IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 April 2024, accepted 9 May 2024, date of publication 14 May 2024, date of current version 22 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3401039

RESEARCH ARTICLE

Fortifying Federated Learning in IIoT: Leveraging Blockchain and Digital Twin Innovations for Enhanced Security and Resilience

SAHAYA BENI PRATHIBA[1], (Member, IEEE), YESHWANTH GOVINDARAJAN[2], VISHAL PRANAV AMIRTHA GANESAN[2], ANIRUDH RAMACHANDRAN[2], ARIKUMAR K. SELVARAJ[3], ALI KASHIF BASHIR[4,5,6], (Senior Member, IEEE), AND THIPPA REDDY GADEKALLU[7,8], (Senior Member, IEEE)

[1]Centre for Cyber Physical Systems, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
[2]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
[3]Department of Data Science and Business Systems, College of Engineering and Technology, SRM Institute of Science and Technology (SRMIST), Kattankulathur 603203, India
[4]Department of Computing and Mathematics, Manchester Metropolitan University, M15 6BH Manchester, U.K.
[5]Woxsen School of Business, Woxsen University, Sangareddy District, Hyderabad, Telangana 502345, India
[6]Department of Computer Science and Mathematics, Lebanese American University, Beirut 13-5053, Lebanon
[7]Division of Research and Development, Lovely Professional University, Phagwara 144001, India
[8]Center of Research Impact and Outcome, Chitkara University, Punjab 140401, India

Corresponding author: Sahaya Beni Prathiba (sahayabeni@ieee.org)

This work is supported by the Vellore Institute of Technology, Chennai, India.

ABSTRACT Ensuring robustness against adversarial attacks is imperative for Machine Learning (ML) systems within the critical infrastructures of the Industrial Internet of Things (IIoT). This paper addresses vulnerabilities in IIoT systems, particularly in distributed environments like Federated Learning (FL) by presenting a resilient framework - Secure Federated Learning (SFL) specifically designed to mitigate data and model poisoning, as well as Sybil attacks within these networks. Sybil attacks, involving the creation of multiple fake identities, and poisoning attacks significantly compromise the integrity and reliability of ML models in FL environments. Our SFL framework leverages a Digital Twin (DT) as a critical aggregation checkpoint to counteract data and model poisoning attacks in IIoT's distributed settings. The DT serves as a protective mechanism during the model update aggregation phase, substantially enhancing the system's resilience. To further secure IIoT infrastructures, SFL employs blockchain-based Non-Fungible Tokens (NFTs) to authenticate participant identities, effectively preventing Sybil attacks by ensuring traceability and accountability among distributed nodes. Experimental evaluation within IIoT scenarios demonstrates that SFL substantially enhances defensive capabilities, maintaining the integrity and robustness of model learning. Comparative results reveal that the SFL framework, when applied to IIoT federated environments, achieves a commendable 97% accuracy, outperforming conventional FL approaches. SFL also demonstrates a remarkable reduction in loss rate, recording just 0.07 compared to the 0.14 loss rate experienced by standard FL systems. These findings highlight the efficiency and applicability of the SFL framework in enhancing data security and traceability within the IIoT ecosystem.

INDEX TERMS Blockchain, data poisoning, decentralized federated learning, digital twin, industrial internet of things, model poisoning, non-fungible tokens, Sybil attack.

I. INTRODUCTION

Amidst the current era of data-centric technologies, Machine Learning (ML) models are fundamental to extracting actionable insights, and this is particularly evident in the Industrial

The associate editor coordinating the review of this manuscript and approving it for publication was Stefano Scanzio.

Internet of Things (IIoT). Digital Twins (DTs), sensory data, and interconnected systems form the backbone of IIoT, where Federated Learning (FL) has emerged as a game-changing architecture. FL's decentralized nature ensures the efficient processing of distributed IIoT data and addresses some of the most pressing privacy concerns by allowing numerous compute nodes to collaboratively hone a shared global model without exposing individual datasets [1], [2]. In addition, FL leverages distributed computational resources to overcome the limitations of a central server, thereby enhancing scalability and responsiveness in IIoT environments [3], [4]. However, while FL is inherently more privacy-preserving than centralized approaches, it is not immune to security threats, such as model poisoning and Sybil attacks, which jeopardize the integrity of collective model training in the IIoT context [5], [6]. Such attacks pose a significant threat to the collaborative learning paradigm, potentially compromising the models integral to industrial applications.

In the IIoT, model poisoning attacks are characterized by malicious participants altering their model updates to degrade the model's performance, while data poisoning directly manipulates the training dataset to skew the learning outcome. Sybil attacks intensify these vulnerabilities, enabling adversaries to fabricate numerous fictitious identities, thereby disproportionately influencing the model training phase [7].

To address these persistent challenges within the IIoT, we propose a comprehensive and robust FL framework, named Secure Federated Learning (SFL). The SFL framework employs blockchain technology to create an unalterable ledger that serializes model updates, effectively deterring unauthorized modifications and preserving the integrity of the data [8], [9]. By incorporating Non-Fungible Tokens (NFTs), which utilize the Edwards-curve Digital Signature Algorithm (EdDSA) for unique identification, the framework thwarts Sybil attacks, assuring the authenticity of each participant in the FL network [10], [11]. The DTs [12], [13], which serve as secure and immutable reference models within the IIoT ecosystem, play an instrumental role in detecting and remedying any incidents of model or data poisoning [14].

A. MOTIVATIONS

In the design of the SFL framework for IIoT environments, significant emphasis was placed on selecting components that bolster security, efficiency, and system integrity. The incorporation of DTs, Blockchain technology, and NFTs was driven by a meticulous evaluation of their functionalities and the distinct advantages they offer. DTs are employed within the SFL framework to serve as accurate and immutable reference models. This strategic choice allows for the benchmarking and integrity verification of the FL model, enabling the detection and rectification of discrepancies that may suggest data or model poisoning. Beyond security, DTs offer the capability for simulation and prediction, facilitating system optimization and proactive security measures by allowing for the analysis of potential system behaviors in a controlled environment. They also provide a reliable foundation for system resilience and recovery, ensuring that, in the event of a breach, the system can revert to a verified and secure state.

Blockchain technology is integrated into the SFL framework to leverage its immutability and traceability, offering an unalterable ledger that records all transactions, including model updates. This choice enhances the security and transparency of the FL process, mitigating centralized points of failure and promoting accountability among participants. Furthermore, the decentralized nature of blockchain serves as a robust defense mechanism against Sybil attacks, ensuring participant authenticity and the legitimacy of contributions to the federated model.

The deployment of NFTs within the SFL framework introduces a novel approach to user authentication and model update validation. NFTs, distinguished by their unique cryptographic signatures, ensure the uniqueness of participant identities, preventing identity duplication and thereby mitigating the risk of Sybil attacks. This not only enhances the security framework but also introduces a mechanism for recognizing and incentivizing contributions to the FL process, fostering a more secure and collaborative environment.

These components—DTs, Blockchain technology, and NFTs—are not mere additions to the SFL framework but are foundational to its ability to address the complex security challenges inherent in IIoT environments. Their integration is based on a strategic consideration of the benefits they bring, ensuring the framework's robustness, reliability, and resilience against a myriad of cyber threats.

B. ARCHITECTURE OF THE SECURE FEDERATED LEARNING FRAMEWORK

Figure 1 presents the integrated architecture of the SFL framework, specifically tailored for IIoT environments. This architecture's heart is the Central Server, which orchestrates the entire FL process. It initiates the training by distributing the global model to the Nodes—various connected devices or computational resources scattered across the IIoT landscape. These Nodes are responsible for conducting local training with their unique datasets, after which they send their model updates back to the Central Server. The Central Server's role extends beyond mere distribution and collection; it is critical in aggregating the model updates from all Nodes. This aggregation is a delicate process that synthesizes distributed learning, updating the global model's parameters to better fit the collective data while ensuring the model's continuous improvement. Ensuring the integrity of this iterative learning process, the Blockchain component is introduced as a secure and immutable ledger. It records each transaction of the model updates, which includes the weights adjusted through local training. This immutability serves a dual purpose: it not only guarantees the traceability of each model's evolution, critical for accountability and auditability but also acts as a robust line of defense against potential model or data inconsistencies that might arise.

