Federated Unlearning Activated Backdoor Attacks

Jian Chen, *Member, IEEE*, Wenlong Shi, Chengyu Hu, *Member, IEEE*, Jianfeng Lu, *Member, IEEE*, Ahmed M. Abdelmoniem, *Senior Member, IEEE*, Chen Wang, *Senior Member, IEEE*

***Abstract*—Federated unlearning (FU) has recently emerged as a promising paradigm for removing client data from trained models, enabling privacy preservation and regulatory compliance in federated learning (FL) systems. However, existing FU methods primarily focus on the efficiency of data removal, leaving security vulnerabilities largely unexplored. In this paper, we propose FedUBA, a Federated Unlearning activated Backdoor Attack, which exploits the unlearning process itself to stealthily trigger backdoor behaviors. Unlike traditional backdoor attacks that embed backdoors via direct data poisoning during the learning process, FedUBA exploits the post-training unlearning process to covertly embed backdoor behaviors. The core of FedUBA lies in misleading the global model to unlearn more information associated with influential samples for the backdoored samples than intended, thereby adversely affecting the backdoored samples. To achieve this, FedUBA employs a principled three-stage framework, which involves generating stealthy backdoor triggers, selecting influential samples with the greatest impact on backdoored samples' predictions via black box sensitivity-based analysis, and crafting malicious unlearning requests to induce the global model into activating the backdoor behavior. By doing so, we can significantly alter predictions on backdoored samples by initiating malicious unlearning requests. Extensive experiments on five realistic datasets demonstrate that FedUBA effectively achieves an 80% attack success rate on backdoored samples by triggering only 0.5% malicious unlearning requests.**

***Index Terms*—Federated unlearning, federated learning, backdoor attack, sensitivity sample.**

I. Introduction

Recently, federated unlearning (FU) [1]–[3] has gained increasing attention as an effective mechanism for eliminating client data from trained models, thereby enhancing user data protection and ensuring compliance with privacy regulations such as the General Data Protection Regulation (GDPR) [4] and the California Consumer Privacy Act (CCPA) [5] in federated learning (FL) systems. Its widespread adoption has led to significant achievements in various domains, including recommendation system [6], natural language processing [7], and computer vision [8]. However, existing studies [9], [10] on FU primarily focus on improving unlearning efficiency, making the security vulnerabilities largely unexplored. This lack of attention exposes a critical blind spot: *if the unlearning process itself is manipulable, could it unintentionally be used for model manipulation?*

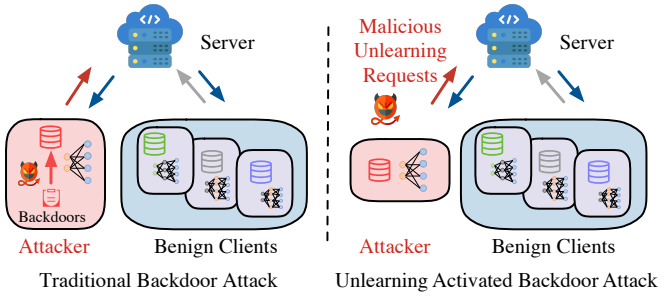


Fig. 1: Differences between traditional backdoor attacks and unlearning-activated backdoor attacks in FL. In traditional backdoor attacks, the attacker embeds backdoor triggers into the training data during the training phase. In contrast, our attack activates backdoor behaviors by crafting malicious unlearning requests that occur after the model has been trained.

Emerging research has begun to explore the vulnerabilities of FU mechanisms. For instance, a very recent study [11] has highlighted that FU is vulnerable to malicious unlearning attacks, where the attacker attempts to trigger a malicious unlearning request to misclassify the targeted data. However, this type of attack suffers from two primary limitations. First, it heavily relies on precise knowledge of the targeted data, which may not always be feasible in real-world FL systems due to data heterogeneity and privacy constraints. Second, the attack scope remains limited to targeted misclassification rather than expanding the scope of potential attacks by exploiting system vulnerabilities that could compromise the integrity of the global model [12].

To step forward, in this paper we propose a novel backdoor attack method, dubbed FedUBA, which leverages the FU process to activate camouflaged backdoors in a stealthy and systematic manner. Unlike traditional backdoor attacks, which embed backdoors by poisoning the training data or directly modifying the model parameters during training, FedUBA strategically manipulates the unlearning process post-training to subtly adjust the model's internal representations (c.f. Fig. 1). To achieve this, our design is driven by the key insight that if a trained model retains latent information about backdoor triggers associated with the targeted class,

This work was supported in part by the National Natural Science Foundation of China under Grants 62502477, 62272183 and 62372343; by the Scientific Research Funds at China University of Geosciences (Wuhan) under Grant 2025022; by the Postdoctoral Fellowship Program of CPSF under Grant Number GZC20250162; by the Key R&D Program of Hubei Province under Grants 2025EHA033; and by the UKRI EPSRC Grant EP/X035085/1. *(Corresponding author: Chen Wang.)*

J. Chen and C. Hu are with the Department of Computer Science, China University of Geosciences (Wuhan), China. Email: {jianchen, huchengyu}@cug.edu.cn

W. Shi and C. Wang are with the Hubei Key Laboratory of Internet of Intelligence, School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. Email: {wenlongshi, chenwang}@hust.edu.cn.

J. Lu is with the School of Computer Science and Technology, Wuhan University of Science and Technology, 430065 Wuhan, China. Email: lujianfeng@wust.edu.cn.

A. M. Abdelmoniem is with School of Electronic Engineering and Computer Science, Queen Mary University of London, UK. He is also with Assiut University, Egypt. Email: ahmed.sayed@qmul.ac.uk.

then strategically manipulating the unlearning process can amplify these hidden triggers, inducing the backdoor samples to be classified into the attacker-desired class. Thus, the core idea of FedUBA is to craft malicious unlearning requests that selectively erase the features of influential samples and subtly manipulate the model's decision boundaries, embedding backdoor behaviors into the global model through unlearning.

Given the above idea, however, backdooring FL is rather challenging with unlearning. (1) Unlike traditional machine learning, where attackers can optimize stealthy backdoor triggers against a unified data distribution, generating such triggers in FL is far more challenging due to data heterogeneity and the limited knowledge of clients' training data. (2) Moreover, randomly unlearning samples is ineffective, as the aggregation rules in FL significantly dilute the impact of unlearning individual samples. This aggregation process, which combines updates from multiple clients, often reduces the effectiveness of these unlearning actions, making it difficult for attackers to significantly influence the global model through this approach.

To address the aforementioned challenges, we introduce a three-step framework. First, we propose *Stealthy Trigger Generation (STG)*, which optimizes the backdoor trigger to closely align with the feature center of the attacker-desired class, ensuring stealthiness and effectiveness even with limited knowledge of the clients' data distributions. Next, *Sensitivity Sample Selection (SSS)* efficiently identifies the most influential samples for a backdoored sample by leveraging sensitivity analysis to quantify the local decision-boundary sensitivity of each sample with respect to backdoor perturbations, serving as a black-box proxy for their influence on backdoored predictions. Finally, we design *Backdoor Effect Activation (BEA)* to craft malicious unlearning requests by pushing the features of unlearned influential samples closer to that of the backdoored samples. These manipulations embed backdoor behavior during unlearning, ultimately causing misclassification on backdoored samples.

We summarize our major contributions as follows:

- We introduce FedUBA as one of the first studies to systematically investigate FU-activated backdoor attacks, revealing the potential security risks of intentionally manipulating the global model's predictive behavior toward backdoored samples in FL systems.
- We propose a novel three-step attacking method, which enables a client to strategically launch FedUBA with only black-box access to the global model, making the attack practical in FU scenarios.
- We theoretically prove that FU can be adversarially manipulated by selectively unlearning sensitivity samples to induce a backdoored change in the model behavior.
- We empirically evaluate FedUBA on multiple datasets, with various FU methods and aggregation rules. The results demonstrate that existing FU frameworks lack robustness against FedUBA. The code of FedUBA has been released for reproducibility purposes[1].

[1]https://github.com/ity207/FedUBA

II. Related Works

A. Federated Unlearning

Generally, existing FU methods can be categorized into two main types [13]: historical update-based methods and parameter manipulation-based methods. The former approach retains previous model updates for future use in unlearning specific data. For instance, FedEraser [14], the earliest FU technique, effectively eliminate the influence of a client's data on the global FL model by retrieving the global model state before a client joins the federation. Building on this foundation, Su et al. [15] propose KNOT, a clustered aggregation mechanism specifically designed for asynchronous FL. Additionally, Ameen et al. [16] leverage the knowledge of a temporary model to reconstruct the unlearned global model, further enhancing the efficiency of the unlearning process.

Another line aims to modify model parameters during training to mitigate the influence of the targeted data. For example, Halimi et al. [17] reverse the learning process for the target client by constraining updates within an l_2-norm ball around a reference model, which is then fine-tuned by the remaining clients. Similarly, MoDE [18] employs a randomly initialized degradation model to facilitate unlearning, while Wang et al. [19] focus on classification tasks by pruning class-related channel parameters in deep learning models to achieve class-targeted unlearning. Recent studies further suggest that FU can be examined not only from the perspective of unlearning utility and efficiency, but also from that of robustness, attack resilience, and privacy leakage. For example, Sheng et al. [20] investigate robust FU under unreliable conditions, Wang et al. [21] discuss poisoning attacks and defenses against FU, and Zhang et al. [22] show that gradient differences may still enable data reconstruction after unlearning. Unlike existing FU techniques, which primarily aim to effectively erase client data by initiating benign unlearning requests, our work focuses on exposing the inherent security vulnerabilities within the unlearning process.

B. Backdoor Attacks in FL

Backdoor attacks [23]–[25] compromise the integrity of FL models by embedding attacker-desired hidden triggers during the training process. These attacks can be broadly categorized into fixed trigger attacks and trigger-optimization attacks. In fixed trigger attacks, the attacker pre-selects a static backdoor trigger without leveraging information from the FL training process. Since fixed triggers may not always be effective for backdoor injection, attackers often enhance their effectiveness through additional techniques, such as manually manipulating poisoned updates. For instance, DBA [26] divides the backdoor trigger into multiple sub-triggers for poisoning, making the attack more stealthy and harder to detect by defenses. Neurotoxin [27] only attacks less frequently updated model parameters to ensure the backdoor remains intact and is not easily erased during subsequent training.

Trigger-optimization attacks, on the other hand, aim to enhance the effectiveness of backdoor triggers by dynamically optimizing them during the training process. For instance, Zhang et al. [28] introduce a technique to maximize the

difference between the latent representations of clean and trigger-stamped samples. Shen et al. [29] propose to optimize the trigger that does not require any additional task label information in the vertical FL setting. While existing backdoor attacks in FL primarily focus on embedding and maintaining hidden triggers, our unlearning-activated paradigm is more practical, as it leverages existing data unlearning mechanisms and enables stealthy activation without modifying the model.

C. Attacks with Unlearning

Existing attacks occur during the unlearning process have primarily been studied in the context of centralized machine learning models. Di et al. [30] first conduct such attacks by injecting both poisoned and camouflage sets into the training dataset. An unlearning request is then triggered to remove the camouflage dataset, activating the poison effect within the model. Qian et al. [31] further expose the vulnerability of deep neural networks during the unlearning process by crafting malicious unlearning requests that identify features capable of causing targeted misclassification. Zhao et al. [32] explore static and sequential malicious attacks in the context of selective forgetting, formulating them as a stochastic optimal control problem to maximize the impact of malicious actions. Furthermore, Chen et al. [33] activate malicious attacks in the regression scenario and Hu et al. [34] craft malicious unlearning requests in the context of machine learning as a service (MLaaS).

Building on this, UBA-Inf [35] focuses on centralized unlearning and uses influence-driven camouflage to activate backdoors, whereas our attack targets FU, where data heterogeneity, client isolation, and aggregation make attack design fundamentally more challenging and motivate our sensitivity-guided selection with a low-budget setting. Moreover, FedMUA [11] studies malicious unlearning for targeted misclassification in FL, while FedUBA aims at persistent backdoor activation under FU. Although both involve selective sample manipulation, they differ in attack objectives, threat models, and technical design. In particular, FedMUA induces target-specific errors via influential samples, whereas our method jointly leverages sensitivity-guided selection and backdoor activation to amplify a trigger-aligned decision direction under a low-budget, black-box setting. Therefore, our approach represents a distinct FU backdoor attack formulation rather than a direct extension of prior work.

