbitrary model attack, and untargeted model poisoning). The research is of value in that it highlights the current shortcomings of swarm learning and puts forward a new privacy-BC Byzantine-resilient swarm learning for e-heath. In addition, the focus on swarm learning differentiates the article from others reviewed here. The PBSL is able to defend against a variety of known security attacks, being resistant to poisoning attacks while protecting data privacy. However, the PBSL does not encompass history-aware threat detection nor does it assess Sybil attack scenarios.

Chinnasamy et al. [50] put forward an access control system that uses smart contracts to achieve greater security while sharing EHRs among various patients and healthcare providers. It offered "an active resolution for secure data sharing in mobility computing while protecting personal health information from potential risks," recognizing "increases in the practicality of lightweight access control architecture, low network expectancy, and significant levels of security and data concealment" (p.1). The authors implemented an Ethereum BC on the AWS cloud and evaluated the system's performance against two criteria: the average time required to evaluate several access user requests in the cloud and the time required to access users' data. Security aspects were evaluated using different attack situations. The authors concluded that compared with traditional approaches, "the operational results show that our architecture could require doctors/users to transmit medical information in an adequate and accurate fashion through mobile cloud settings" and that "our access control system can efficiently detect and block unlawful access to e-health systems, ensuring patient confidentiality and computer security" (p.17). The study provides a potential solution for secure data sharing in mobility computing while protecting personal health information from potential risks. However, several significant security aspects are not covered. The adoption of a hybrid public BC exposes patient identifiers and access patterns to any network participant, raising privacy concerns. Strict identity verification, history-aware threat detection are absent, and a centralized admin controls all smart contract rules, introducing a single point of trust failure. Poisoning, Byzantine, and Sybil attack scenarios are not specified.

Vusumuzi and Godwin [47] proposed an AI-driven ZT framework integrated with BC technologies for secure and privacy-preserving healthcare data sharing. The objective of the study was "to explore how AI can enhance the core principles of ZT, continuous verification, least-privilege access, and adaptive risk assessment, while leveraging BC's immutability and decentralization to ensure auditability and compliance" (p. 1). The results demonstrated that AI can "enhance the core principles of ZT, continuous verification, least-privilege access, and adaptive risk assessment, while leveraging BC's immutability and decentralization to ensure auditability and compliance" (p.1). The proposed framework achieved higher intrusion detection accuracy (95.2%) and superior compliance alignment compared to existing BC-only and AI-only healthcare models and adapted dynamically to evolving threats while maintaining verifiable audit trails through BC. Benchmarking was done against intrusion detection datasets and healthcare-specific synthetic logs were used to test the anomaly detection capabilities of the AI layer. Intrusion detection accuracy was measured at 95.2%. Compared with existing BC-only and AI-only healthcare models, there was superior compliance alignment (HIPAA and GDPR). The authors concluded that "compared to other hybrid frameworks, this study uniquely combines continuous verification, adaptive AI-driven authentication, and BC auditability in a unified architecture, achieving the highest detection accuracy and regulatory alignment among recent approaches" (p. 9). This study represents a significant advance in this field of research, exploring how AI can enhance the core principles of ZT, continuous verification, least-privilege access, and adaptive risk assessment, while leveraging BC's immutability and decentralization. However, it does not encompass decentralized model training and history-aware threat detection and does not investigate major security threats such as Sybil and Byzantine attacks.

It is thus evident that the effectiveness of BC technology in facilitating secure and transparent health information data exchange has been demonstrated via a range of framework-based experiments. At the same time, dynamic threat detection, notably against Sybil and Byzantine attacks, and data poisoning, has remained unproven in several such frameworks, as indicated in Table 2.

3.5 | Toward a PCF (RQ1)

The review of the relevant literature identified the key challenge that this article addresses: to combine BC technology with FL and ZT to allow users in the healthcare sector to perform authentic and traceable data transactions on a distributed decentralized ledger which improves security, enhances privacy-preservation, and builds a trusted platform. This analysis from the literature is summarized in the PCF (Figure 2) that addresses RQ1 and provides the context for the controlled experiment in Phase 2 of the research.

While BC-based FL principles offer a promising privacy-aware learning paradigm for health IoT environments, it is vulnerable to sophisticated threats, particularly zero-day, Sybil, Byzantine, and data poisoning attacks that exploit systemic and architectural flaws [43–45]. These vulnerabilities are amplified by the heterogeneity, real-time requirements, and sensitivity of health data. BC, with its tamper-resistant, decentralized, and verifiable framework, provides robust countermeasures that complement FL's strengths. By integrating smart contracts, distributed consensus, and immutable logging, BC can potentially fortify FL systems against known and unknown threats alike, making it a critical component in the future for secure, decentralized healthcare AI.

4 | Framework Architecture and Design

The core objective of the proposed framework is to enhance the security and privacy of healthcare systems through a three-layer architecture comprising BC, FL, and ZT. BC, as a decentralized and distributed ledger, ensures the immutability of recorded data, thereby preventing data manipulation and minimizing

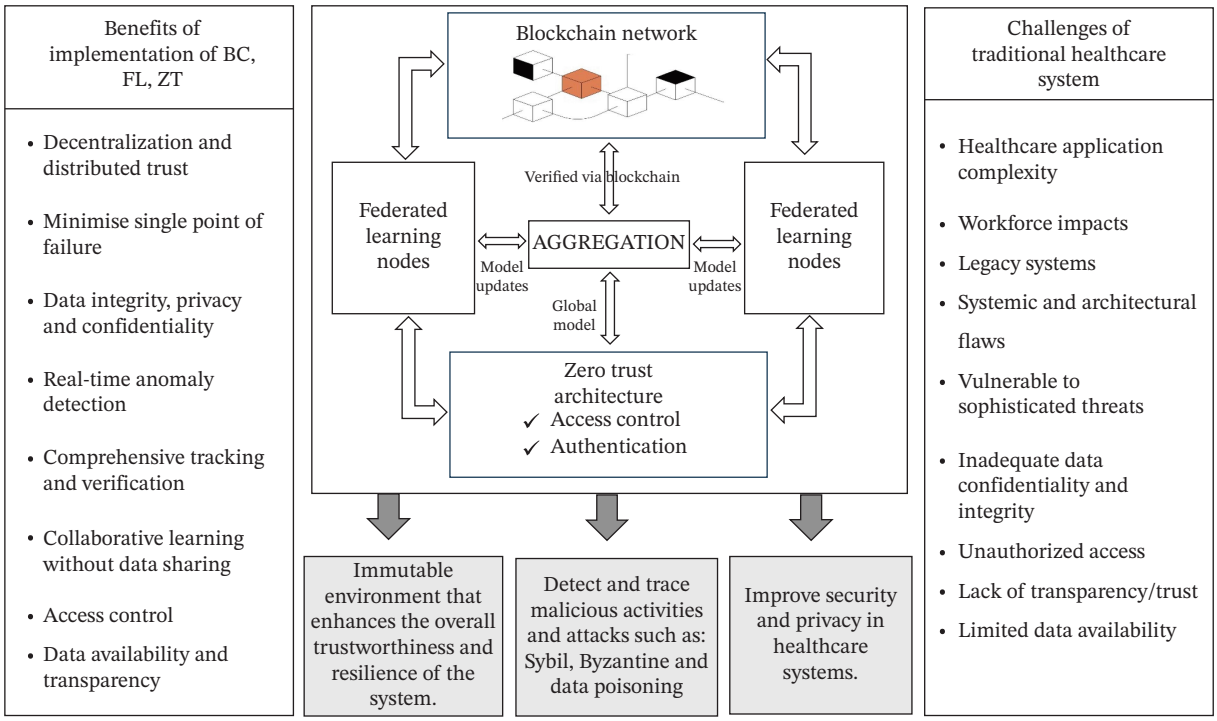

FIGURE 2 | Provisional conceptual framework.

single points of failure. Additionally, this layer supports the enforcement of access control policies using smart contracts that embed predefined rules and conditions.

The integration of FL layer facilitates accurate anomaly detection and risk prediction, whereas the implementation of the ZT layer ensures continuous verification and enforces secure access control mechanisms through role-based access control (RBAC).

4.1 | Smart Contracts for Authorization and Authentication

The developed decentralized framework includes three smart contracts. Each contract is designed with explicit access modifiers and signature verifications, ensuring that only authenticated and role-authorized entities can execute specific operations.

