Artificial Intelligence and Digital Technology

Article

A Blockchain and Federated Learning Based Model for Supply Chain Credit Risk Assessment

Chenxi Li [1,2,*], Jiayi Liu [1,2], Xinzhe Li [2] and Biying Pei [2]

[1] Center for Artificial Intelligence, Jilin University of Finance and Economics, Changchun, 130117, China
[2] Jilin Province Key Laboratory of Fintech, Jilin University of Finance and Economics, Changchun, 130117, China
* Correspondence: Chenxi Li, Center for Artificial Intelligence, Jilin University of Finance and Economics, Changchun, 130117, China; Jilin Province Key Laboratory of Fintech, Jilin University of Finance and Economics, Changchun, 130117, China

Abstract: This paper proposes a novel blockchain and federated learning fusion model for supply chain credit risk assessment. The focus is on combining privacy protection with collaborative learning. The model combines federated learning with blockchain technology. Multiple participants train a global model collaboratively without sharing raw data. Blockchain technology ensures data immutability and transparency. This fusion protects data privacy and ensures the integrity of the collaborative training process. The study evaluates the model's performance in terms of accuracy, privacy protection, and system performance. It also compares the model with traditional centralized and federated learning models. Experimental results show that the proposed model has significant advantages in privacy protection. The use of differential privacy and blockchain immutability effectively reduces the risk of data leakage. However, there is a tradeoff between privacy protection and model performance. The integration of blockchain slightly affects model accuracy. Furthermore, the study demonstrates the model's robustness under different data distributions and varying numbers of nodes. This proves its effectiveness in real world applications, especially in multi-party collaboration contexts. Finally, the paper discusses challenges in optimizing blockchain performance and applying federated learning in privacy sensitive environments. It also outlines prospects for scalability and application in supply chain finance systems.

Keywords: blockchain; federated learning; supply chain credit risk assessment; privacy protection; differential privacy

Published: 24 April 2025

1. Introduction

1.1. Background and Research Motivation

In recent years, the rapid development of global supply chain finance has brought unprecedented opportunities for corporate financing and risk management [1]. However, it has also raised significant challenges in credit risk assessment. Data among supply chain participants are decentralized and highly sensitive [2]. Although traditional centralized data processing can integrate information, it often faces limitations such as restricted data sharing, high privacy leakage risks, and difficulties in cross-organizational collaboration [3]. These issues directly affect the accuracy and timeliness of credit risk assessments.

Federated learning, an emerging distributed machine learning method, allows each participant to independently train models on local data and share only encrypted parameters or gradients [4]. This approach enables the collaborative construction of a global model while protecting data privacy. Nonetheless, federated learning still encounters challenges in practice, such as uneven data distribution and insufficient transparency in

model update processes, especially in supply chain finance scenarios [5]. Blockchain technology, with its decentralized, immutable, and traceable characteristics, offers a novel solution to these problems [6]. By recording model parameter updates and associated metadata on the blockchain and utilizing smart contracts for automatic verification and incentive mechanisms, the security and trustworthiness of the model aggregation process can be enhanced. This approach effectively mitigates the risks related to data privacy and information security.

Therefore, this study aims to develop a supply chain credit risk assessment model that integrates blockchain and federated learning. The proposed model seeks to improve prediction accuracy and overall system trustworthiness while ensuring data privacy and security for all participants. This method not only provides a new technical pathway for risk management in supply chain finance but also offers theoretical support and practical guidance for addressing distributed data security and cross-organizational collaborative learning challenges.

1.2. Existing Research and Limitations

Credit risk assessment in supply chain finance has been a significant topic in financial technology [7]. Traditional credit evaluation methods primarily rely on centralized data processing and analysis. In this model, data from various participants is usually centralized for unified modeling and analysis to predict credit risk [8]. However, while centralized methods can improve prediction accuracy to some extent, they often face challenges related to data privacy protection, information security, and cross-institutional collaboration. Existing literature identifies data privacy breaches, information asymmetry, and obstacles in data sharing as major limitations of centralized credit evaluation methods [9]. To address these issues, federated learning, an emerging distributed learning approach, has gained widespread attention in recent years. Federated learning allows data to remain local while sharing only model parameters or gradients, avoiding the leakage of raw data and resolving the conflict between privacy protection and data sharing [10]. In the financial sector, federated learning has been applied to tasks such as credit scoring and fraud detection [11]. Research has shown that federated learning can enable collaborative modeling between different financial institutions without violating privacy policies. However, despite the great potential of federated learning, its application in supply chain credit risk assessment remains exploratory [12]. Most existing studies focus on areas like traffic flow prediction and healthcare, with limited research applying federated learning to complex supply chain finance environments. At the same time, blockchain technology, with its decentralized, immutable, and traceable characteristics, has shown significant advantages in data certification, information sharing, and security assurance. The introduction of blockchain can effectively address data privacy issues in traditional centralized models, enhancing the credibility and transparency of information exchange. While blockchain has been widely discussed in the context of supply chain finance, most research focuses on the application of blockchain technology alone, lacking systematic exploration of its integration with other advanced technologies, such as federated learning [13]. In the context of supply chain credit risk assessment, how to combine blockchain and federated learning to not only solve data privacy and security issues but also enable efficient model training and credible verification remains an underexplored research direction [14].

In summary, current research faces several key limitations. Traditional centralized credit evaluation methods have significant flaws in data privacy protection and cross-institutional collaboration. While federated learning shows potential in privacy protection, its application and validation in supply chain credit risk assessment remain insufficient. Additionally, blockchain technology has mainly been applied in data certification and security, with limited research on its integration with federated learning, particularly in enhancing model credibility and verification processes in supply chain credit risk evaluation.

Therefore, combining the technological advantages of blockchain and federated learning to create a distributed model that both ensures data privacy and improves credit risk assessment accuracy is a critical issue in the field of supply chain finance. This study aims to address this issue by integrating these two technologies and testing their effectiveness in practical applications.

1.3. Research Objectives and Innovative Contributions

This study aims to propose a supply chain credit risk assessment model that integrates blockchain and federated learning, addressing the shortcomings of existing models in terms of data privacy protection, cross-institutional collaboration, and information security. Specifically, the primary goal of this study is to develop a distributed model that can accurately and efficiently assess credit risk in supply chain finance while ensuring data privacy and security.

First, the proposed model utilizes federated learning to ensure that data remains local to each participant, with only encrypted model parameters or gradients shared. This effectively protects sensitive data while still enabling collaborative training to improve model performance. Compared to traditional centralized models, federated learning overcomes the limitations of data sharing and leverages local data characteristics from multiple participants, thereby enhancing the model's generalization ability.

Second, this study innovatively introduces blockchain technology into the federated learning framework. By utilizing blockchain's immutability and traceability features, the model ensures the security and credibility of parameter exchanges during the training process. Blockchain not only records the model updates from each participant during aggregation but also uses smart contracts to ensure the transparency and consistency of these updates, thereby enhancing the overall system's trustworthiness and reliability.

The innovative contributions of this study are as follows:

1) The introduction of a novel framework combining blockchain and federated learning, which ensures data privacy and the security of model updates through blockchain, while leveraging federated learning to improve model accuracy and collaborative training efficiency.
2) The design of a distributed credit risk assessment model tailored for supply chain finance, filling the gap in research on the application of blockchain and federated learning in this field.
3) The proposal of a reputation-based incentive mechanism and the use of smart contracts in the model training process, which further enhance the collaboration and stability of multi-party participation in the model aggregation process.

