In: Proceedings of the 2024 IEEE Smart World Congress (SWC), 2 - 7 December 2024, Nadi, Fiji, pp. 874-880
DOI: 10.1109/swc62898.2024.00146

Blockchain and Distributed Digital Watermarking effort on Federated Learning: Innovating Intellectual Property Protection

1st Kailin Chao
School of Software
Jiangxi Normal University
Nanchang 330027, China

2nd JunJie Li
School of Software
Jiangxi Normal University
Nanchang 330027, China

3rd Yirui Jiang
School of Water, Energy and Environment
Cranfield University
Bedfordshire MK43 0AL, UK

4th Jianmao Xiao
School of Software
Jiangxi Normal University
Jiangxi Blockchain Data Security and Governance Engineering Research Center
Nanchang 330027, China

5th Yuanlong Cao
School of Computer Information Engineering
Jiangxi Normal University
textitJiangxi Blockchain Data Security and
Governance Engineering Research Center
Nanchang 330022, China
ylcao@jxnu.edu.cn

***Abstract*—Federated Learning with Digital Watermarks (FLDW) have been recognized as a promising solution for property protection. However, the existing FLDW-related technologies neglect the requirements of decentralized settings, leading to recurrent issues such as discrepancies in distributed client data. This paper introduces a Blockchain Federated Learning Intellectual Property Protection Framework (BFLIPR), to address the data security and model validation challenges in decentralized federated learning environments. BFLIPR merges blockchain, digital watermarking, and federated learning technologies. By harnessing the blockchain's tamper-proof properties, digital watermarking's concealment capabilities, and federated learning's distributed feature, the framework offers a solution that aligns with intellectual property protection mechanism, to bolster data security and property safeguarding. Experimental findings demonstrate its high feasibility and robust for data privacy and model security in the federated learning.**

***Index Terms*—Blockchain technology, federated learning, intellectual property protection, digital watermarking, smart contract, watermark consensus mechanism.**

I. Introduction

In the age of large models, federated learning emerges as a highly anticipated, advanced technology founded on the distributed machine learning paradigm. It facilitates model training across multiple local datasets without centralizing the original data[1, 2]. After each training round, local model gradient parameters are aggregated on a central server. This approach leverages multi-party data while preserving privacy and minimizing data transmission, providing an effective solution to the data silo problem[3]. However, the current federated learning still faces challenges, such as single points failure with traditional centralized servers, data manipulation, and inadequate incentives [4, 5]. These hurdles underscore the urgency of exploring safer and more transparent federated learning technologies. Particularly, research into trusted federated learning, which integrates the decentralized and tamper-resistant traits of blockchain technology, is recognized as one of the most promising research future solution [6–8].

Blockchain technology, known for its decentralization, non-tamperability, and transparency, was initially designed for cryptocurrency systems but has since been widely applied across various fields [9]. It constructs a chain by linking transaction data into blocks arranged in chronological order, with each block containing the hash value of the preceding one, thereby ensuring data integrity[9]. Furthermore, blockchain disperses data across multiple network nodes, granting each node the authority to verify and record transactions. This distribution enhances the system's robustness against attacks and improves fault tolerance.The integration of blockchain with federated learning can address challenges such as participant traceability, single-point vulnerabilities, and incentivizing participant engagement (see Fig.1). A federated learning framework integrated with blockchain network was proposed [10]. Decentralized ledger and immutable block structure provides traceable model; while federated learning trains the distributed model and node aggregation. It promotes a more secure collaborative environment among clients [11, 12]. Nevertheless, despite the strides made in enhancing system security and data privacy, federated learning frameworks integrated with blockchain technology still face significant challenges in confirming model ownership, particularly in preventing illicit copying, misuse, and theft of models.

Some research employs digital watermarking technology as an additional approach to verify model authority [13–15]. Digital watermarking, utilized for embedding information, accomplishes authentication and tracking of data by invisibly embedding identifiers [16]. In scenarios involving multiple users, digital watermarking serves to confirm data source, maintain content authenticity, and track unauthorized copying or alterations, thereby safeguarding data ownership. Consequently, the integration of digital watermarking with decentralized federated learning holds promise in addressing the challenge of model validation within federated learning clients while effectively thwarting the risk of model theft. A federated deep neural network combined digital watermarking are proposed for ownership verification and protection [17]. It allows individual watermark embedding of each participant and multi-watermark integration accompanying the entire process of global model aggregation, to prevent illegal copying and misappropriation.

