2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
17-20 October, 2021. Melbourne, Australia

2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-4207-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMC52423.2021.9658998

Secure Federated Learning Model Verification: A Client-side Backdoor Triggered Watermarking Scheme

Xiyao Liu[1], Shuo Shao[1], Yue Yang[1], Kangming Wu[1], Wenyuan Yang[2*], Hui Fang[3]

***Abstract*— Federated learning (FL) has become an emerging distributed framework to build deep learning models with collaborative efforts from multiple participants. Consequently, copyright protection of FL deep model is urgently required because too many participants have access to the joint-trained model. Recently, Secure FL framework is developed to address data leakage issue when central node is not fully trustable. This encryption process has made existing DL model watermarking schemes impossible to embed watermark at the central node. In this paper, we propose a novel client-side Federated Learning watermarking method to tackle the model verification issue under the Secure FL framework. In specific, we design a backdoor-based watermarking scheme to allow model owners to embed their pre-designed noise patterns into the FL deep model. Thus, our method provides reliable copyright protection while ensuring the data privacy because the central node has no access to the encrypted gradient information. The experimental results have demonstrated the efficiency of our method in terms of both FL model performance and watermarking robustness.**

I. INTRODUCTION

Federated learning (FL) [1], as one of the most popular distributed deep learning paradigms, enables multiple participants to collaboratively learn a deep learning (DL) model without sharing training data across the engaged parties. In many privacy-sensitive applications, e.g., medical image analysis systems [2], [3], [4] and personal device recommendation systems [5], [6], large amount of private data from multiple parties is required to build accurate and reliable DL models [7], [8]. In these cases, a centralized data-sharing scheme is not practical as the risks of compromising data privacy and confidentiality become significantly high. In contrast to centralized learning paradigm, FL aggregates locally-computed gradients from data owned by individual parties to joint-learn a DL model so that centralized data sharing is avoided. Consequently, it provides a decentralized method to build the model as well as enhance the data privacy.

Many methods have been proposed to provide copyright protection of centralized DL models [9], [10], [11], [12], [13], [14], [15], [16], [17]. These methods can be categorized into two main classes [18]: watermark embedding into parameters of a DL model [9], [10] and backdoor embedding methods [11], [12], [13], [14], [15]. The first category embeds digital watermark into weights of a DL model to verify ownership of the model [9]. However, the watermark can only be extracted when the weights of the model are accessible. The second category utilizes a backdoor subtask at the training stage [11] so that the watermarked model has an exclusive ability to accomplish the subtask while other unprotected models cannot. This type of schemes are more suitable for the copyright protection of DL models when the weights of a verified DL model are not accessible.

FL based models is an emerging topic despite the progress in protecting centralized DL models. Compared with a centralized DL model, there is a stronger motivation for the copyright protection of FL based model because multiple parties have access to the model that is deployed into production stage. Recently, a watermarking scheme to protect FL based model, named WAFFLE, is proposed in [19]. It introduces a re-training step at the central node to embed a backdoor-based watermark [11] into the aggregated model after each iteration of model update via collecting gradients from client-side local models.

However, WAFFLE is not applicable in many new FL architectures as it is built based on an assumption that the central node is the owner of the FL model who is fully trustable. While this assumption is not always true as a sole participant or a subgroup of participants, i.e. local client nodes, could be the owner of the joint-trained model [7], [20]. Further, it has been proved that original data could be leaked from gradient information [21]. Thus, one of the best practices of FL based training strategy is to encrypt both model weights and gradients so that the central node can not access the model parameters but only aggregate and distribute the gradients [7], [21], [22]. By using such an architecture, data privacy is enhanced in the FL framework. However, WAFFLE is no longer suitable for such a framework as central node has no permission to access the model parameters.

In our paper, we propose a novel client-side FL watermarking method to provide a model verification solution with enhanced data privacy. In specific, the Additively Homomorphic Encryption (AHE) [7], [21], [23] is utilized to encrypt both model weights and gradients for data exchange in the FL framework so that central node has no permission to access both model parameters and gradient information. During the training stage, the model owner, either one of the client

[1]Xiyao Liu, Shuo Shao, Yue Yang, Kangming Wu are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China, Email address: lxyzoewx@csu.edu.cn, ss0822@csu.edu.cn, yangyue2018@csu.edu.cn, WuKangming@csu.edu.cn

[2]Wenyuan Yang is with Shenzhen Graduate School, Peking University, Shenzhen 518055, China, Email address: wyyang@pku.edu.cn

[3]Hui Fang is with the Department of Computer Science, Loughborough University, Loughborough LE11 3TU. E-mail address: h.fang@lboro.ac.uk

*To whom correspondence should be addressed.

