Vulnerabilities of Data Protection in Vertical Federated Learning Training and Countermeasures

Derui Zhu, Jinfu Chen, Xuebing Zhou, Weiyi Shang, *Senior Member, IEEE,* Ahmed E. Hassan, *Fellow, IEEE*, and Jens Grossklags, *Senior Member, IEEE,*

***Abstract*—Vertical federated learning (VFL) is an increasingly popular, yet understudied, collaborative learning technique. In VFL, features and labels are distributed among different participants allowing for various innovative applications in business domains, e.g., online marketing. When deploying VFL, training data (labels and features) from each participant ought to be protected; however, very few studies have investigated the vulnerability of data protection in the VFL training stage. In this paper, we propose a posterior-difference-based data attack, *VFLRecon*, reconstructing labels and features to examine this problem. Our experiments show that standard VFL is highly vulnerable to serious privacy threats, with reconstruction achieving up to 92% label accuracy and 0.05 feature MSE, compared to our baseline with 55% label accuracy and 0.19 feature MSE. Even worse, this privacy risk remains during standard operations (e.g., encrypted aggregation) that appear to be safe. We also systematically analyze data leakage risks in the VFL training stage across diverse data modalities (i.e., tabular data and images), different training frameworks (i.e., with or without encryption techniques), and a wide range of training hyperparameters. To mitigate this risk, we design a novel defense mechanism, *VFLDefender*, dedicated to obfuscating the correlation between bottom model changes and labels (features) during training. The experimental results demonstrate that *VFLDefender* prevents reconstruction attacks during standard encryption operations (around 17% more effective than standard encryption operations).**

***Index Terms*—Privacy-preserving machine learning, vertical federated learning, privacy leakage, data safety, privacy.**

I. INTRODUCTION

Machine learning techniques are increasingly integrated into daily routines, e.g., with recommendation systems [10] or medical diagnosis techniques [26], to improve quality of life. However, the success of machine learning techniques relies on the availability of data, and human-level machine intelligence cannot be achieved without big data as training sets. Accordingly, there is an increasing demand for data sharing to improve model performance. For example, financial companies can dramatically improve their customer risk prediction models with customer data from other banks. However, accessing such data from other organizations is very difficult [36], [50], since data is regarded as a key asset by every organization. In addition, governments are issuing more and stricter policies, e.g., GDPR, that decrease the flow of information across organizational boundaries.

In early 2016, Google proposed a new artificial intelligence (AI) technique, federated learning (FL), to address the data sharing problem [25]. FL is a collaborative learning technique that trains a global model using data from multiple participants [25]. Unlike traditional collaborative learning, the training of FL models does not require a centralized server to collect the data stored by each participant. Instead, to train FL models, the participants keep data locally, and only intermediate data, e.g., gradients, are shared. Therefore, FL promotes the cooperative training of models among different organizations without requiring each organization to share original data. However, even though the original data is not shared during FL model training, significant data leakage risks exist [32].

FL has two important variants, horizontal FL (HFL) and vertical FL (VFL), which differ with regard to label ownership. In HFL, each participant can access the entire model and their own labels, while in VFL, the participants can only access part of the model and only one participant owns labels. Previous studies [14], [15], [58] investigated the risks of leakage of training data in FL, focusing on HFL. In contrast, only a small number of articles have examined the risks of training data leakage in VFL. These risks turn out to be more problematic in the VFL setting compared to the HFL setting [47], [50]. Not only is VFL more widely used than HFL [51], VFL applications are usually associated with highly sensitive data, e.g., financial and government data, where data leakage is a serious concern [17], [27]. To the best of our knowledge, no comprehensive privacy risk analysis, including leakage of labels and features, has been conducted in the context of VFL **training**. Additionally, all related studies were conducted in non-encryption-based VFL training frameworks [7], [13], [29]. However, it is critical to understand how much data from each participant may be leaked during the VFL training process using practically relevant encryption-based training frameworks.

To fill this research gap, we conduct a systematic analysis of data leakage risks in the VFL training stage. In particular, we propose a simple yet efficient posterior-difference-based attack approach, *VFLRecon*, to reconstruct labels and features during VFL training. An adversarial participant can apply the posterior difference of a bottom model between two consecutive training steps to reconstruct the labels or features owned by other participants. Following practical threat model assumptions [35], [40],

D. Zhu and J. Grossklags are with the Department of Computer Science, Technical University of Munich, Garching, Boltzmannstr. 3, 85748, Germany (e-mail: derui.zhu@tum.de; jens.grossklags@in.tum.de).

J. Chen is with the School of Computer Science, Wuhan University, Wuhan, China, 430072 (e-mail: jinfuchen@whu.edu.cn)

W. Shang is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1 Canada (e-mail: wshang@uwaterloo.ca)

X. Zhou is with the Huawei Munich Research Center, Munich, Riesstraße 25, 80992, Germany (e-mail: xuebing.zhou@huawei.com)

Ahmed E. Hassan is with the School of Computing, Queen's University, Kingston, Canada (e-mail: ahmed@cs.queensu.ca)

[58], we assume that the adversarial participants are "honest-but-curious", which means that they contribute truthfully to the VFL training. However, the adversarial participants are capable of recording any intermediate information related to their bottom model updates during VFL training, which can be considered the most realistic scenario [40].

To ensure the practical relevance of our work, we evaluate *VFLRecon* on diverse open-source benchmark datasets ranging from tabular data to images, namely, Sensorless Drive Diagnosis [6], Criteo [3], CIFAR-10 [30], BHI [48], Avazu [2], and CelebA [4]. The experiments are conducted using VFL training frameworks including non-encryption-based and encryption-based operations (encrypted aggregation) [56]. The experimental results show that *VFLRecon* achieves consistent effectiveness in reconstructing training samples during VFL training. We find that the adversarial participants can reconstruct labels with very high accuracy (i.e., $>92\%$ in Criteo) in neural-network-based (NN-based) VFL model training without encryption-based operations when they have half of the features of the training samples. Furthermore, *VFLRecon* can efficiently reconstruct the features of tabular data from other participants with a very small mean square error (MSE), e.g., 0.05 in Criteo, in the same setting. Besides tabular data, we also demonstrate that *VFLRecon* can effectively reconstruct the images held by other participants, with an MSE of 0.04 and 0.03 in CIFAR-10 and BHI, respectively. Surprisingly, similar results are reached in VFL model training with encryption-based aggregation protection. As such, our study reveals that encryption operations are not effective in preventing data leakage in VFL training, thereby highlighting the necessity of designing a more dedicated defense method.

While standard encryption aggregation in VFL training is shown to be ineffective against *VFLRecon*, we propose a gradients-obfuscation-based approach, *VFLDefender*, to mislead adversaries. Indeed, the experimental results demonstrate that we can effectively reduce the correlation between model updates and the input samples. Specifically, the accuracy of reconstructed labels decreases substantially from 0.86 to 0.69, while the MSE increases from 0.01 to 0.14 (shown in Table VI).

Our paper makes the following contributions:

- We present the first comprehensive analysis of data leakage risks **in VFL training**. In particular, we propose a novel simple yet effective attack, *VFLRecon*, to demonstrate the serious leakage risks with regard to **labels and features** in VFL training.
- Moreover, our work highlights that **standard encryption-based aggregation** techniques are **not** capable of preventing data leakage during NN-based VFL training.
- Based on our findings, we propose a **gradients-obfuscation-based** defense approach, *VFLDefender*, which can effectively protect each VFL participant's training data privacy.

The rest of this paper is organized as follows: Section II introduces the background of this work, and Section III discusses prior research. Section IV details our methodology, and Section V presents our experimental setup and data collection. Section VI reports the results and a discussion of our attack evaluation. Section VII demonstrates the approaches, which mitigate the data leakage risks. Section VIII analyzes and discusses the defense performance. Section IX discusses potential limitations, and Section X presents the threats to validity of our study. Finally, Section XI concludes this paper.

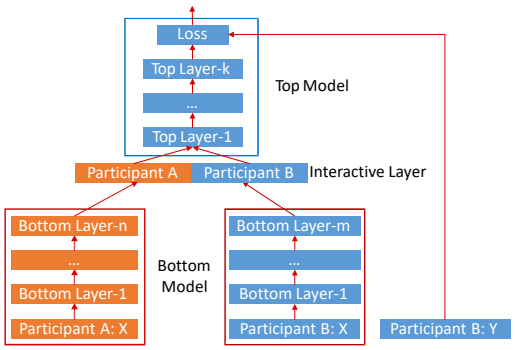

Fig. 1: Neural-network-based VFL model architecture [55], [56]

II. BACKGROUND

In this section, we introduce the background of our work considering primarily two aspects: vertical federated learning, and encryption-based vertical federated learning training.

A. Vertical Federated Learning (VFL)

Vertical federated learning is a distributed machine learning framework, which aims at training an AI model across different participants who share the same sample spaces rather than feature spaces [54]. Figure 1 shows a general architecture of NN-based VFL models. In the VFL setting, each participant holds different features or labels belonging to the same samples. Participants are divided into two groups based on whether they own labels. In general, a participant with labels is categorized as an active participant; otherwise, as a passive participant. Suppose that we have two participants, A and B, where only participant B owns labels. The general NN-based VFL model is then defined as:

$$\mathcal{Y} = h(g(\mathcal{X}^{\mathrm{A}}; \theta_{\mathrm{A}}), g(\mathcal{X}^{\mathrm{B}}; \theta_{\mathrm{B}});\ \theta_{\mathrm{t}}) \tag{1}$$

where $\mathcal{X}^{\mathrm{A}}$ and $\mathcal{X}^{\mathrm{B}}$ are the features owned by participants A and B, respectively. θ_{A} and θ_{B} are the parameters of bottom models g owned by participant A and participant B, respectively. θ_{t} are the parameters of the top model h. Note that the top model is only owned by participant B with data labels.

In general, NN-based VFL models can be trained with the following steps. First, each bottom model takes their local data's features as input to run a forward pass calculation and output the representations of their local features. After that, they upload those representations (refer to embedding) to the top model. Next, the top model aggregates all uploaded representations from each bottom model to compute the final predictions. Comparing the predictions with ground-truth labels, the top model further calculates the gradients with respect to the loss. Then, the gradients are back-propagated to each bottom model from the top model, enabling the VFL model to make an update.

TABLE I: Summary of Notations

Notation	Description
α_{A}	Participant A's output
α_{B}	Participant B's output
σ	Activation function, e.g., Relu, Tanh, etc.
$\mathrm{W_A}$	Weights that connect α_{A} and first layer of top model
$\mathrm{W_B}$	Weights that connect α_{B} and first layer of top model

B. Encryption-based Vertical Federated Learning Training

In general, during the VFL training, each participant sends their local data representations (output of the bottom model) to the top model via plaintext. However, embedding-sharing has been shown to lead to the leakage of original data [11], [43]. As the output of a bottom model is an embedding of the local data from one participant, it is risky to send those outputs to the top model directly without applying any protection mechanisms. As a solution to this problem, encryption techniques, such as additively homomorphic encryption, can protect the bottom model output, allowing the top model to calculate loss and gradients without using the plaintext output from the bottom models [56].

