A Practical Clean-Label Backdoor Attack with Limited Information in Vertical Federated Learning

Peng Chen*, Jirui Yang*, Junxiong Lin*, Zhihui Lu*§, Qiang Duan†, and Hongfeng Chai*¶
*School of Computer Science, Fudan University, Shanghai, China
¶FinTech Institute, Fudan University, Shanghai, China
§Shanghai Blockchain Engineering Research Center, Shanghai, China
†Information Sciences & Technology Department, Pennsylvania State University, Abington, PA, USA

2023 IEEE International Conference on Data Mining (ICDM) | 979-8-3503-0788-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDM58522.2023.00013

***Abstract*—Vertical Federated Learning (VFL) facilitates collaboration on model training among multiple parties, each owning partitioned features of the distributed dataset. Although backdoor attacks have been found as one of the main threats to FL security, research on backdoor attacks in VFL is still in the infant stage. Existing methods for VFL backdoor attacks rely on predicting sample pseudo-labels using approaches such as label inference, which require substantial additional information not readily available in practical FL scenarios. To evaluate the practical vulnerability of VFL to backdoor attacks, we present a target-efficient clean backdoor (TECB) attack for VFL. The TECB approach consists of two phases – i) Clean Backdoor Poisoning (CBP) and Target Gradient Alignment (TGA). In the CBP phase, the adversary trains a backdoor trigger and poisons the model during VFL training. The poisoned model is further fine-tuned in the TGA phase to enhance its efficacy in complex multi-classification tasks. Compared to the existing methods, the proposed TECB achieves a highly effective backdoor attack with very limited information about the target class samples, which is more practical in typical VFL settings. Experimental results verify the superior performance of TECB, achieving above 97% attack success rate (ASR) on three widely used datasets (CIFAR10, CIFAR100, and CINIC-10) with only 0.1% of target labels known, which outperforms the state-of-the-art attack methods. This study uncovers the potential backdoor risks in VFL, enabling the development of secure VFL applications in areas like finance, healthcare, and beyond. Source code is available at: https://github.com/13thDayOfLunarMay/TECB-attack**

***Index Terms*—Vertical Federated Learning, Backdoor Attack, Financial Artificial Intelligence Security, Clean Backdoor Poisoning, Target Gradient Alignment.**

I. Introduction

In recent years, data-driven federated learning (FL) [1], [2] has become a new trend in the field of data mining technology. As a privacy-preserving distributed machine learning paradigm, FL enables collaboratively training a model among multiple participants without exposing their local data. It can be classified into Horizontal Federated Learning (HFL), where participants have different data instances sharing the same features, and Vertical Federated Learning (VFL), where the same data samples split feature spaces among participants. VFL holds significant potential in various domains like finance and healthcare, where data features are distributed among collaborating entities [3], [4].

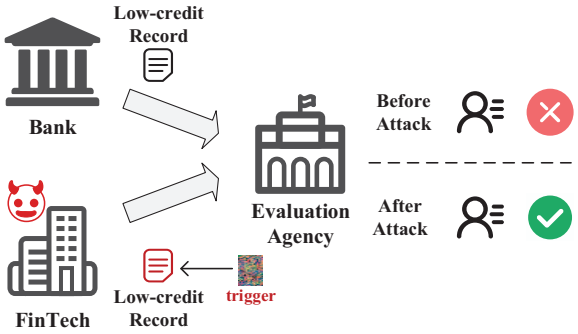


Fig. 1: An example of the backdoor attack in VFL.

However, recent works [5], [6] have illustrated that the FL can be compromised by backdoor attacks. Specifically, a backdoor attack on FL allows an adversary to maliciously craft its local model and thus mislead the global model on trigger-embedded inputs while maintaining the normal behavior for clean inputs.

Fig. 1 illustrates a VFL application scenario susceptible to a backdoor attack [7]. In this scenario, a bank and a FinTech company are involved in VFL for credit analysis, with each entity possessing a subset of users' attributes. An evaluation agency coordinates the VFL model training using labels of users. However, despite the bank's objective of identifying low-credit users, the fintech company has the potential to clandestinely insert a local backdoor to facilitate loan approval for such users, thereby posing a significant security risk to real-world VFL applications.

The inherent security vulnerabilities in FL necessitate a comprehensive understanding of backdoor attacks in order to safeguard FL security. While much of the recent research [5], [6], [8] focuses on HFL, the investigation of backdoor attacks in VFL remains inadequate. The distinct structures of VFL and HFL mean that strategies for addressing backdoor attacks in HFL may not be applicable to VFL. In a VFL setting, the adversary has no access to labels and can only manipulate

Corresponding author: Zhihui Lu, email: lzh@fudan.edu.cn
This work is supported by the National Key Research and Development Program of China (2022YFC3302300, 2021YFC3300600), National Natural Science Foundation of China under Grant (No. 92046024, 92146002, 61873309) and Shanghai Science and Technology Innovation Action Plan Project under Grant (No.22510761000).

979-8-3503-0788-7/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDM58522.2023.00013

their own data and local model. Additionally, the data features and models owned by other participants are not visible to the adversary.

Existing methods for backdoor attacks of VFL, such as the BadVFL [9] and LR-BA [10] methods, suffer from a range of limitations. For instance, they require a portion of samples from each category, which makes them impractical for real-world VFL applications. While the Label Replacement Backdoor (LRB) method [11] mitigates this issue, it relies on local label replacement using specific target data, which can disrupt the sample space and limit the effectiveness of the backdoor attack. Furthermore, these methods use a predefined trigger for backdoor attacks, which lacks a direct connection to the backdoor target and may not align with the training data distribution. As a result, the performance of both the VFL main task and the backdoor task is compromised.

The limitations of current attack methods prompt our main objective: to design an efficient backdoor attack in practical VFL settings with limited label information, achieving high attack effectiveness while minimizing the impact on the VFL main task. To accomplish this, we propose a target-efficient clean backdoor (TECB) attack for VFL. The TECB approach consists of two phases: 1) Clean Backdoor Poisoning (CBP) and 2) Target Gradient Alignment (TGA). In the CBP phase, the adversary locally trains a trigger that contains important features of the target class. This trigger is then injected into the VFL model during the training process. In the TGA phase, the adversary aligns the poisoned data with the target gradient to fine-tune the VFL model, enhancing the attack's effectiveness for complex multi-classification tasks.

Our proposed TECB approach introduces unique advantages that make it a practical approach for backdoor attacks in VFL. One of its key advantages is the removal of label inference in the process of backdoor injection. It only requires label information from less than 0.1% of training samples of the target class, making it more applicable in real VFL scenarios as compared to methods that require extensive auxiliary data for label inference. Additionally, TECB diverges from existing methods that use predefined backdoor triggers. It generates a clean trigger that includes important features of the target class, derived from the very limited set of known target samples. This not only bolsters the effectiveness of the attack but also reduces its impact on the main task of VFL.