Thus, this study not only provides a new technical solution for credit risk assessment in supply chain finance but also offers significant theoretical and practical insights for the combined application of blockchain and federated learning across various domains.

1.4. Related Work

In recent years, supply chain finance has grown rapidly. Credit risk assessment faces challenges in data privacy and secure sharing among multiple parties. Existing research focuses on three directions. The first is blockchain applications in supply chain finance. The second is traditional and ensemble credit risk methods. The third is research on combining blockchain and federated learning.

1.4.1. Blockchain in Supply Chain Finance

Blockchain offers decentralization, immutability, and traceability. These features support information security and process transparency in supply chain finance. Wang et al. used a three-tier supply chain model and game theory to study equilibrium strategies in blockchain-driven accounts receivable chains [15]. Amini et al. proposed a decentralized clearing mechanism. They used smart contracts to automate claim resolution and

payment verification [16]. Li et al. designed Fabric-SCF [17]. This system uses distributed consensus, attribute-based access control, and smart contracts to secure data storage and access in supply chains. These studies highlight blockchain's role in data integrity and trust. However, they do not address privacy-preserving collaborative models like federated learning.

1.4.2. Credit Risk Assessment Methods

Credit risk assessment has evolved from single classifiers to ensemble learning. Xia et al. combined bagging and stacking [18]. Their heterogeneous ensemble credit model improved prediction through pool generation and a trainable fusion. Pławiak et al. developed a deep genetic cascade ensemble of SVMs [19]. This method used multiple kernels and parameter tuning to enhance stability and accuracy. Dumitrescu et al. proposed penalized logistic tree regression (PLTR) [20]. This method integrated decision tree outputs into logistic regression. It balanced interpretability and performance. These methods improve accuracy but rely on centralized data sharing. They do not meet privacy and compliance needs in multi-party settings.

1.4.3. Blockchain and Federated Learning Integration

Some studies combine blockchain and federated learning to address privacy risks. Imteaj & Amini applied federated learning for credit default prediction [21]. They balanced model accuracy and data privacy. Cheng et al. analyzed Secure Boost [22]. They proved that this federated boosting model matches non-federated performance. Others record model parameter hashes on the blockchain. This practice ensures immutability and auditability of training. However, these approaches focus on unstructured scenarios or single attack defenses. They lack an end-to-end design that integrates differential privacy noise, on-chain hash storage, and smart contract validation for supply chain credit risk assessment.

Overall, prior work has advanced secure storage with blockchain, ensemble models, and privacy in federated learning. Yet, there is no unified solution for supply chain credit risk assessment. This paper fills the gap. It proposes a blockchain and federated learning framework. It adds differential privacy noise to local updates. It stores only parameter hashes on-chain and uses smart contracts for integrity checks. This design supports multi-party credit risk assessment.

2. Materials and Methods

2.1. Model Architecture Overview

This study presents a supply chain credit risk assessment model that integrates two cutting-edge technologies: blockchain and federated learning. The goal is to provide an efficient, transparent, and data-privacy-preserving collaborative training platform. By combining blockchain and federated learning, the system ensures data security, transparency, and fairness for all participants, while also optimizing the performance of the global model.

The model architecture consists of two main components: the blockchain module and the federated learning module. The blockchain module is responsible for recording and verifying the model updates of all participants, ensuring the transparency and immutability of each update. The federated learning module coordinates various participants (such as core enterprises, suppliers, banks, etc.) to train models based on their own local data, using a global aggregation algorithm to update the model. The combination of these components allows participants to collaborate and improve the global model without directly exchanging local data, ensuring both data privacy protection and system fairness.

The federated learning and blockchain co-training process is as follows:

1) Local Model Training: Each participant independently trains a model on their local dataset. After each training round, the participant computes the local model parameters and prepares to upload them.
2) Model Parameter Upload and Validation: The uploaded local model parameters are hashed and recorded on the blockchain. After uploading, the model parameters are validated by a smart contract to ensure their legality and accuracy, preventing any malicious tampering.
3) Global Model Aggregation: All uploaded local model parameters are aggregated using the FedAvg algorithm, which computes a weighted average to generate the global model. Blockchain ensures the transparency of the model update process, with each participant's contribution being properly recorded.
4) Model Update and Sharing: After the global model is aggregated, the model parameters and associated metadata are shared with all participants through the blockchain. This ensures that every participant receives the updated global model. The entire process is also validated by the smart contract to ensure that the uploaded model updates are legitimate and that each participant's contribution is acknowledged.

The model architecture is shown in Figure 1.

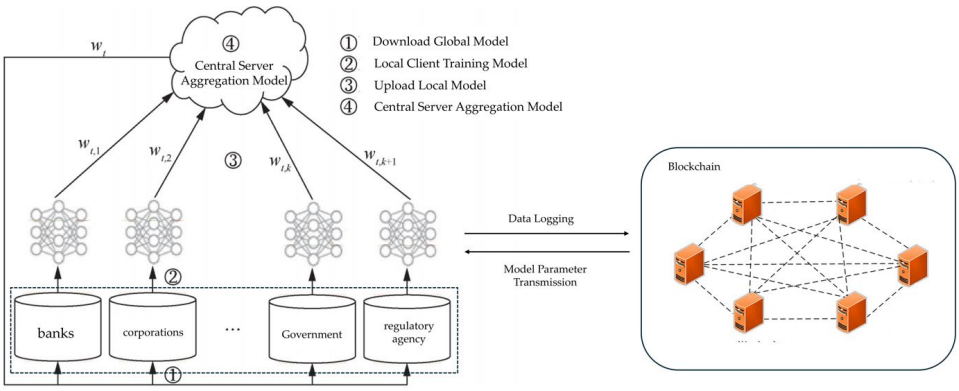

Figure 1. Model Architecture.

2.2. Blockchain Module Design

In this study, the blockchain module is designed to ensure the security, transparency, and immutability of the model update process within the supply chain credit risk assessment model. Blockchain is primarily used to record and validate the updates of model parameters, ensuring that each update is legitimate and unalterable.

The blockchain module operates by recording each model update in a block, which is then added to the blockchain. Each block is linked to its predecessor, creating a chain structure that guarantees the immutability of the data. This section describes the specific design and implementation of the blockchain module, focusing on the block structure, the role of smart contracts in model updates, and the provenance and anti-tampering mechanisms for model parameters.

2.2.1. Blockchain Structure

Each model update generates a new block in the blockchain. The block is divided into two main parts: the block header and the block body.

The block header contains essential information that ensures the integrity and security of the blockchain. It includes the Previous Block Hash, which stores the hash of the previous block, ensuring the chain structure and maintaining the sequence and immutability of the data. The Current Block Hash uniquely identifies the current block, preventing any tampering with the data. The Timestamp records the exact time of each model

update, ensuring that each operation can be traced. Additionally, the Node ID is included to identify which node (e.g., a bank or a supplier) submitted the model update. The hash value $H(B_i)$ of each block B_i is calculated as:

$$H(B_i) = H(\text{Previous Block Hash, Timestamp, Node ID, Model Parameters Hash}) \quad (1)$$

This formula guarantees the uniqueness and security of each block, linking it to its predecessor and preventing any modifications.