With the notation from Table I, we can introduce the encryption mechanisms applied in VFL training. We use $[\cdot]$ to represent an encryption operation. The working process can be described as follows. z is the first layer output of the top model, which is associated with each bottom model's output. The goal of privacy preservation is to calculate z without knowing the value of a bottom model's output. First, participant A encrypts its bottom model output, $[\alpha_A]$, and then uploads it to the top model. Second, the top model generates a noise ϵ_{B} and computes $[z_{\mathrm{A}}] = [\alpha_{\mathrm{A}}] * W_{\mathrm{A}}$ and $z_{\mathrm{B}} = \alpha_{\mathrm{B}} * W_{\mathrm{B}}$. Next, the top model sends $[z_{\mathrm{A}} + \epsilon_{\mathrm{B}}] = [z_{\mathrm{A}}] + \epsilon_{\mathrm{B}}$ to participant A in order to decrypt z_{A}; meanwhile W_{A} is protected from being seen by participant A. Next, participant A decrypts $[z_{\mathrm{A}} + \epsilon_{\mathrm{B}}]$ and sends $z_{\mathrm{A}} + \epsilon_{\mathrm{B}} + \alpha_{\mathrm{A}} * \epsilon_{\mathrm{acc}}$, where ϵ_{acc} is a hyper-parameter ranging from 0 to 1, to the top model. Afterwards, since the noise ϵ_{B} can be eliminated, the top model can calculate its first layer output $z = \sigma(z_{\mathrm{B}} + z_{\mathrm{A}} + \alpha_{\mathrm{A}} * \epsilon_{\mathrm{acc}})$. Then, the top model uses z as input to run its forward pass to compute the final prediction.

III. RELATED WORK

In this section, we present related prior research regarding two aspects: 1) privacy attacks in federated learning, and 2) privacy protections in federated learning.

A. Privacy Attacks in Federated Learning

The training of AI models typically relies on a larger amount of collected data raising heightened concerns about training data leakage. Several works explore data leakage of training data in the HFL setting [41], [58], as well as attacks to identify whether an example is used in the HFL model's training set [35]. In particular, many successful data inversion attacks to reconstruct the HFL model's input data with only the gradients' information have been reported [21], [22].

Further, various privacy attacks have been proposed against HFL, including membership inference, and properties inference, etc. In membership inference [34], [35], [42], the attacker aims to infer whether a data sample is included in another participant's training dataset. Properties inference [34] focuses on reconstructing the data samples belonging to other participants via the intermediate information exchanged.

In contrast to HFL, very few studies have explored the privacy risks in VFL focusing primarily on data leakage in the VFL inference phase. Yang et al. [52] construct a feature reconstruction attack based on trained VFL models by minimizing the distance between the predictions from reconstructed features and target features using zeroth-order gradient estimation. Luo et al. [31] study the feature reconstruction attacks during VFL inference, focusing on logistic, tree-based, and NN-based models, while Fu et al. [13] proposed a label reconstruction attack by fine-tuning a trained bottom model in a semi-supervised manner to predict the sample labels. Importantly, these approaches can only be applied after the VFL model has been trained and are not feasible during the model training phase.

In addition, Fu et al. [13] have also presented several attempts to analyze the potential label leakage risk in the VFL training phase. However, their work is only applicable for reconstructing training labels when the top model (server) is non-trainable or when assuming non-honest adversary participants. Although these situations might arise in extreme cases, they are generally deemed impractical as the common practice requires the top model to be trainable and the participants to be honest, i.e., to faithfully adhere to the training protocol under performance supervision. Besides, Li et al. [29] exploit the norm of gradients in split learning to reconstruct labels during model training. The key limitation of [29] is that they solely support two-party scenarios in which one participant only holds labels, and the other only holds features. Moreover, [29] is restricted to binary classification tasks. Finally, Ye et al. [53] investigate binary feature reconstructing by solving the linear equations in training, but it is only applicable for scenarios in which the feature-holding participants contain at most one layer of neural network trainable parameters, rendering it an unrealistic setup.

To the best of our knowledge, no comprehensive privacy risk analysis, including leakage of labels and features, has been conducted in the context of VFL training. Additionally, all existing related studies are conducted in non-encryption-based VFL training frameworks. Note that data leakage in VFL training is generally regarded as a more serious issue than data leakage during VFL model inference [23]. Furthermore, although recent work [13], [29], [53] attempted to assess label or feature leakage risks in VFL training, the authors concentrated on particular cases of VFL models for very narrow application scenarios, e.g., binary classification, and binary features. Different from prior works on VFL leakage risk analysis, this paper explores label and feature leakage risks in the VFL training process, that applies to any NN-based model.

B. Privacy Protections in Federated Learning

Many prior approaches have been introduced to prevent training data leakage in federated learning. The approaches

can be categorized into two categories. The first category is data satinization [28], [33], e.g., k-anonymization, to remove sensitive information from the training data to reduce the capability of an adversary to obtain or infer sensitive information about the training data. The other category aims to protect the training data from AI model training by adding random noise in the model training process, e.g., differential privacy (DP) [5], [46]. Ranbaduge et al. [38] study the trade-off between model utility and privacy loss in a (ϵ, δ)-differential privacy setting for VFL model training. The DP-based noise can be added to the model input, gradients, and loss functions [46], [49]. Complementing the DP-based defense strategy, Ye et al. [53] propose a protocol to add Gaussian-based noise to the output of each bottom model. However, their defense strategies only protect categorical features.

FL training requires gradients-related information to be exchanged between each participant. However, prior research has shown that the information exchanged can lead to privacy leakage [37], [44], [34], [45]. Encryption-based exchange is a solution for protecting information exchanged. Secure multi-party computation (SMC) is one type of encryption technique that runs secret computations among multiple participants [16]. In early 2016, Google proposed a gradient aggregation algorithm based on SMC to prevent data leakage from HFL training [8]. This prevents the server from obtaining the exact gradient value of each participant. Furthermore, SMC combined with differential privacy allows for HFL training with better privacy protection guarantees [8], [46].

SMC can also be applied to train different VFL models, e.g., tree-based models. A tree-based VFL model can be trained using secure aggregation to calculate each candidate node's information loss, while the statistics about each node are kept secret to each participant [9]. Prior studies also proposed a solution to aggregate bottom model output with homomorphic encryption for NN-based VFL training to prevent data leakage [56]. However, our study finds that the existing encryption solutions cannot prevent data leakage from NN-based VFL training. Therefore, our work proposes a gradient-perturbation-based defense technique to protect data privacy during VFL training.

IV. VFLRECON: DATA RECONSTRUCTION ATTACKS

In this section, we analyze the vulnerability of training data protection in the VFL training stage and present our attack, *VFLRecon*, to better understand the potential impact of adversarial participants in reconstructing training data, i.e., labels and features, from other participants during the VFL training process.

A. Training Data Leakage Risks in Vertical Federated Learning

In the VFL setting, each participant is not able to directly obtain the features or labels of the records with identical sample IDs from other participants. However, it does not mean it is impossible that one participant can reconstruct the features or labels from other participants in the model training phase. Suppose that $\mathcal{L}$ refers to the loss function of the NN-based VFL model, while the adversarial participants hold features $\mathcal{X}^{\text{adv}}$ and bottom model g with parameters θ_{adv}. Eq. 2 represents the gradient calculation of the adversarial bottom model. It clearly shows that those gradients, i.e., $\frac{\partial \mathcal{L}}{\partial \theta_{\text{adv}}}$ with respect to adversarial participants' bottom model, are associated with the other participants' bottom model output (b_2), top model output and ground-truth label. In other words, the distribution (model parameters) changes in the bottom model are correlated with the features and labels from other participants. This offers an attack surface for the adversarial participants to reconstruct other participants' data samples (features or labels). Therefore, this may lead to serious training data leakage in the VFL training stage.

$$\nabla_{\theta_{\text{adv}}} \mathcal{L} = \frac{\partial \mathcal{L}}{\partial h} \nabla_{\theta_{\text{adv}}} h(b_{\text{adv}}, b_{\text{vict}}; \theta_{\text{top}})|_{b_{\text{adv}}=g(\mathcal{X}^{\text{adv}};\theta_{\text{adv}}); b_{\text{vict}}=g(\mathcal{X}^{\text{vict}};\theta_{\text{vict}})} \tag{2}$$

Additionally, VFL models are widely deployed between large entities with a significant share of overlapping user populations, e.g., banks and e-commerce companies [51]. At the same time, customer data is not only subject to strict government regulations, but it is also an important component of entities' core competitiveness strength. Therefore, it is crucial to analyze the potential training data leakage risks during VFL training. This also enables us to design better privacy-preserving mechanisms for VFL training protection.

B. Threat Model

Similar to prior studies [29], [31], [40], we assume the adversaries to be honest-but-curious participants who can hold the data label or not. In this context, "honest-but-curious" means that the adversarial participants may exploit the known information related to their **own bottom model update** to conduct a data reconstruction attack without deviating from the prescribed training protocols. To carry out *VFLRecon*, the adversaries train an additional model (i.e., a shadow model) with the assumptions categorized by different attack goals, i.e., label and feature reconstructions.

Threat model: In label and feature reconstruction scenarios, the adversaries have the following common requirements and knowledge:

- Only exploit the known information related to the updates of the self-owned bottom models, i.e., inputs, parameters, and gradients w.r.t the self-owned bottom models.
- Knowledge about the whole VFL model architecture, which adheres to the typical training protocols adopted in real-world VFL training pipelines.
- A small dataset consisting of complete data samples (all features and labels), which follow the same distribution as the training dataset. We refer to this dataset as shadow data. In Subsection VI-C, we discuss practical solutions to acquire these data.

