FedIPR: Ownership Verification for Federated Deep Neural Network Models

Lixin Fan\textsuperscript{1}, Bowen Li\textsuperscript{2}, Hanlin Gu\textsuperscript{4}, Yan Kang\textsuperscript{1}, Jie Li\textsuperscript{2} and Qiang Yang\textsuperscript{3}\textsuperscript{*}

\textsuperscript{1}AI Group, WeBank Co., Ltd, Shenzhen, China
\textsuperscript{2}Department of CSE, Shanghai Jiao Tong University, Shanghai, China
\textsuperscript{3}Department of CSE, Hong Kong University of Science and Technology, Hong Kong, China
\textsuperscript{4}Department of Mathematics, Hong Kong University of Science and Technology, Hong Kong, China
\{lixinfan, yangkang\}@webank.com, \{li-bowen, lijiecs\}@sjtu.edu.cn, hguaf@connect.ust.hk, qyang@cse.ust.hk

Abstract

Federated learning models must be protected against plagiarism since these models are built upon valuable training data owned by multiple institutions or people. This paper illustrates a novel federated deep neural network (FedDNN) ownership verification scheme that allows ownership signatures to be embedded and verified to claim legitimate intellectual property rights (IPR) of FedDNN models, in case that models are illegally copied, re-distributed or misused. The effectiveness of embedded ownership signatures is theoretically justified by proved conditions under which signatures can be embedded and detected by multiple clients without disclosing private signatures. Extensive experimental results on CIFAR10, CIFAR100 image datasets demonstrate that varying bit-lengths signatures can be embedded and reliably detected without affecting models classification performances. Signatures are also robust against removal attacks including fine-tuning and pruning.

1 Introduction

Federated learning (FL) is a machine learning setting in which many clients collaboratively train a model, and simultaneously, mitigate privacy risks and costs by keeping the training data decentralized [McMahan \textit{et al.}, 2017; Yang \textit{et al.}, 2019; Kairouz \textit{et al.}, 2019]. While preserving data privacy is of the paramount importance, it is also considered a critical issue to prevent adversaries from stealing and misusing models to search for model vulnerabilities [Kairouz \textit{et al.}, 2019]. Moreover, protecting models from being stolen is motivated by the fact that FL models are built upon valuable data owned by multiple clients, and plagiarism of such models must be stopped. This paper illustrates a novel federated deep neural network (FedDNN) ownership verification scheme that can be used to claim legitimate intellectual property rights (IPR) of FedDNN models, in case that models are illegally copied, re-distributed or misused by unauthorized parties. The proposed scheme not only verifies model ownership against external plagiarisms, but also allows each client to claim contributions to the federated model as verified data owners.

DNN watermarking techniques have been proposed to protect DNN Intellectual Property Rights (IPR) [Uchida \textit{et al.}, 2017; Chen \textit{et al.}, 2018; Darvish Rouhani \textit{et al.}, 2018; Adi \textit{et al.}, 2018; Zhang \textit{et al.}, 2018; Fan \textit{et al.}, 2019; Ong \textit{et al.}, 2021; Zhang \textit{et al.}, 2020; Boenisch, 2020], however, it remains an open question concerning whether existing methods are applicable to federated learning settings, in which following technical challenges must be properly addressed. First, a FL client must embed private signatures into DNN models, yet, without disclosing to other parties the presences and extraction parameters of such signatures. Second, when an increasingly large number of signatures are embedded the global DNN model must be able to accommodate private signatures assigned by different clients without compromising model performances of the main task\textsuperscript{1}. Third, the verification of clients’ private signatures must also be kept secret. In a nutshell, a FedDNN ownership protection scheme should entail three capabilities i.e. maintaining main task model performances, preserving signature privacy and avoiding conflicts between multi-client signatures.

This paper presents a general FedIPR framework which demonstrates a number of successful schemes that can be used for different federated learning scenarios and signature verification protection modes. Specifically, theoretical analysis in Proposition 1 of Sect. 2.2 elucidates conditions under which reliable and persistent signatures can be successfully embedded into the same FedDNN model by multiple clients. Extensive experiments in Sect. 3 demonstrate that varying bit-lengths signatures using normalization scale parameters are very persistent in white-box verification mode, while trigger set of backdoor samples can be reliably detected as strong evidence to support claims of legitimate model ownership. To our best knowledge, the FedIPR framework is the first technical solution that supports the protection of DNN ownerships in a federated learning setting. We believe this work will open new avenues for research endeavor to protect Intellectual Property Right of federated learning models.

\textsuperscript{*}Corresponding Author

\textsuperscript{1}The original task for which the federated DNN model is built.
1.1 Related Work

Privacy-preserving deep learning (PPDL) aims to collaboratively train a deep neural network (DNN) model among multiple clients without exposing private training data to each other [Shokri and Shmatikov, 2015; Abadi et al., 2016; Phong et al., 2018; Ryffel et al., 2020].

Backdoor attacks is a security threat that have been extensively studied in federated learning [Bagdasaryan et al., 2018; Sun et al., 2019; Bhagoji et al., 2019; Wu et al., 2020]. Following [Adi et al., 2018], we adopt targeted backdoor samples [Sun et al., 2019; Bagdasaryan and Shmatikov, 2020] as signatures for black-box ownership verification. We show that robust backdoor signatures can provide evidence of suspected plagiarism without accessing to internal parameters of models.

DNN ownership embedding and verification approaches can be broadly categorized into two schools: a) the feature-based methods that embed designated signatures [Uchida et al., 2017; Chen et al., 2018; Darvish Rouhani et al., 2018; Fan et al., 2019; Zhang et al., 2018]; and b) the trigger-set based methods that rely on backdoor training samples with specific labels [Adi et al., 2018; Zhang et al., 2018]. While feature-based methods allow persistent signatures to be reliably detected even under various forms of removal attacks, yet, they must access DNN internal parameters to detect signatures. The benefit of trigger-set based method is that model owners can collect evidence of suspected plagiarism through remote API without accessing to internal parameters of models in question. Interestingly, [Ong et al., 2021] illustrated a black-box and white-box verification method for GAN instead of convolution networks.