This research is supported by the National Natural Science Foundation of China (61602527,61772555,61772553,U1734208), Natural Science Foundation of Hunan Province, China (2020JJ4746)

nodes or a subgroup of the client nodes, uses the backdoor algorithm to embed watermark into the joint-learned model. In this manner, both data privacy and copyright protection of the model are enhanced with the elegant design of the FL architecture which is illustrated in Fig. 1.

The contributions of our work are highlighted as below:

- A novel watermarking method is proposed to provide model copyright protection in a Secure FL framework. To our best knowledge, this is the first scheme to embed watermark into a DL model learned under a Secure FL environment where there are no other existing watermarking methods can achieve.
- A client-side watermarking embedding scheme is designed to provide reliable copyright protection while ensuring the data privacy because the central node has no access to the encrypted gradient information.
- A noise-based trigger set is generated to embed backdoor watermark into DL model. In this manner, the copyright of FL based model can be verified without knowing the parameters in the model.
- Various experiments are conducted to demonstrate that our method is resilient to watermarking removal techniques including fine-tuning, pruning and quantization attacks.

The rest of the paper is structured as follows. In Section II, we provide a literate review of digital watermarking techniques on deep neural networks. We then discuss the framework and present our method in Section III. Afterwards we present our experiment settings and results in section IV. Finally, we conclude our work in section V.

II. RELATED WORK

Many watermarking methods have been proposed to protect the copyright of DL models [9], [10], [11], [12], [13], [14], [15], [16], [17]. Broadly speaking, these methods can be categorized into two classes: white-box watermarks and black-box watermarks [18], [24].

White-box watermarking methods embed watermark directly into the DL model parameters. Since a typical DNN model has millions of parameters, its over-parameterization property could be used to embed watermark for copyright protection without compromising the model accuracy. Uchida et al. [9] propose one of the earliest DL model watermarking methods. During the training of a DL model, a regularizer loss term is added into the cost function to embed a watermark into the weights of the model. In [24], Wang et al. set up an adversarial learning framework to embed watermark into DNN model so that the marked network is robust against various model transformation attacks. Although white-box watermarking methods have the advantage of maintaining DNN model performance, the model parameters are required to be accessible at the watermark verification stage.

Black-box watermarking refers to those schemes which identify copyright information by sending queries to a DL model without accessing its parameters. Yossi Adi et al. [11] design a so-called commitment scheme to combine watermark and backdoor with a cryptographic primitive. In the work, a pair of public key and private key is customized for the output and input of the model trigger set, and the ownership is verified based on the error output of the trigger set. Le Merrer et al. [15] fine tune the model so that it is robust against certain adversarial examples. Similarly, Zhang et al. [12] use three types of irrelevant data samples as watermarks into the training data, which are labeled with classes from the original model output.

Recently, Atli et al. [19] propose an approach to watermark FL DNN models which is the most relevant to our proposed method. In their work, a backdoor-based watermark [11] is embedded at the model updating stage when collected gradients are aggregated at the central node. The main difference between our proposed method and this work is that we provide a client-side watermarking scheme so that the data privacy can be further enhanced by encrypting the gradients for data exchange. The details are explained in the following section.

III. WATERMARK IN SECURE FEDERATED LEARNING

A. Secure Federated Learning Framework

Federated Learning is a distributed machine learning framework aiming to protect data privacy of multiple participants who joint-learn a DL model without sharing their local data. Recently, a potential data leakage risk is identified when the gradients are aggregated at the central node [21]. Therefore, cryptographic methods, e.g., AHE [7], [21], have been used to ensure the data privacy. During the learning, the detailed steps of a Secure FL learning process at each iteration are described as follows. First, central server randomly selects several clients as a set C_i. Second, each selected client uses its own dataset to train a local model L_j^i where the subscript j represents the index of an individual client. Third, the clients utilize same private key to encrypt their gradients and send the encrypted messages to the central node. Once the central node receives all the encrypted gradients from these clients, Secure FedAvg [7] algorithm and AHE enable the server to aggregate these gradients to the encrypted global gradients $Enc(g^i)$ which can be expressed as:

$$Enc(g^i) = \{\prod_{k \in C^i} [Enc(L_j^i - G^{i-1})]^{\frac{n_j}{N^i}}\}^{\eta} \tag{1}$$