In a real-world scenario, e.g., loan risk assessment, a bank, and an e-commerce company may want to collaborate to train a model to assess the potential risk associated with granting a loan to a customer. The personal information held by the bank (i.e., the features) represents a valuable asset that might be of keen interest to the e-commerce company. In addition, the e-commerce company may also be interested in the label

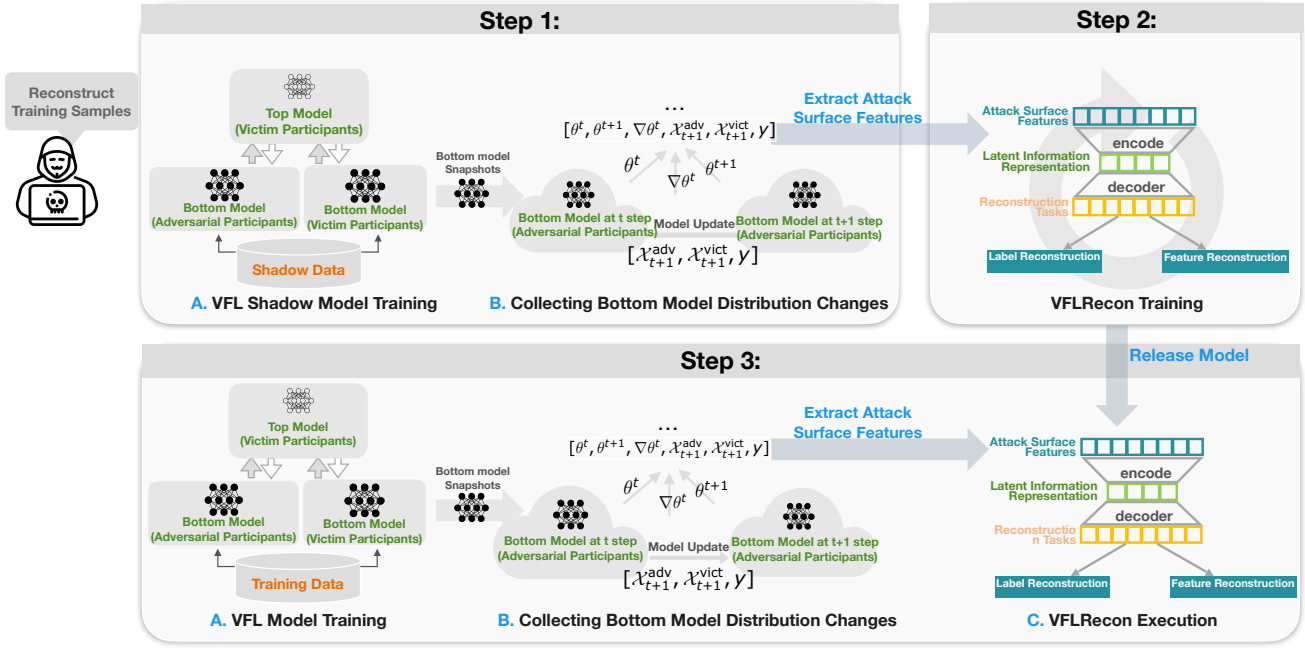

Fig. 2: An overview of *VFLRecon*.

information from the bank. As such, it is reasonable to consider the e-commerce company as a potential adversary with the capability of using *VFLRecon*. More generally, any vertical federated learning application, where data is vertically split, is a candidate for feature and label reconstruction attacks during model training.

C. Algorithm

In this work, we propose an NN-based reconstruction model, $\mathcal{R}(\cdot)$, to reconstruct labels or features from other participants. During VFL model training, the adversarial participants run $\mathcal{R}(\cdot)$ by measuring the posterior difference of the bottom model distributions. We represent the posterior difference of the bottom model distribution using the bottom model output's gradients (δ_g^{adv}), as well as the weights and bottom model outputs before (θ_{adv}, $g(\mathcal{X}^{adv};\theta_{\text{adv}})$) and after ($\theta'_{\text{adv}}$, $g(\mathcal{X}^{adv};\theta'_{\text{adv}})$) bottom model update. In order to model the correlation between those posterior differences and their training samples in two consecutive training steps, we first simulate the VFL shadow model training process to collect the necessary data that depicts the correlation between features or labels of training samples and the bottom model's distribution changes during VFL training. Then, we use the collected data to train an NN-based reconstruction model $\mathcal{R}(\cdot)$ as attackers. The reconstruction loss is defined as:

$$\mathcal{L}_r^{\text{f}} = \|\mathcal{R}(\delta_g^{\text{adv}}, g(\mathcal{X}^{\text{adv}};\theta_{\text{adv}}), g(\mathcal{X}^{\text{adv}};\theta'_{\text{adv}}), \theta_{\text{adv}}, \theta'_{\text{adv}}, \mathcal{X}^{\text{adv}}) - \mathcal{X}^{\text{vict}}\|_2^2 \tag{3}$$

where $\mathcal{R}(\cdot)$ is the reconstruction model that can be an arbitrary NN-based model. Moreover, $\mathcal{X}^{\text{adv}}$ and $\mathcal{X}^{\text{vict}}$ are the raw features of the adversarial participants and victim participants, respectively. Note that Eq. 3 is not suitable for measuring the success of classification tasks. Therefore, in label reconstruction, the loss function is changed as follows:

$$\mathcal{L}_r^{\text{l}} = -\mathbb{E}_y(y \log \mathcal{R}(\delta_g^{\text{adv}}, g(\mathcal{X}^{\text{adv}};\theta_{\text{adv}}), g(\mathcal{X}^{\text{adv}};\theta'_{\text{adv}}), \theta_{\text{adv}}, \theta'_{\text{adv}}, \mathcal{X}^{\text{adv}})) \tag{4}$$

D. Data Reconstruction Attacks

To simplify, we adopt the commonly used framework where adversarial participants own at least one bottom model. Note that *VFLRecon* can be seamlessly adapted to reconstruct features or labels when the adversarial participants only hold the top model. Algorithm 1 describes the whole process of constructing *VFLRecon* to run a specific attack task, reconstructing labels or features from the victim participants. First, the adversarial participants train the VFL shadow model from scratch using the shadow data samples, including complete features and labels. Furthermore, they intentionally record the required information related to the bottom models' distribution change during the shadow model training. After that, the adversarial participants train a reconstruction model, $\mathcal{R}(\cdot)$, using the data collected during the VFL shadow model training. The reconstruction model, $\mathcal{R}(\cdot)$, can be applied to reconstruct training samples' features or labels in realistic VFL training. More specifically, the whole process of the construction and application of $\mathcal{R}(\cdot)$ can be structured into three steps, which are shown in Figure 2.

Step 1: Collecting data for training the reconstruction model. To collect the data for training reconstruction models, the initial step is to collect the data related to the bottom model's distribution changes and reconstruction target. Those data are generated during the shadow VFL model training. We construct a VFL shadow model to mimic the realistic VFL training process and employ the shadow data as training data. Algorithm 1 demonstrates the details of constructing *VFLRecon*.

Algorithm 1 VFLRecon Construction

Input: g: Shadow bottom models. θ_{adv} and θ_{vict} are parameters of adversarial and victim participants' bottom models, respectively;
h: Shadow top model, with parameters, θ_t;
$\mathcal{X}$: Shadow data with complete features and labels, which is a list of tuples ($\mathcal{X}^{\text{adv}}$, $\mathcal{X}^{\text{vict}}$, y);
γ: Learning rate.
Output: $\mathcal{R}(\cdot)$: MLP-based reconstruction model.

1: samples $= \emptyset$
2: **while** $(\mathcal{X}^{\text{adv}}, \mathcal{X}^{\text{vict}}, y) \in \mathcal{X}$ **do**
3: $\quad b_{\text{adv}} = g(\mathcal{X}^{\text{adv}};\ \theta_{\text{adv}})$
4: $\quad b_{\text{vict}} = g(\mathcal{X}^{\text{vict}};\ \theta_{\text{vict}})$
5: $\quad o = h(b_{\text{adv}}, b_{\text{vict}}; \theta_t)$
6: $\quad L = \text{Loss}(o, y)$
7: $\quad \delta_{\text{adv}} = \frac{\partial L}{\partial \theta_{\text{adv}}}$
8: $\quad \theta'_{\text{adv}} = \theta_{\text{adv}} - \gamma \cdot \delta_{\text{adv}}$
9: $\quad b'_{\text{adv}} = g(\mathcal{X}^{\text{adv}};\ \theta'_{\text{adv}})$
10: $\quad$ **if** reconstruction model target is label **then**
11: $\quad\quad$ one record $= \{\frac{\partial L}{\partial b_{\text{adv}}}, b_{\text{adv}}, b'_{\text{adv}}, \theta_{\text{adv}}, \theta_{\text{adv}}', \mathcal{X}^{\text{adv}}, y\}$
12: $\quad$ **else**
13: $\quad\quad$ one record $= \{\frac{\partial L}{\partial b_{\text{adv}}}, b_{\text{adv}}, b'_{\text{adv}}, \theta_{\text{adv}}, \theta_{\text{adv}}', \mathcal{X}^{\text{adv}}, \mathcal{X}^{\text{vict}}\}$
14: $\quad$ **end if**
15: $\quad$ samples $=$ samples $\cup$ one record
16: $\quad$ **Applying SGD to update** $\theta_{\text{adv}}, \theta_{\text{vict}}$ and θ_t
17: **end while**
18: $\mathcal{R} \leftarrow$ MLPModel(samples)
19: **return** $\mathcal{R}(\cdot)$

We first define an empty set of *samples* to store all training records of reconstruction models (Line 1). Next, we iteratively train the VFL shadow model using the complete features and labels (shadow data) (Lines 2 to 17). During model training, we feed the same input $\mathcal{X}^{\text{adv}}$ to the bottom model with parameters before (line 3) and after updating the model (line 9). In addition, we record the data generated during the training process and save them in *samples* (lines 10 to 15).

Step 2: Training the reconstruction model. After we finish the data collection, we use the *collected samples* from step 1 to train an NN-based $\mathcal{R}(\cdot)$ for reconstructing labels or features from other participants during VFL model training (Line 18). We adjust the model output based on the different attack tasks, reconstructing labels or features. As a general rule of thumb, reconstructing the label task takes a sparse vector as the output layer, whereas we take a dense vector as the output layer for reconstructing feature tasks.

Step 3: Executing reconstruction attacks. During the actual VFL model training, the adversarial participants record the data related to their bottom models' changes at each training step to compose the input for $\mathcal{R}(\cdot)$. As *VFLRecon* exploits the changes in the bottom model during training, the adversarial participants are capable of reconstructing training data samples, including features and labels from other participants after participating **only** in one epoch of training.

TABLE II: Overview of datasets.

Dataset	Total samples	Features	Labels
S. Drive Diagnosis	58K	48	11
Criteo	45M	39	2
CIFAR-10	60K	1024	10
BHI	270K	2500	2
Avazu	40M	24	2
CelebA	202K	1024	2

V. EVALUATION SETUP

In this section, we present the experimental setup and metrics to measure the success of *VFLRecon* in reconstructing training samples' features and labels. We further evaluate *VFLRecon* on various datasets ranging from tabular data to images. Moreover, we discuss and analyze the vulnerability of training data protection during VFL training in the last subsection.