2 Federated DNN Ownership Verification

This section first reviews existing DNN signature embedding and verification scheme, followed by illustration of the Federated DNN (FedDNN) framework. In this paper we use signature and watermark interchangeably. There are broadly two categories of DNN signature embedding and verification methods: feature-based vs trigger-set-based. Feature-based signatures are embedded into network parameters and have to be verified by accessing network parameters i.e. in white-box manner. Trigger-set based signatures are embedded into network outputs or labels, and can be verified without accessing network parameters i.e. in black-box manner.

Figure 1: Convolution layer weights (in yellow) and normalization layer weights (in green) used to embed signatures that are to be extracted in white-box manner (see Sect. 2.1 for texts).

Figure 2: An illustration of federated DNN (FedDNN) signature Embedding and Verification scheme. Private signatures are generated and embedded into the local models which are then aggregated using the FederatedAveraging algo. (the left panel). In case that the federated model is plagiarized, each client may invoke verification processes to extract signatures from the plagiarized model in both black-box and white-box manner to claim his/her ownership of the federated model (the right panel).

2.1 Review of DNN Signature Verification

For existing DNN feature-based signature embedding methods [Uchida et al., 2017; Chen et al., 2018; Darvish Rouhani et al., 2018; Zhang et al., 2018; Fan et al., 2019], N-bits target binary signatures $B = \{b_1, \cdots, b_N\} \in \{0, 1\}^N$ are embedded during the learning of parameters $W$ of a DNN model $\mathbb{N}[W]$, by adding regularization terms $R$ to the loss of the original learning task $L = L_D + \alpha R$:

$$R_{B, \theta}(W) = \text{Dist}(B, B_{\theta}(W)), \quad (1)$$

in which $\theta = \{S, E\}$ is a set of hyper-parameters used to extract signature vector

$$B = (W^TE) \in \mathbb{R}^N \quad (2)$$

whereas $W^T = S(W) \in \mathbb{R}^M$ denotes a $M$-dimensional columnized vector of subset of DNN parameters and $E \in \mathbb{R}^{M \times N}$ a pre-determined signature extraction matrix.

Regularization term (1) restricts DNN model parameter $W$ to be distributed within a subspace (see Proposition 1) such that binary strings extracted from DNN parameters $\hat{B} = \text{sign}(B) = \{b_1, \cdots, b_N\} \in \{0, 1\}^N$ is similar to target signature $B^2$. The signature is then successfully verified by a verification process $V$ if Hamming distance $H(\_)$ is less than a preset radius:

$$V(W, (B, \theta)) = \begin{cases} \text{TRUE,} & \text{if } H(B, \hat{B}) \leq \epsilon_H, \\ \text{FALSE,} & \text{otherwise.} \end{cases} \quad (3)$$

Note that the formulation in (1) is a generalization of regularization terms used in representative DNN signature embedding methods (see Figure 1). For instance, Uchida et al. [Uchida et al., 2017] proposed to embed signatures into convolution layer weights i.e. $S(W)$ is the columnized vector of convolution layer weights, and matrix $E$ is predetermined privately. $\text{Dist}(\_)$ in Uchita et al. [Uchida et al., 2017] measures binary cross-entropy $\text{BCE}(B, \hat{B}) = -\sum_{j=1}^N t_j \log(f_j) + (1 - t_j) \log(1 - f_j)$; where $f_j = \frac{1}{1 + \exp(-b_j)}$.

$^2$An analysis of the existence of solutions to the system of constraining inequalities is given in Appendix B.
Fan et al. [Fan et al., 2019] proposed to embed signatures into normalization layer scale parameters i.e. \( S(W) = W_\gamma = \{\gamma_1, \cdots, \gamma_C\} \) where \( C \) is the number of normalization filters, and matrix \( E \) is \( I_{C \times C} \) identify matrix. Instead, Dist(\( \cdot \)) in Fan et al. [Fan et al., 2019], is the hinge loss

\[
\text{HL}(B, B) = \sum_{j=1}^N \max(\alpha - b_j t_j, 0).
\]

For trigger-set based methods, Adi et al. [Adi et al., 2018] first proposed to embed backdoor trigger-set samples \( T = \{(X_1, Y_1), \cdots, (X_J, Y_J)\} \) by incorporating cross-entropy loss of backdoor samples namely,

\[
L_T(W) = \text{CE}(Y, N(T)) = -\sum_{j=1}^J Y_j \log(N(X_j)),
\]

in which \( X_j \) are backdoor samples, \( Y_j \) corresponding one-hot encoding vector of backdoor labels and \( N(X_j) \) the network softmax outputs.

### 2.2 FedIPR: FedDNN Signature Embedding and Verification

A federated learning system consists of \( K \) client participants building local models with their own data \( D_n \) and send local models to a server-side aggregator for secure model aggregation [McMahan et al., 2017; Yang et al., 2019; Kairouz et al., 2019]. It is often assumed the aggregator and other participants are honest-but-curious and thus no leakage of information from participants is allowed. Federated DNN with ownership verification, requires participants that a) keep local model updates secret from the aggregator; and b) keep ownership verification information secret from the aggregators. The first requirement has been fulfilled by techniques like Homomorphic Encryption [Phong et al., 2018], differential privacy [Abadi et al., 2016] or secret sharing [Ryffel et al., 2020], the second requirement is one of the open problems considered in this work. We give below a formal definition of Federated DNN ownership verification scheme, which is pictorially illustrated in Fig. 2

**Definition 1.** A Federated Deep Neural Network (FedDNN) model ownership verification scheme for a given network \( \mathbb{N}[] \) is a tuple \( \mathcal{V} = (G, E, A, V_W, V_B, V_{G}) \) of processes, consisting of,

I) for client \( k, k \in \{1, \cdots, K\}, \) a client-side key generation process \( G(k) \rightarrow B_k, \theta_k, T_k \) which generates target signature \( B_k, \) signature extraction parameters \( \theta_k = \{S_k, E_k\} \) and a trigger set \( T_k = \{(X_1, Y_1), \cdots, (X_J, Y_J)\} \) of backdoor samples \( X_j \) and corresponding output labels \( Y_j \);

II) a client-side FedDNN embedding process \( E \) which minimizes the combined loss of main task, and two regularization terms to embed trigger set backdoor samples \( T_k \) and signature \( B_k \) respectively3.