Specifically, we make the following contributions in this paper.

- We propose a novel TECB approach for effective backdoor attacks in VFL. TECB can circumvent the label inference attack and only require very limited information about the target label. This makes it more feasible for VFL application scenarios.
- We devise a bi-level optimization process CBP, which combines trigger generation and backdoor injection during VFL model training. This process generates a trigger that contains crucial features of the target class, simultaneously incorporating it into the model within the VFL training process.
- We propose the TGA phase to fine-tune the model after the clean-label backdoor in VFL. This step of TECB can further boost the potency of backdoor attacks, especially for large-scale multi-classification tasks that are typical in practical VFL applications.
- Through extensive experimentation on widely used datasets, we demonstrate the superiority of our proposed approach. The TECB approach requires only 0.1% of the training labels from the target class to achieve a success rate of nearly 97% for backdoor attacks, surpassing existing state-of-the-art methods.

II. Related Work

Research on backdoor attacks in VFL is still in the early stages. The existing attack methods mainly rely on label inference for obtaining as many pseudo-labels as possible on the adversary, which forms the basis for local backdoor poisoning. Depending on the phase of backdoor attacks, the attacks can occur either during the VFL training process or in the inference stage.

Typical methods for backdoor attacks during the VFL training phase include LRB [11], BadVFL [9], and Xuan et al [12]. The LRB method [11] assumes that the adversary has access to a limited number of target samples in the training dataset. In the training process of VFL, the adversary swaps the embedded features of these target samples and the returned gradients from the active party with poisoned sample data. This simulates a backdoor attack by manipulating the target labels. However, since only the local embedded features of the adversary can be swapped, this method creates a mismatch in the sample space, undermining the effectiveness of the backdoor attack. The BadVFL method [9] is based on the assumption that the adversary has knowledge of a small set of labels for each category. Using these labels, the adversary conducts a semi-supervised label inference attack on its bottom model during VFL model training. This allows for predicting all pseudo-labels in the training data. Subsequently, the adversary injects a clean-label backdoor [13] during the VFL training stage based on the inferred target samples. This method requires knowledge of partial samples for each category and is highly dependent on a label inference method called model completion [14]. Xuan et al. [12] infers the target class samples based on the gradient information received during VFL training. Then, the adversary replaces local inputs of inferred target class samples with random clean samples and adds the trigger for the backdoor attack. This method relies heavily on a gradient-based label inference attack and requires one target sample. However, its effectiveness is limited when the top model is complex, restricting its applicability in real-world VFL scenarios, particularly in complex multi-classification tasks.

Representative methods for backdoor attacks in the VFL inference phase include LR-BA [10] and TPGD [15]. Similar to BadVFL [9], LR-BA [10] relies on the model completion label inference method [14] for obtaining pseudo-labels of the training data and then executes the attack through backdoor

representation generation and model fine-tuning. The model completion in LR-BA requires the adversary to know the labels of a certain number of samples from each category, which becomes less feasible in practical VFL applications with a large number of data categories. The Targeted Projected Gradient Descent (TPGD) [15] generates adversarial perturbations for the target class during VFL inference using the adversary's local embedded features. However, this method assumes the adversary's ability to alter the active party's labels, which is not realistic in practical VFL settings; therefore, its usage is constrained to serve as a benchmark for targeted backdoor attacks.

Compared to the aforementioned existing methods for VFL backdoor attacks, the TECB method proposed in this paper only needs label information about a small number of samples from the target class. This approach bypasses the need for label inference and is effective for multi-classification tasks of complex models. Unlike using pre-defined backdoor triggers in the existing methods, TECB generates a trigger containing important features of the target class and then implements clean-label backdoor injection, which mitigates the impact of predefined triggers on the backdoor attack performance. TECB also refines the poisoned model using the TGA phase to enhance attack efficacy to multi-classification tasks. Therefore, TECB offers an efficient and practical solution for VFL backdoor attacks.

III. Formulation and Threat model

In this section, the backdoor formulation and the associated threat model are rigorously defined in the VFL setting.

A. Problem Formulation

In the VFL architecture, K ($K > 2$) participants collaboratively train a model using their respective private data for classification tasks. These participants consist of $K-1$ passive parties and one active party. Each passive party possesses only a portion of the feature data, while the active party can access both feature data and the corresponding labels. Without loss of generality, we assume the K-th participant to be the active party. The dataset for training, denoted by $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$, comprises N samples where the feature vector $x_i = \{x_i^k\}_{k=1}^{K}$ is distributed among K parties. The labels $y_i \in \{1, \cdots, C\}$ are kept in the active party K, where C represents the number of categories in the classification task. The objective of VFL is to collaboratively train a model using participants' local data. The formulation of VFL [16], [17] can be expressed as follows:

$$\min_{\Theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathcal{L}\left(G\left(H^1, \cdots, H^K\right), y\right), \tag{1}$$

where each party adopts a bottom model f_k parameterized by θ_k to compute the local output $H^k = f_k(\theta_k; x^k)$. In addition to the bottom model, the active party also has a top model G parameterized by θ_{top}, which aggregates the local outputs of all parties to minimize the loss function $\mathcal{L}$. $\Theta = \{\theta_1, \cdots, \theta_K; \theta_{top}\}$ represents the overall VFL model parameters.

In the VFL framework, when a passive party assumes the role of an adversary, its objective is to establish a mapping association between a trigger and the backdoor target class. This is achieved by injecting a backdoor task during the VFL training process. Subsequently, upon deployment of the VFL model, the adversary can introduce the local trigger to misclassify the corresponding sample as the target class deliberately. The objective of a backdoor attack in VFL is

$$\min_{\Theta} \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}_c} \mathcal{L}\left(F(x;\Theta), y\right)}_{\text{Main Task}} + \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}_p} \mathcal{L}\left(F(x+\delta;\Theta), \tau\right)}_{\text{Backdoor Task}} \tag{2}$$

where F refers to the VFL model, which encompasses both the top model and the bottom models of all participating parties. $\mathcal{D}_c$ and $\mathcal{D}_p$ denote the clean and poisoned datasets respectively, $\mathcal{D}_c, \mathcal{D}_p \in \mathcal{D}$. δ represents the backdoor trigger and τ is the backdoor target class.

Given that $\mathcal{D}_p$ is randomly sampled from the training dataset, the poisoned data instances represent diverse categories and do not strictly correspond to the target class τ. In addition, as a passive party in VFL, the adversary is unable to alter the labels, rendering the backdoor task in Eq. (2) infeasible in the VFL setting. Hence, this study proposes the TECB approach to enable an effective backdoor attack in VFL without modifying the labels held by the active party. By utilizing a very limited number of samples from the target class as auxiliary information, the proposed approach demonstrates the practicality of the backdoor threat to VFL.