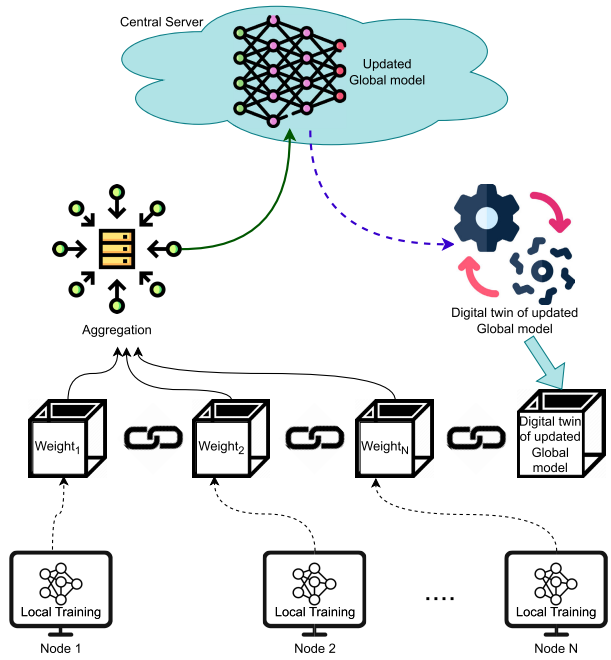

FIGURE 1. Architectural Diagram of Secured Federated Learning (SFL) Framework.

Complementing the Blockchain is the concept of a DT, a reliable replication of the global model that acts as a reference point. By periodically benchmarking the updated global model against this DT, any significant discrepancies can be promptly identified. Such anomalies may suggest data or model poisoning, prompting the system to revert to the DT's state to restore a verified model configuration. The SFL framework's architecture thus encapsulates a cyclical and secure exchange of intelligence between the Nodes and the Central Server, with the Blockchain underpinning the system's trustworthiness. This setup not only amplifies the collaborative power of distributed learning but also fortifies the system against the multifaceted security threats endemic to IIoT domains.

To provide a visual overview of the SFL framework's architecture and design, an architectural diagram is presented in Fig. 1. This diagram illustrates the interconnected components and their interactions, serving as a valuable reference for understanding the structural aspects of the framework.

The intended innovations are evident in the strategic deployment of SFL, where DTs are instantiated and a blockchain-based transaction system for model versioning is established. The commitment to security is further emphasized in the user authentication process, wherein NFTs ensure uniqueness among participants using EdDSA. Finally, a resilient model aggregation mechanism complemented by DT benchmarking signifies a leap toward robust FL systems. The primary contributions of this article are as follows:

- To fortify FL against malicious activities in IIoT, we propose SFL, which combines blockchain and DT.
- In SFL, NFTs distinguished by their unique design and metadata, function as a decentralized authentication method, preventing identity duplication and mitigating Sybil attacks.
- Blockchain integration in SFL ensures an unchangeable record of transactions, facilitating the identification of malicious nodes engaged in poisoning and thereby enhancing overall security.
- DTs serve as a checkpoint in FL, detecting model and data poisoning through efficiency comparison. Discrepancies trigger rollbacks, enhancing security in decentralized architectures.

We organize the paper as follows. Section II discusses the related work and motivating factors for our proposed method. Section III outlines the methodology, including blockchain initialization with DTs, NFT-based user authentication, local training protocols, smart contract-based validation, and the aggregation of model updates against benchmarks with custom algorithms. Section IV provides experimental results and performance comparisons with legacy systems. Lastly, Section V concludes with insights into the future work inspired by the outcome of the current research.

II. RELATED WORKS

The proliferation of FL across distributed systems necessitates robust security against adversarial attacks, particularly model and data poisoning, along with Sybil attacks. In addressing the vulnerabilities associated with model performance and data privacy in FL systems, various research efforts have been undertaken. For instance, a quality-based aggregation method combined with local differential privacy is proposed to preserve both model accuracy and data privacy amid potential adversarial attacks on FL, as demonstrated in [3]. A systematic analysis of secure FL applications outlined in [5] highlights the significance of countering security threats to uphold user privacy and model integrity. To combat poisoning attacks, novel defense strategies such as model analysis, byzantine robust aggregation, and verification-based approaches have been categorized and assessed for effectiveness [15].

Moreover, research has shown the susceptibility of FL models to data poisoning attacks across networked nodes, including Internet of Things (IoT) systems, emphasizing the need for vigilant defense mechanisms [16]. In the context of Zero Touch B5G Networks, the design of automated detection frameworks to identify malicious participants within the FL process stands crucial to safeguard the collaborative learning structure [17]. Additionally, the development of systems like FoolsGold offers resilience against Sybil-based poisoning by evaluating the diversity of node updates [18]. In the realm of IIoT, the first-ever Sybil-based collusion attacks have been proposed, enabling malicious participants to manipulate model aggregation through local poisoning training [19].

The author in [20] utilizes a Markov chain and a K-means clustering algorithm for effective visualization and

mitigation of Sybil attackers in challenging IoT environments. Transitioning to blockchain technology, securing FL involves integrating differential privacy and homomorphic encryption, ensuring more robust data protection in IIoT [21]. Furthermore, enhancing participant privacy while achieving verifiable aggregation results in FL is achieved through additive homomorphic encryption combined with digital signatures [22]. Lastly, the assimilation of FL with blockchain in the context of DT-empowered 6G networks addresses the need for low-latency and reliable edge association, essential for the advancement of IIoT [23].

In amalgamation with the outlined research and considering the persistence of the enumerated vulnerabilities, our work introduces a novel framework employing NFTs and DTs, integrated with blockchain technology, to provide fortified security solutions. Unlike traditional models, SFL utilizes the uniqueness of NFTs for robust user authentication, mitigating the risk of Sybil attacks. Simultaneously, the use of DTs in blockchain aids in benchmarking and efficiently rolling back to secure states in the event of detected anomalies or poisoning attempts. Our approach, thus, contributes to a secure and reliable FL environment, capable of withstanding sophisticated adversarial interferences establishing a formidable defense system for the integrity of decentralized learning.

III. SECURE FEDERATED LEARNING FRAMEWORK

Introducing a novel paradigm in the domain of IIoT, the SFL framework is specifically designed to address the unique security and traceability challenges that IIoT environments face. As IIoT ecosystems incorporate a plethora of connected devices and generate vast amounts of data, harnessing insights securely through FL becomes paramount. The proposed SFL framework unfolds through five key phases to bolster the security and integrity of FL within these complex networks.

The initialization phase sets the groundwork by deploying a central server armed with a foundational global model G. The Central Server orchestrates the FL process by distributing and aggregating the global model G among nodes. It ensures the model's integrity during aggregation before redistributing it for further iterations, which is critical for the scaffolding of an SFL system in the multifaceted and heterogeneous environment of IIoT. Nodes are the IIoT ecosystem's devices or resources that train on local data using the global model G, contributing updates back to the server. This setup enables diverse data learning while maintaining data privacy. To counteract Sybil attacks, which are particularly of concern in IIoT due to numerous nodes, the user authentication phase employs a cryptographic solution, bestowing a unique NFT to each authenticated participant. This measure not only secures the network but also introduces a level of data provenance and traceability which is indispensable in IIoT contexts.