\[
L := L_{D_k}(W_k^t) + \alpha_k L_{T_k}(W_k^t) + \beta_k R_{B_k, \theta_k}(W_k^t),
\]

with ClientUpdate(n, \( W \)) = \( W^{t-1} - \eta \frac{\partial L}{\partial W} \) to be sent to the server for updating iteration \( t \);

III) a server-side FedDNN aggregation process \( A \) which collects updates from \( m \) randomly selected clients and performs model aggregation using the FederatedAveraging algorithm [McMahan et al., 2017] i.e.

\[
W_{t+1} = \frac{1}{n} \sum_{k=1}^{K} n_k W_{t_k}^{t+1},
\]

where \( W_{t_k}^{t+1} \leftarrow \) ClientUpdate(\( k, W_t \)) for \( m \) clients,

IV) a client-side white-box verification process \( V_W \) which checks whether signatures extracted from the federated model \( \hat{B}_k = \text{sign}(S_k(W)E_k) \) is similar to the client target signature \( B_k \),

\[
V_{W}(W, \{(B_k, \theta_k)\}) = \begin{cases} \text{TRUE, if } H(B_k, \hat{B}_k) \leq \epsilon_H, \\ \text{FALSE, otherwise}; \end{cases}
\]

V) a client-side black-box verification process \( V_B \) which checks whether the detection error of designated labels \( Y_j \) generated by trigger set backdoor samples \( X_j \) is smaller than \( \epsilon_y \),

\[
V_{B}(\mathbb{N}, T_k) = \begin{cases} \text{TRUE, if } \mathbb{E}_{T_k}(I(Y_j \neq N(X_j)) \leq \epsilon_y, \\ \text{FALSE, otherwise}; \end{cases}
\]

in which \( I(\cdot) \) is the indicator and \( \mathbb{E} \) the expectation over trigger set \( T_n \);

A fundamental challenge for federated DNN signature embedding is to ensure that signatures embedded into local models can be reliably detected from the federated model. For trigger-set based signatures, this seems not an issues as backdoor samples with arbitrarily assigned labels can always be learned with over-parameterized models as demonstrated in [Allen-Zhu et al., 2018; Zhang et al., 2017] (also see Figure 4 (c) and (d)). For feature-based signatures, however, it remains an open question whether there is a common solution \( W \) for different clients to embed their own designated signatures. The following analysis elucidates the condition under which a feasible solution is guaranteed.

**Definition 2.** Let \( U^{M \times KN} \) be matrix combined with \( \{E_1^{M \times N}, E_2^{M \times N}, \cdots, E_K^{M \times N}\} \) by column. Let \( \bar{U}^{M \times KN} \) be matrix combined with

\footnote{A client \( k \) may opt-out and not embed signatures or trigger set backdoor samples by setting \( \alpha_k = 0 \) or \( \beta_k = 0 \). Following [Sun et al., 2019], we adopt random sampling strategy in experiments to assign non-zero values to \( \alpha_k, \beta_k \) to simulate the situation that client make decisions by their own.}
\{(B_1E_1)^{M \times N}, (B_2E_2)^{M \times N}, \ldots, (B_NE_K)^{M \times N}\} \text{, by column, where } B_k = (t_{k1}, t_{k2}, \ldots, t_{KN}) \in \{+1, -1\}^N, \text{ is signature of } k\text{th client.}

In order to satisfy the required condition \(t_{kj}(W^TE_k) > 0\), let’s consider the following two alternatives: (i) \(Ux = 0, x \geq 0\) for some non-zero \(x\); (ii) \(\exists W\) such that \(W^TU \geq 0\). Actually exactly one of the two statements is true according to the Gordan’s theorem ([Alon and Berman, 1986]), which is a simple modifications of Farkas’ Lemma ([Dinh and Jeyakumar, 2014]). And the following propositions 1 gives three conditions of \(U\) to satisfy \(W^TU \geq 0\). The proof is shown in Appendix.

**Proposition 1.** If \(U\) or \(\bar{U}\) as defined above satisfy any one of following conditions, then there exists \(W\) such that \(W^TU \geq 0\).

1. \(\text{rank}(U) = KN\).
2. \(\exists \) all elements of one row of \(\bar{U}^{M \times KN}\) are positive,
3. \(\) the dot product of any two columns of \(\bar{U}^{M \times KN}\) are positive.

In addition, when the feature-based sign loss is binary cross-entropy regularization \(BCE_{\theta}(W^T)\), there exits the common model parameters \(W\) under three conditions such that \(WU\) is less than zero (\(\sigma(WU) < 0.5\)) as target signature \(B\) is 0, or larger than zero as target signature \(B\) is 1.

3 Experiments

This section illustrates the empirical study of our protection framework on the FedDNN models. The network architectures we investigated include the well-known AlexNet and ResNet-18, which are tested with typical CIFAR10 and CIFAR100 classification tasks. In particular, our experiments embed binary signatures to the last convolution layer of AlexNet and ResNet, corresponding to 256 channels and 512 channels of convolution kernel weights \(W^k\) and normalization scale weights \(W^\gamma\). For federated learning setting, we simulate a \(K = 20\) clients horizontal federated learning system in a stand-alone machine. In each distribution round, the server sample clients with uniform distribution.

We adopt adversarial samples as trigger set \(T\), which are trained by Projected Gradient Descent (PGD)[Nguyen et al., 2015], the original data \(T_{source}\) is in the standard benchmark data, trigger \(T\) can mislead the classifier to targeted label designated ahead.

3.1 Evaluation Metrics

To evaluate the FedDNN model signature embedding quantitatively, we use a set of metrics to measure the fidelity and reliability of the proposed feature-based signatures and trigger-set based signatures.

**Fidelity:** we use classification accuracy on the main task \(Acc_{main}\) as the metrics for fidelity. It is expected classification accuracy should not be degraded by the embedding of signatures into the federated model.

**Reliability:** averaged detection rate \(\eta\) of embedded signatures is used to quantify the reliability of a signature verification scheme. For feature-based signatures, detection rate \(\eta_{F}\) is calculated as \(\eta_{F} = 1 - D_{\text{hamming}}\), where \(D_{\text{hamming}}\) measures Hamming distance \(H(B, \bar{B})\) between extracted binary signature string and the target signatures (\(M\) bits length in total). For trigger-set based signatures, detection rate \(\eta_{T}\) is calculated as the ratio of backdoor samples that are classified as designated labels w.r.t. the total number of all trigger set samples.