B. Threat model

In this study, we assume that the adversary is a passive party, while all other parties are trusted and act with integrity. This is because the active party has direct control over the labels, making it easy to execute a backdoor attack.

Adversary's capacity. The adversary, while malicious, acts as a passive party in the VFL setting, adhering strictly to the VFL protocol. It doesn't manipulate information from other participants, especially labels controlled by the active party. The adversary's operations are limited to transmitting local features to the active party and receiving gradient information in return.

Adversary's objective. In the VFL multi-classification tasks, the adversary's objective is to inject a backdoor into the model during the training phase. When the poisoned model is deployed for applications, it will misclassify any local data with the backdoor trigger as the designated target class but maintains classification capacity for all other clean data.

Adversary's knowledge. In backdoor attacks, the adversary requires only very few training samples labeled target class. For example, in our CINIC-10 experiments, only 4 target label samples out of 180,000 total training samples were required to successfully carry out the backdoor attack. Although this requirement deviates from the original VFL setting, it is realistic in practical VFL applications because the adversary can acquire a small number of target class samples through various means, such as direct purchasing [10], [14]. Apart from

these limited target samples, the adversary has no information about the models and data of other parties.

IV. Methodology

In this section, we present our proposed TECB approach for the VFL backdoor attack. As illustrated in Fig. 2, the TECB approach consists of two steps. Firstly, the CBP phase locally trains the backdoor trigger and injects it into the model during the VFL training process, using a limited number of target class samples, as detailed in Algorithm 1. Subsequently, the poisoned VFL model is refined using the TGA phase (see Algorithm 2). This refinement step boosts the effectiveness of the VFL backdoor attack in complex multi-classification tasks.

A. Clean Backdoor Poisoning in VFL

In this step of TECB, the adversary locally trains a trigger that includes crucial features of the backdoor target class while training the VFL model. At the same time, the adversary injects the generated trigger into the VFL model. As a result, when samples containing this trigger are used in the VFL model, they are wrongly classified as the target class.

To generate the trigger, we use the PGD method [18], [19] for optimization:

$$\delta_{t+1} = \Pi_\epsilon \left(\delta_t - \alpha \cdot \operatorname{sgn} \left(\nabla_\delta \mathcal{L} \left(F \left(x + \delta_t; \Theta^* \right), \tau \right) \right) \right), \quad (3)$$

where t is the step index, $\nabla_\delta \mathcal{L} \left(F \left(x + \delta_t \right), \tau \right)$ denotes the gradient of the loss function for the backdoor target class with respect to the trigger, α is the step size, and Π_ϵ keeps δ within an ϵ-ball at each step, $F(\Theta^*)$ refers to the pre-trained VFL model. $sgn(\cdot)$ denotes a sign function.

The trigger generation depends on using a limited number of target class τ samples to compute the gradient with respect to the trigger on a pre-trained model $F(\Theta^*)$. Therefore, the adversary requires some target class samples. Furthermore, without an extra label inference attack, the adversary, as a passive party, can only get the gradient of the target class with respect to the trigger δ from the active party during model training, with the intermediate model serving as the pre-trained model $F(\Theta^*)$. This necessitates the incorporation of trigger generation with VFL model training, which can be expressed by the following bi-level optimization:

$$\min_{\Theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\min_{\delta \in \mathcal{C}} \mathcal{L} \left(F(x + \delta; \Theta), y \right) \right] \quad (4)$$

$$\text{s.t.} \quad \mathcal{C} = \{ \delta : \|\delta\|_\infty \leq \epsilon, \delta = 0 \ \forall x \notin \mathcal{D}_t \} \quad (5)$$

where the trigger δ is trained according to all known target samples $\mathcal{D}_t$ available to the adversary, $\mathcal{D}_t \in \mathcal{D}$. In essence, this trigger serves as a universal trigger for the target class τ [20], [21].

In Eq. (4), the internal optimization searches for a backdoor trigger for a given model (intermediate VFL model i.e., Θ^*) while the external minimization of the loss function essentially trains the VFL model parameters Θ with a trigger δ embedded. This bi-level optimization problem in VFL can be efficiently solved by iteratively optimizing these two sub-problems [22]–[24].

Algorithm 1 Clean Backdoor Poisoning in VFL

Require: Training dataset $\mathcal{D}$; number of CBP epochs $\mathcal{T}$; target dataset $\mathcal{D}_t$ with target label τ
Ensure: trigger δ, model parameters Θ
1: Initialize: $\delta \leftarrow 0$.
2: **while** not reached $\mathcal{T}$ **do**
3: **for** each batch B in $\mathcal{D}$ **do**
4: **Adversary** A: updates $\{x_i^A = x_i^A + \delta\}_{i \in \{B \cap \mathcal{D}_t\}}$.
5: **for** each party $k = 1, 2, \ldots, K$ in parallel **do**
6: k computes embedded features $\{H_i^k\}_{i \in B}$ using its bottom model f_k.
7: **end for**
8: **Active party**:
9: computes Eq. (1), then updates θ_{Top} using $\frac{\partial \mathcal{L}}{\partial \theta_{Top}}$.
10: sends $\{\frac{\partial \mathcal{L}}{\partial H_i}\}_{i \in B}$ to all parties.
11: **Adversary** A:
12: computes $\nabla_\delta \mathcal{L}$ with $\{\frac{\partial \mathcal{L}}{\partial H_i} \frac{\partial H_i}{\partial \delta}\}_{i \in \{B \cap \mathcal{D}_t\}}$
13: updates δ with Eq. (3)
14: **for** each party $k = 1, 2, \ldots, K$ in parallel **do**
15: k computes $\nabla_{\theta_k} \mathcal{L} = \{\frac{\partial \mathcal{L}}{\partial H_i} \frac{\partial H_i^k}{\partial \theta_k}\}_{i \in B}$.
16: k updates model parameters θ_k.
17: **end for**
18: **end for**
19: **end while**

Algorithm 1 outlines the procedure of CBP. In each iteration of VFL model training, the adversary examines the current batch B to identify any target class sample in $\mathcal{D}_T$. If such samples are found, the adversary injects the trigger into the corresponding feature vectors of the local dataset (line 4). Subsequently, each party k computes embedded features via the bottom model f_k (line 5-7), where H_i^k represents the embedded features of the i-th data from the k-th party. Upon receiving these embedded features, the active party computes the gradients of the loss function with respect to the top model and embedded features of each party, as described in (1). The top model is then updated, and the gradients of loss with respect to embedded features $\{\frac{\partial \mathcal{L}}{\partial Hi}\}_{i \in B}$ are transmitted to each party (lines 8-10), where H_i denotes the aggregation of the embedded features from all parties of the i-th data. After receiving the gradients, the adversary computes $\nabla_\delta \mathcal{L}$ and updates the trigger using Eq. (3). The current model parameters serve as the pre-trained model $F(\Theta^*)$ (lines 11-13). Lastly, each party updates its bottom model based on the gradients sent by the active party (lines 14-17). Note that if batch B does not contain $\mathcal{D}_t$, i.e., $B \cap \mathcal{D}_t = \emptyset$, the adversary does not perform any trigger update and injection. In this case, Algorithm 1 reverts to the standard VFL model training process.