Moving to the local training and update preparation phase, post-authentication, participating nodes undertake localized model training using their specific datasets. The resultant

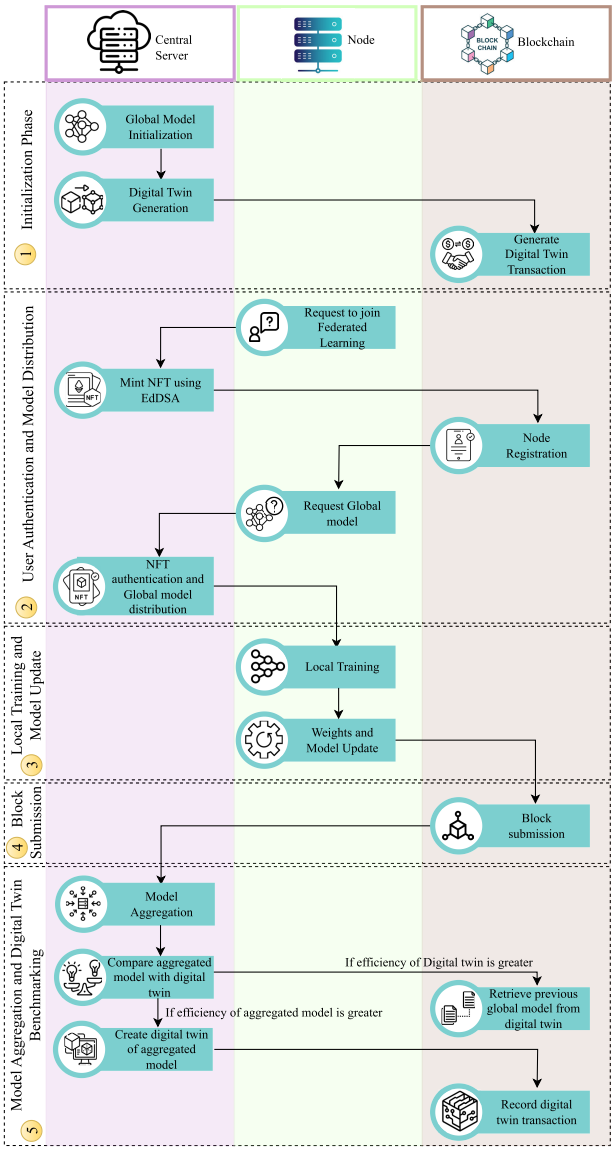

FIGURE 2. Flowchart depicting the Sequential Process of SFL.

model updates are not directly integrated; instead, in the submission of the block phase, they are transacted on the blockchain B, establishing a persistent, immutable record of model evolution. Herein lies the critical application of the blockchain's immutable property: each transaction on the blockchain, once recorded, cannot be altered or deleted. This immutability ensures that every model update, once submitted to the blockchain, becomes a permanent, unchangeable record of the model's evolution. This systematic logging not only facilitates a rigorous defense against potential data poisoning but also enables the traceability of each model modification back to its origin. The immutability of blockchain is crucial in this context as it guarantees that the integrity of the recorded model updates remains intact, thwarting any attempts to tamper with or falsify the history of model evolution. This systematic logging facilitates a

rigorous defense against potential data poisoning, enabling the traceability of each model modification to its origin. Embedded within this blockchain infrastructure are smart contracts, constituting an integral part of the submission of the block phase, which autonomously scrutinizes each model transaction. These contracts serve as an immediate filtration mechanism, identifying and isolating potential model poisoning attempts by detecting deviations from established norms.

In the final phase of model aggregation and DT benchmark, the blocks in the blockchain B are periodically invoked by the central server and aggregated with the global model G. Discrepancies in performance between the aggregated global model G and its digital counterpart signal potential data or model inconsistencies. In such scenarios, the system reverts to the pristine state represented by the DT, ensuring the consistent and reliable performance of the federated model. Through this structured approach, SFL aims to enhance the robustness and security of FL systems, addressing and mitigating prevalent vulnerabilities inherent in the domain. Fig. 2 illustrates the detailed workflow of the SFL framework, contextualizing it within the scope of IIoT.

A. INITIALIZATION PHASE

In this phase of deploying SFL, a series of systematic and crucial steps are taken to ensure the robustness and security of the system from the get-go. To commence, a centralized server is designated with the responsibility of instantiating the primary global model G. This global model G, constructed based on diverse user data and ample insights, serves as the baseline for all forthcoming collaborative training. As an immediate auxiliary step, a DT of this global model G is crafted. This DT precisely mirrors the architecture and parameters of the original global model G at the point of its inception. Contrary to real-time mirrors, this DT remains static, preserving the initial state of the model. Static DT is used so that only the endpoint of a global model G is stored, which helps to avoid the overhead and increased costs associated with constantly mirroring the whole model in a dynamic DT.

The process begins by constructing the genesis block, which acts as the cornerstone of the blockchain B. It is labeled as the first block with a unique index and is timestamped with ''creation time''. The genesis block contains no transactions, is denoted by an empty list, and is assigned an introductory proof value. Its previous hash is set to 0, signifying the initial block in the chain. Following the creation of the genesis block, a blockchain B is initiated by appending this block to an empty list, forming the foundational structure of the blockchain B.

The *CreateDigitalTwinTransaction* algorithm of this phase initiates by capturing the current timestamp and determining the version of the DT. Subsequently, it constructs a DT transaction dictionary, denoted as D, with specific keys for essential information such as the reference model, hash, location, timestamp, and version. The global model (G) serves as the reference model, while the DT hash and twin location are assigned to their respective keys. The timestamp of the transaction is set to the captured timestamp, and the version of the DT is specified as well. In essence, this algorithm systematically organizes and encapsulates relevant details into a structured dictionary, creating a DT transaction for use in a blockchain system.

This phase facilitates a reliable point of reference, especially valuable in scenarios requiring model verification or when a rollback to the original model becomes imperative.

B. USER AUTHENTICATION AND MODEL DISTRIBUTION

Following the initialization phase, the next critical phase addresses user authentication, which is a pivotal element in guaranteeing the integrity of FL processes by avoiding Sybil attacks that pose a substantial threat to FL systems. Malicious entities can overwhelm the network with numerous pseudo-identities, attempting to skew aggregated model updates or gain disproportionate influence. Additionally, our system is designed to counter data poisoning attackers, who introduce tainted or misleading data during local training sessions to compromise the integrity of the global model. Lastly, we must consider the threat posed by model poisoning attackers, who deliberately manipulate the model updates they submit in an attempt to corrupt the aggregated global model and degrade its performance

To mitigate the risk of such attacks, SFL utilizes the unique and immutable properties of NFTs to ensure that each participant in the FL network is represented by a single, authenticated entity. Our methodology incorporates EdDSA to authenticate the uniqueness of each participant's digital signature. EdDSA provides a deterministic alternative to the Schnorr signature, essential for ensuring the authenticity of the signature which will in turn set the authenticity of NFT due to its efficiency and security. This signature, along with their third-party data, is encapsulated within a block as ed-NFT and added to the blockchain B.

EdDSA relies on the specification of public parameters denoted as EdParams=(E_c, K', q, K, b, H). Within this framework, E_c represents a twisted elliptic curve, K' is indicative of an additive cycle group featuring the generator K and order q. b signifies the bit-length of confidential EdDSA scalars, $H : \{0,1\}^* \rightarrow \{0,1\}^{2b}$ serves as a hash function producing a 2b-bit output. $H_L(\cdot)$ designates the initial half of hash value as $(h_{0,\ldots,}h_{b-1})$ and $H_R(\cdot)$ corresponds to the latter half expressed as $(h_b, \ldots, h_{2b-1})$. The procedural outline of EdDSA is elucidated in the Algorithm 1.

When a third-party entity wishes to enter the FL network, SFL mints a tailored NFT (ed-NFT), as detailed in Algorithm 2, that includes specific user details u_i and a timestamp t_i captured. This process effectively prevents an individual from presenting multiple identities, which is a common scenario in Sybil attacks.