After decryption, the global gradients g^i is equivalent to:

$$g^i = \eta \sum_{k \in C^i} \frac{n_j}{N^i} (L_j^i - G^{i-1}) \tag{2}$$

Here, n_j refers to the j-th client's quantity of its own dataset and N^i refers to the number of the dataset samples of all the clients selected in the i-th iteration. η is the learning rate of server and G^{i-1} is the global model in $i-1$ iteration. $Enc(*)$ represents the AHE encryption method (e.g. Paillier's method [25]). Finally, the central server returns the encrypted global gradients to each client which decrypts the gradients and updates its local model by using Eq. (3).

$$G^i = G^{i-1} - g^i \tag{3}$$

In such a learning framework, the central server only aggregates and distributes the encrypted gradients, thus it cannot derive any private information from the local gradients.

B. Watermarking Scenario

The scenario of the Secure Federated Learning to be verified is discussed in this subsection: Alice is an enterprise who has a number of clients Bobs to use its services. Alice decides to improve the quality of its services by training a high-performance DL-based model. However, its own data is not sufficient to train such a model. Therefore, the initiator, Alice, plans to use Secure Federated Learning to build a reliable model with joint-efforts from Bobs. At the training stage, Alice hires Carol to perform as the central server of the FL and utilizes AHE (e.g. paillier's methods [25]) to protect the privacy of its clients' local data. Therefore, none of these parties will possibly get any private information of other parties.

However, since training a Federated Learning model needs to send the model weights to each client, a malicious client Eva may copy the model which does not belong to her. To protect the copyright of the model, Alice intends to embed watermark into the model so that she could verify unauthorized use of the model. In this scenario, Alice is one of the client-side nodes in the FL system. We assume that:

- Alice has the full control of its own training procedure, which means Alice could set the hyper-parameters of its local computation of its gradients.
- Alice can discretionarily select data to train the model in the Federated Learning process.
- Alice has no control on any actions of other clients, i.e. Bobs and Eva and central node Carol, who follow the Federated Learning training procedure.

C. Watermark in Secure Federated Learning

In our work, we use a backdoor-based watermarking method as Alice cannot change the global model weights due to the restriction of the Secure FL framework. Backdoor embedding aims to force a deep learning model to classify the data of trigger set x^* into a pre-defined class. Beyond that, the model performs well with the common classification task trained with ordinary samples x. Formally, the target of backdoor watermark is to train a model to achieve

$$\begin{cases} f(x) = f^*(x) \\ f(x^*) = l^* \neq f^*(x^*) \end{cases} \tag{4}$$

where x refers to ordinary sample, while x^* refers to data of trigger set, $f(x)$ represents the model prediction of sample x, while $f^*(x)$ is the ground-truth label of sample x.

The illustration of watermarking stage and copyright verification stages of the Secure FL are shown in Fig. 1. During the training, the initiator Alice adds a trigger set into its normal training data samples to embed the backdoor