The block body contains detailed information about the model update. The Model Parameters Hash stores the hash of the updated model parameters. The SHA-256 hashing algorithm is used to ensure the immutability of these parameters. The Model Version field records the version of the model with each update, maintaining version control. The Update Log captures the specific changes made during the update process, ensuring transparency and traceability. For instance, if θ new represents the current model's parameters, the hash of these parameters is calculated using the SHA-256 algorithm:

$$H(\theta_{new}) = \text{SHA} - 256(\theta_{new}) \quad (2)$$

The block body stores this hash value $H(\theta_{new})$, the model version, and the update log, ensuring that each update is accurately recorded and retrievable.

2.2.2. Role of Smart Contracts in Model Update and Validation

Smart contracts in the blockchain module automatically validate the legitimacy of each model update, ensuring that every update adheres to predefined rules. The smart contract eliminates the need for manual intervention, improving the efficiency and fairness of the update process.

The smart contract performs the following functions:

1) Validation of Model Parameters: It checks whether the uploaded model parameters meet the required format and dimension standards.
2) Validation of Update Legality: It ensures that the model update adheres to predefined rules, such as permissible parameter changes and model training guidelines.
3) Incentive Mechanism: The smart contract manages the incentive system, rewarding nodes that contribute to the model training based on their update frequency, accuracy, and other performance metrics.

The smart contract verification process is represented by the following pseudocode:

$$\text{If } \theta_{new} \in \mathbb{R}^n \text{ and ValidUpdate}(\theta_{new}) \text{ then ExecuteUpdate } (\theta_{new}) \quad (3)$$

Here, ValidUpdate ensures that the model parameters are formatted correctly, and ExecuteUpdate applies the valid update.

2.2.3. Model Parameter Provenance and Anti-Tampering Mechanism

The decentralized and immutable nature of blockchain ensures that the provenance of model parameters is recorded in a secure manner. Each time a model's parameters are updated, the hash of these parameters is stored on the blockchain, ensuring that each model update is traceable and immutable.

Encrypted Hash Provenance: Every locally trained model parameter θ_i is hashed using the SHA-256 algorithm, generating a hash value $H(\theta_i)$, which is stored on the blockchain. This guarantees the integrity and authenticity of the model parameters:

$$H(\theta_i) = \text{SHA} - 256(\theta_i) \quad (4)$$

Anti-Tampering Mechanism: Since each block contains the hash of the previous block, any attempt to tamper with the data would alter the hash, disrupting the blockchain structure. This triggers an alert system that ensures the integrity of the data. The combination of hash algorithms and blockchain's linked structure offers a robust defense against tampering.

2.3. Federated Learning Module Design

2.3.1. Local Model Architecture

In this study, we adopt the Multilayer Perceptron (MLP) as the local model architecture for each participant. MLP is a classical deep learning model consisting of multiple fully connected layers, where each neuron is connected to all neurons in the previous layer. The model is designed with an input layer, several hidden layers, and an output layer.

Each participant trains its local model based on its local dataset D_i. During each training step, the participant updates its local model parameters w_i based on the loss function $L(w)$ using gradient descent. The parameter update rule at the t-th round of training is as follows:

$$w_i^{(t)} = w_i^{(t-1)} - \alpha \nabla L(w_i^{(t-1)}, D_i) \tag{5}$$

Where $w_i^{(t)}$ is the model weight of participant ii at the tt-th round, α is the learning rate, and $\nabla L(w_i^{(t-1)}, D_i)$ is the gradient of the loss function computed over the local dataset D_i.

The local model consists of an input layer that receives the features of the local data, hidden layers using activation functions to perform nonlinear mappings, and an output layer that produces the final prediction. The architecture of the MLP may vary depending on the complexity and characteristics of the data. Through this design, each participant independently performs model training and generates a local model without exposing their local data.

The model formulas are shown in Figure 2.

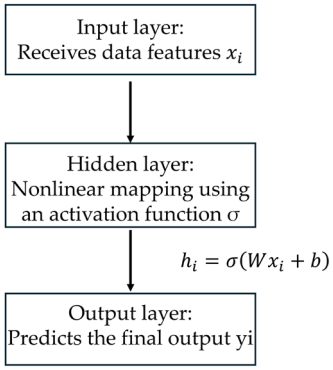

Figure 2. Model Formulas.

2.3.2. FedAvg Aggregation Strategy and Communication Mechanism

In federated learning, the aggregation strategy determines how to combine the local model updates from multiple participants into a global model. In this study, we adopt the FedAvg (Federated Averaging) algorithm as the aggregation strategy. This algorithm performs weighted averaging of the local model parameters from each participant to generate the global model.

Let $w_i^{(t)}$ be the local model parameters of participant i at the t-th round. The global model is updated by performing a weighted average of all local models:

$$w^{(t)} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{|D_i|}{\sum_{j=1}^{N} |D_j|} \cdot w_i^{(t)} \right) \tag{6}$$

Where N is the number of participants, $|D_i|$ is the size of the local dataset of participant i, and $w_i^{(t)}$ is the local model weights after the tt-th round of training.

After each round of training, each participant uploads its updated model parameters to a central server or blockchain for verification. The server or blockchain aggregates all uploaded local models to generate the global model, which is then distributed back to all

participants for the next training round. The FedAvg algorithm ensures that the local updates are aggregated fairly based on the data size, and prevents any data leakage in a centralized training setup.

The algorithm flow of FedAvg is shown in Figure 3.

Algorithm Federated Averaging

Server executes:initialize w_0 **for** each round $t = 1, 2, \ldots$ **do**

S_t = (random set of $\max(\cdot K, 1)$ clients)

for each client $k \in S_t$ **in parallel do**

$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

end for

$w_{t+1} \leftarrow \sum_{t=1}^{n_k} w_{t+1}^k$

end for

ClientUpdate(k, w): *// Executed on client* k

for each local epoch i from 1 to **do**

batches $\leftarrow$ (data $_k$ split into batches of size)

for batch b in batches **do**

$w \leftarrow w - \eta(w; b)$

end for

end for

return w to server =0

Figure 3. Algorithmic Flow of FedAvg.

2.3.3. Model Update Cycle and Coordination Mechanism

The model update cycle and coordination mechanism are crucial for maintaining the stability and collaboration of federated learning. In this study, we adopt a periodic update mechanism. Each participant performs several rounds of local training on its own dataset, and after completing the training, the local model parameters are uploaded to the server or blockchain for aggregation.

The local training and global update cycles are coordinated as follows:

1) Local training: Each participant trains its local model over several rounds using its local dataset D_i. The training process is independent, and no data exchange occurs between participants.
2) Model upload: After local training, the participant uploads the updated model parameters $w_i^{(t)}$ to the blockchain or central server for validation.
3) Global aggregation: The server or blockchain performs weighted averaging of the uploaded local models to generate the new global model $w^{(t)}$, which is then distributed to all participants.
4) Model distribution**:** Once the global model is updated, the new global model is distributed back to all participants for the next round of training.