The ed-NFT serves as a means of authenticating users; once minted, it assigns a private key to them. When a node, authenticated through its ed-NFT, requests to join

Algorithm 1 EdDSA

KeyProduce(EdParams):

1: Generate a private key $pk \leftarrow \{0,1\}^b$ randomly and calculate the hash value $H(pk) = (h_0, h_1 \ldots, h_{2b-1})$
2: Set h_0, h_1, h_2, h_{b-1} as 0 and set h_{b-2} as 0. Subsequently, utilize the modified vector $(h_{0,\ldots,}h_{b-1})$ to establish a secret scalar s$\in Z_q$ i.e., s $\sum_{i=0}^{b-1} h_i \bullet 2^i mod\ q$
3: Compute the public key $bk = [s]K$.

EdDSA_Sign(pk, bk, M):

4: Calculate a random nonce, denoted as r, by applying the hash function to the latter half of $H(pk)$(which is $H_R(pk) = (h_b, \ldots, h_{2b-1})$), combined with M, and then take the result modulo q
5: Determine R by computing $[r]K$. Calculate the hashed value c as the hash of the tuple (R, bk, M) and subsequently, obtain S by computing $(r + c * s)\ mod\ q$.
6: Assemble the digital signature, which is comprised of the pair (R, S).

EdDSA Verify(bk, M, σ):

7: Calculate the hashed value c' as the hash of the tuple (R, bk, M).
8: iff $\left[2^3 \bullet S\right]K = \left[2^3\right]R + \left[2^3 \bullet c'\right]bk$ holds output 1; otherwise, output 0.

Algorithm 2 NFT-Minting With EdDSA for Third Party

Input: user details (u_i) (A set of unique details for a user)
Output: ed-NFT

1: **Function** MintNFT(u_i)
2: Capture the current timestamp (t_i).
3: Initialize a list of raw data with u_i and t_i.
4: Generate unique token id (tid) with hash function for raw data (Standard cryptographic hash, like SHA-256)
5: Construct the user metadata dictionary (M) with keys ''user details'', ''timestamp:
6: Set u_i as the value of ''user details''.
7: Set t_i as the value of ''timestamp''.
8: Generate signature (σ_i) using EdDSA_sign() algorithm for M.
9: Construct the ed-NFT with attributes ''signature'', ''user data'' and ''token id''
10: Set s_i as the value of attribute ''signature''.
11: Set M as the value of attribute ''user data''.
12: Set tid as the value of attribute ''token id''.
13: Add ed-NFT to blockchain B
14: **return** ed-NFT
15: **end function**

a training round, the central server validates the request using the participant's ed-NFT and the EdDSA verification mechanism. If blockchain B contains ed-NFT and it is not an already active ed-NFT and EdDSA_Verify(ed-NFT, σ_i) returns ''Authenticated'' as the output then the node is successfully verified. Upon successful verification, the server dispatches the global model G to the authenticated nodes. This verification process ensures that only legitimate, authenticated nodes are allowed to contribute to the local training.

C. LOCAL TRAINING AND UPDATE PREPARATION

Following the distribution of the global model G, the local training phase gets underway. In this stage, each participating node embarks on the task of training the model using its proprietary dataset according to the traditional FedAvg mechanism as follows [24].

$$F_j(w) = \sum_{i \in P_j} \frac{f_i(w)}{n_j} \tag{1}$$

where the function $f_j(w)$ signifies the loss incurred in predicting the output y_j from input x and w represents the weight vectors. The objective is to minimize this loss. The variable j denotes the total number of participants involved in the ongoing learning round, and $F_j(w)$ characterizes the local objective function specific to the j^{th} participant. Within this context, n represents the total number of samples across all participants, and n_j corresponds to the locally held samples by the j^{th} participant. Additionally, P_j, where $n_j = |P_j|$, signifies the partition assigned to the j^{th} participant from the entire dataset P.

After each node completes its localized training, the derived model updates or adjustments are consolidated. These updates represent the unique learnings garnered from individual datasets, ready to be merged into the global model G. The model updates from multiple nodes are combined into a block. This block is then prepared for submission to the smart contract for verification before being attached to the blockchain B.

D. BLOCK SUBMISSION

After local training, nodes submit model updates that undergo rigorous validation through a smart contract, serving as the first line of defense against model and data poisoning attacks. For our implementation, we employed the Ethereum blockchain, specifically leveraging its smart contract capabilities to enforce the integrity and validation processes of the submitted model updates. The choice of Ethereum is motivated by its widespread adoption, robust smart contract functionality, and its ability to integrate with existing IIoT platforms. This automated process ensures adherence to predefined criteria, swiftly rejecting non-compliant or potentially malicious updates. This proactive measure prevents unnecessary overhead, as it mitigates the need to advance to the stage where a DT is employed for recovery.

Upon successful validation by the smart contract, the model update is encapsulated into a transaction T using the *CreateModelTransaction* algorithm, which takes the sender's information, model hash, the function $F_j(w)$, and the number of data points n_j as inputs to generate a model transaction. The

process begins by capturing the current timestamp, denoted as t_i. It then constructs a model transaction dictionary, T, with keys including sender, datapoints, weight, and hash. The sender's identity is set as the value of the sender, the number of data points as the value of data points, the function $F_j(w)$ as the value of the weight, and the model hash as the value of the hash. The timestamp t_i is assigned to the time key in the dictionary. Ultimately, the algorithm returns the model transaction dictionary T with organized information necessary for recording model-related transactions. This transaction is then methodically prepared and positioned for addition to the blockchain B, symbolizing the technical completion of the validation and submission process.

Simultaneously, this architecture the integrity of blockchain B as outlined in Algorithm 3. This mechanism, designed to maintain blockchain B's consistency, employs a custom hashing function and checks every block's integrity against its predecessor. By comparing digests and credentials, it ensures that the blockchain B remains tamper-proof. Any inconsistency immediately flags the block as INVALID, while consistent chains are marked VALID, ensuring the trustworthiness of the entire blockchain B.

Algorithm 3 Chain Integrity Checking

Input: active block, Blockchain B
Output: Bool (VALID if blockchain B is consistent, INVALID otherwise)

Function GenerateDigest(record)
Get the Record comprising Timestamp, preceding digest, random value, and details (covering updates from local trainers and DT logs)
digest output = CustomHashFunction(record)
return digest output
end function
Set base block to B's genesis block
Set block tracker = 1
while Block tracker is below the total blocks in B **do**
    active block = B [block tracker]
    if active block.preceding digest != GenerateDigest(B [block tracker - 1]) **then**
        return INVALID
    end if
    prior verification = base block.credential
    recent verification = active block.credential
    hash mechanism = CustomHashFunction(recent verification – prior verification)
    if leading segment of the hash mechanism isn't according to the predetermined pattern **then**
        return INVALID
    end if
    base block = active block
    block tracker = block tracker + 1
end while
return VALID

E. MODEL AGGREGATION AND DIGITAL TWIN BENCHMARKING

Periodically, the global model G initiates a block extraction procedure from the blockchain B. Each block embodies a series of transactions (or) model updates, which is the result of localized training computations. Through the traditional FedAvg algorithm [24] equation, these updates are methodically consolidated, updating the global model G's parameters $f\ (w)$ to represent an integrated dataset while ensuring data privacy as follows.

$$f\ (w) = \sum_{j=1}^{J} \frac{n_j}{n} F_j(w) \tag{2}$$

After the aggregation, an evaluation procedure is triggered as illustrated in Algorithm 4 wherein the performance metrics, specifically accuracy and loss, of the refined global model G are compared with those of the pre-existing DT. We focus on accuracy and loss metrics as primary performance parameters due to their critical importance in assessing the efficacy and reliability of machine learning models, especially in federated learning contexts. Accuracy provides a straightforward measure of the model's predictive capability, directly impacting its practical utility in real-world applications. Loss metrics offer insights into the model's learning process, highlighting how well the model generalizes from the training data. This comparison serves not only as a quality check but also as a safeguard against anomalies. While we understand federated learning's inherently non-deterministic nature, significant deviations in performance metrics, especially when they fall below established thresholds without clear non-malicious explanations, raise concerns that warrant further investigation. The DT, a static representation of a previous optimal model state, functions as a benchmark. Any deviations in performance are analytically scrutinized.

