VFLIP: A Backdoor Defense for Vertical Federated Learning via Identification and Purification

Yungi Cho[1], Woorim Han[1], Miseon Yu[1], Younghan Lee[1], Ho Bae[2(✉)], and Yunheung Paek[1(✉)]

[1] Department of ECE and ISRC, Seoul National University, Seoul 08826, Republic of Korea
{q1w1ert1,rimwoo98,altjs543,201younghanlee,krypaek}@snu.ac.kr
[2] Department of Cyber Security, Ewha Womans University, Seoul 03760, Republic of Korea
hobae@ewha.ac.kr
http://sor.snu.ac.kr , https://www.spai.co.kr/

Abstract. Vertical Federated Learning (VFL) focuses on handling vertically partitioned data over FL participants. Recent studies have discovered a significant vulnerability in VFL to *backdoor attacks* which specifically target the distinct characteristics of VFL. Therefore, these attacks may neutralize existing defense mechanisms designed primarily for Horizontal Federated Learning (HFL) and deep neural networks. In this paper, we present the first backdoor defense, called *VFLIP*, specialized for VFL. VFLIP employs the *identification* and *purification* techniques that operate at the inference stage, consequently improving the robustness against backdoor attacks to a great extent. VFLIP first identifies backdoor-triggered embeddings by adopting a participant-wise anomaly detection approach. Subsequently, VFLIP conducts purification which removes the embeddings identified as malicious and reconstructs all the embeddings based on the remaining embeddings. We conduct extensive experiments on CIFAR10, CINIC10, Imagenette, NUS-WIDE, and Bank-Marketing to demonstrate that VFLIP can effectively mitigate backdoor attacks in VFL. https://github.com/blingcho/VFLIP-esorics24

Keywords: Vertical Federated Learning · Backdoor Attack · AI Security

1 Introduction

Federated learning (FL) is a privacy-preserving machine learning framework that enables multiple participants to collaboratively train a model without directly sharing their private data. Instead, participants exchange local computations, such as model weights, gradients, and embeddings. FL can be categorized into two types based on the distribution of data among participants: Horizontal Federated Learning (HFL) and Vertical Federated Learning (VFL) [20]. In HFL,

J. Garcia-Alfaro et al. (Eds.): ESORICS 2024, LNCS 14985, pp. 291–312, 2024.
https://doi.org/10.1007/978-3-031-70903-6_15

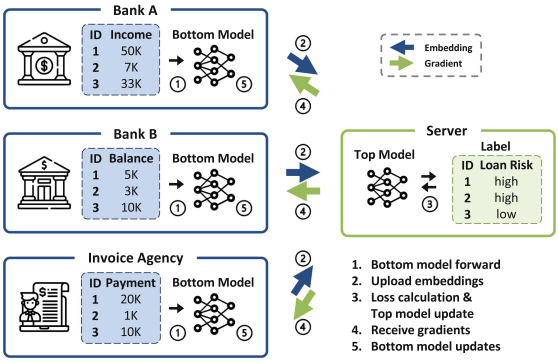


Fig. 1. An illustration of VFL with the split neural network

each participant has distinct sets of samples that share the same features. However, in VFL, participants handle the same samples, but each possesses a unique subset of features for these samples. In sectors where data privacy is of utmost importance, such as finance and healthcare, it is common for subsets of features to be distributed across multiple organizations [13]. In such situations, VFL offers a compelling solution by employing a split neural network architecture [1,6,15,30]. As illustrated in Fig. 1, each participant operates a *bottom model* tailored to its unique subset of features. The subsets are disjoint across participants yet pertain to the same set of samples (e.g., IDs 1,2,3). Instead of sharing the raw features with sensitive information, participants compute and share *embeddings* derived from the bottom model. The central server hosts a *top model*, which uses the embeddings to infer the labels of the samples.

FL confronts a range of security threats arising from the involvement of unreliable or malicious participants who deviate from the majority's intention [18,25,28,30]. During the training stage, some participants may send malicious local computations to the server to manipulate the model's behavior. Relatively much work has been done to reduce the impact of such malicious computations in HFL [2,5,21]. However, little has been conducted to defend against malicious participants in VFL. One of the most well-known security threats posed by the malicious participants in VFL is the *backdoor attack* [1,30], where an attacker subtly manipulates the training data by planting a backdoor trigger during the model's training stage. The trigger is carefully designed to alter the predictions to a target label of the attacker's choice during inference. Sadly, we have discovered that the unique characteristics inherent in VFL make it difficult to apply the HFL defense mechanisms, which deal with computations derived from samples that share the same features [2,23]. Moreover, these challenges cannot be effectively mitigated by existing backdoor defense mechanisms designed for DNNs [1,16,17,19,29]. Consequently, there is an urgent need for a defense mechanism tailored to countering backdoor attacks specialized for VFL.

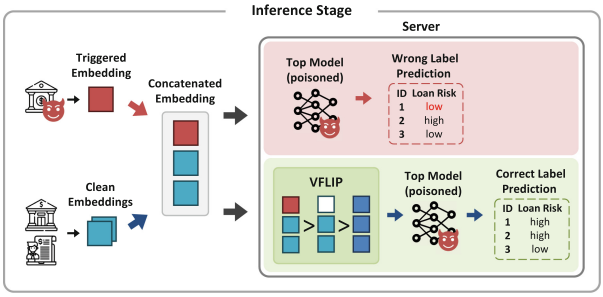


Fig. 2. A brief summary of VFLIP. VFLIP identifies the backdoor-triggered embeddings and purifies all the embeddings through removal and reconstruction.

The first challenge **(C1)** in designing the defense mechanism for VFL is that the server aggregates *embeddings* from the participants to predict the label of a given sample in the inference stage [20]. This unique aspect of the split neural network architecture inherent in VFL introduces a new attack surface that can be exploited by attackers. By exploiting the newly exposed attack surface, a recent work [1] proposed a novel backdoor attack with *embedding-level* backdoor triggers. They demonstrated that such attacks can be both effective and stealthy, creating a robust backdoor trigger that manipulates the model's prediction to a targeted label. The second challenge **(C2)** stems from the unique nature of vertically partitioned feature configuration in VFL. Such configuration significantly complicates the process of detecting attackers among participants. In HFL, each local update (e.g., model weights or gradients) is computed from data with the same feature space. This allows direct comparison of local updates to identify attackers who deviate from the majority of participants [2,5,23]. In contrast, VFL presents a different setting where each participant only possesses a subset of features. Each embedding computed from a different feature subset is difficult to compare. Thus, it is necessary to design a new approach to identify malicious embeddings, which differ from most embeddings.

To address these challenges, we propose a backdoor defense for VFL via Identification and Purification (VFLIP) at the inference stage, which employs a Masked Auto-Encoder (MAE) as a key component. VFLIP consists of two phases: identification and purification. In the **identification** phase, we introduce a participant-wise anomaly detection method with the majority voting to resolve the above challenges **(C1-2)**. This approach is inspired by the attack strategy of the previous study [1,30]. In these studies, as the participants lack the ability to manipulate the labels on the server, a two-fold approach is employed. During training, the trigger is injected into samples with the target label establishing a connection between the trigger and the target label. In contrast, for inducing misclassification during inference, the trigger is introduced into samples associated with non-target labels. Given the inter-participant correlations within the sample, as highlighted in previous studies [8,15,18], the embedding correlation

among participants during training inevitably diverges from that observed during the inference stage. Such discrepancy allows the training of MAE to incorporate an anomaly detection approach during the inference stage, facilitating the identification of abnormal relations caused by malicious participants. Thus, VFLIP identifies the embeddings that show abnormal relationships from most other embeddings. Subsequently, we conduct the **purification** phase using the MAE to minimize the impact of the backdoor-triggered embeddings. This process removes the identified backdoor-triggered embeddings and reconstructs all of the embeddings based on the remaining embeddings.