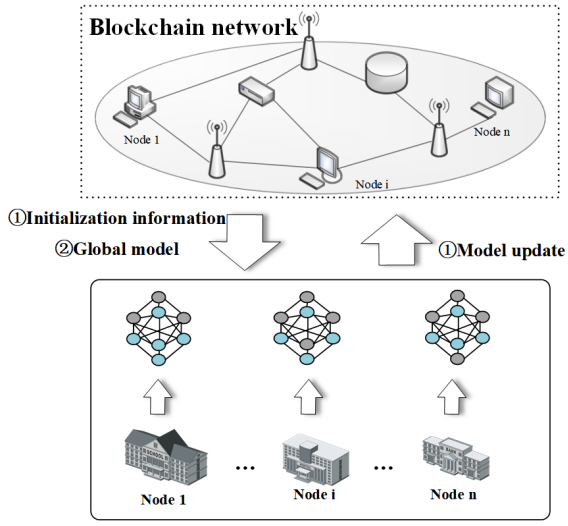


Fig. 1: Blockchain-based federated learning framework.

Nevertheless, there remains an inadequate amount of current research that thoroughly explores the utilization of blockchain, federated learning, and digital watermarking in safeguarding model authority within distributed scenarios. The challenges include: i) The single watermarking utilized in traditional machine learning necessitates innovation to facilitate multi-watermark embedding within blockchain-based federated learning frameworks, which can resolve conflicts arising from multiple participants. ii) Federated learning with digital watermarking lacks storage protection for in distributed system.

Addressing the aforementioned challenges, this paper introduces a decentralized blockchain-based federated learning framework. Built upon blockchain, it integrates digital watermarking and federated learning, coupled with a personalized consensus mechanism. The framework can verifying model ownership, traceability of transaction behavior, and multiple-participation equitability.

This paper makes the following contributions:

i. Introduce distributed digital watermarking technology, integrated blockchain consensus mechanisms and smart contracts.

ii. Propose a blockchain-based federated learning framework, with blockchain's tamper-proof feature, federated learning's privacy feature, and the copyright mechanism of digital watermarking. It constructs an integrated framework for model copyrights verification , and efficiently managment.

The remainder of the paper is structured as follows: Section II introduces the architecture of the proposed framework. Section III meticulously scrutinizes the experimental findings . Finally, the conclusions are presented in Section IV.

II. METHODOLOGY

Integrating blockchain and smart contract based on the Federated Learning Model Intellectual Property Rights (FedIPR) framework[17], this paper proposes Blockchain Federated Learning Model Intellectual Property (BFLIPR) framework for privacy information security standards. The BFLIPR framework establishes a secure and dependable decentralized distributed scenario for implementing digital watermarking in federated learning. This section presents the system architecture and design of BFLIPR, blockchain-based digital watermarking generation and embedding, and blockchain-based reliable digital watermarking verification mechanism.

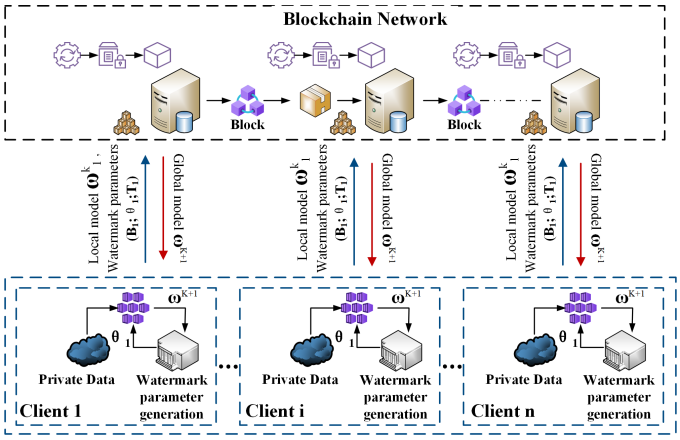


Fig. 2: BFLIPR framework architecture.

A. System architecture and design

In the BFLIPR system framework (see Figure 2), each client is mandated to upload their private watermark parameters and model gradient parameters to the chain via smart contracts upon submitting model parameters $(B_i; \theta_i; T_i)$.

The private watermark parameters of participants are standardized into matrices and stored on the blockchain by the smart contract. Subsequently, the processed global watermark matrix is disseminated through the consensus mechanism E^t. Among them, the private watermark parameters $(B_i; \theta_i; T_i)$ remain available and invisible to other participants.

Building upon FedIPR, the BFLIPR framework employs smart contracts to integrate and optimize various components of the system framework, while also facilitating a pluggable modular design. This section certainly introduces blockchain-enhanced federated learning, the innovative digital watermarking module, and the smart contract-driven blockchain network module.