The update cycle ensures that all participants collaborate in training the model, and the use of blockchain ensures the privacy and security of the model updates. The synchronization of local training and global model updates is achieved through the coordination between the server or blockchain, which manages the training cycles and maintains fairness.

2.3.4. Smart Contract and Encryption Algorithm

In this section, we describe in detail how differential privacy and smart contracts are employed in the blockchain-based federated learning framework to ensure the security, transparency, and data privacy protection of the supply chain credit risk assessment model. Differential privacy is used to protect the local data privacy of each participant,

while the smart contract is responsible for verifying the legality of model updates, controlling the model upload process, and implementing an incentive mechanism.

1) Differential privacy noise injection mechanism

To safeguard participant privacy, we apply differential privacy to inject noise into model parameters for each local model update. Specifically, before each model parameter upload, the participant adds noise to its local model parameters W_i according to the Laplace mechanism. Assuming the privacy budget is ϵ and the model parameter sensitivity is Δ, noise η is drawn from a Laplace distribution, and the noise scale is $b = \frac{\Delta}{\epsilon}$. The formula is as follows:

$$\widetilde{W}_i = W_i + \eta, \eta \sim \text{Laplace}(0, \frac{\Delta}{\epsilon}) \tag{7}$$

Where W_i is the local model parameter of participant ii, and $\widetilde{W}_i$ is the noised model parameter. Through differential privacy, each participant's contribution does not reveal sensitive information about its local data, and the impact of each model update is constrained within a controllable range, effectively protecting participant data privacy.

2) Application of smart contracts

In this study, the smart contract is deployed on the Hyperledger Fabric blockchain platform and is primarily used for model update verification, upload control, and incentive management. The smart contract's functions include:

a) Model parameter verification: Each model update is verified by the smart contract, ensuring that the uploaded model parameters meet preset format requirements and have not been maliciously tampered with. The smart contract checks whether the uploaded model parameters fall within reasonable bounds, ensuring that each submitted update conforms to the training specifications.

b) Upload control: The smart contract controls the upload behavior of each participant, ensuring that each participant can only submit one model update per training round and preventing unauthorized nodes from uploading model parameters. For each model submission, it first verifies that the participant has upload permission and checks whether the participant has already submitted an update during the current round.

c) Incentive mechanism: To encourage active contributions, the smart contract also calculates and distributes rewards. Each participant receives rewards based on the quality of its model update and its contribution to the global model. The reward allocation is determined by the participant's contribution, such as the model's accuracy on a validation set. The formula is as follows:

$$R_i \leftarrow \alpha \cdot R_i + (1 - \alpha) \cdot f(Acc_i) \tag{8}$$

Where R_i is the reputation score of participants i, α is the weighting coefficient, Acc_i is the model accuracy of participant ii in this training round, and $f(Acc_i)$ is the reputation increment calculated from the model accuracy. The smart contract allocates rewards to each participant based on its reputation score.

3) Smart contract pseudocode

The main logic of the smart contract includes verifying the legality of the uploaded model parameters, limiting upload frequency, calculating rewards, and distributing them.

Figure 4 below illustrates the smart contract execution process.

Algorithm Smart Contract Pseudo-code: Model Update Submission

```
function SUBMITMODELUPDATE(participant_id, model_update)
    if (participant_id not in AuthorizedParticipants) or (State[participant_id].submitted_in_round_r) then
        return Reject("Unauthorized or duplicate submission")
    end if
    if (not verify_format (model_update) ) or (not within_bounds (model_update) ) then
        return Reject("Invalid model parameters")
    end if
    if evaluate_accuracy(model_update) ¡ eps_min then
        return Reject("Model accuracy below threshold")
    end if
    State[participant_id].submitted_in_round_r = true
    Ledger.append(hash(model_update), participant_id)
    State[participant_id].temp_model = model_update
    acc = evaluate_accuracy(model_update)
    repDelta = f(acc)
    State[participant_id].new_reputation = alpha * State [participant_id] . reputation + (1 - alpha) * repDelta
    State[participant_id].reward = g(repDelta)
    return Accept("Update received")
end function
if (all participants submitted) or (round_timeout) then
    Model_Global_new = aggregate({State[*].temp_model})
    Ledger.store(Model_Global_new)
    for each participant i do
        finalize(State[i].reputation = State[i].new_reputation)
        transfer Token(State[i].reward) to participant_i
        State[i].submitted_in_round_r = false
    end for
    broadcast(Model_Global_new)
end if
```

Figure 4. The process of executing smart contracts.

In the pseudocode, $(verify\ format)$ and $(within\ bounds)$ validate the format and value range of the model parameters, while $(evaluate\ accuracy)$ computes the accuracy of the submitted model. Functions $f(acc)$ and $g(\Delta rep)$ calculate the reputation increment and reward amounts, respectively. Through smart contract execution, every participant's contribution is recorded in a transparent and verifiable manner, and the incentive mechanism ensures participants remain motivated and committed.

4) Synergy between differential privacy and smart contracts

Differential privacy and smart contracts work together to guarantee both privacy protection and regulatory compliance in federated learning. Differential privacy ensures that each model update does not leak sensitive data from any participant, while the smart contract enforces compliance during the upload process, validates the model updates, and fairly distributes rewards based on the quality of the contributions. This design achieves privacy protection, transparent model updates, and active multi-party participation, ultimately strengthening the reliability of supply chain credit risk assessment.

2.4. Model Collaboration and Fusion Mechanism

In this study, the design of the model collaboration and fusion mechanism aims to ensure that multiple parties in the blockchain-federated learning framework can efficiently and transparently perform model training and updates, while ensuring the correct aggregation of model parameters and fairness in the participation of multiple parties. This section describes the model synchronization mechanism during federated training, the interaction process of model parameters and metadata in the blockchain, and the reputation-based incentive mechanism and node participation control.

2.4.1. Model Synchronization Mechanism during Federated Training

In federated learning, multiple parties (such as banks, core enterprises, suppliers, etc.) train models based on their local data. After each round of training, the updated model

parameters need to be uploaded to the server for aggregation. To ensure the synchronization of models among multiple parties, the FedAvg aggregation strategy is adopted. This strategy combines the local model parameters from each party by weighted averaging to generate a global model.

In each round of training, each node first performs local training based on its own data to obtain local model parameters W_i. These parameters are then uploaded to the blockchain via a secure channel for record-keeping. All uploaded model parameters are validated by a smart contract to ensure their legality and accuracy. Then, using the FedAvg algorithm, the server aggregates the local model updates from each party to generate the global model W_{global}, as shown in the following formula:

$$W_{\text{global}} = \sum_{i=1}^{n} \frac{n_i}{N} \cdot W_i \tag{9}$$

Where n_i is the local data size of the iii-th node, N is the total data size across all nodes, and W_i is the local model parameters of the iii-th node. This method allows the models to collaborate effectively while ensuring the precision and consistency of the global model under the premise of data privacy protection.