Capitalizing on the denoising capability of the MAE [27], VFLIP adeptly reconstructs all the embeddings, minimizing the influence of malicious embeddings with stealthy triggers. Figure 2 presents a brief summary of VFLIP, showing that even when an attacker attempts to manipulate the model prediction by providing a backdoor-triggered embedding, the VFL model with VFLIP can predict the correct label. Our main contributions are as follows.

1. We propose VFLIP, a simple yet powerful method for defending against backdoor attacks in VFL, which conducts participant-wise anomaly detection with majority voting. To the best of our knowledge, this is the first study to defend against the backdoor attacks specialized for VFL with split architecture.
2. We conduct extensive experiments on CIFAR10, CINIC10, Imagenette, NUS-WIDE, and Bank-Marketing. This demonstrates that VFLIP effectively defends against the state-of-the-art attack methods by reducing the attack success rate from 84.4% to 7.57% on average.
3. We design an adaptive attack strategy for compromising the VFLIP's MAE. Through this, we demonstrate that it is hard for the attackers to compromise the MAE without significantly decreasing their attack performance.

2 Preliminaries

2.1 Vertical Federated Learning

The fundamental concept of VFL with a split neural network is dividing the model into two parts: bottom models that take local data as inputs and produce embeddings, and a top model that makes a final decision based on the embeddings from the participants [1,6]. Each participant has a bottom model and a subset of the joint features, while the server holds the top model and the labels. Following the previous VFL setup [1,6], we suppose that there are N participants and a server, with the collaborative goal of training a model and subsequently performing inference on a sample using the trained model. VFL model training is conducted on a dataset $\mathcal{D} = \{(x^k, y^k)\}_{k=1}^{K}$ where x represents a joint data sample, y is the corresponding label, K is the total number of data samples, and k is the index for each data sample. In the feature-partitioned environment, a joint data sample can be expressed as $x = [x_1, \cdots, x_n]$. The i-th participant holds a vertically partitioned local dataset, denoted as $\mathcal{D}_i = \{x_i^k\}_{k=1}^{K}$. The i-th participant's bottom model B_i maps local data x_i to the embedding h_i. For

simplicity, the parameters of the bottom models are denoted as $\theta_{B_1,\ldots,B_N}$. The server owns the top model $T(h_1, \cdots, h_N)$ parameterized as θ_T. We denote the loss function of VFL as $\mathcal{L}$. The objective function can be formulated as follows:

$$\underset{\theta_{\mathrm{B}_1,\ldots,\mathrm{B}_N},\theta_{\mathrm{T}}}{\operatorname{argmin}} \sum_{k=1}^{K} \mathcal{L}(T([B_1(x_1^k), \cdots, B_N(x_N^k)]), y^k) \tag{1}$$

The training stage for VFL consists of five main steps: 1) **Batch index selection**: The server selects indices, denoted as idx, from $\mathcal{D}$ and shares it with the VFL participants; 2) **Bottom model forward pass**: Each participant computes their embeddings h_i^{idx} with $B_i(x_i^{idx})$ and sends it to the server; 3) **Top model forward pass**: The server concatenates all embeddings from participants corresponding to idx and computes model prediction through $T(h_1^{idx}, \cdots, h_N^{idx})$; 4) **Top model backward propagation**: The server calculates the loss with labels. Using the loss, the server computes the gradients of the top model and updates it. Afterward, the server sends back the gradients associated with the participants' embeddings; 5) **Bottom model backward propagation**: Each participant performs backward propagation using the gradients received from the server and updates their bottom models. This process is repeated during the training stage. For the inference stage, steps 1 to 3 are executed, and in step 1, idx is chosen from the test set, not the training set $\mathcal{D}$. We note that VFLIP can be applied in scenarios where the server possesses features and participates in training with their own bottom model. This paper focuses on scenarios where the server cannot access the features of the training data.

2.2 Backdoor Attacks in VFL

The objective of a backdoor attack is to manipulate the model so that it correctly predicts clean samples but misclassifies backdoor-triggered samples as the target label [1,9,30]. Depending on whether the attacker can manipulate the labels of the training set, backdoor attacks can be divided into clean-label attacks and dirty-label attacks. The clean-label backdoor attack injects a trigger only into the samples of the target label [12,24]. On the other hand, the dirty-label attack injects a trigger into the samples of non-target labels and manipulates the labels to the target label [9]. Since the labels on the server cannot be manipulated by the attackers in VFL, it is only susceptible to clean-label backdoor attacks [1,30].

Recent studies have proposed backdoor attacks tailored for VFL. Xuan *et al.* [30] propose BadVFL, a data-level backdoor attack where the attacker plants backdoor triggers in their local data to manipulate the top model. To strengthen the connection between the trigger and the target label, the attacker replaces the local data of the target label with that of non-target labels before injecting a trigger. Bai *et al.* [1] introduce VILLAIN, which proposes an additive backdoor trigger on the embedding level, aiming for a stealthy backdoor attack. By adjusting the magnitude of this trigger, the attacker can control the trade-off between stealthiness and attack power. The details of the attack methods are provided in Appendix A.1.

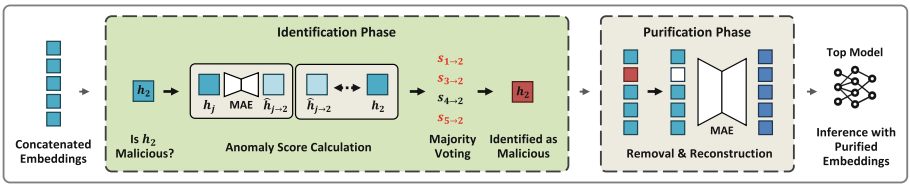


Fig. 3. An overview of VFLIP. VFLIP calculates the anomaly scores for each embedding with the MAE. The voting mechanism is conducted based on the anomaly scores to determine whether an embedding is malicious. Embeddings identified as malicious are removed, and then all the embedding is reconstructed through MAE.

2.3 Threat Model

We follow the threat model of previous studies [1,30], but introduce a few modifications to consider strong attackers. There are one or more attackers among the VFL participants, but the number of attackers is limited to less than half of the total number of participants based on previous FL studies [2,18]. The attacker can modify their local data or embedding. This potentially allows the attacker to poison a substantial portion of the training set. However, they cannot modify any operation on the server or the benign participants. The attacker either has or does not have the label knowledge for their training set. The attacker without the label knowledge owns a small amount of labeled auxiliary data to infer labels about their training set [1,6,30]. We note that even if the attacker knows the labels for the training set, they cannot change the labels in the server. The defender is the server that owns the top model and the labels. The server cannot access participants' operations.

3 Method

This section introduces VFLIP, a novel backdoor defense method for VFL. The primary objective of VFLIP is to diminish the influence of attackers who upload a backdoor-triggered embedding during the inference stage. VFLIP achieves this by identifying the backdoor-triggered embedding and purifying the concatenated embedding before feeding them to the top model. Leveraging the capability of the MAE for both tasks, VFLIP exhibits a mechanism illustrated in Fig. 3.

Identification. Using the discrepancy between the embedding correlation of training and inference stage made by conducting backdoor attacks in VFL, VFLIP detects the abnormal relationships between the embeddings of participants for each sample that are not observed during training. Here, the anomaly detection approach is applied by training the MAE with embeddings from the training stage. Subsequently, during the inference stage, the trained MAE is used to identify the backdoor-triggered embeddings that exhibit abnormal relationships with the majority of other embeddings.

