Label Leakage in Vertical Federated Learning: A Survey

Yige Liu[1] , **Yiwei Lou**[1] , **Yang Liu**[2] , **Yongzhi Cao**[1,3,*] and **Hanpin Wang**[1,4]
[1]Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education;
School of Computer Science, Peking University, Beijing, China
[2]Institute for AI Industry Research, Tsinghua University, Beijing, China
[3]Zhongguancun Laboratory, Beijing, China
[4]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China
{yige.liu, cyfqylyw}@stu.pku.edu.cn, liuy03@air.tsinghua.edu.cn, {caoyz, whpxhy}@pku.edu.cn

Abstract

Vertical federated learning (VFL) is a distributed machine learning paradigm that collaboratively trains models using passive parties with features and an active party with additional labels. While VFL offers privacy preservation through data localization, the threat of label leakage remains a significant challenge. Label leakage occurs due to label inference attacks, where passive parties attempt to infer labels for their privacy and commercial value. Extensive research has been conducted on this specific VFL attack, but a comprehensive summary is still lacking. To bridge this gap, our paper aims to survey the existing label inference attacks and defenses. We propose two new taxonomies for both label inference attacks and defenses, respectively. Beyond summarizing the current state of research, we highlight techniques that we believe hold potential and could significantly influence future studies. Moreover, experimental benchmark datasets and evaluation metrics are summarized to provide a guideline for subsequent work.

1 Introduction

Federated learning (FL) [McMahan *et al.*, 2016] is a distributed machine learning paradigm that enables multiple participants to train a model collaboratively. FL can be classified into three different forms based on the different divisions of data characteristics [Yang *et al.*, 2019]: horizontal federated learning (HFL), vertical federated learning (VFL), and federated transfer learning (FTL). In VFL, only one participant (*active party*) owns sample labels (and may also own the features simultaneously), while the other participants (*passive party*) exclusively hold the sample features. The models between the passive and active parties are not accessible to each other. While this setup provides some level of privacy preservation, VFL still faces numerous threats, including reconstruction attacks [Vepakomma *et al.*, 2019; Sun *et al.*, 2021], feature inference attacks [Abuadbba *et al.*, 2020; Pasquini *et al.*, 2021], and label inference attacks (LIA) [Fu *et al.*, 2022a; Liu and Lyu, 2022; Kariyappa and Qureshi, 2023]. Among these challenges, LIA is typically launched by the *honest-but-curious* passive parties who act as attackers attempting to infer labels owned by the active party from gradients or embeddings. LIA is a specific attack in VFL scenarios because the sample features and labels are not simultaneously owned by the participating parties. Therefore, studying LIA under VFL presents an intriguing and interesting challenge.

In addition, extensive research has been conducted on FL, leading to ample surveys covering various aspects such as FL challenges and applications [Wen *et al.*, 2023], communication and computation [Almanifi *et al.*, 2023], security and privacy threats [Rodríguez-Barroso *et al.*, 2023], and others [Kairouz *et al.*, 2021; Soltani *et al.*, 2023]. However, it is important to note that most of these surveys provide an overall perspective on FL but do not delve into VFL aspects. Therefore, despite the widespread use of VFL in the real world, there is a lack of surveys specifically focused on VFL. Liu *et al.* [2024] provided an overall overview of VFL, but unfortunately, the discussion of LIA was limited and several pertinent details were not included.

To fill the existing gap in the comprehensive summary of the VFL-specific label leakage threat and promote further research on VFL, we conduct this survey on label leakage in VFL scenarios. We focus on two aspects: attack and defense, and propose two new taxonomies for each of them. In addition to summarizing existing research, we highlight techniques that we believe hold potential and could significantly influence future studies. To facilitate a fair and effective evaluation of LIA research, we also compile a summary of experimental benchmark datasets and evaluation metrics.