A. Experimental Setup

We implement *VFLRecon* with Pytorch and conduct experiments on a server with four 24GB Quadro RTX 6000 GPUs and 512GB RAM running Ubuntu 20.04 LTS. We train the NN-based VFL model in both a general VFL training framework [39] and an encryption-based VFL training framework [56]. The NN-based VFL model consists of bottom models with two hidden layers for each participant, where each hidden layer has 50 units. The top model has two hidden layers, each with 100 units. To reconstruct labels, *VFLRecon* consists of three hidden layers with 1000, 600, and 200 units, respectively. Moreover, *VFLRecon* has three hidden layers with 800, 500, and 100 units, respectively, when it is applied to reconstruct features. To train the NN-based VFL model and *VFLRecon*, we use Adam [24] as an optimizer and "He Uniform" [18] as the initializer. The initial learning rate is set to 0.001. We conduct our label and feature reconstruction experiments on six well-known benchmark datasets, including three tabular datasets (Sensorless Drive Diagnosis, Avazu and Criteo) and three image datasets (CIFAR-10, BHI and CelebA). The overview of our datasets is shown in Table II. We separate the original datasets into two disjointed parts, i.e., a small partial dataset (*shadow* data) and a large partial dataset (normal VFL model training). The VFL shadow model simulates the training process of the VFL model to generate data for *VFLRecon* training using the small amount of *shadow* data. The larger partial dataset is employed for VFL model training, which serves as the target that *VFLRecon* aims to reconstruct.

To better understand the vulnerability of training data protection during a VFL training process, we conduct further experiments in a setting with **encryption-based privacy-preserving** VFL training algorithms [55]. The experiments are conducted with the open-source FATE platform [1].

B. Datasets

In this subsection, we give a brief description of the datasets listed in Table II.

Sensorless Drive Diagnosis is a dataset containing 58,509 data records related to drive signals. Each record has 48 features. The records are categorized into 11 classes.

Avazu is a benchmark dataset for click-through rate (CTR) prediction tasks. It contains around 40 million online ad impressions, each labeled as clicked (1) or not clicked (0). The dataset includes 24 features. In this work, we conduct empirical experiments on 500k data records with balanced sampling from the original data.

CelebA is a large-scale face attributes dataset containing 200k RGB images, which are aligned using facial landmarks. This involves randomly selecting a subset of images, center-cropping them, and resizing them to a resolution of 32×32 for training the models and evaluating the attacks.

Criteo is a public dataset that contains user click histories, which is used for recommendation system tasks. The recommendation scenario is a practical application of VFL. The original dataset contains billions of user records. Limited by our computing resources, we sample 500,000 data records from the original dataset to conduct our analysis.

CIFAR-10 is a well-known label-balanced dataset and contains 60,000 images categorized into 10 classes, each of which consists of 6,000 images.

BHI is a medical dataset that only includes breast cancer images. Each patient's X-rays are distributed among multiple hospitals. We conduct image reconstruction tasks on this dataset.

To conduct reconstruction attacks using *VFLRecon*, we sample a very small amount of data from each dataset, e.g., 1000 records, to generate shadow data that can be accessed by adversarial participants.

C. Evaluation Metrics

To understand the vulnerability of training data protection in VFL training, we use the following metrics to measure how successfully the adversarial participants can apply *VFLRecon* to reconstruct labels or features owned by other participants during VFL model training.

Accuracy is applied to evaluate the performance of label reconstruction. Accuracy calculates the percentage of correctly reconstructed labels from the whole training set.

$$\text{Accuracy} = \frac{\text{the number of correctly classified labels}}{\text{the number of all labels}} \quad (5)$$

Mean Square Error (MSE) is a metric to compare the difference between training features and reconstructed features. We use MSE to measure the performance of the feature reconstruction attack. Suppose that y_i is a ground-truth value, $\hat{y}_i$ is the predicted value, n is the number of records, then the MSE can be calculated as:

$$\text{MSE} = \frac{\sum(y_i - \hat{y}_i)^2}{n} \quad (6)$$

VI. ATTACK EVALUATION

In this section, we evaluate how successfully *VFLRecon* can reconstruct other participants' partial features and labels during VFL model training. In addition, we provide a comprehensive understanding of the vulnerability of training data protection at the VFL training stage. We start by assessing the success of reconstruction attacks on features and labels with six very different benchmark datasets, ranging from tabular data to images. After that, we analyze the potentially significant factors that led to the success of *VFLRecon*. The data and code are available at https://sites.google.com/view/vflrecon/vfl-reconstruct.

A. VFLRecon for Reconstructing Labels

To determine how much label information can be leaked during VFL training, we first randomly sampled a small amount of data from the whole dataset as shadow data. After that, we locally trained an NN-based VFL shadow model and collected the data containing the bottom model snapshots and gradients during model updates. In particular, to discover the vulnerability of training data protection in general VFL training, we conducted experiments on both **non-encryption-based** and **encryption-based** VFL training settings.

To evaluate the effectiveness of *VFLRecon*, we ran our label reconstruction attack experiments on NN-based VFL models on the six datasets presented in Subsection V-B. We utilized accuracy as the metric to evaluate the success of the label reconstruction attacks. Due to the relative absence of related work in VFL privacy research on protecting training data, we employed a common and intuitive approach to formulating a baseline. That is, we reconstruct labels from other participants based on a prediction model trained using shadow data. Specifically, we train a baseline attacker model to predict the labels of the training samples using *shadow* data as training data. The adversary's features serve as input for this baseline attacker, while the training samples' labels are the output. We also compare our approach to prior studies [13] and [29]. [13] proposes one attack approach related to label reconstruction during model training, focusing on the scenario where the top model serves as an aggregation function without any trainable parameters. Similarly, [29] can only be applied to two-party scenarios in which one participant holds labels only, and the other holds features only. To demonstrate the effectiveness of our approach and make a fair comparison, we tailor our approach to their scenarios.

Results: *VFLRecon* can effectively reconstruct labels in different types of datasets, e.g., tabular data and images. Figure 3 shows that *VFLRecon* performs significantly better than the baseline attacks across all datasets, regardless of the data type. The adversarial participants, only owning half of the samples' features, can create a VFL shadow model with 100 complete data samples (including all features). When the batch size is 16 for the VFL shadow model training, the accuracy of label reconstruction is over 85% for all six datasets. Especially, in the two common benchmark datasets Avazu and CelebA, *VFLRecon* can achieve an accuracy of around 90% in label reconstruction. However, as can be seen in the figure, with the increasing complexity of the dataset, the label reconstruction accuracy decreases from 92% (Criteo) to 85% (CIFAR-10).

***VFLRecon* can effectively reconstruct labels in both encryption-based and non-encryption-based VFL training frameworks.** Note that encryption-based training frameworks are considered secure methods to prevent data leakage in the model training stage [44], [58]. However, Figure 3 shows that

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2024.3361813

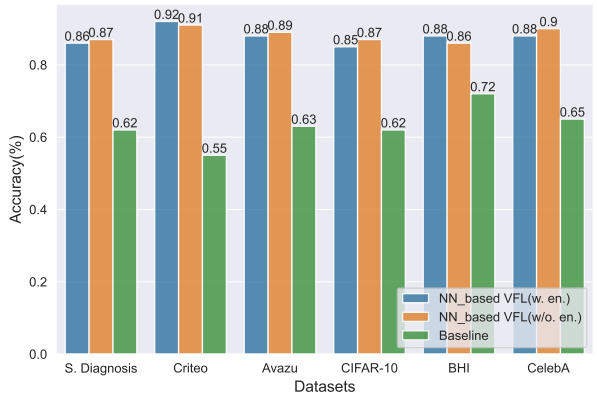

Fig. 3: The label reconstruction attacks on different datasets. S.Diagnosis refers to the Sensorless Drive Diagnosis dataset. "w.en." is the target model trained in an encryption-based VFL training framework, and "w/o." is the model trained in a non-encryption-based VFL training framework.

our approach achieves a very similar performance when reconstructing labels in the **encryption-based** VFL training setting (i.e., an average accuracy of 87.75%) and the **non-encryption-based** VFL training setting (i.e., an average accuracy of 87.75%) for both tabular and image data. The results indicate that **encryption-based** VFL frameworks are **not capable** of preventing label leakage during training. *VFLRecon* effectively reconstructs the labels from other participants. Additionally, *VFLRecon* shows that the existing encryption-based frameworks also suffer from weak training data protection in the VFL training stage.

TABLE III: Label reconstructions over Criteo, Avazu, and CelebA datasets during VFL training.

	Criteo	Avazu	CelebA	Average
Li et al. [29]	88.62%	82.64%	86.49%	85.92%
Ours	91.24%	89.45%	90.08%	90.26%

***VFLRecon* is a more generic approach to measuring the leakage risks of training sample labels.** Table III presents the experimental results for the approach from prior work [29] and our approach. The results show that *VFLRecon* achieves a better accuracy of 91.24%, 89.45%, and 90.08% compared to [29] with an accuracy of 88.62%, 82.64%, and 86.49% in datasets Criteo, Avazu, and CelebA, respectively. Additionally, compared with [13] in the Sensorless Drive Diagnosis dataset, when the top models are non-trainable (only aggregation), the label reconstruction accuracy of [13] can reach 100% while *VFLRecon* reaches 96%. However, when the top models are trainable (which is the common practice), the label reconstruction accuracy from [13] decreases from 100% to 56%, while *VFLRecon* still reaches an accuracy of 92%. We find that when increasing the number of layers in the top model, [13] shows gradually diminishing effectiveness.

Remark: The labels of training samples are prone to leakage to other participants during VFL training. The standard encryption mechanisms applied in VFL training cannot protect those labels.

B. VFLRecon for Reconstructing Features

Training samples, including features and labels, are regarded as a key asset for many organizations. We have shown that our proposed approach, *VFLRecon*, is capable of reconstructing the labels of training samples from other participants during VFL training. Besides effective label reconstruction, to understand how much information about samples' features may be leaked during VFL training, we investigate whether *VFLRecon* can effectively reconstruct the training data features from other participants during VFL training. In other words, we focus on studying whether the bottom model changes disclose information about features from other participants.

To investigate how well the adversarial participants can reconstruct the training data features, we first trained a VFL shadow model to collect the required data introduced in Section IV as the training data of *VFLRecon*. In particular, we assumed that the adversarial participants own half of the features of the training samples during VFL training. Moreover, to examine the essential weakness of training data feature protection in VFL training, we also ran the feature reconstruction in both **encryption-based** and **non-encryption-based** training settings.

The features in the original dataset might be independent or correlated to each other. The correlation between features contains sensitive information about the training samples and poses serious privacy leakage risks. For example, the *income* feature may have a positive correlation with *age* features in a company dataset owned by a VFL participant. If adversaries have prior knowledge about the individuals' *age*, it is easy to infer who earns more than others in that company. Therefore, we also evaluated whether *VFLRecon* can reveal the correlation between features.

Similar to label reconstruction, we assessed feature reconstruction on NN-based VFL models in six different datasets. For the experiments on CIFAR-10, each participant possessed one part of an image. The participants then collaborated to predict the content of the images. The adversaries can apply *VFLRecon* during the collaboration. We used MSE as a metric to measure the success of feature reconstruction attacks.

