Label Inference Attacks against Federated Unlearning

Wei Wang[1][0009−0009−0625−4603], Xiangyun Tang[1⋆][0000−0002−5511−0720], Yajie Wang[2⋆][0000−0002−7023−4844], Yijing Lin[3][0000−0003−2702−7679], Tao Zhang[4][0000−0002−2639−4357], Meng Shen[2][0000−0001−5706−6383], Dusit Niyato[5][0000−0002−7442−7416], and Liehuang Zhu[2][0000−0003−3277−3887]

[1] the Key Laboratory of Ethnic Language Intelligent Analysis and Security Management of MOE, Minzu University of China, Beijing, China
{wangwei,xiangyunt}@muc.edu.cn

[2] School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China
wangyajie0312@foxmail.com

[3] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China

[4] School of Computer Science and Engineering, Beijing Jiaotong University, Beijing, China

[5] School of Cyberspace Science and Technology, Nanyang Technological University, Singapore

Abstract. Federated Unlearning (FU) has emerged as a promising solution to respond to "the right to be forgotten" of clients, by allowing clients to erase their data from global models without compromising model performance. Unfortunately, researchers find that the parameter variations of models induced by FU expose clients' data information, enabling attackers to infer the label of unlearning data, while label inference attacks against FU remain unexplored. In this paper, we introduce and analyze a new privacy threat against FU and propose a novel label inference attack, `ULIA`, which can infer unlearning data labels across three FU levels. To address the unique challenges of inferring labels via the models variations, we design a gradient-label mapping mechanism in `ULIA` that establishes a relationship between gradient variations and unlearning labels, enabling inferring labels on accumulated model variations. We evaluate `ULIA` on both IID and non-IID settings. Experimental results show that in the IID setting, `ULIA` achieves a 100% Attack Success Rate (ASR) under both class-level and client-level unlearning. Even when only 1% of a user's local data is forgotten, `ULIA` still attains an ASR ranging from 93% to 62.3%.

Keywords: Federated Unlearning · Label Inference Attack · Gradient-Label Mapping · Federated Learning · Privacy Protection.

[⋆] The corresponding authors: Xiangyun Tang (xiangyunt@muc.edu.cn), Yajie Wang (wangyajie0312@foxmail.com). This work was supported in part by the National Nature Science Foundation of China under Grant 62302539, 62402040 and the Key Laboratory of Ethnic Language Intelligent Analysis and Security Management of MOE (OPU-202502)

1 Introduction

Federated Learning (FL), as a decentralized machine learning paradigm, has gained widespread adoption in various domains such as finance [18] and smart cities [8] due to its inherent capability to protect user privacy. FL allows multiple users to jointly train a global model by sharing local models rather than raw data with a central server, thereby mitigating privacy leakage [19]. In addition to the privacy protection, existing data security legislation, such as General Data Protection Regulation (GDPR) [24] and California Consumer Privacy Act (CCPA) [5], emphasizes the "Right to be Forgotten", affirming users' authority to demand the unlearning of their data from global models during FL.

To respond the unlearning demand, Federated Unlearning (FU) has emerged as a promising solution [17]. FU methods, such as historical information-based unlearning [32] and rapid retraining [16], involve the server collaborating with users to remove the influence of unlearning data from the global model, in accordance with users' unlearning requests, while ensuring that the model's performance remains consistent with its state prior to the unlearning operation. FU can be categorized into sample-level, class-level, and client-level [22], where the unlearning requests correspond to a set of samples, all samples associated with a set of classes, or the entire local dataset of a user, respectively.

Although existing FU methods can unlearn data from global models while preserving model performance, they are vulnerable to a significant privacy leakage threat. After FU operations, the server holds two versions of models before and after unlearning. The adjustments of the resulting parameter on the models are not entirely independent but are closely related to the characteristics of the unlearning data, such as the labels of the unlearning data [21]. Hence, *the variations in the models before and after FU expose users' data information*, enabling the server as attackers to infer private information about the unlearning data. In this paper, we focus on label inference attacks against FU, where the server, by analyzing the variations of models before and after unlearning, infers the labels of unlearning data.

Label inference attacks allow the server to steal private labels of users that should have been unlearning or protected in FU, raising significant privacy concerns. Furthermore, ensuring the privacy of labels is a fundamental guarantee, as they often represent sensitive or critical information for the participant [20]. For example, in a healthcare scenario, multiple medical institutions collaboratively train a global model using diabetic patient data. If a patient requests the removal of their medical data due to privacy concerns, the label inference attacks on FU enable the attacker to infer the diagnosis, leading to a severe breach of privacy.

However, the privacy risks of label inference attacks on FU remain underexplored. It is challenging to infer the labels of unlearning data from the model differences induced by FU. The model differences reflect the accumulated impact of all the unlearning data and their associated labels. Consequently, when attackers are unaware of the number of unlearning data samples or labels, accurately inferring labels from the accumulated parameter differences is challenging. Furthermore, when the quantity of unlearning data is small, the resulting changes in model parameters may be too subtle to adequately capture the features of the unlearning data, making it more difficult for attackers to infer the labels.

In this paper, we propose ULIA, a novel label inference attack that infers the labels of unlearning data across three FU levels: sample-level, class-level, and client-level, by analyzing the variations in the model parameters of FU. ULIA addresses the inherent challenge of accurately inferring labels from model differences induced by FU, where the accumulated impact of unlearning data on model parameters obscures individual label effects. To overcome this, we propose a *gradient-label mapping mechanism*, which establishes a relationship between gradient variations and unlearning labels. This allows the attacker to separate the specific parameter shifts attributable to individual unlearning label, thus enhancing the accuracy of label inference. Moreover, ULIA incorporates a dynamic filtering strategy that prioritizes label categories with the higher likelihood of matching model parameter changes, focusing on those labels that exhibit the most significant alterations, ensuring effective label inference, even in scenarios with sparse or subtle unlearning data.