2 Vertical Federated Learning

VFL, a distributed collaborative machine learning paradigm designed for distributed environments, is primarily derived from FL. According to the classification of FL [Yang *et al.*, 2019], HFL participants share the same feature space but have different samples [McMahan *et al.*, 2016; Zeng *et al.*, 2023; Xue *et al.*, 2024], and VFL participants share the same sample or user space but have different features [Gu *et al.*, 2023; Li *et al.*, 2023; Wu *et al.*, 2023b]. Therefore, the VFL architecture typically consists of multiple passive parties with different features and one active party with additional labels. Due to the inaccessibility between VFL embedding models

[*] corresponding author

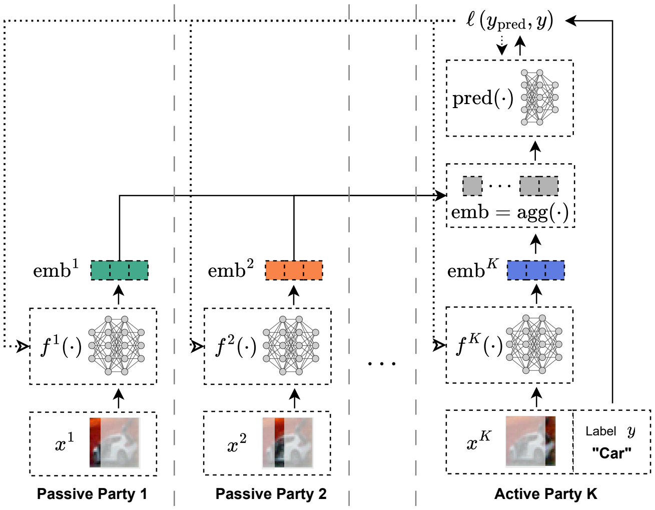

Figure 1: An illustration of VFL. Each participant trains a local model and sends embedding vectors to the active party, which aggregates these embedding vectors and combines them with labels to train a prediction model (shown as solid arrows). The active party then computes and backpropagates the gradients to each participant (shown as dashed arrows) to update the model.

(bottom models) and the prediction model (top model), VFL can offer stronger privacy preservation compared to HFL models with shared parameters.

In addition, it is worth noting that the structure of the VFL model also satisfies the split learning (SL) [Vepakomma *et al.*, 2018] concept. SL decomposes the machine learning model into multiple parts and trains it by multiple participants collaboratively. The classical architectures of SL can be classified into three categories based on the form of data cuts: simple vanilla split learning, U-shaped split learning, and vertical split learning (VSL). However, among these architectures, simple vanilla split learning can be considered a two-party VFL. VSL enables multiple participants with different data partitions to collaboratively train the model, which can be regarded as a special and state-of-the-art VFL method [Bai *et al.*, 2023]. Therefore, in the paper, we will focus on label inference attacks and defenses using the VFL concept.

As shown in Figure 1, VFL [Yang *et al.*, 2019] collaboratively trains a machine learning model with K participants and N samples, including only one *active party* that owns the data labels $\{y_i\}_{i=1}^N$, and multiple *passive parties* with non-label data. Each passive party k owns the different feature datasets $\{x_i^k\}_{i=1}^N$. Note that the active party can also have the feature dataset. Without loss of generality, we assume that all participants have a feature dataset and a local model $f_k(\cdot)$ parameterized by θ_k, while the active party has a prediction model $\text{pred}(\cdot)$ parameterized by φ additionally. Then the collaborative training task can be formulated as:

Forward Propagation. Each participant trains its local model and sends corresponding embedding vectors $\text{emb}_i^k = f_k(\theta_k; x_i^k)$ to the active party. The active party aggregates these embedding vectors into whole embedding $\text{emb}_i = \text{agg}\left(\text{emb}_i^1, \ldots, \text{emb}_i^K\right)$ by a certain aggregation function and trains the prediction model through labels with the loss function as follows:

$$\mathcal{L}(\theta, \varphi; x, y) = \frac{1}{N}\sum_{i=1}^{N} \ell\left(\text{pred}\left(\varphi; \text{emb}_i\right), y_i\right), \tag{1}$$

where $\ell(\cdot)$ denotes the sample loss such as cross-entropy loss and mean squared error loss.