The three contracts are as follows:

i. PermissionHandle.sol: It is responsible for managing user access permissions. Specifically, it maintains the status of user access requests, which may be pending, accepted, or denied.

ii. UserModelUpload.sol: It acts as a verification layer that ensures users possess the required access permissions prior to engaging in system interactions. Users interface with this contract to upload FL model parameters, which are subsequently validated by admin ("admins" are users but the difference is they validate and do access control for the framework; they do not train the data like a standard user does). Additionally, the contract manages the storage and retrieval of IPFS hashes associated with each model upload. Users can view the status of both their current and previous model parameter submissions. For each transaction or function call, the contract records the associated digital signature and message hash. Furthermore, it provides functionality to interact with the admin contract for downloading the global model, thereby enabling distributed participation.

iii. Admin.sol: It contains predefined admins' wallet addresses and their corresponding roles. The contract enforces RBAC with a ZT security model, requiring that each function invocation be accompanied by a valid hashed message signature, irrespective of admin status. Key functions are set out in Table 3.

Figure 3 indicates the four main steps that are undertaken for users to upload their locally trained model to the BC and for admins to retrieve the model and update the result back to the chain. Step 1 involves the process of users using their own authorized data that is stored on their local device and will begin to train the data based on a global model and output the locally trained model parameters, ready to be uploaded via the BC and stored on the decentralized IPFS. At Step 2, the users take the model parameters output from the training process and upload the file to IPFS which stores the model across users rather than a central server (2a). After this process, users can check their model verification status via the BC and view the history of their uploads, and if their models have been denied or approved for the corresponding round (2b). At Step 3, the admins will use the BC and IPFS in conjunction to check what round the FL aggregation is on, change the global model if required, and obtain users locally trained model parameter so aggregation can take place. The final Step 4 overlaps with Steps 1–3 but also has its own separate process. Admins will set permissions on who can access the solution that determines if users can upload their local model parameters to begin with. All permissioned users' local models are downloaded and aggregated by an admin, and the results are then placed on the BC. Finally, once aggregation has taken place, a new global model is uploaded, and the new FL round begins.

TABLE 3 | Functions of the Admin.sol smart contract.

Admin.sol functions	Description
Round management	Admins with the RoundSetter role are authorized to set the global model, initiate training rounds, and terminate rounds.
Permission control	Admins with the PermissionHandler role can decide to accept or deny user access.
Model aggregation	Admins with the validator role may download the global model, access individual user models, and perform aggregation.
User model validation	The status of each user's model submission is updated in the UserModelUpload contract. Admins with validator role are authorized to accept or reject the submitted local model parameters.
Role enforcement	The system verifies that only designated admins can execute privileged operations, ensuring compliance with segmented role definitions (zero trust element).

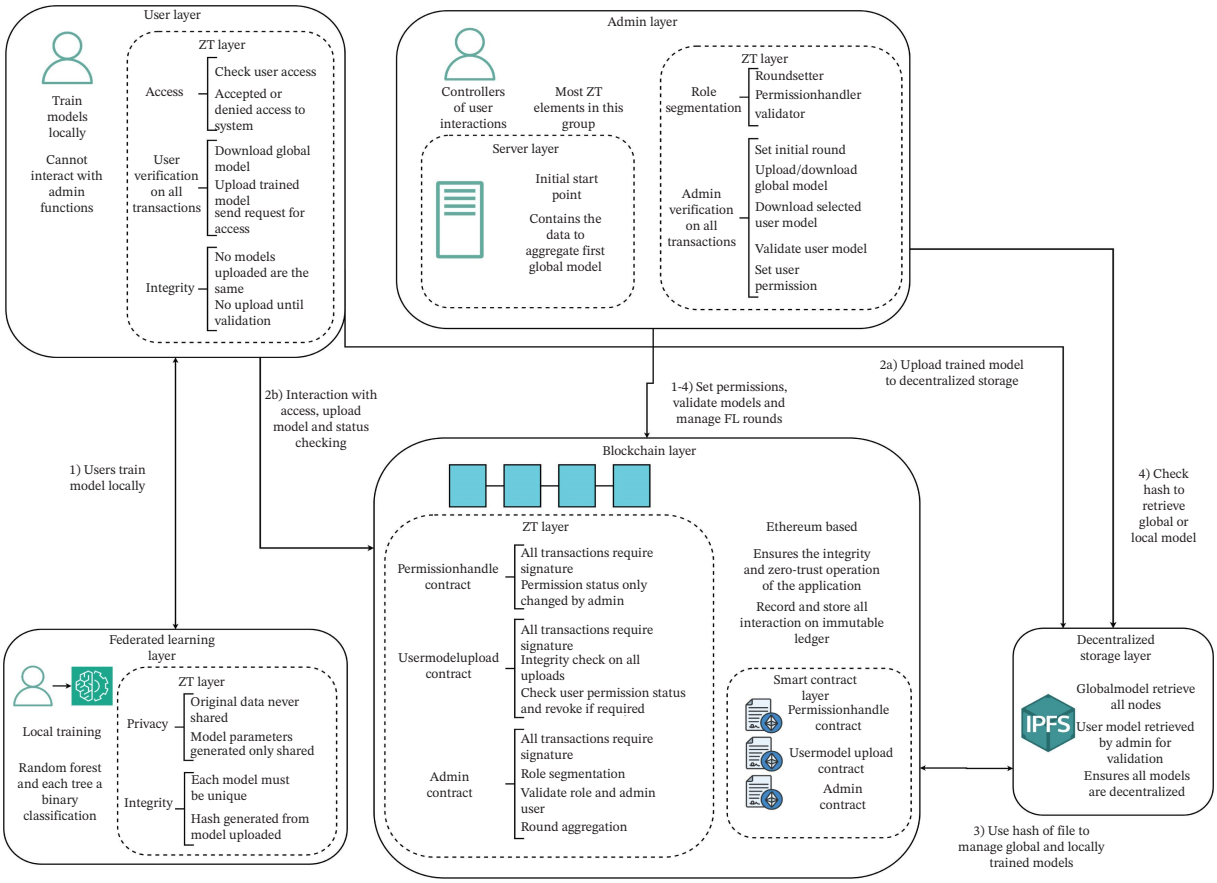

FIGURE 3 | Framework-system architecture.

4.2 | Operational Layers of the System Workflow

The system workflow revolves around four main operational layers: permission access management, FL round management, user model upload and validation, and digital authentication and ZT enforcement. Further detail regarding each of these layers is provided below.

4.2.1 | Permission Access Management

A user initiates access by submitting a signed digital request to the system's permission handler admin. The request includes identity credentials, public wallet address, and hashed metadata. The admin reviews the pending requests and decides whether to approve or deny each request but the RBAC in the smart contract ensures that the admin cannot modify permissions related to access for the model uploads section. This mechanism enforces ZT principles by maintaining least-privilege access control. All transactions are cryptographically signed and immutably recorded on-chain.

4.2.2 | FL Round Management

After user validation, the system performs local FL training to allow authorized users to upload their health model parameter

document, which is the result of the training done by the user on each end. The resulting model is uploaded to the BC, while the actual file is securely stored in IPFS. A hash of the model parameter is assigned to the user's wallet address for traceability. All uploads are digitally signed and verified using BC and ZT mechanisms, ensuring authentication of users and IoT devices, as well as verification of transactions for data integrity. The RoundSetter starts a new FL round by uploading the initial global model to its IPFS node, which generates a unique hash that represents the uploaded model and is stored on-chain. All authorized users retrieve the global model from IPFS using the model parameters hash stored in the smart contract on the BC. They update their local models based on the global model values and upload their updated local parameters to IPFS. An active round will be ended by updating the global model, storing its new hash on-chain, and incrementing the round number in the smart contract. The new global model automatically terminates the previous round. Only admins with the RoundSetter privilege may initiate or end rounds. Any attempt by unauthorized users to perform these actions is reverted by the smart contract, preserving system integrity and enforcing decentralized governance.

4.2.3 | User Model Upload and Validation

FL enables the training of AI models by averaging local updates aggregated from multiple health data users such as medical imaging systems and IoT devices, without the need for direct access to the local data. This approach preserves user privacy by ensuring that sensitive data never leaves the local device. For example, each IoT device independently trains a local FL model using its own dataset.

Authorized users can upload their locally trained model, and the hashes of these models are stored on the BC to ensure integrity and traceability. Users can only upload one pending model at a time, with duplicate model uploads being reverted. Admins with the validator role reviews and evaluates each uploaded local model. The admin compares the submitted model parameters with the global model to identify any significant discrepancies that could indicate malicious activity. Based on validation results, the admin either approves or rejects the submitted model. The validation decision is recorded on the BC, ensuring transparency and visibility for both admin and users.

All operations employ digital signature verification as the foundation of ZT security. Each transaction, whether uploading, downloading, or validating and every event such as "accept" or "reject" are associated with a signed message that includes a unique hash which is recorded on the decentralized ledger. The smart contract verifies the authenticity of the signature before executing any function.

Figure 4 depicts the sequence of operations that are undertaken between different entities for the BC, ZT, and FL architecture. The figure traces the flow of all actions and the data exchanged between each entity. There are seven entities that make up the sequences: admin, crypto wallet, Ethereum network, admin contract, PermissionHandler contract, UserModelUpload contract, and all users.

