mathematics

MDPI

Article

Enhancing the Security and Privacy in the IoT Supply Chain Using Blockchain and Federated Learning with Trusted Execution Environment

Linkai Zhu [1,†], Shanwen Hu [2,*,†], Xiaolian Zhu [1], Changpu Meng [3,4] and Maoyi Huang [5]

[1] Information Technology School, Hebei University of Economics and Business, Shijiazhuang 050061, China; linkai@hueb.edu.cn (L.Z.); xiaolianzhu@hueb.edu.cn (X.Z.)
[2] Faculty of Data Science, City University of Macau, Macau 999078, China
[3] iFLYTEK Co., Ltd., Hefei 230088, China; cpmeng@iflytek.com
[4] School of Computing and Information Technology, Faculty of Engineering and Information Sciences, University of Wollongong, Wollongong 2522, Australia
[5] Product Development, Ericsson, 41756 Gothenburg, Sweden; maoyi.huang@ericsson.com
* Correspondence: d21091100159@cityu.mo
† These authors contributed equally to this work.

Abstract: Federated learning has emerged as a promising technique for the Internet of Things (IoT) in various domains, including supply chain management. It enables IoT devices to collaboratively learn without exposing their raw data, ensuring data privacy. However, federated learning faces the threats of local data tampering and upload process attacks. This paper proposes an innovative framework that leverages Trusted Execution Environment (TEE) and blockchain technology to address the data security and privacy challenges in federated learning for IoT supply chain management. Our framework achieves the security of local data computation and the tampering resistance of data update uploads using TEE and the blockchain. We adopt Intel Software Guard Extensions (SGXs) as the specific implementation of TEE, which can guarantee the secure execution of local models on SGX-enabled processors. We also use consortium blockchain technology to build a verification network and consensus mechanism, ensuring the security and tamper resistance of the data upload and aggregation process. Finally, each cluster can obtain the aggregated parameters from the blockchain. To evaluate the performance of our proposed framework, we conducted several experiments with different numbers of participants and different datasets and validated the effectiveness of our scheme. We tested the final global model obtained from federated training on a test dataset and found that increasing both the number of iterations and the number of participants improves its accuracy. For instance, it reaches 94% accuracy with one participant and five iterations and 98.5% accuracy with ten participants and thirty iterations.

Keywords: federated learning; Trusted Execution Environment (TEE); blockchain; supply chain; Internet of Things (IoT)

MSC: 68M25

Citation: Zhu, L.; Hu, S.; Zhu, X.; Meng, C.; Huang, M. Enhancing the Security and Privacy in the IoT Supply Chain Using Blockchain and Federated Learning with Trusted Execution Environment. *Mathematics* **2023**, *11*, 3759. https://doi.org/10.3390/math11173759

Academic Editor: Cheng-Chi Lee

Received: 18 July 2023
Revised: 24 August 2023
Accepted: 29 August 2023
Published: 1 September 2023

1. Introduction

The exponential growth of Internet of Things (IoT) applications in supply chain management has introduced both opportunities and challenges. IoT devices play a crucial role in enabling intelligent control, automated operations, optimized scheduling, quality testing, and efficient delivery within production lines [1]. They also facilitate the intelligent positioning, monitoring, and management of transportation vehicles, distribution centers, and warehouses while providing real-time tracking and traceability of goods [2]. In the supply chain, enterprises rely heavily on suppliers for diverse raw materials or services necessary for manufacturing their products. IoT devices assist in the smart selection,

management, and evaluation of suppliers, as well as the real-time monitoring of raw material quality, quantity, and location [3]. Although generating substantial data, the need to share and coordinate data among participants in real time arises to enhance efficiency and reduce costs. However, concerns surrounding data leakage, potential misuse, and exposure to competitors pose significant challenges concerning data privacy and trust. Moreover, the process of data sharing in the supply chain complicates the establishment of the consensus on data value and quality, as sharing parties may intentionally provide low-quality or even falsified data to benefit themselves [4]. Hence, addressing the challenge of achieving data collaboration and intelligent analysis while ensuring data security and privacy within the supply chain is of utmost importance.

Federated learning is a distributed machine learning technique that enables multiple participants to train models locally and share model parameters or updates with a central server for aggregation [5]. This approach allows each participant to improve their model's performance by leveraging data from other participants without directly sharing their raw data [6]. Federated learning offers advantages in terms of data privacy protection and reduced communication overhead, making it well suited for supply-chain-management scenarios [7–9]. Nevertheless, existing federated learning techniques still encounter security and privacy challenges. For instance, the central server could be vulnerable to hacking or tampering by internal personnel, resulting in the leakage or corruption of model parameters or updates [10,11]. Dishonest or malicious behavior among participants, such as transmitting erroneous or malicious model parameters or updates, can significantly impact the quality of the models [12]. Additionally, participants' join or exit events may lead to an unstable or inconsistent model training process [13]. To address these issues, robust and reliable security and privacy-protection mechanisms need to be introduced.

Figure 1 illustrates a typical cross-border data-sharing use case within a global supply chain, where various IoT devices' data are stored in the cloud, accessible from different regions or countries involved in the supply chain. However, this scenario exposes sensitive data to potential harm from malicious users. This paper presents a novel federated learning framework that leverages the blockchain and the Trusted Execution Environment (TEE) to enhance the security and privacy of supply chain data. Our contributions are as follows.

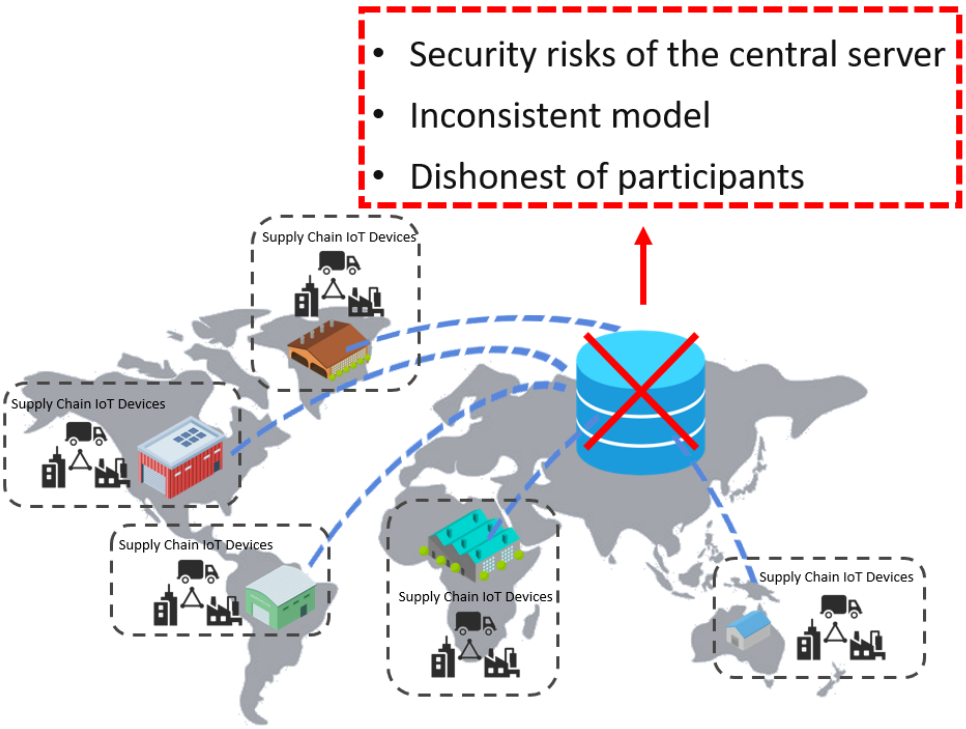


Figure 1. Possible challenges of traditional centralized global supply chain scenario.

(1) The framework introduces a consortium blockchain-based federated learning protocol that capitalizes on the decentralized, tamper-resistant, and traceable characteristics of blockchain technology to enable secure transmission, storage, and verification of model parameters or updates.

(2) We utilize TEE to safeguard local participant data and computation processes from leakage or tampering. We introduce a distributed local federated learning architecture based on TEE to further enhance data privacy while ensuring the quality of local client models.

(3) We conduct experimental evaluations using datasets, and the results demonstrate that our framework effectively enhances data security and privacy while achieving high model accuracy and maintaining low communication overhead.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 introduces the proposed framework. Section 4 gives the security analysis. Section 5 presents the experimental results and evaluates various performance aspects of the proposed framework. Section 6 concludes this paper.