We evaluate the performance of ULIA under three advanced FU methods across the three FU levels on real-world datasets. In the IID setting, ULIA achieves an Attack Success Rate (ASR) ranging from 100% to 62.3%. Even in the non-IID setting, ULIA is still able to achieve $96.4\% - 58.5\%$ ASR.

The main contributions of this paper are as follows:

- To the best of our knowledge, we are the first to reveal the label leakage issue of FU. We propose ULIA, a novel attack inferring unlearning data labels across three FU levels, by analyzing the variations in the model parameters induced by unlearning operations.
- The attack works regardless of the quantity of unlearning data, as we design a gradient-label mapping mechanism that establishes a relationship between gradient variations and unlearning labels, enabling inferring unlearning labels on accumulated model variations.
- We evaluate our attacks with real-world datasets under three advanced FU methods, both in IID and non-IID settings. The experimental results show that ULIA demonstrates outstanding attack performance and adaptability.

2 Related Works

Federated Unlearning. Existing FU methods can be broadly categorized into two main approaches: (1) *Historical information-based unlearning*, which is recorded and analyzed during training to assess the impact of specific data or clients on the global model, enables efficient unlearning. The typical techniques explored under this approach include: Gradient correction adjusts the gradient information in the model training process to eliminate the contribution of target data or target clients to the global model [32]. The knowledge distillation strategy restores the performance of the global model through knowledge transfer and model tuning, thereby approximating unlearning [27]. Approximate unlearning of target data is achieved by pruning network layers [25] and refining the loss function [9]. (2) *Rapid retraining*, which efficiently restores the unlearning global model by optimizing retraining algorithms or performing partial retraining. The typical techniques explored under this approach include: First-order Taylor expansion methods are employed to effectively utilize gradient and curvature information, thereby identifying a more optimal descent direction [16]. The optimal number of unlearning rounds can be accurately determined by assessing the contribution of target clients during each training iteration [13]. Clients are grouped into suitable clusters for aggregation, ensuring that retraining occurs solely within the cluster of the target client during the unlearning process [23].

Label Inference Attacks. Label inference attacks aim to infer the labels of training or unlearned data, thereby compromising the privacy of participating clients. Previous studies have primarily implemented label inference attacks through the following techniques. (1) *Based on gradient information*, where attackers exploit the label information reflected in the gradient signs and magnitudes to perform inference. [31] discovers the relationship between labels and gradient signs in the cross-entropy loss function, which becomes a fundamental basis for label inference attacks. [4] demonstrates that attackers exploit inherent vulnerabilities in vertical federated learning (VFL) to infer sensitive labels owned by one party through the output features of the bottom model or the gradient information returned by the server. (2) *Classification and clustering methods*, where attackers train gradient classifiers or utilize the clustering properties of embeddings and gradients to make inferences. [12] observes that in two-party scenarios, the gradient norm associated with target labels is often larger than that of non-target labels, providing a basis for classification-based methods. [15] proposes a K-means clustering-based attack method that classifies gradients or embeddings using cosine similarity. (3) *Model reconstruction*, where attackers simulate the predictive model and labels of the active party to construct surrogate models and labels, minimizing prediction errors to infer the true labels. [30] proposes using label smoothing techniques to prevent the model from becoming overconfident in label predictions. [1] introduces an early stopping strategy to reduce the impact of gradient magnitude peaks on the attack accuracy.

Highlights of the proposed attack. Existing label inference attacks mainly focus on inferring the labels of training data. With the emergence of FU, the analysis of local and global model parameters' changes generated by unlearning requests to infer the labels of forgotten data has still remained unexplored. In this paper, we propose `ULIA`, a novel label inference attack against FU, inferring unlearning data labels by analyzing the model variations induced by unlearning operations, which reveals and analyzes the new privacy leakage issue of FU.

3 Preliminaries

Before entering the sample-level FU process, clients participate in FL, training global models under the server's orchestration. The FU process is launched when a client launches an unlearning request. The server and clients cooperate to eliminate the target data from the global model, based on a unlearning strategy.

Federated Learning. The primary objective of FL is to enable collaborative and efficient training on distributed data without directly sharing the raw data. Given N clients P_i $(i = 1, \ldots, N)$, each holding local data D_i, the overall dataset is defined as $D = \bigcup_{i=1}^{N} D_i$. At the beginning of the FL training process, the server initializes the global model $\theta_{\text{global}}^{(0)}$ and distributes it to all clients. At the t-th round, client i optimizes the global model $\theta_{\text{global}}^{(t)}$ based on its local data D_i, and updates the local model parameters $\theta_{\text{local},i}^{(t+1)}$:

$$\theta_{\text{local},i}^{(t+1)} = \theta_{\text{global}}^{(t)} - \eta \cdot \sum_{(x,y)\in D_i} \nabla \ell(f_\theta(x), y) \tag{1}$$

where η is the learning rate, and $\nabla \ell(f_\theta(x), y)$ is the gradient of the loss function ℓ for model f_θ with respect to input x and label y.

The server aggregates the local model parameters $\theta_{\text{local},i}^{(t+1)}$ uploaded by all participating after their individual training steps, in order to update the global model parameters $\theta_{\text{global}}^{(t+1)}$:

$$\theta_{\text{global}}^{(t+1)} = \sum_{i=1}^{N} w_i \theta_{\text{local},i}^{(t+1)} \tag{2}$$

where $w_i = \frac{|D_i|}{|D|}$ represents the weight of client i based on the size of its local dataset D_i relative to the total dataset D.