3.2 Fidelity

Fidelity of the proposed model verification scheme was evaluated under different settings, including varying signature bit length, varying number of triggers per client and different datasets and model architectures.

**Trigger-set Signature:** varying number of clients may decide to embed different number of trigger set samples (as signatures) into the federated model, and Figure 3 (c) and (d) show that model performances of the main task \(Acc_{main}\) remain almost constant when 20 to 600 trigger set samples are embedded by, respectively, each of 5 and 10 clients. There is a negligible accuracy drop (less than 1%) with respect to the model performance without embedding any trigger set signatures.

**Feature-based Signature:** Figure 3 (a) and (b) illustrates model performance \(Acc_{main}\) measured with different length (\(M\)) of binary signatures embedded into normalization layer scale parameter (\(W_{\gamma}\)). It was observed that long bit-lengths (200 bits per client) of signatures lead to slight model performance drop up to 2% for AlexNet on CIFAR10 classification main task. Similar performance drop up to 2% was also observed for ResNet on CIFAR100 classification task, when up to 350 bits signatures were used for each client of 10 clients. The drop of classification accuracy \(Acc_{main}\) is due to the sub-optimal solution restricted to the subspace defined by large number of binary signature constrains (see Proposition 1). Note that performance drop can actually be mitigated by assigning binary signatures across different layers of normalization scale parameters.

3.3 Reliability

Reliability of the proposed model verification scheme was evaluated under different settings, including varying signature bit length, varying number of triggers per client and different datasets and model architectures.

**Trigger-set based signature:** reliability of trigger-set signatures were evaluated under two settings, i.e., \(n_B = 5\) or 10 clients are randomly selected to embed trigger-set signatures generated by Projected Gradient Descent (PGD) adversarial attack method [Nguyen et al., 2015].

Figure 4 (c) and (d) illustrate the trigger set detection rates \(\eta_T\) on these adversarial sample \(T\), respectively, with AlexNet on CIFAR10 classification and ResNet18 on CIFAR100 classification tasks. The results show that the trigger set detection rates \(\eta_T\) almost keep constant even the trigger number per client increases. Moreover, detection rates \(\eta_T\) of signatures embedded in the more complex ResNet18 is more stable than those signatures embedded in AlexNet. Also, the detection

\footnote{The trigger set samples are regarded as correctly detected when the designated targeted adversarial labels are returned.}
Figure 3: Model performances in a federated learning system with 20 clients. Figure (a) and (b), respectively, illustrate CIFAR10 with AlexNet and CIFAR100 with ResNet18 classification accuracy \( \text{Acc}_{\text{main}} \), when \( n_W = 5, 10 \) clients embed varying bit-lengths signatures. Figure (c) and (d), respectively, illustrate CIFAR10 with AlexNet and CIFAR100 with ResNet18 classification accuracy \( \text{Acc}_{\text{main}} \) for \( n_B = 5, 10 \) clients embedding varying number of trigger-set samples.

Figure 4: In a federated learning system of 20 clients, figure (a) and (b), respectively, illustrate the case when \( n_W = 5, 10 \), the signature detection rate \( \eta_F \) with varying bit length per client, figure (a) describes the case that AlexNet with CIFAR10 dataset, figure (b) describes the case that ResNet with CIFAR100 dataset. Figure (c), (d) illustrate the case when \( n_B = 5, 10 \), the trigger detection rate \( \eta_T \) with varying trigger per client, figure (c) describes the case that AlexNet with CIFAR10 dataset, figure (d) describes the case that ResNet with CIFAR100 dataset.

rate is not influenced by the varying number of clients and, thus, varying number of total trigger-set samples used. We ascribe the stable detection rate \( \eta_T \) to the generalization capability of over-parameterized networks as demonstrated in [Allen-Zhu et al., 2018; Zhang et al., 2017].

**Feature-based Signature:** Figure 4 (a) and (b) illustrate binary signature detection rates \( \eta_F \) in white-box manner, in which (a) is with AlexNet for CIFAR10 and (b) with ResNet18 for CIFAR100 classification tasks. First, note that the detection rates \( \eta_F \) remain constant (100%) within the regime, whereas the total bit lengths assigned by multiple \( (n_W = 5, 10) \) clients does not exceed the capacity of network parameters used to embed signatures. This limit is, respectively, 256 and 512 convolution channels at the last layer for AlexNet and ResNet18. Therefore, binary signatures of all bits can be reliably detected, which is in accordance to the analysis disclosed in Proposition 1. When the total bit lengths exceed the limit e.g. in Figure 4 (a), 100 bits signatures are assigned by 5 clients, the detection rate \( \eta_F \) drops to about 80% due to the conflicts of overlapping signature assignments. Nevertheless, the dropped detection rate still guarantees very high confidence in claiming the ownership of verified models.

The results illustrated in Figure 4 give rise to the capability of feature based signature \( B \) into FedDNN model: the bit length of signatures of total clients \( \{M\}_{i=1}^{n_W} \) can not exceed the channel number of normalization scale weights \( W^\gamma \) in selected convolutional layers.

## 4 Robustness

Strategies like Differential Privacy[Wei et al., 2020], Homomorphic Encryption[Phong et al., 2018] and client selection[McMahan et al., 2017] are widely used for privacy and efficiency in federated learning. Those strategies intrinsically bring performance decades on the main classification task. Respectively, we evaluate the detection rate \( \eta \) of signature under those strategies. Moreover, the attacker may try to remove the signatures while inheriting most model performance in federated learning. Specifically, we conduct two removal attacks including fine-tuning and pruning to identify whether the signatures can be reliably detected under those removal attacks.