Algorithm 1: VFLIP MAE Training

Input : Masked Auto-Encoder MAE, Training set $\mathcal{H}^{\text{train}}$, Training epoch E_{mae}, Learning rates β_1 (for "N-1 to 1") and β_2 (for "1 to 1"), Masks m filled with 1 for each local embedding, Number of participants N

Output: Trained MAE, Threshold t for the anomaly score

Initialize weights and bias for MAE
for *each train epoch* $e = 1, 2, ..., E_{\text{mae}}$ **do**
  for *each minibatch* B *from* $\mathcal{H}^{\text{train}}$ **do**
    `/* "N-1 to 1" strategy */`
    $\mathcal{L}_{\text{N-1}} \leftarrow 0$
    for *each embedding* h *from* B **do**
      $h_i \leftarrow$ randomly draw one local embedding from h
      $m_i \leftarrow$ Mask filled with 1 for h_i
      $\tilde{m}_i \leftarrow 1 - m_i$
      $\hat{h} \leftarrow \text{MAE}(\tilde{m}_i \odot h)$
      $\mathcal{L}_{\text{N-1}} \leftarrow \mathcal{L}_{\text{N-1}} + ||m_i \odot (h - \hat{h})||_2$
    Update MAE to minimize $\mathcal{L}_{\text{N-1}}$ with learning rate β_1
    `/* "1 to 1" strategy */`
    $\mathcal{L}_1 \leftarrow 0$
    for *each embedding* h *from* B **do**
      $h_i, h_j \leftarrow$ randomly draw two local embeddings from h
      $m_i, m_j \leftarrow$ Mask filled with 1 for h_i, h_j
      $\hat{h} \leftarrow \text{MAE}(m_j \odot h)$
      $\mathcal{L}_1 \leftarrow \mathcal{L}_1 + ||m_i \odot (h - \hat{h})||_2$
    Update MAE to minimize $\mathcal{L}_1$ with learning rate β_2
Compute the thresholds $t = [t_1, ..., t_N]$ for each participant over the $\mathcal{H}^{\text{train}}$
return Trained MAE, Threshold t

Purification. The purification phase aims to mitigate the impact of the backdoor-triggered embedding by leveraging the identification results. To achieve this goal, VFLIP removes the embeddings identified as backdoor-triggered. Subsequently, VFLIP feeds the remaining embeddings to the MAE and reconstructs all the embeddings. Leveraging the denoising capability of the MAE [27], this process is particularly effective in alleviating the influence of malicious participants employing small-magnitude triggers.

The following subsections describe the MAE training and provide details about the identification and purification phase that utilizes the MAE.

3.1 MAE Training

Initially, VFLIP trains the MAE, parameterized as θ_{MAE}, with the poisoned training set after the training stage of VFL. The training set for the MAE is

embeddings gathered from the last epoch of the VFL training stage, denoted as $\mathcal{H}^{\text{train}}$. The input of the MAE is the concatenated embedding, denoted as $h = [h_1, \cdots, h_N]$. The MAE outputs the reconstructed concatenated embedding, which is represented as $\hat{h} = [\hat{h}_1, \cdots, \hat{h}_N]$. The architecture of MAE is composed of an encoder and a decoder. Both of them use a fully connected network.

Training Strategies. VFLIP employs two MAE training strategies: "N-1 to 1" and "1 to 1". The "N-1 to 1" strategy randomly chooses one embedding h_i to be restored (Line 4-6 in Algorithm 1). The selected h_i is masked from h, and MAE reconstructs $\hat{h}_i$ using the masked h (Line 7-9 in Algorithm 1). Next, the "1 to 1" strategy randomly selects two embeddings, denoted as h_i and h_j (Line 14 in Algorithm 1). h_i is the target to restore, and h_j is used as the input for restoration. Here, all the embeddings except h_j are masked from h, and MAE reconstructs $\hat{h}_i$ using the masked h (Line 15-17 in Algorithm 1).

Following the above strategies, the loss is calculated only for the selected embedding h_i (Line 10, 17 in Algorithm 1). MAE is trained by alternately optimizing the loss for the two strategies (Line 11, 18 in Algorithm 1). The objective functions for each strategy are as follows:

$$\underset{\theta_{\text{MAE}}}{\text{argmin}} \; ||m_i \odot (h - \text{MAE}(\tilde{m}_i \odot h))||_2 \tag{2}$$

$$\underset{\theta_{\text{MAE}}}{\text{argmin}} \; ||m_i \odot (h - \text{MAE}(m_j \odot h)))||_2 \tag{3}$$

Here, m_i represents a masking value where only the part corresponding to h_i is filled with 1, while the rests are filled with 0. $\tilde{m}_i$ represents the opposite of m_i, where 0 s and 1 s are reversed. Equation 2 and Eq. 3 are for "N-1 to 1" and "1 to 1" strategy, respectively. The ablation study for these training strategies is provided in Subsect. 4.5.

Standardization and Drop-Out. To enhance the MAE performance, VFLIP employs standardization and drop-out. Standardization is a data preprocessing technique where, instead of directly using h as input, MAE uses standardized h. VFLIP's standardization is based on the mean and standard deviation of $\mathcal{H}^{\text{train}}$. Additionally, drop-out is used for data augmentation. During training, randomly generated masks partially remove h to prevent MAE from overfitting to specific dimensions.

3.2 VFLIP Mechanism

Identification. In the identification phase, VFLIP conducts the participant-wise anomaly detection with majority voting. In this process, anomaly scores for one participant's embedding are calculated from other participants' embeddings. Then, based on these anomaly scores, the participant's embedding is determined whether it is malicious or benign through majority voting. This process is conducted for each participant's embedding.

Algorithm 2: VFLIP mechanism

Input : Concatenated embedding h, Trained Masked Auto-Encoder MAE, Masks m filled with 1 for each embedding, Number of participants N, Thresholds $\{t_i\}_{i=1}^{N}$ for the anomaly score, VFL top model T

Output: VFL model prediction P

1 **Initialize** $votes[N]$ with 0
`/* votes is an array for counting votes */`
2 **Initialize** m_{mal} with $[0, ..., 0]$
`/*` m_{mal} `is a mask for malicious participants' part */`

`/* Identifying the backdoor-triggered embedding */`
3 **for** *each local embedding* $h_i = h_1, ..., h_N$ **do**
4 **for** *each local embedding* $h_j = h_1, ..., h_N$ **do**
5 **if** h_i *is not* h_j **then**
6 $s_{j \to i} \leftarrow ||m_i \odot (h - \text{MAE}(m_j \odot h)||_2$
7 **if** $s_{j \to i} > t_i$ **then**
8 $votes[i] \leftarrow votes[i] + 1$
9 **if** $votes[i] > \frac{N}{2}$ **then**
10 $m_{\text{mal}} \leftarrow m_{\text{mal}} + m_i$

`/* Purifying the concatenated embedding */`
11 $h_{\text{removed}} \leftarrow (1 - m_{\text{mal}}) \odot h$
12 $\hat{h}_{\text{purified}} \leftarrow \text{MAE}(h_{\text{removed}})$
13 $P \leftarrow T(\hat{h}_{\text{purified}})$
14 **return** VFL model prediction P

Anomaly Score Calculation. For an embedding of one participant, VFLIP calculates N-1 anomaly scores based on the embeddings of other N-1 participants (Line 6 in Algorithm 2). To be specific, for h_i, VFLIP masks all the embeddings except for h_j in the concatenated embedding and reconstructs $\hat{h}$ using the MAE. The part of $\hat{h}$ corresponding to h_i, based on h_j, is denoted as $\hat{h}_{j \to i}$. VFLIP defines the anomaly score $s_{j \to i}$ as follows:

$$\hat{h}_{j \to i} = m_i \odot \text{MAE}(m_j \odot h) \tag{4}$$

$$s_{j \to i} = ||\hat{h}_{j \to i} - h_i||_2 \tag{5}$$

To understand this anomaly score, it is essential to delve into the characteristics of VFL backdoor attacks [1,30]. The attacker injects a backdoor trigger only into the target label samples during the training stage. Therefore, at the inference stage, if the attacker inserts the backdoor trigger into non-target label samples to manipulate the VFL model predictions, it results in a relatively high anomaly score for the attacker's embedding because the MAE incorrectly generates the attacker's embedding as the MAE did not learn about the relationships between the backdoor-triggered embedding and the embeddings of the non-target labels.