Compared to prior attempts to exploit unlearning for backdoor attacks in FL such as FUBA [36] and BadFU [37], our FedUBA differs in several fundamental aspects. First, existing methods rely on training-phase manipulation (e.g., camouflage samples or coordinated multi-adversary strategies) to implant latent backdoor behaviors that are later activated during unlearning. In contrast, FedUBA does not depend on such training-stage assumptions; instead, it directly exploits the unlearning process itself by removing influential samples, thereby activating the backdoor effect. Second, FedUBA introduces a sensitivity-driven mechanism to identify the critical samples that have the greatest impact on backdoored predictions. This enables a fundamentally different attack strategy from prior approaches that mainly depend on data construction or adversarial coordination. Third, FedUBA operates under a more practical and constrained setting, requiring only a single malicious client with black-box access to the global model and a very small fraction of malicious unlearning requests without access to gradients, intermediate representations, model parameters, or other clients' private data.

D. Backdoor Defenses in FL

Representative backdoor defenses in FL mainly aim to identify malicious client updates or suppress suspicious model behaviors during aggregation. For example, the direction alignment inspection [38] detects backdoor updates by examining the directional consistency of client gradients, FLAME [39] constrains malicious updates via clustering and noise-based model sanitization, and the multi-metrics-based method [40] combines multiple statistical indicators to adaptively identify anomalous clients. More recently, the individual unlearning-based detection [41] attempts to identify backdoored models by tracing model behavior changes through per-client unlearning. It can be seen that most existing FL backdoor defenses focus on identifying abnormal training-time updates, poisoned client gradients, or suspicious aggregation behavior. In contrast, our FedUBA activates the backdoor through malicious unlearning requests after model training, which makes the attack signal weaker and shifts it from the training stage to the unlearning stage. As a result, these defenses may have limited visibility into FedUBA, while defenses based on model sanitization or client-level statistics may not effectively capture the structured decision-boundary deformation induced during FU. This motivates the need for FU-specific defenses beyond conventional FL backdoor detection.

III. Threat Model

In this section, we formally define our threat model, and outline the attacker's goals, knowledge and capabilities.

A. Attacker's Goal

In the FL setting, clients collaboratively train a global model on the server. However, one or more untrusted clients may act as potential attackers, aiming to initiate malicious unlearning requests to misclassify trigger-containing data. Unlike previous backdoor attacks in FL, which aim to maintain the backdoor's availability at all times, the goal of our FedUBA is to activate the backdoor at the right moment while remaining undetectable. FedUBA seeks to minimize the attack success rate (ASR) to enhance stealth before launching the attack. Once the backdoor is activated, FedUBA maintains the same standards for ASR and accuracy as conventional backdoor attacks. More specifically, we can define the attacker's goal as follows:

- **Goal I: backdoor effectiveness goal.** This goal refers to the intentional misclassification behavior on the backdoored data by the unlearned global model during the testing phase. It aims to ensure that the unlearned global model can predict the attacker-desired prediction for the backdoored data.

- **Goal II: model utility goal.** This goal aims to maintain the prediction performance of the malicious unlearned global model on non-backdoored data. It is intended that the malicious unlearned global model generate predictions that are consistent with those of the local client model trained without any unlearning requests for non-backdoored data. This goal helps preserve the stealthiness of the attack to avoid detection.

B. Attacker's Knowledge

In our scenario, the attacker is assumed to have the knowledge of the general application domain of the victim task and to choose an attacker-desired semantic class. Such knowledge is realistic in many practical deployments because the task semantics are usually exposed by the service itself. For example, in a face-recognition service, the attacker may infer the target identity or identity category from public-facing usage context and collect publicly available photos of that target. Also, in traffic-sign classification, the attacker may know the target sign category from the task definition and obtain related images from public datasets or Internet resources; and in medical-image analysis, disease categories and visually related benchmark samples are often accessible from open repositories.

In addition, the attacker is assumed to be able to obtain a small number of representative target-class examples and auxiliary surrogate data that are semantically related to the task but strictly separated from the victim's training set. Importantly, these assumptions do not imply the access to the victim model's internals or private training data; they only reflect realistic semantic prior knowledge and external data sources that can be collected from public resources or application context. Therefore, FedUBA remains a black-box attack, even though it allows realistic task-level side information and external surrogate data.

We also notice that the assumption of collecting semantically related surrogate data is realistic for many practical FL deployments where public benchmark datasets, web resources, or domain-related repositories are available. However, for highly specialized domains such as private medical imaging or proprietary enterprise tasks, obtaining closely aligned surrogate data may be more challenging. In such cases, the transferability of the generated trigger may decrease, which could reduce the effectiveness of FedUBA.

C. Attacker's Capability

Generally, the attacker's capability is constrained by capping the number of malicious unlearning requests, including each client's budgets and the ownership-based access control. The attacker also operates in a strictly black-box manner and can selectively request the unlearning of locally samples, but cannot inject poisoned data, modify labels, or access other clients' data, gradients and intermediate model parameters. Moreover, the impact on predictions for backdoored data is bounded by the number of attackers, whereas the predictions on non-backdoored data remain largely unaffected.

IV. Problem Formulation

We begin by selecting y_d as the attacker-desired class. A malicious client C_m may act as an attacker with its training data, denoted as $\mathcal{D}_m = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{m}$, along with the local model $\mathcal{M}_m(., \boldsymbol{w}_m)$. The attacker first generates backdoor triggers $\boldsymbol{\delta}$ to better align with y_d. Next, the attacker identifies p high sensitivity training samples, represented as $\mathcal{D}_\text{s} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{p}$ that are relevant to the backdoored data. The attacker then applies a malicious unlearning modification on each $\boldsymbol{x}_j$ in $\mathcal{D}_\text{s}$, denoted as $\boldsymbol{x}_j - \boldsymbol{\delta}$. The malicious client then updates the local model $\mathcal{M}_m(., \boldsymbol{w}_m^{'})$ via the modified $\mathcal{D}_\text{s}$ and subsequently sends it to the server for further unlearning. Specifically, FedUBA performs the following two steps:

- **Step I.** In each round of FL training process, each client begins by downloading the global model $\mathcal{M}_G(., \boldsymbol{w})$ from the server and then independently trains its local model $\mathcal{M}_i(., \boldsymbol{w}_i)$ using its own local training data $\mathcal{D}_i$. The server then gathers updates from all clients and aggregates these updates to obtain the global model. The aggregated model $\mathcal{M}_{i+1}(., \boldsymbol{w}_{i+1})$ can be used as the global model for the next training round.
- **Step II.** Client C_m first generates stealthy trigger $\boldsymbol{\delta}$ and identifies p high sensitivity training samples. At a certain time, C_m sends data removal requests $\boldsymbol{x}_j - \boldsymbol{\delta}$ for each of these p sample. The server then aims to remove the influence of these requests from $\mathcal{M}_G(., \boldsymbol{w})$. In the FU process, client i could send the aggregated model $\mathcal{M}_i(., \boldsymbol{w}_i^{'})$ trained without $\mathcal{D}_f$ to the server. The server can then employ FU algorithms [14], [15] to update $\mathcal{M}_G(., \boldsymbol{w})$ without retraining.

The goal of FedUBA is to misclassify the backdoored data and maintain the prediction performance on non-backdoored data after unlearning, which can be formalized as:

$$\boldsymbol{\delta}^* = \arg\min_{\boldsymbol{\delta} \in \boldsymbol{\Delta}} \sum_{(\boldsymbol{x}, y_d) \in D_m} \mathcal{L}\left(\mathcal{M}_G\left(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{w}\right), y_d\right), \quad (1)$$

$$\mathcal{M}_G(\boldsymbol{x}_i - \boldsymbol{\delta}^*, \boldsymbol{w}^u) = y_d, \quad (2)$$

$$\mathcal{M}_G(., \boldsymbol{w}^u) = U(\mathcal{M}_m(., \boldsymbol{w}_m^{'}), \sum_{i=1, i \neq m}^{K} \mathcal{M}_i(., \boldsymbol{w}_i)), \quad (3)$$

where $U(\cdot)$ denotes the unlearning algorithm, $\mathcal{M}_m(., \boldsymbol{w}_m^{'})$ is the unlearned local model, $\boldsymbol{\delta}^*$ is the optimized backdoor trigger, and $\mathcal{M}_G(., \boldsymbol{w}^u)$ represents the unlearned global model. Note that all the notations used in this paper are summarized in Table I.

V. Attack Methodology

Recall that the goal of FedUBA is to deliberately manipulate the FU process, leading the unlearned model to produce attacker-desired predictions on backdoored data. To achieve this, FedUBA mainly consists of three steps (c.f. Fig. 2):

Step I. Stealthy Trigger Generation: FedUBA first leverages surrogate samples and the attacker-desired class samples to train a surrogate model, and then generates a trigger that is semantically aligned with the attacker-desired class by solving the trigger-generation objective in Eq. 5.

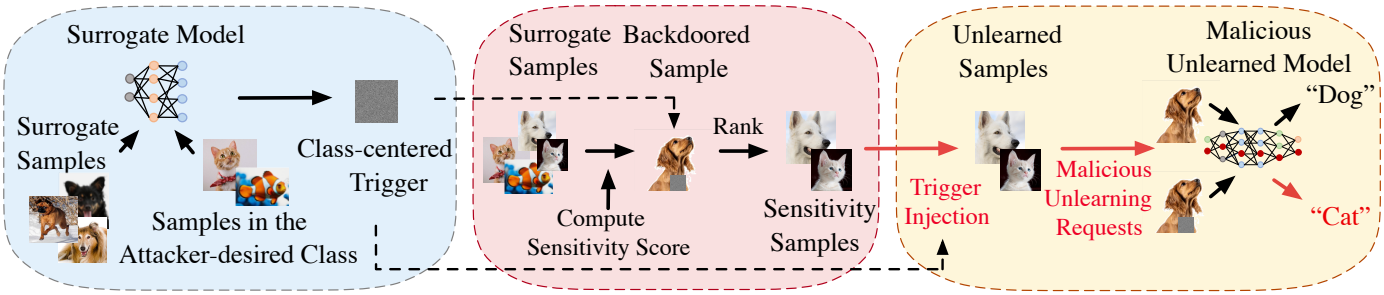


Fig. 2: The attack flow of FedUBA contains three steps: stealthy trigger generation, sensitivity sample selection, and backdoor effect activation, which systematically designs stealthy triggers, selects high sensitivity samples on the backdoored data, and crafts unlearning requests to induce unintended backdoor activation in the global model.

Step II. Sensitivity Sample Selection: A small subset of training samples in the malicious client with significant influence on the model's prediction of the backdoored data is identified through sensitivity analysis. Specifically, FedUBA computes the sensitivity score of each local sample by measuring the logit deviation caused by the backdoored noise. Samples with larger sensitivity scores are selected as the critical subset for enhancing the backdoor effect during unlearning.

Step III. Backdoor Effect Activation: Based on the selected sensitivity samples, malicious unlearning requests are constructed by perturbing the feature representations of these samples to align with those of the backdoored sample. This feature manipulation shifts the model's representations, ultimately resulting in the desired misclassification of the backdoored sample in the unlearned model.

It is worth emphasizing that the motivation of the above design differs from simply reusing a camouflaged backdoor pipeline from prior work. In FUBA [36] and BadFU [37], the attacker primarily relies on stronger training-phase preparation or adversarial optimization to implant latent malicious behavior before unlearning is invoked. In contrast, our goal is to make the attack feasible under a stricter FU setting with only black-box access and limited malicious unlearning budget. Under this constraint, STG is introduced to generate a transferable trigger from limited surrogate knowledge, SSS is needed because direct influence estimation is inaccessible, and BEA is designed to convert the selected critical samples into effective unlearning-time manipulation.