In line with the label reconstruction evaluation in Subsection VI-A, we took the model that reconstructed the features of other participants based only on the features possessed by the adversarial participants as the baseline. Furthermore, to investigate whether the reconstructed features retained the correlation between features in the original samples, we separately calculated the correlation scores between each pair of features for the original and reconstructed samples.

Results: *VFLRecon* can effectively reconstruct features in both tabular and image data in both encryption-based and non-encryption-based frameworks. Figure 5 shows that our approach has a much lower MSE than the baseline approach in

both VFL training frameworks, indicating the high quality of the reconstructed features. In addition, *VFLRecon* performs well across different datasets, ranging from tabular to image data (see Figure 5), and it performed similarly for **encryption-based** (i.e., an average MSE of 0.03) and **non-encryption-based** (i.e., an average MSE of 0.04) frameworks. The minimum MSE (0.01) is achieved for the Sensorless Drive Diagnosis dataset, in both settings.

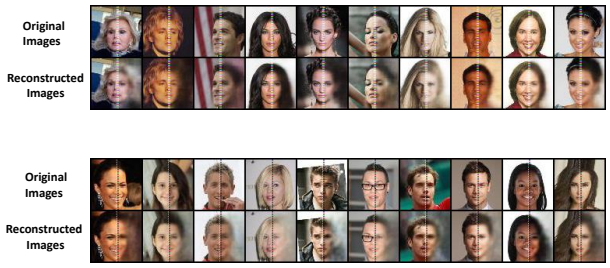

Fig. 4: The visualization of image reconstruction in CelebA.

In general, image reconstruction is more challenging than tabular data reconstruction due to the inherent complexity introduced by the increased feature dimensionality. Nevertheless, our experiments show that *VFLRecon* can faithfully recover images up to a high degree of similarity to their original counterparts. The feature reconstruction MSEs in the **encryption-based** environment are 0.04, 0.03 and 0.01, with the baseline being 0.23, 0.25 and 0.22, in the CIFAR-10, BHI and CelebA datasets, respectively. Figure. 4 visualizes the reconstructed images for the CelebA dataset when adversaries hold half of each image. The models were trained without encryption techniques.

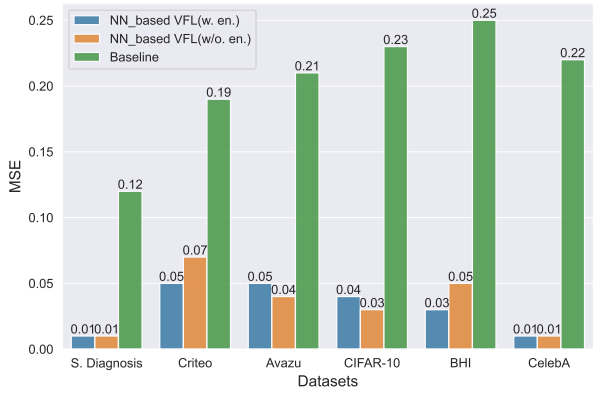

Fig. 5: The feature reconstruction attacks on different datasets. S.Diagnosis refers to the Sensorless Drive Diagnosis dataset. "w.en." is the target model trained in an encryption-based VFL training framework, and "w/o." is the model trained in a non-encryption-based VFL training framework.

***VFLRecon* is able to reconstruct the hidden correlation between features.** Figure 6 depicts the correlation (using the Pearson correlation coefficient) between features in the original dataset and the reconstructed features using *VFLRecon*. As shown in Figure 6, *VFLRecon* can effectively reconstruct the correlations between features. For example, feature 3 has a correlation of -0.45 to feature 4 in the original dataset. In our reconstructed features, the corresponding correlation is -0.25. These results suggest a high utility of the reconstructed features for downstream tasks by the adversary. Furthermore, the reconstructed features provide a potential attack surface for model property inference attacks.

Remark: Training data features can easily leak to adversarial participants during VFL training, and standard encryption mechanisms may be insufficient to prevent such leakage. Additionally, the correlation between the features can be reconstructed with high accuracy.

C. Discussion

In this subsection, we investigate further influencing factors impacting the vulnerability of training data protection in VFL training. The previously illustrated experimental results already reveal that *VFLRecon* can successfully reconstruct labels and features from other participants during VFL training. By more deeply investigating factors influencing such data reconstruction (vulnerability in training data protection), practitioners can better understand the characteristics of training data leakage. Such characteristics can be used to proactively design improved privacy-preserving mechanisms to protect their training data during VFL training.

Potential impacts on the vulnerability of training data protection in VFL training. A prior study [13] reports that the percentage of features, batch size, feature partition strategy, shadow data size, and model update process might impact the label reconstruction performance on a trained VFL model. Therefore, we conducted experiments to investigate if these factors affect the effectiveness of *VFLRecon* on reconstructing labels and features from other participants during model training.

Ablation experiment setup. We first ran our experiments in the NN-based VFL model on the Sensorless Drive Diagnosis dataset. Next, we allowed the adversarial participants to own half of the features. To study how the **percentage of features** affects the weakness of training data protection, we increased the percentage of features owned by the adversarial participants from 5% to 15%, 25%, 50%, and 75% of complete features. For **batch size**, we set up the batch size ranging from 1 to 128. In terms of **number of participants**, we consider multiple participants, i.e., 2, 3, and 4 participants in our experiment. For **feature partition strategy**, we use three different feature partition strategies, i.e., random, Gaussian, and Gibbs partitions. Regarding the **model update process**, we consider three different common optimizations in our experiment, i.e., Adam, SGD and AdaDelta. For **shadow data size**, we conduct further experiments to examine the correlation between our proposed reconstruction attacks and shadow data size. The experiments otherwise use the same setting as reported in Section VI-A. We also applied a similar process to evaluate how successfully *VFLRecon* reconstructs labels and features. Finally, we compared the performance of label and feature reconstruction to understand which factors are important in determining the weakness of training data protection during VFL training in terms of the metrics introduced in Subsection V-C.

TABLE IV: The experiments to explore the effectiveness of *VFLRecon* with different factors, i.e., the number of participants, feature partition strategy, and model updates process in the Sensorless Drive Diagnosis dataset. "w. en." is the model trained in the encryption-based VFL training framework, and "w/o." is the model trained in the non-encryption-based VFL training framework. Recon. refers to reconstruction.

Number of Participants						
	2		3		4	
	Label Recon. Accuracy	Feature Recon. MSE	Label Recon. Accuracy	Feature Recon. MSE	Label Recon. Accuracy	Feature Recon. MSE
NN_based VFL(w. en.)	87.32%	0.01	87.11%	0.01	86.99%	0.01
NN_based VFL(w/o. en.)	86.22%	0.01	85.98%	0.01	85.77%	0.01
Feature Partitions Strategy						
	Radom		Gaussian		Gibbs	
	Label Recon. Accuracy	Feature Recon. MSE	Label Recon. Accuracy	Feature Recon. MSE	Label Recon. Accuracy	Feature Recon. MSE
NN_based VFL(w. en.)	87.32%	0.01	86.99%	0.01	81.61%	0.03
NN_based VFL(w/o. en.)	86.22%	0.01	87.49%	0.01	80.98%	0.03
Model Update Process						
	Adam		SGD		AdaDelta	
	Label Recon. Accuracy	Feature Recon. MSE	Label Recon. Accuracy	Feature Recon. MSE	Label Recons. Accuracy	Feature Recon. MSE
NN_based VFL(w. en.)	87.32%	0.01	88.12%	0.01	86.78%	0.01
NN_based VFL(w/o. en.)	86.22%	0.01	87.49%	0.01	86.96%	0.01

*1) **Percentage of features**:* The experimental results demonstrate that **the more features the adversarial participants hold, the easier they can reconstruct the labels or features of training samples from other participants.** Figure 7 (left part) shows that, when the adversarial participant holds 75% of the features of the complete samples, our approach can achieve an accuracy of 91% with encryption-based VFL training. More importantly, such a high accuracy can be achieved without the need to have a large portion of features. Having only 25% of the features stored by adversarial participants, our approach still achieves a highly efficient attack accuracy of 81%.

Figure 7 (right part) shows the impact of using different percentages of features to conduct feature reconstruction. As expected, if an adversarial participant owns more features during VFL model training, it is easier for the attacker to steal the feature values from other participants. However, the quality of reconstructed features remains stable when the percentage of features held by adversarial participants is higher than 25%.

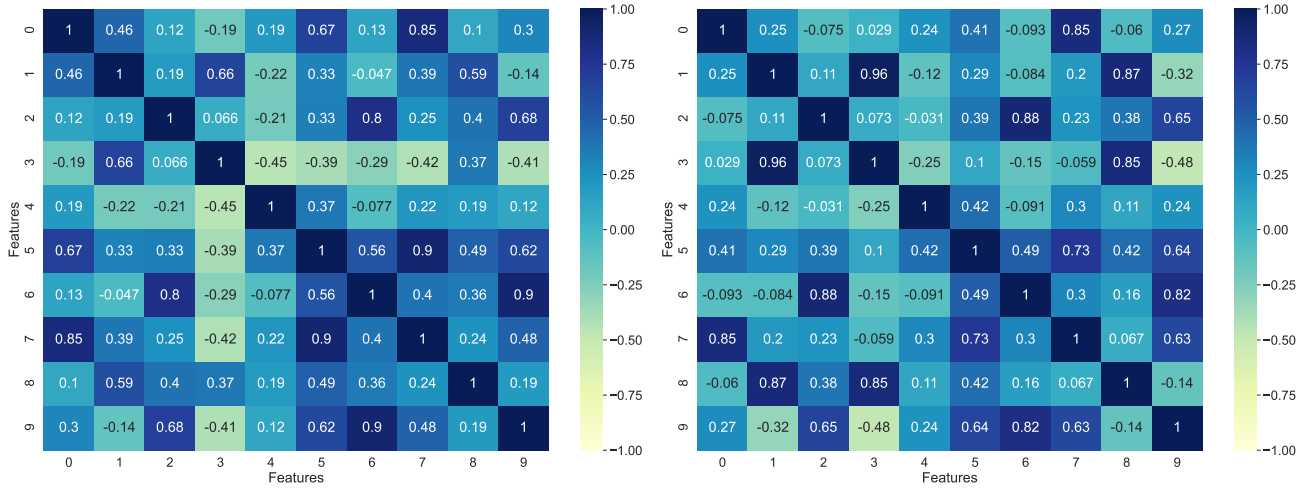

Fig. 6: Visualization of Pearson correlation coefficient for 10 randomly selected features in the Sensorless Drive Diagnosis dataset. The left figure refers to the Pearson coefficient of the features in the original data, while the right figure is the Pearson coefficient of the features in the reconstructed data.