Majority Voting. If the anomaly score $s_{j \to i}$ exceeds the threshold t_i, h_i gets a vote (Line 7-8 in Algorithm 2). The threshold t_i is determined by the $\mu_i + \rho \cdot \sigma_i$, where μ_i and σ_i represent the mean and standard deviation of all anomaly scores from $\mathcal{H}^{\text{train}}$ for i-th participant. Since VFLIP aims to detect abnormal cases that are not observed in the VFL training stage, it is reasonable to set the threshold based on the distribution of anomaly scores obtained from $\mathcal{H}^{\text{train}}$. ρ is the hyper-parameter for controlling the threshold. If the number of votes h_i received is greater than half of the number of total participants ($\frac{N}{2}$), it is considered as a backdoor-triggered embedding (Line 9-10 in Algorithm 2).

Purification. VFLIP removes the malicious embeddings from the concatenated embedding (Line 11 in Algorithm 2) and feeds them into MAE to obtain the purified $\hat{h}$ (Line 12 in Algorithm 2). Subsequently, the top model uses $\hat{h}$ as input to obtain the final prediction (Line 13 in Algorithm 2).

4 Experiments

4.1 Experiments Setup

Dataset descriptions. Following the previous VFL studies [1,18,30], we evaluate the effectiveness of VFLIP using five datasets: three image datasets (i.e., CIFAR10 [14], CINIC10 [4], Imagenette [11]), one image-text combined dataset (i.e., NUS-WIDE [3]), and one financial dataset (i.e., Bank Marketing (BM) [22]). CIFAR10 and CINIC10 have 10 classes consisting of 32×32 pixel images. Imagenette is a 10-class subset of the Imagenet dataset. Each image is resized by 224×224 pixels. NUS-WIDE has 81 classes with 634 image features and 1000 text features. We select the five classes following the previous study [18]. BM has two classes with 40 features.

Default Training Setup. We validate VFLIP under a four-participant scenario with a single attacker and an eight-participant scenario with three attackers. Based on previous VFL studies [1,6], the data features are vertically split among the participants. The optimization method is Stochastic Gradient Descent (SGD). The bottom model architecture is VGG19 [26] for the image datasets and a 4-layer fully connected network (FCN) for NUS-WIDE and BM. The top model uses a 3-layer FCN. The VFL training epoch is set to 50 for CIFAR10, CINIC10, Imagenette, and NUS-WIDE, and is set to 40 for BM.

Attacks. We evaluate VFLIP on two SOTA attacks: BadVFL [30] and VILLAIN [1]. The attacker without label knowledge conducts the label inference attack proposed by the previous study [1]. The attacker injects a backdoor trigger after E_{bkd} epoch. For the attacker without label knowledge, the poisoning budget is set to 10% and E_{bkd} is set to 20. For the attacker with label knowledge, the poisoning budget is set to 50% and E_{bkd} is set to 5. The attacker's target label is set to 0. The details for attack settings are provided in Appendix A.2.

Table 1. Evaluation for a single attacker on five datasets. No DEF (no defense): Result without any defense mechanism.

Dataset	Label Knowledge	Attack	Defense											
			Accuracy (%) ↑ (Higher is better)						Attack Success Rate (%) ↓ (Lower is better)					
			NO DEF	DP-SGD	MP	ANP	BDT	VFLIP	NO DEF	DP-SGD	MP	ANP	BDT	VFLIP
CIFAR10	w/o	BadVFL	77.34	**78.04**	75.35	75.53	73.28	75.14	30.58	30.81	25.55	28.14	29.20	**13.95**
		VILLAIN	**76.84**	75.04	75.34	73.40	73.15	75.22	86.96	20.40	79.17	64.83	83.00	**3.19**
	with	BadVFL	76.85	**77.30**	74.83	75.41	72.46	75.46	97.50	99.99	87.13	96.49	95.51	**5.52**
		VILLAIN	75.46	**75.52**	74.94	74.11	74.33	73.86	99.84	64.13	99.76	99.80	99.77	**2.79**
CINIC10	w/o	BadVFL	**64.60**	64.43	64.37	64.34	61.94	62.45	21.13	28.27	18.05	17.89	19.52	**13.90**
		VILLAIN	**63.66**	63.14	63.30	63.33	60.88	62.27	75.65	11.53	64.30	65.03	70.36	**3.74**
	with	BadVFL	**64.37**	62.46	62.22	63.88	60.40	63.21	99.46	80.31	91.51	96.82	99.25	**4.92**
		VILLAIN	**62.43**	61.73	61.93	62.09	60.91	61.05	99.97	85.41	99.91	99.94	99.97	**2.34**
Imagenette	w/o	BadVFL	**75.48**	74.98	72.54	75.07	71.22	73.37	61.02	78.01	54.37	60.21	58.46	**14.41**
		VILLAIN	**74.11**	73.83	72.55	73.12	71.25	71.60	96.66	23.80	93.95	96.50	96.50	**2.87**
	with	BadVFL	74.02	**74.97**	70.84	73.64	69.43	71.70	92.21	93.20	73.36	92.08	90.62	**3.79**
		VILLAIN	72.31	**73.27**	69.50	71.37	69.45	69.32	97.38	89.29	96.70	97.33	97.42	**1.87**
NUS-WIDE	w/o	BadVFL	**83.55**	83.32	82.99	81.42	81.66	81.26	72.65	57.68	68.66	61.82	73.12	**9.28**
		VILLAIN	**83.74**	83.16	81.78	80.75	81.21	81.39	89.34	23.73	80.13	72.33	88.75	**7.65**
	with	BadVFL	**82.79**	81.45	80.72	82.52	81.76	79.93	99.99	95.55	95.11	99.98	99.97	**12.45**
		VILLAIN	**82.44**	80.50	80.47	79.26	80.68	80.51	100.00	96.75	99.91	99.22	100.00	**6.30**
BM	w/o	BadVFL	**93.74**	93.70	88.06	93.58	93.44	90.35	26.90	22.25	15.38	31.39	27.20	**10.45**
		VILLAIN	**94.39**	92.77	91.41	94.28	90.23	91.13	45.98	52.60	31.24	45.03	44.94	**5.59**
	with	BadVFL	93.50	93.67	90.19	**93.91**	87.92	91.75	92.14	64.41	78.52	94.30	88.67	**12.54**
		VILLAIN	93.73	92.51	90.50	**93.75**	89.51	90.94	99.98	55.93	99.86	100.00	99.94	**9.43**

Table 2. Evaluation for multiple attackers with label knowledge on five datasets.