TABLE I: Summary of Notations

Notation	Description
$\mathcal{D}_m$	Local training data on the malicious client
$\mathcal{D}_s$	Sensitivity samples for backdoored samples
$\mathcal{D}_t$	Samples with attacker-desired label
y_d	Attacker-desired label
$\mathcal{M}_G(\cdot, \boldsymbol{w})$	Global model with weight $\boldsymbol{w}$
$\mathcal{M}_m(\cdot, \boldsymbol{w}_m)$	Malicious client's local model with weight $\boldsymbol{w}_m$
$\mathcal{M}_G(., \boldsymbol{w}^u)$	Global model after unlearning
$\boldsymbol{\delta}$	Backdoor trigger perturbation
$\boldsymbol{\delta}^*$	Optimized trigger
ϵ	Perturbation bound for backdoor trigger
$\mathcal{L}(\cdot)$	Loss function
$f_{\boldsymbol{\theta}_v}$	The victim model
$f_{\boldsymbol{\theta}_{sur}}$	The surrogate model
Δ	Feasible perturbation set for backdoor trigger
$U(\cdot)$	FU algorithm

A. *Stealthy Trigger Generation*

A key limitation of existing backdoor attacks in FL lies in the arbitrary design of triggers, which often leads to misalignment between the trigger and the attacker-desired class. To enhance the effectiveness of our attack, we aim to generate a trigger that closely aligns with the semantics of the targeted class and serves as its representative feature. To this end, we formulate an optimization problem to guide the trigger generation process.

We begin by assuming the access to a victim model trained on the original dataset, denoted by $f_{\boldsymbol{\theta}_v}$. Given this model and a set of target-class data $\mathcal{D}_t$, our goal is to learn a trigger $\boldsymbol{\delta}$ that maximizes the confidence with which the backdoored inputs are classified into the targeted class. This leads to the following optimization objective:

$$\boldsymbol{\delta}^* = \arg\min_{\boldsymbol{\delta}\in\boldsymbol{\Delta}} \sum_{(\boldsymbol{x},t)\in\mathcal{D}_t} \mathcal{L}\left(f_{\boldsymbol{\theta}_v}\left(\boldsymbol{x}+\boldsymbol{\delta}\right), y_d\right), \tag{4}$$

where $\boldsymbol{\Delta}$ denotes the set of possible trigger patterns. $\mathcal{L}\left(f_{\boldsymbol{\theta}_v}\left(\boldsymbol{x}+\boldsymbol{\delta}\right), y_d\right)$ represents the loss of predicting $\boldsymbol{x}+\boldsymbol{\delta}$ into the attacker-desired class y_d.

However, access to the victim model is unrealistic in practical attack settings. To overcome this, we draw inspiration from black-box attack techniques [42] and introduce a surrogate model, trained using available data from both surrogate examples and the targeted class. To improve the generalization of the surrogate model, we adopt a two-stage training strategy. First, we pre-train the model on surrogate data to extract robust low-level features. Then, we fine-tune it on the target-class samples for a few epochs to guide the model toward recognizing discriminative patterns specific to the targeted class. With this surrogate model in place, the optimization for trigger generation becomes:

$$\boldsymbol{\delta}^* = \arg\min_{\boldsymbol{\delta}\in\boldsymbol{\Delta}} \sum_{(\boldsymbol{x},y_d)\in D_t} \mathcal{L}\left(f_{\boldsymbol{\theta}_{sur}}\left(\boldsymbol{x}+\boldsymbol{\delta}\right), y_d\right). \tag{5}$$

To solve the above optimization problem, we adopt mini-batch stochastic gradient descent strategy. Specifically, a mini-batch of samples is drawn from the target-class training data in each iteration. We compute the gradient of the objective function, and then average these gradients across the batch. The trigger is subsequently updated using the averaged gradient and projected back onto the set $\mathbf{\Delta}$. Note that $\mathbf{\Delta}$ can be regarded as an l_∞-norm ball (i.e., $\mathbf{\Delta} = \boldsymbol{\delta} : |\boldsymbol{\delta}|\infty \leq \epsilon$). Also, $\boldsymbol{\delta}$ is constrained to the range $[-\epsilon, +\epsilon]$.

B. Sensitivity Sample Selection

Given the optimized trigger, the next step is to craft malicious unlearning requests. Here we have to emphasize that randomly altering training samples is ineffective, as the aggregation mechanism in FL tends to dilute local updates, and may even counteract the desired attack effect. Therefore, identifying the most influential samples that exert the greatest positive impact on the model's predictions for the backdoored inputs is essential.

Prior works rely on influence function-based analysis [43], which requires gradient or even Hessian information. This requirement is unrealistic in typical FL settings. To mitigate this constraint, we propose a fully black-box mechanism that identifies critical samples via forward inference alone. The insight here is that backdoor features tend to induce sharp local decision boundaries, and samples residing near these boundaries possess a distinctive property: even a tiny perturbation to the input elicits a disproportionately large shift in the model's output. This local sensitivity naturally indicates which samples are aligned with the backdoor direction. Thus, selecting high-sensitivity samples acts as an effective surrogate for gradient-based influence estimation, facilitating malicious unlearning without requiring any model parameters.

Specifically, given a local sample x_i and the victim model $f_{\boldsymbol{\theta}_v}$, the attacker first queries the victim model to obtain the original logits $f_{\boldsymbol{\theta}_v}(x_i)$. A small backdoored perturbation $\boldsymbol{\delta}^*$ is then added to x_i, producing the perturbed output $f(x_i + \boldsymbol{\delta}^*)$. We define the sensitivity score of x_i as the logit deviation:

$$s_i = \left\| f(x_i + \boldsymbol{\delta}^*) - f(x_i) \right\|. \tag{6}$$

The attacker computes s_i for all samples in the local dataset, ranks them in descending order, and selects the top-n samples:

$$\mathcal{D}_\text{s} = \text{Top-}n\{s_i \mid x_i \in \mathcal{D}\}. \tag{7}$$

By now, we can obtain the high-sensitivity sample set $\mathcal{D}_\text{s}$ which represents the most critical samples for manipulating the predictions of backdoored inputs. Note that our black-box method for $\mathcal{D}_\text{s}$ identification requires neither gradients nor the parameter access but only two forward passes per sample.

C. Backdoor Effect Activation

Building upon the identification of $\mathcal{D}_\text{s}$, the subsequent step involves formulating malicious unlearning requests for these samples to activate the backdoor effect. To achieve stealthiness, we propose a feature-level manipulation strategy that modifies the features of influential samples rather than their labels.

Algorithm 1 FedUBA Algorithm

Input: Malicious client $\mathcal{M}_m(., \boldsymbol{w}_m)$ with its learning algorithm $\mathcal{L}_m$ and training data $\mathcal{D}_m$; D_t is the attacker-desired class data; $\mathcal{I}$ is the total iteration number; α is the step size.

for each iteration $i \in (1, \mathcal{I} - 1)$ **do**
  $\boldsymbol{\delta}_{i+1} \leftarrow \boldsymbol{\delta}_i - \alpha \sum_{(\boldsymbol{x}_i, t) \in D_t} \nabla_{\boldsymbol{\delta}} \mathcal{L}\left(f_{\boldsymbol{\theta}_\text{sur}}(x_i + \boldsymbol{\delta}_i), t\right)$
  $\boldsymbol{\delta}^* \leftarrow \boldsymbol{\delta}_{i+1}$;
  Constraint: $\|\boldsymbol{\delta}_j\| \leq \epsilon$
end for
Return: $\boldsymbol{\delta}^*$
Calculate sensitivity score for each sample in $\mathcal{D}_n \leftarrow Eq.\ 6$;
Select s samples: $\mathcal{D}_\text{s} \leftarrow Eq.\ 7$;
for $j = 1; j <= p; j++$ **do**
  Update unlearned sample: $\boldsymbol{x}'_j \leftarrow Eq.\ 8$
end for
Malicious local model unlearn $\{\boldsymbol{x}'_j\}_{j=1}^p$: $\mathcal{M}_m(., \boldsymbol{w}'_m)$
Output: Updated malicious local model: $\mathcal{M}_m(., \boldsymbol{w}'_m)$

The core idea is to gradually move the influential samples closer to the target backdoored input in feature space. This targeted movement subtly manipulates the decision boundary, inducing the model to misclassify the backdoored input in favor of the attacker-desired label. By embedding the manipulation within unlearning requests, the attacker disguises their data while achieving model behavior manipulation.

In practice, the attacker modifies each influential sample $\boldsymbol{x}_j \in \mathcal{D}_\text{s}$ by applying a perturbation $\boldsymbol{\delta}^*$, such that the modified sample $\boldsymbol{x}'_j$ becomes:

$$\boldsymbol{x}'_j = \boldsymbol{x}_j - \boldsymbol{\delta}^*. \tag{8}$$

Also, the magnitude of the perturbation should be bounded, maintaining the stealthiness of the unlearning request (i.e., $\|\boldsymbol{\delta}^*\| \leq \epsilon$). It is worth emphasizing that the choice of ϵ significantly influences the effectiveness of the attack. A larger value of ϵ allows for more aggressive perturbations, thereby enhancing the ability to activate the backdoor effect. However, this benefit comes with a trade off as excessive perturbation may lead to a decline in model utility. Therefore, balancing attack strength and model utility is a critical consideration when tuning this parameter. The complete procedure of the proposed FedUBA framework is summarized in Algorithm 1.

VI. Theoretical Analysis

A. Unlearning-Induced Decision Boundary Deformation

Most existing FU studies implicitly assume that unlearning operations uniformly remove the influence of specified training samples while preserving the decision behavior on the remaining data. In contrast, we show that FU induces structured and sample-dependent decision boundary deformation, rather than information removal. This deformation constitutes the fundamental mechanism that enables adversarial exploitation.

Let $f_{\boldsymbol{\theta}}(\cdot)$ denote a trained global model with parameters $\boldsymbol{\theta}$. Consider a backdoor trigger $\boldsymbol{\delta}$ applied to an input $\boldsymbol{x}$, yielding

a backdoored input $\boldsymbol{x}+\boldsymbol{\delta}$. We define the backdoor-sensitive region as:

$$\mathcal{R}_\delta = \left\{ \boldsymbol{x} \;\middle|\; \left|\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x})^\top \boldsymbol{\delta}\right| > \frac{1}{|\mathcal{D}|}\sum_{\bar{\boldsymbol{x}}\in\mathcal{D}} \left|\nabla_{\bar{\boldsymbol{x}}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top \boldsymbol{\delta}\right| \right\}. \quad (9)$$

which characterizes samples whose magnitude of output variation along the trigger direction is significantly higher than the average over the training data. Samples in $\mathcal{R}_{\boldsymbol{\delta}}$ typically lie near decision boundaries aligned with the backdoor feature direction and thus act as critical counter-evidence that suppresses backdoor activation.

FU of a sample set $\mathcal{S}$ can be approximated as a first-order parameter update:

$$\Delta\boldsymbol{\theta}_{\mathcal{S}} \approx -\eta \sum_{\boldsymbol{x}_i\in\mathcal{S}} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}_i), \quad (10)$$

where $\Delta\boldsymbol{\theta}_{\mathcal{S}}$ denotes the change in the model parameters induced by performing FU on the sample set, $\ell(\cdot)$ is the training loss and η is the unlearning step size. This perturbation induces a prediction shift for an arbitrary input $\boldsymbol{x}$ as:

$$\Delta f(\boldsymbol{x}) \approx \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})^\top \Delta\boldsymbol{\theta}_{\mathcal{S}}. \quad (11)$$

which represents a first-order approximation of the change in the model output at input $\boldsymbol{x}$ induced by a small parameter update $\Delta\boldsymbol{\theta}_{\mathcal{S}}$, thereby indicating that the impact of unlearning is governed by the gradient geometry of the unlearned samples. Consequently, unlearning reshapes the decision boundary in specific directions determined by $\mathcal{S}$, rather than uniformly erasing information.

B. Backdoor Amplification via Unlearning

We next show how unlearning induced decision boundary deformation directly leads to backdoor amplification when the unlearning process is selectively manipulated.

We first consider a target sample $\boldsymbol{x}_b$ and observe the effect of unlearning on it. Let

$$\Delta f(\boldsymbol{x}_b) = f_{\boldsymbol{\theta}'}(\boldsymbol{x}_b) - f_{\boldsymbol{\theta}}(\boldsymbol{x}_b) \quad (12)$$

denote the change in the model's output for $\boldsymbol{x}_b$, where $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are the model parameters before and after unlearning, respectively. Then, the influence induced by a selected high sensitivity sample set $\mathcal{S}_{\mathrm{SSS}}$ can be measured as $\|\Delta f(\boldsymbol{x}_b)\|_{\mathcal{S}_{\mathrm{SSS}}}$.