4.2.4 | Authentication and ZT Enforcement

The admin is the starting point for the management of the upload processes and uses the crypto wallet entity for the digital signatures and for conducting all transactions. The Ethereum network entity is where the BC resides and holds the three smart contract entities and all data that are passed through the chain. The admin contract sets the sequences for permission handling of regular users and validation of all uploaded models residing in the IPFS entity. The PermissionHandler contract will handle all users' permissions to access the solution, and their status is only changed once an admin grants or denies them access; and if accepted, users will interact with the UserModelUploads contract to upload their locally trained model via the BC to IPFS. Finally, the admin entity will look through all these and update the status to the relevant contract entities and broadcast these changes to all users and the BC.

4.3 | Security Workflow of Smart Contracts Operation

Evaluating the security and privacy of healthcare data involved three algorithms designed to authenticate and authorize all users, validate locally trained models, and control access for submitting these models. These assessments were designed to validate the integrity, confidentiality, and authenticity of the data flow within the decentralized framework.

Algorithm 1 serves as the main function embedded in all smart contracts to enforce the ZT principle. The process includes generating a digital signature and a hashed message to verify all users including admins and to authenticate their interactions within the system.

The inputs for this algorithm are the generated hashed message and the address of the user who signed the message for authentication. Step 1 obtains the signature of the signed message and reads if it is a 32-byte Ethereum standard signature. Step 2 extracts the r, s, and v parameters from the signature, which contain the authenticity of the signature, the message hash, and the original signer of the signature. Steps 3-4 concern the process of receiving the parameter and outputting them so they can be checked against the current user interacting with a smart contract function.

Algorithm 2 is designed for admins with the permission handler role, enabling them to verify users' permissions for uploading locally trained models and downloading global models.

The algorithm accepts 2 inputs that are checked in Algorithm 1. These are the construction of the digital signature to verify the user of the function, and the output is the status changing of a user to be accepted to interact with the solution or not. The "Require" statements enforce ZT validation by requiring that the signature is cryptographically valid and adheres to the verification requirements of Algorithm 1. Step 3 retrieves the existing permission record associated with the caller to be used in conjunction with Steps 4 to 10, where the user does not already exist in the system: a new record is created, the user's status is set to Pending, their address is stored, and they are added to the list of

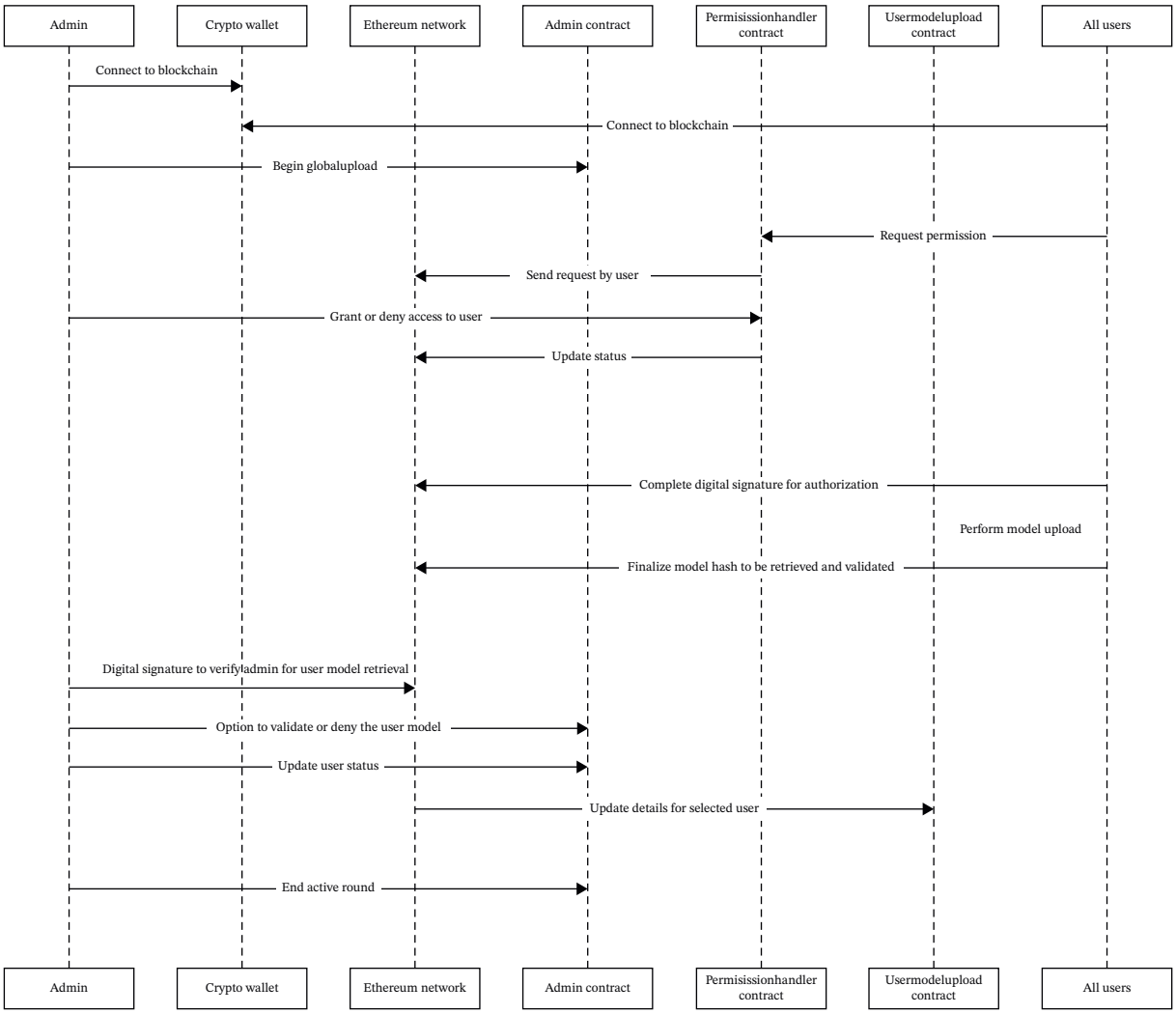

FIGURE 4 | Flow of actions and the data exchanged between entities in the proposed framework (integrating blockchain, FL, and ZT within healthcare systems).

ALGORITHM 1 | ZT verify signature applied to all smart contract functions.

Input: messageHash, signature, expectedSigner

Output: true if signature is valid and matches expectedSigner, otherwise false

1. Compute ethSignedMessageHash ⟵ Hash from "Ethereum Signed Message:\n32" and messageHash
2. (*r, s, v*) ⟵ extract elements from signature to verify in all contract functions signature using SplitSignature procedure
3. signer ⟵ result of RecoverAddress (ethSignedMessageHash, *v, r, s*)
4. return (signer = expectedSigner)

all users, followed by emitting the PermissionRequested event. Steps 11 to 13 handle where a user was previously denied; their status is reset to Pending and the event is re-emitted.

Algorithm 3 is designed for admins with the validator role to verify users' submitted local models after each FL round. If all predefined criteria are met, the transaction is approved and completed; otherwise, it is rejected according to the predefined smart contract rules.

The inputs identify the user, specify the training round, and indicate whether the update is to be accepted or denied. The output of the algorithm is the emission of a RequestValidated event, which is stored on the BC. Steps 3-4 check that the signature is valid based on the logic of Algorithm 1; the user must have an approved permission status setting in Algorithm 2, and the caller must be a known admin to the smart contract. Steps 5-6 retrieve all model uploads associated with the specified user. Steps 7–16 iterate these uploads, updating the status and recording timestamps.

ALGORITHM 2 | Permission handling of users wanting to upload the locally trained FL model.

Input: messageHash, signature

Output: Emit event PermissionRequested if applicable

1. Require: Signature is valid for messageHash
2. Require: Match requirement of Algorithm 1 to be accepted through ZT validation
3. user permission record associated with caller
4. **if** user does not exist **then**
5. Create new permission record for caller
6. user.exists ⟵ true
7. user.status ⟵ Pending
8. user.address ⟵ caller
9. Add caller to allUsers list
10. Emit PermissionRequested (caller)
11. **else if** user.status = Denied **then**
12. user.status ⟵ Pending
13. **Emit** PermissionRequested (caller)

ALGORITHM 3 | Validate user models to be accepted or denied.