2. Related Work

(1) Trusted Execution Environment

Intel SGX [14] is a hardware-based security architecture technology that enables the creation of TEEs on Intel processors for cloud platforms and server environments. A TEE is a secure region of memory that isolates the execution of sensitive code and data from any untrusted system components, such as the operating system or the hypervisor. SGX extends CPU instructions to encapsulate the secure operations of legitimate software in an enclave, which is the basic unit of protection in SGX. Unlike other TEE technologies, such as TrustZone, SGX can run multiple secure enclaves on the same processor, each equivalent to a TEE. SGX also provides mechanisms for the attestation and sealing of enclaves, which allow for proving the integrity and confidentiality of the enclave's execution and data. SGX has been widely used to achieve privacy-preserving machine learning [15–19] by running the machine learning algorithms inside enclaves and protecting the data and models from unauthorized access.

(2) Federated Learning

Federated learning is a machine learning technique that enables distributed model training on multiple local data sources without sharing the raw data [5,20,21]. This technique leverages local model parameters, which do not reveal the original data, to construct a global model that captures the data value while preserving data privacy and security. Federated learning realizes a novel paradigm of "data available but not visible", which empowers decentralized data to be utilized for more accurate model training without violating data protection. In contrast to conventional centralized machine learning approaches, federated learning enables the collaboration of multiple parties on a shared model without exposing their private data, augmenting privacy and security [22]. It also mitigates the communication overhead and computational costs associated with centralized machine learning, as only the updates to the local model are transmitted to the central server instead of the whole dataset. Federated learning can be divided into three types: vertical federated learning, horizontal federated learning, and federated transfer learning [23]. Vertical federated learning is applicable to scenarios where data features are highly correlated but data samples are disjoint, and it can achieve cross-domain knowledge sharing. Horizontal federated learning is applicable for scenarios where data samples are highly correlated but data features are disjoint, and it can achieve cross-institutional knowledge fusion. Federated transfer learning is applicable to scenarios where data samples and data features are both disjoint but have some correlation, and it can achieve cross-domain knowledge transfer.

FedAvg is a federated learning algorithm that was introduced in a 2016 paper by Google researchers [24]. Federated learning exists to train deep neural networks on many devices, such as smartphones or tablets. These devices may have different types of data, and their users may not want to share their data with others for privacy reasons. FedAvg

allows these devices to train a local model on their own data without sending the data to a central server [25]. Instead, the device only sends the model parameters, which are numbers that represent how the model learns from the data. The central server then combines the parameters from all the devices by taking their weighted average, where the weights depend on how many data each device has. The server then sends the averaged parameters back to all the devices, which update their local models with the new parameters. This cycle of local training and global averaging continues until the global model reaches a good performance. FedAvg has some benefits over traditional learning methods that require all the data to be sent to a central server, such as saving communication bandwidth, handling diverse data sources, and improving model accuracy and generalization ability.

(3) Blockchain

Blockchain technology is a novel form of distributed ledger technology that employs cryptographic techniques and consensus protocols to maintain a shared record of transactions across multiple nodes, creating a sequential chain of data blocks. This technology exhibits several distinctive features, such as decentralization, immutability, distributed storage, anonymity, transparency, and smart contracts [26,27]. Decentralization refers to the absence of a central authority or intermediary institution that governs or regulates the blockchain. All participants have equal rights to join the blockchain network and contribute to its security through consensus protocols [28]. Immutability implies that the data stored on the blockchain are permanent and irreversible. Distributed storage indicates that the blockchain data are not located on a centralized server but dispersed across multiple nodes. This enhances the reliability and security of blockchain technology, making it resilient to attacks or tampering and having high fault tolerance. Anonymity and transparency denote that blockchain technology utilizes public keys and private keys to encrypt and decrypt data, safeguarding the privacy and security of users, while all transaction records are publicly available on the blockchain, and anyone can access the blockchain data [29]. Smart contracts are a type of self-executing code based on blockchain technology that can automatically perform under predefined conditions.

Based on the various features of blockchain technology, many interesting and valuable applications have emerged in various fields. Power Ledger is a blockchain-based energy-trading platform that allows consumers to sell their excess solar energy to other consumers and verify the source and quality of the renewable energy they purchase. IBM Food Trust is a blockchain-based food-traceability platform that connects farmers, processors, distributors, and retailers. It enables data sharing and traceability along the food supply chain and improves food safety, quality, and efficiency. Medicalchain is a blockchain-based platform that aims to create a decentralized electronic health record system, where patients can securely store and share their health data with authorized medical professionals and access remote healthcare services. Estonia, as a pioneer in the field of e-government, has implemented digital identity, e-residency, e-voting, e-health, and e-justice based on the blockchain, improving the security, transparency, and efficiency of public information processes [30–33].

The consortium blockchain is a special form of blockchain that allows only authorized nodes to join its network, which usually represents different physical organizations or enterprises that need to collaborate or share a task or resource [34]. Consortium blockchain typically uses specific consensus agreements to ensure that all nodes agree on the validity and sequence of transactions while providing a high degree of control and customizability [35]. The consortium blockchain is transparent, decentralized, highly controllable, and customizable. The consortium blockchain is suitable for applications that require sharing and collaboration among multiple organizations, such as cross-organizational transactions and collaborative operations.

A brief comparison of related works of literature with respect to various methodologies is shown in Table 1. Bonawitz [36] adopted federated learning, but their research does not incorporate the use of TEE (Trusted Execution Environment) or blockchain methodologies. A significant limitation of this work is its inability to fully guarantee data security and

privacy during the federated learning process. Chen et al. [37] integrates both federated learning and TEE but does not utilize the blockchain. Its limitation lies in only being able to guarantee data security and privacy during the data-aggregation process to a certain degree, implying there may be vulnerabilities or scenarios where data could be at risk. Li et al. [38] utilize federated learning and the blockchain but not TEE. This can only ensure data security and privacy within the local model to a certain extent. This might indicate potential challenges or vulnerabilities when scaling or in more complex scenarios.

Table 1. Comparison of related works of literature with respect to various methodologies.

Paper	Federated Learning	TEE	Blockchain	Features
[36]	Yes	No	No	Cannot fully guarantee the data security and privacy in the federated learning process.
[37]	Yes	Yes	No	Can guarantee data security and privacy in the data aggregation process to a certain extent.
[38]	Yes	No	Yes	Can guarantee data security and privacy in the local model to a certain extent.
Ours	Yes	Yes	Yes	Can fully guarantee data security locally and privacy in the data-aggregation process.

3. Proposed Framework

The summary of notations used in the methodology can be seen in Table 2.

Table 2. Description of notations.

Notations	Description
D_i	Local data for device i
E	Number of training epochs
α	Learning rate
p, g	Diffie–Hellman parameters
SK_i	Private key for Diffie–Hellman of device i
PK_i	Public key for Diffie–Hellman of device i
N	Nonce generated by TEE
T	Attestation token generated by TEE
SK_{TEE}	Private key for Diffie–Hellman of TEE
PK_{TEE}	Public key for Diffie–Hellman of TEE
w	Model weights
$\hat{y}$	Model prediction
y	True label
∇	Gradient
E_{weight}	Encrypted weight
$\mathcal{D}$	Device sets
K	Number of global iterations
η	FedAvg learning rate
$\{\mathbf{w}_i\}_{i\in\mathcal{D}}$	Local model weight for each device in $\mathcal{D}$
$H_{ID,i}$	Device identity hash value for device i
$\mathbf{w}_0$	Initial global weight
B_t	Block containing verified local model weights
$\mathbf{a}_{t'}$	Aggregated update at iteration t'
$\nabla f(\mathbf{a}_{t'}, \mathcal{D})$	Gradient of objective function
$\mathbf{w}_{t'}$	Global weight at iteration t'

3.1. System Overview

In this paper, we propose a secure and efficient federated learning framework based on a Trusted Execution Environment (TEE) and the blockchain for industrial Internet of Things (IoT) supply chain applications. Our framework aims to address the challenges of data privacy, model security, and model verifiability in federated learning, which is a promising

technique to enable collaborative learning among multiple parties without sharing raw data. We consider an IoT supply chain scenario where N stakeholders (e.g., manufacturers, warehouses, logistics providers, and retailers) collect data $\mathcal{D}_i = \{(\mathbf{x}_i^j, y_i^j)\}_{j=1}^{n_i}$ from various sensors installed on products, equipment, or vehicles along supply chain stages. These data are used to train local models on each stakeholder's device using TEE technology, which provides hardware-level isolation and protection for local model parameters $\mathbf{w}_i$ from being exposed to aggregation servers or malicious attackers during the training process. The local models are then aggregated by a central server using a federated averaging (FedAvg) algorithm over a blockchain network, which ensures secure aggregation and tamper-resistant storage of model parameters among distributed nodes. The server updates global model parameters $\mathbf{w}$ by computing the weighted average of local model parameters, where $n = \sum_{i=1}^{N} n_i$ is the total number of data points. The server sends the updated global model parameters $\mathbf{w}$ back to stakeholders via the blockchain consensus mechanism, which allows stakeholders to verify the correctness and integrity of global model updates. The stakeholders can use the updated global model to perform product recognition and classification tasks on their own data without revealing sensitive information to others. The framework iterates until the convergence criterion is reached or a predefined number of communication rounds are completed. Figure 2 illustrates the overview of our proposed framework.