B. Target Gradient Alignment

Given the clean trigger is generated using just a few samples from the target class and lacks guidance from other categories,

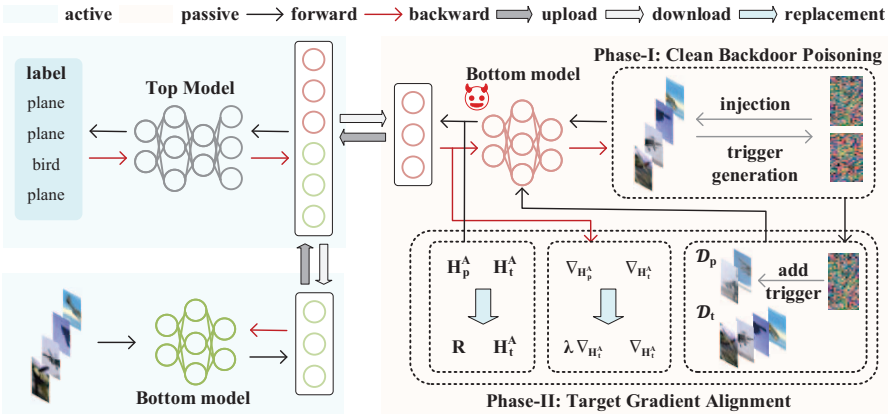


Fig. 2: The overview of TECB approach. (R represents a random vector, H_p^A and H_t^A denote the embedded features of the samples p, t in the poisoned dataset $\mathcal{D}_p$ and the target dataset $\mathcal{D}_t$, respectively. The corresponding gradients from the active party are represented as $\nabla_{H_p^A}$ and $\nabla_{H_t^A}$.)

its effectiveness could be compromised in practical VFL scenarios that involve complex multi-classification tasks. To address this issue, we propose the TGA phase to refine the poisoned model in VFL by incorporating poisoned data from multiple categories. The objective of TGA is expressed as follows:

$$\min_{\Theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_p} \mathcal{L}\left(F(x+\delta;\Theta),\tau\right) \tag{6}$$

where $\mathcal{D}_p$ denotes the poisoned dataset containing multiple categories.

TGA in Eq. (6) incorporates the clean trigger into the poisoned data of multi-categories during the VFL model training, thereby establishing an association with the target class τ. This alignment between backdoor injection during training and the backdoor task during inference can greatly enhance the performance of the backdoor attack.

Considering that the trigger is trained using the target dataset $\mathcal{D}_t$, when added to data samples in $\mathcal{D}_p$, it should make these samples closely resemble the target samples from $\mathcal{D}_t$, thus allowing them to be misclassified as the backdoor target τ [25]. Therefore, when the adversary is unable to manipulate the label, the TGA strategy in Eq. (6) can be achieved by utilizing the gradients of the clean target dataset as a substitute:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_p} \nabla_{H^A} \mathcal{L}\left(x+\delta,\tau\right)) \approx \mathbb{E}_{(x,y)\sim\mathcal{D}_t} \nabla_{H^A} \mathcal{L}\left(x,y\right) \tag{7}$$

where $\nabla_{H^A}\mathcal{L}$ denotes the gradients of loss with respect to the embedded features that the adversary received from the active party, and for brevity, we refer to it as ∇_{H^A}.

The TGA procedure is described in Algorithm 2, which uses the results of the CBP phase (i.e., the trained trigger and model) as inputs to further poison the VFL model. In each training epoch, the adversary checks each batch B to determine the count of known target data samples in the batch, and then randomly selects an equal number of samples from the remaining data in that batch to form the poisoned data, denoted as S. The trigger is then appended to these

Algorithm 2 Target Gradient Alignment in VFL

Require: Training dataset $\mathcal{D}$; trained trigger δ and model Θ; target dataset $\mathcal{D}_t$; Number of TGA epochs T
Ensure: Updated model Θ

1: **while** not reached T **do**
2: **for** each batch B in $\mathcal{D}$ **do**
3: **Adversary** A:
4: Updates $\{x_i^A = x_i^A + \delta\}_{i\in S}$, where $S \subset (B \setminus (B \cap \mathcal{D}_t))$ and $|S| = |B \cap \mathcal{D}_t|$.
5: Records the indices $P =< m, n >$, where m and n are the indices of $B \cap \mathcal{D}_t$ and S in B respectively.
6: **for** each party $k = 1, 2, \ldots, K$ in parallel **do**
7: k computes embedded features $\{H_i^k\}_{i\in B}$ using its bottom model f_k.
8: **end for**
9: **Adversary** A:
10: Sets $\{H_i^A\}_{i\in S}$ to random vectors.
11: **Active party**:
12: Computes Eq. (1), then updates θ_{Top} using $\frac{\partial \mathcal{L}}{\partial \theta_{Top}}$.
13: Sends $\{\frac{\partial \mathcal{L}}{\partial H_i}\}_{i\in B}$ to all parties.
14: **Adversary** A:
15: Replaces $\frac{\partial \mathcal{L}}{\partial H_n}$ with $\lambda \frac{\partial \mathcal{L}}{\partial H_m}$, for all $< m, n > \in P$.
16: **for** each party $k = 1, 2, \ldots, K$ in parallel **do**
17: k computes $\nabla_{\theta_k}\mathcal{L} = \{\frac{\partial \mathcal{L}}{\partial H_i}\frac{\partial H_i^k}{\partial \theta_k}\}_{i\in B}$.
18: k updates model parameters θ_k.
19: **end for**
20: **end for**
21: **end while**

poisoned samples, and the indices of both the target samples and poisoned samples in the same batch are recorded as P (lines 3-5).

Subsequently, every participant in the VFL framework computes embedded features utilizing its bottom models (lines 6-8). The adversary then substitutes the embedded features of

the poisoned samples with random vectors [11] (lines 9-10). This is to ensure that the poisoned samples, when combined with the clean trigger, do not establish an association with their true labels. The active party then calculates the loss using the embedded features received from all parties, updates the top model accordingly, and sends the gradients of the loss with respect to the embedded features back to all parties (lines 11-13). Upon receiving the gradients, the adversary replaces the gradients of the poisoned data with those corresponding to the clean target data, scaled by a factor λ (lines 14-15). This process ensures that the poisoned data are aligned with the clean target class. Finally, all parties update their local models based on the received gradients (lines 16-19). This process guarantees alignment between the backdoor inference and the training poisoning objective, thereby substantially enhancing the performance of the TECB approach.