Input: walletAddress, roundId, updatedStatus, messageHash, signature

Output: Emit event RequestValidated

1. Require: Signature is valid for messageHash
2. Require: Match requirement of Algorithm 1 to be accepted through ZT validation
3. Require: Match requirement of Algorithm 2 for permission value to be set as 1 in contract
4. Require: Caller is an admin with role Verifier
5. userUploads all uploads for walletAddress
6. Foreach upload in userUploads **do**
7. if upload.status = Pending **then**
8. upload.status ⟵ updatedStatus
9. upload.timestamp ⟵ current time
10. **for** j ⟵ 0 to updateCount − 1 **do**
11. record ⟵ updatesByRound [j]
12. **if** record.userWalletAddr = walletAddress and record.status = Pending **then**
13. record.status ⟵ updatedStatus
14. record.timestamp ⟵ current time
15. break
16. break
17. Emit RequestValidated(walletAddress, roundId, updatedStatus)

Algorithm 4 specifies the history-aware trust filter applied during Byzantine attack detection at the aggregation stage. The filter evaluates each client's current $F1$-score against both the peer median and client's own history, flagging and removing the anomalous participant before aggregation. Three predefined thresholds govern the detection criteria: $\tau\_c = 0.10$ controls the minimum gap between a client's current $F1$ and the peer median required to consider it a candidate for removal, $\tau\_h = 0.08$

governs the minimum deviation between a client's historical mean $F1$ and the historical median across clients, and $\tau\_d = 0.12$ captures a sharp drop either from the client's own previous round or from its historical mean. These values were selected based on the Breast Cancer Wisconsin dataset. Legitimate clients consistently produce $F1$ variations below 0.08 between rounds, while a 30% label-flip attack induces drops of 0.15–0.25. The trust history is updated after every round using all observed client $F1$-scores, including those of flagged clients, ensuring the filter improves in sensitivity as rounds progress.

The application of Algorithms 1–4 resulted in a ZT yet decentralized solution for FL validation and security. The connection of Algorithm 1 ensures that ZT principles are enforced as it is applied to all interactable functions in all 3 smart contracts presented earlier. Access control in Algorithm 2 ensures that only authenticated users are allowed to be accepted and necessary levels of privacy, as all identifiers in the signature are hashed before signing. Algorithm 3 primarily ensures integrity ensuring only the admin with the correct role assigned can validate on chain. Algorithm 4 adds a further layer of robustness by detecting and excluding Byzantine clients at the aggregation stage, leveraging temporal consistency in $F1$-score behavior to identify malicious participants without requiring access to the raw data.

5 | Results and Evaluation

This section evaluates the effectiveness of the proposed framework and analyzes the security and performance of the developed decentralized application. The experimental evaluation focuses on three main criteria: model performance, security and privacy, and cost analysis (See Table 3).

- Model performance was measured for different rounds of training and testing using classification accuracy, F1-score, precision, and recall.
- Security and privacy was measured by the framework's ability to detect and mitigate three major cyber threats: Sybil, Byzantine, and data poisoning and unauthorized access attempts. The metrics used for detection of the attacks include MAD rule and Mean–std rule metrics for data poisoning, history-aware filter on $F1$-score for Byzantine attacks, and digital signature with access control for Sybil attacks.
- Cost analysis was done in terms of system latency, gas cost, and throughput.

5.1 | FL Performance and Attack Detection (Poisoning and Byzantine)

In this paper, the $F1$-score is used as the primary evaluation metric for the FL model, as it provides a more meaningful assessment than accuracy alone in medical-style binary classification tasks, where class imbalance is common and misclassification can have significant consequences. While accuracy, precision, and recall are also reported, the $F1$-score better captures the balance between precision and recall, making it more suitable for this context. On the BC side, evaluation focuses on system performance, using latency, gas cost, and approximate throughput to assess the efficiency and scalability of recording

ALGORITHM 4 | History-aware trust filter for Byzantine client detection.

Input: clientMetrics (set of {client_id, $F1\_i$, n_samples} for all clients in round t), trustHistory (log of past $F1$ scores per client)

Output: approvedClients (subset of clientMetrics with anomalous client removed), updatedTrustHistory

medianF1 ⟵ median({$F1\_i$: client_i ∈ clientMetrics})

medianHistory ⟵ median({μ_i: client_i ∈ clientMetrics}), where μ_i = mean of all $F1$ scores for client_i in trustHistory (if no history, μ_i ⟵ $F1\_i$)

For each client_i in clientMetrics do

prev$F1_i$ ⟵ most recent $F1$ score for client_i in trustHistory (if none, prev$F1_i$ ⟵ $F1\_i$)

currentGap_i ⟵ median$F1$ − $F1\_i$

historyGap_i ⟵ medianHistory − μ_i

dropFromPrev_i ⟵ prev$F1_i$ − $F1\_i$

dropFromHistory_i ⟵ μ_i − $F1\_i$

score_i ⟵ currentGap_i + max(historyGap_i, 0) + max(dropFromPrev_i, 0) + max(dropFromHistory_i, 0)

candidates ⟵ {client_i: currentGap_i > τ_c AND (historyGap_i > τ_h OR dropFromPrev_i > τ_d OR dropFromHistory_i > τ_d)}

if |candidates| = 0 then

approvedClients ⟵ clientMetrics

else

flagged ⟵ client_i ∈ candidates with highest score_i

approvedClients ⟵ clientMetrics − {flagged}

updatedTrustHistory ⟵ trustHistory ∪ {(client_i, $F1\_i$, t): client_i ∈ clientMetrics}//record ALL clients including flagged

return approvedClients, updatedTrustHistory

and validating model updates. A summary of key performance metrics is provided in Table 4.

Without any adversarial detection mechanisms in place, FL systems are highly vulnerable to data poisoning and Byzantine attacks, where malicious clients can submit corrupted or misleading updates. Over time, these harmful contributions accumulate, leading to degraded model performance, unstable training, and in some cases, complete failure of the global model. Introducing detection and mitigation mechanisms helps identify and remove such malicious participants before aggregation, improving the robustness of the training process. As a result, the model is able to converge more reliably and achieve better overall performance, leading to more dependable outcomes.

The evaluation is conducted across multiple adversarial scenarios, including data poisoning and Byzantine attacks, to assess the proposed framework under varying threat models. The data poisoning experiments focus on user-level detection across training rounds, while the Byzantine scenarios evaluate the broader impact on global model performance and the effectiveness of the proposed defense during aggregation. To assess the model performance and effectiveness of the threat detection mechanism under varying levels of compromise, five users participated in the training process across five aggregation rounds. Users 1, 2, and 4 were simulated as legitimate participants throughout all rounds, downloading each global model, performing local training, and storing data to the BC and IPFS as expected. In contrast, Users 3 and 5 behaved as legitimate participants during the first two rounds but intentionally introduced poisoned data into their local datasets from Rounds 3 to 5 (Figure 5).

During Rounds 1 and 2, no data poisoning was detected, as all users behaved legitimately. However, in Rounds 3, 4, and 5, the detection mechanism successfully identified data poisoning attempts from Users 3 and 5. Specifically, User 3's dataset was heavily compromised (approximately 75% poisoned), while User 5's dataset contained a moderate level of poisoning (around 50%). Consequently, both users were excluded from participating in the global model aggregation process to preserve model integrity. As a result, only Users 1, 2, and 4 contributed to the aggregation process during in Rounds 3, 4, and 5.

The experimental results indicate that the proposed federated aggregation and threat detection approach successfully detect and mitigate data poisoning attacks. Table 5 details the "False" and "True" values indicating the status of threat detection for each user in every round. The value "False" denotes that all user activities were legitimate, whereas "True" indicates that the system detected a data poisoning attempt for that user during the specified round. When a poisoning attempt is detected, the corresponding client update is excluded from the aggregation for that round, preventing it from influencing the global model metrics.

As noted above in the context of the successful threat detection approach, a minor performance decline, in terms of prediction accuracy, followed the exclusion due to reduced data contribution. However, no significant degradation was observed, confirming that the poisoning did not affect the global model. By Rounds 4 and 5, all key performance metrics including accuracy, precision, recall, and $F1$-score showed a clear upward trend, demonstrating the model's resilience and ability to recover from adversarial interference. Repeated experiments under identical conditions consistently detected the same adversarial users, validating the reliability and reproducibility of the detection mechanism. Overall, the results confirm that the proposed aggregation strategy effectively maintains robustness, scalability, and integrity of the global model under poisoning attacks.

The detection of poisoned local models demonstrated strong robustness and some scalability when performed on an isolated incident. In the isolated case, a single user was poisoned with 20% of the dataset and the threat detection flagged this user in each aggregation round. This shows the ability for anomalies and poisoning to be detected even with a lower rate compared to the previous tests.

However, when poisoning with 20% was applied to all 5 users in unison to simulate a more coordinated-like attack scenario, some users were being detected but not immediately, and it only began detection consistently around later rounds, i.e., 4-5 in the experimental study. When the poisoning was reapplied with greater threshold, i.e., at approximately 30% across all users' datasets, the models were detected at a more consistent rate thanks to the historical comparison. These tests show that the FL aggregation process is highly sensitive to isolated incidents or those with varying levels of data poisoning but does show some robustness and scalability by identifying coordinated attacks to a certain extent, although detection in coordinated attack scenarios occurs later (around Rounds 3–5) compared to the more immediate detection observed in isolated cases.