Backpropagation. The active party backpropagates the gradient $\nabla_{\text{emb}_i^k}\mathcal{L} = \frac{\partial \mathcal{L}}{\partial \text{emb}_i^k}$ to each participant and gradient $\nabla_\varphi \mathcal{L}$ to its own. With the chain rule, each participant computes the gradient of the local model and both the local and prediction models update the parameters as shown below:

$$\nabla_{\theta_k}\mathcal{L} = \frac{\partial \mathcal{L}}{\partial \text{emb}_i^k} \cdot \frac{\partial \text{emb}_i^k}{\partial \theta_k} = \nabla_{\text{emb}_i^k}\mathcal{L} \cdot \frac{\partial \text{emb}_i^k}{\partial \theta_k}, \tag{2}$$

$$\theta_k^{t+1} = \theta_k^t - \eta \cdot \nabla_{\theta_k}\mathcal{L},\ \varphi^{t+1} = \varphi^t - \eta \cdot \nabla_\varphi \mathcal{L}, \tag{3}$$

where t denotes the iteration and η denotes the learning rate.

3 Label Inference Attacks

As a specific attack designed for VFL scenarios, LIA is typically initiated by the passive parties to infer the label for its privacy or commercial value. The security assumptions of LIA can be defined as follows.

Threat Model. In most cases, LIA assumes that all passive parties are *honest-but-curious* and do not collude with each other, and they can not access the labels of the active party. They possess different local datasets and corresponding indexes, but they are unable to modify the orders. So they need to follow the VFL protocol mentioned in Section 2 to collaboratively train the model. However, these passive parties can also act as attackers and attempt to infer labels using the available information, such as embeddings, gradients, and a limited amount of auxiliary data. In the worst case, they can even use a malicious local optimizer to maximize their attack performance [Fu *et al.*, 2022a]. Different LIA approaches have different auxiliary data assumptions (e.g., labeled samples and label distribution) that will be introduced later. Moreover, attacks can be launched at any epoch or iteration of VFL, including the training and inference phases.

In this section, we propose a new taxonomy of LIA and provide a comprehensive summary of existing LIA approaches. The taxonomy first separates LIA into two categories based on the VFL models, including the neural networks and other models (tree-based models and regression models). Since most LIAs focus on neural networks, we further classify LIA on neural networks into the attack approaches it uses. This taxonomy provides a clear overview of the existing LIA research landscape. In addition, we explore the related techniques and works in this field and hope to inspire future research on LIA.

3.1 Attacks on Neural Networks

Neural networks serve as the most commonly utilized model in VFL, making them a prime target for LIA. It is worth mentioning that besides the classification presented below, these attacks can be further subdivided based on the type of neural network employed, such as CNN [Kariyappa and Qureshi, 2023; Liu and Lyu, 2022] and GNN [Arazzi *et al.*, 2023], as well as the specific task they aim to exploit, including classification, recognition [Liu and Lyu, 2022; Zheng *et al.*, 2022; Kariyappa and Qureshi, 2023], and recommendation [Fu *et al.*, 2022a; Sun *et al.*, 2022; Kariyappa and Qureshi, 2023].

LIA with Gradient Sign and Magnitude

During the training of VFL models, the most direct label-related information attackers can obtain is the gradients backpropagated by the active party. Therefore, an intuitive approach is to infer the label from them. Zhao *et al.* [2020] were the first to discover and propose the following property regarding the relationship between gradient signs and labels:

Property 1. *In the classification task, when the last layer of the neural network is predicted using the softmax function and cross-entropy loss function with one-hot labels, only the gradient corresponding to the ground-truth label in the last layer has a negative sign, while the gradients for the other labels are positive.*

Based on the aforementioned findings, a series of studies [Liu *et al.*, 2020; Wainakh *et al.*, 2022; Wainakh *et al.*, 2021; Zhang *et al.*, 2022; Fu *et al.*, 2022a; Liu and Lyu, 2022; Zou *et al.*, 2022] have utilized Property 1 to implement LIA in VFL scenario.