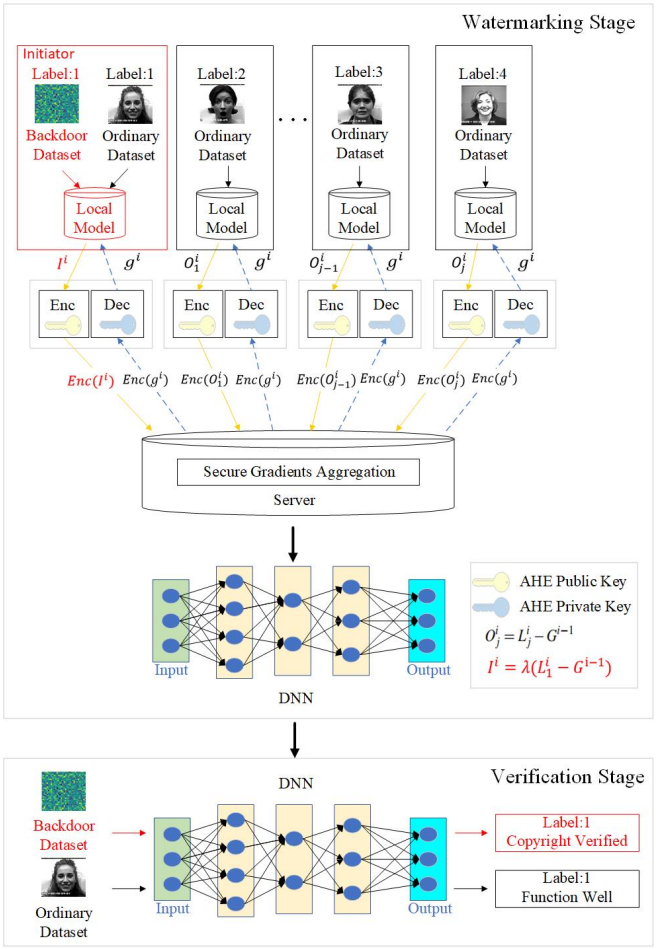

Fig. 1. Watermark and Verification in Secure Federated Learning

watermark into the joint-trained model. As Alice is not guaranteed to be selected at each training iteration to send its gradients to the central node for aggregation, a so-called "scaling factor" λ is defined in our work to improve the embedding process. When Alice is selected to participate into the model update, it uses the following equation to calculate its gradients:

$$O_j^i = L_j^i - G^{i-1} \tag{5}$$

$$I^i = \lambda(L_j^i - G^{i-1}) \tag{6}$$

where O_j^i is the j-th ordinary client's parameters delivered to the server in i-th iteration and I^i represents the initiator's parameters. In Eq. (6), λ is the most important hyper-parameter. We design the most suitable scaling factor setting equation for the FL model learning which is expressed as:

$$\lambda = \frac{N}{n} \tag{7}$$

where n is the number of clients selected at each iteration and N is the number of clients. In condition of a negligible local learning rate, the model weights change consistently in the

training procedure. The gradients of each client within several iterations are approximately equal, which is expressed in Eq. (8). Therefore, it is approximately equivalent to the case that the initiator node is selected every iteration. In this manner, our method has a consistent updating process to improve the stability of model convergence and performs well in watermarking.

$$L_j^i - G^{i-1} \approx L_j^{i+k} - G^{i-1+k}, k = 1,2,\ldots,N/n \tag{8}$$

D. Copyright Verification

The copyright verification process is discussed in this subsection. When Alice decides to verify an unauthorized model deployment by Eva, she recruits a fully-trustworthy third party Trent as their arbitrator. In the copyright verification protocol, they perform the following steps:

1) Alice sends some samples in the backdoor dataset and their corresponding labels to Trent.
2) Trent gets the application programming interface (API) so that he can input the samples to Eva's model and receive outputs.
3) Trent uses Eva's model to make classifications of Alice's samples.
4) Trent checks whether the classifications of those backdoor datasets are in accord with the label provided by Alice. If the accuracy is above the threshold γ, the ownership of the model belongs to Alice and Trent could sentence Eva to infringement.

If a k-classification model does not have a backdoor embedded watermark, it classifies the samples from the backdoor dataset to random labels which probability of the correct classification is $1/k$. We expect the probability of successful copyright verification to be less than 0.01 on the model without backdoor. Assuming that Alice provides n backdoor samples, the threshold γ satisfies:

$$\sum_{j=\lceil \gamma n \rceil}^{n} \binom{n}{j} \left(\frac{1}{k}\right)^j \left(\frac{k-1}{k}\right)^{n-j} \leq 0.01 \tag{9}$$

In our protocol, there are a couple of distinctive advantages. Firstly, Trent does not need to access the weights of Eva's model so that it avoids that Eva cheats Trent by providing fake model weights. As the model is deployed to all the clients of Eva's via the API access, It is impossible for Eva to avoid scrutiny. Secondly, Alice could just provide small number of samples in her backdoor dataset so that sufficient data is preserved for future verification.