**Robustness Against Differential Privacy:** we adopt the Gaussian noise-based method to provide differential privacy guarantee for federated learning. Specifically, We vary the standard deviation \( \sigma \) of Gaussian noise on the local gradient before clients send gradients to the server. As Figure 5 (a) shows, the main task performance \( \text{Acc}_{\text{main}} \) decreases severely as the \( \sigma \) of noise increases, while the detection rate of feature-based signature \( \eta_F \) and trigger-based signature \( \eta_T \) drop little while the \( \text{Acc}_{\text{main}} \) is within usable range (more than 85%). In a concrete way, when \( \text{sigma} = 0.003 \), classification accuracy \( \text{Acc}_{\text{main}} \), detection rate \( \eta_F \) and \( \eta_T \) keep a high performance,
Figure 5: Figure describes the robustness of our FedIPR ownership verification scheme: In a federated learning system of 20 clients training AlexNet with CIFAR10 dataset, in which \( n_W = 10 \) clients embed white-box signatures, \( n_B = 10 \) clients embed black-box signatures. The dot lines in figure (a)(b)(c)(d) illustrates the main task classification accuracy \( \text{Acc}_{\text{main}} \) under diverse settings. Respectively, figure (a) illustrates feature-based detection rate \( \eta_F \) and trigger-based detection rate \( \eta_T \) under varying differential private noise sigma; figure (b) illustrates feature-based detection rate \( \eta_F \) and trigger-based detection rate \( \eta_T \) under different sample fraction ratio while federated training; figure (c) illustrates feature-based signature detection rate \( \eta_F \) against model pruning attack with varying pruning rate; figure (d) illustrates feature-based signature detection rate \( \eta_F \) against model finetuning attack in 50 epochs.

which demonstrates the robustness of signature under differential privacy strategy.

**Robustness Against Client Selection:** we decrease the fraction ratio \( c \) of clients selected in each epoch to for communication efficiency. Figure 5 (b) shows that the signature could not be removed even the fraction ratio \( c \) is as low as 0.25. More specifically, when the fraction ratio is larger than 0.2, main classification accuracy \( \text{Acc}_{\text{main}} \) and detection rate \( \eta \) keep constant. This result gives a lower bound of client sampling rate in which signatures can be effectively embedded and verified.

**Robustness Against Pruning:** the target of model pruning is to reduce redundant parameters without compromise the performance. We evaluate the main task performance \( \text{Acc}_{\text{main}} \) and signature detection rate \( \eta \) under pruning attack with varying pruning rate. Figure 5 (c) shows signature detection rate \( \eta \) while varying network parameters are pruned. It was observed that the detection rate \( \eta_T \) of signature embedded in normalization layer is stable all the time, while \( \eta_F \) with \( W_k \) are severely degraded, this fact shows that the signature on normalization parameters are more robust against pruning attack.

**Robustness Against Fine-tuning:** attacks on embedded signatures by fine-tuning were launched to train the network without the presence of the regularization term, i.e., \( L_T \) and \( R_B \). In Figure 5 (d), it was observed that the detection rate \( \eta_F \) of signature embedded with normalization layer (\( W_\gamma \)) remains at 100% (blue curve). In contrast, the detection rate of signature embedded with convolution layer (\( W_k \)) drops significantly (purple curve). The superior robustness of signatures embedded in normalization layer is in accordance to observations reported in [Fan et al., 2019].

### 5 Discussion and Conclusion

This paper illustrated a novel ownership verification scheme to protect Intellectual Property Right (IPR) of Federated DNN models against external plagiarizers who illegally copy, redistribute the models. To our best knowledge, it is the first ownership verification scheme that aims to protect IPR of federated learning models. This work addresses a crucial issue remained open in federated learning research, since the protection of valuable federated learning models is as important as protecting data privacy.

On the technical side, this work demonstrated that reliable and persistent signatures can be embedded into local models without disclosing the presence and extraction parameters of these signatures. In particular, normalization scale parameters based signatures are extremely robust against removal attacks including fine-tuning and pruning. It is our wish that the formulation illustrated in this paper will lead to signature embedding and verification in various federated learning settings.
References


Appendix

A Notation of Parameters

Table 1 summarizes the notation in the whole paper.

B Proof of existence for feature based regularization

In this part, we give a proof of the Proposition 1, which illustrates the three conditions under which reliable and persistent signatures can be successfully embedded into the same FedDNN model by multiple clients.

Proposition 2. If $U$ or $\bar{U}$ as defined above satisfy any one of the following conditions, then there exists $W$ such that $W^T\bar{U} \geq 0$.

1. $\text{rank}(U) = KN$,
2. $\exists$ all elements of one row of $\bar{U}^{M \times KN}$ are positive,
3. The dot product of any two columns of $\bar{U}^{M \times KN}$ are positive.

Proof. For the condition (1), if $\text{rank}(U) = K$, then the column of $U (U_1, U_2, \ldots, U_{KN})$ is independent, and the column of $\bar{U} (\bar{U}_1, \bar{U}_2, \ldots, \bar{U}_{KN})$ is also independent. Thus,

$$y_1\bar{U}_1 + y_2\bar{U}_2 + \cdots + y_{KN}\bar{U}_{KN} = 0$$

$$\iff y_1 = y_2 = \cdots = y_{KN} = 0$$ (9)

Therefore the solution of $\bar{U}y = \bar{0}$ is only $\bar{0}$, moreover, $\bar{U}y = \bar{0}$ doesn’t have non-negative solutions except $\bar{0}$. According to Gordan’s theorem ([Alon and Berman, 1986]), Either $\bar{U}y > 0$ has a solution y, or $\bar{U}y = \bar{0}$ has a nonzero solution y with $\bar{y} \geq \bar{0}$. Since the latter statement is wrong, there exists $W = \bar{y}^T$ such that $W\bar{U} > 0$.

For the second condition, it is obvious that $\bar{U}y = \bar{0}$ doesn’t have non-negative solutions except $\bar{0}$. Therefore the conclusion is true based on Gordan’s theorem.

For the third condition, let the columns of $\bar{U}$ be $(\bar{U}_1, \bar{U}_2, \ldots, \bar{U}_{KN})$. We obtain a contradiction by considering $\bar{U}y = \bar{0}$ has a nonzero solution y with $\bar{y} \geq \bar{0}$, then

$$0 = \bar{U}y = \bar{U}y^T = \sum_{i,j} \bar{U}_i\bar{U}_j^TY_{ij}y_{ij}$$ (10)

Since $y_i \geq 0$ and $\bar{U}_i\bar{U}_j^T > 0, y_{ij} = 0$ for $i = 1, 2, \ldots, KN$. This shows $\bar{U}y = \bar{0}$ has a nonzero solution y with $\bar{y} \geq \bar{0}$, which infers the existence of $W^1 \times M$ such that $W\bar{U} > 0$. □

Remark: The proposition only demonstrates the existence of solution for signature regularization term $R_{B, \theta}(W)$. Embedding signature does not influence the performance of the main task confirmed in experiments, because deep neural networks are typically over parameterized. Deep neural networks have many local minima, whose error very close to the global minimum $\cdot \cdot \cdot \cdot \cdot$. Therefore, the embedding regularizer only needs to guide model parameters to one of a number of good local minima so that the final model parameters embed the signature well.