2.4.2. Interaction of Model Parameters and Metadata in Blockchain

Blockchain technology plays a crucial role in ensuring the transparency and immutability of model parameter updates in this study. In each round of federated learning training, the local model updates from each party are recorded on the blockchain. Specifically, each party uploads the hash value $W_i(\text{i.e.}, H(W_i))$ of their local model parameters W_i, along with other relevant metadata (such as node identifier, timestamp, model version) to the blockchain. Smart contracts play a key role in this process by verifying the legitimacy of the uploaded model parameters and executing necessary control logic. The interaction process can be outlined as follows:

1) Each participating party calculates the hash value $H(W_i)$ of its local model parameters and uploads it along with other metadata (such as the participant ID, timestamp) to the blockchain.
2) The smart contract verifies whether the uploaded model parameters conform to the predefined format and checks for any malicious tampering.
3) Once the model parameters are validated, the blockchain records the transaction and generates a new block for the model update.
4) The server retrieves the model hashes of all parties from the blockchain, aggregates the local model parameters using the FedAvg algorithm, and generates the global model.

This blockchain-based interaction mechanism not only enhances the transparency of data exchange but also ensures the traceability and immutability of the model update process, thereby improving the reliability of the supply chain credit risk assessment.

2.4.3. Reputation Incentive and Node Participation Control

To encourage active participation from each party and ensure high-quality model updates, a reputation-based incentive mechanism is introduced. The smart contract calculates the reputation score R_i for each node based on its contribution in the model training process (e.g., model accuracy). The reputation score of a node directly affects its future participation opportunities and the amount of reward it receives. The implementation of the reputation incentive mechanism is as follows:

1) Reputation score calculation: Each participating node calculates its reputation increment based on the accuracy of the model it uploads. The reputation score R_i is updated after each round of training as:
$$R_i^{new} = \alpha \cdot R_i^{old} + (1 - \alpha) \cdot f(Acc_i) \tag{10}$$

Where α is a weighting coefficient, R_i^{old} is the previous round's reputation score, and $f(Acc_i)$ is the reputation increment calculated from the model's accuracy Acc_i.

2) Upload control and reward distribution: The smart contract controls the upload behavior of each participant, ensuring that each party can only submit one model update per training round and prevents unauthorized nodes from uploading model parameters. Each model update is validated for the participant's upload permission and whether they have already submitted an update in the current training round.

Through this mechanism, each node is incentivized to provide higher-quality model updates based on its contribution, promoting faster convergence of the model and improving its overall performance.

3. Results

3.1. Experimental Design

This study designs several experiments to evaluate the effectiveness of the blockchain and federated learning-based supply chain credit risk assessment model. The experiments cover various aspects, including data privacy protection, model accuracy, and the transparency and security of multi-party collaboration. The company financial data used in the experiments comes from multiple publicly available financial databases, including Wind Information, Bloomberg, and Reuters platforms. These datasets contain 35 financial indicators and 67,900 data records, covering key financial parameters such as the current ratio, asset growth rate, and profit growth rate. These data ensure the reliability and repeatability of the experimental results.

The experiments in this study are conducted on the Hyperledger Fabric blockchain platform, combined with the federated learning framework. The study simulates collaborative training among multiple roles in the supply chain (such as core enterprises, banks, suppliers, etc.) without exchanging data. The experimental platform uses Hyperledger Fabric 1.4.1 as the blockchain platform, with Golang for developing smart contracts and Python 3.8 for training the federated learning models. All experiments are deployed in a Docker environment to ensure the collaboration of different nodes and the protection of data privacy.

To verify the advantages of combining blockchain and federated learning, this study compares the proposed federated learning + blockchain model with three other models. First, the centralized model concentrates all data on a single server for model training, using traditional centralized machine learning methods. In this model, all participants upload data to the central server, where the training and updates occur. In the federated learning model, each participant independently trains models based on local data and then uploads the model parameters to the central server for aggregation. In this method, the data remains local and is not exchanged, ensuring data privacy. Finally, the federated learning + blockchain model combines blockchain technology within the federated learning framework for model training and updating. It ensures data privacy protection while validating and recording each model update through blockchain, ensuring transparency and immutability.

To comprehensively evaluate the experimental results, this study selects the following evaluation metrics. Accuracy is the most commonly used metric to assess the overall predictive ability of the model on the test dataset. F1 score considers both precision and recall, making it particularly suitable for evaluating models on imbalanced datasets. AUC measures the model's classification ability across different thresholds, helping to assess its performance in complex scenarios, especially for imbalanced categories. Data leakage rate evaluates whether there is a risk of data leakage during the training and updating process, which is a key indicator for testing privacy protection capabilities.

3.2. Experiment 1: Performance Comparison of Multiple Models in Supply Chain Credit Risk Assessment

This experiment compares the performance of the Centralized Model, Federated Learning Model, and Federated Learning + Blockchain Model in supply chain credit risk assessment, with a focus on accuracy and privacy protection. The comparison evaluates whether the Federated Learning Model can maintain accuracy comparable to the traditional Centralized Model while preserving data privacy. It also explores the enhanced privacy protection achieved by integrating Federated Learning and Blockchain.

In the experiment, the Centralized Model uploads all data to a central server for training, which has strong fitting capabilities but lacks privacy protection. The Federated Learning Model adopts distributed training, with each participant independently training a model on local data and only uploading encrypted model parameters for aggregation, ensuring data privacy. The Federated Learning + Blockchain Model further integrates blockchain technology to ensure data immutability while providing privacy protection and recording verification information for each model update.

As shown in Table 1 and Figure 5, despite the introduction of blockchain technology, the Federated Learning + Blockchain model has a slightly lower accuracy than the centralized model and the federated learning model. However, its loss convergence is more stable, showing a better training process. In terms of privacy protection, the Federated Learning + Blockchain model shows a significant advantage by ensuring the immutability and transparency of data, enhancing the credibility of multi-party collaboration and data security.

Table 1. Comparison of Results from Multiple Models in Credit Risk Assessment.

Model	Accuracy (%)	Loss	F1 (%)
Centralized Model	88.5	0.15	88.6
Federated Learning Model	87.8	0.17	88.0
Federated Learning + Blockchain	87.5	0.18	87.5

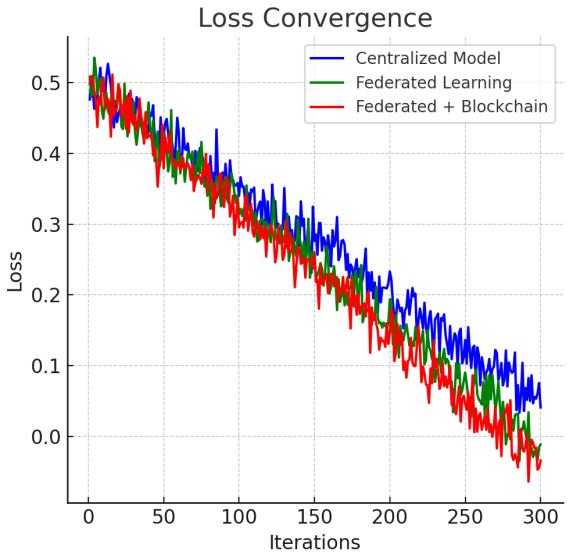

Figure 5. Loss Rate Curve.

3.3. Experiment 2: Verification of Multi-Node Collaborative Training and Model Aggregation

This experiment aims to validate the effectiveness of the proposed Federated Learning + Blockchain model. The model involves multiple nodes participating collaboratively in training. It also evaluates the performance of the FedAvg aggregation strategy. This strategy generates a global model under different scenarios. The context is specifically

supplying chain credit risk assessment. We examine the model's convergence, final performance, and robustness. We do this by varying the number of participating nodes. We also simulate heterogeneous (Non-IID) data distributions.