Federated Unlearning. When a target client K sends an unlearning request in the t-th round, FU performs the data removal request, resulting in the post-unlearning global model θ'_{global}. FU can be categorized into the following three main types based on different forgetting requests and objectives.

- *Sample-level unlearning* seeks to remove specific sensitive samples from a client's local dataset and eliminate their influence on the global model to protect privacy [29]. The local dataset D_i is updated by removing S_f, resulting in a revised dataset $D'_i = D_i \setminus S_f$. The client initializes its post-unlearning local model $\theta'_{\text{local},i}$ with the global model parameters $\theta_{\text{global}}^{(t)}$ from the t-th round and retrains it on the updated dataset D'_i.
- *Class-level unlearning* focuses on completely erasing certain class-specific information from the global model, renders the model incapable of classifying those classes [2,26]. When the influence of a specific class C_f needs to be removed, each client updates its local dataset by removing the data associated with C_f, resulting in modified datasets $D'_i = D_i \setminus \{(x, y) \mid y \in C_f\}$.
- *Client-level unlearning* aims to fully remove the contributions of specific clients and to ensure that their data has no residual impact on the global model or subsequent training processes [28]. The global model removes the influence of the target clients K and updates using only the remaining clients. The remaining dataset is represented as $D' = \bigcup_{\substack{i=1 \\ i\neq K}}^{N} D_i$.

4 Attack Model and Problem Formalization

4.1 Attack Model

This paper focuses on label inference attacks against FU. When a target client submits an unlearning request to remove samples from the global model or to exit collaborative training, the server and client participate in FU to ensure that the model no longer reflects the influence of the forgotten data. We assume that the server is semi-honest. By exploiting access to the model parameters, the semi-honest server as the attacker aims to infer the labels of the data that the target client has requested to forget.

Attacker's Goal. The attacker's goal is to infer the labels of the unlearning data by analyzing the parameter differences of models before and after unlearning. Our attack specifically targets a single client's unlearning request but considers different types of forgotten data.

Attacker's Knowledge. The semi-honest server, acting as the attacker, possesses legitimate data within FU. It can access two versions of the local model parameters and global model parameters, before and after unlearning.

4.2 Problem Formalization

Throughout this paper, we study the problem of label inference attacks against FU. We denote the forgotten dataset as $D_{\text{forgotten}} = \{(x_i, y_i) \mid i = 1, \ldots, M\}$, where x_i represents the input features, and $y_i \in \{0,1\}^C$ denotes the corresponding one-hot encoded class labels for C classes. Let the global model $\mathcal{F}(\cdot;\theta)$ be parameterized by θ, which maps the input space $\mathcal{X}$ to a C-dimensional output space $\mathbb{R}^C$. As a potential attacker, the server analyzes parameter differences between the two versions of both the local and the global models to infer the labels of the forgotten data, formally expressed as the following objective function:

$$\hat{y}_i = \arg\max_{y_i} \mathcal{L}\left(\mathcal{F}(x_i;\theta_{\text{global}}), y_i\right) \tag{3}$$

where $\hat{y}_i$ is the inferred label for the forgotten data point x_i, and $\mathcal{L}$ is the loss measuring the discrepancy between the model output and the predicted label.

5 `ULIA`: Label Inference Attack

In this section, we propose `ULIA`, a novel label inference attack that infer the labels of unlearning data across three FU levels: sample-level, class-level, and client-level, by analyzing the model differences in FU. Figure 1 illustrates the proposed label inference attack in the context of FU. The implementation of the `ULIA` attack can be carried out in the following four steps.

- *Analysis of parameter changes.* This step compares local and global model parameters before and after unlearning to quantify the influence of forgotten data, providing a foundation for the attack.