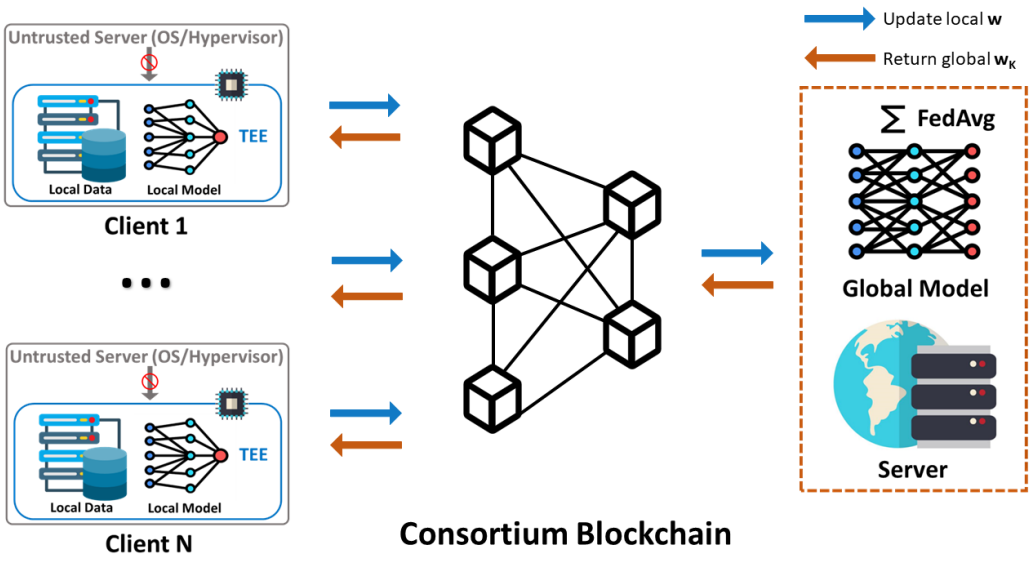


Figure 2. Overview of the proposed framework.

3.2. Threat Model

This paper investigates a federated learning system that consists of a central server and multiple clients. Each client has its own local dataset that is hidden from others. The server starts the model training by distributing the initial parameters to the clients and then receives and aggregates their updates after each round. The server is assumed to be honest and will not reveal or change the data or parameters. The communication between the clients is also assumed to be secure and will not be eavesdropped on or tampered with. Side-channel attacks are ignored when using SGX. However, some clients may be malicious and have motives to sabotage the federated learning system. Three types of attack objectives for malicious clients are identified: robustness attacks, privacy attacks, and free-riding attacks. Robustness attacks aim to impair the model's accuracy or availability, making it unsuitable or untrustworthy for its intended tasks. Privacy attacks aim to infer other clients' private data or model parameters, obtaining sensitive information or competitive advantage. Free-riding attacks aim to exploit other clients' contributions to improve their own model performance without paying the corresponding cost.

3.3. Local Model in TEE for Federated Learning

In order to address the problem of the security of the local model, we implement the CNN model training inside the enclave. The algorithm details a local model in a TEE for federated learning. Federated learning is a distributed machine learning technique that allows multiple parties to collaboratively train a model without sharing their raw data. A TEE is a secure area of a processor that protects the code and data from being tampered with or leaked by other processes. The overview is shown in Figure 3.

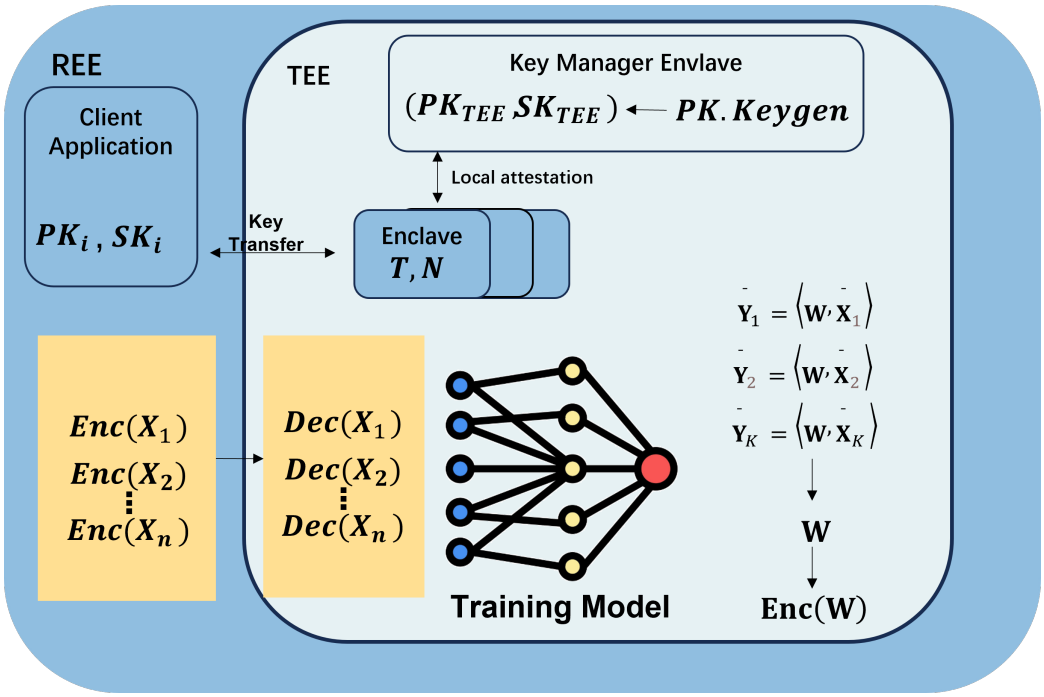


Figure 3. The architecture of the local model in a trusted execution environment.

Diffie–Hellman key exchange is based on the property of modular exponentiation that $(a^b)^c \mod p = a^{bc} \mod p$. Since $PK_i = g^{SK_i} \mod p$ and $PK_{\text{TEE}} = g^{SK_{\text{TEE}}} \mod p$, we have

$$\begin{aligned} S &= PK_i^{SK_{\text{TEE}}} \mod p \\ &= (g^{SK_i} \mod p)^{SK_{\text{TEE}}} \mod p \\ &= g^{SK_i SK_{\text{TEE}}} \mod p \\ &= (g^{SK_{\text{TEE}}} \mod p)^{SK_i} \mod p \\ &= PK_{\text{TEE}}^{SK_i} \mod p \end{aligned} \tag{1}$$

Gradient descent update is based on the idea of finding the minimum of a function by moving in the opposite direction of its gradient. The gradient is the vector of partial derivatives that points to the steepest ascent of the function. The learning rate α controls the step size of the update. The loss function Loss measures the discrepancy between the model prediction $\hat{y}$ and the true label y. The model prediction $\hat{y}$ is a function of the model weights w and the input x. The update rule can be derived as follows:

$$\begin{aligned} w &\leftarrow w - \alpha \cdot \nabla \text{Loss}(\hat{y}, y) \\ &= w - \alpha \cdot \frac{\partial}{\partial w} \text{Loss}(\hat{y}, y) \\ &= w - \alpha \cdot \frac{\partial}{\partial w} \text{Loss}(\text{Model}(w, x), y) \\ &= w - \alpha \cdot \frac{\partial}{\partial w} L(w) \end{aligned} \tag{2}$$

where $L(w)$ is a shorthand notation for $\text{Loss}(\text{Model}(w, x), y)$.

Encryption and decryption in TEE are based on the assumption that there exists an encryption function Encrypt and a decryption function Decrypt that have the following properties:

- For any message m and key kk, $\text{Decrypt}(k, \text{Encrypt}(k, m)) = m$.
- For any ciphertext cc and key kk, $\text{Encrypt}(k, \text{Decrypt}(k, c)) = c$.
- It is computationally infeasible to recover m from $\text{Encrypt}(k, m)$ without knowing k.