The experiment focuses on the performance of model aggregation. It examines the impact of the number of participating nodes on training outcomes. These outcomes include convergence speed and final accuracy. The study also investigates the model's robustness under Non-IID data distributions. Analysis compares the global model from collaborative training with local models trained individually by each node. We evaluate improvements in accuracy and generalization ability. We also verify the role of model aggregation in effectively integrating multi-party knowledge. This integration aims to enhance overall assessment effectiveness.

In the experiment, we simulate multiple supply chain participants. These participants act as federated learning nodes (e.g., core enterprises, suppliers, banks). Each node uses its local private data to train a model. Nodes submit model updates according to a predefined communication protocol. This process might be coordinated via blockchain. An aggregation mechanism, such as FedAvg, integrates model contributions from different nodes. It generates an updated global model. This global model is then distributed back to the nodes for the next training round. This iterative process of collaborative training and aggregation aims to create a shared global model. The target model should synthesize information from all parties, exhibit superior performance, and possess good generalization capabilities for credit risk assessment.

Experiment 2 verified the effectiveness of multi-node training and aggregation within the proposed fusion framework. We simulated scenarios with varying node counts (N = 3, 5, 10). We also used different data distributions (IID, low Non-IID, high Non-IID). The FedAvg algorithm was used for model aggregation. (Parameter exchange relies on the framework's design; its trustworthiness is verified in subsequent experiments). The experiment primarily evaluated key performance indicators like AUC, Accuracy, and F1-score. We also analyzed model convergence speed and cross-node performance consistency. Detailed data are presented in Table 2 and Figure 6. The results consistently showed that the global model from collaborative training significantly outperformed the average performance of models trained locally in isolation. For instance, under the N = 5 IID setting, the global AUC was 0.852, while the average local AUC was only 0.741. Increasing the number of nodes under IID conditions yielded slight performance improvements (e.g., N = 10 IID AUC reached 0.861). However, data heterogeneity (Non-IID) led to decreased final model performance (e.g., N = 5 High Non-IID AUC dropped to 0.813). It also slowed down convergence. Non-IID conditions also increased performance variance among nodes. Nevertheless, the federated learning approach still showed significant advantages over purely local training. This holds true even under challenging high Non-IID conditions.

Table 2. Comparison Results with Different Number of Nodes and Data Distribution.

Scenario	AUC	Accuracy	F1-Score	Convergence Rounds	Std Dev of Local AUCs
Avg. Local Model (N = 5)	0.741	0.832	0.758	N/A	N/A
N = 3, IID	0.845	0.908	0.855	58	0.011
N = 5, IID	0.852	0.915	0.866	55	0.009
N = 10, IID	0.861	0.92	0.871	52	0.008
N = 5, Low Non-IID (α = 1)	0.831	0.897	0.84	68	0.025
N = 5, High Non-IID (α = 0.1)	0.813	0.885	0.821	80	0.048

N=10, Low Non-IID (α=1)	0.839	0.901	0.847	65	0.022

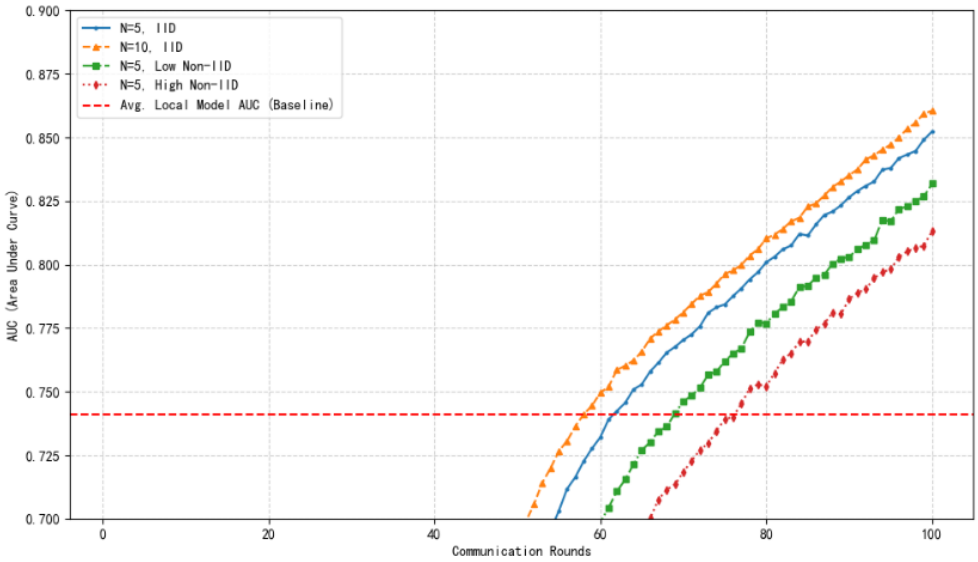

Figure 6. Accuracy Comparison Chart.

3.4. Experiment 3: Privacy Protection Capability Evaluation

This experiment aims to quantitatively evaluate the privacy protection capability of the Federated Learning model, which integrates differential privacy and blockchain, in the application of supply chain credit risk assessment. The focus is on examining the core privacy mechanism — differential privacy (Laplace mechanism) applied to local model parameters and the design of storing only parameter hash values on the blockchain — in preventing sensitive data leakage during collaborative training. By simulating privacy attack scenarios and using relevant privacy metrics, this experiment seeks to measure the advantages of this model in protecting participant data privacy, compared to unprotected federated learning or traditional centralized methods.

The evaluation focuses on the following aspects:

1) The effectiveness of the differential privacy mechanism, which analyzes the degree of protection of the local training data of participants under different privacy budget ε settings.
2) The model's resistance to privacy attacks, such as Membership Inference Attacks (MIA), especially when the attacker may have access to noisy parameters $\widetilde{w}_i$ in transmission or can analyze on-chain metadata and parameter hash records.
3) The contribution of blockchain's process integrity (through on-chain hash verification and smart contract rule execution) to prevent malicious behaviors that may indirectly lead to privacy leakage (such as invalid parameter injection to probe the system).

In this evaluation, federated learning retains the original data locally, providing basic privacy protection. The core evaluation target is the effect of differential privacy (Laplace mechanism) on adding noise to local model parameters $\widetilde{w}_i$ before uploading them to generate noisy parameters $\widetilde{w}_i$. Additionally, the evaluation examines the role of storing only parameter hashes $H(\widetilde{w}_i)$ on the blockchain, rather than the parameters themselves, in reducing the risk of direct information leakage on-chain. The evaluation also incorporates an analysis of how smart contracts validate the format, range, and upload behavior of noisy parameters $\widetilde{w}_i$, ensuring process compliance and increasing the difficulty of manipulation by attackers. The overall privacy protection capability will be assessed by simulating different attackers (for example, an "honest but curious" aggregator or other participants attempting to infer information from $\widetilde{w}_i$ or on-chain records), calculating attack

success rates, measuring information leakage, and comparing the model's utility loss under different privacy budget ε values.