Byzantine attack scenarios demonstrated that the integrated system exhibits nonlinear degradation under adversarial conditions, which is mitigated by a lightweight history-aware trust filter that leverages temporal consistency to improve robustness.

The FL configurations (3 or 5 clients over 5 or 7 rounds) were evaluated under consistent data and training settings, varying only client count, rounds, and adversarial conditions. A baseline without attacks established a reference, while the Byzantine setup introduced a single client performing a 30% label-flipping attack, and robustness was improved using a history-aware trust filter that excludes clients only when their current $F1$ is significantly below peers and anomalous relative to their past performance. The following table highlights the metrics that have been used for the evaluation. Table 6 shows the final FL comparison across benign, byzantine, and history-aware defended settings.

To assess the impact of Byzantine attacks and the response of the proposed framework, a series of FL experiments were conducted under controlled configurations. The FL setups included 3 and 5 clients over 5 and 7 communication rounds, respectively, with consistent data distribution and training parameters; only the number of clients, rounds, and adversarial conditions were varied. A baseline scenario without attacks was first established to provide a performance reference. Subsequently, a Byzantine setting was introduced, where a single malicious client performed a 30% label-flipping attack, leading to nonlinear degradation in model performance. To mitigate this effect, a lightweight history-aware trust filter was employed, leveraging temporal consistency to identify anomalous client behavior. The filter excludes client updates only when their current $F1$-score

TABLE 4 | Key performance metrics.

Category	Metric/rule	Formula	Description
Model performance	Accuracy	$(TP + TN)/(TP + TN + FP + FN)$	Overall proportion of correct predictions, where TP denotes True Positive, TN denotes True Negative. FP and FN represent False Positive and False Negative, respectively.
	Precision	$(TP)/(TP + FP)$	Proportion of predicted positives that are correct. This indicates low false alarms.
	Recall	$(TP)/(TP + FN)$	Proportion of actual positives correctly identified (sensitivity). It is very desirable in the health-related datasets, to avoid missing diagnosis of a disease.
	$F1$-score	$(2 \cdot (\text{Precision} \cdot \text{Recall}))/(\text{Precision} + \text{Recall})$	The harmonic mean of precision and recall, providing a balanced measure that accounts for both false positives and false negatives.
Threat detection metrics	MAD rule	$\lvert F1_i - \text{median}(F1)\rvert > 2.5 \cdot \text{MAD}$, where MAD represents Median absolute deviation	A robust method for detecting outliers based on the median. A user is flagged if their $F1$-score deviates significantly from the median by more than a chosen multiple (typically 2.5) of the median absolute deviation.
	Mean–std rule	$\lvert F1_i - \mu\rvert > k\sigma$	An outlier detection method based on the mean and standard deviation. A data point is flagged as an anomaly if it lies more than k standard deviations (typically 2 or 3) away from the mean.
	History-aware trust filter	$(\text{median}\ (F1) - F1_i) > \tau\_c$ AND $((\text{median}\ (F1_{\text{history}}) - \mu_i) > \tau_h$ OR $((F1_i^{(t-1)} - F1_i^t) > \tau_d)$ where $F1_i$ is the $F1$-score of client i, μ_i is the mean historical $F1$-score of Client i, and τ_c, τ_h, and τ_d are predefined thresholds for current deviation, historical deviation, and temporal drop, respectively.	A client is flagged as anomalous if its current performance deviates significantly from the median of participating clients and either exhibits historically poor performance or a sharp drop compared to previous rounds. This rule combines current deviation, historical behavior, and temporal consistency to identify Byzantine participants.
	Krum	$\text{core}\ (i) = \Sigma\lVert F1_i - F1_j\rVert$ for $j \in$ nearest $(n - f - 2)$ neighbors; select $m = n - f$ clients with lowest scores, where n is the total number of clients and f is the assumed number of Byzantine clients.	An aggregation method that scores each client by the sum of distances to its nearest neighbors. Clients with the lowest scores least isolated from the majority are selected for aggregation, reducing the influence of adversarial outliers.
	Trimmed Mean	$\text{TrimmedMean}\ (F1) = (1/(n - 2k))\ \Sigma F1_i$ for $i = k + 1$ to $n - k$, where the $F1$ values are sorted in ascending order and $k = \lfloor n \cdot \beta \rfloor$ for trim fraction β.	An aggregation method that removes the highest and lowest β fraction of client $F1$-scores before computing the mean. This reduces the influence of extreme values introduced by malicious clients.
	ECDSA signature verification	Algorithm 1	Nodes unable to produce a valid ECDSA signature over their own address cause the smart contract to revert the transaction and deny registration.
	Access control enforcement	Algorithm 2	Nodes not vetted by the RequestHandler remain at Status_Pending and are excluded from FL participation.

(Continues)

TABLE 4 | (Continued)

Category	Metric/rule	Formula	Description
Cost analysis metrics	Latency	$t_{end} - t_{start}$ where t_{start} and t_{end} represent time when the smart operation starts and ends, respectively.	Time (ms) taken to execute a smart contract operation.
	Gas cost	Gas used · Gas price	Computational cost required to execute a smart contract operation.
	Throughput	*Tx*/sec	Number of transactions processed per second.

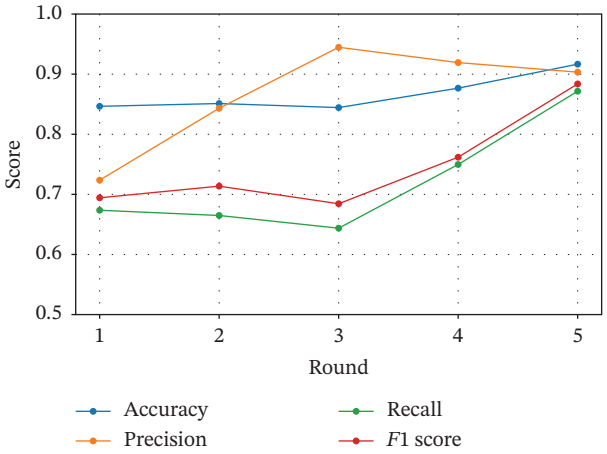

FIGURE 5 | Performance of global model across aggregation rounds in the presence of data poising by certain users and in certain rounds.

TABLE 5 | Summary of threat detection results (by user and round).

Round	User 1	User 2	User 3	User 4	User 5
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	True	False	True
4	False	False	True	False	True
5	False	False	True	False	True

deviates significantly from peer performance and their own historical trends. Table 5 presents the comparative results across benign, Byzantine, and defended settings, based on the evaluation metrics described below.

Figure 6 provides a round-by-round view of performance, illustrating convergence behavior and recovery from adversarial impact under the proposed defense. Table 6 also benchmarks the history-aware filter against two established Byzantine-robust aggregation methods, Krum and Trimmed Mean under identical attack settings. The history-aware filter matches Krum across all four configurations, with both methods consistently outperforming Trimmed Mean. This result confirms that the proposed lightweight filter achieves comparable Byzantine robustness to the classical Krum method, while operating directly on $F1$-score metrics rather than requiring access to full gradient vectors, constitutes a practical advantage in the BC-based FL setting where only model performance metrics are exchanged and recorded on-chain.

The framework achieved an approximate throughput of 31.651 transactions per second (TPS) (tx/s) under the local Ganache benchmark, indicating the rate at which BC operations can be processed. The results in Table 7 demonstrate that most BC operations exhibit consistently low latency, typically ranging between approximately 19 ms and 39 ms, indicating efficient transaction processing within the permissioned environment.

Lightweight operations such as permission requests (requestPermission) and round completion (completeRound) incur minimal latency and gas costs, reflecting their limited computational complexity. In contrast, more resource-intensive

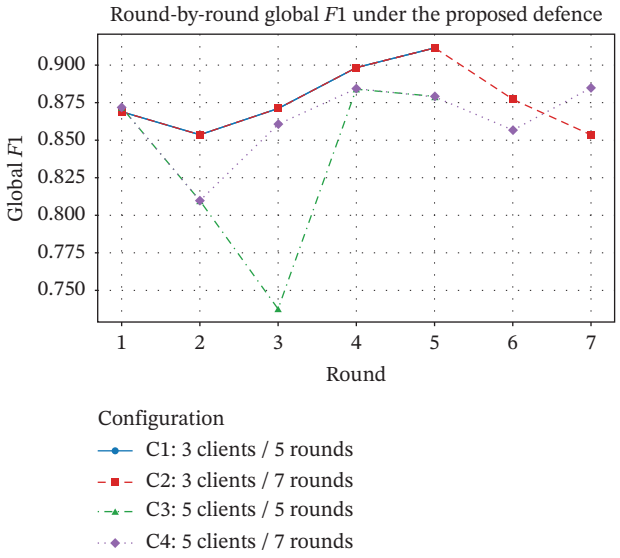

FIGURE 6 | Global F1 performance under baseline, Byzantine, and history-aware defended conditions across all configurations.

operations, particularly sendUpload, exhibit the highest latency (38.771 ms). Gas consumption varies, notably across operations, with the upload step, again, incurring the highest gas use of 272,472. In contrast, lightweight functions such as permission requests and round completion remain relatively low-cost consumption. Overall, while BC integration introduces measurable overhead, these costs consumptions are concentrated in a limited set of operations and remain manageable for moderate-scale deployments.