The encryption equation uses the public key of the rich execution environment (REE) as the key to encrypt the weight: $E_{\text{weight}} = \text{Encrypt}(PK_i, w)$. The decryption equation uses the shared secret key as the key to decrypt the encrypted weight: $w = \text{Decrypt}(S, E_{\text{weight}})$. Since $S = PK_i^{SK_{\text{TEE}}} = PK_{\text{TEE}}^{SK_i}$, we have:

$$\begin{aligned} w &= \text{Decrypt}(S, E_{\text{weight}}) \\ &= \text{Decrypt}(PK_i^{SK_{\text{TEE}}}, E_{\text{weight}}) \\ &= \text{Decrypt}(PK_i^{SK_{\text{TEE}}}, \text{Encrypt}(PK_i, w)) \\ &= w \end{aligned} \tag{3}$$

Algorithm 1 consists of four main parts:

- Input and output: The input includes the local data D_i, the number of training epochs E, the learning rate α, and the Diffie–Hellman parameters p and g. The output is the weight w of the local model.
- Interaction between REE and TEE: The REE is the normal operating system that runs outside the TEE. The REE and TEE communicate through a secure channel to exchange public keys for Diffie–Hellman key exchange, which is a method to generate a shared secret key without revealing it to an eavesdropper. The public key of the REE is $PK_i = g^{SK_i} \mod p$, where SK_i is the private key of the REE. The public key of the TEE is $PK_{\text{TEE}} = g^{SK_{\text{TEE}}} \mod p$, where SK_{TEE} is the private key of the TEE. The shared secret key is $S = PK_i^{SK_{\text{TEE}}} \mod p = PK_{\text{TEE}}^{SK_i} \mod p$.
- Local attestation process in TEE: The TEE generates a nonce N and an attestation token T, which is a digital signature that proves the identity and integrity of the TEE. The signature uses the private key of the TEE, denoted by SK_{TEE}, and can be verified by anyone who knows the public key of the TEE, denoted by PK_{TEE}. The nonce prevents replay attacks by ensuring that the token is fresh and unique. The TEE sends N and T to the REE, which verifies the token by checking if $\text{Verify}(PK_{\text{TEE}}, N, T)$ returns true.
- Local model in TEE: The TEE initializes the model weights w and trains them for E epochs using gradient descent on the local data D_i. For each sample (x, y) in D_i, the model computes the prediction $\hat{y} = \text{Model}(w, x)$ and the gradient $\nabla = \nabla\text{Loss}(\hat{y}, y)$, where Loss is a loss function that measures the discrepancy between $\hat{y}$ and y. The model updates the weights by subtracting a fraction of the gradient: $w \leftarrow w - \alpha \cdot \nabla$, where α is the learning rate. After training, the TEE encrypts the weight using the public key of the REE: $E_{\text{weight}} = \text{Encrypt}(PK_i, w)$, where Encrypt is an encryption function. The encrypted weight is sent to the REE, which decrypts it using the shared secret key $w = \text{Decrypt}(S, E_{\text{weight}})$, where Decrypt is a decryption function. The decrypted weight is returned as the output of the algorithm.

The weight "w" is expected to improve with each iteration. The improvement here is in the context of minimizing the loss function specific to the local dataset D_i. Each iteration updates the weight based on the gradient of the loss, directing it toward an optimal value for those specific data.

The algorithm assumes that some functions are predefined, such as ModelModel, LossLoss, SignSign, VerifyVerify, EncryptEncrypt, and Decrypt. These functions may vary depending on the specific implementation of federated learning and TEE.

Algorithm 1 Local Model in TEE for Federated Learning

Input:
Local data: D_i
Number of training epochs: E
Learning rate: α
Diffie–Hellman parameters: p, g
Output: weight w

Generate private key for Diffie–Hellman: SK_i
Compute public key for Diffie–Hellman: $PK_i \leftarrow g^{SK_i} \mod p$
Interaction between REE and TEE:
REE: Send PK_i to TEE
TEE: Receive PK_i from REE
Local Attestation Process in TEE:
TEE: Generate nonce: N
TEE: Compute attestation token: $T = \text{Sign}(SK_{\text{TEE}}, N)$
TEE: Send N and T to REE
REE: Receive N and T from TEE
REE: Verify attestation token: $\text{Verify}(PK_{\text{TEE}}, N, T)$
In TEE:
TEE: Generate fresh private key for Diffie–Hellman: SK_{TEE}
TEE: Compute public key for Diffie–Hellman: $PK_{\text{TEE}} \leftarrow g^{SK_{\text{TEE}}} \mod p$
Local Model in TEE:
Initialize model weights: w
for $e = 1$ to E **do**
    for each sample (x, y) in D_i **do**
        Compute model prediction: $\hat{y} = \text{Model}(w, x)$
        Compute gradient: $\nabla = \nabla \text{Loss}(\hat{y}, y)$
        Update model weights: $w \leftarrow w - \alpha \cdot \nabla$
Interaction between REE and TEE:
TEE: Send encrypted weight: $E_{\text{weight}} = \text{Encrypt}(PK_i, w)$ to REE
REE: Receive encrypted weight: E_{weight} from TEE
REE: Decrypt weight using Diffie–Hellman: $w = \text{Decrypt}(SK_i, E_{\text{weight}})$
return: w

3.4. Federated Averaging on Blockchain

Federal learning involves data exchange among multiple parties. To ensure the security and integrity of the data, this paper adopts the FedAvg algorithm combined with the consortium blockchain to achieve secure data transmission. The consortium blockchain is a distributed ledger system based on blockchain technology that can improve the security and credibility of data-sharing and interaction among multiple parties. The FedAvg algorithm uses the average of the model parameters to achieve collaborative learning among participants. An overview of this process is shown in Figure 4. And the flow of reaching consensus on the blockchain is shown in Figure 5.

FedAvg is a communication-efficient algorithm for distributed training with many clients who keep their data locally for privacy. A central server communicates the global model parameter to each client and aggregates the updated local model parameters from clients. The core formula of FedAvg is

$$w^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{(t+1)} \tag{4}$$

where $w^{(t+1)}$ is the global model parameter at round $t+1$, K is the number of clients, n_k is the number of local data samples on client k, n is the total number of data samples across all clients, and $w_k^{(t+1)}$ is the local model parameter updated by client k on round $t+1$. The formula is derived by having each selected client perform SGD on its local data using

the global model parameter as the initial value and having the server take a weighted average of the received local model parameters. FedAvg can reduce communication costs and improve privacy compared to centralized methods that require sending raw data or gradients to the server.

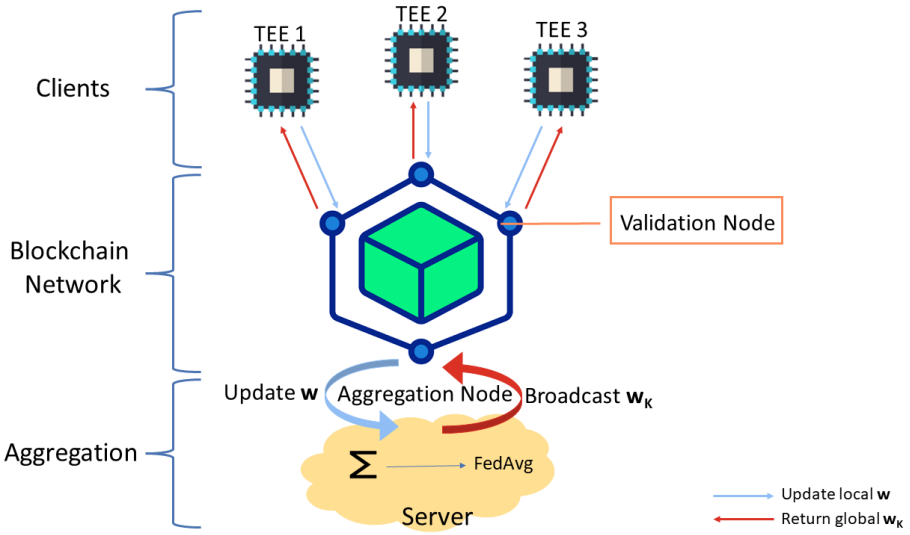


Figure 4. FedAvg on the blockchain.

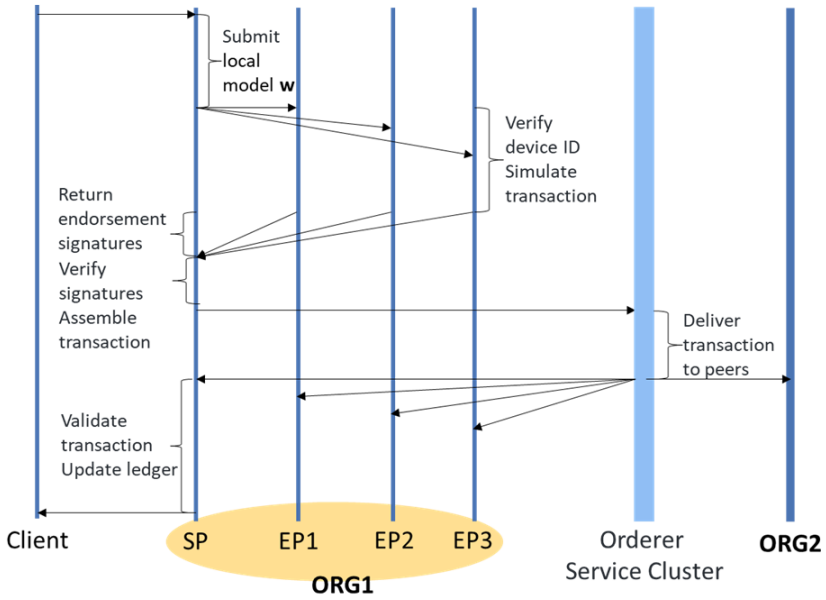


Figure 5. Flow of reaching consensus on the blockchain.