IV. EXPERIMENT

A. Experiment Settings

We conduct empirical experiments to demonstrate the effectiveness of the proposed method by using Tensorflow (version 2.3.0) and Tensorflow-Federated (version 0.17.0). Two datasets, including MNIST [26] and CIFAR10 [27] are used in these experiments. The MNIST is a grayscale image dataset with 60,000 training data of 10 digits and additional 10,000 for testing set. While the CIFAR10 consists of 50,000 RGB images for training and 10,000 for testing. In the experiments, we split the data into 100 folds randomly so that each fold of images is allocated to an individual client node to simulate the FL environment. Thus, there are 100 client nodes in total and each client node has 600 MNIST images and 500 CIFAR10 images. They follow the independent identically distribution (IID) assumptions [1]. These nodes in the Federated Learning system contribute their own data to joint-train a global model without leaking their own local data to each other.

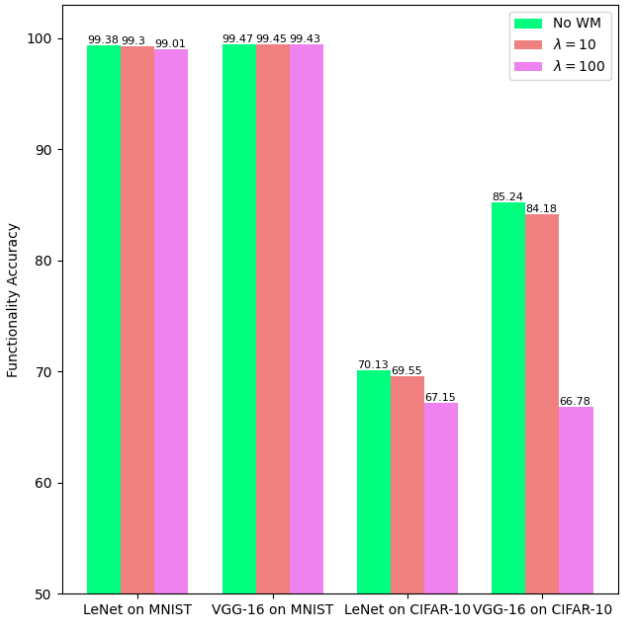

Fig. 2. Functionality-preserving evaluation of the backdoor embedded watermarking models

Two different convolutional neural network architectures are adopted in our work to evaluate the performance of the proposed method. The first CNN is the classical LeNet architecture [28] which consists of two convolution layers each followed by 2×2 max-pooling layer, and two fully connected layers. The second is the VGG-16 backbone network [29], which has 13 convolution layers and 3 fully connected layers.

In the watermarking embedding stage, 100 images of Gaussian noise, which are easily acquired, are randomly generated, and they are set a given label as our trigger set of the backdoor watermarking. Then these images are used together with the data allocated on client nodes to embed the backdoor subtask into the DNN. Here, we train the two DNNs on MNIST classification task with 2,000 iterations and CIFAR10 task with 5,000 iterations respectively. We ensure that the backdoor subtask has been embedded into the two models successfully as 100% accuracy is achieved to classify the backdoor images into their allocated classes.

B. Functionality-preserving

Functionality-preserving is one of the most important requirements for FL model watermarking evaluation. In the

TABLE I

ACCURACY UNDER PRUNING ATTACK

	LeNet				VGG-16			
	MNIST		CIFAR10		MNIST		CIFAR10	
	Functionality Accuracy	Verification Accuracy	Functionality Accuracy	Verification Accuracy	Functionality Accuracy	Verification Accuracy	Functionality Accuracy	Verification Accuracy
10%	99.15%	100.00%	68.91%	100.00%	99.41%	100.00%	83.90%	100.00%
30%	99.20%	100.00%	68.78%	100.00%	99.40%	100.00%	82.81%	100.00%
50%	99.15%	100.00%	67.71%	99.00%	98.28%	92.00%	78.00%	71.00%
70%	98.48%	100.00%	54.18%	90.00%	11.35%	0.00%	14.44%	0.00%