To validate the effectiveness of the sensitivity-based selection, we compare it with randomly selected samples $\mathcal{S}_{\mathrm{rand}}$. Formally, we can write the incremental effect of a sample set $\mathcal{S}$ as follows:

$$\Delta f(\boldsymbol{x}_b;\mathcal{S}) = f_{\boldsymbol{\theta}-\eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathcal{S})}(\boldsymbol{x}_b) - f_{\boldsymbol{\theta}}(\boldsymbol{x}_b), \quad (13)$$

where $\mathcal{L}(\mathcal{S})$ is the loss over $\mathcal{S}$ and η is the learning rate.

Hence, the original formula can be proved that selectively unlearning a benign sample set $\mathcal{S}\subset\mathcal{R}_{\boldsymbol{\delta}}$ that maximizes prediction sensitivity to $\boldsymbol{\delta}$ induces a larger prediction shift on $\boldsymbol{x}_b$ than unlearning a randomly selected sample set of the same size, i.e.,

$$\|\Delta f(\boldsymbol{x}_b)\|_{\mathcal{S}_{\mathrm{SSS}}} \;>\; \|\Delta f(\boldsymbol{x}_b)\|_{\mathcal{S}_{\mathrm{rand}}}. \quad (14)$$

To make the above result more quantitative, let $m(\boldsymbol{x}_b) = f_{y_d}(\boldsymbol{x}_b) - \max_{y\neq y_d} f_y(\boldsymbol{x}_b)$ denote the classification margin of the backdoored sample toward the attacker-desired class. Under the same first-order approximation, the margin shift induced by unlearning a sample set $\mathcal{S}$ can be written as

$$\Delta m(\boldsymbol{x}_b;\mathcal{S}) \approx \nabla_{\boldsymbol{\theta}} m(\boldsymbol{x}_b)^\top \Delta\boldsymbol{\theta}_{\mathcal{S}}. \quad (15)$$

Therefore, if the selected set $\mathcal{S}_{\mathrm{SSS}}$ is more strongly aligned with the backdoor direction than a random set of the same size, then it follows that $|\Delta m(\boldsymbol{x}_b;\mathcal{S}_{\mathrm{SSS}})| > |\Delta m(\boldsymbol{x}_b;\mathcal{S}_{\mathrm{rand}})|$. This provides a quantitative interpretation of our theory: sensitivity-guided unlearning should produce a larger logit-margin displacement, which in turn yields a higher probability of crossing the decision boundary and activating the backdoor. This prediction is consistent with our empirical comparison against Rand-FedUBA in our experiment, where replacing SSS with random selection leads to a clear reduction in ASR.

As a result, our theoretical analysis yields both qualitative intuition and a measurable quantity, namely the induced logit-margin shift, which establishes a direct connection between sensitivity-based sample selection and the backdoor activation strength. We can prove that FU induces structured decision boundary deformation that can be adversarially manipulated. FedUBA is the first to theoretically demonstrate that unlearning procedures themselves constitute a novel attack surface, enabling post-training activation of backdoor behaviors without data poisoning or model tampering.

VII. Evaluation

In this section, we begin by outlining the datasets, evaluation metrics, and aggregation rules employed in our experiments. Subsequently, we perform experiments in both IID and Non-IID settings to assess the performance of FedUBA.

A. Experiment Setup

1) Datasets: We use the following datasets to evaluate the performance of FedUBA in our experiments. Specifically, PathMNIST, CIFAR-10, and GTSRB are adopted as the primary benchmark datasets, while CIFAR-100 and Tiny-ImageNet are additionally included to evaluate the scalability of FedUBA in larger classification settings.

PathMNIST[2]. This dataset is based on a study on predicting patient survival based on histological slides of colorectal cancer tissue. It comprises $100,000$ non-overlapping image patches obtained from hematoxylin and eosin-stained histological images, along with an independent test set of $7,180$ patches collected from different clinical centers. The dataset includes 9 tissue types. The original image size of $3\times224\times224$ is resized to $3\times28\times28$. In our setup, we partition $90,000$ training samples across 30 clients, each with $3,000$ samples.

CIFAR-10[3]. It is a widely-used benchmark dataset for image classification tasks. It contains $60,000$ color images of size $3\times32\times32$, evenly distributed among 10 classes ($6,000$ images per class). For our evaluation, we randomly divide

[2] https://medmnist.com/
[3] http://www.cs.toronto.edu/~kriz/cifar.html

the dataset among 20 clients, each having $3,000$ samples. In each training round, 10 clients are randomly selected for participation, and the rest of the data is used for evaluation.

GTSRB[4]. It is a benchmark for traffic sign recognition, containing $51,839$ color images that cover 43 classes of traffic signs (e.g., speed limits, stop signs, and directional signs). The images are typically resized to 32×32 or 64×64 pixels and feature real-world variations such as lighting changes and occlusions, making it more complex and realistic. In our evaluation, we randomly select $39,000$ samples into 20 different local datasets across clients, with each client holding $1,950$ samples. For each training round, 10 clients are randomly selected for training, while the remaining images are used as the test set.

CIFAR-100. This dataset consists of 20 major categories, with a total of 100 subcategories. There are 500 training samples and 100 testing samples in each subcategory, with 600 $32 \times 32 \times 3$ color images per class.

Tiny-ImageNet[5]. It is commonly used for benchmarking image classification algorithms. It contains 200 image classes, with each class having 500 training images, 50 validation images, and 50 test images. It is employed as an alternative dataset to craft backdoor triggers for attacking models trained on CIFAR-10.

Additionally, we use Tiny-ImageNet and the following dataset to train substitute models to generate backdoor triggers.

DermaMNIST[6]. Derived from the HAM10000 dataset, DermaMNIST consists of $10,015$ dermatoscopic images categorized into 7 skin disease classes. It serves as a substitute dataset to generate backdoor triggers for the PathMNIST dataset.

2) Evaluation Metrics: We utilize the following metrics to assess the performance of FedUBA:

Attack Success Rate. The ASR quantifies the proportion of backdoored samples that are incorrectly classified into the attacker-desired class by the malicious unlearned global model $\widetilde{\mathcal{M}_G}$. It is formally defined as:

$$ASR = \frac{\sum_{(\mathbf{x}_b) \in \mathcal{D}_b} \mathbb{I}(\widetilde{\mathcal{M}_G}(\mathbf{x}_b) = y_d)}{m}, \tag{16}$$

where $\mathbb{I}$ denotes indicator function, and $\mathcal{D}_b = \{(\mathbf{x}_b)\}_{t=1}^{m}$ is the set of backdoored dataset. ASR-B refers to the ASR resulting solely from the FU process without FedUBA.

Unlearned Global Testing Accuracy ($\widetilde{Acc_G}$): $\widetilde{Acc_G}$ denotes the proportion of testing samples that are correctly classified by the malicious unlearned global model $\widetilde{\mathcal{M}_G}$. It is calculated as:

$$\widetilde{Acc_G} = \frac{\sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_{test}} \mathbb{I}(\widetilde{\mathcal{M}_G}(\mathbf{x}_i) = y_i)}{n}. \tag{17}$$

where $\mathcal{D}_{test} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ is the testing dataset.

Clean Global Testing Accuracy (Acc_G). Acc_G represents the accuracy of the benign global model $\mathcal{M}_G$ on the test dataset. It is defined as:

$$Acc_G = \frac{\sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_{test}} \mathbb{I}(\mathcal{M}_G(\mathbf{x}_i) = y_i)}{n}. \tag{18}$$

3) FU Methods and Aggregation Rules: We begin by incorporating two representative FU techniques,namely FedEraser [14] and KNOT [15], both of which have been extensively studied in the previous literature. We further consider an implicit-unlearning method based on knowledge distillation, i.e., FUKD [44]. Unlike retraining-based FU, FUKD transfers the retained knowledge of the post-unlearning model through distillation without explicitly retraining on the remaining clients' data. These methods were chosen due to their wide recognition and relevance to the objectives of our work. Following this, we explore three widely used aggregation rules in FL, including FedAvg [45], and two Byzantine-robust ones, i.e., Median [46], Trimmed-mean [46]

FedAvg: FedAvg [45] is the most prevalent aggregation method in FL. It computes the global model by averaging all participating local updates. While highly effective in benign environments, FedAvg lacks robustness against adversarial manipulations.

Median: Median [46] is a coordinate-wise robust aggregation technique. For each model parameter, the server sorts all corresponding local values and selects the median. This operation mitigates the impact of extreme or malicious updates in adversarial settings.

Trimmed-mean: Trimmed-mean [46] enhances robustness by discarding the top and bottom k extreme values from the sorted set of parameter updates before averaging the remaining ones. The trimming parameter k determines the level of resilience to outliers while maintaining information fidelity.

4) Baseline: Given the absence of prior research on studying the vulnerabilities of FL to unlearning-activated backdoor attacks, we employ the following five baselines:

Rand-FedUBA: This baseline is constructed using a straightforward approach that performs attacks by randomly selecting data for unlearning. Recall that FedUBA consists of three components: **STG+ISS+BEA**. In our evaluation, Rand-FedUBA replaces the ISS component with random sample selection (denoted as '**Rand**'), while keeping the other two components unchanged.

FCBA [24]: FCBA is a novel backdoor technique in FL that constructs an expanded combinatorial set of triggers to enhance the model's inherent response. By aggregating a broader set of triggers, it forms a more comprehensive backdoor pattern in the global model. For comparison, FCBA replaces only the STG component with its own trigger generation method, while leaving the other two components intact.

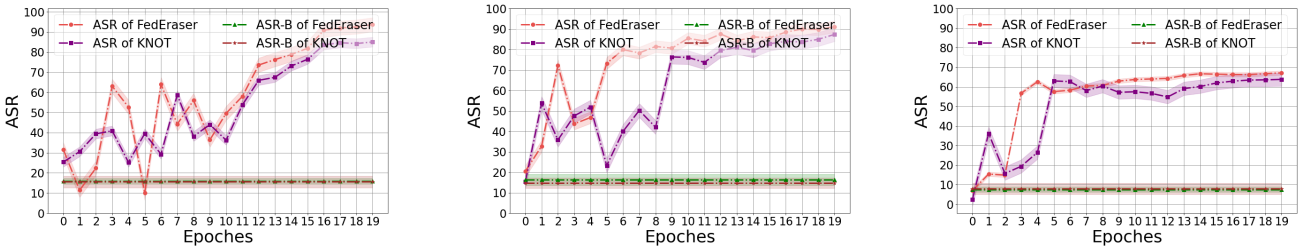
SIBA [47]: SIBA was originally designed for traditional machine learning, where it formulates trigger generation process as a bi-level optimization problem with constraints on sparsity and invisibility. For comparison, SIBA only replaces the STG component with its own trigger generation method.

FUBA [36]: FUBA is a backdoor attack via unlearning that implants latent backdoor behavior during training and

[4] https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

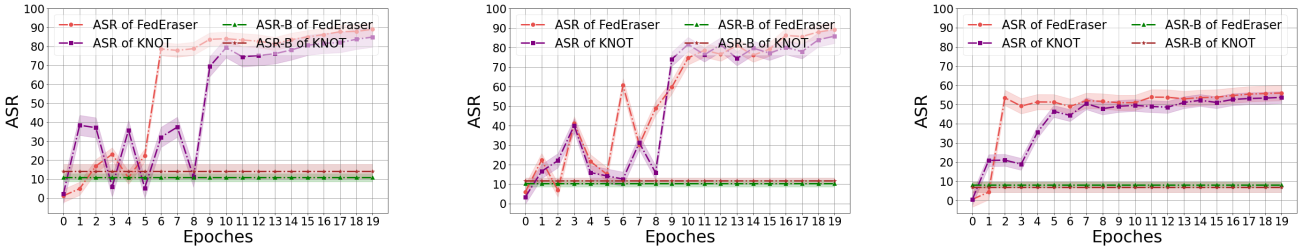
[5] http://cs231n.stanford.edu/tiny-imagenet-200.zip

[6] https://medmnist.com/



(a) PathMNIST (b) CIFAR-10 (c) GTSRB

Fig. 3: ASR of FedUBA on different FU algorithms in the IID setting.