Algorithm 2 consists of four main parts:

- Input and output: The algorithm takes the following input, the set of devices $\mathcal{D}$, the number of global iterations K, the FedAvg learning rate η, the local model weight $\{\mathbf{w}_i\}_{i\in\mathcal{D}}$, and the device identity hash value $H_{ID,i}$ for all devices $i \in \mathcal{D}$. The output of the algorithm is the global weight $\mathbf{w}_K$ after global training.
- Hash Value Verification: To verify the identity information of each device, the algorithm reads the corresponding smart contract from the blockchain. If the device's identity hash value $H_{ID,i}$ matches the verified hash value, the corresponding local model weight $\mathbf{w}_i$ is stored in a list; otherwise, the local model weight $\mathbf{w}_i$ is discarded.
- Blockchain Uploading: All verified local model weights are stored in a block, which is then transmitted to the Fabric blockchain network. The algorithm waits for the

consensus result from the Raft network. If consensus is reached, the accepted block is received at all replicas, and the local model weights of the verified devices are extracted for use in the aggregation step.

- Aggregation: The local model weights of all verified devices are aggregated. The average value of these verified local model weights $\mathbf{a}_{t'}$ is computed and used to update the global model weight $\mathbf{w}_{t'}$. Here, $\mathbf{w}_{t'-1}$ represents the global model weight from the previous iteration. The FedAvg strategy is used to update the global model weight $\mathbf{w}_{t'} \leftarrow \mathbf{w}_{t'-1} - \eta \nabla f(\mathbf{a}_{t'}, \mathcal{D})$, where $f(\cdot)$ is the objective function and $\mathcal{D}$ is the global dataset.

Algorithm 2 FederatedAveraging on Blockchain

Input:
Device sets: $\mathcal{D}$
Number of global iterations: K
FedAvg learning rate: η
Local model weight: $\{\mathbf{w}_i\}_{i\in\mathcal{D}}$ and device identity hash value: $H_{ID,i}$ for all $i \in \mathcal{D}$
Output: global weight $\mathbf{w}_K$

procedure FEDAVG($\mathcal{D}, K, \eta, \{\mathbf{w}_i\}_{i\in\mathcal{D}}, \{H_{ID,i}\}_{i\in\mathcal{D}}$)
  Initialize the global weight $\mathbf{w}_0$
  for $t = 1, 2, \ldots, K$ **do**
    for $i \in \mathcal{D}$ **do**
      if the device ID hash value in $H_{ID,i}$ is validated by the corresponding smart contract **then**
        Store the verified local model weight in a list: $\{\mathbf{w}_i\}_{i\in\mathcal{D},\text{verified}}$.
      else
        Discard the local model weight $\mathbf{w}_i$ of device i due to mismatched ID hash value.
    Form a block B_t containing all verified local model weights $\{\mathbf{w}_i\}_{i\in\mathcal{D},\text{verified}}$.
    Upload the block B_t to the Fabric blockchain: $B_t \rightarrow Upload$
    Wait for the consensus result from the Raft network: $Consensus \rightarrow Wait$
    if consensus achieved **then**
      Receive the accepted block at all replicas, extract the local model weights of verified devices
      Aggregate the received updates using FedAvg: $\mathbf{a}_{t'} \leftarrow \frac{1}{C}\sum_{i=1}^{C}\mathbf{w}_i$, where $\{\mathbf{w}_i\}_{i=1}^{C}$ are the verified local model weights in the accepted block.
      Update the global weight using FedAvg: $\mathbf{w}_{t'} \leftarrow \mathbf{w}_{t'-1} - \eta \nabla f(\mathbf{a}_{t'}, \mathcal{D})$, where $f(\cdot)$ is the objective function and $\mathcal{D}$ is the global dataset.
  return $\mathbf{w}_K$

4. Security Analysis

Federated learning is a distributed collaborative learning technique that enables training a global model using local data from multiple clients without sharing the data. However, federated learning faces various security and privacy threats, such as data poisoning by malicious clients, model poisoning, inference attacks, free-riding attacks, and robustness attacks. To address these threats, we propose an algorithm for training a local model in TEE.

4.1. Local Training Model in TEE

TEE is a hardware-isolation technology that ensures that the code and data running within it are protected from external interference or leakage. The algorithm presented in this paper leverages TEE to provide the following security assurances. The proposed algorithm incorporates several security measures to mitigate security and privacy threats in federated learning. These measures include secure communication using the Diffie–Hellman key exchange protocol, local attestation, encryption and decryption mechanisms, and local model training within the TEE.

Secure communication between the REE and TEE is achieved through the utilization of the Diffie–Hellman key-exchange protocol. This protocol, based on mathematical principles, enables the two parties to negotiate a shared key over a public channel without directly sharing the key itself. By relying on the computational hardness of the discrete logarithm problem, the protocol ensures that computing the private key from the public parameters is a challenging task. As a result, this approach prevents potential man-in-the-middle attacks and replay attacks, bolstering the security of the communication channel.

To verify the identity and integrity of the TEE, a local attestation process is implemented. In this process, the TEE generates a random number (nonce) and an attestation token, which are then transmitted to the REE. Upon receiving these parameters, the REE employs the TEE's public key to verify the authenticity of the attestation token and its correspondence to the received nonce. The successful verification of these parameters establishes trust in the TEE, effectively preventing impersonation or tampering attacks. To protect the confidentiality of the model weights, encryption and decryption mechanisms are employed. The TEE utilizes the Diffie–Hellman public key received from the REE to encrypt the model weights. Subsequently, the encrypted weights are securely transmitted to the REE, which possesses the corresponding Diffie–Hellman private key required for decryption. As only the REE holds the private key, unauthorized entities are unable to decrypt or modify the weights, safeguarding against theft or unauthorized manipulation. Furthermore, local model training within the TEE ensures privacy and prevents inference attacks or GAN attacks. This process involves the initialization of model parameters using locally generated Diffie–Hellman public and private keys. Multiple rounds of training are then conducted on the local dataset, enabling the TEE to update the model weights accordingly. Since only the TEE possesses knowledge of its private key and the weights, external entities are unable to infer any sensitive information about the data or weights, reinforcing privacy and thwarting potential attacks.

By leveraging the TEE and integrating mechanisms for secure communication, local attestation, encryption, and local model training, the proposed algorithm establishes robust security measures. These measures collectively mitigate various security and privacy threats in the context of federated learning. The combination of secure communication, trust establishment, data encryption, and local training within the TEE ensures the confidentiality, integrity, and authenticity of the communication and data, bolstering the overall security of the federated learning process.

4.2. Aggregate Security

Federated learning also faces various security and privacy challenges, such as malicious client attacks, local weight leakage or tampering, etc. To address these challenges, this paper optimizes the aggregation part of federated learning and proposes a scheme that combines aggregation algorithms with blockchain technology, thereby improving the security and efficiency of federated learning. The scheme proposed in this paper mainly includes two steps: consortium chain-based verification and consortium chain-based transmission.

In the consortium chain-based verification step, each participant needs to upload their local weights to the chain and attach their own identity digital signature after completing the local model training. A smart contract is deployed on the consortium chain to verify whether the local weights come from legitimate clients, i.e., whether they match the identity digital signature. Only local weights that pass the verification can be recorded on the blockchain and used for subsequent global model updates. Otherwise, local weights will be rejected and discarded. This verification mechanism can effectively prevent malicious clients from forging or tampering with local weights, thereby ensuring the quality and accuracy of the global model. At the same time, this verification mechanism can also solve the trust and attack problems that exist in traditional federated learning and improve the security and reliability of federated learning. Through the consortium-chain-based verification mechanism, we can enhance trust and collaboration among participants and prevent data pollution and model poisoning attacks.

In the consortium-chain-based transmission step, each participant no longer needs to send their local weights to the aggregation server after uploading them to the chain; instead, the aggregation server obtains all participants' local weights from the blockchain and performs centralized aggregation. This transmission mechanism can effectively ensure the anti-tampering and security of local weights, prevent local weights being maliciously modified or stolen during transmission, and ensure the traceability and immutability of local weights. At the same time, this transmission mechanism can also reduce communication overhead and improve efficiency, solving the communication and efficiency problems that exist in traditional federated learning and improving the performance and effect of federated learning. Through the consortium-chain-based transmission mechanism, we can protect participants' data privacy and model knowledge and prevent data leakage and model theft attacks.