Notably, for VFL, the gradient of the active party backpropagation can be written as:

$$\nabla_{\mathrm{emb}_i}\mathcal{L} = \frac{\partial \mathcal{L}}{\partial z_i} \cdot \frac{\partial z_i}{\partial \mathrm{emb}_i} = g_i \cdot \frac{\partial z_i}{\partial \mathrm{emb}_i}. \tag{4}$$

To satisfy Equation (4), no network structure should be included between embedding vectors emb_i and the logit z_i. Otherwise, $\frac{\partial z_i}{\partial \mathrm{emb}_i}$ will contain parameters that are not accessible to the passive party. An alternative approach is that the attacker can directly access the gradients of the last layer, but this is a strong assumption for LIA. Furthermore, it is worth noting that this approach has another limitation: the attack can only be performed during the training phase since there are no gradients available in the inference phase.

In addition, Wainakh *et al.* [2021] found the relationship between gradients and labels from the batch perspective:

Property 2. *The gradient magnitude correlates with the number of label occurrences in the batch of training data and is independent of the label types.*

For example, g_i denotes the gradient corresponding to label i. One occurrence of label i in the training batch results in a gradient value $g_i - \alpha$, so two occurrences of label j result in a gradient value $g_j - 2\alpha$. Based on Properties 1 and 2, Wainakh *et al.* [2021] proposed the LLG algorithm to implement the LIA for batch training.

Except for gradient sign and gradient magnitude, Aggarwal *et al.* [2021] found that for any finite number of label classes, the labels of a dataset can be accurately inferred from the reported log-loss scores of a single carefully constructed prediction vector if arbitrary precision arithmetic is allowed. They also proposed some practical attack algorithms that can infer labels without the need for model training. These algorithms leverage concepts from number theory and combinatorics, offering a novel approach to label inference.

LIA with Classification and Cluster

Considering that the gradient can reflect the label information, attackers can utilize the backpropagated gradients to train a gradient classifier. Meanwhile, by combining relevant auxiliary data, they can infer the labels corresponding to different gradients. Li *et al.* [2022] focused on LIA in the two-party scenario. They observed that during training, the model tends to be less confident about a positive example being positive than a negative example being negative. This observation is reflected in the gradients, as the norm of the gradient corresponding to the target label is typically larger than that of the non-target label. Therefore, it is possible to infer whether a gradient is derived from the target label by training an adversarial gradient binary classifier.

Additionally, based on the relationship between the forwarded embeddings and the gradients of the backpropagation, Bai *et al.* [2023] proposed an algorithm to infer whether a sample belongs to the target label. They argued that for a well-trained local embedding model, sending an unchanged embedding will result in a smaller gradient, while a larger gradient indicates that an embedding with a non-target label was sent. Therefore, combined with a small amount of auxiliary data with known labels, the gradient classifier can be obtained by replacing the embedding to infer label categories.

Furthermore, apart from gradients, labels can also be leaked through forward-propagating embeddings, which was first noticed by [Sun *et al.*, 2022]. They employed the spectral attack [Tran *et al.*, 2018] (a singular value decomposition based method) to differentiate the embedding distribution between target and non-target samples. By leveraging known label distributions or data imbalances, they assigned labels based on the differentiated embedding distribution.

Another similar approach considers the spatial distribution characteristics of gradients and embeddings. It observes that gradients with the same labels tend to be close to each other, while gradients with different labels are far apart. This clustering pattern also holds for embeddings of well-trained models, indicating that both gradients and embeddings exhibit clustering characteristics. In the studies conducted by [Liu and Lyu, 2022; Arazzi *et al.*, 2023], different clustering algorithms for LIA were employed to perform gradient or embedding classification. Specifically, Liu and Lyu [2022] used the *K-means* algorithm and proposed the cosine similarity metric for gradients and Euclidean distance for embeddings. They also demonstrated that these two metrics are interchangeable under normalized data samples to develop a unified attack for both gradients and embeddings to improve LIA efficiency. On the other hand, Arazzi *et al.* [2023] first investigated LIA in VFL scenarios using a zero-background knowledge strategy with a particular focus on GNN structures. They leveraged an unsupervised learning algorithm to identify optimal clusters for embeddings generated by the attacker-controlled local model when the number of classes is unknown.