1) Blockchain-enhanced federated learning and innovative digital watermarking module

In addition to the original FedIPR framework's combination of the black-box and white-box models for model embedding and verification, BFLIPR has optimized the framework by employing lightweight nested verification algorithms within smart contracts to integrate with the black-box model. The white-box model remains an optional component of the system framework. The original design intent of the white-box model is still innovatively achieved on the blockchain, reducing the complexity of the integrated framework. BFLIPR uses below two watermarking methods based on FedIPR[17]:

- **Feature watermarking**: The watermarking is embedded directly into the parameters of the federated learning. By altering the model's weights, it incorporates specific patterns that are challenging to detect. Specifically, for a given set of model parameters W, feature watermarking can be achieved by adding an optimization term $L_{\text{watermark}}(W)$ to the original loss function $L_{\text{original}}(W)$. The purpose of the optimization item is to adjust the model parameters to include specific watermark information without significantly affecting the original task performance. Its mathematical expression is:

$$L_{\text{total}}\mathbf{W} = L_{\text{original}}\mathbf{W} + \lambda L_{\text{watermark}}\mathbf{W}, \tag{1}$$

  where λ is the adjustment term used to balance watermark embedding and model performance.
- **Backdoor-based Watermarks**: The core idea of this type of watermark is to create a specific set of input-output pairs (i.e., a trigger set). When the model encounters a specific input pattern (trigger), it produces a predefined output that proves ownership of the model. The mathematical form of this watermark is: given a set of trigger samples $\mathcal{T} = \{(x_i, y_i)\}$, the model $f(x_i)$ should output when encountered y_i.

2) Smart contract-driven blockchain network module

BFLIPR innovative integrates blockchain network module with others through smart contracts and consensus mechanisms. This integration allows all participants to maintain a continuously expanding list of records (blocks) in the decentralized network, documenting crucial model transactions and data changes during training, thereby playing a pivotal role in the entire framework. To ensure data consistency and system robustness, this paper adopts the Practical Byzantine Fault Tolerance (PBFT) consensus mechanism as the experimental algorithm. The PBFT consensus mechanism used in the framework is as follows.

The PBFT system mainly contains Master Node (Leader) and Slave Node (Followers) (see Code 1). The master node is responsible for initiating new requests or proposals, while the slave nodes are responsible for validating and voting on proposals. The workflow of PBFT as below:

Algorithm 1: PBFT Consensus Mechanism Process

Input: Client request request, Total nodes N, Faulty nodes f
Output: Consensus confirmation C

if *Node is primary* **then**
  sequenceNumber ← assignUniqueSequenceNumber();
  m_{pre} ← createPrePrepareMessage(request, sequenceNumber);
  broadcast m_{pre} to all nodes;
end
prePrepares ← **new list** to collect pre-prepare messages;
prepares ← **new list** to collect prepare messages;
commits ← **new list** to collect commit messages;
while *receiving pre-prepare* m_{pre} *from primary* **do**
  if *validate(*m_{pre}*)* **then**
    broadcast prepare message based on m_{pre};
  end
end
while *receiving prepare message* m_{prep} **do**
  if *validate(*m_{prep}*)* **then**
    add m_{prep} to prepares;
    if *count(prepares) >2f* **then**
      broadcast commit message based on m_{prep};
    end
  end
end
while *receiving commit message* m_{commit} **do**
  if *validate(*m_{commit}*)* **then**
    add m_{commit} to commits;
    if *count(commits) >2f* **then**
      execute request and return result;
    end
  end
end

- **Pre-prepare:** The master node receives the request from the client, assigns it a unique sequence number, and then broadcasts the request and sequence number to all slave nodes in the form of a message. This message is digitally signed with the private key of the master node, ensuring its non-repudiation and verification of origin. The message can be represented as $M_{\text{pre}} = \text{sign}\left(H\left(\text{request}, \text{sequence number}\right), \text{SK}_{\text{leader}}\right)$ where H is the hash function used to generate a unique representation of the request and sequence number, and $\text{SK}_{\text{leader}}$ is the master node's private key. In the BFLIPR framework, a watermark matrix is sent along with the verification message E^k.
- **Preparation phase (Prepare):**The slave node receives

and uses the public key of the master node to verify (PK_{leader}) whether the digital signature of the pre-prepared message sent by the master node is correct.The verification process ensures the authenticity of the message and the correctness of the sequence number, preventing replay attacks and message tampering. After successful authentication, the slave node broadcasts a prepare message indicating that it is ready to process the request.

- **Commit phase (Commit):**After receiving a sufficient number of prepare messages (usually required to exceed the total number of network nodes), $\frac{2}{3}$ the slave node broadcasts a commit message. This step marks the node's final approval of the request. The key to the commit phase is to ensure most nodes in the network agree on the current request status, thereby reaching a consensus. After receiving the submission message from the majority of other nodes, the node will execute the request and return the result to the client. The broadcast of the submission message can be expressed as:

$$\text{Commit} = \text{true if } |\{\text{valid prepare messages}\}| > 2f, \tag{2}$$

where f is the maximum possible number of malicious nodes in the network.