(a) PathMNIST (b) CIFAR-10 (c) GTSRB

Fig. 4: ASR of FedUBA on different FU algorithms in the Non-IID setting.

TABLE II: Accuracy of FedUBA under various FU methods and aggregation rules in the IID setting (%)

FU Method	Aggregation Rule	PathMNIST		CIFAR-10		GTSRB	
		$Acc_G(\%)$	$\widetilde{Acc}_G$ (%)	$Acc_G(\%)$	$\widetilde{Acc}_G(\%)$	$Acc_G(\%)$	$\widetilde{Acc}_G$ (%)
FedEraser	FedAvg	**79.23±0.25**	**79.93±0.24**	85.82±0.74	**85.16±0.45**	93.76±0.35	92.24±0.41
	Median	76.62±0.63	74.61±0.43	85.41±0.48	84.74±0.53	93.92±0.85	92.73±0.34
	Trimmed-mean	78.69±0.46	77.17±0.22	**85.88±1.24**	84.87±0.79	**93.91±0.69**	**93.63±0.47**
KNOT	FedAvg	**75.23±0.64**	74.82±0.64	81.27±0.54	**81.79±0.43**	89.72±0.43	90.31±0.32
	Median	73.27±0.75	72.89±0.57	80.94±0.32	80.57±0.52	**90.75±0.75**	**91.23±0.93**
	Trimmed-mean	74.14±0.53	**75.23±0.87**	**82.03±0.24**	81.73±0.75	90.27±1.23	89.29±0.63
FUKD	FedAvg	**74.61±0.58**	74.08±0.51	**82.74±0.61**	**82.21±0.56**	91.03±0.67	90.44±0.62
	Median	73.52±0.71	73.36±0.63	82.13±0.57	81.68±0.49	**91.54±0.74**	**90.97±0.58**
	Trimmed-mean	74.05±0.62	**74.41±0.55**	82.65±0.66	82.04±0.61	91.28±0.69	90.85±0.64
Average		75.48±0.61	75.17±0.56	83.21±0.59	82.75±0.56	91.80±0.70	91.30±0.57

activates it during unlearning, typically through camouflage-sample design and training-stage manipulation.

BadFU [37]: BadFU is a backdoor attack method in FU that leverages stronger adversarial optimization to induce malicious behaviors during the unlearning process. We use it as a baseline to compare against FedUBA under a more restricted black-box setting.

Additionally, there are currently no established defense mechanisms specifically designed to counter malicious unlearning attacks. In light of this, we explore potential defense strategies and use them as baseline defenses.

FAT [48]: Federated adversarial training (FAT) is used for improving model robustness against adversarial attacks in FL setting. Since the malicious unlearning requests generated by FedUBA can be regarded as adversarial data, FAT may serve as a possible defense against such attacks.

FADngs [49]: FADngs focuses on detecting anomalous behaviors by utilizing a contrastive learning approach to train local models. This method generates more discriminative representations that are optimized for anomaly detection, by leveraging shared density functions.

FLAME [39]: FLAME is a representative backdoor-specific defense for federated learning. It mitigates malicious model updates by combining clustering-based client filtering with norm clipping and calibrated Gaussian noise injection, thereby suppressing backdoor behaviors during model aggregation.

MASA [41]: MASA is an individual unlearning-based defense that identifies potentially backdoored federated models by analyzing model behavior changes induced by per-client unlearning. Unlike standard robust aggregation defenses, MASA leverages unlearning itself as a detection and mitigation mechanism.

5) Parameter Setting: All experimental evaluations and baseline implementations are implemented using Python 3.8

TABLE III: Accuracy of FedUBA under various FU methods and aggregation rules in the Non-IID setting (%)

FU Method	Aggregation Rule	PathMNIST		CIFAR-10		GTSRB	
		$Acc_G(\%)$	$\widetilde{Acc}_G$ (%)	$Acc_G(\%)$	$\widetilde{Acc}_G(\%)$	$Acc_G(\%)$	$\widetilde{Acc}_G$ (%)
FedEraser	FedAvg	**75.09±0.58**	**75.73±0.21**	**85.31±0.49**	**84.18±0.84**	93.77±0.46	92.89±0.85
	Median	69.72±0.98	68.97±0.37	84.75±0.76	83.67±1.47	93.67±0.12	92.85±0.32
	Trimmed-mean	70.45±0.42	74.36±0.85	84.55±1.58	84.04±1.32	**93.86±0.75**	**93.28±0.25**
KNOT	FedAvg	**72.97±1.67**	**73.03±0.58**	81.39±0.64	79.31±0.57	89.21±1.64	88.05±1.67
	Median	67.81±1.36	65.39±0.34	**86.32±0.72**	**84.79±0.75**	**90.32±0.78**	**89.59±0.54**
	Trimmed-mean	68.39±1.54	69.20±1.21	80.97±0.43	81.35±1.53	87.27±1.21	88.45±0.82
FUKD	FedAvg	**71.64±1.12**	**71.92±0.73**	83.74±0.81	82.63±0.92	91.42±0.84	90.76±0.77
	Median	68.95±1.28	68.41±0.69	**84.91±0.87**	**83.58±0.96**	**91.88±0.73**	**91.03±0.61**
	Trimmed-mean	70.26±1.21	70.88±0.81	83.48±0.79	82.95±0.88	91.37±0.82	90.84±0.70
Average		70.59±1.16	70.88±0.69	83.94±0.79	82.94±0.90	91.42±0.84	90.86±0.73

with the PyTorch library. The experiments are conducted on a workstation equipped with an Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz and an NVIDIA GeForce RTX 4090 GPU. During the training phase, we adopt different image classification models tailored to each dataset to comprehensively evaluate the performance of FedUBA under varying model architectures. In the non-IID setting, we adopt a commonly used data partitioning scheme based on Dirichlet sampling [50], where each client receives a subset of samples per class according to a Dirichlet distribution with a concentration parameter set to 0.5. Specifically, we utilize MobileNetV2 [51] for PathMNIST, ResNet18 [52] for CIFAR-10, and VGG11 [53] for GTSRB.

In the unlearning attack phase, we first train a substitute model using both public datasets. This substitute model is then leveraged to generate backdoor triggers. A set of samples is selected and combined with the learned triggers to construct the unlearning dataset. The upper bound of the backdoor trigger intensity ϵ is treated as a hyperparameter. Its influence is analyzed through ablation studies, while it is fixed to $32/255$ in other experiments to evaluate the overall attack performance. This noise threshold, commonly used in centralized backdoor literature [54].

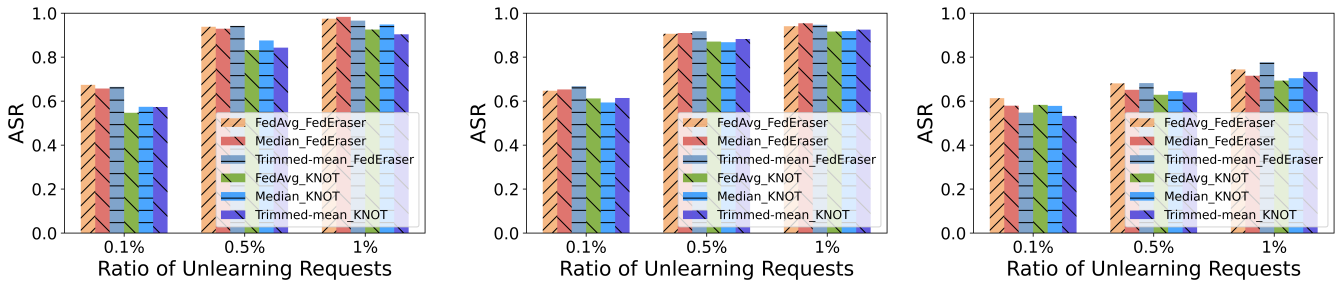
B. Effectiveness of FedUBA

We begin by conducting a comparative analysis to evaluate the attack performance of FedUBA. The experimental results across various datasets, FU methods, and aggregation rules under both IID and Non-IID settings are presented in Table II and Table III. It is important to highlight that the attacker has two primary objectives: Goal I is measured by ASR-B and ASR, while Goal II is assessed through $\widetilde{Acc}_G$ and Acc_G.

In the IID setting, the number of malicious clients is fixed at 2, the forgetting rate is set to 0.5%, and the maximum perturbation of the backdoor trigger is constrained to $32/255$. As shown in Fig. 3 and Table II, the experimental results demonstrate that FedUBA can achieve a significant difference between ASR and ASR-B under different unlearning epoches. The ASR-B varies noticeably across datasets. For PathMNIST and CIFAR-10, the average ASR-B reaches around 15%, while GTSRB yields about 7%. These results indicate that the backdoor triggers generated by FedUBA are able to exploit class-dependent noise and already exert a certain influence on the model even before the attack. After activating the attack, the ASR increases significantly. This improvement is primarily attributed to the fact that FedUBA enhances the model's sensitivity to backdoor triggers through the use of unlearning algorithms. Notably, for PathMNIST and CIFAR-10, ASR reaches up to 89% across various unlearning and aggregation strategies. For GTSRB, a relatively high ASR of about 65% is also observed. The differences in ASR mainly because that PathMNIST and CIFAR-10 have fewer classes than GTSRB, and GTSRB contains more realistic images with richer scene. Although the average ASR-B reaches approximately 15% on PathMNIST and CIFAR-10, it remains substantially lower than the post-activation ASR, which exceeds 80% in most settings. This indicates that the trigger alone introduces only a weak semantic bias before malicious unlearning requests are issued. From a practical perspective, ASR-B should be interpreted together with the activated ASR and the baseline prediction variability, rather than by its absolute value alone. Moreover, such a level may be comparable to natural prediction fluctuations caused by class overlap, model uncertainty, and dataset-specific noise, making it less likely to raise suspicion under backdoor detection procedures.

Table II presents the result of $\widetilde{Acc}_G$ and Acc_G. Due to dataset-specific characteristics and differences in model architectures, the absolute accuracy varies across datasets. However, for each individual dataset, the prediction accuracy of the global model shows negligible change within approximately 1% before and after the attack. This suggests that FedUBA has minimal impact on clean samples and can effectively preserves model utility. After further introducing FUKD, we observe a consistent trend. Specifically, the average $\widetilde{Acc}_G$ in Table II remains 75.17% on PathMNIST, 82.75% on CIFAR-10, and 91.30% on GTSRB, indicating only a limited utility drop compared with the retraining-based settings. This result suggests that although knowledge distillation induces less controllable model updates than retraining-based FU, FedUBA still remains feasible in this unlearning scenario.

Moving to the Non-IID setting, as shown in Fig. 4 and Table III, FedUBA also shows demonstrate competitive performance. The ASR exceeding 85% on PathMNIST and CIFAR-



(a) PathMNIST (b) CIFAR-10 (c) GTSRB

Fig. 5: ASR of FedUBA under different ratios of unlearning requests.

10, and surpassing 50% on GTSRB. This observation indicates that the generated backdoor triggers inherently embed critical features of the attacker-desired class, and confirms that our attack significantly enhances the ASR. Additionally, the results of $\widetilde{Acc}_G$ and Acc_G reveal that the global model's predictive performance on clean testing samples remains largely unaffected after our attack. However, compared to the IID setting, a slight degradation is observed in both ASR and $\widetilde{Acc}_G$. This decline can be attributed to the inherent heterogeneity in data distributions across clients in Non-IID scenarios. Such diversity poses challenges to federated aggregation and complicates the task of conducting effective backdoor attacks. A similar phenomenon is observed after adding FUKD. As shown in Table III, the average $\widetilde{Acc}_G$ remains 70.88% on PathMNIST, 82.94% on CIFAR-10, and 90.86% on GTSRB. Compared with the IID case, the utility degradation under knowledge distillation becomes slightly more evident in the heterogeneous Non-IID setting, which is consistent with the intuition that implicit FU is more sensitive to data heterogeneity. Nevertheless, the overall degradation is still moderate, supporting that FedUBA can generalize beyond the specific retraining-based FU pipelines considered originally.