V. Experiments

A. Experiment Setting

1) Models and Datasets: This paper employs three widely adopted public datasets, namely CIFAR10 [26][1], CIFAR100 [26][2] and CINIC-10 [27] [3], for evaluating the TECB performance. CIFAR10 comprises a training set with 50,000 images and a test set with 10,000 images. Both sets have an equal distribution across 10 categories. Similarly, CIFAR100 has a training set of 50,000 images and a test set of 10,000 images, with the images evenly distributed across 100 categories. CINIC-10 contains 270,000 images, 4.5 times that of CIFAR10. Its training and test sets include 180,000 and 90,000 images, respectively, and like CIFAR10, it has 10 categories that are evenly represented in the dataset. CINIC-10 serves as an expanded version of the CIFAR10 dataset and is utilized to evaluate the algorithm's performance under larger and more complex classification tasks. The input images of all datasets are 32x32 pixels.

In our experiments, we choose to concentrate on a two-party VFL setup for simplicity, as expanding to multiple participants is a simple extension of the two-party VFL. [14], [28]. In our attack scenario, the VFL comprises one passive party, which takes on the role of the adversary, and an active party. The data and model splitting setting follow [9], [10], [14]. Specifically, in all three datasets, each image is bisected along the middle line with each participant retaining one-half. In our TECB approach, the adversary holds half of the image, whereas the active party possesses the other half as well as the labels. Both participants employ a ResNet-18 [29] architecture as the bottom model. The top model on the active party is composed of a 4-layer fully connected neural network, where each layer is supplemented with a Batch Normalization and a ReLU activation function.

In order to ensure a fair comparison with previous studies [9]–[11], we set the backdoor target classes for the CIFAR10, CIFAR100, and CINIC-10 datasets as "airplane," "airplane," and "apple" respectively. In the TECB approach, the backdoor training data consists of four randomly selected samples from the target class($\mathcal{D}_t$), held by the adversary, and four samples randomly selected from the remaining training set($\mathcal{D}_p$). To evaluate the performance of the TECB approach, we construct a strong backdoor task test set by adding the trained trigger to all non-target categories in the test data [21]. Specifically, we select 9000, 9900, and 81000 samples from the CIFAR10, CIFAR100, and CINIC-10 test sets, respectively. This comprehensive test set allows us to assess the ability of TECB to misclassify non-target categories as the backdoor target class.

2) Evaluation Metrics: To evaluate the performance of the TECB approach, we utilize two common metrics: attack success rate (ASR) and main task accuracy (MTA) [30]. ASR measures the percentage of samples in the backdoor test set that is predicted as the target class by the poisoned model. On the other hand, MTA assesses the accuracy of the clean test set on the poisoned model. Notably, the CIFAR10 and CINIC-10 datasets use top-1 accuracy for evaluation, while the CIFAR100 dataset employs top-5 accuracy due to its larger number of categories [14].

3) Compared Methods: To evaluate the efficacy of our proposed TECB approach, we compare it with three types of VFL backdoor methods: i) methods of backdoor attacks in the training phase, namely LRB [11] and BadVFL [9]; ii) methods of backdoor attacks during the VFL inference phase, including TPGD [15] and LR-BA [10]; and iii) VFL baseline method with no attacks (Baseline), which is used as a benchmark reflecting the VFL's inherent performance.

For a fair comparison with the LRA method, we set the number of labeled samples to 4 and the amplification ratio to 10 [10], [11]. For BadVFL, we choose to demonstrate performance based on optimal selection, as BadVFL requires selecting both original and target categories. The TPGD attack is set with a learning rate of 0.5 and 50 epochs for all three datasets. For LR-BA, we adhere to the settings specified in [10].

4) Implementation Details: We adopt the Baseline hyperparameters for the three datasets as suggested in [10], [14]. Specifically, the number of epochs is set to 100. For the CIFAR10 and CIFAR100 datasets, the batch size is 64, and the learning rate is 0.2, while for the CINIC-10 dataset, the batch size is 2048, and the learning rate is 0.8. The minor adjustments relative to [10], [14] are implemented to expedite the model training process. We employ the SGD optimizer with momentum set to 0.9 for all three datasets. Additionally, the learning rate decays by a factor of 0.1 at 50 and 85 epochs with MultiStepLR during the training process. Our experiments are based on the Pytorch framework implemented on two 3090 GPU cards. To ensure the reliability of the experimental results, we conduct five independent tests for all experiments and take the average value as the final result.

For all three datasets, we perform the CBP phase of the TECB approach for the first 50 epochs [31] of VFL training and start the TGA phase since the 80th training epochs. We

[1]https://www.cs.toronto.edu/~kriz/cifar.html
[2]https://www.cs.toronto.edu/~kriz/cifar.html
[3]https://datashare.ed.ac.uk/handle/10283/3192

TABLE I: Performance comparison with state-of-the-art methods on the CIFAR10, CIFAR100 and CINIC-10 datasets. (Best results are highlighted in bold.)

Dataset	MTA						ASR					
	Baseline	**LRB**	**BadVFL**	**TPGD**	**LR-BA**	**TECB**	**Baseline**	**LRB**	**BadVFL**	**TPGD**	**LR-BA**	**TECB**
CIFAR10	**80.73**	74.34	76.00	80.73	79.90	79.56	2.05	3.04	89.00	25.28	98.2	**99.04**
CIFAR100	**79.88**	71.46	67.00	79.88	74.93	78.55	2.48	2.35	81.00	2.21	86.5	**97.09**
CINIC-10	**76.13**	66.55	67.00	76.13	68.84	73.20	1.62	2.46	77.00	12.35	96.78	**98.62**

set the factor λ to 5 for the CIFAR10 and CIFAR100 datasets, whereas, for the CINIC-10 dataset, the amplification factor λ is set to 12. The step size α is set to 0.05, and ϵ is 1.0 for the three datasets.

B. Performance Evaluation and Comparison

The experiment results obtained for the proposed TECB approach and the state-of-the-art backdoor attacks in VFL are listed in Table I. We can see from the table that TECB achieves the highest ASR for all three datasets – 99.04%, 97.09%, and 98.62% for CIFAR10, CIFAR100, and CINIC-10, respectively. The table also shows that TECB maintains an excellent MTA among the tested attack methods for all datasets – only 1.17%, 1.33%, and 2.93% less than the accuracy of the baseline VFL (without backdoor). Therefore, the experiment results verify that the proposed TECB approach is more effective in VFL backdoor attacks than the existing attack methods.