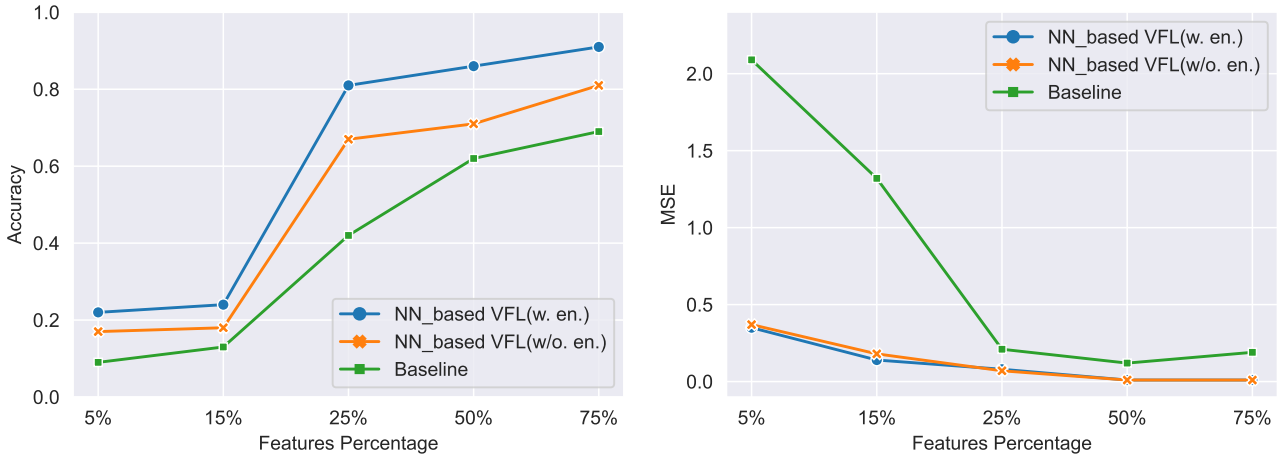

Fig. 7: Effect of adversarial participant's features percentage on feature reconstruction attacks in VFL training on the Sensorless Drive Diagnosis dataset. "w. en." is the model trained in the encryption-based VFL framework, while "w/o." is the model trained in the non-encryption-based VFL training framework.

Even when adversarial participants only hold 25% of the total features, our approach achieves a very low MSE (0.08). As such, without needing a large portion of features at hand, *VFLRecon* can successfully and effectively reconstruct other participants' feature values.

*2) **Batch size**:* **Batch size does not play an important role in data reconstruction attacks.** Regarding the different choice of batch sizes (Figure 8), our results show that the success of *VFLRecon* is rather unaffected by this factor. The majority of the MSE in our approach is less than 0.05 across different batch sizes. For example, *VFLRecon* still achieves an MSE of 0.04 when using a batch size of 128 in the encryption-based VFL model training stage.

*3) **Number of participants**:* Table IV shows experimental results in label reconstruction attacks on the setting with different participants in the Sensorless Drive Diagnosis dataset. The results show that the number of participants has no significant impact on our label reconstruction attack performance in encryption- and non-encryption-based VFL model training.

*4) **Feature partition strategy**:* Table IV shows the performance results using different feature partition strategies. The results show that using an exponential partition strategy, *VFLRecon* achieves the best label reconstruction attack accuracy, i.e., 87.49%, in non-encryption-based VFL model training. Therefore, reasoning about feature partition strategies is important when designing privacy-preserving VFL applications.

*5) **Model update process**:* Table IV shows the results for attack accuracy using three different optimizations. We find that *VFLRecon* achieves a similar attack accuracy, i.e., about 87%, for the three optimizers. Such results imply that the model update process has little impact on *VFLRecon*.

*6) **Shadow data size**:* The experimental results demonstrate that our approach only requires a very small amount of shadow data to conduct effective reconstruction attacks, e.g., 1000 samples (0.2%) in the Criteo dataset containing 500,000 records. It is important to note that as adversaries access more shadow data, the effectiveness of reconstruction attacks increases. When the amount of shadow data surpasses a certain threshold, the improvement of reconstruction effectiveness becomes less pronounced. As previously shown, 1000 samples are enough during attack experiments (Figure 3 and Figure 5) for the six studied datasets with sizes ranging from 58,509 to 500,000. In fact, the actual needed shadow data that can conduct an effective attack maybe even less, as illustrated in Figure 9. It is practical and straightforward to collect such an extremely small amount of shadow data [42], e.g., via model-based synthesis and statistics-based synthesis [12], [57]. Specifically, the adversary can generate a small number of samples without labels based on some strategies and use the inference service to call the trained VFL model (target model) to generate the labels. Moreover, the adversary may also use non-technical strategies such as purchasing a small amount of data from other participants or data brokers directly.

Remark: Several configuration factors, i.e., percentage of features, feature partition strategy and amount of shadow data available to adversarial participants, have a considerable impact on the leakage risks of training samples in the VFL training stage. In contrast, the number of participants and choice of optimizers exert minimal impact on the effectiveness of *VFLRecon*.

VII. DEFENSES AGAINST TRAINING DATA LEAKAGE

Section VI has shown the high potential for leakage of training data in the VFL training stage. In this section, we propose a practical defense strategy.

A. VFLDefender: Preventing Training Data Leakage during VFL Training

To defend against data leakage, we propose a gradients-obfuscation-based approach. With gradients-based model updates, the training samples guide the VFL model to learn the distribution of the training data. Gradients are an effective

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2024.3361813

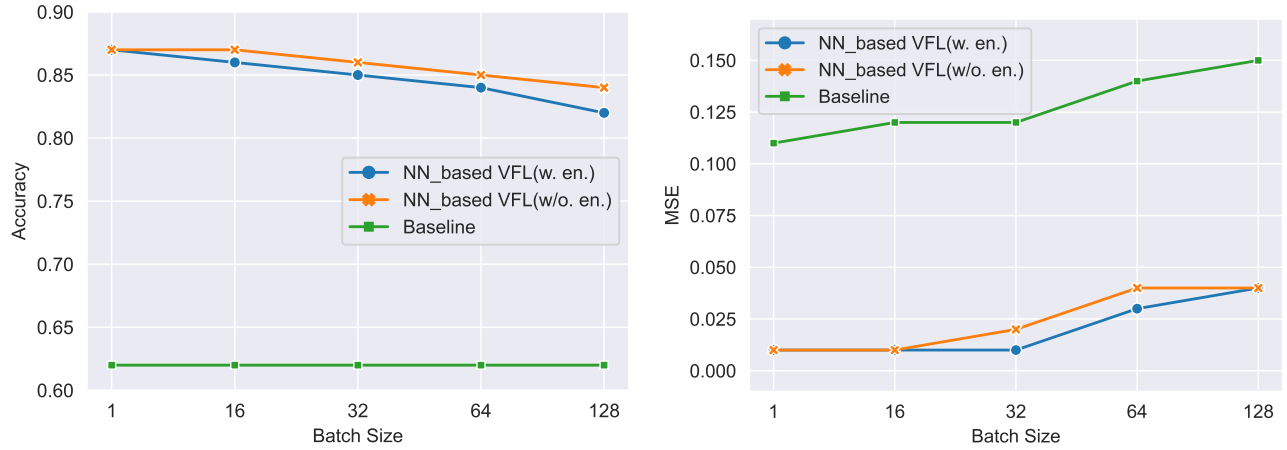

Fig. 8: Labels and features reconstruction in different batch sizes in Sensorless Drive Diagnosis Datasets. S. Diagnosis refers to Sensorless Drive Diagnosis dataset. "w. en." is the model trained in the encryption-based VFL training framework, and "w/o." is the model trained in the non-encryption-based VFL training framework.

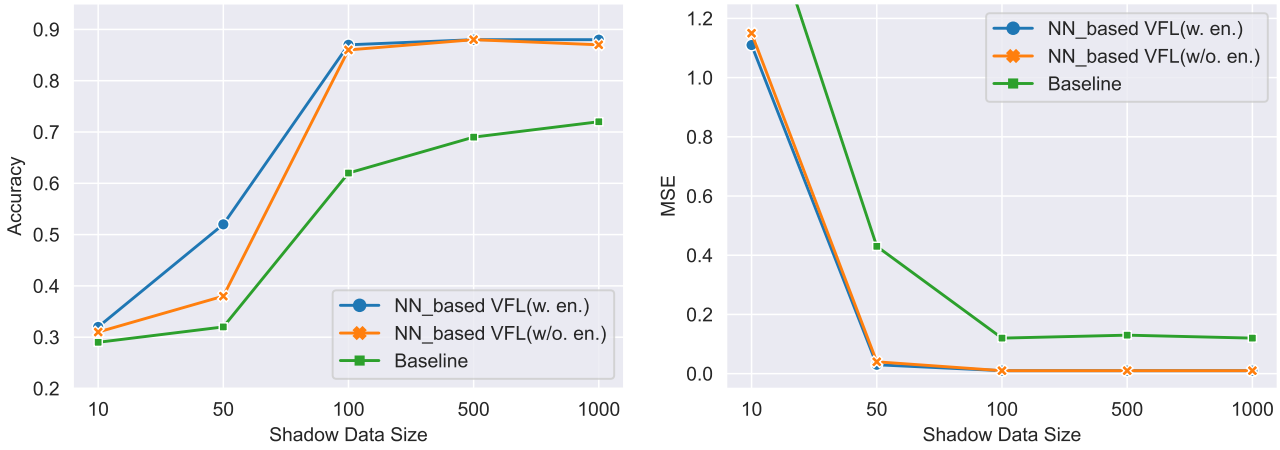

Fig. 9: Label and feature reconstruction in settings with different amounts of necessary shadow data in Sensorless Drive Diagnosis dataset. S. Diagnosis refers to the Sensorless Drive Diagnosis dataset. w. en. is the model trained in an encryption-based VFL training framework, and w/o. is the model trained in a non-encryption-based VFL training framework.

metric to measure how much the distribution changes were caused by the training samples. **If two or more samples produce the same gradients, the correlation between model changes and the training samples becomes weak**. Therefore, we aim to perturb the back-propagated gradients to decrease the correlation between the bottom model's distribution changes and the training samples. Adding random noise to gradients is one of the most common approaches to protecting the information contained in gradients [19], [58]. However, the magnitude of the noise scale has a significant impact on model utility [13], [58]. To ensure model utility, we designed a simple mechanism, *VFLDefender*, to add as little noise as possible to the gradients of the output layer. Our approach is to randomize the norm of the gradients without changing their direction dramatically.