C. Impact of Ratio of Unlearning Requests

In this section, we investigate the impact of varying unlearning rates on the attack effectiveness. All experiments are conducted under the IID setting, with a fixed backdoor perturbation bound of $32/255$ and two malicious clients involved. As shown in Fig. 5, under the FedEraser algorithm, the results across all datasets consistently demonstrate that increasing the unlearning rate leads to higher ASR. For the PathMNIST and CIFAR-10 datasets, an ASR exceeding 60% can be achieved even with a forgetting rate as low as 0.1%. Furthermore, the ASR reaches over 90% at a unlearning rate of 0.5%. This trend can be attributed to the relatively ideal conditions of these datasets, including high-quality images that facilitate effective extraction of class-representative features by our method. For the GTSRB dataset, although it reflects a more realistic and complex visual environment, the ASR still improves significantly with increasing forgetting rates. An ASR exceeding 50% is observed at just 0.1%, underscoring that FedUBA maintains strong effectiveness even in more challenging scenarios.

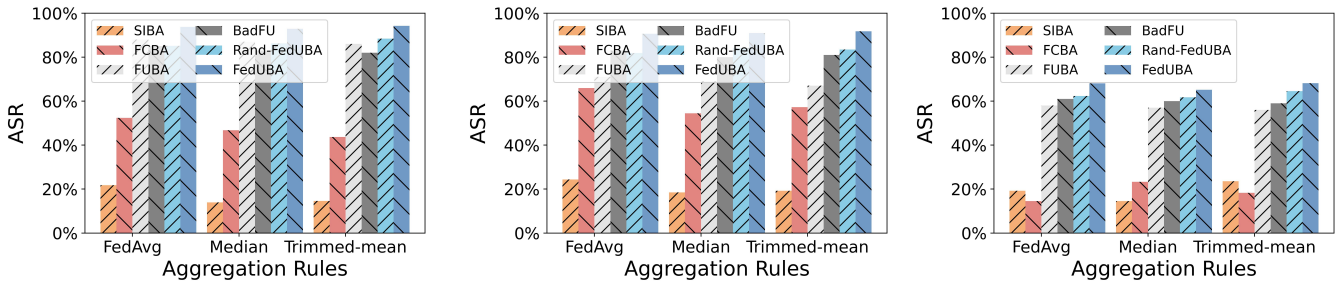
Fig. 5 also presents the attack performance under varying unlearning rates using the KNOT algorithm. A similar upward trend in ASR is observed as the forgetting rate increases, especially within the 0.1% to 0.5% range. However, when compared to FedEraser, the performance under KNOT is consistently lower. This discrepancy is primarily due to the asynchronous nature of KNOT, which inherently introduces variability in the unlearning process. By decoupling client updates, the asynchronous design inadvertently attenuates the effectiveness of the attack, resulting in diluted ASR outcomes.

D. Effectiveness of FedUBA and Baseline Attacks

In this section, we conduct comparative experiments based on five baseline backdoor attacks. Specifically, the SIBA method generates invisible triggers by formulating a bi-level optimization problem, while the FCBA method employs combined triggers to perform backdoor attacks. Rand-FedUBA, on the other hand, randomly selects samples to modify their features. Similar to FedUBA, both methods manipulate only the features without altering the labels. In addition, we further include two recent FU-oriented attack baselines, namely FUBA and BadFU. We uniformly set the maximum perturbation budget to $32/255$, involving two malicious clients and a fixed unlearning rate of 0.5%.

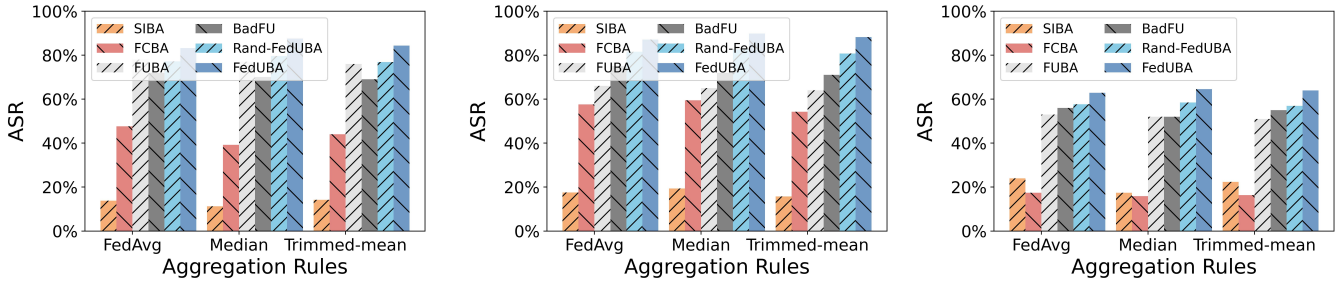
Fig. 6 shows the performance of various baseline attacks under FedEraser across different datasets. FedUBA achieves the highest ASR on all three datasets, Rand-FedUBA is the closest variant but still slightly worse, while SIBA, FCBA, FUBA, and BadFU remain clearly below FedUBA. On the PathMNIST and CIFAR-10 datasets, FedUBA still delivers an approximately $2\times$ improvement over the weaker baselines, and it maintains a clear advantage over FUBA and BadFU. On the more realistic GTSRB dataset, the gap becomes even more evident, where FedUBA shows about a $3\times$ gain over the weakest baselines and still preserves a noticeable margin over FUBA and BadFU. Notably, the FCBA method also achieves around 50% ASR on PathMNIST and CIFAR-10, which can be attributed to the relatively low number of classes and their distinct class-specific features, making simple feature mixing method relatively effective.

Fig. 7 further illustrates the experimental results under the KNOT unlearning algorithm. FedUBA remains the best-performing method across all datasets, whereas the other



(a) PathMNIST (b) CIFAR-10 (c) GTSRB

Fig. 6: ASR of FedUBA and Baseline Attacks under FedEraser algorithm.



(a) PathMNIST (b) CIFAR-10 (c) GTSRB

Fig. 7: ASR of FedUBA and Baseline Attacks under KNOT algorithm.

TABLE IV: Attack performance under different perturbation bounds with FedEraser

Perturbation bound	Aggregation Rule	PathMNIST			CIFAR-10			GTSRB		
		ASR	$Acc_G(\%)$	$\widetilde{Acc}_G$ (%)	ASR	$Acc_G(\%)$	$\widetilde{Acc}_G(\%)$	ASR	$Acc_G(\%)$	$\widetilde{Acc}_G$ (%)
24/255	FedAvg	81.29	**79.23±0.25**	**79.14±0.43**	78.52	85.82±0.74	85.32±0.57	**56.23**	93.76±0.35	**92.96±0.74**
	Median	**83.53**	76.62±0.63	75.39±0.32	77.83	85.41±0.48	84.69±0.35	52.74	**93.92±0.85**	92.76±0.21
	Trimmed-mean	82.75	78.69±0.46	77.53±0.22	**79.51**	**85.88±1.24**	**85.74±0.89**	54.38	93.91±0.69	92.62±0.42
32/255	FedAvg	93.79	**79.23±0.25**	**79.93±0.24**	90.61	85.82±0.74	**85.16±0.45**	68.07	93.76±0.35	92.24±0.41
	Median	92.91	76.62±0.63	74.61±0.43	90.97	85.41±0.48	84.74±0.53	65.16	**93.92±0.85**	92.73±0.34
	Trimmed-mean	**94.23**	78.69±0.46	77.17±0.22	**91.75**	**85.88±1.24**	84.87±0.53	**68.17**	93.91±0.69	**93.63±0.47**
40/255	FedAvg	97.34	**79.23±0.25**	**78.45±0.24**	96.42	85.82±0.74	**84.95±0.42**	82.85	93.76±0.35	**92.97±0.31**
	Median	97.96	76.62±0.63	75.73±0.31	97.21	85.41±0.48	84.37±0.63	82.76	**93.92±0.85**	92.70±0.57
	Trimmed-mean	**98.47**	78.69±0.46	77.54±0.47	**98.45**	**85.88±1.24**	84.28±1.53	**83.30**	93.91±0.69	92.95±0.39
Average		91.36	78.18±0.45	77.28±0.32	89.03	85.70±0.82	84.90±0.66	68.18	93.86±0.63	92.84±0.43

baselines, including FUBA and BadFU, suffer a more obvious reduction under asynchronous unlearning. Due to the nature of asynchronous FU adopted in KNOT, all attacks experience a noticeable degree of performance degradation compared to the FedEraser setting, but the degradation is especially evident on GTSRB and more moderate on PathMNIST and CIFAR-10. This performance drop is primarily attributed to the increased randomness introduced by KNOT during the aggregation. Specifically, KNOT processes unlearning requests in an asynchronous and decoupled manner, which breaks the coordination between the attacker's trigger optimization and the global model updates, thereby disrupting the effectiveness of carefully crafted backdoor strategies.

E. Impact of ϵ

In this section, we investigate the impact of hyperparameters, with a particular focus on the upper bound of perturbation used for backdoor generation in FedUBA. Specifically, we vary the perturbation upper bound from $24/255$ to $40/255$ in increments of $8/255$ and evaluate the attack performance across different datasets and unlearning algorithms. For all experiments, the number of malicious clients is fixed at 2, and the forgetting rate is set to 0.5%.

The experimental results under the FedEraser unlearning frameworks are summarized in Table IV, respectively. Overall, we observe that small perturbation bounds still achieve notable ASR across various settings. As the perturbation bound increases, the ASR generally improves. Specifically, for the PathMNIST and CIFAR-10 datasets, an upper bound of $24/255$ is already sufficient to achieve an ASR of approximately 80%. Further increasing the bound to $40/255$ only results in a marginal improvement (around 10%). In contrast, the GTSRB dataset, which initially exhibits lower ASR, sees a more substantial improvement of nearly 20% when the

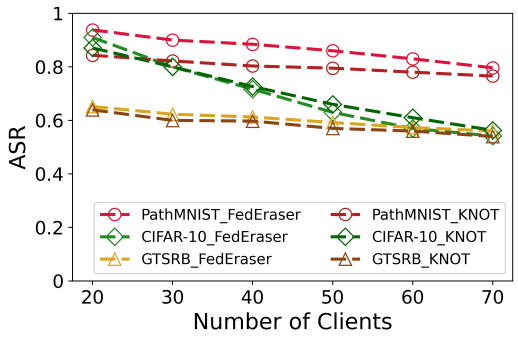


Fig. 8: Impact of the number of clients.

bound increases from $24/255$ to $40/255$. PathMNIST and CIFAR-10 are relatively clean with less noise, making it easier to extract class-representative features. On the other hand, GTSRB reflects real-world conditions and includes noise from various sources, making feature extraction more challenging.

F. Multi-Client Scenario

In this section, we evaluate the performance of FedUBA under a multi-client scenario. Experiments are conducted on the PathMNIST, CIFAR-10, and GTSRB datasets, with the number of clients varying from 20 to 70. The number of malicious clients is fixed at 2, and the ratio of malicious unlearning requests is set to 0.5%. As shown in Fig. 8, ASR consistently declines with increasing client numbers across all settings, indicating that larger clients enhance robustness by increasing aggregation diversity and reducing the influence of malicious updates. Specifically, the ASR on PathMNIST under the FedEraser setting decreases from 93.79% to 79.63%, while under KNOT it drops from 95.22% to 79.56%, demonstrating the consistently stronger defense of KNOT and similar trends are observed for CIFAR-10 and GTSRB.

G. Effectiveness of Defenses Against FedUBA

To assess the effectiveness of FedUBA against different baseline defenses, we evaluate the ASR of FedUBA under FedEraser and KNOT settings across two representative defense strategies discussed earlier, namely FAT and FADngs. All experiments are conducted under the same parameter settings as in the IID setting. The ratio of malicious unlearning requests is set to 0.5%, and the number of malicious clients is fixed at 2. The results are presented in Fig. 9, where ASR is plotted against varying values of the adversarial sample ratio r and shrinkage intensity ρ across three benchmark datasets.

Under the FAT defense illustrated in Fig. 9(a), FedUBA demonstrates consistently higher ASR under the FedEraser setting compared to the KNOT setting across all datasets. For instance, when $r = 0.1$, the ASR on PathMNIST under the FedEraser setting surpasses 0.85, whereas it stays below 0.75 under the KNOT setting. A similar performance gap is observed on CIFAR-10. In contrast, GTSRB exhibits substantially lower ASR under both defenses, indicating that either the dataset characteristics or model architecture make it inherently more resistant to FedUBA. As presented in Fig. 9(b), a similar trend is observed, where FedUBA under the KNOT setting consistently achieves lower ASR than under the FedEraser setting, despite a slight overall reduction in ASR for both defenses. For instance, on GTSRB, the ASR under the FedEraser setting drops to approximately 0.5, while the KNOT setting achieves even lower ASR. This result implies that FADngs contributes additional robustness through feature-distribution alignment.