TABLE II

ACCURACY UNDER QUANTIZATION ATTACK

	LeNet				VGG-16			
	MNIST		CIFAR10		MNIST		CIFAR10	
	Functionality Accuracy	Verification Accuracy	Functionality Accuracy	Verification Accuracy	Functionality Accuracy	Verification Accuracy	Functionality Accuracy	Verification Accuracy
8bits	99.14%	100.00%	68.63%	100.00%	99.17%	100.00%	83.93%	100.00%
6bits	99.18%	100.00%	68.54%	100.00%	99.15%	100.00%	83.78%	100.00%
4bits	99.15%	100.00%	67.34%	100.00%	99.12%	100.00%	82.29%	100.00%
2bits	97.14%	100.00%	20.44%	1.00%	11.35%	0.00%	11.76%	0.00%

first experiment, we compare backdoor embedded watermarking FL models with normal models without watermark. As illustrated in Fig. 2, the normal LeNet model and VGG16 model achieve 99.38% and 99.47% accuracy on the MNIST dataset respectively. While the corresponding models with backdoor embedded watermark achieve 99.30% and 99.45% on the same dataset. The performance differences are 0.08% and 0.02% on these two models which are unnoticeable. When these two models are tested on the CIFAR10 dataset, the models without watermark achieves 70.13% and 85.24% respectively. Comparing to the performance of the normal models, the two models with embedded watermark achieves 69.55% and 84.18% and the dropped performance is also insignificant.

In addition, we evaluate the scaling factor for the backdoor watermark embedding process. Since the trigger set is relatively smaller comparing to the training dataset, in [30], a scaling factor $\lambda = 100$ is recommended at the embedding stage. As illustrated in Fig. 2, in our experiments, the scaling factor λ is set to 10 based on the Eq. (7) which is more suitable to preserve the performance of the FL models.

C. Watermark Robustness

Watermark robustness against various of attacks is another important attribute to evaluate FL model watermarking methods. In our work, we conduct experiments to test the robustness of our watermarking method against three typical attacks, including fine-tuning attack, pruning attack, and quantization attack.

1) Fine-Tuning Attack: Model fine-tuning is a popular way to update an existing model slightly so that it could perform well on a similar task. In FL model watermarking attacks, fine-tuning attack refers to the action of removing backdoor watermark from the FL model by fine-tuning the model. In our experiments, we fine-tune the FL watermarking models with 50 iterations. As illustrated in Fig. 3, the verification of the backdoor watermark on the MNIST task only drops insignificantly from 100% to 97% and 96%. The

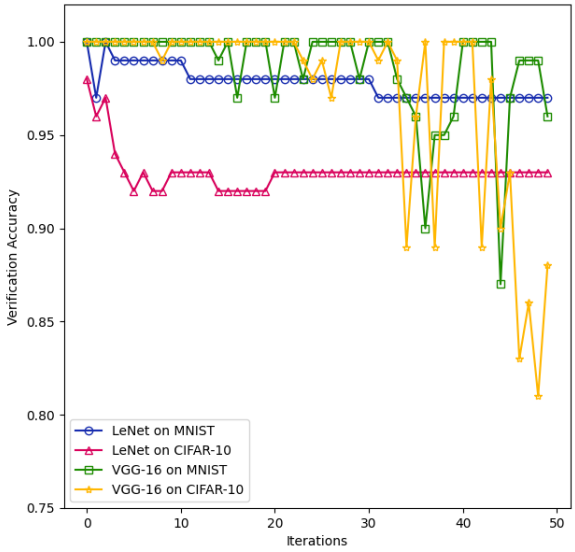

Fig. 3. Verification Accuracy under Fine-Tuning Attack

verification accuracy of both LeNet and VGG models on the CIFAR10 dataset is also acceptable although the drop is relatively larger comparing to the MNIST classification models.

2) Pruning Attack: Pruning attack refers to actions of simplifying a watermarking model by pruning unimportant neurons so that the watermark is removed. By making the pruning operations, the DL model still can keep high accuracy on its classification task. In our experiment, we adopt the network pruning method used in [9] which removes the neurons with the lowest weights.

In Table I, we show the pruning results of all four models with different percentage levels that indicate how many neurons are pruned. As shown in Table I, it is found that the verification accuracy of the backdoor watermark are remarkable at 10%, 20% and 50% pruning levels. Despite the fact that the verification accuracy of MNIST-VGG16 and CIFAR-VGG16 models drop to 0% when 70% neurons

are pruned from the FL models, the performance of the classification models drops to 11.35% and 14.44% which means these attacked models have lost their values.