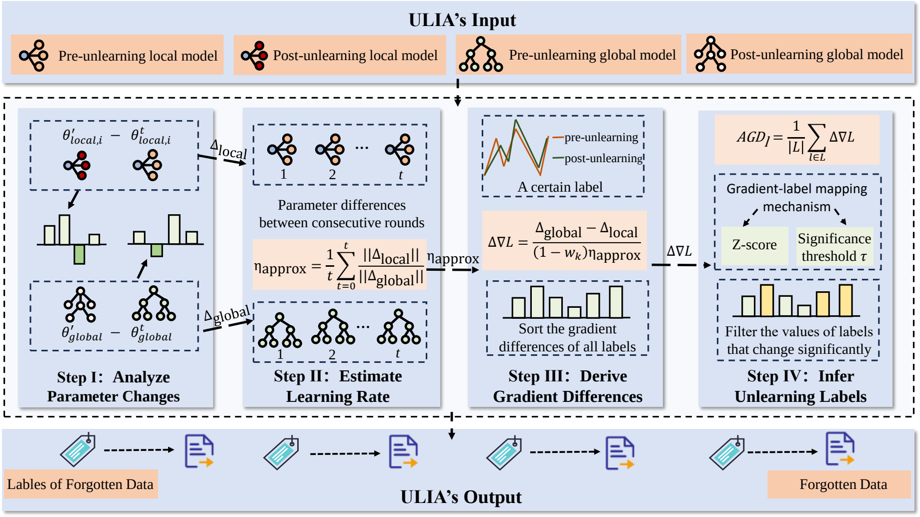

ULIA's Input
Pre-unlearning local model
Post-unlearning local model
Pre-unlearning global model
Post-unlearning global model
$\theta'_{local,i} - \theta^{t}_{local,i}$
$\theta'_{global} - \theta^{t}_{global}$
Step I: Analyze Parameter Changes
Δ_{local}
Δ_{global}
1
2
t
Parameter differences between consecutive rounds
$\eta_{approx} = \frac{1}{t}\sum_{t=0}^{t}\frac{||\Delta_{local}||}{||\Delta_{global}||}$
η_{approx}
Step II: Estimate Learning Rate
pre-unlearning
post-unlearning
A certain label
$\Delta\nabla L = \frac{\Delta_{global} - \Delta_{local}}{(1-w_k)\eta_{approx}}$
Sort the gradient differences of all labels
Step III: Derive Gradient Differences
$\Delta\nabla L$
$AGD_l = \frac{1}{|L|}\sum_{l\in L}\Delta\nabla L$
Gradient-label mapping mechanism
Z-score
Significance threshold τ
Filter the values of labels that change significantly
Step IV: Infer Unlearning Labels
Lables of Forgotten Data
ULIA's Output
Forgotten Data

Fig. 1: Overview of Unlearning Label Inference Attack. It shows the four main steps: (1) analyzing parameter changes, (2) estimating the learning rate, (3) deriving gradient differences, and (4) inferring the forgotten labels. The diagram highlights how ULIA utilizes pre- and post-unlearning models to infer labels.

- *Estimation of learning rate.* By analyzing parameter differences across multiple rounds, an effective learning rate is estimated to reconstruct gradients.
- *Derivation of gradient differences.* Using the estimated learning rate and parameter differences, approximate gradient differences are reconstructed.
- *Inferring the Labels of Forgotten Data.* By utilizing the gradient-label mapping mechanism, the degree of match between gradient changes and forgotten data labels is measured, enabling more accurate selection of labels.

5.1 Analysis of Parameter Changes

The first step in ULIA focuses on analyzing the changes in the model parameters caused by the unlearning process. By examining both the local and global model parameter changes, attackers can infer the influence of forgotten data on FL. Changes of local model parameters provide insights into how the removal of specific data affects individual clients, while changes of global model parameters reveal the aggregated impact of the unlearning operation across all clients.

Changes of local model parameters Δ_{local} quantify the variation in the local model parameters of the target client before and after unlearning. This change, which captures the impact of removing the influence of forgotten data.

$$\Delta_{\text{local}} = \theta'_{\text{local},i} - \theta^{(t)}_{\text{local},i} \tag{4}$$

where $\theta^{(t)}_{\text{local},i}$ represents the local model parameters before unlearning at the t-th communication round. $\theta'_{\text{local},i}$ denotes the updated local model parameters.

Similarly, the global parameter change Δ_{global}, obtained from the model parameters before and after unlearning, is defined as:

$$\Delta_{\text{global}} = \theta'_{\text{global}} - \theta^{(t)}_{\text{global}} \tag{5}$$

where $\theta^{(t)}_{\text{global}}$ indicates the global model parameters before unlearning. θ'_{global} refers to the updated global model parameters after unlearning.

The changes in the local model parameters Δ_{local} reflect the adjustments made by the client during the update process to remove the influence of the unlearning data, revealing the contribution of specific data to the client's local model. By analyzing the changes in the global model parameters Δ_{global} before and after unlearning, the extent of the influence of the unlearning data on the global model can be observed.

5.2 Estimation of Effective Learning Rate

The learning rate controls the step size of gradient updates in FL, directly affecting parameter changes. We estimate the learning rate by leveraging the differences between the local model parameters and the global model parameters from previous training rounds as given in Equation (6). This enables `ULIA` to conduct attacks with more limited knowledge, enhancing its adaptability.

$$\eta_{\text{approx}} = \frac{1}{t} \sum_{t=0}^{t} \frac{\|\Delta_{\text{local}}\|}{\|\Delta_{\text{global}}\|} \tag{6}$$

where $\|\cdot\|_2$ denotes the $L2$-norm, t is the number of rounds used for averaging.

The reasons for averaging over the previous t rounds are as follows. Firstly, performing the averaging process helps mitigate the impact of noise and outliers in individual updates, leading to a more robust and reliable estimate. Secondly, it preserves the integrity of the gradient direction, as the learning rate influences only the magnitude of the update, not the direction. This ensures that the estimated learning rate can be effectively used to derive gradient differences. By estimating an effective learning rate, attackers obtain the key parameters required to infer the gradients corresponding to the forgotten data.

5.3 Derivation of Gradient Differences

Attackers can infer the gradient updates associated with forgotten data by analyzing the difference between local and global parameter changes. This discrepancy provides insight into the contribution of the forgotten data to model optimization, forming a direct link between parameter variations and data characteristics. By leveraging an estimated learning rate, attackers can approximate the gradient difference introduced by the forgotten data.