1) Comparison with Backdoor Attacks in the Training Stage: Compared to BadVFL, TECB achieves significant improvement in ASR – 10.04%, 16.09%, and 22.92% higher for CIFAR10, CIFAR100, and CINIC10, respectively. TECB also outperforms BadVFL in MTA with 3.56%, 11.55%, and 6.2% higher accuracy respectively for these three datasets. In addition, BadVFL depends on a model-complete label inference method that requires 4 labeled data from *each* category as an auxiliary dataset. This means that for the CIFAR10, CIFAR100, and CINIC-10 datasets, a total of 40, 400, and 40 labeled data points are needed to cover all classes, as mentioned in reference [14]. In contrast, our TECB approach achieves superior results in backdoor attacks by only requiring 4 labeled samples from the target class.

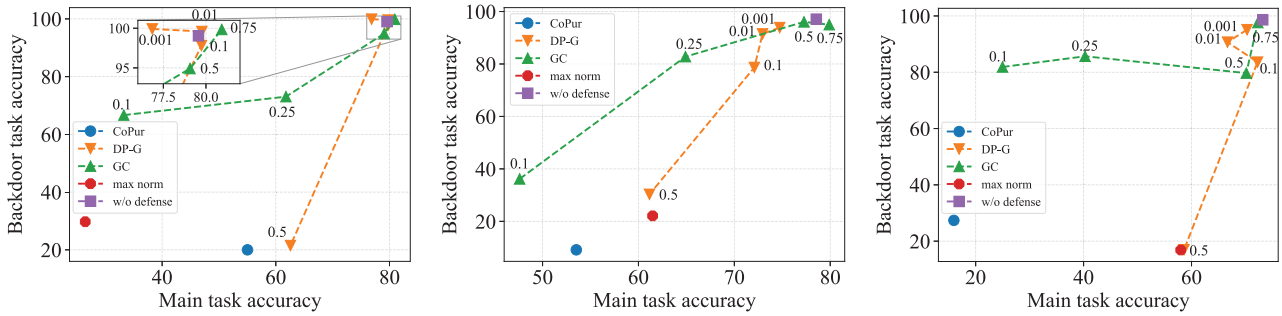
LRB is another benchmark attack in the VFL training stage. While LRB only needs a few target class samples, it results in poor backdoor performance in VFL with complex and large-scale data. Specifically, the local replacement of embedded features in LRB causes feature misalignment within samples, which impacts both ASR and MTA metrics. In contrast, TECB prevents feature misalignment and generates the trigger using samples from the target class, resulting in an efficient backdoor attack that does not compromise the performance of the main task.

2) Comparison with Backdoor Attacks in the Inference Stage: The results in Table I demonstrate that the TPGD method, although maintains the same MTA as the baseline model, suffers low ASR – only 25.28%, 2.21%, and 12.35% on the three datasets, mainly due to its lack of capability in poisoning the VFL model. Furthermore, TPGD assumes an unrealistic VFL setting where the adversary can manipulate the active party's label and obtain gradient information during the inference phase. On the contrary, the TECB approach can inject a backdoor trigger into the VFL model using information from only 4 target samples without label manipulation. TECB significantly enhances ASR – 73.76%, 94.88%, and 86.27% higher than TPGD on the three datasets respectively with only slight reductions (1.17%, 1.33%, and 2.93% for the three datasets) in MTA. These results indicate that TECB can achieve a highly effective backdoor attack in a practical VFL setting with minimal disruption to the main task.

LR-BA, another method of attacking during the inference phase, demonstrates strong performance in VFL backdoor attacks. It achieves ASR metrics of 98.2%, 86.5%, and 96.87% on the CIFAR10, CIFAR100, and CINIC-10 datasets, respectively. However, this method also requires 4 labeled data samples from *each* class, i.e., a total of 40, 400, and 40 labeled training samples respectively for the three datasets, to perform the backdoor attack. Our TECB approach has a significant advantage in terms of auxiliary information requirements – requires only four samples from the target class on these datasets to carry out an efficient backdoor attack. Even with limited information, TECB achieves performance improvements over LR-BA – 0.84%, 10.59%, and 1.84% higher ASR respectively on the three datasets. Remarkably, TECB demonstrates a substantial gain in ASR on the CIFAR100 dataset, highlighting the effectiveness of our TGA method employed in TECB for complex multi-class backdoor attacks, compared to the model completion-based pseudo label prediction scheme of the LR-BA method [14].

3) Attack Efficacy under Defense: To assess the robustness of our TECB approach, we implemented several representative backdoor defense methods in the VFL framework when testing TECB performance. Broadly, these defense methods fall into two categories. The methods in the first category, including DP-G [14], max norm [32], and gradient compression (GC) [14], [33], aim to defend against backdoor attacks by perturbing the gradients sent to the adversary from the active party during the VFL training process. The second category, represented by the CoPur method [15], strives to purify the embedded features sent by the adversary during the VFL inference process for eliminating the poisoned features. In the experiments, we set the noise scale in DP-G to 0.5, 0.1, 0.01, and 0.001 and the compression rate in GC to 0.75, 0.5, 0.25, and 0.1, respectively.

The results in Fig. 3 illustrate the existence of a trade-



(a) CIFAR10 (b) CIFAR100 (c) CINIC-10

Fig. 3: Defence Evaluation on the TECB approach. The lines labeled DP-G and GC display the protection performance against TECB with various noise scales (0.5, 0.1, 0.01, 0.001) and compression rates (0.75, 0.5, 0.25, 0.1), respectively. Meanwhile, the protection performances of the CoPur and max norm methods are depicted as single points. The ideal position for a defense method on the figure is the lower right corner, indicating the highest MTA and lowest ASR.

TABLE II: Results of ablation study on different phases of TECB. (Best results are highlighted in bold.)

Dataset	MTA			ASR		
	TECB-No-GA	**TECB-NO-CLB**	**TECB**	**TECB-No-GA**	**TECB-NO-CLB**	**TECB**
CIFAR-10	74.08 ± 5.19	66.42 ± 15.52	$\mathbf{79.56 \pm 2.06}$	85.58 ± 18.17	27.31 ± 13.64	$\mathbf{99.04 \pm 1.92}$
CIFAR-100	60.24 ± 13.87	77.84 ± 2.26	$\mathbf{78.55 \pm 1.46}$	74.17 ± 21.15	6.51 ± 2.64	$\mathbf{97.09 \pm 3.58}$
CINIC-10	70.35 ± 3.58	$\mathbf{75.83 \pm 0.32}$	73.20 ± 1.17	81.02 ± 20.43	2.99 ± 0.31	$\mathbf{98.62 \pm 1.88}$

off between model utility and backdoor protection in VFL. While different defense strategies can partially diminish the impact of backdoor tasks, they simultaneously deteriorate the performance of the main task. From our experiments on three datasets, DP-G provides a relatively effective defense that reduces the backdoor efficacy of all three datasets at a noise scale of 0.5, but at the cost of much-reduced accuracy for the main task. Similarly, CoPur causes significant degradation in main task performance in order to provide an effective defense against backdoor attacks. Therefore, the obtained results demonstrate that typical defense methods cannot easily mitigate the TECB attack without severe sacrifice in main task performance, which verifies the robustness of the TECB approach for resisting VFL defense.