C Experiment Settings

This section illustrates the experiment settings of the empirical study on our FedIPR framework for the FedDNN models.

DNN Model Architectures. The deep neural network architectures we investigated include the well-known AlexNet and ResNet-18. Feature-based binary signatures are embedded into convolution kernel weights $W^k$ and normalization scale weights $W^v$ of multiple convolution layers in AlexNet and ResNet-18. Table 2 shows the detailed model architectures and parameter shape of AlexNet and ResNet-18, which we employed in all the experiments.

Dataset. We evaluate classification tasks on standard CIFAR10 dataset and CIFAR100 dataset. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. CIFAR-100 has 100 classes containing 600 images each, there are 500 training images and 100 testing images per class. Respectively, we conduct stand image classification tasks of CIFAR10 and CIFAR100 with AlexNet and ResNet-18. According to the way we split the dataset for clients in federated learning, the experiments are divided into iid setting and non-iid setting. The results for both IID setting and Non-IID federated learning setting are provided in the section ??.

Federated Learning Settings. We simulate a with $K = 20$ clients horizontal federated learning system in a stand-alone machine with 8 Tesla V100-SXM2 32 GB GPUs and 72 cores of Intel(R) Xeon(R) Gold 61xx CPUs.

In each communication round, the server samples clients with uniform distribution of a certain fraction ratio to participate training (the fraction ratio we explore includes 1.0, 0.8, 0.5, 0.2, 0.1 and 0.05). The clients update the weight updates, server adopts Federatedavg([McMahan et al., 2017] algorithm in to aggregate the model updates. The detailed experiment Hyper parameters we employ to conduct our federated learning are listed in the table 4.

Embedding Process $E$

Feature-based signature: For the feature-based signature embedding scheme, we constrain the sign of network parameters $W^v$ and $W^k$ with regularization terms $R_{B, \theta}$ including Hinge Like loss, and cross-entropy loss targeted at different bit length of each client. We change the number of clients and the bit length of signature per client under diverse federated learning setting. The algorithm is shown in Algorithm 2.

Trigger-set Signature: The trigger-set embedding process adopts a batch-poisoning backdoor method: in each iteration of backdoor training, both normal samples and backdoor samples are used in the same data batch for model training.

We adopt adversarial samples as trigger set $T$: we train the adversarial samples with Projected Gradient Descent (PGD)[Nguyen et al., 2015], from original data $T_{source}$ in the standard benchmark data. The PGD parameters are listed in the table 5. After the training process of PGD, the trigger $T$ can mislead the classifier to targeted label designated ahead.

Verification process $V$

After obtaining the model, we could extract the signature and test the trigger set from the model. Verification process of signature and trigger-set is shown Algorithm 3 and 4.

Removal attack
Table 1: Notation description

<table>
<thead>
<tr>
<th>W</th>
<th>Model weights</th>
<th>( W^k )</th>
<th>Convolution kernel weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>Model outputs</td>
<td>( O^k )</td>
<td>Middle layer activation outputs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( O^i )</td>
<td>Images outputs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( O^c )</td>
<td>Classification labels outputs</td>
</tr>
</tbody>
</table>

**Key generation Process: G**

<table>
<thead>
<tr>
<th>B</th>
<th>Target signature</th>
<th>( \tilde{B} )</th>
<th>( \text{sign}(B) )</th>
<th>( N )</th>
<th>Bits number of targeted signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Dimension of ( B )</td>
<td>( \theta = {S, E} )</td>
<td>Hyper-parameters of extracting signature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Trigger set</td>
<td>( K )</td>
<td>Number of clients</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Embedding Process: E**

<table>
<thead>
<tr>
<th>( L_D )</th>
<th>Main task loss</th>
<th>( L_T )</th>
<th>trigger-set-based loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{B, \theta} )</td>
<td>Feature-based regularization</td>
<td>( \text{HL}() )</td>
<td>Hinge loss</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{BCE}() )</td>
<td>Cross entropy loss</td>
</tr>
</tbody>
</table>

**Aggregation Process: A**

| \( n_k \) | The aggregate weights for \( k_{th} \) clients |

**Verification Process: V**

<table>
<thead>
<tr>
<th>( V_W )</th>
<th>white-box verification</th>
<th>( \epsilon_W )</th>
<th>threshold of signature detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_B )</td>
<td>black-box verification</td>
<td>( \epsilon_B )</td>
<td>threshold of signature detection</td>
</tr>
</tbody>
</table>

Following previous DNN watermarking methods, we report model performances under fine-tuning and pruning attacks. For **finetuning**, we adopt the code\(^5\) and follow their implementation in Algorithm 5. For **pruning**, we adopt the code\(^6\) and follow their implementation in Algorithm 5. The hyper parameters are shown in table 3.

---

\(^{5}\text{https://github.com/dingsheng-ong/ipr-gan}\)
\(^{6}\text{https://github.com/zepx/pytorch-weight-prune/blob/develop/pruning/methods.py}\)
<table>
<thead>
<tr>
<th>layer name</th>
<th>output size</th>
<th>weight shape</th>
<th>padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>32 × 32</td>
<td>64 × 3 × 5 × 5</td>
<td>2</td>
</tr>
<tr>
<td>MaxPool2d</td>
<td>16 × 16</td>
<td>2 × 2</td>
<td></td>
</tr>
<tr>
<td>Conv2</td>
<td>16 × 16</td>
<td>192 × 64 × 5 × 5</td>
<td>2</td>
</tr>
<tr>
<td>MaxPool2d</td>
<td>8 × 8</td>
<td>2 × 2</td>
<td></td>
</tr>
<tr>
<td>Conv3</td>
<td>8 × 8</td>
<td>384 × 192 × 3 × 3</td>
<td>1</td>
</tr>
<tr>
<td>Conv4</td>
<td>8 × 8</td>
<td>256 × 384 × 3 × 3</td>
<td>1</td>
</tr>
<tr>
<td>Sign Embedding (W_k)</td>
<td>8 × 8</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>Conv5</td>
<td>8 × 8</td>
<td>256 × 256 × 3 × 3</td>
<td>1</td>
</tr>
<tr>
<td>Sign Embedding (W_γ)</td>
<td>8 × 8</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>MaxPool2d</td>
<td>4 × 4</td>
<td>256 × 256 × 3 × 3</td>
<td>1</td>
</tr>
<tr>
<td>Linear</td>
<td>10</td>
<td>10 × 4096</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Left: modified AlexNet architecture. Right: modified ResNet-18 architecture