In FL, the global model update at round $t+1$ follows the standard aggregation rule, as described in Equation (2). When a client requests unlearning, its local model parameters are updated without the forgotten data, deviating from their original update path. The parameter changes before and after unlearning are defined as shown in Equations (4) and (5). Since the global model aggregates updates from all clients, its change can be rewritten as:

$$\varDelta_{\text{global}} = \sum_{i=1}^{N} w_i \varDelta_{\text{local},i}. \tag{7}$$

Computing the difference between global and local parameter changes gives:

$$\varDelta_{\text{global}} - \varDelta_{\text{local}} = (w_k - 1)\varDelta_{\text{local}}. \tag{8}$$

which reveals that the discrepancy between local and global updates is directly proportional to the forgotten data's gradient contribution. Since the attackers do not directly observe $\varDelta \nabla L$, they approximate the discrepancy using the estimated learning rate η_{approx}, yielding:

$$\varDelta \nabla L = \frac{\varDelta_{\text{global}} - \varDelta_{\text{local}}}{(1 - w_k)\eta_{\text{approx}}}. \tag{9}$$

This equation provides a mechanism for reconstructing the gradient updates influenced by the forgotten data. Given that gradient variations capture feature importance, the attackers can analyze the magnitude and sparsity of $\varDelta \nabla L$ to infer the forgotten data's characteristics. By analyzing the approximated gradient differences, the attackers can identify which feature dimensions were most influenced by the forgotten data. Significant variations in specific gradient components indicate that these features were closely associated with the removed data. Furthermore, examining the concentration and distribution of gradient changes allows the attackers to assess the relative importance of individual features.

The difference between local and global parameter updates provides a strong signal of the forgotten data's impact on model training. As this discrepancy maintains a linear relationship with the forgotten data's gradient contribution, it serves as a crucial indicator for reconstructing its characteristics. These findings reveal inherent weaknesses in existing FU mechanisms, demonstrating that even after explicit data removal, residual traces may still persist in the global model, posing potential privacy risks.

5.4 Inferring the Labels of Forgotten Data

The final step of the proposed method focuses on inferring the labels of forgotten data by analyzing the derived gradient differences. The unlearning operation introduces significant changes to the gradients associated with the forgotten data, making them more prominent in the analysis. By linking these gradient changes to the weights in the model's output layer, attackers can accurately infer the labels of the forgotten data. For each label category $l \in L$, the average gradient difference is computed as:

$$\text{AGD}_l = \frac{1}{|L|} \sum_{l \in L} \varDelta \nabla L_l \tag{10}$$

where AGD_l represents the average gradient difference for a specific label category l, and L is the set of all label categories.

This step emphasizes the innovative integration of gradient difference reconstruction with targeted label analysis, illustrating how the method effectively translates parameter and gradient variations into actionable insights, thereby enabling precise identification of the labels of forgotten data.

However, when multiple label categories are forgotten by the client, the server does not know the exact number of forgotten categories. Therefore, it cannot simply assume that the category with the largest gradient change accounts for all the forgotten categories. To address this, we propose a *gradient-label mapping mechanism*, which establishes a one-to-one correspondence between gradient changes and the labels of forgotten data, and employs a dynamic filtering strategy to select labels that are more likely to correspond to the forgotten data. Specifically, this mechanism leverages two key methods: (1) It quantifies the changes induced by each label category during model parameter updates by utilizing the properties of the Z-score [3].

$$Z_l = \frac{\mathrm{AGD}_l - \mu_{\mathrm{AGD}}}{\sigma_{\mathrm{AGD}}} \tag{11}$$

where μ_{AGD} represents the average of gradient differences across all label categories, and σ_{AGD} represents the standard deviation of the gradient differences across all label categories. (2) It dynamically infers the number of forgotten data categories. After computing the Z-score for each label category, a predefined significance threshold τ is applied to filter a candidate set $L_{\mathrm{candidate}}$ as given in Equation (12), which includes the label categories most likely corresponding to the forgotten data of the client.

$$L_{\mathrm{candidate}} = \{l \mid Z_l > \tau\} \tag{12}$$

6 Experiments

6.1 Experimental Settings

All experiments are performed on a workstation equipped with an Intel(R) Core(TM) i7-13700K processor, 64GB of RAM, and three NVIDIA RTX 4090 GPU cards. `ULIA` is implemented using Python 3.8 and PyTorch 2.1.0.

Datasets and model. We select the following two datasets, which are widely used in FU. 1) **MNIST** [11], a handwritten digit classification dataset consisting of $60,000$ training images and $10,000$ test images. 2) **CIFAR-10** [10], a color image classification dataset containing $50,000$ training samples and $10,000$ test samples. We conduct experiments on both *IID* and *non-IID* data distributions, and perform training on the popular deep learning model **ResNet-18** [6].

Table 1: Attack performance of ULIA applied to typical unlearning methods. The attack is performed under the condition that the quantity of forgotten samples accounts for 10% of the client's total local data. $\mathcal{L}$ represents the number of categories of forgotten data labels.

Methods		**FedEraser**			**Rapid Retrain**			**SGA-EWC**		
Datasets	$\mathcal{L}$	**Sample**	**Class**	**Client**	**Sample**	**Class**	**Client**	**Sample**	**Class**	**Client**
The attacker is aware of the number of label categories										
MNIST	1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	2	0.970	1.000	1.000	0.970	1.000	1.000	0.940	1.000	1.000
	3	0.850	1.000	1.000	0.910	1.000	1.000	0.810	1.000	1.000
CIFAR	1	0.970	1.000	1.000	1.000	1.000	1.000	0.950	1.000	1.000
	2	0.890	1.000	1.000	0.920	1.000	1.000	0.860	1.000	1.000
	3	0.800	1.000	1.000	0.860	1.000	1.000	0.780	1.000	1.000
The attacker does not know the number of label categories										
MNIST	1	0.958	1.000	1.000	0.973	1.000	1.000	0.923	1.000	1.000
	2	0.898	1.000	1.000	0.925	1.000	1.000	0.855	1.000	1.000
	3	0.780	1.000	1.000	0.850	1.000	1.000	0.738	1.000	1.000
CIFAR	1	0.919	1.000	1.000	0.924	1.000	1.000	0.887	1.000	1.000
	2	0.837	1.000	1.000	0.872	1.000	1.000	0.785	1.000	1.000
	3	0.747	1.000	1.000	0.817	1.000	1.000	0.692	1.000	1.000

FU methods. To comprehensively evaluate ULIA, we apply it to the following three typical FU methods. 1) **FedEraser** [14], calibrates the historical updates of retained clients, enabling the server to efficiently reconstruct the global model. 2) **Rapid Retrain** [16], utilizes gradient and curvature information to identify more optimal descent directions, facilitating efficient retraining. 3) **SGA-EWC** [27], proposes an efficient FU framework using reverse Stochastic Gradient Ascent (SGA) and Elastic Weight Consolidation (EWC) to quickly adjust model parameters and eliminate the influence of specific data.