In *VFLDefender*, we employed the same symbols in Eq. 2 to represent the gradients of the output layer: $\delta_o = \frac{\partial L}{\partial h}$. Before adding noise to δ_o, we clip and normalize δ_o to $\hat{\delta}_o$, then reset $\hat{\delta}_o$ in terms of Eq. 7. Note that $\hat{\delta}_o$ is a vector, and $\hat{\delta}_i$ is the i-th element in $\hat{\delta}_o$. $t^{\max}$ and $t^{\min}$ are maximum and minimum clipping thresholds, respectively.

$$\forall \hat{\delta}_i \in \hat{\delta}_o;\ \hat{\delta}_i = \begin{cases} \text{rand}(0, t^{\max}), & \text{if } \hat{\delta}_i >= 0 \\ \text{rand}(t^{\min}, 0), & \text{if } \hat{\delta}_i < 0 \end{cases} \tag{7}$$

Algo. 2 shows the details of our proposed defense algorithm. During VFL model training, each bottom model's owner first feeds their self-owned samples to the models and uploads the output to the top model (lines 1-3). The top model aggregates all bottom model outputs to make a final prediction (line 4). After that, the top model calculates the output layer's gradients (δ_o) in terms of the selected loss function and the ground-truth labels (line 5). Furthermore, the top model clips the δ_o and applies l2-norm-based normalization to transform it into $\hat{\delta}_o$ (lines 6-7). Then, the top model randomizes the norm of $\hat{\delta}_o$ while keeping the gradients' direction unchanged (lines 8-14). After that, the randomized gradients, $\hat{\delta}_o$, are back-propagated

Algorithm 2 VFLDefender

Input: K: The number of bottom models;
g: Bottom models. Each bottom model's parameters are $\theta_i, i = 1, ..., K$;
h: Top model, with parameters, θ_t;
X_1^K: Training data features; it consists of $(X_1, ..., X_K)$; X_i is the features owned by bottom model i;
y: Ground truth label;
γ: Learning rate;
$t^{\max}, t^{\min}$: Maximum and minimum clipping thresholds, respectively.
Output: θ_1^K, θ_t.

1: **for** $k = 1$ **to** K **do**
2: $\quad b_k = g(X_k; \theta_k)$
3: **end for**
4: $o = h(b_1^K; \theta_t)$
5: $L = \text{Loss}(o, y)$
6: $\delta_o = \text{Clipping}(\frac{\partial L}{\partial o};\ t^{\max},\ t^{min})$
7: $\hat{\delta}_o = \text{Normalize}(\delta_o)$
8: **for all** $\hat{\delta} \in \hat{\delta}_o$ **do**
9: $\quad$ **if** $\hat{\delta} > 0$ **then**
10: $\quad\quad \hat{\delta}_{oi} = rand(0, t^{\max})$
11: $\quad$ **else**
12: $\quad\quad \hat{\delta}_{oi} = rand(t^{\min}, 0)$
13: $\quad$ **end if**
14: **end for**
15: **for** $k = 1$ **to** K **do**
16: $\quad \theta_k = \theta_k - \gamma \cdot \hat{\delta}_o \cdot \frac{\partial o}{\partial \theta_k}$
17: **end for**
18: $\theta_t = \theta_t - \gamma \cdot \hat{\delta}_o \cdot \frac{\partial o}{\partial \theta_t}$
19: **return** θ_1^K, θ_t

to each model layer. The bottom and top models update their parameters using the perturbed gradients (lines 15-18).

VIII. DEFENSE EVALUATION

In this section, we present and discuss the evaluation results against training data leakage during VFL model training.

A. Defense Evaluation

We evaluate our defense approach using the Sensorless Drive Diagnosis, CIFAR-10 and Criteo datasets. Specifically, we first apply *VFLDefender* to train the VFL model. During model training, we conduct label and feature reconstruction attacks using the same setting as in Subsection VI-A and Subsection VI-B, respectively. Additionally, to highlight the effectiveness of *VFLDefender*, we first assess the success of *VFLRecon* on label and feature reconstruction with different random noise variance.

Furthermore, we examine whether differential privacy and other privacy-preserving technologies can be applied to prevent data leakage during model training. Specifically, we compare *VFLDefender* with DP-SGD [5] with different privacy budgets (10, 100), and Marvell [29].

Results: Random noise solutions cannot prevent training data leakage from VFL training without substantial model utility loss. Table V shows the results when random noise is added to the output of a top model for the Sensorless Drive Diagnosis dataset. We observe a noticeable relationship between the noise variance and attack performance in the two attack tasks (i.e., label and feature reconstruction). For example, when adding random Gaussian noise with a variance of 0.1, the accuracy of label reconstruction is only 14% and the MSE of feature reconstruction is 1.5. However, the more noise is added, the worse the model's utility becomes. Consequently, the random-noise-based solutions have to be considered ineffective given the increasing model utility loss.

TABLE V: Results of labels and features reconstruction under the protection of random noise solutions for Sensorless Drive Diagnosis dataset. Perf. refers to performance; Acc. refers to accuracy; and MSE refers to mean square error.

	Label Reconstruction			Feature Reconstruction		
Attacker Perf. (baseline)	**62%**			**0.22**		
Attacker Perf. (**our attack, w/o. defence**)	86%			0.01		
Noise Var.	0.001	**0.01**	0.1	0.001	**0.01**	0.1
VFL Model Acc. Loss	-1%	**-30%**	-73%	-1%	**-30%**	-73%
Metrics	Accuracy			MSE		
Attacker Perf. (**our attack, w. defence**)	85%	**57%**	14%	0.019	**0.26**	1.5

Limiting a bottom model's change decreases the vulnerability of training data in VFL training. Applying the *VFLDefender* protection approach, Table VI shows that the attack performance decreases dramatically for the studied datasets. For example, in the dataset of Sensorless Drive Diagnosis, the attack accuracy decreases from 86.22% to 69.48% in terms of label reconstruction. Regarding feature reconstruction, the MSE changes from 0.01 to 0.14. Furthermore, it is important to note that these figures are even close to the attack performance of the baseline approach. These results strongly suggest that *VFLDefender* can decrease the vulnerability of training data. At the same time, limiting a bottom model's change might be expected to decrease the model's utility. However, in our experiments, the VFL model accuracy loss is only around 1%. In contrast, while the experimental results also show that DP is likewise able to protect the privacy of training data, the approach would decrease the model accuracy dramatically (about 35% when privacy budget $\epsilon = 10$).

Remark: Obfuscating the gradients adds uncertainty to the correlation between bottom|top models' distribution change and training samples. *VFLDefender* can efficiently protect the training data during VFL training while maintaining model utility.

B. Discussion

The experimental results in Table V and Table VI show that following the basic approach to add random noise into gradients is possible to prevent training data leakage at the VFL training stage. Such a result is expected since generally injecting noise is a way to perturb the correlation between

TABLE VI: Result of labels and features reconstruction under the protection of *VFLDefender* for Sensorless Drive Diagnosis, Criteo and CIFAR-10 datasets. Recon. refers to reconstruction; Acc. refers to accuracy; and MSE refers to mean square error.

Methods	Sensorless Drive Diagnosis			Criteo			CIFAR-10		
	Acc. Loss	Label Recon. Acc.	Feature Recon. MSE	Acc. Loss	Label Recon. Acc.	Feature Recon. MSE	Acc. Loss	Label Recon. Acc.	Feature Recon. MSE
Baseline	-	62.19%	0.22	-	55.39%	0.19	-	62.49%	0.23
w/o defense	-	86.22%	0.01	-	91.24%	0.07	-	87.18%	0.03
DP-SGD[5] ($\epsilon = 10$)	-34.13%	55.28%	0.21	-38.21%	53.13%	0.18	-35.26%	63.12%	0.22
DP-SGD[5] ($\epsilon = 100$)	-27.55%	57.18%	0.19	-29.29%	56.72%	0.17	-26.39%	64.09%	0.22
Marvell [29]	-2.30%	78.44%	0.09	-2.44%	82.41%	0.09	-3.74%	77.29%	0.07
Our approach	-1.04%	69.48%	0.14	-1.31%	58.27%	0.17	-0.45%	64.47%	0.19

the self-owned bottom model's changes and features or labels of training samples. However, a small amount of noise is not enough to obfuscate those correlations, while a large amount of noise leads to a dramatic model utility decrease (see Table V). Differing from adding random noise, *VFLDefender* aims to add an adaptive noise to the clipped gradients while keeping the gradients' direction unchanged. Therefore, *VFLDefender* can largely preserve the most informative signals in model training while obfuscating the correlation between model changes and target features or labels.

Apart from the abovementioned defense strategies, there are also other possible defenses against training data leakage, e.g., DP. In our defense evaluation, we find that DP can protect the privacy of the training data. However, model accuracy decreases dramatically by 34.13% and 27.55% using DP with a privacy budget of 10 and 100, respectively, in the context of the Sensorless Drive Diagnosis dataset. The performance results for the other studied datasets, Criteo and CIFAR-10, are similar. Such results imply that the DP-based algorithms are not suitable for the studied settings.

Furthermore, our results in Figure 7 show that the accuracy of label reconstruction decreases by about 57% when the percentage of features held by the adversarial participant drops from 25% to 15%. Inspired by this observation, we conjecture that influencing the percentage of features held by the participants may be used to increase the difficulty of reconstruction attacks during VFL training. A possible approach is that the victim participants construct additional useless features within their local data. As these features would not be related to the learning task, their impact on the performance of the final NN-based VFL model would be negligible.

IX. LIMITATIONS

Our evaluation is conducted with six benchmarking datasets with diverse characteristics using NN-based VFL models. Although our studied datasets cover different domains and sizes, our evaluation results may still not generalize to other datasets and other models. Our results in the ablation experiments show that it is easier for adversarial participants who hold more features to reconstruct labels from other participants. Therefore, the success of the attack approach may necessitate a considerable percentage of features. Finally, when participants do not work together to design the final VFL architecture, participants might have no information about the final model architecture. Such missing information may disturb the attack surface. While our approach is both data- and model-agnostic (i.e., it can be seamlessly applied to any type of model and data), further performance advancement may be achieved through a more dedicated design that is tailored for specific model architectures and data modalities.

X. THREATS TO VALIDITY

External threat. A threat to external validity is the generalizability of our approach to statistical-based VFL models. Our study is evaluated on the general NN-based VFL model architecture, i.e., the feed-forward models and six benchmark public datasets. More case studies on other datasets and other non-NN-based VFL models would further improve the evaluation of our approach.

Internal threat. Our work relies on prior knowledge of a small amount of data with the same distribution as the training data. Though we propose a variety of strategies to obtain the shadow data, there are many other feasible approaches. Different shadow data collection approaches may lead to different attack performances and may impact the vulnerability of training data protection.

Construct threat. In the evaluation of possible approaches for mitigating data leakage risks during VFL training, we only study three viable defense strategies. Other possible defense strategies could be explored in future research to complement our evaluation.