Dataset	Attack	Defense											
		Accuracy (%) ↑ (Higher is better)						Attack Success Rate (%) ↓ (Lower is better)					
		NO DEF	DP-SGD	MP	ANP	BDT	VFLIP	NO DEF	DP-SGD	MP	ANP	BDT	VFLIP
CIFAR10	BadVFL	**74.57**	74.31	70.29	72.99	72.98	70.59	99.37	100.00	96.24	99.16	99.06	**8.12**
	VILLAIN	**71.58**	70.56	70.61	71.30	69.36	68.44	100.00	94.57	98.86	100.00	100.00	**5.68**
CINIC10	BadVFL	**61.95**	60.45	58.65	61.67	57.87	58.66	99.88	99.98	91.07	99.64	99.81	**5.37**
	VILLAIN	58.52	55.98	56.46	**58.55**	56.14	55.98	100.00	98.09	99.97	100.00	100.00	**2.24**
Imagenette	BadVFL	**71.77**	71.20	70.85	70.54	67.04	67.54	98.50	98.85	96.85	98.15	97.96	**3.00**
	VILLAIN	**69.71**	69.01	67.90	69.03	67.52	66.13	99.84	96.20	98.97	99.69	99.70	**3.06**
NUS-WIDE	BadVFL	**81.81**	78.80	79.11	81.44	80.04	78.67	99.83	97.50	98.69	99.85	99.82	**14.12**
	VILLAIN	**82.76**	75.54	79.47	81.35	80.37	80.62	99.97	80.21	99.63	99.57	99.95	**7.90**
BM	BadVFL	93.51	93.73	90.43	**93.95**	89.10	91.01	99.98	99.09	99.79	100.00	99.92	**17.65**
	VILLAIN	93.64	92.07	93.63	**93.83**	92.41	90.28	100.00	96.88	100.00	100.00	100.00	**17.12**

Defense Baselines. To the best of our knowledge, there are no backdoor defenses specialized for VFL with split architecture. Following the previous study [1], we apply the existing backdoor defenses designed for DNNs, such as Model Pruning (MP) [19], Adversarial Neural Pruning (ANP) [29], and Backdoor Defense via Transform (BDT) [17] to defend against backdoor attacks in VFL and compare them with VFLIP. For applying BDT to the VFL scenario, we add noise to the embeddings following the previous study [1]. Moreover, as DP-SGD [1] is known for improving the robustness against backdoor attacks

to some extent [7], we analyze the ability of DP-SGD to defend against VFL backdoor attacks.

Defense Settings. To find the defense hyperparameter for MP, ANP, and BDT in VFL, we explore various security-utility trade-off hyperparameters in a wide range like the previous study [21], and report the results with the lowest ASR while maintaining accuracy. For MP, we vary the pruning ratios from 10% to 90%. For ANP, similar to MP, we vary the ANP ratio from %0.1 to 2%. For BDT, we increase the noise level until the accuracy is no longer maintained.

In the case of DP-SGD, we select the lowest ϵ values among those showing stable model convergence: 6.35 for CIFAR10 and CINIC10, 4.65 for Imagenette and NUS-WIDE, and 8.84 for BM. The clipping value is determined based on the median of gradients' magnitude when DP-SGD is not applied, as guided by the previous study [1].

In VFLIP, the architecture of MAE employs a 3-layer FCN for both the encoder and the decoder. We set the learning rate of the "N-1 to 1" strategy to 0.01 and the "1 to 1" strategy's learning rate to 0.1. For identification, ρ is set to 2 for CIFAR10, CINIC10, NUS-WIDE, and BM, and 2.5 for Imagenette. Dropout is set to 10%. The MAE training epoch is set to 20 for CIFAR10, CINIC10, NUS-WIDE, and BM, and it is set to 50 for Imagenette.

Metrics. We employ two metrics to assess the robustness of VFLIP: clean accuracy (ACC) and attack success rate (ASR). ACC is the probability of correctly predicting the true label in the absence of backdoor triggers. ASR represents the probability of the model misclassifying the labels of the backdoor-triggered samples as the attacker's target label.

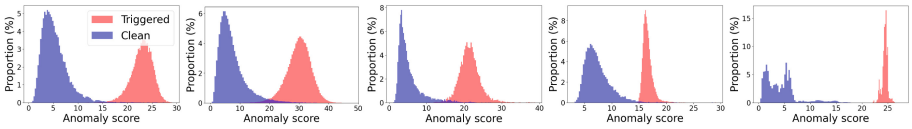
4.2 Main Results

Table 1 presents the results of a four-participant scenario with a single attacker. The results for label inference accuracy are provided in Appendix A.3. Each experiment is repeated five times with different seeds, and the average result is reported. In most cases, other defense techniques fail to mitigate backdoor attacks in VFL. On the other hand, VFLIP achieves the average ASR decrease of 85.56%, 80.50%, 91.84%, 89.97%, and 81.19%, and the average ACC drop of 2.21%, 2.38%, 3.37%, 2.84%, and 3.36% in CIFAR10, CINIC10, Imagenette, NUS-WIDE, and BM, respectively. Even though there is a slight accuracy drop compared with no defense, VFLIP demonstrates that it can effectively mitigate backdoor attacks in VFL.

4.3 Multiple Attackers

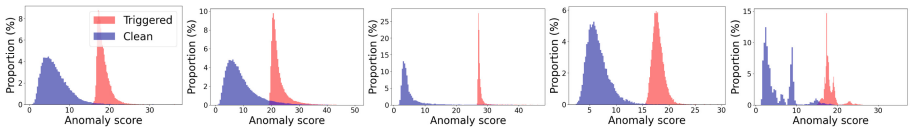
To demonstrate the defense capabilities of VFLIP under multiple attackers, we conduct experiments in an eight-participant scenario with three label-knowledgable attackers, as presented in Table 2. Each experiment is repeated

five times with different seeds, and the average result is reported. The attackers share the information they possess, their objective, and their attack strategy and carry out all malicious actions simultaneously. Thus, the malicious portion in the concatenated embeddings is increased, making the attack more powerful. While other defense techniques fail to mitigate the backdoor attacks, for ASR, VFLIP achieves 6.9%, 3.8%, 3.03%, 11.01%, and 17.39% on average for CIFAR10, CINIC10, Imagenette, NUS-WIDE, and BM, respectively. This indicates that VFLIP has the capacity to defend against multiple attackers. The ACC drop of VFLIP is 4.86%, 4.83%, 5.51%, 3.21%, and 3.13% on average for CIFAR10, CINIC10, Imagenette, NUS-WIDE, and BM, respectively. Although there is a slight increase in ACC drop compared to the four-participant scenario, we note that VFL typically involves collaboration with a small number of participants [28,31], so situations resulting in an accuracy drop greater than observed in these experiments are rare.



(a) CIFAR10 (b) CINIC10 (c) Imagenette (d) NUS-WIDE (e) BM

Fig. 4. Anomaly score distribution with BadVFL on five datasets.



(a) CIFAR10 (b) CINIC10 (c) Imagenette (d) NUS-WIDE (e) BM

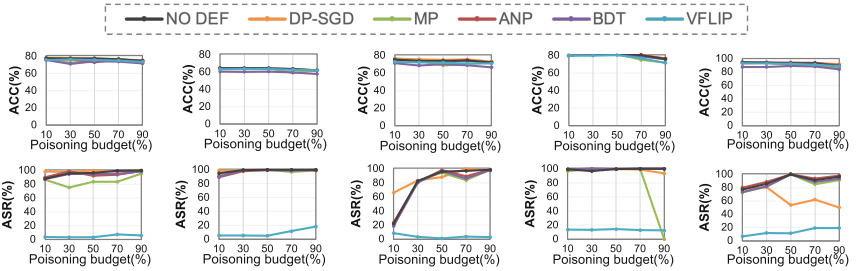
Fig. 5. Anomaly score distribution with VILLAIN on five datasets.

4.4 Anomaly Score Distribution

To demonstrate VFLIP's ability to identify backdoor-triggered embeddings based on the anomaly score, we visualize the anomaly score distribution of clean embeddings and backdoor-triggered embeddings. Figure 4 and Fig. 5 present the distribution of the anomaly scores for each type of attack on five datasets. While there may be a slight overlap between the distributions, most backdoor-triggered embeddings are clearly separated from clean embeddings.