B. Blockchain-based digital watermarking generation and embedding

BFLIPR introduces innovations in the generation and embedding of digital watermarking. As shown in Figure 3, The digital watermarking generation and embedding details as below:

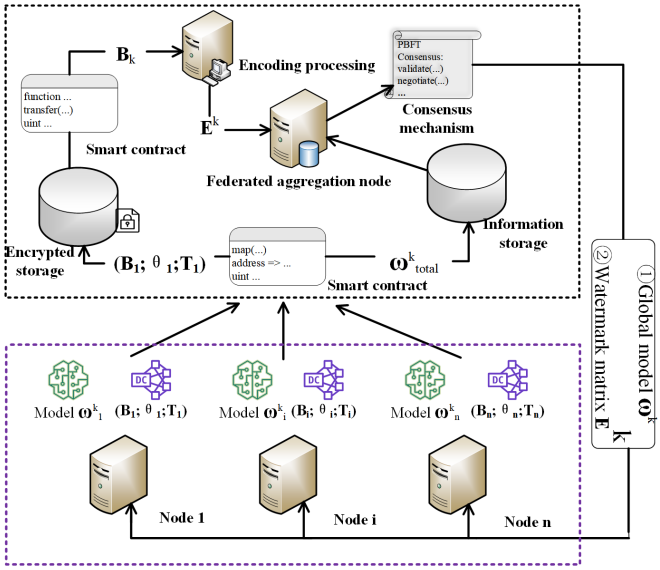


Fig. 3: Generation and embedding process of BFLIPR .

1) Integration and optimization of global feature watermark coding and training process

- **Generation stage:**The global feature watermark of each round is generated by all client feature watermarks B_k through complex encoding technology. The encoding process includes converting the watermark into a series of binary strings $B_k^{\text{matrix}} = \text{Encode}(B_k)$. These strings are designed to map directly into model parameters and are difficult to detect by third parties. Before completing the encoding, the smart contract encrypts these client watermark parameters and stores them on the blockchain through the PBFT consensus mechanism to ensure the consistency and security of the information; after the encoding is completed, the PBFT consensus mechanism will store the global characteristics watermark and model parameters together, sending preparation information to all clients, completing global broadcast, and storing this important transaction on the chain.
- **Embedding stage:**The feature watermark is integrated into model training through a customized loss function $L_{\text{watermark}}W$. The loss function:

$$L_{watermark}\boldsymbol{W} = \sum_{i=1}^{N} f(B_{k,i}, W_i), \tag{3}$$

is a specially designed regularization to optimize model parameters including watermark information. Parameter updates employ advanced optimization algorithms such as stochastic gradient descent with momentum $W_{\text{new}} = W - \eta\nabla L_{\text{total}}W$ to minimize the impact on model performance.

2) Backdoor watermark generation and activation strategy of the model

- **Generation phase :** The set of triggers based on the backdoor watermark $\mathcal{T}_k$ is generated through adversarial learning techniques. Each trigger (x_i, y_i) is designed for a specific layer of the model to maximize the activation of a specific neural network path. The optimization formula used during the generation process:

$$\mathcal{T}_k = \boldsymbol{x}i + \delta : \delta = \arg\min|\delta| \leq \epsilon L(y_i, N(\boldsymbol{x}_i + \delta)). \tag{4}$$

Simultaneously, all trigger samples and their parameters undergo encryption via smart contracts post-generation and are subsequently stored on the blockchain. The PBFT consensus mechanism ensures the precision and security of this data. During this process, the trigger parameters are not globally broadcasted, thus furnishing non-tamperable and enduring protection for subsequent watermark information verification.
- **Embedding stage:** The model's training strategy is meticulously calibrated to ensure the effective embedding of triggers while minimizing impact on the model's primary performance. This entails fine-tuning parameters such as the learning rate and employing optimization algorithms like stochastic gradient descent with momentum to activate and maintain the watermark within a specific layer. The objective is to heighten the model's responsiveness to these particular samples while minimizing

interference with overall performance. Throughout this process, smart contracts autonomously manage the embedding process of trigger collections, including sample selection, adjustment of embedding time and frequency, and dissemination of watermark information. This automation reduces the necessity for human intervention, thereby enhancing the efficiency and reliability of the entire embedding process.

C. Blockchain-based digital watermarking verification mechanism

The verification process of the BFLIPR framework leverages blockchain to enhance security, transparency, and consistency. This process encompasses advanced verification mechanisms for signature watermarks and backdoor-based watermarks.