C. Ablation Study

1) Impact of Each Phase in TECB: The TECB comprises two phases – first the CBP phase and then the TGA phase. We conduct ablation studies to evaluate the impact of each phase on the overall performance of the TECB approach. TECB-No-GA represents the exclusive use of the CBP phase while TECB-NO-CLB entails solely using the TGA phase where a 9-pixel pattern [34] serves as the trigger to align the model with the target gradient.

Table II presents the experimental results of different phase combinations, comprising the mean and variance from five independent trials. In these experiments, TECB-No-GA achieves ASR metrics of 85.58%, 74.17%, and 81.02% on CIFAR10, CIFAR100, and CINIC-10 datasets respectively with high variances (18.17%, 21.15%, and 20.43% on the three datasets). The results demonstrate that while using CBP alone may achieve a certain level of attack efficacy, the attack is not robust, especially in complex multi-classification scenarios such as the CIFAR100 dataset, where the variance in ASR is as high as 21.15%. The poor ASR results of TECB-NO-CLB on all three datasets indicate that the TGA phase alone is not sufficient in achieving an effective backdoor attack.

Both TECB-No-GA and TECB-NO-CLB exhibit varying degrees of impact on the main task performance. The performance decline in TECB-No-GA could be because CBP might skew the model towards the target class, causing a bias that compromises classification for other categories. This issue becomes severe in complex multi-classification tasks like CIFAR100, where we observed a significant 29.64% decrease in MTA. On the other hand, the decreased performance of TECB-NO-CLB may be attributed to a mismatch between the distribution of the 9-pixel pattern trigger and the training data. This mismatch could potentially disrupt the model training process. The impact of this issue is especially prominent in the CIFAR10 dataset, which has fewer data points and categories, making it less resilient to noise [35].

The above results confirm the suitability of the two-phase design of TECB, which combines CBP and TGA in the attack method, to achieve a successful backdoor attack in VFL.

2) Impact of Combining the CBP and TGA phases: TECB performs the CBP and TGA phases in two steps. To evaluate the impact of the sequential combination of these two phases on TECB performance, we conducted a comparative study of two operation configurations of the TECB approach. In the first

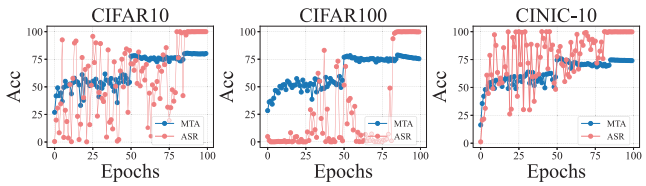


Fig. 4: Performance of TECB during VFL training with CBP in the first 50 epochs and TGA in 80-100 epochs.

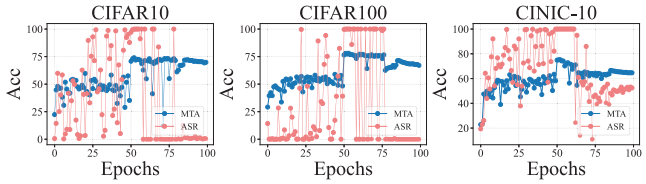


Fig. 5: Performance of TECB during VFL training with CBP in the first 50 epochs and TGA in 50-100 epochs.

configuration, TECB performs CBP for the first 50 epochs of the VFL model training and then runs TGA during the 80-100th epochs (till the end of model training). In the other configuration, TECB runs CBP for the first 50 epochs and then switches to TGA immediately after completing the 50th epoch. The obtained results of ASR and MTA varying with the number of training epochs for these two configurations are plotted in Fig. 4 and 5.

Fig. 4 demonstrates that in the first configuration, the main task and the backdoor task both converge stably to their optimal results by the end of the VFL training. On the contrary, as shown in Fig. 5, when CBP runs in the initial 50 epochs and TGA takes over for the subsequent 50 epochs, although the backdoor attack can achieve optimal results across all three datasets by the end of the CBP phase, the backdoor task's performance declines significantly or even completely disappears as the main VFL task converges.

The experimental results imply that configurations of the combination of CBP and TGA phases may impact the overall TECB performance. The configuration of using CBP for the first 50 epochs and running TGA for the last 20 epochs of the training process allows TECB to achieve a robust and efficient VFL backdoor attack. The choice of employing the CBP phase in the initial 50 epochs takes advantage of the fact that gradients can leak more information at the early stage of model training, facilitating the generation of the backdoor trigger [31]. In the TGA phase, randomly selected poisoned data may unexpectedly align with the target class label. This circumstance, similar to adversarial training [20], [36], can lead to a decline or total elimination of backdoor attack effectiveness with more TGA training rounds.

D. Analysis of Hyperparameter in TECB

1) Setting of target dataset: The TECB approach requires a set of auxiliary target sample labels for backdoor attacks, denoted as $\mathcal{D}_t$. The size of $\mathcal{D}_t$ is a crucial hyperparameter

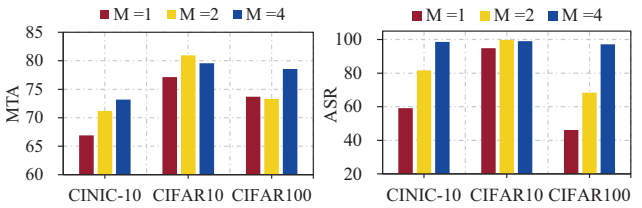


Fig. 6: Comparison of different numbers of target samples known to the adversary in the TECB approach. M is the number of target dataset $\mathcal{D}_t$

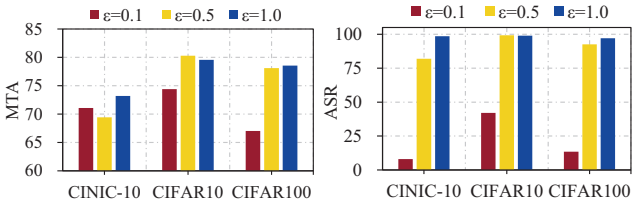


Fig. 7: Comparison of different perturbation ϵ in the TECB approach.

influencing both the effectiveness of TECB and its realistic threat. We measured the TECB performance (ASR and MTA) on three datasets with different sizes of $\mathcal{D}_t$ and the obtained results are plotted in Fig. 6. This figure illustrates that our TECB approach achieves impressive backdoor attack performance with very limited auxiliary target data. TECB performs very well on the CIFAR10 dataset even with only a single sample known from the target class. For more complex multi-class datasets like CIFAR100, TECB yields good ASR and MTA performance with just four samples from the target class. The results in Fig. 6 verify that TECB can realize an effective backdoor attack in VFL with very limited information about target samples and therefore offers an efficient attack in practical VFL scenarios.