LIA with Model Reconstruction

Model reconstruction, also known as gradient inversion, was initially proposed in the HFL scenario for recovering the raw training data, with DLG [Zhu *et al.*, 2019] being the most prominent method. This approach is also applicable in VFL scenarios. For model reconstruction-based LIA, the attacker simulates the active party's prediction model $\mathcal{M}$ and the ground-truth label y by constructing the surrogate model $\mathcal{M}'$ and surrogate label y'. The attacker then trains the surrogate model $\mathcal{M}'$ to minimize the difference between its loss gradient $\nabla_{\mathrm{emb}}\mathcal{L}'$ and the real gradient $\nabla_{\mathrm{emb}}\mathcal{L}$ and updates the surrogate label to achieve the inference.

In [Zhang *et al.*, 2022; Erdoğan *et al.*, 2022; Zou *et al.*, 2022; Arazzi *et al.*, 2023; Kariyappa and Qureshi, 2023], model reconstruction was employed to infer labels. In addition, Zhang *et al.* [2022] proposed to construct surrogate labels with *Label Smoothing* instead of one-hot labels. It avoids the model being overconfident in label predictions. For example, when considering categories, *planes* are correlated with *birds* but differ from *tables*. However, using one-hot labels would treat *planes* as equally different from both *birds* and *tables*, which does not accurately reflect the real distribution.

Furthermore, Arazzi *et al.* [2023] found a correspondence between the peak in gradient magnitude and the significant drop in attack accuracy. To address this, they proposed *Early Stopping Strategy* to enhance the performance of the attack. The strategy aims to identify the optimal point at which to stop the attack process, improving the overall attack accuracy.

Moreover, Kariyappa and Qureshi [2023] introduced a novel loss function called *ExPloit Loss* for model reconstruction. This loss function satisfies the following optimization objectives when the real gradient and the ground-truth label distribution are known: 1) ensuring that the gradient of the surrogate model closely matches the real gradient; 2) aligning the surrogate label distribution with the ground-truth distribution; 3) minimizing the entropy of surrogate labels to resemble one-hot type ground-truth labels; and 4) achieving high prediction accuracy for the surrogate model, where the predicted labels closely match the surrogate labels. Due to this loss function, the attacker can infer the label by optimizing the surrogate model and label.

LIA with Model Completion

Model completion is a fine-tuning approach used in VFL. At the end of VFL training, passive parties are given a well-trained local embedding model (bottom model). The embeddings generated by this model exhibit a strong correlation with the labels. Leveraging this correlation, attackers can perform fine-tuning on the bottom model using a small amount of auxiliary labeled data, which enables the model to predict or infer labels more accurately.

According to [Fu *et al.*, 2022a], the attacker can employ a specially designed optimizer to train the local model. This optimizer ensures that improved embeddings are sent to the active party in each iteration, thereby increasing the dependency of the prediction model (top model) on the bottom model and enhancing the accuracy of the attack. Additionally, the study found that by abusing the bottom model, the attacker can even infer labels outside of the training dataset. Furthermore, Zheng *et al.* [2022] explored defense methods against such an attack approach, which will be discussed in detail later.

3.2 Attacks on Other Models

Except for neural network models, LIA has also been studied in regression models and tree-based models.

In the context of regression models, Tan *et al.* [2022] proposed the attacker can utilize the residual variables to infer labels in logistic regression models. Specifically, the residual variable is calculated by solving a system of linear equations constructed from the local dataset and the received decrypted gradients. Xie *et al.* [2023] focused on LIA against continuous labels instead of discrete types. In their study, the target labels are scores rather than categories, and they employed model reconstruction for label reconstruction and inference.

In the context of tree-based models, Takahashi *et al.* [2023] conducted the attack by extracting graph structures from the data records used to train the tree-based model. They then applied community detection to cluster the learned graphs, where clusters with the same labels imply the same labels.

3.3 Limitations and Future Directions

A successful LIA relies not only on the approaches mentioned above but also on the utilization of important techniques. Therefore, this section also presents some related works and techniques in the field, which may inspire future directions.