4.3. Adaptability

Our framework leverages Trusted Execution Environment (TEE) and blockchain technology to address data security and privacy challenges in federated learning, especially within the IoT supply chain management domain. However, our approach is general and can be extended to various contexts where data security and privacy are paramount. The core principles of our framework are ensuring data security, data integrity, and user privacy. These principles are applicable across various domains where federated learning and data security concerns overlap, such as healthcare, smart cities, or industrial IoT. Moreover, any collaborative machine learning or data analytics scenario where data privacy is a concern can potentially benefit from our framework. Examples include finance, healthcare, and e-commerce, where transactional data, patient records, and user purchase history need to be protected. The combination of TEE and blockchain ensures not only data security but also the traceability and accountability of computations, making it suitable for any scenario requiring trustworthiness and auditability. Furthermore, our system is modular and adaptable. Different components (e.g., a different blockchain or a different TEE) can be integrated based on the specific requirements of another context, allowing for flexibility in deployment. Therefore, our framework has a wide range of potential applications beyond IoT supply chain management.

5. Results

In this section, we provide an overview of the implementation and evaluation of our proposed framework.

5.1. Experimental Methodology

We present the experimental results of our proposed scheme for privacy-preserving federated learning based on TEE and the blockchain. We first evaluated the performance of the local model training in TEE for federated learning. Then, we conducted experiments to compare the performance of federated learning with blockchain-based parameter updates with different numbers of nodes and transactions. We collected a dataset from real-world IoT devices of a supply chain, which includes production data from multiple suppliers, sales data from retailers, and product data. The detailed configuration of the experimental environment is presented in Table 3.

Table 3. Experiments setup.

Experiments Setup	Specification
CPU	Intel Xeon Processor 2695 CPU 2.40 GHz
Ram	64 GB
Operating System	Ubuntu 18.04
Implementation	Python 3.7.0
Libraries	Pytroch 1.6.0
System Setup	Hyperledger Fabric 2.0, Docker 20.10.7

5.2. Performance Evaluation

We conducted an experiment to evaluate our framework's performance in local clients using federated learning from Figure 6. We compared the time costs of local training and aggregated three models (LeNet, VGG16, and ResNet) with 200 data samples each on two datasets (Fashion MNIST and MNIST) under two scenarios: using TEE in Intel SGX or using REE. We varied the number of local nodes from 5 to 30 and measured the total time required for each scenario. Figure 6 shows the results of our experiment. We observed that ResNet had similar trends on both datasets, as shown in Figure 6a,d. The training and aggregation time remained stable until 15 local nodes but increased when reaching 20 local nodes or more. The increase was about 1.6 s for both scenarios. Figure 6b,c show the results for LeNet and VGG on the MNIST dataset. We found that using TEE increased the time cost by about 1.57 s compared to using REE when using 30 local nodes for VGG model. However, this was only 1.7 s slower than not using SGX at all for aggregation. The other models had similar results. The main reason for the increased time cost in TEE was the memory limitation of Intel SGX, which affected the efficiency of federated learning.

We evaluated our framework's accuracy by testing the final global model obtained from federated training on a test dataset. We varied the number of participants and the number of iterations in our experiment. We considered four scenarios with different numbers of iterations: 5, 10, 20, and 30. We also varied the number of participants from 1 to 30. Figure 7 shows our accuracy results for each scenario. We found that increasing the number of both iterations and participants improved the accuracy of our global model. For example, when using only one participant and five iterations, our model achieved an accuracy of 94%. However, when using ten participants and thirty iterations, our model reached an accuracy of 98.5%. This indicates that more iterations and participants allow our framework to capture more features from the training data and generate a better classifier. The main reason for this improvement is that each participant randomly selects samples from their local data for each iteration. When there are few participants or iterations, some samples may not be used for training at all, which reduces the diversity and representativeness of our global model. By increasing both factors, we can ensure that our framework covers more samples from different distributions and learns a more accurate model.

For the evaluation of the system throughput, we mainly measured the overall scale of the system in recording verification and data acquisition in the blockchain. We constructed three experimental scenarios with different numbers of participants joining, where the transaction number was set to 500, 1000, and 2000. Figure 8 shows the relationship between different numbers of participants and system throughput under the condition of completing the transaction number of the three experimental scenarios. When the number of participants was two, the system throughput performance showed a significant upward trend as the set's transaction number increased, rising from about 20 TPS to about 34 TPS. However, as the number of participants increased, the system throughput decreased. Under normal circumstances, the increase in participants would have a positive effect on system throughput. The reason for this, we believe, is that the transaction number was set too high, resulting in too many data generated by participants, causing data-congestion problems for the system and a performance bottleneck. We also conducted throughput experiments for four and six participants in different stages of the FedAvg on Chain process, such as local weight upload, verification, and aggregation, as shown in Figure 9. Figure 9a is a graph of the relationship between transaction volume and throughput for four participants and Figure 9b is for six participants. A common point for four and six participants is that the throughput of local weight uploaded to the blockchain stage is lower than that of the other two stages, and as the transaction volume increases, the throughput performance of four participants is better than that of six participants, for the same reason that too much data generation leads to performance bottlenecks. Only when the transaction volume is too large will it cause a decline in the throughput performance. FedAvg on Chain

does not have thousands of iterations, ensuring that the system can run normally with multiple participants.

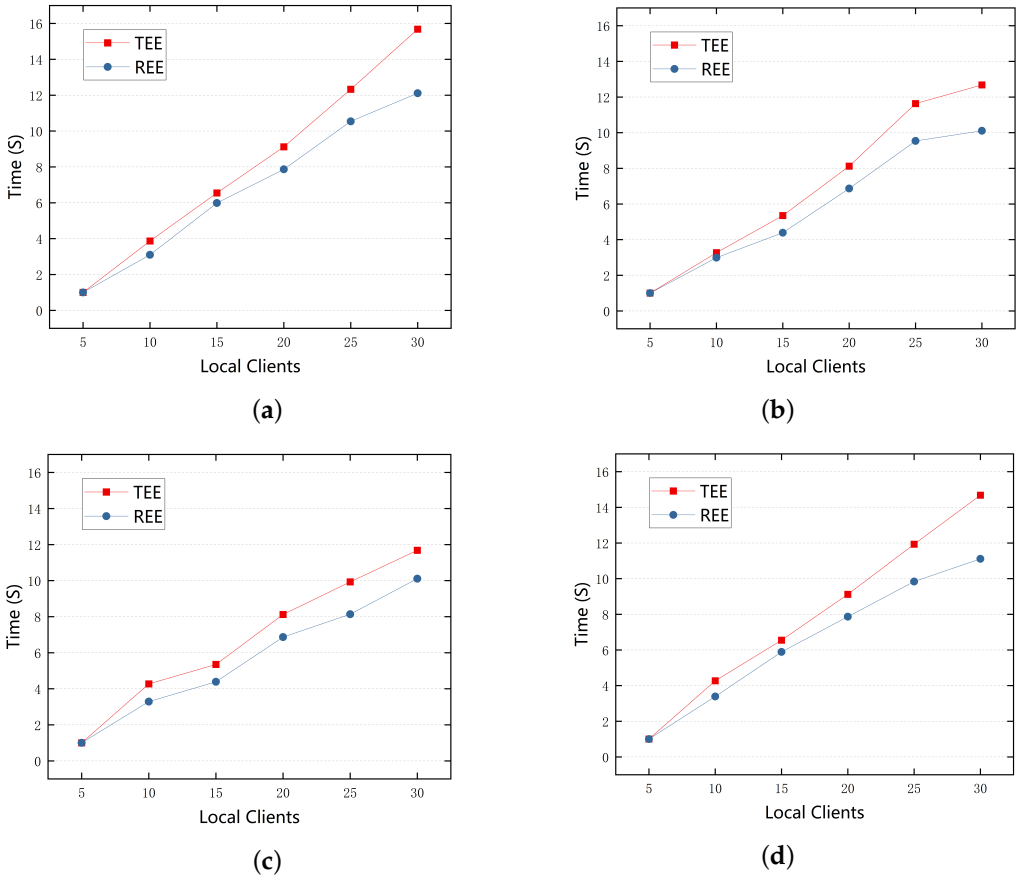


Figure 6. Comparison of durations of training processes between the TEE and REE for various deep learning models and the Fashion MNIST dataset. (**a**) ResNet-Fashion MNIST. (**b**) LeNet-MNIST. (**c**) VGG16-MNIST. (**d**) ResNet-MNIST.