3.4.1. Privacy-Utility Trade-off Analysis

This experiment aims to verify the impact of the differential privacy mechanism on the global model's utility. We set different privacy budget ε values ($\varepsilon = 0.1, 0.5, 1, 2, 5, and\ \infty$, where $\varepsilon = \infty$ represents the baseline without differential privacy protection). Under fixed hyperparameters, the FL + DP + Blockchain framework is used to train the model and evaluate the global model's main performance metrics on a standard test set. These metrics include accuracy, F1 score, AUC, and the number of convergence rounds. The experimental results are summarized in Table 3:

Table 3. Privacy-Utility Trade-off Comparison Results.

Privacy Budget (ε)	Global Accuracy (%)	Global F1 Score (%)	Global AUC	Convergence Rounds
0.1	82.3	81.5	0.83	90
0.5	84.7	84	0.85	85
1	86.1	85.4	0.87	80
2	87	86.5	0.88	75
5	87.8	87.1	0.89	70
∞ (Baseline)	88.5	88	0.91	65

The Figure 7 shows that as the privacy budget ε increases (i.e., as privacy protection decreases), the global model's accuracy, F1 score, and AUC gradually improve, while the number of convergence rounds decreases. This indicates that under weaker privacy protection, the model can achieve higher utility more quickly; conversely, under stronger privacy protection, the global model's utility slightly decreases. To further illustrate the impact of the privacy budget on model utility, we plotted the following charts:

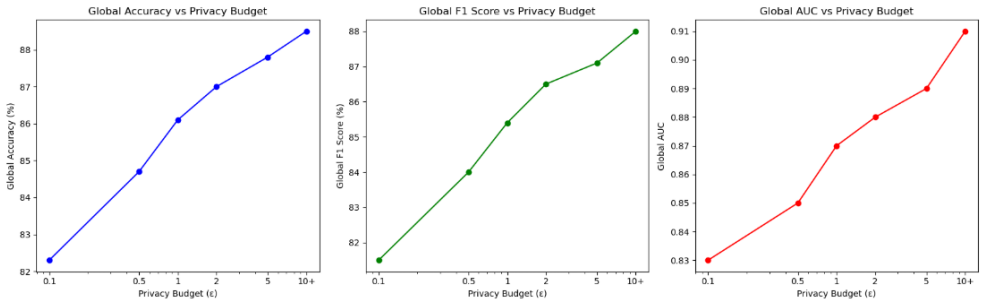

Figure 7. Accuracy, F1 score and AUC under different privacy budgets.

3.4.2. Evaluation of Resistance to Membership Inference Attacks

This experiment aims to evaluate the model's resistance to membership inference attacks after applying a differential privacy mechanism (Laplace mechanism). The experiment trains the model under different privacy budget ε values (0.1, 0.5, 1, 2, 5, and the baseline condition without differential privacy, i.e., $\varepsilon = \infty$) and then simulates a membership inference attack scenario. The attacker tries to determine whether a specific data record was included in the training dataset. The evaluation metric is the success rate of the membership inference attack; a lower success rate indicates better privacy protection.

Table 4 of the experimental results shows that, as the ε value increases (i.e., the strength of privacy protection decreases), the success rate of membership inference attacks significantly increases. For example, when $\varepsilon = 0.1$, the attack success rate is only about 10.5%, while under the baseline condition ($\varepsilon = \infty$) the success rate rises to 45.0%. This

trend indicates that a lower privacy budget can effectively reduce the risk of membership inference attacks, thereby enhancing the protection of the participants' data privacy.

Table 4. Resistance to Membership Inference Attacks under Differential Privacy.

Privacy Budget (ε)	Membership Inference Attack Success Rate (%)
0.1	10.5
0.5	15.2
1	20.4
2	25.7
5	31.3
∞ (Baseline)	45

The figures clearly display a negative correlation between the privacy budget and the attack success rate. This provides a theoretical basis for selecting an appropriate privacy budget in practical applications. Overall, the experimental results demonstrate the effectiveness of the differential privacy mechanism in the model, as it significantly reduces the success rate of membership inference attacks and plays a key role in protecting data privacy. Figure 8 shows the success rate under different privacy budgets:

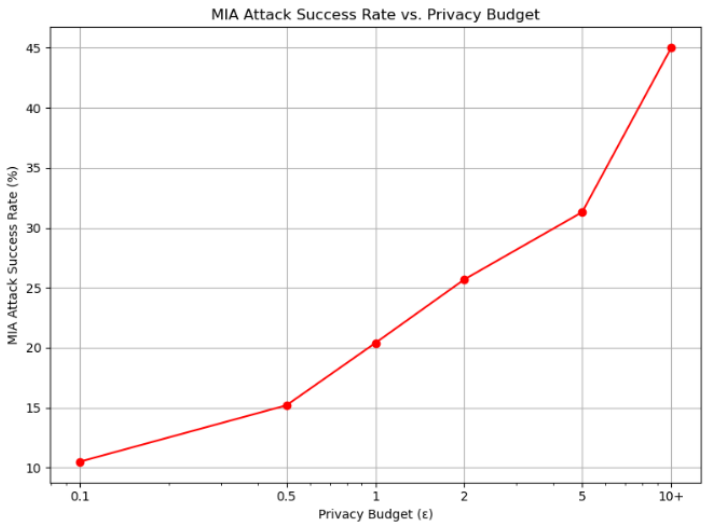

Figure 8. Success Rates under Different Privacy Budgets.

3.4.3. Contribution of Blockchain Integrity to Privacy Protection

This experiment aims to verify the role of blockchain mechanisms in preventing malicious attacks. The experiment simulates two attack scenarios, namely tampering attacks and replay attacks, to analyze the effectiveness of storing parameter hashes on the blockchain and verifying them using smart contracts to ensure the integrity of the model update process. A qualitative analysis is also provided for the proof-of-existence aspect. This demonstrates how immutable records on the blockchain indirectly enhance privacy protection.

In the experiment, the FL + DP + Blockchain framework is used. All nodes validate the format and range of the noisy model parameters via smart contracts before uploading. The proportion of invalid submissions that are successfully rejected is recorded under different attack scenarios. The results show that under tampering attack scenarios, the system has a rejection rate of 88.0%, and under replay attack scenarios, the rejection rate is 89.0%. The proof-of-existence aspect is validated through log records and qualitative analysis. It shows that the immutable hash records on the blockchain provide credible proof of each node's contribution, thereby reducing the risk that malicious nodes gain undue

profit through data tampering or replay attacks. Table 5 shows the denial rate data for the two main attack scenarios:

Table 5. Contribution of Blockchain Integrity to Privacy Protection.

Attack Scenario	Detection/Rejection Rate (%)
Tampering Attack	88
Replay Attack	89

4. Discussion

In this study, the proposed blockchain + federated learning fusion model demonstrated significant advantages in supply chain credit risk assessment. By integrating blockchain technology, the model effectively ensured data privacy and promoted collaborative training through the federated learning framework, reducing the risk of data leakage. Although the model showed advantages in privacy protection, there was only a limited improvement in accuracy, especially after introducing blockchain, which caused some performance decline. This phenomenon suggests a certain privacy-utility trade-off between privacy protection and model performance. As privacy protection is strengthened, the accuracy of the model slightly decreases, but in practical applications, the value brought by privacy protection is likely more important, especially when dealing with sensitive data and multi-party collaboration where privacy protection is indispensable.