In contrast to many BC-enabled FL studies that primarily emphasize system architecture, access control, auditability, or secure storage, this work extends the evaluation to include detailed behavioral analysis under adversarial conditions. Moreover, the incorporation of a history-aware trust filter introduces an explicit mitigation layer, enabling not only the observation of Byzantine effects but also active robustness enhancement.

Takeaways: The experimental results demonstrate that the proposed framework achieves robust and reliable performance under both data poisoning and Byzantine attack scenarios. Using *F*1-score as the primary metric, the system effectively detects and excludes malicious clients, preventing degradation of the global model. In poisoning experiments, adversarial users were consistently identified from the onset of malicious behavior, with only minor temporary performance decline and clear recovery in later rounds, confirming model resilience and reproducibility. The framework also shows strong sensitivity to isolated attacks and reasonable scalability under coordinated scenarios, although detection may be slightly delayed at lower poisoning levels. Under Byzantine settings, the introduction of a history-aware trust filter significantly mitigates nonlinear performance degradation, consistently improving *F*1-scores and enabling reliable removal of malicious participants across configurations. From a system perspective, BC integration introduces moderate but manageable overhead, with most operations exhibiting low latency (≈19–39 ms) and acceptable throughput (~31.65 tx/s), while the upload step

TABLE 6 | FL *F*1-score comparison across baseline (benign), Byzantine, history-aware defended, Krum, and Trimmed Mean settings.

Configuration	Clients	Rounds	Baseline *F*1-score	Byzantine *F*1-score	History-aware defense *F*1-score	Krum *F*1-score	Trimmed Mean *F*1-score	Attacker removed
C1	3	5	0.9021	0.7762	0.9117	0.9036	0.8955	5/5 rounds
C2	3	7	0.8586	0.7824	0.8536	0.9045	0.8923	7/7 rounds
C3	5	5	0.7007	0.8280	0.8791	0.8691	0.8549	4/5 rounds
C4	5	7	0.7007	0.8340	0.8849	0.9016	0.8966	7/7 rounds

TABLE 7 | The blockchain operation performance summary (latency and gas usage).

Operation	Mean latency (ms)	Mean gas used	Comment
requestPermission	23.830	31.526	Low latency permission request
setPermission	24.196	39.822	Admin-side approval/denial overhead
startRound	26.879	130.012	Round-start management cost
sendUpload	38.771	272.472	Most expensive core upload step
validateRequest	28.394	107.536	Validation/approval overhead
completeRound	19.462	37.458	Low latency round closure

remains the primary bottleneck due to higher computational and verification costs. In summary, the findings confirm that the proposed approach effectively balances security, robustness, and performance, making it suitable for practical deployment in adversarial environments.

5.2 | Sybil Attack Detection

A two-phase scenario Sybil attack simulation was conducted on a network of 100 nodes under three different case studies, i.e., (i) low, (ii) balanced, and (iii) high number of malicious nodes. In this experimental study, the low case consists of 93 honest and 7 malicious nodes, balanced consists half of the nodes to be malicious, while high consists of 7 honest and 93 malicious nodes. These three cases are aimed at evaluating the framework's ability to distinguish legitimate participants from adversarial identities along with economic impact, latency, and costs under the varying intensity of malicious activities. Each of the three cases is tested under two scenarios (Table 8).

5.2.1 | Scenario 1: Without Digital Signature

The attacker attempted to register Sybil nodes using only Ethereum addresses without valid ECDSA signatures across all three case studies. In every case, all attempts failed, as the smart contract enforced signature verification via ecrecover and reverted invalid transactions. Consequently, no Sybil node was successfully registered under any of the three cases, and all malicious nodes remained at Status_None (Unregistered). Honest nodes were equally unable to register without a valid signature, demonstrating strong resistance to basic identity spoofing.

5.2.2 | Scenario 2: With Digital Signature

The attacker generated unique key pairs and funded them with sufficient Ether to broadcast signed transactions. While these nodes successfully registered, the permission layer put them into Status_Pending as only the admin can elevate nodes to Status_Granted after manual vetting. In the low scenario, 7 Sybil nodes successfully registered but remained at Status_Pending, while all 93 honest nodes were approved. In the balanced scenario, all 50 Sybil nodes were blocked at Status_Pending, while all 50 honest nodes were granted access. In the high scenario, all 93 Sybil nodes remained at Status_Pending, while the 7 honest nodes were correctly elevated to Status_Granted. Across all three scenarios, every Sybil node was confined to Status_Pending without exception.

Although these nodes were registered, they possessed no operational permissions. They were unable to download global model weights or upload local updates. The attack therefore had zero impact on the FL process across all three scenarios. Table 9 presents the full cost and performance breakdown per scenario in Phase 2.

As attack intensity increases, the economic burden shifts entirely onto the attacker. In the low scenario, the attacker spends only £3.16 for 7 Sybil nodes, whereas in the balanced scenario, this rises to £22.55 for 50 nodes, and in the high scenario reaches £41.95 for 93 nodes, none of which gain any access across any scenario. The per-node cost remains consistent regardless of scenario, confirming that scaling the attack scales the cost linearly with no additional return.

In the scenario where an admin is compromised and grants unauthorized permissions to malicious nodes, the framework preserves integrity through a secondary security layer. As shown in Figure 3, the model aggregation layer acts as a final gatekeeper, detecting and mitigating data poisoning attempts before they affect the global model.

There is a degree of trust placed in the admin role and around a centralized authority. This design choice was made deliberately, and the framework does not vest all privileges in a single admin. As enforced by the smart contract, the admin role is subdivided into three independently assigned subroles, RoundHandler, RequestHandler, and verifier, each being

TABLE 8 | A comparison of Sybil attack outcomes under different cryptographic conditions across two phases.

Attribute	Scenario 1	Scenario 2
Digital signature	No signature	Valid ECDSA signature
Node registry count	0 Sybils registered (all scenarios)	Low: 7/balanced: 50/High: 93
Honest node status	Status_None	Status_Granted (all scenarios)
Sybil node status	Status_None	Status_Pending (all scenarios)
Economic impact	0	Low: ~£3.16/balanced: ~£22.55/High: ~£41.95
FL model impact	Zero	Zero (nodes are in isolation)

TABLE 9 | Cost and performance with honest and Sybil nodes in Phase 2 (Eth to GBP cost 2800 recorded on 03/04/2026).

Use case	Node type	Count	Avg latency (ms)	Std latency (ms)	Avg gas usage	Std gas usage	Avg cost (£)	Std cost (£)	Total cost (£)
Low	Honest	93	28.23	6.52	80546.65	5.96	0.4511	5×10^{-5}	41.95
	Sybil	7	32.29	5.47	80543.14	9.44	0.4510	79×10^{-5}	3.16
Balanced	Honest	50	31.67	5.39	80545.68	7.18	0.4511	6×10^{-5}	22.55
	Sybil	50	30.97	4.76	80545.92	7.11	0.4511	59×10^{-5}	22.55
High	Honest	7	31.82	4.07	80546.57	9.07	0.4511	76×10^{-5}	3.16
	Sybil	93	32.47	4.42	80544.58	7.82	0.4511	65×10^{-5}	41.95

assigned to a separate address and restricted to a distinct set of functions. The RoundHandler can only initiate and manage training rounds; the RequestHandler can only grant or deny user permissions, and the verifier can only validate uploaded model contributions and complete rounds. The smart contract reverts any attempt by one role to execute another's functions, meaning no single compromised admin wallet can simultaneously manipulate round management, access control, and model validation. This role separation follows the principle of least privilege, which is itself a core tenet of ZT.

In the scenario where an admin wallet is compromised, the impact is limited to that subrole's function only. Should the RequestHandler be compromised and grant unauthorized permissions to malicious nodes, the model aggregation layer acts as a secondary gatekeeper, with the history-aware trust filter (Algorithm 4) detecting and excluding anomalous client contributions before they affect the global model. All admin actions are furthermore cryptographically signed and immutably recorded on-chain, providing a tamper-evident audit trail that supports postincident forensic analysis.

Takeaways: The proposed framework achieves robust Sybil attack mitigation by combining cryptographic identity verification with permissioned access control, ensuring that malicious nodes remain isolated and have no impact on model training. The attack is further discouraged economically, as costs scale linearly with the number of Sybil identities without yielding any benefit. This demonstrates both security and practical deployability under adversarial conditions.