Details of parameter settings. Before the unlearning process, FU is performed for 100 rounds on 10 clients, using a Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.01, and a batch size of 64. Additionally, we set the significance threshold as $\tau = 2$ to more accurately infer the labels.

Evaluation metrics. Like previous works[7,33], we employ the widely-used Intersection over Union (IoU) method to evaluate the ASR of ULIA in inferring the labels of the forgotten data. We perform 100 attack tests and calculate the average ASR of ULIA. The ASR is calculated as follows:

$$\mathrm{ASR} = \frac{|L^i_{\mathrm{true}} \cap L^i_{\mathrm{pred}}|}{|L^i_{\mathrm{true}} \cup L^i_{\mathrm{pred}}|} \tag{13}$$

Table 2: The impact of the quantity of forgotten samples on the ASR of ULIA. 1%, 2%, and 5% represent the percentage of samples requested to be forgotten, relative to the client's total local data.

Methods		**FedEraser**			**Rapid Retrain**			**SGA-EWC**		
Datasets	$\mathcal{L}$	**1%**	**2%**	**5%**	**1%**	**2%**	**5%**	**1%**	**2%**	**5%**
The attacker is aware of the number of label categories										
MNIST	1	0.910	0.960	1.000	0.930	0.970	1.000	0.870	0.920	0.980
	2	0.820	0.850	0.900	0.850	0.890	0.920	0.760	0.810	0.880
	3	0.700	0.750	0.810	0.760	0.820	0.870	0.670	0.710	0.770
CIFAR	1	0.830	0.890	0.940	0.850	0.920	0.980	0.770	0.850	0.900
	2	0.760	0.810	0.850	0.780	0.840	0.880	0.690	0.750	0.810
	3	0.680	0.720	0.770	0.700	0.760	0.810	0.650	0.690	0.740
The attacker does not know the number of label categories										
MNIST	1	0.825	0.886	0.918	0.838	0.891	0.933	0.783	0.848	0.887
	2	0.783	0.833	0.872	0.827	0.868	0.907	0.697	0.745	0.803
	3	0.665	0.718	0.761	0.729	0.778	0.814	0.642	0.675	0.719
CIFAR	1	0.778	0.847	0.902	0.793	0.853	0.916	0.747	0.806	0.843
	2	0.713	0.773	0.813	0.748	0.798	0.843	0.658	0.693	0.732
	3	0.648	0.691	0.735	0.662	0.714	0.759	0.623	0.648	0.668

where $|L^i_{true} \cap L^i_{pred}|$ represents the size of the intersection between the true label set L^i_{true} and the predicted label set L^i_{pred} for the i-th attack test.

6.2 Attack Performance Evaluation

We set the number of categories of forgotten data labels to 1, 2, and 3, and perform attacks on three different FU methods at three levels, under two conditions: whether the attacker knows the number of forgotten data labels. Meanwhile, the quantity of forgotten samples is set to 10% of the target client's data. The attack performance of ULIA applied to typical unlearning methods on the MNIST and CIFAR-10 datasets is reported in Table 1.

ULIA achieves 100% ASR for both class-level and client-level unlearning, because the large volume of forgotten data causes significant shifts in the model's parameters, making the inference easier. The changes in model weights and gradients are directly correlated with the forgotten data. By analyzing the gradient differences between the global model and local models, ULIA effectively identifies the forgotten data. For sample-level unlearning, the ASR gradually decreases as the number of forgotten data labels increases. However, on the MNIST dataset, when the number of forgotten label categories is 1, the ASR reaches 95.8% on FedEraser, 97.3% on Rapid Retrain, and 92.3% on SGA-EWC. This indicates that when the number of forgotten label categories is small, our attack still achieves strong performance, approaching 100%. Even with 3 forgotten label categories, ULIA still achieves 73.8% ASR. Similarly, on the CIFAR-10 dataset,

Table 3: Impact of the Non-IID Data Distribution. The attack is performed under the condition that the quantity of forgotten samples accounts for 10% of the client's total local data.

Methods		**FedEraser**			**Rapid Retrain**			**SGA-EWC**		
Datasets	$\mathcal{L}$	**Sample**	**Class**	**Client**	**Sample**	**Class**	**Client**	**Sample**	**Class**	**Client**
MNIST	1	0.872	0.955	0.918	0.893	0.964	0.932	0.834	0.892	0.878
	2	0.828	0.914	0.858	0.852	0.934	0.872	0.775	0.848	0.812
	3	0.735	0.872	0.798	0.755	0.902	0.822	0.645	0.798	0.745
CIFAR	1	0.812	0.914	0.868	0.835	0.925	0.882	0.765	0.864	0.825
	2	0.714	0.885	0.808	0.742	0.908	0.824	0.692	0.805	0.778
	3	0.625	0.834	0.752	0.652	0.845	0.782	0.585	0.748	0.695

ULIA maintains 69.2% ASR under the same conditions. This difference is small compared to the ASR when attacker knows the number of forgotten data labels, indicating that ULIA is still able to effectively infer the labels even without complete information, demonstrating strong attack performance and adaptability.