Compared to traditional centralized models and federated learning models, the blockchain + federated learning model not only improved privacy protection but also enhanced data collaboration transparency and security. In particular, when handling sensitive data, blockchain ensures data immutability and further enhances model trustworthiness through the mechanism of smart contracts. While privacy protection is ensured, the model's accuracy remains within a reasonable range. However, the introduction of blockchain in this study also brought about challenges such as communication overhead and performance trade-offs, especially in high data heterogeneity environments, where the model experienced slower convergence and accuracy degradation.

Future research could further explore optimizing the integration of blockchain and federated learning, for example, by improving communication protocols or optimizing blockchain performance to minimize its impact on model performance. Additionally, introducing heterogeneous models or multi-task learning techniques may further enhance the model's adaptability and generalization ability.

5. Conclusion

The blockchain + federated learning fusion model proposed in this study effectively improved the accuracy of supply chain credit risk assessment while ensuring data privacy protection. By combining differential privacy and blockchain technologies, the model optimized data protection mechanisms and reduced the risk of data leakage. Although the accuracy of the model slightly decreased due to privacy protection, its advantages in privacy protection and data collaboration remained evident.

This study provides a new solution for future supply chain financial systems, particularly in the context of privacy protection, data transparency, and multi-party collaboration. It demonstrates that the combination of blockchain and federated learning can effectively advance the development of supply chain finance. This research not only enriches the academic understanding of privacy protection technologies but also provides valuable insights for practical applications.

References

1. R. Qiao and L. Zhao, "Highlight risk management in supply chain finance: Effects of supply chain risk management capabilities on financing performance of small-medium enterprises," *Supply Chain Manag.*, vol. 28, no. 5, pp. 843-858, 2023, doi: 10.1108/SCM-06-2022-0219.

2. G. Zhang, Z. Yang, and W. Liu, "Blockchain-based decentralized supply chain system with secure information sharing," *Comput. Ind. Eng.*, vol. 182, p. 109392, 2023, doi: 10.1016/j.cie.2023.109392.
3. J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18209-18237, 2018, doi: 10.1109/ACCESS.2018.2820162.
4. S. Shen, T. Zhu, D. Wu, *et al.*, "From distributed machine learning to federated learning: In the view of data privacy and security," *Concurrency Comput. Pract. Exp.*, vol. 34, no. 16, p. e6002, 2022, doi: 10.1002/cpe.6002.
5. G. Zheng, D. Ivanov, and A. Brintrup, "An adaptive federated learning system for information sharing in supply chains," *Int. J. Prod. Res.*, pp. 1–23, 2025, doi: 10.1080/00207543.2024.2432469.
6. S. Dong, K. Abbas, M. Li, and J. Kamruzzaman, "Blockchain technology and application: An overview," *PeerJ Comput. Sci.*, vol. 9, p. e1705, 2023, doi: 10.7717/peerj-cs.1705.
7. S. Guo, R. Niu, and Y. Zhao, "Credit evaluation and rating system for farmers' loans in the context of agricultural supply chain financing based on AHP-ELECTRE III," *Agric. Econ. (Zeměděl. Ekon.)*, vol. 70, no. 11, pp. 541–555, 2024, doi: 10.17221/434/2023-AGRICECON.
8. J. Galindo and P. Tamayo, "Credit risk assessment using statistical and machine learning: Basic methodology and risk modeling applications," *Comput. Econ.*, vol. 15, pp. 107–143, 2000, doi: 10.1023/A:1008699112516.
9. L. Theodorakopoulos, A. Theodoropoulou, and C. Halkiopoulos, "Enhancing decentralized decision-making with big data and blockchain technology: A comprehensive review," *Appl. Sci.*, vol. 14, no. 16, p. 7007, 2024, doi: 10.3390/app14167007.
10. Z. Li, V. Sharma, and S. P. Mohanty, "Preserving data privacy via federated learning: Challenges and solutions," *IEEE Consum. Electron. Mag.*, vol. 9, no. 3, pp. 8–16, May 1, 2020, doi: 10.1109/MCE.2019.2959108.
11. M. Abdul Salam, K. M. Fouad, D. L. Elbably, et al., "Federated learning model for credit card fraud detection with data balancing techniques," *Neural Comput. Appl.*, vol. 36, pp. 6231–6256, 2024, doi: 10.1007/s00521-023-09410-2.
12. L. Kong, G. Zheng, and A. Brintrup, "A federated machine learning approach for order-level risk prediction in supply chain financing," *Int. J. Prod. Econ.*, vol. 268, p. 109095, 2024, doi: 10.1016/j.ijpe.2023.109095.
13. J. S. Bellagarda and A. M. Abu-Mahfouz, "An updated survey on the convergence of distributed ledger technology and artificial intelligence: Current state, major challenges and future direction," *IEEE Access*, vol. 10, pp. 50774-50793, 2022, doi: 10.1109/ACCESS.2022.3173297.
14. A. Rahdari et al., "A Survey on Privacy and Security in Distributed Cloud Computing: Exploring Federated Learning and Beyond," in *IEEE Open J. Commun. Soc.*, doi: 10.1109/OJCOMS.2025.3560034.
15. C. Wang, X. Chen, X. Xu, et al., "Financing and operating strategies for blockchain technology-driven accounts receivable chains," *Eur. J. Oper. Res.*, vol. 304, no. 3, pp. 1279-1295, 2023, doi: 10.1016/j.ejor.2022.05.013.
16. H. Amini, M. Bichuch, and Z. Feinstein, "Decentralized payment clearing using blockchain and optimal bidding," *Eur. J. Oper. Res.*, vol. 309, no. 1, pp. 409-420, 2023, doi: 10.1016/j.ejor.2022.12.024.
17. D. Li, D. Han, N. Crespi, et al., "A blockchain-based secure storage and access control scheme for supply chain finance," *J. Supercomput.*, vol. 79, pp. 109–138, 2023, doi: 10.1007/s11227-022-04655-5.
18. Y. Xia, C. Liu, B. Da, et al., "A novel heterogeneous ensemble credit scoring model based on bstacking approach," *Expert Syst. Appl.*, vol. 93, pp. 182-199, 2018, doi: 10.1016/j.eswa.2017.10.022.
19. P. Pławiak, M. Abdar, and U. R. Acharya, "Application of new deep genetic cascade ensemble of SVM classifiers to predict the Australian credit scoring," *Appl. Soft Comput.*, vol. 84, p. 105740, 2019, doi: 10.1016/j.asoc.2019.105740.
20. E. Dumitrescu, S. Hué, C. Hurlin, et al., "Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects," *Eur. J. Oper. Res.*, vol. 297, no. 3, pp. 1178-1192, 2022, doi: 10.1016/j.ejor.2021.06.053.
21. A. Imteaj and M. H. Amini, "Leveraging asynchronous federated learning to predict customers' financial distress," *Intell. Syst. Appl.*, vol. 14, p. 200064, 2022, doi: 10.1016/j.iswa.2022.200064.
22. K. Cheng et al., "SecureBoost: A lossless federated learning framework," *IEEE Intell. Syst.*, vol. 36, no. 6, pp. 87–98, Nov.–Dec. 2021, doi: 10.1109/MIS.2021.3082561.