Algorithm 2: Evidencc Contract Implementation

```
Input: FederatedLearningData, DigitalWatermark
Output: Result of the request
Function SaveData(id: string, data: string):
    newData.id ← id;
    newData.data ← data;
    newData.timestamp ← Now;
    // If the data already exists,
       update the timestamp, otherwise
       add new data
    if Length(DataMap) >0 then
        if DataMap[High(DataMap)].id ← id then
            DataMap[High(DataMap)].timestamp ←
              Now;
        end
        else
            Length(DataMap, Length(DataMap) + 1);
            DataMap[High(DataMap)] ← newData;
        end
    end
    else
        SetLength(DataMap, Length(DataMap) + 1);
        DataMap[High(DataMap)] ← newData;
    end
Function GetData(id: string):
    foreach data in DataMap do
        if data.id ← id then
            Result ← data;
            return;
        end
    end
    // If not found, an empty data
       structure is returned
    FillChar(Result, SizeOf(Result), 0);
```

1) Decoding and verification of feature watermarking

First, the verifier needs to extract the watermark information from the model parameters using the same algorithm used when embedding the watermark $\widetilde{B_k}$ to ensure the accuracy of the extracted information. The extracted watermark information $\widetilde{B_k}$ is compared with the original watermark information B_k stored on the blockchain through equation (5), and the similarity between them is calculated:

$$Similarity(B_k, \tilde{B}_k) = 1 - \frac{\text{HammingDistance}(B_k, \tilde{B}_k)}{N}. \quad (5)$$

In the model design stage, equation (6) is integrated and deployed the verification mechanism in the smart contract. Therefore, the actual execution of the verification process is automated by smart contracts, effectively ensuring the accuracy and consistency of verification.

2) Decoding and verification of backdoor watermark

During the verification process, the trigger sample set $\mathcal{T}_k$ is tested, and the smart contract calls the trigger parameters stored on the blockchain to ensure the consistency and security of the test:

$$Accuracy_{\text{backdoor}} = \frac{1}{|\mathcal{T}k|} \sum (\boldsymbol{x}_i, y_i) \in \mathcal{T}_k \mathbb{I}[N(\boldsymbol{x}_i) = y_i]. \quad (6)$$

The process is pre-deployed within the smart contract. The automation feature of the smart contract ensures the efficiency and reliability of the verification process, minimizing the requirement for human intervention and eliminating the likelihood of verification errors.

III. EXPERIMENT

This section performs an empirical study on the proposed BFLIPR, assessing the feasibility and robustness of watermarking within a decentralized blockchain context. The findings demonstrate that BFLIPR can offer comparable reliability and robustness to FedIPR operating within a traditional environment. Classic AlexNet and ResNet-18 architectures are utilized to evaluate performance on the CIFAR10 dataset, known for their representativeness. CIFAR-10 comprises 60,000 images across 10 categories, as detailed in Tables I and II.

The CIFAR-10 dataset comprises images that are uniformly distributed, with each image accompanied by a corresponding label, rendering it highly suitable for supervised learning tasks. Given its relatively modest size, the CIFAR-10 dataset proves invaluable for testing and validating the efficacy of federated learning algorithms. In this paper, a simulated federated learning platform is utilized, wherein clients upload their local models in each round of communication, while the server employs the Fedavg algorithm [13] to aggregate these local models. Additionally, concerning blockchain configuration, clients also upload watermark information and federated learning data to the blockchain throughout the training process.

A. feasibility

The experimental results of the FedIPR and BFLIPR frameworks executing the AlexNet model (see Figure 5 and Figure 6) and ResNet (see Figure 7 and Figure 8) with 10 clients are presented. Train-Accuracy serves as a commonly utilized evaluation metric for assessing the performance of a classifier.

TABLE I: Dataset category

Category	Airplane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Boat	truck
Quantity	6000	6000	6000	6000	6000	6000	6000	6000	6000	6000
Specification	32*32	32*32	32*32	32*32	32*32	32*32	32*32	32*32	32*32	32*32

TABLE II: Data set distribution

Category	Training Set	Test Set	Total
Quantity	50000	10000	60000

It quantifies the ratio of correctly classified samples within a category to the total number of samples. The formula is as follows:

$$Accuracy = \frac{\alpha + \beta}{\alpha + \gamma + \delta + \beta}, \tag{7}$$

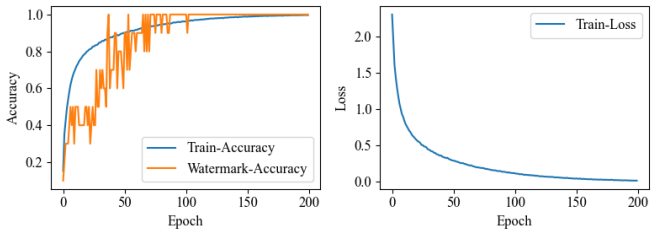


Fig. 4: FedlPR-AlexNet model training results.