Auxiliary Data. Almost all approaches use auxiliary data when implementing LIA, whose types can be summarized as:

- Small amount of data with known labels (which can be involved in training) [Wainakh *et al.*, 2021; Liu and Lyu, 2022; Fu *et al.*, 2022a; Xie *et al.*, 2023].
- Prior distribution of labels [Sun *et al.*, 2022; Kariyappa and Qureshi, 2023].
- Number of label classes [Kariyappa and Qureshi, 2023].
- Imbalanced or biased labels (e.g., the percentage of people with a certain disease in the natural population is almost always much lower than 50%) [Li *et al.*, 2022; Sun *et al.*, 2022].

Different quantities and qualities of auxiliary data typically imply different levels of security assumptions for LIA, with a lesser dependence on auxiliary data indicating more robust LIA capabilities. We recommend that future research should incorporate more realistic auxiliary data, and we also advocate for considering this factor during comparisons.

LIA from Embeddings. The majority of current LIA approaches primarily rely on backpropagated gradients and often overlook the potential leakage of labels through embeddings generated by well-trained local models. However, in the case of well-trained local models, embeddings sent by passive parties can also reveal label information. For instance, as mentioned earlier, Sun *et al.* [2022] employed the spectral attack to classify embeddings to infer label categories. Additionally, Zheng *et al.* [2022] devised a specific loss function for embeddings to prevent them from clustering and inadvertently leaking labels.

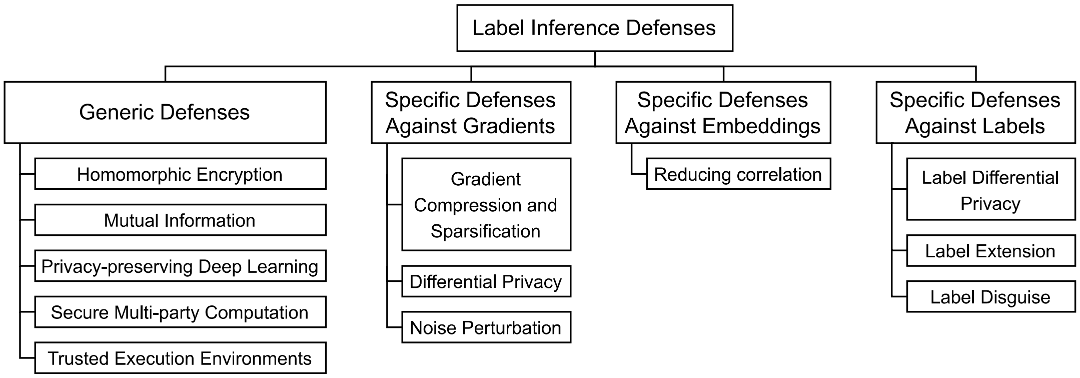
Label Inference Defenses
Generic Defenses
Homomorphic Encryption
Mutual Information
Privacy-preserving Deep Learning
Secure Multi-party Computation
Trusted Execution Environments
Specific Defenses Against Gradients
Gradient Compression and Sparsification
Differential Privacy
Noise Perturbation
Specific Defenses Against Embeddings
Reducing correlation
Specific Defenses Against Labels
Label Differential Privacy
Label Extension
Label Disguise

Figure 2: Taxonomy of label inference defenses.

Learning Rate Adjustment. In VFL scenarios, the attacker is typically the party with less influence or control. However, in specific LIA approaches (model completion [Fu *et al.*, 2022a] and embedding replacement [Bai *et al.*, 2023]), the attack accuracy can be effectively increased by using malicious local optimizer adjusting the learning rate to make the top prediction model more dependent on the attacker's bottom model. Therefore, when the attacker performs LIA using the embeddings generated by the bottom model (e.g., cluster and model completion), it will get better attack performance.

Backdoor Attacks. Backdoor attacks aim to train a model with a hidden backdoor implant that behaves normally on clean inputs and misclassifies inputs that contain *triggers*. In certain backdoor attacks against VFL [Bai *et al.*, 2023; Naseri *et al.*, 2023], attackers only want to add triggers to data with the target label and not change other data. To achieve this, they first need to infer the labels, which means that LIA is sometimes employed as a part of backdoor attacks.