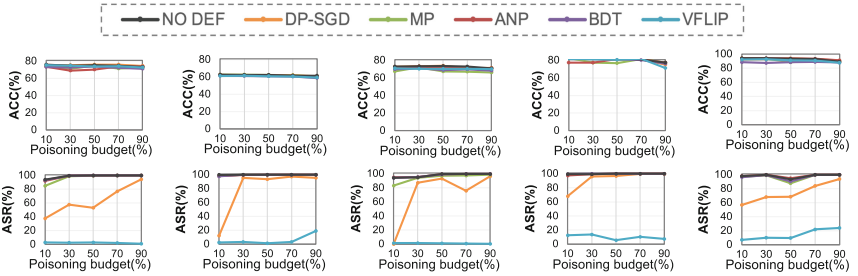
4.5 Ablation Study

In this subsection, we assess how different factors affect the robustness of VFLIP. The default attack setting is the same as the main results with label-knowledgable attackers.



(a) CIFAR10 (b) CINIC10 (c) Imagenette (d) NUS-WIDE (e) BM

Fig. 6. Impact of poisoning budget with BadVFL on five datasets.



(a) CIFAR10 (b) CINIC10 (c) Imagenette (d) NUS-WIDE (e) BM

Fig. 7. Impact of poisoning budget with VILLAIN on five datasets.

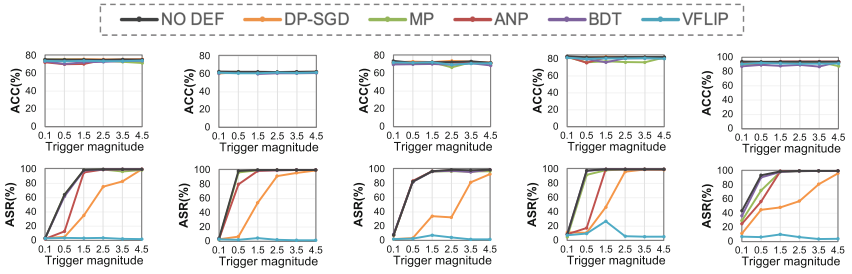
Impact of Poisoning Budget. In VFL, the backdoor attacker can easily adjust their poisoning budget. Therefore, considering an attacker who uses a large poisoning budget is essential. Figure 6 and Fig. 7 presents the results of increasing the poisoning budget from 10% to 90%. DP-SGD mitigates VILLAIN to some extent when the poisoning budget is small, but as the poisoning budget increases, the ASR significantly rises. Other defenses fail to mitigate both of the attacks in most cases. On the other hand, VFLIP shows stable defense performance across poisoning budgets.

Impact of Trigger Magnitude. Since there is no limit to the magnitude of the embeddings in VFL [18], VILLAIN can largely perturb the embeddings to increase the attack performance. Moreover, VILLAIN can attempt to evade identification mechanisms by sending a backdoor-triggered embedding that closely resembles a clean embedding using a small trigger magnitude. This requires the capability to defend against attacks with various trigger magnitudes. Figure 8 presents experiments against VILLAIN with trigger magnitude ranging from 0.1 to 4.5. VFLIP outperforms other defense techniques in all datasets. This indicates that VFLIP can effectively mitigate attacks regardless of trigger magnitudes. Notably, VFLIP maintains the defensive performance even against the small-magnitude triggers (below scale 2). This is attributed to the purification phase which denoises the trigger by reconstructing all the embeddings.

Impact of Bottom Model Architectures. The bottom model architecture can change depending on the attacker. Therefore, we evaluate various bottom model architectures. For image datasets, Resnet-20 [9] and MobileNet [10] are evaluated. For NUS-WIDE and BM, 5-layer FCN and 3-layer FCN are evaluated. The results are provided in Appendix A.4. In the experiments, VFLIP outperforms all other defenses. It indicates that VFLIP demonstrates the defense capability without relying on a specific bottom model architecture.

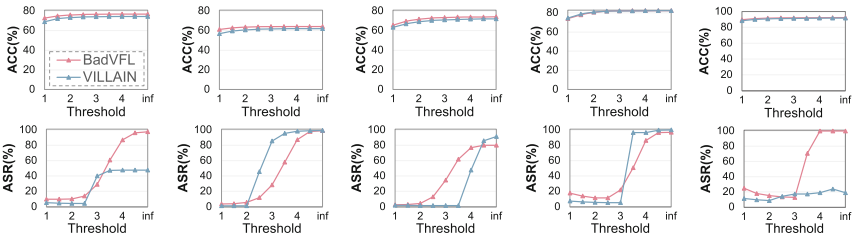
Impact of the Anomaly Score Threshold ρ for VFLIP. Most defense techniques face a trade-off between security and utility. In VFLIP, the threshold ρ for the anomaly score introduces a trade-off between robustness and accuracy. Figure 9 illustrates the performance with respect to ρ. The results show that when ρ is set to 2 or 2.5, there is a negligible decrease in ACC while ASR is still low. It indicates that there exists a proper trade-off point in VFLIP.

Impact of the MAE Training Strategies. VFLIP uses two MAE training strategies. To evaluate each strategy individually, experiments are conducted by training MAE with only one strategy at a time. The results are provided



(a) CIFAR10 (b) CINIC10 (c) Imagenette (d) NUS-WIDE (e) BM

Fig. 8. Impact of trigger magnitude on five datasets.



(a) CIFAR10 (b) CINIC10 (c) Imagenette (d) NUS-WIDE (e) BM

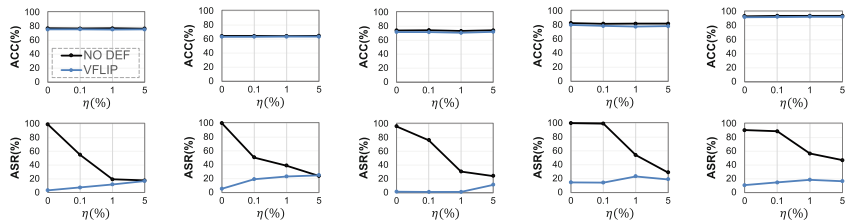
Fig. 9. Impact of anomaly score threshold ρ for VFLIP on five datasets.

in Appendix A.5. While using a single training strategy shows slightly better performance in some cases, using two training strategies shows more stable and better performance in most cases.

5 Adaptive Attack

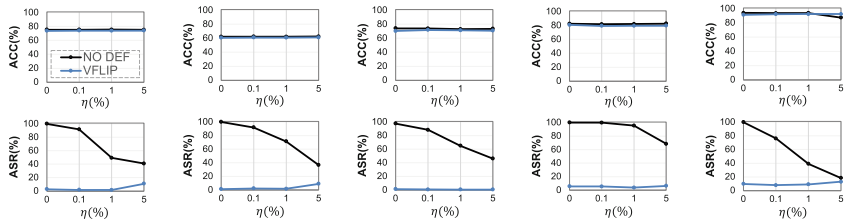
To demonstrate the robustness of VFLIP against the attacker who knows the VFLIP mechanism, we design and evaluate an adaptive backdoor attack strategy. This strategy aims to poison $\mathcal{H}^{\text{train}}$ so that the MAE learns relationships between the backdoor trigger and non-target embedding, reducing the anomaly scores. For this, at the last epoch, they insert backdoor triggers into samples of non-target labels with a certain probability. Thus, the abnormal cases are included in $\mathcal{H}^{\text{train}}$. It makes VFLIP difficult to identify backdoor-triggered embeddings. Conversely, this weakens the connection between the backdoor trigger and the target label in the top model, reducing the attack success rate. To balance this trade-off, the attacker adjusts the probability of inserting triggers into non-target samples, denoted as η.

Figure 10 and Fig. 11 indicates that as η increases, the ASR for VFLIP slightly increases. On the other hand, the ASR significantly declines when there is no defense. This implies that the connection between the backdoor trigger and



(a) CIFAR10 (b) CINIC10 (c) Imagenette (d) NUS-WIDE (e) BM

Fig. 10. Evaluation for the adaptive attacks with BadVFL against VFLIP.