Should the aggregated model's metrics ϕ be commensurate with or superior to the DT's ϕ_i metrics, the model is deemed fit for adoption as the current global model G. In this scenario, a new DT is instantiated, mirroring this updated model state.

Correspondingly, a blockchain B entry is recorded, affording future traceability and auditing capabilities. Conversely, a significant and unexplained decline in performance metrics prompts a closer review. This cautious approach recognizes the possibility of legitimate variability in model performance due to the diverse nature of local datasets while remaining vigilant against potential security threats. In addressing suspected poisoning, the immutability of the blockchain B proves instrumental. As every model update is chronologically and securely logged, anomalous entries indicative of malicious intent can be identified and isolated.

To counteract the effects of potential poisoning, the system defaults to the DT, leveraging its preserved state to counteract the detrimental effects of the poisoned updates.

Algorithm 4 Benchmarking Aggregated Efficiency Against Global Model

Input: ϕ_i : Efficiency of the central (global) model
ϕ: Aggregated efficiency computed from local models
Output: Decision (Update or No Update) based on the comparison

Procedure BenchmarkAndUpdate (ϕ_i, Φ):
if $\Phi > \phi_i$ **then**
    $\phi_i = \Phi$ // Update the global model 's efficiency
    The DT of the least efficient model is discarded
    DT of the updated model is stored and a transaction is stored in blockchain B
    return ''Global model Efficiency Updated''
else
    The aggregated model is discarded and the DT of the last model is Instantiated as the global model G.
end if
return ''Review transactions for potential anomalies, including indications of model or data tampering.''
end procedure

Thus SFL maintains traceability and emphasizes resilience against malicious attempts. It establishes a robust framework for maintaining collaborative learning integrity.

F. ATTACKER MODEL AND SECURITY ANALYSIS FOR SFL IN IIOT

In the domain of IIoT, the SFL framework is meticulously designed to counteract a spectrum of sophisticated cyber threats that uniquely challenge the security and integrity of FL systems. Understanding the attacker model is paramount for evaluating the resilience of the SFL framework against potential security vulnerabilities.

Threat Actors and Capabilities:

- **Sybil Attackers:** Individuals or entities capable of creating multiple fake identities (Sybil attacks) to disrupt the FL process. These attackers aim to infiltrate the network with pseudoidentities, potentially skewing aggregated model updates or gaining undue influence over the learning process.
- **Data Poisoning Attackers:** Malicious participants or external adversaries may introduce tainted or misleading data during local training, aiming to compromise the global model's integrity.
- **Model Poisoning Attackers:** Attackers who manipulate the model updates they submit, attempting to corrupt the aggregated global model to degrade its performance or introduce backdoors.

Attacker Objectives:

- **Disruption of Learning Process:** To degrade the performance or reliability of the FL system, making it ineffective or untrustworthy.
- **Compromise of Data Integrity:** To inject false data or model updates, aiming to skew the learning outcomes to the attacker's advantage.
- **System Infiltration:** To gain unauthorized access or influence within the FL network for long-term exploitation or sabotage.

Attack Vectors and Methods:

- **Sybil Attacks through Fake Identities:** Leveraging the creation of numerous pseudoidentities to gain influence or disrupt the aggregation process.
- **Data Poisoning via Tainted Local Updates:** Submitting manipulated data or model updates during local training phases to corrupt the global model.
- **Model Poisoning through Malicious Updates:** Introducing subtle but malicious alterations to model updates aimed at degrading the global model's accuracy or introducing vulnerabilities.

The SFL framework incorporates several defense mechanisms across its phases to address these threats directly. During the Initialization Phase, the creation of a static DT and the use of a secure blockchain for the genesis block construction lay the groundwork for a traceable and tamper-evident model lineage. The User Authentication and Model Distribution Phase introduces unique and immutable NFT-based authentication to combat Sybil attacks and ensure that each participant in the FL network is a single, authenticated entity. In the Local Training and Update Preparation Phase, localized model training incorporates mechanisms to detect and mitigate data poisoning attempts. The Block Submission Phase utilizes smart contracts for rigorous validation of model updates, serving as a robust line of defense against model poisoning attacks. Finally, the Model Aggregation and DT Benchmarking Phase leverages the immutable ledger of blockchain to verify the integrity of model updates, ensuring the system can revert to a known good state in the event of detected inconsistencies.

IV. RESULTS

To evaluate the performance of SFL in an IIoT ecosystem, the simulations were executed on a custom server equipped with an AMD Ryzen 9 5900X CPU, 32 GB DDR4 RAM, paired with a 1 TB NVMe SSD and 4 TB HDD. The graphics processing was handled by an NVIDIA GeForce RTX 3080 GPU. Throughout the simulations,[1] the system's RAM usage peaked at 24 GB during the most intensive phases of model training and aggregation, demonstrating efficient resource utilization given the dataset's size and complexity. The processing time for each epoch averaged 2 minutes, underscoring the SFL framework's computational efficiency.

For our dataset of choice, we used MNIST [25], with the training involving 30 independent nodes. Unlike previous approaches that often overlook the distribution variability's impact on model performance, we meticulously divided the dataset into 30 equal parts, ensuring each node received a

[1] https://github.com/YeshwanthGovindarajan/SFL_Model

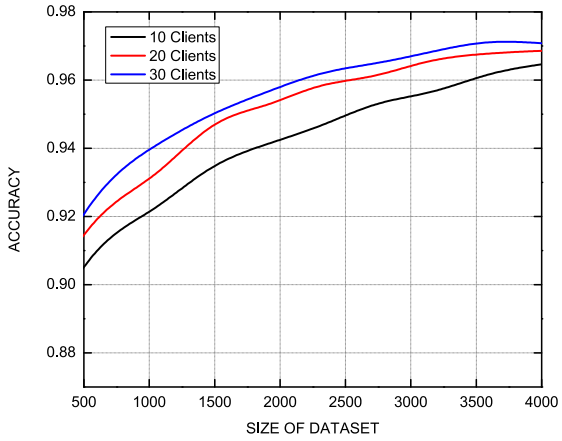

FIGURE 3. The relationship between accuracy, and individual dataset sizes for different numbers of nodes.

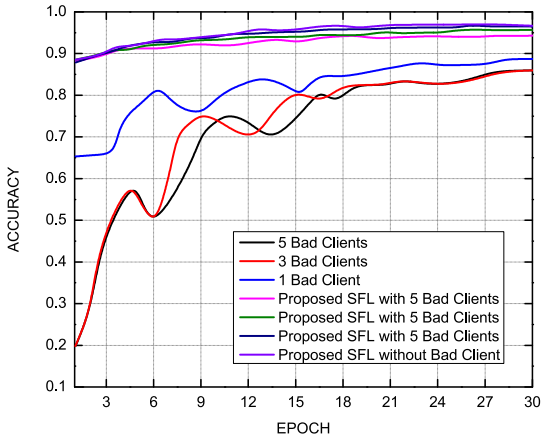

FIGURE 4. Performance of SFL under adversarial conditions.

distinct subset, a method demonstrating our commitment to evaluating FL under realistic conditions. The training was programmed to run for 25 epochs at a fixed learning rate of 0.01. The model updates, while maintaining user privacy, were periodically aggregated from the different nodes throughout the experiment. The aggregation process was observed to consume less than 1 GB of RAM, highlighting the SFL's optimization in handling data across the network.