<table>
<thead>
<tr>
<th>Hyper-parameter</th>
<th>Removal Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pruning Rate</td>
<td>0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9</td>
</tr>
<tr>
<td>Finetuning Learning Rate</td>
<td>0.0001</td>
</tr>
<tr>
<td>Finetuning Learning Loss</td>
<td>only Cross Entropy loss</td>
</tr>
<tr>
<td>Batch size</td>
<td>16</td>
</tr>
<tr>
<td>Finetuning Epochs</td>
<td>50</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.01</td>
</tr>
<tr>
<td>Learning rate decay</td>
<td>0.01 per epoch</td>
</tr>
<tr>
<td>Vanilla Classification model</td>
<td>CNN with three convolution layers</td>
</tr>
</tbody>
</table>

Table 3: Training parameters for Removal Attack

<table>
<thead>
<tr>
<th>Hyper-parameter</th>
<th>AlexNet</th>
<th>ResNet-18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation function</td>
<td>ReLU</td>
<td>ReLU</td>
</tr>
<tr>
<td>Optimization method</td>
<td>SGD</td>
<td>SGD</td>
</tr>
<tr>
<td>Momentum</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Batch size</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Backdoor batch size</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Data Distribution</td>
<td>IID and non-IID</td>
<td>IID and non-IID</td>
</tr>
<tr>
<td>Global Epochs</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Local Epochs</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Learning rate decay</td>
<td>0.99 at each global Epoch 0.99 at each global Epoch</td>
<td></td>
</tr>
<tr>
<td>Federated Fraction</td>
<td>[He et al., 2015]  [He et al., 2015]</td>
<td></td>
</tr>
<tr>
<td>Client numbers</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Feature-based Signature Client numbers</td>
<td>5,10</td>
<td>5,10</td>
</tr>
<tr>
<td>Regularization Term</td>
<td>BCE loss, Hinge-like loss</td>
<td>BCE loss, Hinge-like loss</td>
</tr>
<tr>
<td>α of Regularization Loss</td>
<td>0.2, 0.5, 1, 5 0.2, 0.5, 1, 5</td>
<td></td>
</tr>
<tr>
<td>Feature-based Signature parameters W</td>
<td>W_k and W_γ W_k and W_γ</td>
<td></td>
</tr>
<tr>
<td>Trigger-based Signature Client numbers</td>
<td>5,10</td>
<td>5,10</td>
</tr>
<tr>
<td>Trigger-based Signature type</td>
<td>Adversarial sample</td>
<td>Adversarial sample</td>
</tr>
</tbody>
</table>

Table 4: Training parameters for Federated AlexNet_p and ResNet-p-18, respectively († the learning rate is scheduled as 0.01, 0.001 and 0.0001 between epochs [1-100], [101-150] and [151-200] respectively).
Hyper-parameter | Projected Gradient Descent
--- | ---
Optimization method | Projected Gradient Descent
Norm type | L2
Norm of noise | 0.3
Learning rate | 0.01
PGD Batch size | 128
Targeted at Specific Labels | True
Iterations | 80
Learning rate decay | None
Vanilla Classification model | CNN with three convolution layers

Table 5: Training parameters for Projected Gradient Descent Adversarial Training

**Algorithm 2** Signature Embedding Process for FedIPR
1. Each client $k$ with its own signature tuple $(B_k, \theta_k, T_k)$
2. for $i$ in communication round $E$ do
3. Server distributes the global model parameters $W_i$ to each client
4. Sample clients with fraction ratio $C$ into subset $s$ of $K$ clients
5. **Local Training:**
6. for $k$ in number of selected users subset $s$ do
7. Sample minibatch of $m$ samples $X \{X^{(1)}, \ldots, X^{(m)}\}$ and targets $Y \{Y^{(1)}, \ldots, Y^{(m)}\}$
8. if enable backdoor then
9. sample $t$ samples of $T_k$ and backdoor targets $Y_{T_k}$ \( t = 2 \), default by [Adi et al., 2018]
10. concatenate $X$ with $T$, $Y$ with $Y_{T_k}$
11. compute cross-entropy loss $L_c$ using $X$ and $Y$
12. for layer $l$ in targeted layers set $L$ do
13. compute Regularization term $R_l$ using $\theta_k$ and $W_l$
14. $W_i \leftarrow \sum_{l \in L} R_l$
15. $L = L_c + R$
16. Backpropagate using $L$ and update $W_i^k$
17. **Server Update:**
18. Aggregate the $\{W_i^k\}_{k=1}^K$ with FederateAvg algorithm

**Algorithm 4** Black-box Trigger-set Based Signature Verification
**Input:** Model $N$ offered by adversaries, Trigger set $T$ and $Y_T$ provided by user.
1. **procedure** TRIGGER-SET DETECTION
2. Fed the Trigger-set $T$ into model $(N)$ to derive the classification label $O_c$
3. Match $O_c$ with target label of target trigger-set label $Y_T$
4. Compute the trigger-set detection rate $V_B(T, Y_T, W)$

**Output:** $V_B(T, Y_T, W)$

**Algorithm 5** Removal attack
**Input:** Model $N$, trigger set $T$ and $Y_T$, target signature $B$
1. **procedure** PRUNING
2. for $p$ in different pruning percentage do
3. Pruning the model $N$ in $p$ percentage.
4. Test the signature and trigger-set detection rate.
5. **procedure** FINE TUNING
6. for epochs in 50 do
7. Train the model $N$ only in main task (classification task)
8. Test the signature and trigger-set detection rate.
D Ablation Study

D.1 Influence of feature-based signature regularization parameter $\alpha$

In this section, we test the influence which the feature-based signature regularization parameter $\alpha$ brings. Our experiments demonstrates $\alpha$ only affects the fidelity.