(a) CIFAR10 (b) CINIC10 (c) Imagenette (d) NUS-WIDE (e) BM

Fig. 11. Evaluation for the adaptive attacks with VILLAIN against VFLIP.

the target label in the top model is substantially weakened before completely compromising the VFLIP's MAE. This demonstrates that to compromise the MAE, the attacker must sacrifice their ASR.

6 Conclusion

In this paper, we propose VFLIP, a novel backdoor defense for VFL. VFLIP identifies the backdoor-triggered embedding and purifies their malicious influences. Additionally, we demonstrate that to compromise VFLIP's MAE, the attacker has to significantly sacrifice their ASR. Through extensive experiments, we demonstrate that VFLIP is robust and effective for defending against the backdoor attacks in the VFL. While this paper focuses on backdoor attacks in VFL, future research will need to explore these backdoor attacks in the broader context of VFL, considering additional complex problems such as non-IID issues.

Acknowledgements. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2023-00277326), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [IITP-2023-RS-2023-00256081 (under the artificial intelligence semiconductor support program to nurture the best talents), No. 2022-0-00516 (Derivation of a Differential Privacy Concept Applicable to National Statistics Data While Guaranteeing the Utility of Statistical Analysis), RS-2021-II212068 (Artificial Intelligence Innovation Hub, EWU), and RS-2022-00155966 (Artificial Intelligence Convergence Innovation Human Resources Development, EWU)], the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2024, Seoul R&D Program(CY230117) through the Seoul Business Agency(SBA) funded by Seoul Metropolitan Government, 2024 AI Security Prototype Development Support Program funded by Ministry of Science and ICT and Korea Internet & Security Agency, and azoo.ai.

A Appendix

A.1 VFL Backdoor Attacks

BadVFL [30] is a data-level backdoor attack that consists of label inference and backdoor injection. For backdoor injection, BadVFL first replaces the target label local data with non-target label local data. Subsequently, the trigger is injected into the replaced data. BadVFL employs the pre-defined static trigger like the previous study [9].**VILLAIN** [1] is an embedding-level backdoor attack. Initially, VILLAIN conducts a label inference attack during the training stage. For backdoor injection, VILLAIN utilizes trigger fabrication, backdoor augmentation, and learning rate adjustment. For trigger fabrication, VILLAIN chooses an additive trigger in the embedding level, rather than a replacement trigger. First, VILLAIN chooses M dimensions with the highest standard deviation as the trigger area. Next, the trigger value is designed by using the average standard deviation of the selected dimension, denoted by σ. Ultimately, a repeating pattern of σ serves as a backdoor trigger, represented as $\gamma \cdot [\sigma, \sigma, -\sigma, -\sigma, \cdots, \sigma, \sigma, -\sigma, -\sigma]$ where γ is a hyperparameter for the trigger magnitude. Next, the trigger undergoes two types of augmentation, during backdoor injection. One method involves randomly deleting parts of the trigger, and the other involves multiplying the trigger by a random value within the range of $[\underline{\lambda}, \bar{\lambda}]$. Moreover, before the backdoor injection, VILLAIN amplifies their local learning rate, causing the top model to be more dependent on the attacker's embeddings. Once the backdoor injection begins, the local learning rate is decreased to a smaller value. The backdoor injection process is conducted after training for E_{bkd} epochs. They empirically show that the existing backdoor defense for DNNs cannot mitigate VILLAIN.

A.2 Attack Settings

For the label inference attack [1], the label inference module [6] uses a batch size of 64 and a learning rate of 0.002. For swapping, the number of candidates in the minibatch is 3 $\times$ poisoning budget $\times$ batch size. If the gradient magnitude of the previous embedding is smaller than the average and the gradient magnitude of the swapped embedding is less than 10 times the previous gradient magnitude, it is identified as a sample having the target label. Following previous studies [1,6], the attacker increases their local learning rate to enhance their malicious actions. For datasets other than Imagenette, the attacker's learning rate is multiplied by two, whereas it is increased by 1.2 times for Imagenette. The BadVFL trigger size is set to 5×5 for CIFAR10 and CINIC10, 40×40 for Imagenette, 60 for NUS-WIDE, and 8 for BM. The VILLAIN trigger size is 75% of the attacker's embedding dimension. The trigger magnitude γ for VILLAIN is set to 3 for CIFAR10, CINIC10, NUS-WIDE, and BM, and 4 for Imagenette. We select the attacker as a participant holding features in the middle of the sample following the previous study [1].

Table 3. Evaluation for various bottom model architectures.

Dataset	Architecture	Attack	Defense											
			Accuracy (%) ↑ (Higher is better)						Attack Success Rate (%) ↓ (Lower is better)					
			NO DEF	DP-SGD	MP	ANP	BDT	VFLIP	NO DEF	DP-SGD	MP	ANP	BDT	VFLIP
CIFAR10	Resnet-20	BadVFL	**67.53**	66.38	66.10	65.86	65.28	65.03	97.52	95.33	96.95	92.36	96.85	**5.70**
		VILLAIN	63.77	62.00	**63.78**	62.97	61.52	62.00	100.00	81.93	100.00	100.00	100.00	**3.74**
	MobileNet	BadVFL	67.48	**68.60**	65.36	65.65	62.07	66.05	99.26	99.92	88.46	94.93	98.15	**7.88**
		VILLAIN	64.39	**65.10**	62.98	62.86	61.84	62.10	100.00	51.25	100.00	100.00	100.00	**2.22**
CINIC10	Resnet-20	BadVFL	**54.70**	53.50	50.18	54.00	52.42	53.45	99.37	98.59	97.03	99.68	99.62	**12.00**
		VILLAIN	**52.73**	50.48	49.33	52.14	49.74	51.26	100.00	99.77	99.97	100.00	100.00	**2.28**
	MobileNet	BadVFL	56.94	**57.20**	56.24	56.26	54.64	55.36	99.92	99.96	99.79	99.88	99.89	**5.55**
		VILLAIN	**53.39**	52.23	52.84	52.90	51.77	51.67	100.00	100.00	100.00	100.00	100.00	**2.92**
Imagenette	Resnet-20	BadVFL	**70.67**	67.08	66.66	64.99	68.27	66.75	87.19	93.56	80.67	75.85	86.48	**24.36**
		VILLAIN	**67.82**	63.32	63.17	65.74	63.95	63.37	100.00	99.83	99.61	99.91	99.88	**1.62**
	MobileNet	BadVFL	68.94	**69.78**	64.35	69.23	66.47	65.44	96.13	80.31	68.73	96.28	95.62	**12.47**
		VILLAIN	**69.36**	67.37	61.77	62.50	63.54	66.67	99.74	69.23	99.70	99.85	99.82	**2.90**
NUS-WIDE	5-layer FCN	BadVFL	80.03	79.57	75.08	**80.49**	76.86	75.68	98.69	100.00	68.73	99.17	98.81	**24.28**
		VILLAIN	81.32	75.82	**81.97**	81.54	81.12	77.62	100.00	37.71	100.00	100.00	100.00	**5.84**
	3-layer FCN	BadVFL	**82.54**	81.91	76.98	82.49	81.58	80.65	99.60	96.66	99.88	99.90	99.94	**9.47**
		VILLAIN	82.75	**83.32**	76.66	82.82	81.29	80.69	100.00	93.45	88.19	100.00	99.98	**5.07**
BM	5-layer FCN	BadVFL	93.64	93.72	89.03	**94.00**	86.93	92.01	89.80	71.74	71.15	91.91	84.82	**14.96**
		VILLAIN	93.72	92.73	91.71	**94.05**	87.78	91.81	99.90	62.57	99.66	99.93	99.55	**14.80**
	3-layer FCN	BadVFL	93.96	93.36	87.43	**94.37**	87.48	91.34	93.03	64.50	68.72	94.75	88.40	**17.61**
		VILLAIN	94.00	91.68	87.70	**94.33**	86.53	89.28	99.97	85.33	99.60	100.00	99.86	**16.41**

Table 4. Evaluation for each training strategy.