In assessing the impact of a node's dataset size on the accuracy of the model, our simulations demonstrate that the variance of dataset sizes among nodes, ranging from 500 to 4000, marginally influences the overall model precision. This finding contrasts with common assumptions in the existing literature that larger dataset sizes at individual nodes significantly enhance model accuracy, underscoring the efficiency of our SFL system in handling diverse data sizes. As displayed in Fig. 3, varying the number of participating nodes per epoch specifically, 10, 20, or 30 out of a total of 30 reveals that the model's accuracy, which spans from 88% to 97%, is largely stable regardless of the number of nodes included in each epoch. Notably, the accuracy figures when considering either 20 or 30 nodes are quite similar, underscoring SFL's proficiency in maintaining high accuracy without the necessity of encompassing an extensive number of nodes. This stability, even with fewer nodes, sets our SFL system apart from traditional FL systems that exhibit marked accuracy decreases with reduced node participation. Furthermore, an increase in the nodes' dataset size correlates positively with enhancements in model accuracy.

In Fig. 4 we assessed the robustness of the SFL model against adversarial attacks, specifically focusing on threats posed by Sybil attackers, data poisoning attackers, and model poisoning attackers, as outlined in our attacker model. We compared its performance with that of a standard FL system under varying conditions. Three distinct scenarios were considered to demonstrate the impact of malicious nodes on system accuracy. Scenario 1 served as our baseline, representing an SFL system with no malicious nodes, which achieved a high benchmark accuracy of 97%.This baseline performance itself showcases the superior initial accuracy of our system. In Scenarios 2 and 3, we introduced adversarial elements into the standard FL framework and SFL to observe their effect on model performance. In Scenario 2, a conventional FL system was challenged by the presence of 1, 3, and 5 malicious nodes. The negligible accuracy decrease in our SFL model under adversarial conditions starkly illustrates its robustness, a testament to the effectiveness of our security protocols which are absent in conventional FL systems. The simulation of adversarial conditions revealed that the SFL framework's defensive mechanisms did not significantly affect the system's performance or resource usage, with only a nominal increase in processing time (less than 10% across scenarios) and no additional RAM usage. The accuracy of this non-secure FL system was observed to decrease significantly as the number of malicious nodes increased. Specifically, the accuracy dropped to 87% in the presence of 1 malicious node, 85% with 3 malicious nodes, and fell to 81% when 5 malicious nodes were introduced. These results indicated a clear inverse relationship between the number of malicious nodes and the accuracy of a standard FL model. In contrast, Scenario 3 showcased the performance of our proposed SFL framework under the same adversarial conditions. Remarkably, the SFL model displayed a consistent accuracy, with only a negligible decrease of 0.05% from the baseline, despite the inclusion of up to 5 malicious nodes. This performance stability, nearly paralleling the ideal unassailed scenario, highlights the robustness and resilience of our SFL model in adverse environments. These experimental findings underscore the effectiveness of our SFL system in maintaining data integrity and accuracy even when faced with sophisticated adversarial attacks, thereby establishing the SFL as a viable solution for enhancing the security of FL networks.

In terms of scalability, the performance of SFL remained consistent across different node counts, as illustrated in Fig. 5. The graph depicts four distinct lines corresponding to scenarios with a random number of nodes, 10 nodes,

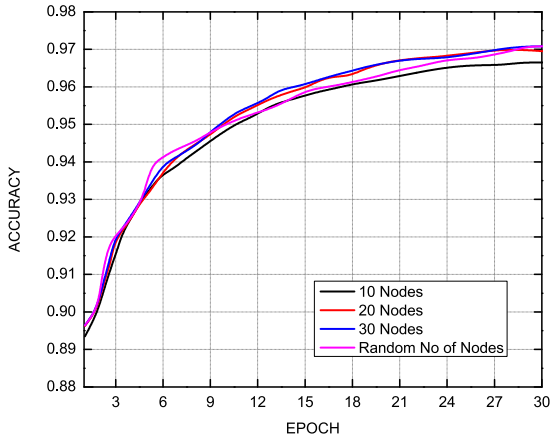

FIGURE 5. The relationship between accuracy, and epoch for different numbers of nodes.

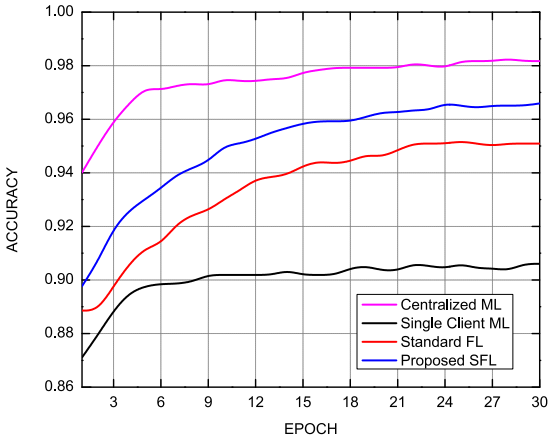

FIGURE 7. The relationship between accuracy and number of epochs of various learning models.

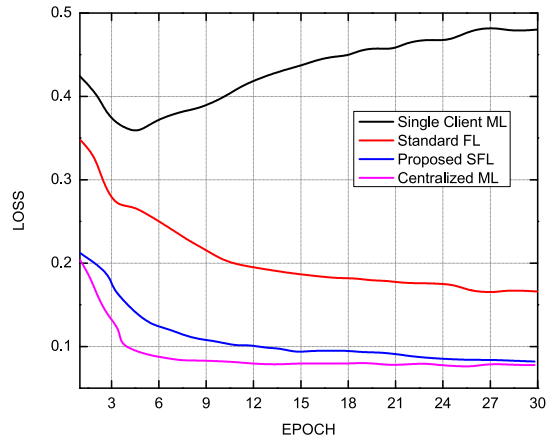

FIGURE 6. The relationship between loss and number of epochs of various learning models.

20 nodes, and 30 nodes. It is noteworthy that the accuracy across the 4 scenarios held steady and did not show any signs of degradation, even with a growing number of nodes. In particular, the performance under a random number of nodes showcased a robust efficiency, highlighting the adaptability of the SFL system. The trajectories for the 20 and 30-node scenarios are indistinguishable, indicating that the model upholds its efficacy and learning capability regardless of the number of nodes involved, a direct outcome of our novel aggregation and synchronization protocols. This underpins the scalability of the SFL framework, ensuring it delivers consistent performance in various node configurations.

The scalability tests, depicted in Fig. 5, further emphasize the SFL's efficient use of computational resources. Despite increasing the number of nodes from 10 to 30, the system's RAM usage and processing times remained largely unaffected, illustrating the SFL framework's capability to scale without proportional increases in resource demand.

Fig. 6 and Fig. 7, provide substantial insights through the quantitative analysis of model accuracy and loss metrics over successive epochs. In the single-node ML scenario, an accuracy of 0.90 is observed, which is likely due to limited data diversity exposure. This is considerably lower than the standard FL approach, which leverages a more expansive dataset and obtains an accuracy of 0.96 after 30 epochs.

SFL framework demonstrates a significant improvement in accuracy, reaching 0.97, which closely approaches the centralized ML benchmark of 0.98. This underscores the effectiveness of our security protocols, illustrating that they not only preserve data privacy and integrity but also enable a learning efficiency that closely mirrors that of centralized ML systems.

We compare loss trajectories across 20 epochs for four training scenarios in Fig. 7. The single-node ML model exhibits an increase in loss after 5 epochs, indicating limitations in leveraging diverse data sources. Standard FL shows a moderate loss decrease, stabilizing at 0.14 with collaborative benefits. The centralized model serves as a baseline with the lowest final loss of 0.07. Notably, the SFL framework closely mirrors the convergence of the centralized model, achieving a final loss of 0.08. Our findings, particularly the high accuracy and low loss achieved by the SFL framework, present clear evidence of its superiority over both traditional FL and single-node ML models. This not only confirms the effectiveness of our security and privacy-enhancing protocols but also establishes SFL as a benchmark in FL efficiency.