3) Quantization Attack: Quantization attack refers to using fewer bits to represent weights in order to remove the model watermark. In our experiment, we report the verification accuracy of the four models with different number of bits for quantization. As shown in Table II, when the weights are quantized by using 4 bits, 6 bits and 8 bits, the watermarking models have preserved classification performance. At the same time, the verification accuracy of the backdoor watermark is 100%. Similar to the pruning attack, both the model and the watermark are collapsed when 2 bits is used to quantize VGG-16 models.

V. CONCLUSION

In this paper, we propose a novel client-side watermarking scheme for Secure FL based deep model learning framework. To our best knowledge, this is the first scheme to embed watermark into a DL model learned under a Secure FL environment where there are no other existing watermarking methods can achieve. The proposed method has several distinctive advantages, including the model verification without accessing model parameters, embedding backdoor-based watermark when central node has no access to the gradient information, and robustness to various watermarking removal attacks. We conduct various experiments to prove that our proposed method achieves excellent performance in terms of functionality preserving as well as its robustness against typical counter-watermark attacks including fine-tuning, pruning and quantization.

As for future work, since non-independent identically distributed (non-IID) datasets are one of the greatest challenge in Federated Learning, it remains a significant topic to measure the efficiency of our scheme in non-IID datasets. Besides, other practical problems like communication cost, implementation complexity and time delay should also be carefully considered in future work.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[2] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein *et al.*, "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[3] X. Li, Y. Gu, N. Dvornek, L. H. Staib, P. Ventola, and J. S. Duncan, "Multi-site fmri analysis using privacy-preserving federated learning and domain adaptation: Abide results," *Medical Image Analysis*, vol. 65, p. 101765, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841520301298

[4] B. Liu, B. Yan, Y. Zhou, Y. Yang, and Y. Zhang, "Experiments of federated learning for covid-19 chest x-ray images," *arXiv preprint arXiv:2007.05592*, 2020.

[5] M. Ammad-Ud-Din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *arXiv preprint arXiv:1901.09888*, 2019.

[6] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.

[7] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[8] M. Anderson, "Technology device ownership," 2015, http://www.pewinternet.org/2015/ 10/ 29/ technology-device-ownership-2015.

[9] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, 2017, pp. 269–277.

[10] J. Wang, H. Wu, X. Zhang, and Y. Yao, "Watermarking in deep neural networks via error back-propagation," *Electronic Imaging*, pp. 22–1, 2020.

[11] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proceedings of 27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1615–1631.

[12] J. Zhang, Z. Gu, J. Jang, H. Wu, M. P. Stoecklin, H. Huang, and I. Molloy, "Protecting intellectual property of deep neural networks with watermarking," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018, pp. 159–172.

[13] J. Guo and M. Potkonjak, "Watermarking deep neural networks for embedded systems," in *Proceedings of 2018 IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 2018, pp. 1–8.

[14] S. Szyller, B. G. Atli, S. Marchal, and N. Asokan, "Dawn: Dynamic adversarial watermarking of neural networks," *arXiv preprint arXiv:1906.00830*, 2019.

[15] E. Le Merrer, P. Perez, and G. Trédan, "Adversarial frontier stitching for remote neural network watermarking," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9233–9244, 2020.

[16] L. Fan, K. W. Ng, and C. S. Chan, "Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks," *arXiv preprint arXiv:1909.07830*, 2019.

[17] R. Namba and J. Sakuma, "Robust watermarking of neural network with exponential weighting," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 2019, pp. 228–240.

[18] F. Boenisch, "A survey on model watermarking neural networks," *arXiv preprint arXiv:2009.12153*, 2020.

[19] B. G. Atli, Y. Xia, S. Marchal, and N. Asokan, "WAFFLE: Watermarking in federated learning," *arXiv preprint arXiv:2008.07298*, 2020.

[20] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[21] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[22] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[23] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–35, 2018.

[24] T. Wang and F. Kerschbaum, "RIGA: Covert and robust white-box watermarking of deep neural networks," *arXiv preprint arXiv:1910.14268*, 2019.

[25] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.

[26] L. Y, C. C, and B. C, "Mnist handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[27] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Tech. Rep.*, 2009.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11. Ieee, 1998, pp. 2278–2324.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.