4 Defenses

There are various defenses to mitigate the label leakage threat. In this section, we propose a new taxonomy of them. It first categorizes these defenses into four main types: *generic defenses* and *specific defenses* against gradients, embeddings, and labels. Then we classify these defenses into particular techniques as demonstrated in Figure 2. Specifically, generic defenses are not only designed for LIA, but also can generally defend against other attacks in VFL scenarios, such as feature inference attacks, reconstruction attacks, and sample ID attacks. In contrast, specific defenses are tailored to address different attack approaches used in LIA. These defense strategies primarily focus on safeguarding components that have the potential to leak labels, such as gradients, embeddings, and even labels per se.

4.1 Generic Defenses

Commonly used generic defenses are summarized and recalled in this section.

Homomorphic Encryption. Different from conventional encryption, homomorphic encryption (HE) is a special cryptosystem that supports arithmetic operations on ciphertexts and ensures that the ciphertext computation after decryption is the same as the plaintext computation. In VFL, HE serves as a defense mechanism that prevents attackers from exploiting information in gradients or embeddings [Mohassel and Zhang, 2017; Fu *et al.*, 2022b; Tan *et al.*, 2022; Zhou *et al.*, 2022]. Cheng *et al.* [2021] focused on addressing the privacy concerns associated with tree-based models and proposed the *SecureBoost* algorithm to protect the privacy of each participant.

Mutual Information. The essence of LIA is to exploit the mutual information among labels, gradients, and embeddings. Therefore, an intuitive defense is to eliminate this mutual information. Takahashi *et al.* [2023] proposed a defense mechanism called *ID-LMID* to reduce the mutual information between labels and instance spaces. In addition, some specific defenses against raw data and intermediate outputs [Zou *et al.*, 2023], gradients [Zou *et al.*, 2022], embeddings [Sun *et al.*, 2022; Erdoğan *et al.*, 2022], and labels [Zou *et al.*, 2022; Qiu *et al.*, 2023] can also be considered as mutual information-based methods.

Privacy-preserving Deep Learning. Privacy-preserving deep learning (PPDL) [Shokri and Shmatikov, 2015] is a comprehensive privacy-enhancing method that contains three defense strategies: differential privacy, gradient compression, and random selection. In [Fu *et al.*, 2022a; Bai *et al.*, 2023], they utilized this method to defend against LIA. However, it is important to note that while PPDL effectively protects privacy, it may also lead to a reduction in the model's quality due to the application of DP and gradient compression.

Secure Multi-party Computation. Secure multi-party computation [Knott *et al.*, 2021] leverages cryptographic techniques to facilitate private computation of distributed data held by multiple parties. These techniques involve cryptographic primitives such as secret sharing [Mohassel and Zhang, 2017; Zhou *et al.*, 2022] and homomorphic encryption. Unfortunately, due to their substantial computation and communication overheads, these methods are not commonly employed in VFL scenarios.

Trusted Execution Environments. Trusted execution environments (TEE) [Mo *et al.*, 2021] establish a secure region in the central processor using a combination of hardware and software techniques. This ensures the protection of programs and data loaded inside fulfill confidentiality and integrity. However, TEE is primarily designed for CPU, which can result in underwhelming performance when it comes to VFL computation on GPU. Additionally, the specialized hardware required for TEE implementation can increase costs and potentially limit the generalizability of the solution.

4.2 Specific Defenses against Gradients

Exploiting the properties of gradients (such as gradient sign and magnitude) and the correlation between gradients and labels is a widely used approach in LIA. As a result, numerous defenses against gradients have been derived.