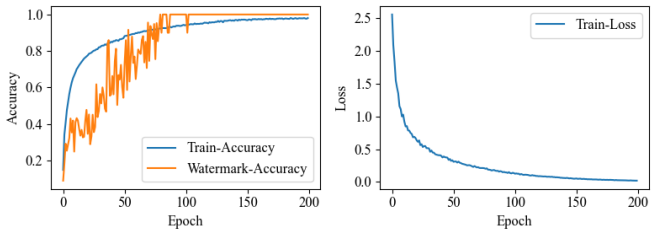


Fig. 5: BFLIPR-AlexNet model training results.

In the task of determining whether an image depicts a dog. If the detection correctly identifies a dog, it means the image is indeed of a dog. Conversely, if the detection fails to identify a dog, it suggests the image does not depict a dog. However, if the detection incorrectly identifies a non-dog image as a dog, or fails to identify a dog in an image, it indicates a discrepancy in the detection result. Watermark-Accuracy serves as a crucial metric for assessing the accuracy of watermark embedding and extraction. It quantifies the extent to which the watermark signal has been altered or compromised from its original state. Meanwhile, Train-Loss reflects the loss function, which gauges the effectiveness of the model. This article employs the cross-entropy function (Cross Entropy) for this purpose, as depicted in the following formula:

$$H(p, q) = -\sum_{i=1}^{n} p(x_i) \log q(x_i), \tag{8}$$

When training the network, the input data and labels are determined, and the true probability distribution P_{true} is defined. The value represented by this function represents the difference between the true probability distribution P_{true} and the predicted probability distribution $P_{\text{predicted}}$. The smaller the value, the better the prediction result.

As shown in the above two sets of pictures, BFLIPR achieved similar training results in the three indicators of Train-Accuracy, Watermark-Accuracy, and Train-Loss compared with FedIPR during the training process of AlexNet model and ResNet model. BFLIPR attains high fidelity due to its unique design and characteristics, making it particularly suitable for operation within a blockchain environment. It meticulously incorporates the decentralization, security, transparency, and tamper-proof attributes of blockchain, optimizing these features. Within blockchain, maintaining data integrity

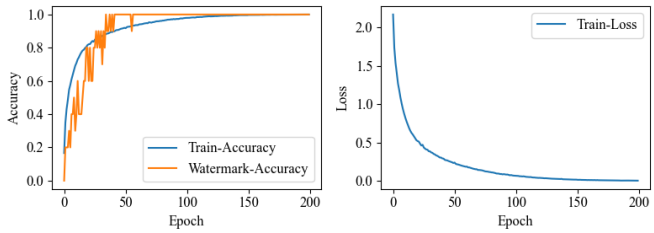


Fig. 6: FedlPR-ResNet model training results.

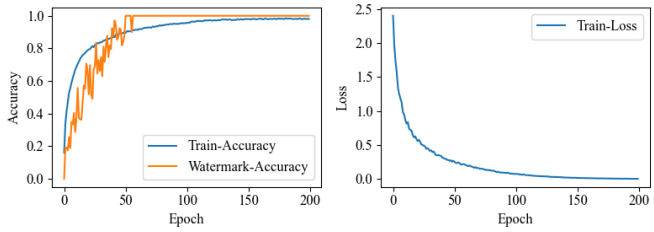


Fig. 7: BFLIPR-ResNet model training results.

and trustworthiness is paramount. BFLIPR achieves intellectual property protection for models by embedding imperceptible watermark information. This watermarking technology seamlessly integrates information into models without impacting their performance, while also ensuring the watermark remains intact and identifiable even if the model is tampered with or copied. This affords model creators robust legal protection and prevents unauthorized copying or misuse of models.

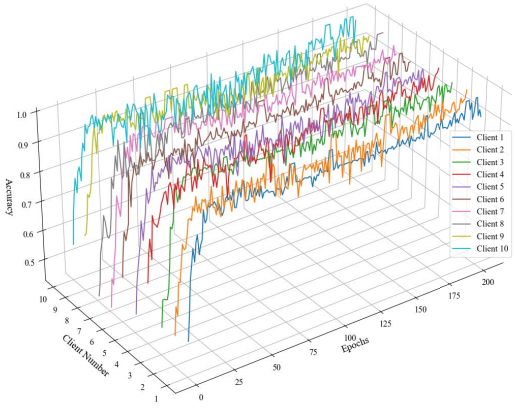
Moreover, during training, BFLIPR can adapt to the loss function, enabling the model to preserve the integrity and accuracy of the watermark information while striving for high

performance. This balanced training strategy enhances the model's performance and fortifies the robustness of the watermark information, ensuring the model can operate normally within the blockchain environment.