XI. CONCLUSION

VFL [20] is an increasingly popular approach to collaborative learning. However, our work offers further evidence that VFL suffers from significant data leakage risks during model training. More specifically, we demonstrate that *VFLRecon* achieves a high accuracy in label reconstruction and a low MSE in feature reconstruction across several studied datasets **even** against **encryption-based** VFL training. We also illustrate the impact of various factors including the amount of features available to the adversarial participants, batch size, shadow data size, and the different domains of datasets. Furthermore, we show that adversarial participants can efficiently train *VFLRecon* with a very small amount of shadow data. To mitigate the vulnerability of training data during VFL training, we propose a defense strategy, *VFLDefender*, to perturb the correlation between model updates (gradients) and training samples. The experimental results reveal that *VFLDefender* is highly effective in preventing training data leakage during VFL training, with an accuracy loss of only around **1%**. Moreover, our work provides valuable insights for VFL system designers on the critical importance of privacy-preserving VFL.

ACKNOWLEDGMENTS

We thank the editors and reviewers for their valuable feedback.

REFERENCES

[1] Heterogeneous neural networks – FATE. https://fate.readthedocs.io/en/latest/federatedml_component/hetero_nn/. (Accessed on 01/02/2022).
[2] Avazu dataset. https://www.kaggle.com/c/avazu-ctr-prediction/data, 2014.
[3] Criteo dataset. https://labs.criteo.com/2013/12/download-terabyte-click-logs/, 2014.
[4] Celeba dataset. https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html, 2016.
[5] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318, 2016.
[6] Martyna Bator. Dataset for sensorless drive diagnosis data set. https://archive.ics.uci.edu/ml/datasets/dataset+for+sensorless+drive+diagnosis, 2015. (Accessed on 12/28/2021).
[7] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.
[8] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1175–1191, 2017.
[9] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, Dimitrios Papadopoulos, and Qiang Yang. SecureBoost: A lossless federated learning framework. *IEEE Intelligent Systems*, 36(6):87–98, 2021.
[10] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The YouTube video recommendation system. In *2010 ACM Conference on Recommender Systems (RecSys)*, pages 293–296. ACM, 2010.
[11] Vasisht Duddu, Antoine Boutet, and Virat Shejwalkar. Quantifying privacy leakage in graph embedding. In *MobiQuitous 2020 – 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 76–85, 2020.
[12] Ju Fan, Tongyu Liu, Guoliang Li, Junyou Chen, Yuwei Shen, and Xiaoyong Du. Relational data synthesis using generative adversarial networks: A design space exploration. *Proceedings of the VLDB Endowment*, 13(11):1962–1975, 2020.
[13] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X. Liu, and Ting Wang. Label inference attacks against vertical federated learning. In *31st USENIX Security Symposium (USENIX Security)*, pages 1397–1414, 2022.
[14] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients – How easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.
[15] Jiahui Geng, Yongli Mou, Feifei Li, Qing Li, Oya Beyan, Stefan Decker, and Chunming Rong. Towards general deep leakage in federated learning. *arXiv preprint arXiv:2110.09074*, 2021.
[16] Oded Goldreich. Secure multi-party computation. Technical report, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, 1998.
[17] Bin Gu, An Xu, Zhouyuan Huo, Cheng Deng, and Heng Huang. Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11):6103–6115, 2021.
[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *IEEE International Conference on Computer Vision (CVPR)*, pages 1026–1034, 2015.
[19] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the GAN: Information leakage from collaborative deep learning. In *2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 603–618, 2017.
[20] Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. In *International Workshop on Federated Learning for User Privacy and Data Confidentiality (FL-NeurIPS)*, 2019.
[21] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 7232–7241, 2021.
[22] Malhar Jere, Tyler Farnan, and Farinaz Koushanfar. A taxonomy of attacks on federated learning. *IEEE Secur. Priv.*, 19(2):20–28, 2021.
[23] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.
[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[25] Jakub Konečnỳ, H. Brendan McMahan, Felix Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.
[26] Igor Kononenko. Machine learning for medical diagnosis: History, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109, 2001.
[27] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Comput. Ind. Eng.*, 149:106854, 2020.
[28] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering (ICDE)*, pages 106–115, 2007.
[29] Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. Label leakage and protection in two-party split learning. In *International Conference on Learning Representations (ICLR)*, 2022.
[30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.
[31] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. Feature inference attack on model predictions in vertical federated learning. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 181–192. IEEE, 2021.
[32] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.
[33] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):3–es.
[34] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (S&P)*, pages 691–706. IEEE, 2019.
[35] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (S&P)*, pages 739–753. IEEE, 2019.
[36] Olga Ohrimenko, Felix Schuster, Cédric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. Oblivious multi-party machine learning on trusted processors. In *25th USENIX Security Symposium (USENIX Security)*, pages 619–636, 2016.
[37] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018.
[38] Thilina Ranbaduge and Ming Ding. Differentially private vertical federated learning. *arXiv preprint arXiv:2211.06782*, 2022.
[39] Daniele Romanini, Adam James Hall, Pavlos Papadopoulos, Tom Titcombe, Abbas Ismail, Tudor Cebere, Robert Sandmann, Robin Roehm, and Michael A. Hoeh. PyVertical: A vertical federated learning framework for multi-headed SplitNN. *arXiv preprint arXiv:2104.00489*, 2021.
[40] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. Updates-leak: Data set inference and reconstruction attacks in online learning. In *29th USENIX Security Symposium (USENIX Security)*, pages 1291–1308, 2020.
[41] Daniel Scheliga, Patrick Mäder, and Marco Seeland. PRECODE – A generic model extension to prevent deep gradient leakage. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3605–3614. IEEE, 2022.
[42] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (S&P)*, pages 3–18. IEEE, 2017.

[43] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 377–390, 2020.

[44] Lili Su and Jiaming Xu. Securing distributed machine learning in high dimensions. *arXiv preprint arXiv:1804.10140*, 2018.

[45] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 2512–2520, 2019.

[46] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H. Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[47] Yuncheng Wu, Shaofeng Cai, Xiaokui Xiao, Gang Chen, and Beng Chin Ooi. Privacy preserving vertical federated learning for tree-based models. *Proceedings of the VLDB Endowment*, 13(12):2090–2103, 2020.

[48] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:cs.LG/1708.07747*, 2017.

[49] Liu Yang, Di Chai, Junxue Zhang, Yilun Jin, Leye Wang, Hao Liu, Han Tian, Qian Xu, and Kai Chen. A survey on vertical federated learning: From a layered perspective. *arXiv preprint arXiv:2304.01829*, 2023.

[50] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[51] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. *Vertical Federated Learning*, pages 69–81. Springer International Publishing, Cham, 2020.

[52] Ruikang Yang, Jianfeng Ma, Junying Zhang, Saru Kumari, Sachin Kumar, and Joel J. Rodrigues. Practical feature inference attack in vertical federated learning during prediction in artificial internet of things. *IEEE Internet of Things Journal*, 2023.

[53] Peng Ye, Zhifeng Jiang, Wei Wang, Bo Li, and Baochun Li. Feature reconstruction attacks and countermeasures of DNN training in vertical federated learning. *arXiv preprint arXiv:2210.06771*, 2022.

[54] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[55] Qiao Zhang, Cong Wang, Hongyi Wu, Chunsheng Xin, and Tran V. Phuong. GELU-Net: A globally encrypted, locally unencrypted deep neural network for privacy-preserved learning. In *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3933–3939, 2018.

[56] Yifei Zhang and Hao Zhu. Additively homomorphical encryption based deep neural network for asymmetrically collaborative machine learning. *arXiv preprint arXiv:2007.06849*, 2020.

[57] Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. Improving sign language translation with monolingual data by sign back-translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1316–1325. Computer Vision Foundation / IEEE, 2021.

[58] Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated Learning*, pages 17–31. Springer, 2020.

Derui Zhu Derui Zhu is a visiting researcher at the University of Alberta and pursuing his doctoral study under the supervision of Prof. Jens Grossklags at the Professorship of Cyber Trust, Technical University of Munich. His research interests mainly focus on the intersection between security, privacy, and software engineering, in particular, trustworthy AI with particular attention given to the vulnerability analysis of data protection in AI models. Contact him at derui.zhu@tum.de.

Jinfu Chen Jinfu Chen is an Associate Professor at School of Computer Science, Wuhan University. His research interest lies in software performance engineering, software performance testing, software log mining, code clone detection and vulnerability detection. He published various high-quality papers at renowned software engineering journals and conferences such as TSE, EMSE, ICSE, FSE, and ASE. He is a recipient of the SIGSOFT Distinguished Paper Award at ICSE 2020. Contact him at jinfuchen@whu.edu.ca.

Xuebing Zhou Dr. Xuebing Zhou leads research and development of Privacy Enhancing Technologies at the Huawei Cyber Security and Privacy Lab. Her work covers privacy-preserving AI, privacy risk assessment, anonymization techniques for smart devices, and transparency-enhancing tools for end users. She received her doctoral degree in the field of privacy protection and biometrics from Technical University Darmstadt. Before she joined Huawei in 2014, she worked at Fraunhofer IGD and the Center for Advanced Security Research Darmstadt. Contact her at xuebing.zhou@huawei.com

Weiyi Shang Weiyi Shang is an Associate Professor at the University of Waterloo. His research interests include AIOps, big data software engineering, software log analytics, and software performance engineering. He serves as a Steering committee member of the SPEC Research Group. He is ranked the top worldwide SE research star in a recent bibliometrics assessment of software engineering scholars. He is a recipient of various premium awards, including the SIGSOFT Distinguished Paper Award at ICSE 2013 and ICSE 2020, Best Paper Award at WCRE 2011 and the Distinguished Reviewer Award for the Empirical Software Engineering journal. His research has been adopted by industrial collaborators (e.g., BlackBerry and Ericsson) to improve the quality and performance of their software systems that are used by millions of users worldwide. Contact him at wshang@uwaterloo.ca.

Ahmed E. Hassan Ahmed E. Hassan (Fellow, IEEE and ACM) received the PhD degree in computer science from the University of Waterloo. He is an ACM SIGSOFT Influential Educator, an NSERC Steacie fellow, the Canada Research Chair (CRC) in Software Analytics, and the NSERC/BlackBerry Software Engineering Chair with the School of Computing, Queen's University, Canada. His research interests include mining software repositories, empirical software engineering, load testing, and log mining. He spearheaded the creation of the Mining Software Repositories (MSR) conference and its research community. He also serves on the editorial boards of IEEE Transactions on Software Engineering, Springer Journal of Empirical Software Engineering, and PeerJ Computer Science. More information at: http://sail.cs.queensu.ca

Jens Grossklags Jens Grossklags is Professor of Cyber Trust in the Department of Computer Science at the Technical University of Munich. His research and teaching activities focus on interdisciplinary challenges in the areas of security, privacy, and technology policy. Grossklags received a Ph.D. in Information Management and Systems from the University of California, Berkeley. He is a Senior Member of IEEE and the ACM. Contact him at jens.grossklags@in.tum.de.