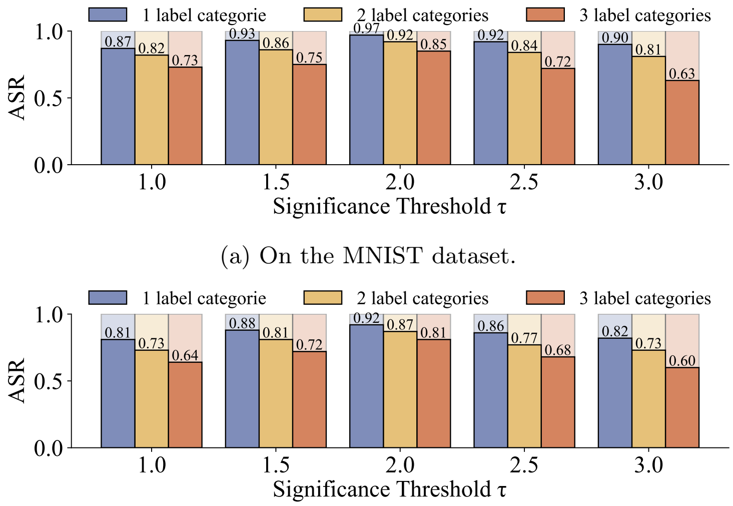
6.3 Sensitivity Evaluation

In this subsection, we study the impact of three key factors, the quantity of forgotten samples, the non-IID data distribution and the significance threshold.

Impact of the Quantity of Forgotten Samples. The quantity of forgotten samples affects the magnitude of model parameter changes before and after unlearning. Therefore, we set the quantity of forgotten samples to 1%, 2%, and 5% to evaluate ULIA. The impact of the quantity of forgotten samples on the ASR of ULIA is shown in Table 2.

The experimental results show that as the number of forgotten samples increases, the ASR of ULIA under different FU methods improves. Taking the MNIST dataset as an example, on FedEraser, when the number of forgotten label categories is 1 and the forgotten sample percentage is 1%, the ASR of ULIA is 82.50%, while when the forgotten sample percentage is 5%, the ASR of ULIA is 91.83%. This is because as the number of forgotten samples increases, the gradient differences corresponding to the labels become more pronounced. However, even with 3 forgotten label categories and only 1% of the quantity of forgotten samples, ULIA can still achieve the ASR of 64.17%. Therefore, even in situations where the quantity of forgotten samples is small and the number of forgotten label categories is large, ULIA is still capable of effectively handling and adapting to these conditions.

Impact of the Non-IID Data Distribution. The distribution of data across clients significantly influences the performance of ULIA. In non-IID settings, the skewed data distribution causes uneven parameter updates, complicating the inference process. To evaluate the impact of non-IID distribution on ULIA, we conduct experiments under varying levels of data heterogeneity. The effect on the ASR is shown in Table 3.



(a) On the MNIST dataset.

(b) On the CIFAR-10 dataset.

Fig. 2: The impact of different significance threshold on the ASR of ULIA.

In the non-IID data environment, the ASR of ULIA is generally lower than that in the IID setting. This is due to the heterogeneity of the non-IID data distribution, which leads to large differences in data characteristics across clients, affecting the global model update process and making it more difficult to infer forgotten data. Nevertheless, ULIA still maintains a certain attack performance in the non-IID environment. On the MNIST dataset, the ASR ranges from 64.5% to 96.4%, while on the CIFAR-10 dataset, it ranges from 58.5% to 92.5%, demonstrating its strong adaptability and considerable attack performance.

Impact of Different Significance Threshold τ. The significance threshold τ is used to filter candidate forgotten label categories, and it has a significant impact on the ASR. Therefore, we performed experiments with different values of τ, using ULIA applied in the Rapid Retrain method as an example. The experimental results on the MNIST dataset are shown in Figure 2(a), while the results on the CIFAR-10 dataset are shown in Figure 2(b).

The experimental results indicate that as τ increases, the ASR first increases and then decreases. Around $\tau = 2$, ULIA achieves the best attack performance, regardless of whether the number of forgotten label categories is 1, 2, or 3. Therefore, selecting an appropriate value of τ is crucial for the attack performance of ULIA. If τ is set too high, some smaller but still important gradient changes may be missed, thus affecting the attack effectiveness. Conversely, if τ is set too low, too many label categories may be incorrectly inferred as forgotten categories, which could compromise the accuracy of the attack.

7 Conclusion

In this paper, we have analyzed the label inference attacks against Federated Unlearning (FU). Our research has uncovered a significant privacy vulnerability within the FU framework. We have introduced ULIA, a novel label inference attack that can infer the labels of unlearning data at three FU levels: sample-level, class-level, and client-level, by examining the model variations caused by FU. Our experiments show that ULIA demonstrates outstanding attack performance and adaptability. In future work, we will explore defense strategies on FU that protect against the label inference attacks, contributing to the development of more robust privacy-preserving techniques in FL systems.