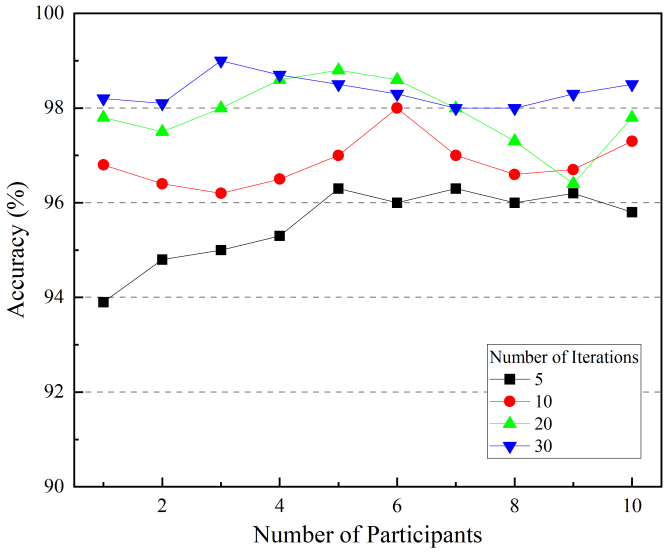


Figure 7. Federated learning accuracy.

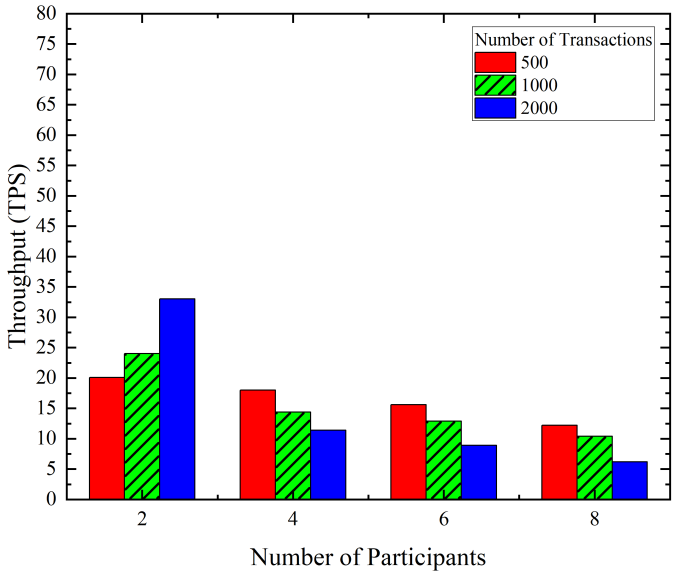


Figure 8. Throughput.

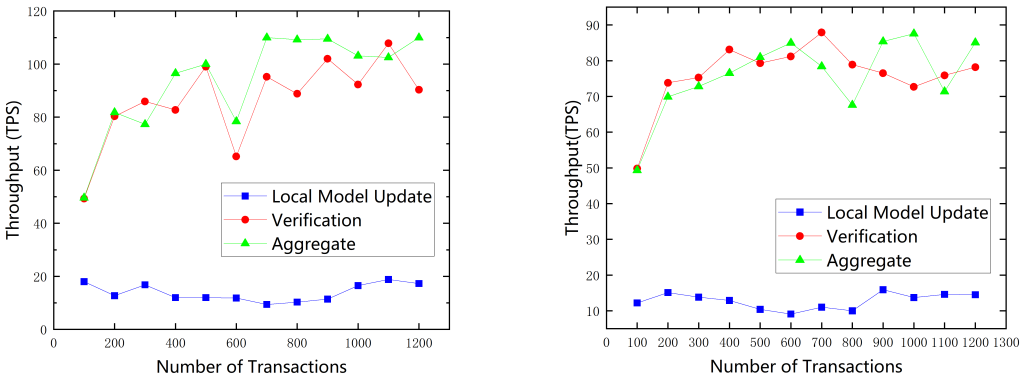


Figure 9. The comparison of throughput between 4 nodes and 6 nodes.

We also tested the transaction latency of the system, as well as the evaluation of the system throughput. We still constructed scenarios with 500, 1000, and 2000 transactions and conducted experiments on the average transaction latency of the system with different numbers of participants. Figure 10 shows that as the number of participants increases, the average transaction time delay shows a positive trend. In the case of eight participants, the average time delay for 500 transactions can reach about 40 s. The number of participants remains unchanged, and the increase in transaction volume is also the reason for the increase in average time delay. The reason is that as the number of participants and transactions increase, the system blockchain will increase the overhead of the consensus algorithm and the network communication load, resulting in a decrease in system performance and a delay in transaction completion time. For the evaluation of time delay, we also conducted tests for four and six participants in the same experimental scenarios as the throughput. Figure 11a is a graph of the relationship between transaction volume and throughput for four participants, and Figure 11b is for six participants. As shown in the figure, whether it is four or six participants, the local weight upload stage has a higher time delay than the verification and aggregation stages, which is opposite to the throughput situation. The reason is also that the system reaches a performance bottleneck, resulting in a decrease in throughput and an increase in time delay. Such results provide a reference direction for the subsequent optimization of this system.

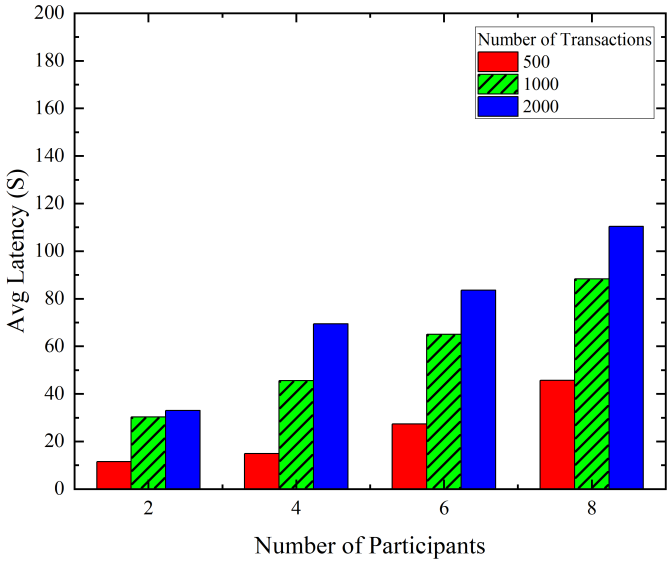
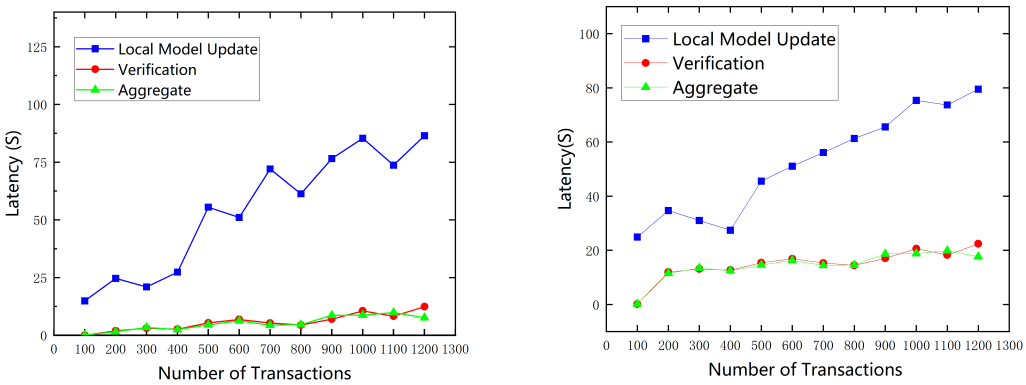


Figure 10. Average Latency.



(**a**) (**b**)

Figure 11. The comparison of latency between 4 nodes and 6 nodes.

6. Conclusions

In this paper, we propose a TEE and consortium-blockchain-based federated learning framework for the IoT of the supply chain. The main objective of this framework is to ensure the tamper resistance of data in federated learning, protect the data security in the whole federated learning process, and achieve the security and effectiveness of the aggregation results. The local model runs in the TEE. The framework uses an Intel SGX-based TEE to ensure the secure execution of the local model and the trusted output of the local data and then uses consortium blockchain technology to realize the secure transmission of the local data, and finally obtains the aggregated parameter results. In this framework, each blockchain node is equipped with a verification node, which verifies whether the data source is consistent with the set participants and ensures the trustworthiness of data usage. Finally, through blockchain technology, the aggregated global model parameters are added to the blockchain and returned to all local models. We set different numbers of participants and different datasets to test the framework. The experimental results show that the local data-processing time of our proposed framework is almost the same as that of the original federated learning model. In addition, our framework has high throughput performance for normal use by multiple participants.