2) Setting of perturbation ϵ: Trigger perturbation in backdoor attacks is commonly restrained, e.g., to 16/255, to make it undetectable to the human being [37]. In the VFL architecture, the trigger locally injected by the adversary as a passive party is undetectable to other parties. Nevertheless, examining how the trigger perturbation range impacts TECB performance offers useful insights into this attack.

Fig. 7 presents TECB's performance with perturbations (ϵ) of 0.1, 0.5, and 1.0. TECB performs well with ϵ at 0.5 on all datasets, showing effective backdoor concealment in VFL even without ensuring the trigger's human imperceptibility. In our experiments, we use ϵ of 1.0 for optimal backdoor attack performance. We observed that smaller ϵ values negatively impact the main task. This could be because a smaller trigger fails to effectively capture the target class's important features, degrading VFL's main task performance in the TGA phase.

VI. Conclusion

In this study, we propose the target-efficient clean backdoor (TECB) attack as a practical approach for backdoor attacks in VFL. The TECB approach uses very limited samples from the target class to implement CBP locally, addressing the issue of the adversary's inability to access labels in practical VFL settings. The TECB approach generates a trigger that embodies critical features of the target class, which is used to poison the model during the VFL training process. Furthermore, TECB conducts TGA to enable efficient backdoor attacks on complex multi-classification tasks. Experimental results have verified the efficiency of the proposed TECB attack. The findings of this study mark a significant stride towards unveiling the risk of hidden backdoors in VFL and paving the way for the future development of secure VFL. In the future, we plan to conduct theoretical analyses and explore effective defense mechanisms to address the vulnerabilities of VFL to backdoor attacks.

References

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

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

[3] O. Li, J. Sun, X. Yang, W. Gao, H. Zhang, J. Xie, V. Smith, and C. Wang, "Label leakage and protection in two-party split learning," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=cOtBRgsf2fO

[4] P. Wei, H. Dou, S. Liu, R. Tang, L. Liu, L. Wang, and B. Zheng, "Fedads: A benchmark for privacy-preserving cvr estimation with vertical federated learning," *arXiv preprint arXiv:2305.08328*, 2023.

[5] L. Haoyang, Q. Ye, H. Hu, J. Li, L. Wang, C. Fang, and J. Shi, "3dfed: Adaptive and extensible framework for covert backdoor attack in federated learning," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2023, pp. 1893–1907.

[6] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 070–16 084, 2020.

[7] J. Chen, G. Huang, H. Zheng, S. Yu, W. Jiang, and C. Cui, "Graph-fraudster: Adversarial attacks on graph neural network-based vertical federated learning," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 2, pp. 492–506, 2022.

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

[9] M. Naseri, Y. Han, and E. De Cristofaro, "Badvfl: Backdoor attacks in vertical federated learning," *arXiv preprint arXiv:2304.08847*, 2023.

[10] Y. Gu and Y. Bai, "Lr-ba: Backdoor attack against vertical federated learning using local latent representations," *Computers & Security*, vol. 129, p. 103193, 2023.

[11] T. Zou, Y. Liu, Y. Kang, W. Liu, Y. He, Z. Yi, Q. Yang, and Y.-Q. Zhang, "Defending batch-level label inference and replacement attacks in vertical federated learning," *IEEE Transactions on Big Data*, pp. 1–12, 2022.

[12] Y. Xuan, X. Chen, Z. Zhao, B. Tang, and Y. Dong, "Practical and general backdoor attacks against vertical federated learning," *arXiv preprint arXiv:2306.10746*, 2023.

[13] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, "Clean-label backdoor attacks on video recognition models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 443–14 452.

[14] C. Fu, X. Zhang, S. Ji, J. Chen, J. Wu, S. Guo, J. Zhou, A. X. Liu, and T. Wang, "Label inference attacks against vertical federated learning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1397–1414.

[15] J. Liu, C. Xie, S. Koyejo, and B. Li, "Copur: Certifiably robust collaborative inference via feature purification," *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 645–26 657, 2022.

[16] Y. Liu, X. Zhang, Y. Kang, L. Li, T. Chen, M. Hong, and Q. Yang, "Fedbcd: A communication-efficient collaborative learning framework for distributed features," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4277–4290, 2022.

[17] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, "Vertical federated learning," *arXiv preprint arXiv:2211.12814*, 2022.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb

[19] H. Huang, X. Ma, S. M. Erfani, J. Bailey, and Y. Wang, "Unlearnable examples: Making personal data unexploitable," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=iAmZUo0DxC0

[20] A. Shafahi, M. Najibi, Z. Xu, J. Dickerson, L. S. Davis, and T. Goldstein, "Universal adversarial training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5636–5643.

[21] Y. Zeng, M. Pan, H. A. Just, L. Lyu, M. Qiu, and R. Jia, "Narcissus: A practical clean-label backdoor attack with limited information," *arXiv preprint arXiv:2204.05255*, 2022.

[22] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S.-T. Xia, and B. Li, "Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=kcQiIrvA_nz

[23] R. Liu, J. Gao, J. Zhang, D. Meng, and Z. Lin, "Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 10 045–10 067, 2021.

[24] Y. Zhang, M. Qiu, and H. Gao, "Communication-efficient stochastic gradient descent ascent with momentum algorithms," in *IJCAI*, 2023.

[25] J. Geiping, L. H. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein, "Witches' brew: Industrial scale data poisoning via gradient matching," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=01olnfLIbD

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

[27] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "Cinic-10 is not imagenet or cifar-10," *arXiv preprint arXiv:1810.03505*, 2018.

[28] D. Yao, L. Xiang, H. Xu, H. Ye, and Y. Chen, "Privacy-preserving split learning via patch shuffling over transformers," in *2022 IEEE International Conference on Data Mining (ICDM)*, 2022, pp. 638–647.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[31] W. Wei and L. Liu, "Gradient leakage attack resilient deep learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 303–316, 2021.

[32] O. Li, J. Sun, X. Yang, W. Gao, H. Zhang, J. Xie, V. Smith, and C. Wang, "Label leakage and protection in two-party split learning," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=cOtBRgsf2fO

[33] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[34] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," in *Usenix Security*, 2021.

[35] W. Wei and L. Liu, "Gradient leakage attack resilient deep learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 303–316, 2021.

[36] C.-H. Weng, Y.-T. Lee, and S.-H. B. Wu, "On the trade-off between adversarial and backdoor robustness," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 973–11 983, 2020.

[37] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.