To comprehensively evaluate the efficiency of our SFL framework, we included an analysis of computational overhead alongside accuracy and loss metrics. Fig. 8 illustrates the computational overhead across four different models: Centralized ML, Single-client ML, Standard FL, and our Proposed SFL approach, over ten training iterations. Our Proposed SFL demonstrates a slightly higher computational overhead than Standard FL, attributable to the enhanced security protocols it employs. Specifically, while Standard FL exhibits a modest increase in CPU time, from 1.2 to 1.5 seconds per iteration, our SFL approach starts at 1.3 seconds and climbs to 1.7 seconds by the 10th iteration. Centralized ML remains the most efficient, given its unified computational environment, whereas Single-client ML incurs

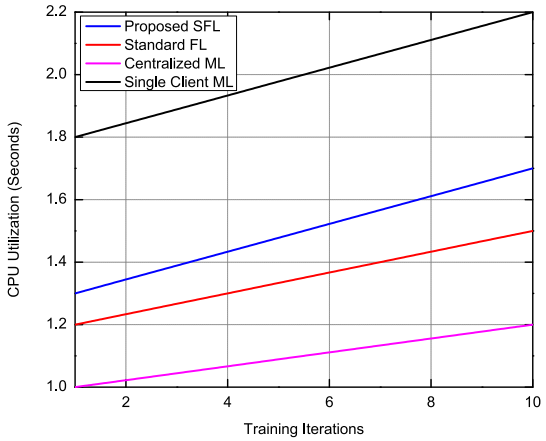

FIGURE 8. Computational Overhead Across Training Iterations for Different ML Approaches.

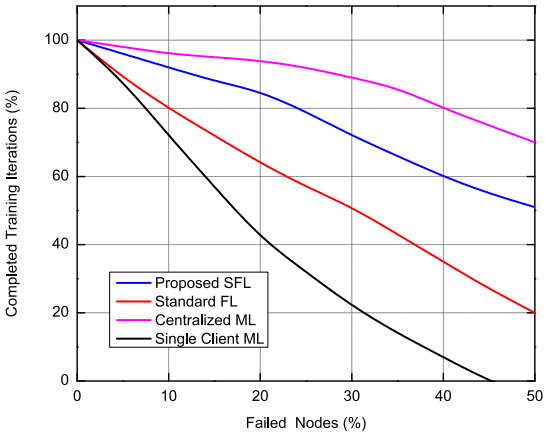

FIGURE 9. Fautl Tolerance Comparison.

the highest overhead due to processing all tasks on a single node.

This increase in computational time is a considered trade-off for the significant improvements in security and robustness that our SFL framework offers. The data suggests that while SFL incurs a small efficiency cost, it considerably strengthens the system's resilience against adversarial threats, as demonstrated by its consistent accuracy under attack scenarios, previously shown in Fig. 4. Such a trade-off is essential for practical IIoT applications where security cannot be compromised. The scalability tests further support the viability of our framework, showing that even with the slight increase in computational overhead, the system maintains consistent performance as the number of nodes scales, a testament to the SFL's design efficiency.

Our evaluation of the SFL framework's performance includes an examination of fault tolerance, an essential characteristic of IIoT systems. Fig. 9 plots the fault tolerance of the SFL framework against three other models: Centralized ML, Single-client ML, and Standard FL. Fault tolerance is measured by the ability to complete training iterations in the presence of node failures. The proposed SFL framework shows a slight decrease in the percentage of completed training iterations as the number of failed nodes increases, starting at 95% completion without failures and maintaining above 60% even with 50% node failure. This performance is notably better than that of the Standard FL, which starts at a similar completion rate but drops more rapidly, highlighting the SFL's enhanced resilience. As expected, Centralized ML shows minimal impact from node failures, sustaining near-complete training iterations, While Single-client ML is significantly affected, with a sharp decrease in completion rate as node failures rise. This resilience of the SFL framework can be attributed to its design features that collectively provide robustness against node failures, ensuring that learning can proceed with minimal disruption, a vital attribute for stable IIoT operations.

A. STRATEGIC TRADE-OFFS: BALANCING SECURITY, SCALABILITY, AND PRIVACY IN SFL DEPLOYMENT

In the proposed SFL system, the acceptance of certain limitations—centralization in the initialization phase, reliance on blockchain technology, and inherent challenges in ensuring complete model and data privacy—serves as strategic trade-offs, each carefully weighed to harness specific advantages vital to the system's overall design and objectives. Centralization, while potentially limiting scalability, ensures a secure and controlled deployment of the initial global model, crucial for establishing a trusted foundation. The system's dependence on blockchain, despite introducing computational overhead, is justified by the immutable and transparent nature of blockchain transactions, enhancing security and user authentication. This approach prioritizes integrity and trust, fundamental in a FL context. Finally, while absolute privacy remains challenging, the use of DTs and NFT-based authentication strategically balances the need for system integrity against privacy concerns, providing a robust mechanism for identity verification and model validation. Together, these trade-offs are consciously made to optimize the system's security, scalability, and privacy, reflecting a nuanced understanding of the practical implications and technological constraints inherent in deploying an SFL framework.

Despite the inherently distributed nature of FL, the comparable performance confirmed our security integrations' efficacy, thus establishing the SFL framework as a prominent contender that surpasses standard configurations and secures the model's proficiency against adversarial activities.

While the theoretical foundations of our proposed framework are well-established, its practical implementation in industrial settings may face challenges such as computational overhead and cost-intensive processes. Careful deployment and monitoring of DTs are essential to prevent excessive creation, which could exacerbate computational burdens. However, the SFL framework offers robust security measures and leverages decentralized approaches for parallel computing, mitigating these challenges. With ongoing research and technical advancements, the computational overhead

will be reduced, making the SFL framework increasingly feasible. Ultimately, the exceptional security provided by SFL suggests that, despite current limitations, continued development could outweigh computational costs, paving the way for widespread adoption in industrial applications.

V. CONCLUSION

This research paper presents an SFL framework that significantly enhances the security and robustness of distributed ML systems within the IIoT landscape. In the current IIoT ecosystem, achieving high data security and maintaining operational traceability are paramount, yet traditional FL approaches often fall short in these areas. To address these challenges, our research introduces an SFL framework, which incorporates DTs with blockchain technology to enhance security measures. Additionally, we utilize NFTs as a novel mechanism for authentication and update validation within the distributed network. This integration has not only mitigated the risks associated with adversarial attacks in IIoT environments but has also led to substantial improvements in model performance. Our SFL architecture achieves an accuracy of 97% and a loss of just 0.08, rivaling centralized ML systems and outperforming standard FL models. These results are particularly pertinent for IIoT settings where security, reliability, and traceability are crucial. Moving forward, our focus will be on optimizing the transactional efficiency associated with DTs to further enhance the SFL framework for IIoT applications.

REFERENCES

[1] S. Moon and W. Hee Lee, "Privacy-preserving federated learning in healthcare," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Feb. 2023, pp. 1–4.

[2] K. S. Arikumar, S. B. Prathiba, M. Alazab, T. R. Gadekallu, S. Pandya, J. M. Khan, and R. S. Moorthy, "FL-PMI: Federated learning-based person movement identification through wearable devices in smart healthcare systems," *Sensors*, vol. 22, no. 4, p. 1377, Feb. 2022.

[3] Y. Gao, L. Zhang, L. Wang, K. R. Choo, and R. Zhang, "Privacy-preserving and reliable decentralized federated learning," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 1–13, Jul. 2023.

[4] A. K. S. B. Prathiba, R. S. Moorthy, G. Srivastava, and T. R. Gadekallu, "Software defined networking assisted electric vehicle charging: Towards smart charge scheduling and management," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 1, pp. 163–173, Jan./Feb. 2024.

[5] H. N. C. Neto, J. Hribar, I. Dusparic, D. M. F. Mattos, and N. C. Fernandes, "A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends," *IEEE Access*, vol. 11, pp. 41928–41953, 2023.