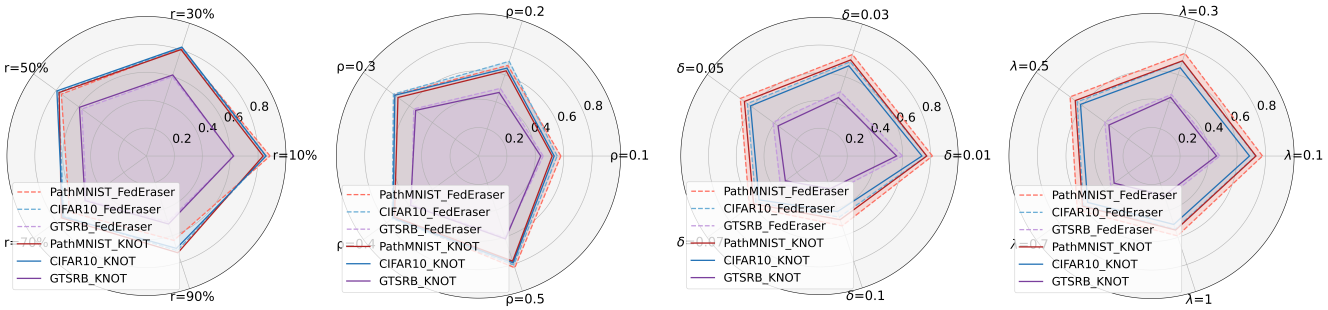
TABLE V: Attack performance of FedUBA on larger-scale benchmarks under different FU methods (%).

Dataset	Setting	FU Method	ASR	Acc_G	$\widetilde{Acc_G}$
CIFAR-100	IID	FedEraser	68.42	66.83±0.58	65.97±0.64
		KNOT	63.37	64.46±0.71	63.81±0.76
	Non-IID	FedEraser	61.18	64.74±0.82	63.63±0.77
		KNOT	56.46	62.28±0.93	61.34±0.88
Tiny-ImageNet	IID	FedEraser	57.84	45.36±0.69	44.51±0.73
		KNOT	52.63	43.92±0.77	43.08±0.81
	Non-IID	FedEraser	50.27	43.87±0.84	42.78±0.88
		KNOT	45.91	42.15±0.91	41.12±0.94

Fig. 9(c) and Fig. 9(d) further report the results under FLAME and MASA. For FLAME, we vary the Gaussian noise level δ from 0.01 to 0.1. A clear trend is observed under both FedEraser and KNOT: increasing δ consistently reduces ASR. For example, under FedEraser, the ASR on PathMNIST decreases from 81.46% to 52.96%, while under KNOT it decreases from 77.35% to 48.27%; similar trends can also be observed on CIFAR-10 and GTSRB. For MASA, we vary the fusion degree λ from 0.1 to 1.0. The results show that smaller λ values lead to weaker defense effectiveness, and the ASR is relatively higher at $\lambda = 0.3$, which is consistent with the observation in [41] that a smaller fusion degree may overly compromise local distinctive features and thus reduce the detection quality. As λ increases toward 0.7 and 1.0, MASA becomes more effective.

H. Scalability on a Larger-Scale Benchmark

To further evaluate the scalability of FedUBA, we additionally conduct experiments on CIFAR-100 and Tiny-ImageNet. CIFAR-100 evaluates scalability to a larger label space, while Tiny-ImageNet further increases both image resolution and visual complexity. The experimental setting follows the same protocol as in the main experiments, and the results are summarized in Table V. Overall, FedUBA remains effective on both datasets under FedEraser and KNOT, although the ASR is lower than that on CIFAR-10 and PathMNIST. This is expected because the larger label space, higher-resolution inputs, and finer-grained inter-class similarity make it more difficult for the generated trigger to consistently shift a backdoored sample toward the attacker-desired class. On CIFAR-100, FedUBA achieves ASRs of 68.42% and 63.37% in the IID setting under FedEraser and KNOT, respectively, and 61.18% and 56.46% in the Non-IID setting. On Tiny-ImageNet, the ASR further decreases but remains effective, reaching 57.84% and 52.63% in the IID setting and 50.27% and 45.91% in the Non-IID setting.Meanwhile, the clean accuracy degradation



(a) FAT (b) FADngs (c) FLAME (d) MASA

Fig. 9: ASR of baseline defenses against FedUBA.

TABLE VI: Time overhead for FedUBA's components (s).

Time overhead / Component \ Dataset	PathMNIST	CIFAR-10	GTSRB
STG	456.62	672.23	482.74
SSS	24.49	38.08	51.64
BEA	0.87	1.03	1.25

TABLE VII: Comparison of benign and unlearned samples using feature magnitude and KL divergence on different datasets.

Dataset	Averaged Feature Magnitude		KL Divergence	
	Benign Samples	Unlearned Samples	Benign Samples	Unlearned Samples
PathMNIST	1.86	2.24	0.06	0.08
CIFAR-10	0.74	0.93	0.05	0.07
GTSRB	1.02	1.28	0.11	0.16

remains limited across both datasets, indicating that FedUBA can scale to larger and more complex visual benchmarks while largely preserving model utility.

I. Evaluation on Run-time Overhead

Lastly, we evaluate the computational overhead of FedUBA, conducted on an NVIDIA GeForce RTX 4090 GPU. The run-time for FedUBA is divided into three parts: STG, SSS and BEA. The experimental results are summarized in Table VI. For the STG phase, the run-time varies across datasets due to differences in input complexity and data scale. Among the three datasets, CIFAR-10 exhibits the highest time cost, followed by GTSRB and PathMNIST. The higher cost in CIFAR-10 is attributed to its greater data dimensionality and the slower convergence of trigger optimization, which require more optimization iterations for effective trigger generation.

In the SSS phase, we report the time to identify 100 sensitivity samples. The process is relatively efficient, with run-times of $24.49s$, $38.08s$, and $51.64s$ for PathMNIST, CIFAR-10, and GTSRB, respectively. The increase in time for GTSRB reflects the added complexity in computing sample influence under its multi-class traffic sign distribution. In the final BEA phase, we measure the time taken to generate $1,000$ malicious unlearning samples. The overhead is minimal across all datasets, as this stage involves lightweight feature perturbation rather than computationally intensive sample selection or trigger optimization.

J. Stealthiness Analysis of Unlearning Requests

To further substantiate the stealthiness of FedUBA, we provide a quantitative comparison between benign and unlearned samples from two complementary perspectives: *averaged feature magnitude* and *KL divergence* between the model outputs before and after unlearning. The former characterizes the overall intensity of sample representations extracted from the penultimate layer, while the latter quantifies the change in predictive distribution induced by unlearning. From the results in Table VII, we can see that across all three datasets, the averaged feature magnitude of unlearned samples is only moderately larger than that of benign samples, increasing from 1.86 to 2.24 on PathMNIST, from 0.74 to 0.93 on CIFAR-10, and from 1.02 to 1.28 on GTSRB. A similar pattern is observed for KL divergence, which rises from 0.06 to 0.08 on PathMNIST, from 0.05 to 0.07 on CIFAR-10, and from 0.11 to 0.16 on GTSRB. These results indicate that unlearned samples do not exhibit dramatic deviations from benign samples, supporting the stealthiness of FedUBA.

To provide a more intuitive view, Fig. 10 visualizes representative samples before and after the FedUBA attack on PathMNIST, CIFAR-10, and GTSRB datasets. The visualization follows the same attack configuration as the main experiments, where the perturbation budget is fixed to $\epsilon = 24/255$, the number of malicious clients is set to two, and the malicious unlearning rate is set to 0.5%. For each dataset, the left side shows the original benign sample with its ground-truth label, while the right side shows the corresponding attacked sample associated with the attacker-specified target label. The zoomed-in patches further highlight the local regions where the perturbations are introduced. As shown in the normal image view, the attacked samples largely preserve the main semantic content and visual structure of the original samples, and the perturbations remain visually subtle. This visual evidence is consistent with the quantitative results in Table VII, suggesting that FedUBA does not introduce obvious visual artifacts into the unlearning requests.

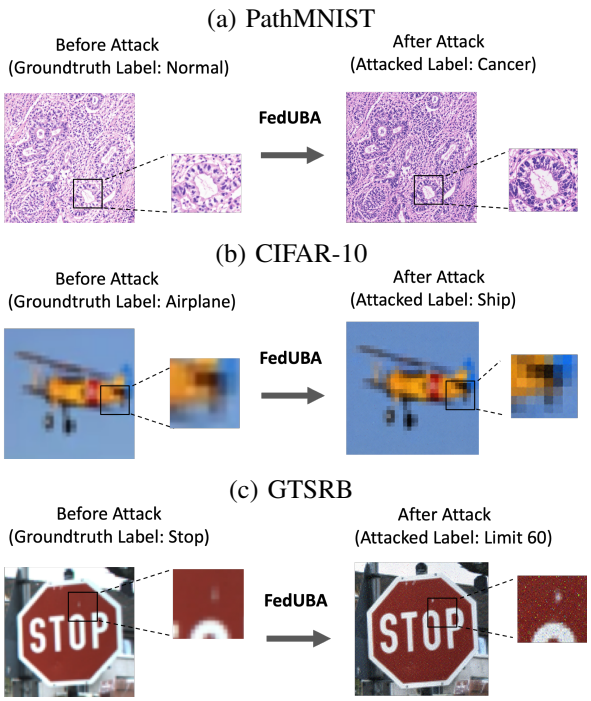


Fig. 10: Visualization examples on PathMNIST, CIFAR-10, and GTSRB.

VIII. Ethical and Mitigation Discussions

This paper studies FedUBA for defensive and scientific purposes only. Although this work is attack-oriented, its purpose is to present a new formulation of FU-activated backdoor attacks and to motivate corresponding defenses. The goal is to help the community understand how FU mechanisms may be misused, so that more robust and secure FU protocols can be designed. We do not advocate malicious deployment of the attack. To reduce misuse risk, we limit our discussion to the minimum technical detail necessary to support reproducibility and security evaluation, and we frame the attack together with two potential mitigation directions. (1) *Unlearning request auditing:* the server can screen candidate unlearning requests by monitoring whether the requested samples induce abnormal feature shifts, unusually concentrated class-wise forgetting patterns, or excessive logit-margin changes on a small trusted validation set. (2) *Robust FU with consistency constraints:* FU updates can be regularized to preserve prediction consistency on held-out clean data and to bound the representation drift caused by each unlearning request, thereby limiting the decision-boundary deformation exploited by FedUBA.

Despite its effectiveness, FedUBA has several limitations. First, the attack assumes access to semantically related surrogate data, which may not always be available in highly specialized FL domains. Second, our current evaluation focuses on image classification tasks, and extending the attack to other modal FL systems requires further investigation. Future work will explore unlearning triggered backdoors in emerging federated agent systems, where long-term memory, multi-agent collaboration, and dynamic knowledge sharing may introduce new attack surfaces and activation pathways.

IX. Conclusion

In this work, we have presented FedUBA, a novel and effective backdoor attack framework that exploits FU mechanisms to stealthily activate camouflaged backdoors. To achieve this, FedUBA introduces a three-stage pipeline, which collaboratively enable inconspicuous yet highly effective backdoor behaviors. These backdoors are triggered via strategically crafted unlearning requests, allowing the attack to remain stealthy and persistent. Extensive experiments conducted on multiple datasets and FU methods demonstrate the effectiveness of FedUBA. We believe this study reveals a previously overlooked threat surface in FU, and opens up new research directions in adversarial machine learning under the FU setting.

References

[1] X. Gao, X. Ma, J. Wang, Y. Sun, B. Li, S. Ji, P. Cheng, and J. Chen, "Verifi: Towards verifiable federated unlearning," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 6, pp. 5720–5736, 2024.

[2] S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, Y. Yao, C. Y. Liu, X. Xu, H. Li *et al.*, "Rethinking machine unlearning for large language models," *Nature Machine Intelligence*, pp. 1–14, 2025.