6 | Conclusion

Although modern healthcare systems have adopted various security mechanisms, they continue to face persistent vulnerabilities particularly in the protection of sensitive medical data, the authentication of users and IoT devices, and the assurance of data integrity [52, 53]. BC technology has emerged as a promising solution to address these by providing a decentralized and immutable ledger system that ensures traceability and resistance to tampering. By employing cryptographic hashing, BC enables secure data encryption while facilitating reliable authentication and verification of both users and connected medical devices. In addition, it supports real-time monitoring and guarantees the availability, confidentiality, and integrity of health data. One of the most transformative aspects of BC lies in its capacity to enhance patient privacy and foster greater user control over personal health information. Through transparent and verifiable access mechanisms, BC promotes data ownership and accountability, thereby strengthening user engagement and trust in digital healthcare systems. The experiment showed that when BC is deployed in conjunction with the proposed federated aggregation and threat detection approach, data poisoning, Byzantine, and Sybils attacks can be successfully detected and mitigated. BC, when integrated with FL, can provide a decentralized, transparent, and immutable environment that enhances the overall trustworthiness and resilience of the learning system. This integration is particularly promising in healthcare, where auditability, accountability, and tamper-proofing are critical.

Modern healthcare systems continue to face security and privacy challenges around sensitive medical data, user and IoT-device authentication, access control, and data integrity. This study proposed and evaluated a BC-enabled FL framework strengthened by ZT principles, smart-contract-based access control, digital signature verification, and role-based authorization. The experimental results show that the framework can detect and mitigate data poisoning, Byzantine behavior, Sybil attacks, and unauthorized access attempts. Poisoned and Byzantine updates were identified and excluded before aggregation, while Sybil nodes either failed registration or remained isolated in Status_Pending, producing no impact on the FL process. The BC layer further supported transparency, traceability, and tamper-resistant recording of permission requests, model uploads, validation decisions, and FL round management.

This research clearly has its limitations. The experimental Phase 2 is based on a simple testbed in which a single dataset was used. In addition, there could have been a measurement of latency and of the performance time taken for aggregating multiple users at once. At the same time, the BC is running on a local environment. While it simulates the necessary security and authentication elements, it makes it more challenging to assess other factors such as the communication overhead and TPS. These limitations restrict assessment of real-world communication overheads, network congestion, and large-scale transaction performance. A further concern is that should all three admins be simultaneously compromised, the entire network would be at risk, which is an inherent characteristic of BC architectures. In this context, future work could profitably evaluate the framework using larger and more heterogeneous healthcare IoT datasets, more users, and real-world or permissioned BC deployments. In addition, future studies could explore Decentralised Autonomous Organisations (DAOs) to assess possible benefits and trade-offs for complete decentralized management.

Nevertheless, the authors maintain that this study makes significant contributions in three main areas. Firstly, as regards problem identification, the systematic analysis of existing vulnerabilities and threats in healthcare systems, including Sybil and Byzantine attacks, and data poisoning, highlighted critical security and privacy issues. Secondly, the design, development, and implementation of a proof-of-concept framework that integrates BC, ZT principles, and FL into one robust and secure architecture and constitutes a novel contribution to existing research literature in this field of study. It illustrates how the framework can be applied to a healthcare setting though ensuring no raw data are ever shared; all users and transactions are authenticated regardless of the user's role or access; and the decentralized environment can allow for high availability. Thirdly, the evaluation of the proposed risk-prediction framework through healthcare-based scenarios allows a comprehensive assessment of its security, reliability, and performance.

Choquet [31] recently noted that "as the healthcare industry continues to embrace digital transformation, BC technology will play a pivotal role in enabling the shift to patient-driven interoperability, ultimately improving the quality, efficiency, and accessibility of healthcare services" (p. 2). The main contribution of this work is a proof-of-concept framework that enhances the potential role and value of BC by its integration with ZT and FL in a unified security architecture. Although developed for healthcare, the approach is also applicable to other sensitive-data domains such as finance, public-sector systems, and critical infrastructure, where privacy, auditability, resilience, and trusted collaboration are essential.

Author Contributions

Conceptualization: M.W. and S.M.; methodology: M.W., S.M., J.C., and M.R.; validation: M.W., S.M., J.C., and M.R.; formal analysis: M.W., S.M., J.C., M.R., and M.H.A.R.; data curation: M.W., S.M., J.C., M.R., and M.H.A.R.; writing–original draft preparation: M.W., S.M., J.C., and M.R.; writing–review and editing: M.W., S.M., J.C., M.R., and M.H.A.R.; visualization: M.W. and S.M.; project administration: M.W.

Funding

This research received no external funding.

Disclosure

All authors have read and agreed to the published version of the manuscript.

Ethics Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

All sources used in this article are available via the cited reference list.

References

1. B. Metin, F. Özhan, and M. Wynn, "Digitalisation and Cybersecurity: towards an Operational Framework," *Electronics* 13, no. 21 (2024): 4226, https://doi.org/10.3390/electronics13214226.

2. S. Dhar and I. Bose, "Securing IoT Devices Using Zero Trust and Blockchain," *Journal of Organizational Computing and Electronic Commerce* 31 (2020): 1–17, https://doi.org/10.1080/10919392.2020.1831870.

3. S. S. Bhuyan, U. Kabir, J. M. Escareno, et al., "Transforming Healthcare Cybersecurity from Reactive to Proactive: Current Status and Future Recommendations," *Journal of Medical Systems* 44, no. 98 (2020): 98, https://doi.org/10.1007/s10916-019-1507-y.

4. R. Akkaoui, "Blockchain for the Management of Internet of Things Devices in the Medical Industry," *IEEE Transactions on Engineering Management* 70, no. 8 (2023): 2707–2718, https://doi.org/10.1109/tem.2021.3097117.

5. Y. K. Aluri, S. Katlagunta, T. Musunuru, A. Mylarapu, K. V. Kuruba, and V. E. Jyothi, "Healthcare Supply Chain Security with Zero Knowledge Proof Authentication in Blockchain for Personalized Healthcare Monitoring," in *3rd IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics, ICDCECE 2024* (2024).

6. G. Vukotich, "Healthcare and Cybersecurity: Taking a Zero Trust Approach," *Health Services Insights* 16 (2023): 11786329231187826, https://doi.org/10.1177/11786329231187826.

7. The White House, *M-22-09 Federal Zero Trust Strategy* (Washington, DC: The White House, 2022).

8. The White House, *Executive Order on Improving the Nation's Cybersecurity* (Washington, DC: The White House, 2021), https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-thenations-cybersecurity/.

9. L. Alton, "Intelligent Systems vs. Artificial Intelligence: How to Help Your Clients," *TechTarget. Data Science Central* (2018): https://www.datasciencecentral.com/intelligent-systems-vs-artificial-intelligence-how-to-help-your/.

10. H. Kumar, A. Martin, M. Wynn, and M. Lakshmi, "Research Challenges in Deep Generative Models for Healthcare and Medical Applications," in *Advancements in Deep Generative Models for Healthcare and Medical Applications*, ed. S. Balasubramaniam and S. N. Kadry (Abingdon, UK: Routledge, 2025).

11. S. Mjeat, M. Yousif, S. Bader, O. Mohammed, and A. H. Saeed, "A Public Key Infrastructure Based on Blockchain for IoT-based Healthcare Systems," *Journal of Cybersecurity and Information Management* 15 (2025): 233–243.

12. J. Gill and P. Johnson, *Research Methods for Managers*, 3rd ed. (London, UK: Sage, 2002).

13. E. Bell, A. Bryman, and B. Harley, *Business Research Methods* (Oxford, UK: Oxford University Press, 2018).

14. H. Snyder, "Literature Review as a Research Methodology: An Overview and Guidelines," *Journal of Business Research* 104 (2019): 333–339, https://doi.org/10.1016/j.jbusres.2019.07.039.

15. C. Hart, *Doing a Literature Review: Releasing the Research Imagination* (Thousand Oaks, CA, USA: Sage Publishing, 2018).

16. A. Fink, *Conducting Research Literature Reviews: From the Internet to Paper* (Thousand Oaks, CA, USA: Sage Publishing, 2019).

17. B. Levering, "Concept Analysis as Empirical Method," *International Journal of Qualitative Methods* 1 (2002): 35–48, https://doi.org/10.1177/160940690200100104.

18. Y. Jabareen, "Building a Conceptual Framework: Philosophy, Definitions, and Procedure," *International Journal of Qualitative Methods* 8, no. 4 (2009): 49–62, https://doi.org/10.1177/160940690900800406.

19. J. Popay, H. Roberts, A. Sowden, M. Petticrew, L. Arai, and M. Rodgers, *Guidance on the Conduct of Narrative Synthesis in Systematic Reviews* (Lancaster, UK: Lancaster University, 2006).

20. A. L. Porter, A. Kongthon, and J. C. Lu, "Research Profiling: Improving the Literature Review," *Scientometrics* 53, no. 3 (2002): 351–370, https://doi.org/10.1023/A:1014873029258.