References

1. Arazzi, M., Conti, M., Koffas, S., Krcek, M., Nocera, A., Picek, S., Xu, J.: Blindsage: Label inference attacks against node-level vertical federated graph neural networks. arXiv preprint arXiv:2308.02465 (2023)
2. Che, T., Zhou, Y., Zhang, Z., Lyu, L., Liu, J., Yan, D., Dou, D., Huan, J.: Fast federated machine unlearning with nonlinear functional theory. In: International conference on machine learning. pp. 4241–4268. PMLR (2023)
3. Fei, N., Gao, Y., Lu, Z., Xiang, T.: Z-score normalization, hubness, and few-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 142–151 (2021)
4. Fu, C., Zhang, X., Ji, S., Chen, J., Wu, J., Guo, S., Zhou, J., Liu, A.X., Wang, T.: Label inference attacks against vertical federated learning. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 1397–1414 (2022)
5. Harding, E.L., Vanto, J.J., Clark, R., Hannah Ji, L., Ainsworth, S.C.: Understanding the scope and impact of the california consumer privacy act of 2018. Journal of Data Protection & Privacy **2**(3), 234–253 (2019)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y.: Acquisition of localization confidence for accurate object detection. In: Proceedings of the European conference on computer vision (ECCV). pp. 784–799 (2018)
8. Jiang, J.C., Kantarci, B., Oktug, S., Soyata, T.: Federated learning in smart city sensing: Challenges and opportunities. Sensors **20**(21), 6230 (2020)
9. Jiang, Y., Tong, X., Liu, Z., Ye, H., Tan, C.W., Lam, K.Y.: Efficient federated unlearning with adaptive differential privacy preservation. In: 2024 IEEE International Conference on Big Data (BigData). pp. 7822–7831. IEEE (2024)
10. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
11. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
12. Li, O., Sun, J., Yang, X., Gao, W., Zhang, H., Xie, J., Smith, V., Wang, C.: Label leakage and protection in two-party split learning. arXiv preprint arXiv:2102.08504 (2021)
13. Lin, Y., Gao, Z., Du, H., Ren, J., Xie, Z., Niyato, D.: Blockchain-enabled trustworthy federated unlearning. arXiv preprint arXiv:2401.15917 (2024)
14. Liu, G., Ma, X., Yang, Y., Wang, C., Liu, J.: Federaser: Enabling efficient client-level data removal from federated learning models. In: 2021 IEEE/ACM 29th international symposium on quality of service (IWQOS). pp. 1–10. IEEE (2021)

15. Liu, J., Lyu, X.: Clustering label inference attack against practical split learning. arXiv e-prints pp. arXiv–2203 (2022)
16. Liu, Y., Xu, L., Yuan, X., Wang, C., Li, B.: The right to be forgotten in federated learning: An efficient realization with rapid retraining. In: IEEE INFOCOM 2022-IEEE Conference on Computer Communications. pp. 1749–1758. IEEE (2022)
17. Liu, Z., Jiang, Y., Shen, J., Peng, M., Lam, K.Y., Yuan, X., Liu, X.: A survey on federated unlearning: Challenges, methods, and future directions. ACM Computing Surveys **57**(1), 1–38 (2024)
18. Long, G., Tan, Y., Jiang, J., Zhang, C.: Federated learning for open banking. In: Federated learning: privacy and incentive, pp. 240–254. Springer (2020)
19. Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G.: A survey on security and privacy of federated learning. Future Generation Computer Systems **115**, 619–640 (2021)
20. Qiu, P., Zhang, X., Ji, S., Du, T., Pu, Y., Zhou, J., Wang, T.: Your labels are selling you out: Relation leaks in vertical federated learning. IEEE Transactions on Dependable and Secure Computing **20**(5), 3653–3668 (2022)
21. Shaik, T., Tao, X., Xie, H., Li, L., Zhu, X., Li, Q.: Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy. IEEE Transactions on Neural Networks and Learning Systems (2024)
22. Sheng, X., Bao, W., Ge, L.: Robust federated unlearning. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. pp. 2034–2044 (2024)
23. Su, N., Li, B.: Asynchronous federated unlearning. In: IEEE INFOCOM 2023-IEEE Conference on Computer Communications. pp. 1–10. IEEE (2023)
24. Voigt, P., Von dem Bussche, A.: The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing **10**(3152676), 10–5555 (2017)
25. Wang, J., Guo, S., Xie, X., Qi, H.: Federated unlearning via class-discriminative pruning. In: Proceedings of the ACM Web Conference 2022. pp. 622–632 (2022)
26. Wang, Z., Gao, X., Wang, C., Cheng, P., Chen, J.: Efficient vertical federated unlearning via fast retraining. ACM Transactions on Internet Technology **24**(2), 1–22 (2024)
27. Wu, L., Guo, S., Wang, J., Hong, Z., Zhang, J., Ding, Y.: Federated unlearning: Guarantee the right of clients to forget. IEEE Network **36**(5), 129–135 (2022)
28. Yuan, W., Yin, H., Wu, F., Zhang, S., He, T., Wang, H.: Federated unlearning for on-device recommendation. In: Proceedings of the sixteenth ACM international conference on web search and data mining. pp. 393–401 (2023)
29. Zhang, L., Zhu, T., Zhang, H., Xiong, P., Zhou, W.: Fedrecovery: Differentially private machine unlearning for federated learning frameworks. IEEE Transactions on Information Forensics and Security (2023)
30. Zhang, X., Zhou, X., Chen, K.: Data leakage with label reconstruction in distributed learning environments. In: International Conference on Machine Learning for Cyber Security. pp. 185–197. Springer (2022)
31. Zhao, B., Mopuri, K.R., Bilen, H.: idlg: Improved deep leakage from gradients. arXiv preprint arXiv:2001.02610 (2020)
32. Zhao, Y., Wang, P., Qi, H., Huang, J., Wei, Z., Zhang, Q.: Federated unlearning with momentum degradation. IEEE Internet of Things Journal (2023)
33. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-iou loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 12993–13000 (2020)