[6] J. Yu, L. Lu, Y. Chen, Y. Zhu, and L. Kong, "An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 337–351, Feb. 2021.

[7] Y. Zhang, Y. Zhang, Z. Zhang, H. Bai, T. Zhong, and M. Song, "Evaluation of data poisoning attacks on federated learning-based network intrusion detection system," in *Proc. IEEE 24th Int. Conf. High Perform. Comput. Commun.*, Dec. 2022, pp. 2235–2242.

[8] J. Yang, K. Yang, Z. Xiao, H. Jiang, S. Xu, and S. Dustdar, "Improving commute experience for private car users via blockchain-enabled multitask learning," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 21656–21669, Dec. 2023.

[9] J. Shen, H. Sheng, S. Wang, R. Cong, D. Yang, and Y. Zhang, "Blockchain-based distributed multiagent reinforcement learning for collaborative multiobject tracking framework," *IEEE Trans. Comput.*, vol. 73, no. 3, pp. 778–788, Mar. 2024.

[10] Y. Liu, Z. Fang, M. H. Cheung, W. Cai, and J. Huang, "Mechanisms design for blockchain storage sustainability," *IEEE Commun. Mag.*, vol. 61, no. 8, pp. 102–107, Aug. 2023.

[11] B. Cao, X. Wang, W. Zhang, H. Song, and Z. Lv, "A many-objective optimization model of industrial Internet of Things based on private blockchain," *IEEE Netw.*, vol. 34, no. 5, pp. 78–83, Sep. 2020.

[12] M. S. Rodrigo, D. Rivera, J. I. Moreno, M. Àlvarez-Campana, and D. R. López, "Digital twins for 5G networks: A modeling and deployment methodology," *IEEE Access*, vol. 11, pp. 38112–38126, 2023.

[13] S. B. Prathiba, G. Raja, S. Anbalagan, S. Gurumoorthy, N. Kumar, and M. Guizani, "Cybertwin-driven federated learning based personalized service provision for 6G-V2X," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4632–4641, May 2022.

[14] J. Wu, J. Zhu, J. Zhang, P. Dang, W. Li, Y. Guo, L. Fu, J. Lai, J. You, Y. Xie, and C. Liang, "A dynamic holographic modelling method of digital twin scenes for bridge construction," *Int. J. Digit. Earth*, vol. 16, no. 1, pp. 2404–2425, Oct. 2023.

[15] G. Xia, J. Chen, C. Yu, and J. Ma, "Poisoning attacks in federated learning: A survey," *IEEE Access*, vol. 11, pp. 10708–10722, 2023.

[16] G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, and J. Liu, "Data poisoning attacks on federated machine learning," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11365–11375, Jul. 2022.

[17] S. Ben Saad, B. Brik, and A. Ksentini, "Toward securing federated learning against poisoning attacks in zero touch B5G networks," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 2, pp. 1612–1624, Jun. 2023.

[18] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating Sybils in federated learning poisoning," 2020, *arXiv:1808.04866*.

[19] X. Xiao, Z. Tang, C. Li, B. Xiao, and K. Li, "SCA: Sybil-based collusion attacks of IIoT data poisoning in federated learning," *IEEE Trans. Ind. Informat.*, vol. 19, no. 3, pp. 2608–2618, Mar. 2023.

[20] A. K. Mishra, A. K. Tripathy, D. Puthal, and L. T. Yang, "Analytical model for Sybil attack phases in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 379–387, Feb. 2019.

[21] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, and Y. Liang, "Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4049–4058, Jun. 2022.

[22] Y. Ren, Y. Li, G. Feng, and X. Zhang, "Privacy-enhanced and verification-traceable aggregation for federated learning," *IEEE Internet Things J.*, vol. 9, no. 24, pp. 24933–24948, Dec. 2022.

[23] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5098–5107, Jul. 2021.

[24] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[25] L. Deng, "The MNIST database of handwritten digit images for machine learning research [Best of the Web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

SAHAYA BENI PRATHIBA (Member, IEEE) received the bachelor's and master's degrees in computer science and engineering and the Ph.D. degree from Anna University, Chennai, in 2022. She has secured 23rd rank among 2581 candidates in Master of Engineering. She is currently an Assistant Professor with the Centre for Cyber Physical Systems, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai. Her research interests include 5G/6G, vehicle-to-everything, software defined networking, autonomous vehicular networks, industry 5.0, and metaverse. She was a recipient of the Anna Centenary Research Fellowship. She is serving as a reviewer for IEEE TRANSACTION and reputed Elsevier journals.

YESHWANTH GOVINDARAJAN is currently pursuing the B.Tech. degree in computer science engineering with VIT Chennai. He is also interning and gaining practical experience at Highperformr.ai, he actively contributes to the application of these technologies in real-world scenarios. His research interests include artificial intelligence, machine learning, and neural networks.

VISHAL PRANAV AMIRTHA GANESAN is currently pursuing the B.Tech. degree in computer science engineering with VIT University. He is also interning at Highperformr.ai, he is passionate about applying these technologies to solve real-world challenges and actively contributing to innovative projects. His research interests include artificial intelligence, machine learning, and neural networks.

ANIRUDH RAMACHANDRAN is currently pursuing the B.Tech. degree in computer science engineering with VIT Chennai. He is also interning and gaining practical experience with the Department of IT and Backend Development, TTK Ltd. His research interests include artificial intelligence, database management, and machine learning.

ARIKUMAR K. SELVARAJ received the bachelor's and master's degrees in computer science and engineering and the Ph.D. degree from Anna University, Chennai, in 2022. He is currently an Assistant Professor with the Department of Data Science and Business Systems, SRM Institute of Science and Technology (SRMIST), Kattankulathur. He has published 25 conferences and journal papers. He has 12 years of experience. His current research interests include wireless sensor networks, machine learning, software defined networking, autonomous vehicular networks, and the IoT.

ALI KASHIF BASHIR (Senior Member, IEEE) is currently a Reader with the Department of Computing and Mathematics, Manchester Metropolitan University, U.K. He is also the Leader of the Future Networks Laboratory and the Head of the Advanced Cybersecurity Testbed. He is supervising/co-supervising many Ph.D.'s, postdoctorals, and research associates. He also enjoys several honorary and adjunct professorship in many countries, such as China, Canada, Lebanon, United Arab Emirates, India, and Pakistan. Along with his students and colleagues, he has published over 250 high-impact articles in top venues. He has obtained over e4 million in external funding from U.K., South Korean, Japanese, European, Asian, and Middle Eastern agencies. He is also a Co-I of GM AI and GM Cyber Foundry, each having e6 million in funding. He is a member of more than ten IEEE technical societies and a Distinguished Speaker of ACM. He has chaired several international conferences and workshops and has delivered over 40 invited and keynote talks across the globe. He is serving as the Editor-in-Chief for IEEE TECHNOLOGY POLICY AND ETHICS and *Journal of Autonomous Intelligence* and an Editor for over ten international journals, including *Scientific Reports*, *Nature*, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.

THIPPA REDDY GADEKALLU (Senior Member, IEEE) received the bachelor's degree in computer science and engineering from Acharya Nagarjuna University, India, in 2003, the master's degree in computer science and engineering from Anna University, Chennai, Tamil Nadu, India, in 2011, and the Ph.D. degree from Vellore Institute of Technology, Vellore, Tamil Nadu, in 2017. He is currently working at Lovely Professional University, India. He has more than 14 years of experience in teaching. He has more than 200 international/national publications in reputed journals and conferences. His current research interests include machine learning, the Internet of Things, deep neural networks, blockchain, and computer vision. He was recently recognized as one of the top 2% scientists in the world as per the survey conducted by Elsevier, in 2021 and 2022. He is an Editor of several publishers, such as Springer, *PLOS One, Scientific Reports, Nature*, and Wiley. He also acted as the Guest Editor in several reputed publishers, such as IEEE, Elsevier, Springer, and MDPI.