21. Indeed Editorial Team, "Experimental Research: Definition, Types and Examples. Indeed," (2024), https://www.indeed.com/career-advice/career-development/experimental-research.

22. D. Dua and C. Graff, "Breast Cancer Wisconsin (Diagnostic) Data Set. UCI Machine Learning Repository," (2019), https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data.

23. S. Lee, J.-H. Huh, and H. Woo, "Security System Design and Verification for Zero Trust Architecture," *Electronics* 14, no. 4 (2025): 643, https://doi.org/10.3390/electronics14040643Y.

24. C. Thornell, "3 Ways to Use Zero Trust to Reduce Cybercrime Risk in Healthcare," *Akamai* (2023): https://www.akamai.com/blog/security/use-zero-trust-to-reduce-cybercrime-risk.

25. L. Coventry and D. Branley, "Cybersecurity in Healthcare: a Narrative Review of Trends, Threats and Ways Forward," *Maturitas* 113 (2018): 48–52, https://doi.org/10.1016/j.maturitas.2018.04.008.

26. O. C. Edo, D. Ang, P. Billakota, and J. C. Ho, "A zero-trust Architecture for Health Information Systems," *Health Technology* 14, no. 1 (2024): 189–199, https://doi.org/10.1007/s12553-023-00809-4.

27. I. Milligan-Pate, "Zero Trust for Healthcare," (2022), https://www.zscaler.com/blogs/company-news/zero-trust-healthcare.

28. Y. Zhong, M. Zhou, J. Li, et al., "Distributed Blockchain-based Authentication and Authorization Protocol for Smart Grid," *Wireless Communications and Mobile Computing* 1 (2021): 5560621, https://doi.org/10.1155/2021/5560621.

29. IBM, "What is Interoperability in the Healthcare Sector?" (2026), https://www.ibm.com/fr-fr/think/topics/interoperability-in-healthcare.

30. W. J. Gordon and C. Catalini, "Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability," *Computational and Structural Biotechnology Journal* 16 (2018): 224–230, https://pubmed.ncbi.nlm.nih.gov/30069284/, https://doi.org/10.1016/j.csbj.2018.06.003.

31. J. Choquet, "Blockchain Facilitation of Patient-Driven Interoperability: Overcoming Challenges and Harnessing Opportunities in Healthcare," *Journal of Health Care and Prevention* (2024): https://www.omicsonline.org/open-access/blockchain-facilitation-of-patientdriven-interoperability-overcoming-challenges-and-harnessing-opportunities-in-healthcare-130391.html.

32. T. V. Le and C. L. Hsu, "A Systematic Literature Review of Blockchain Technology: Security Properties, Applications and Challenges," *Journal of Internet Technology* 22 (2021): 789–802.

33. C. Daah, A. Qureshi, I. Awan, and S. Konur, "Enhancing Zero Trust Models in the Financial Industry Through Blockchain Integration: A Proposed Framework," *Electronics* 13, no. 5 (2024): 865, https://doi.org/10.3390/electronics13050865.

34. H. Wang, Z. Zheng, S. Xie, H. N. Dai, and X. Chen, "Blockchain Challenges and Opportunities: A Survey," *International Journal of Web and Grid Services* 14, no. 4 (2018): 352, https://doi.org/10.1504/ijwgs.2018.095647.

35. Y. Liu, X. Hao, W. Ren, et al., "A Blockchain-Based Decentralized, Fair and Authenticated Information Sharing Scheme in Zero Trust internet-of-things," *IEEE Transactions on Computers* 72, no. 2 (2023): 501–512, https://doi.org/10.1109/tc.2022.3157996.

36. M. Sultana, A. Hossain, F. Laila, K. Abu Taher, and M. N. Islam, "Towards Developing a Secure Medical Image Sharing System Based on Zero Trust Principles and Blockchain Technology," *BMC Medical Informatics and Decision Making* 20, no. 1 (2020): 256, https://doi.org/10.1186/s12911-020-01275-y.

37. G. Ramezan and E. Meamari, "zk-IoT: Securing the Internet of Things with Zero-Knowledge Proofs on Blockchain Platforms," in *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (IEEE, May 2024), 1–7, https://doi.org/10.1109/ICBC59979.2024.10634342.

38. K. W. Narayan and N. Shah, "Decentralizing Health Care: History and Opportunities of Web3," *JMIR Formative Research* 8 (2024): e52740, https://doi.org/10.2196/52740.

39. T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "FL: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine* 37, no. 3 (2020): 50–60, https://doi.org/10.1109/msp.2020.2975749.

40. M. Ali, F. Naeem, M. Tariq, and G. Kaddoum, "FL for Privacy Preservation in Smart Healthcare Systems: a Comprehensive Survey," *IEEE Journal of Biomedical and Health Informatics* 27, no. 2 (2022): 778–789.

41. S. Singh, S. Rathore, O. Alfarraj, A. Tolba, and B.-W. Yoon, "A Framework for privacy-preservation of IoT Healthcare Data Using FL and Blockchain Technology," *Future Generation Computer Systems* 129 (2022): 380–388, https://doi.org/10.1016/j.future.2021.11.028.

42. P. Kairouz, H. McMahan, B. Avent, et al., "Advances and Open Problems in FL," *Foundations and Trends in Machine Learning* 14, no. 1–2 (2021): 1–210.

43. S. Pouriyeh, O. Shahid, R. M. Parizi, et al., "Secure Smart Communication Efficiency in Federated Learning: Achievements and Challenges," *Applied Sciences* 12, no. 18 (2022): 8980, https://doi.org/10.3390/app12188980.

44. S. R. Pokhrel, L. Yang, S. Rajasegarar, and G. Li, "Robust Zero Trust Architecture: Joint Blockchain Based Federated Learning and Anomaly Detection Based Framework," in *Proc. SIGCOMM Workshop Zero Trust Architecture for Next Generation Communications (ZTA-NextGen '24)* (New York, NY, USA: Association for Computing Machinery, 2024), 7–12, https://doi.org/10.1145/3672200.3673878.

45. M. M. Salim, M. Kim, S. Kumar Singh, and J. H. Park, "Zero-Trust Blockchain-Enabled Framework for Scalable and Secure IoT Networks," *Future Generation Computer Systems* 175 (2026): 108093, https://www.sciencedirect.com/science/article/pii/S0167739X25003875, https://doi.org/10.1016/j.future.2025.108093.

46. J. M. Bernabé Murcia, E. Cánovas, J. García-Rodríguez, A. M. Zarca, and A. Skarmeta, "Decentralised Identity Management Solution for Zero-Trust Multi-Domain Computing Continuum Frameworks," *Future Generation Computer Systems* 162 (2025): 107479, https://www.sciencedirect.com/science/article/pii/S0167739X24004291, https://doi.org/10.1016/j.future.2024.08.003.

47. M. Vusumuzi and M. Godwin, "AI-Driven Zero-Trust Models for Blockchain-Supported Healthcare Ecosystems," in *Proceedings of the 2025 International Conference on Artificial Intelligence and its Applications (icARTi '25)* (New York, NY, USA: Association for Computing Machinery, 2025), 1–11, https://doi.org/10.1145/3774791.3774801.

48. M. A. Aleisa, "Blockchain-Enabled Zero Trust Architecture for Privacy-Preserving Cybersecurity in IoT Environments," *IEEE Access* (2025): https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=%26arnumber=10839415.

49. X. Zhu, T. Lai, and H. Li, "Privacy-Preserving Byzantine-resilient Swarm Learning for e-healthcare," *Applied Sciences* 14, no. 12 (2024): 5247, https://doi.org/10.3390/app14125247.

50. P. Chinnasamy, A. Albakri, M. Khan, A. A. Raja, A. Kiran, and J. C. Babu, "Smart contract-enabled Secure Sharing of Health Data for a Mobile Cloud-based E-health System," *Applied Sciences* 13, no. 6 (2023): 3970, https://doi.org/10.3390/app13063970.

51. A. Vishnukumar, V. Govarthan, S. Deva, S. Barath, V. Yuvanesh, and K. Suvathika, "Sybil-Resistant Decentralized blockchain-enabled Federated Learning for Edge AI: Robustness and Incentive Mechanisms,"

Proc. ICSTSN 2025 (2025): 1–5, https://doi.org/10.1109/ICSTSN67075.2025.11398054.

52. S. Mallick, S. Sobhanayak, and R. K. Lenka, "Blockchain-Enhanced IoT Ecosystem for Healthcare: Transformative Potentials, Applications, Challenges, Solutions, and Future Perspectives," *Computers & Industrial Engineering* 197 (2024): 110538, https://doi.org/10.1016/J.CIE.2024.110538.

53. W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-Based FL for Securing Internet of Things: a Comprehensive Survey," *ACM Computing Surveys* 55, no. 9 (2023): 1–4, https://doi.org/10.1145/3560816.