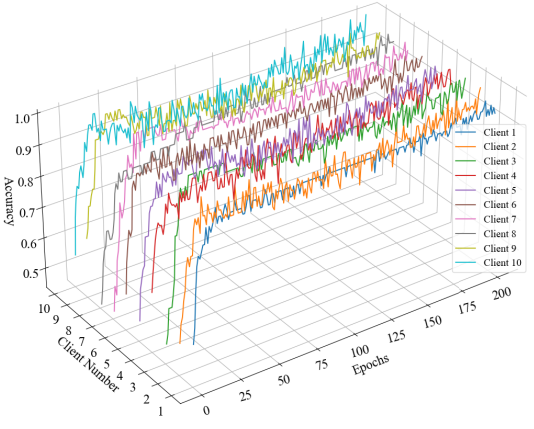
In federated learning, the distribution of data among clients is typically disparate and independently distributed. Consequently, the efficiency of embedding and extracting each client's watermark becomes a crucial indicator of model robustness when data distribution is inconsistent. This paper simulates 10 clients on the established federated learning platform. The FedlPR and BFLIPR frameworks are employed to run the AlexNet and ResNet models, evaluating watermark accuracy on the CIFAR10 dataset.

B. Robustness

When employing both the FedlPR and BFLIPR frameworks with the AlexNet model, the watermark accuracy of each client initially displays fluctuations during the early stages of training (see Figure 9). These fluctuations may arise from various factors, including model initialization randomness, data transmission instability, and random noise. However, as training progresses, these fluctuations gradually diminish, and

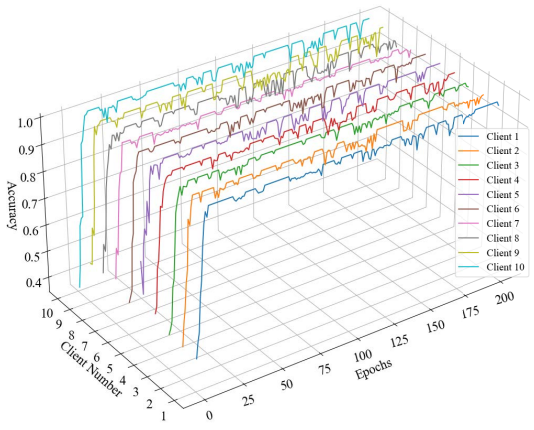


(a) BFLIPR

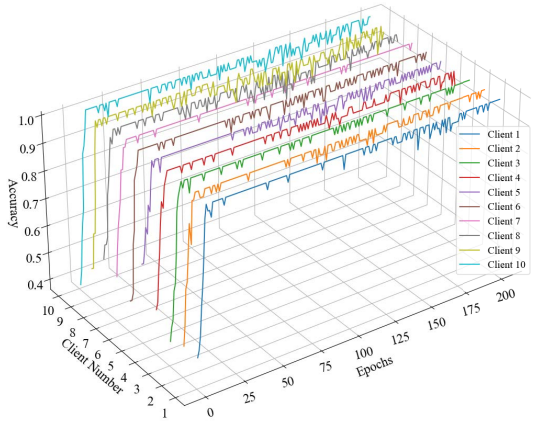


(b) FedIPR

Fig. 8: AlexNet client watermark accuracy.



(a) BFLIPR



(b) FedIPR

Fig. 9: Resnet Client watermark accuracy.

watermark accuracy begins to converge. This convergence trend indicates that both the FedlPR and BFLIPR frameworks effectively learn and optimize under the AlexNet model. Within the initial 25 epochs, the convergence trend of the BFLIPR framework appears smoother compared to the FedlPR framework. In the early stages of model training, the difference in watermark accuracy among clients in the BFLIPR framework is smaller, indicating higher stability and consistency. The optimization of the BFLIPR framework for blockchain enables it to adapt and converge more effectively during the initial phases of model training.