[3] W. Wang, C. Zhang, Z. Tian, and S. Yu, "Fedu: Federated unlearning via user-side influence approximation forgetting," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 03, pp. 2550–2562, 2025.

[4] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[5] E. L. Harding, J. J. Vanto, R. Clark, L. Hannah Ji, and S. C. Ainsworth, "Understanding the scope and impact of the california consumer privacy act of 2018," *Journal of Data Protection & Privacy*, vol. 2, no. 3, pp. 234–253, 2019.

[6] W. Wu, J. Jiang, and C. Hu, "Aegis: Post-training attribute unlearning in federated recommender systems against attribute inference attacks," in *Proceedings of the ACM Web Conference*, 2025, pp. 3783–3793.

[7] Z. Liu, Y. Jiang, J. Shen, M. Peng, K.-Y. Lam, X. Yuan, and X. Liu, "A survey on federated unlearning: Challenges, methods, and future directions," *ACM Computing Surveys*, vol. 57, no. 1, pp. 1–38, 2024.

[8] S. Zhao, J. Zhang, X. Ma, Q. Jiang, Z. Ma, S. Gao, Z. Ying, and J. Ma, "FedWiper: Federated unlearning via universal adapter," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 4042–4054, 2025.

[9] H. Gu, W. Ong, C. S. Chan, and L. Fan, "Ferrari: federated feature unlearning via optimizing feature sensitivity," in *Proceedings of NeurIPS*, vol. 37, 2024, pp. 24 150–24 180.

[10] Z. Liu, Y. Jiang, W. Jiang, J. Guo, J. Zhao, and K.-Y. Lam, "Guaranteeing data privacy in federated unlearning with dynamic user participation," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 3, pp. 2072–2085, 2025.

[11] J. Chen, Z. Lin, W. Lin, W. Shi, X. Yin, and D. Wang, "Fedmua: Exploring the vulnerabilities of federated learning to malicious unlearning attacks," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 1665–1678, 2025.

[12] M. Naseri, Y. Han, and E. De Cristofaro, "BadVFL: Backdoor attacks in vertical federated learning," in *Proceedings of IEEE Symposium on Security and Privacy*, 2024, pp. 2013–2028.

[13] Z. Zhong, W. Bao, J. Wang, S. Zhang, J. Zhou, L. Lyu, and W. Y. B. Lim, "Unlearning through knowledge overwriting: Reversible federated unlearning via selective sparse adapter," in *Proceedings of IEEE/CVF CVPR*, 2025.

[14] G. Liu, X. Ma, Y. Yang, C. Wang, and J. Liu, "Federaser: Enabling efficient client-level data removal from federated learning models," in *Proceedings of IWQoS*, 2021, pp. 1–10.

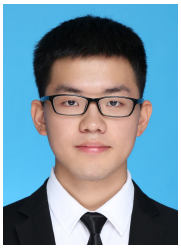
[15] N. Su and B. Li, "Asynchronous federated unlearning," in *Proceedings of INFOCOM*, 2023, pp. 1–10.

[16] M. Ameen, P. Wang, W. Su, X. Wei, and Q. Zhang, "Speed up federated unlearning with temporary local models," *IEEE Transactions on Sustainable Computing*, pp. 1–16, 2025.

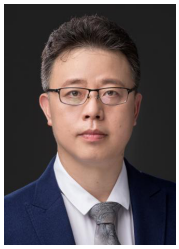
[17] A. Halimi, S. R. Kadhe, A. Rawat, and N. B. Angel, "Federated unlearning: How to efficiently erase a client in fl?" in *Proceedings of ICML*, 2022.

[18] Y. Zhao, P. Wang, H. Qi, J. Huang, Z. Wei, and Q. Zhang, "Federated unlearning with momentum degradation," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 8860–8870, 2023.
[19] J. Wang, S. Guo, X. Xie, and H. Qi, "Federated unlearning via class-discriminative pruning," in *Proceedings of the ACM Web Conference*, 2022, pp. 622–632.
[20] X. Sheng, W. Bao, and L. Ge, "Robust federated unlearning," in *Proceedings of ACM CIKM*, 2024, pp. 2034–2044.
[21] W. Wang, Q. Ma, Z. Zhang, Y. Liu, Z. Liu, and M. Fang, "Poisoning attacks and defenses to federated unlearning," in *Companion Proceedings of the ACM on Web Conference*, 2025, pp. 1365–1369.
[22] F. Zhang, W. Li, Y. Hao, X. Yan, Y. Cao, and W. Y. B. Lim, "Verifiably forgotten? gradient differences still enable data reconstruction in federated unlearning," *CoRR, arXiv: 2505.11097*, 2025.
[23] Y. Miao, R. Xie, X. Li, Z. Liu, K.-K. R. Choo, and R. H. Deng, "Efficient and secure federated learning against backdoor attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 5, pp. 4619–4636, 2024.
[24] T. Liu, Y. Zhang, Z. Feng, Z. Yang, C. Xu, D. Man, and W. Yang, "Beyond traditional threats: A persistent backdoor attack on federated learning," in *Proceedings of AAAI*, 2024, pp. 21 359–21 367.
[25] M. Fan, Z. Hu, F. Wang, and C. Chen, "Bad-PFL: Exploring backdoor attacks against personalized federated learning," in *Proceedings of ICLR*, 2025.
[26] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *Proceedings of ICLR*, 2020.
[27] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, and J. Gonzalez, "Neurotoxin: Durable backdoors in federated learning," in *Proceedings of ICML*, 2022, pp. 26 429–26 446.
[28] H. Zhang, J. Jia, J. Chen, L. Lin, and D. Wu, "A3fl: Adversarially adaptive backdoor attacks to federated learning," in *Proceedings of NeurIPS*, vol. 36, 2023, pp. 61 213–61 233.
[29] W. Shen, W. Huang, G. Wan, and M. Ye, "Label-free backdoor attacks in vertical federated learning," in *Proceedings of AAAI*, 2025, pp. 1–9.
[30] J. Z. Di, J. Douglas, J. Acharya, G. Kamath, and A. Sekhari, "Hidden poison: Machine unlearning enables camouflaged poisoning attacks," in *Proceedings of NeurIPS ML Safety Workshop*, 2022.
[31] W. Qian, C. Zhao, W. Le, M. Ma, and M. Huai, "Towards understanding and enhancing robustness of deep learning models against malicious unlearning attacks," in *Proceedings of SIGKDD*, 2023, pp. 1932–1942.
[32] C. Zhao, W. Qian, R. Ying, and M. Huai, "Static and sequential malicious attacks in the context of selective forgetting," *Proceedings of NeurIPS*, vol. 36, 2023.
[33] J. Chen, W. Shi, W. Lin, C. Wang, W. Liu, H. Sun, and G. Liu, "Unlearning attacks for regression learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2025.
[34] H. Hu, S. Wang, J. Chang, H. Zhong, R. Sun, S. Hao, H. Zhu, and M. Xue, "A duty to forget, a right to be assured? exposing vulnerabilities in machine unlearning services," in *Proceedings of NDSS*, 2024.
[35] Z. Huang, Y. Mao, and S. Zhong, "UBA-Inf: Unlearning activated backdoor attack with influence-driven camouflage," in *Proceedings of USENIX Security Symposium*, 2024, pp. 4211–4228.
[36] X. Sheng, W. Bao, Y. Guo, and S. Fu, "FUBA: Backdoor federated learning via federated unlearning," *IEEE Transactions on Artificial Intelligence*, to appear. DOI: 10.1109/TAI.2025.3630110.
[37] B. Lu, H. Hu, Y. Miao, S. Sohail, C. He, S. Wang, and X. Chen, "BadFU: Backdoor federated learning through adversarial machine unlearning," in *Proceedings of RAID*, 2025, pp. 773–788.
[38] J. Xu, Z. Zhang, and R. Hu, "Detecting backdoor attacks in federated learning via direction alignment inspection," in *Proceedings of IEEE/CVF CVPR*, 2025, pp. 20 654–20 664.
[39] T. D. Nguyen, P. Rieger, H. Chen, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, S. Zeitouni *et al.*, "FLAME: Taming backdoors in federated learning," in *Proceedings of USENIX Security Symposium*, 2022, pp. 1415–1432.
[40] S. Huang, Y. Li, C. Chen, L. Shi, and Y. Gao, "Multi-metrics adaptively identifies backdoors in federated learning," in *Proceedings of ICCV*, 2023, pp. 4652–4662.
[41] J. Xu, Z. Zhang, and R. Hu, "Identify backdoored model in federated learning via individual unlearning," in *Proceedings of WACV*, 2025, pp. 7960–7969.
[42] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proceedings of S&P*, 2017, pp. 3–18.
[43] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proceedings of ICML*, 2017, pp. 1885–1894.
[44] C. Wu, S. Zhu, and P. Mitra, "Federated unlearning with knowledge distillation," *CoRR, arXiv: 2201.09441*, 2022.
[45] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of AISTATS*, 2017, pp. 1273–1282.
[46] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of ICML*, 2018, pp. 5650–5659.
[47] Y. Gao, Y. Li, X. Gong, Z. Li, S.-T. Xia, and Q. Wang, "Backdoor attack with sparse and invisible trigger," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 6364–6376, 2024.
[48] G. Zizzo, A. Rawat, M. Sinn, and B. Buesser, "Fat: Federated adversarial training," in *Proceedings of NeurIPS Workshop on SpicyFL*, 2020.
[49] B. Dong, D. Chen, Y. Wu, S. Tang, and Y. Zhuang, "Fadngs: Federated learning for anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 2, pp. 2578–2592, 2025.
[50] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *CoRR, arXiv: 1909.06335*, 2019.
[51] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of IEEE/CVF CVPR*, 2018, pp. 4510–4520.
[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE/CVF CVPR*, 2016, pp. 770–778.
[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of ICLR*, 2015.
[54] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proceedings of AAAI*, 2020, pp. 11 957–11 965.

Jian Chen (Member, IEEE) is currently an Associate Professor in the Department of Computer Science, China University of Geosciences (Wuhan), China. He received his Ph.D. degree from the School of Electronic Information and Communications at the Huazhong University of Science and Technology in 2023. He received B.S. degree from Hubei University of Technology in 2014 and the M.S. degree from Huazhong University of Science and Technology in 2018. His recent research interests focus on AI security and federated learning.



Wenlong Shi received the B.E. degree from Wuhan University of Technology, China, in 2022. He is currently pursuing the M.S. degree in Information and Communication Engineering at Huazhong University of Science and Technology, China. His research interests include machine unlearning and data poisoning attacks.



Chengyu Hu (Member, IEEE) received the M.S. degree in automation and control from Wuhan University of Technology, China, in 2003, and the Ph.D. degree in automation control from the Huazhong University of Science and Technology, China, in 2010. He is currently a Professor and the Vice Dean of the School of Computer Science, China University of Geosciences, Wuhan, China. His research interests include evolutionary algorithms, reinforcement learning, and cloud computing.

Jianfeng Lu (Member, IEEE) received the PhD degree in computer application technology from the Huazhong University of Science and Technology, in 2010. He worked with Zhejiang Normal University from 2010 to 2021, served as a visiting researcher with the University of Pittsburgh, in 2013, and is currently a professor with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China. His research interests include crowdsensing, federated learning and game theory.

Ahmed M. Abdelmoniem (Senior Member, IEEE) received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2017. He is an associate professor with the Queen Mary University of London, U.K. and leads the Scalable Adaptive Yet Efficient Distributed (SAYED) Systems Research Group. Formerly, he was a research scientist with KAUST, Saudi Arabia, and a senior researcher with Huawei's Future Networks Lab in Hong Kong. He is an investigator on several U.K. and international grants totaling nearly USD 1.5mil in funding. His research interests include the intersection of distributed systems, machine learning, and computer networks. He is a member of ACM and USENIX.

Chen Wang (Senior Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Automation, Wuhan University, China, in 2008 and 2013, respectively. From 2013 to 2017, he was a postdoctoral research fellow in the Networked and Communication Systems Research Lab, Huazhong University of Science and Technology, China. Thereafter, he joined the faculty of Huazhong University of Science and Technology where he is currently an associate professor. His research interests are in the broad areas of wireless networking, Internet of Things, and mobile computing, with a recent focus on privacy issues in wireless and mobile systems. He is a senior member of ACM.