The left image of Figure 6 shows the model performance drops seriously as $\alpha$ increases, especially when $\alpha$ equals 1 and 5. The right image of Figure 6 explains the reliability keeps the similar trend even the $\alpha$ changes from 0.2 to 5.

D.2 Diversity of embedding position of signature

We embed feature-based signatures into last two layers of AlexNet to explore the capacity of white-box embedding. As shown in the Table 2, the Conv. layer 4 and Conv. layer 5 have each 256 channels of convolution kernels, we test the fidelity, reliability of white box signature. We compare the results of the case with single Conv. layer 5 and the case with Conv.4 and Conv.5, the results is described in the Figure 7.

Figure 7 Left shows that the signature embedding into multiple layers yields no compromise of fidelity, the main task slightly decodes as the bit length increases. Figure 7 Right describe the signatures into multiple layers of the neural network amplify the capacity of signatures, because the multiple layers enable more bit length of signatures into the model. Two-layer case enables twice the bit length of signature as one-layer case in the same detection signature rate.

This result also proves that: the bit length of signatures of total clients $\{M_i\}_{i=1}^N$ can not exceed the channel number of normalization scale weight $W^N$ in selected convolution layers, which is consistent with Proposition 1.

D.3 Signature into Kernel Weights

The parameters $W$ chosen for embedding signatures includes convolution layer weights $S(W)$ (the columnized vector of convolution layer weights) and normalization layer scale parameters $S(W) = W_z = \{\gamma_1, \cdots, \gamma_C\}$ where $C$ is the number of normalization filters.

The table 6 illustrates the main task classification accuracy (fidelity) with the increase of bit length of signature, the model performance is slightly affected only when the signature embedding conflict with each other. Figure 8 illustrate the reliability (signature detection rate) of signature detection on convolution weights $W^K$. Convolution kernel parameters naturally have more parameters for embedding signatures, so the capacity is correspondingly larger the the case with normalization weights $W^K$.

Remark The blue line in Figure 8 does not decrease because the total signature length $KN = 500 \times 5 = 2500$ is closed to number of embedding weights $256 \times 9 = 2294$

D.4 Cross Entropy Loss

The regularization term we employ for signature embedding include both binary cross entropy (BCE) loss and Hinge-like (HL) loss: Binary cross-entropy BCE $\mathbf{BCE}(B, B') = -\sum_{j=1}^N t_j \log(f_j) + \log(1 - f_j)$; where $f_j = \frac{1}{1+\exp(-b_j t_j)}$, and Hinge loss $\mathbf{HL}(B, B') = \sum_{j=1}^N \max(\alpha - b_j t_j, 0)$, where signatures $B = (t_1, \cdots, t_N) \in \{-1, 1\}^N$.

We conduct experiments with the same setting of 20 clients in BCE regularization and Hinge regularization, whose results show both two approaches are influenced with signature length similarly.

Specifically, when the bit length is in the capacity of signature embedding, the fidelity and reliability between BCE loss and Hinge loss are the same (shown in Table 7. Moreover, when the signature embedding conflict with each other, the reliability of BCE loss is slightly better then the HL loss.

To conclude, the HL regularization term is a stronger constrain than BCE regularization term, when the diverse signature embedding of clients conflict with each other, hinge like loss affects more fidelity and reliability.
Figure 6: Results of CIFAR10 with AlexNet when embedding signature in different regularization parameter $\alpha$. Left is main task classification accuracy as the signature length number varies in different $\alpha$. Right images is signature detection rate with different number of signature length in different $\alpha$.

Table 6: Model classification accuracy with embedding in convolution layer ($W_k$) under two conditions (5 or 10 clients add signature)

<table>
<thead>
<tr>
<th>Bits Number</th>
<th>30</th>
<th>60</th>
<th>90</th>
<th>120</th>
<th>150</th>
<th>180</th>
<th>210</th>
<th>240</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Acc</td>
<td>0.9135</td>
<td>0.9131</td>
<td>0.9132</td>
<td>0.9129</td>
<td>0.9125</td>
<td>0.9123</td>
<td>0.9128</td>
<td>0.913</td>
</tr>
<tr>
<td>Bits Number</td>
<td>270</td>
<td>300</td>
<td>330</td>
<td>360</td>
<td>390</td>
<td>420</td>
<td>450</td>
<td>480</td>
</tr>
<tr>
<td>Model Acc</td>
<td>0.9142</td>
<td>0.914</td>
<td>0.9125</td>
<td>0.9132</td>
<td>0.9114</td>
<td>0.9121</td>
<td>0.9122</td>
<td>0.9094</td>
</tr>
</tbody>
</table>

Table 7: Model classification accuracy in BCE loss and Hinge loss under two conditions (5 or 10 clients add signature)

<table>
<thead>
<tr>
<th>Bits Len.</th>
<th>BCE loss</th>
<th>Hinge loss</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Client5</td>
<td>Client10</td>
</tr>
<tr>
<td>20</td>
<td>0.9152</td>
<td>0.9157</td>
</tr>
<tr>
<td>40</td>
<td>0.9146</td>
<td>0.91</td>
</tr>
<tr>
<td>60</td>
<td>0.9137</td>
<td>0.9087</td>
</tr>
<tr>
<td>80</td>
<td>0.9136</td>
<td>0.9136</td>
</tr>
<tr>
<td>100</td>
<td>0.9135</td>
<td>0.9123</td>
</tr>
<tr>
<td>120</td>
<td>0.9123</td>
<td>0.913</td>
</tr>
<tr>
<td>140</td>
<td>0.912</td>
<td>0.9124</td>
</tr>
<tr>
<td>160</td>
<td>0.9118</td>
<td>0.9139</td>
</tr>
<tr>
<td>180</td>
<td>0.9113</td>
<td>0.9082</td>
</tr>
<tr>
<td>200</td>
<td>0.9111</td>
<td>0.9121</td>
</tr>
</tbody>
</table>
Figure 7: Results of embedding signature into last two normalization layers ($W^\gamma$ of AlexNet). Left image is the model classification accuracy in different signature length; right is the signature detection rate in different signature length.

Figure 9: Results in embedding signature into last normalization layer ($W^\gamma$ of AlexNet with two different regularization: Hinge regularization and BCE regularization. Left image is the comparison of signature detection rate between Hinge loss and BCE loss when 5 clients choose to add signature; right is similar comparison when 10 clients choose to add signature.