When both the FedlPR and BFLIPR frameworks are executed using the ResNet model, the watermark accuracy of each client attains a faster convergence speed compared to the AlexNet model (see Figure 10). This enhancement is attributed to the deeper network structure and more potent feature extraction capabilities inherent in the ResNet model itself. By introducing residual connections, the ResNet model effectively mitigates common issues like vanishing gradients and representation bottlenecks in deep neural networks. Consequently, the model can more efficiently learn and extract

crucial information from the training data. Hence, with the ResNet model, both the FedlPR and BFLIPR frameworks can swiftly adjust model parameters, resulting in accelerated convergence. After the client watermark accuracy reaches 100% for the first time under the ResNet model, the fluctuation range of the BFLIPR framework in subsequent training sessions is smaller than that of the FedlPR framework (see Figure 10). This observation underscores the BFLIPR framework's superior convergence and stability under the ResNet model. This advantage may be attributed to the BFLIPR framework's specialized optimization for blockchain, allowing it to exhibit adaptability and robustness within deep neural network models.

IV. CONCLUSION

This paper introduces the Blockchain Federated Learning Model Intellectual Property (BFLIPR) framework, a secure and decentralized solution addressing prevalent challenges in data security and model validation within federated learning models. Through the innovative integration of blockchain, digital watermarking, and federated learning technologies, BFLIPR enhances the capabilities of safeguarding data security and intellectual property. By incorporating non-tamperable blockchain and a watermark consensus mechanism supporting smart contracts, BFLIPR elevates model transparency and verification capabilities. Theoretical advancements and empirical study findings substantiate the feasibility and robustness of the BFLIPR framework in decentralized environments. Future work will focus on further enhancing the BFLIPR framework, expanding its applicability to diverse scenarios, optimizing efficiency, and investigating its adaptability in different client environments.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 61962026, the Natural Science Foundation of Jiangxi Province under Grant No. 20224ACB202007, Jiangxi Provincial Natural Science Foundation under Grant No. 20224BAB212015, Jiangxi Provincial 03 Special Project, and 5G Project (20224ABC03A13, 20232ABC03A26).

REFERENCES

[1] Li, L., Fan, Y., Tse, M., & Lin, K. Y. (2020). A review of applications in federated learning. *Computers & Industrial Engineering*, 149, 106854.

[2] Konečný, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint* arXiv:1610.02527.

[3] Ye, M., Fang, X., Du, B., Yuen, P. C., & Tao, D. (2023). Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3), 1-44.

[4] Beltrán, E. T. M., Pérez, M. Q., Sánchez, P. M. S., Bernal, S. L., Bovet, G., Pérez, M. G., ... Celdrán, A. H. (2023). Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*.

[5] Fu, L., Zhang, H., Gao, G., Zhang, M., & Liu, X. (2023). Client selection in federated learning: Principles, challenges, and opportunities. *IEEE Internet of Things Journal*.

[6] Nguyen, D. C., Ding, M., Pham, Q. V., Pathirana, P. N., Le, L. B., Seneviratne, A., ... Poor, H. V. (2021). Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet of Things Journal*, 8(16), 12806-12825.

[7] Qu, Y., Uddin, M. P., Gan, C., Xiang, Y., Gao, L., & Yearwood, J. (2022). Blockchain-enabled federated learning: A survey. *ACM Computing Surveys*, 55(4), 1-35.

[8] Ma, C., Li, J., Shi, L., Ding, M., Wang, T., Han, Z., & Poor, H. V. (2022). When federated learning meets blockchain: A new distributed learning paradigm. *IEEE Computational Intelligence Magazine*, 17(3), 26-33.

[9] Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4), 352-375.

[10] Li, Y., Chen, C., Liu, N., Huang, H., Zheng, Z., & Yan, Q. (2020). A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network*, 35(1), 234-241.

[11] Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology?—a systematic review. *PloS one*, 11(10), e0163477.

[12] Xu, M., Chen, X., & Kou, G. (2019). A systematic review of blockchain. *Financial Innovation*, 5(1), 1-14.

[13] Tekgul, B. G., Xia, Y., Marchal, S., & Asokan, N. (2021, September). Waffle: Watermarking in federated learning. In 2021 40th International Symposium on Reliable Distributed Systems (SRDS) (pp. 310-320). IEEE.

[14] Yuan, X., Ma, X., Zhang, L., Fang, Y., & Wu, D. (2021). Beyond class-level privacy leakage: Breaking record-level privacy in federated learning. *IEEE Internet of Things Journal*, 9(4), 2555-2565.

[15] Hwang, D., Mun, H., & Lee, Y. (2022, April). Improving response time of home IoT services in federated learning. In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (pp. 157-163).

[16] Van Schyndel, R. G., Tirkel, A. Z., & Osborne, C. F. (1994, November). A digital watermark. In Proceedings of 1st international conference on image processing (Vol. 2, pp. 86-90). IEEE.

[17] Li, B., Fan, L., Gu, H., Li, J., & Yang, Q. (2022). FedIPR: Ownership verification for federated deep neural network models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4), 4521-4536.