Dataset	Label Knowledge	Attack	Defense					
			Accuracy (%) ↑ (Higher is better)			Attack Success Rate (%) ↓ (Lower is better)		
			1 to 1	N-1 to 1	VFLIP	1 to 1	N-1 to 1	VFLIP
CIFAR10	w/o	BadVFL	**76.32**	73.48	75.62	19.52	**8.93**	13.50
		VILLAIN	74.46	71.09	**75.33**	3.94	4.76	**3.67**
	with	BadVFL	**75.79**	71.45	75.56	3.75	4.65	**3.30**
		VILLAIN	73.33	66.41	**73.82**	4.42	**1.99**	2.78
CINIC10	w/o	BadVFL	**63.42**	62.60	63.18	17.74	13.35	**12.95**
		VILLAIN	62.18	60.75	**62.80**	2.94	3.31	**2.84**
	with	BadVFL	63.04	60.51	**63.39**	6.32	5.57	**5.23**
		VILLAIN	**61.18**	59.62	60.74	2.59	2.13	**1.17**
Imagenette	w/o	BadVFL	71.74	35.79	**72.94**	16.76	42.49	**14.00**
		VILLAIN	71.01	25.05	**71.32**	3.77	26.42	**2.68**
	with	BadVFL	**72.73**	41.28	71.82	10.79	22.91	**1.31**
		VILLAIN	**71.01**	28.73	70.35	1.62	0.00	**1.28**
BM	w/o	BadVFL	81.35	79.46	**81.65**	**9.39**	29.83	9.80
		VILLAIN	80.45	80.23	**81.51**	8.64	10.64	**6.38**
	with	BadVFL	79.00	78.54	**80.31**	17.67	21.44	**14.54**
		VILLAIN	79.62	79.80	**80.83**	**4.82**	8.28	5.65
NUS-WIDE	w/o	BadVFL	**93.47**	49.97	91.27	15.97	0.00	**14.71**
		VILLAIN	89.74	49.97	**91.91**	10.32	8.28	**7.31**
	with	BadVFL	89.01	50.00	92.13	21.26	100.00	**10.78**
		VILLAIN	90.80	50.00	**91.30**	11.07	100.00	**9.82**

A.3 Results for Label Inference Attacks

Table 5. Accuracy of label inference attacks on five datasets.

Label Inference Attack	Label Inference Accuracy ↑ (Higher is better)									
	CIFAR10		CINIC10		Imagenette		NUS-WIDE		BM	
	DP-SGD	Others	DP-SGD	Others	DP-SGD	Others	DP-SGD	Others	DP-SGD	Others
[1]	88.16	89.27	94.59	93.15	98.97	95.20	94.16	94.85	83.63	89.14

A.4 Impact of Bottom Model Architecture

Table 3 presents the results for various bottom architectures.

A.5 Impact of the MAE Training Strategies

Table 4 presents the results for each training strategy in VFLIP (Table 5).

References

1. Abadi, M., et al.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016)
2. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: byzantine tolerant gradient descent. Adv. neural inform. process. syst. **30** (2017)
3. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of Singapore. In: Proceedings of the ACM International Conference on Image and Video Retrieval, pp. 1–9 (2009)
4. Darlow, L.N., Crowley, E.J., Antoniou, A., Storkey, A.J.: CINIC-10 is not imagenet or CIFAR-10 (2018)
5. Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to Byzantine-Robust federated learning. In: 29th USENIX Security Symposium (USENIX Security 20), pp. 1605–1622 (2020)
6. Fu, C., et al.: Label inference attacks against vertical federated learning. In: 31st USENIX Security Symposium (USENIX Security 22), pp. 1397–1414 (2022)
7. Gao, K., Bai, Y., Gu, J., Yang, Y., Xia, S.T.: Backdoor defense via adaptively splitting poisoned dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4005–4014 (2023)
8. Gharibshah, Z., Zhu, X.: Local contrastive feature learning for tabular data. In: Proceedings of the 31st ACM International Conference on Information and Knowledge Management, pp. 3963–3967 (2022)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

10. Howard, A.G., et al.: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
11. Howard, J., Gugger, S.: Fastai: a layered API for deep learning. Information **11**(2), 108 (2020)
12. Huang, W.R., Geiping, J., Fowl, L., Taylor, G., Goldstein, T.: Metapoison: practical general-purpose clean-label data poisoning. Adv. Neural. Inf. Process. Syst. **33**, 12080–12091 (2020)
13. Jin, X., Chen, P.Y., Hsu, C.Y., Yu, C.M., Chen, T.: Cafe: catastrophic data leakage in vertical federated learning. Adv. Neural. Inf. Process. Syst. **34**, 994–1006 (2021)
14. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
15. Lai, J., Wang, T., Chen, C., Li, Y., Zheng, Z.: VfedAd: a defense method based on the information mechanism behind the vertical federated data poisoning attack. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 1148–1157 (2023)
16. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Anti-backdoor learning: training clean models on poisoned data. Adv. Neural. Inf. Process. Syst. **34**, 14900–14912 (2021)
17. Li, Y., Zhai, T., Jiang, Y., Li, Z., Xia, S.T.: Backdoor attack in the physical world. arXiv preprint arXiv:2104.02361 (2021)
18. Liu, J., Xie, C., Koyejo, S., Li, B.: CoPur: certifiably robust collaborative inference via feature purification. Adv. Neural. Inf. Process. Syst. **35**, 26645–26657 (2022)
19. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: defending against backdooring attacks on deep neural networks. In: Bailey, M., Holz, T., Stamatogiannakis, M., Ioannidis, S. (eds.) RAID 2018. LNCS, vol. 11050, pp. 273–294. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00470-5_13
20. Liu, Y., et al.: Vertical federated learning: concepts, advances and challenges (2023)
21. Lyu, L., Yu, H., Yang, Q.: Threats to federated learning: a survey. arXiv preprint arXiv:2003.02133 (2020)
22. Moro, S., Cortez, P., Rita, P.: A data-driven approach to predict the success of bank telemarketing. Decis. Support Syst. **62**, 22–31 (2014). https://doi.org/10.1016/j.dss.2014.03.001
23. Nguyen, T.D., et al.: {FLAME}: Taming backdoors in federated learning. In: 31st USENIX Security Symposium (USENIX Security 22), pp. 1415–1432 (2022)
24. Shafahi, A., et al.: Poison frogs! targeted clean-label poisoning attacks on neural networks. Adv. Neural Inf. Process. Syst. **31** (2018)
25. Shejwalkar, V., Houmansadr, A.: Manipulating the byzantine: optimizing model poisoning attacks and defenses for federated learning. In: NDSS (2021)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
27. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference On Machine Learning, pp. 1096–1103 (2008)
28. Wei, K., et al.: Vertical federated learning: challenges, methodologies and experiments. arXiv preprint arXiv:2202.04309 (2022)
29. Wu, D., Wang, Y.: Adversarial neuron pruning purifies backdoored deep models. Adv. Neural. Inf. Process. Syst. **34**, 16913–16925 (2021)
30. Xuan, Y., Chen, X., Zhao, Z., Tang, B., Dong, Y.: Practical and general backdoor attacks against vertical federated learning. In: Koutra, D., Plant, C.,

Gomez Rodriguez, M., Baralis, E., Bonchi, F. (eds.) Machine Learning and Knowledge Discovery in Databases: Research Track (2023). https://doi.org/10.1007/978-3-031-43415-0_24

31. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) **10**(2), 1–19 (2019)