However, our proposed framework also has some limitations that need to be addressed in future work. First, our system relies on TEE and blockchain technology, which may increase the system's complexity and cost, as well as the hardware and network require-

ments. We plan to explore how to reduce the system overhead and resource consumption while maintaining security and efficiency. Second, our system uses Intel SGX as a specific implementation of TEE, but Intel SGX also has some known security vulnerabilities, such as Foreshadow and Plundervolt, which may affect the reliability and robustness of our system. We plan to investigate how to deploy our system on different TEE platforms and how to prevent or detect these potential attacks.

Author Contributions: Conceptualization, L.Z. and S.H.; methodology, L.Z.; software, S.H.; validation, L.Z., S.H. and M.H.; formal analysis, S.H. and C.M.; investigation, X.Z., C.M. and M.H.; resources, L.Z. and S.H.; writing—original draft preparation, L.Z. and S.H.; writing—review and editing, L.Z. and S.H.; visualization, L.Z., S.H. and M.H.; supervision, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by MOST-FDCT Projects (0058/2019/AMJ, 2019YFE0110300) (Research and Application of Cooperative Multi-Agent Platform for Zhuhai-Macao Manufacturing Service).

Data Availability Statement: The data can not be shared due to the project's privacy policy.

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

References

1. Ben-Daya, M.; Hassini, E.; Bahroun, Z. Internet of Things and Supply Chain Management: A Literature Review. *Int. J. Prod. Res.* **2019**, *57*, 4719–4742. [CrossRef]
2. Qu, T.; Lei, S.; Wang, Z.; Nie, D.; Chen, X.; Huang, G.Q. IoT-Based Real-Time Production Logistics Synchronization System under Smart Cloud Manufacturing. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 147–164. [CrossRef]
3. Tao, F.; Zuo, Y.; Xu, L.D.; Zhang, L. IoT-Based Intelligent Perception and Access of Manufacturing Resource toward Cloud Manufacturing. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1547–1557.
4. Wen, Q.; Gao, Y.; Chen, Z.; Wu, D. A Blockchain-Based Data Sharing Scheme in the Supply Chain by IIoT. In Proceedings of the 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), Taipei, Taiwan, 6–9 May 2019; pp. 695–700.
5. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 12. [CrossRef]
6. Li, A.; Zhang, L.; Tan, J.; Qin, Y.; Wang, J.; Li, X.-Y. Sample-level Data Selection for Federated Learning. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10. [CrossRef]
7. Demertzis, K.; Iliadis, L.; Pimenidis, E.; Tziritas, N.; Koziri, M.; Kikiras, P.; Tonkin, M. Federated Blockchained Supply Chain Management: A CyberSecurity and Privacy Framework. In Artificial Intelligence Applications and Innovations. 2021. Available online: http://hdl.handle.net/11615/60214 (accessed on 30 October 2022).
8. Zheng, G.; Kong, L.; Brintrup, A. Federated Machine Learning for Privacy Preserving, Collective Supply Chain Risk Prediction. *Int. J. Prod. Res.* **2023**, 1–18. [CrossRef]
9. Liu, Y.; Yu, W.; Ai, Z.; Xu, G.; Zhao, L.; Tian, Z. A Blockchain-Empowered Federated Learning in Healthcare-Based Cyber Physical Systems. *IEEE Trans. Netw. Sci. Eng.* **2022**. [CrossRef]
10. Wu, J.M.T.; Teng, Q.; Huda, S.; Chen, Y.-C.; Chen, C.-M. A Privacy Frequent Itemsets Mining Framework for Collaboration in IoT Using Federated Learning. *ACM Trans. Sens. Netw.* **2023**, *19*, 27. [CrossRef]
11. Lu, Y.; Huang, X.; Dai, Y.; Maharjan, S.; Zhang, Y. Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4177–4186. [CrossRef]
12. Qammar, A.; Karim, A.; Ning, H.; Ding, J. Securing Federated Learning with Blockchain: A Systematic Literature Review. *Artif. Intell. Rev.* **2023**, *56*, 3951–3985. [CrossRef]
13. Korkmaz, C.; Kocas, H.E.; Uysal, A.; Masry, A.; Ozkasap, O.; Akgun, B. Chain FL: Decentralized Federated Machine Learning via Blockchain. In Proceedings of the 2020 Second International Conference on Blockchain Computing and Applications (BCCA), Antalya, Turkey, 24–26 November 2020; pp. 140–146. [CrossRef]
14. Intel Corporation. *Intel Software Guard Extensions Programming Reference*; Intel Corporation: Santa Clara, CA, USA, 2014.
15. Zhang, Y.; Wang, Y.; Liu, J.; Shi, W. SGX-FPGA: Trusted Execution Environment for CPU-FPGA Heterogeneous Architecture. In Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 5–9 December 2021; pp. 1–6.
16. Götzfried, J.; Eckert, M.; Schinzel, S.; Müller, T. Cache attacks on intel sgx. In Proceedings of the 10th European Workshop on Systems Security, Paris, France, 23–24 April 2017; p. 2.
17. Brasser, F.; Müller, U.; Dmitrienko, A.; Kostiainen, K.; Capkun, S.; Sadeghi, A.R. Software grand exposure:SGX cache attacks are practical. In Proceedings of the 11th USENIX Workshop on Offensive Technologies (WOOT17), Vancouver, BC, Canada, 14–15 August 2017; p. 15.

18. Van Bulck, J.; Minkin, M.; Weisse, O.; Genkin, D.; Kasikci, B.; Piessens, F.; Silberstein, M.; Wenisch, T.F.; Yarom, Y.; Strackx, R. Foreshadow: Extracting the keys to the intel sgx kingdom with transient out-of-order execution. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore, MD, USA, 15–17 August 2018; pp. 991–1008.
19. VanNostrand, P.M.; Kyriazis, I.; Cheng, M.; Guo, T.; Walls, R.J. Confidential Deep Learning: Executing Proprietary Models on Untrusted Devices. In Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Austin, TX, USA, 23–27 March 2020; pp. 1–62.
20. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.* **2021**, *115*, 619–640. [CrossRef]
21. Zhang, Y.; Zeng, D.; Luo, J.; Xu, Z.; King, I. A Survey of Trustworthy Federated Learning with Perspectives on Security, Robustness, and Privacy. *arXiv* **2023**, arXiv2302.10637.
22. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16), Vienna, Austria, 24–28 October 2016; pp. 308–318.
23. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowl.-Based Syst.* **2021**, *216*, 106775. [CrossRef]
24. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
25. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.-C.; Yang, Q.; Niyato, D.; Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
26. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A survey on the security of blockchain systems. *Future Gener. Comput. Syst.* **2020**, *107*, 841–853. [CrossRef]
27. Zheng, Z.; Xie, S.; Dai, H.-N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
28. Dai, H.-N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A survey. *IEEE Internet Things J.* **2019**, *6*, 8076–8094. [CrossRef]
29. Maesa, D.D.F.; Mori, P. Blockchain 3.0 applications survey. *J. Parallel Distrib. Comput.* **2020**, *138*, 99–114. [CrossRef]
30. Wang, Y.; Zhang, Y.; Li, Z.; Liu, Y. Integrating blockchain technology into the energy sector—From theory of blockchain to research and application of energy blockchain. *Comput. Sci. Rev.* **2020**, *37*, 100275. [CrossRef]
31. Mirabelli, G.; De Benedetto, L.; Dassisti, M. Blockchain-based solutions for agri-food supply chains: A survey. *Int. J. Simul. Process Model.* **2021**, *17*, 1–15. [CrossRef]
32. Hasselgren, A.; Kralevska, K.; Gligoroski, D.; Pedersen, S.A.; Faxvaag, A. Blockchain in healthcare and health sciences—A scoping review. *Int. J. Med. Inform.* **2020**, *134*, 104040. [CrossRef]
33. Kassen, M. Blockchain and e-government innovation: Automation of public information processes. *Inf. Syst.* **2022**, *103*, 101862. [CrossRef]
34. Dib, O.; Brousmiche, K.-L.; Durand, A.; Thea, E.; Hamida, E.B. Consortium blockchains: Overview, applications and challenges. *Int. J. Adv. Telecommun.* **2018**, *11*, 51–64.
35. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium blockchain for secure energy trading in industrial internet of things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 3690–3700. [CrossRef]
36. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konecny, J.; Mazzocchi, S.; McMahan, H.B.; et al. Towards federated learning at scale: System design. *Proc. Mach. Learn. Syst.* **2019**, *1*, 374–388.
37. Chen, Y.; Luo, F.; Li, T.; Xiang, T.; Liu, Z.; Li, J. A training-integrity privacy-preserving federated learning scheme with trusted execution environment. *Inf. Sci.* **2020**, *522*, 69–79. [CrossRef]
38. Li, Y.; Chen, C.; Liu, N.; Huang, H.; Zheng, Z.; Yan, Q. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Netw.* **2020**, *35*, 234–241. [CrossRef]