Gradient Compression and Sparsification. Gradient compression is a defense strategy aimed at improving communication efficiency and preserving privacy. One specific technique within gradient compression is gradient sparsification, which involves sharing only a subset of the most important gradients, typically those with the largest absolute values. In various studies [Wainakh *et al.*, 2022; Zou *et al.*, 2022; Fu *et al.*, 2022a; Liu and Lyu, 2022; Arazzi *et al.*, 2023; Bai *et al.*, 2023], researchers have employed this strategy and evaluated parameters like compression ratio and sparsification ratio. However, the experimental results indicate that gradient compression and sparsification have a negative impact on the model's quality and are not particularly effective in defending against LIA. This is because the compressed or sparsified gradients still retain the most significant features, which are often highly correlated with the labels.

Differential Privacy. Differential privacy (DP) [Dwork, 2006] is a widely used privacy-preserving technique that limits the impact of individual data on the overall dataset within a certain range, thereby mitigating the risk of differential attacks. In the context of FL, the DP-SGD algorithm [Abadi *et al.*, 2016] is a famous approach. It involves clipping the gradient and adding noise that satisfies DP during model updating. Researchers in [Liu and Lyu, 2022; Bai *et al.*, 2023] have successfully applied this algorithm to reduce the risk of privacy leakage from gradients. Additionally, the shuffle method [Erlingsson *et al.*, 2019; Cheu *et al.*, 2019] can be used to achieve privacy amplification for DP.

Noise Perturbation. Adding noise to the gradient [Li *et al.*, 2022; Wainakh *et al.*, 2022; Zhang *et al.*, 2022; Fu *et al.*, 2022a; Arazzi *et al.*, 2023; Xie *et al.*, 2023; Kariyappa and Qureshi, 2023] is an intuitive way to perturb the gradient. These noises can alter the distribution of the gradients and in some cases, even change the gradient sign, directly impacting LIA. Moreover, the addition of noise is often associated with DP, with Gaussian and Laplace noise being commonly used. Although these noises may reduce the accuracy of the model, they serve as effective defenses against LIA.

4.3 Specific Defenses against Embeddings

For the well-trained models, since embeddings and labels are highly correlated, an intuitive defense is to reduce this correlation. Inspired by the electrostatic equilibrium and Coulomb's law, particularly the phenomenon of mutual repulsion between like charges (electric charges of the same sign), Zheng *et al.* [2022] proposed a potential energy loss function. This loss function makes it challenging for an attacker to finetune the bottom model using a small number of samples to infer labels. In addition, Sun *et al.* [2022] proposed an additional optimization objective $dCor\left(f\left(X\right), Y\right)$ for the active party, which aims to reduce the distance correlation between the embedding and the label. To address the potential data leakage through embeddings, Vepakomma *et al.* [2019] introduced an additional loss function during training: the logarithm of the distance correlation (DCOR) [Székely *et al.*, 2007] between the input and the embedding, which was also used in [Pasquini *et al.*, 2021; Erdoğan *et al.*, 2022].

4.4 Specific Defenses against Labels

The variability of the labels per se plays a crucial role in information leakage, which has prompted research efforts to develop defenses specifically targeting labels.

Label Differential Privacy. Label DP is a relaxation of DP in which only the privacy of the labels needs to be protected, while the features are public or non-sensitive. Ghazi *et al.* [2021] used randomized responses to flip the labels and used the generated noisy labels to calculate the loss for training. They demonstrated that their algorithm can achieve label DP. In contrast, Wu *et al.* [2023a] investigated the connection between the label DP and LIA. Their study showed that it is not reasonable to equate label DP privacy with limiting the accuracy of LIA. In other words, high privacy does not imply the infeasibility of label inference. Similarly, the study by [Busa-Fekete *et al.*, 2021] suggests the idea that label DP may be more vulnerable than initially thought.

Label Extension. Label extension [Qiu *et al.*, 2023] is a method used to obfuscate the label information contained in the gradient by extending the labels. This prevents attackers from using gradients to reconstruct the label inference models. Qiu *et al.* [2023] proposed the random label extension (RLE) method, which simply extends the original labels with random vectors. Additionally, they proposed a model-based adaptive label extension (MLE) method. In MLE, the dimension where locating the original label is designed to dominate the training process, which aims to improve the performance of the original task when applying the defense.

Label Disguise. Label disguise [Zou *et al.*, 2022] is a method that transforms original labels to *soft fake labels* on the active party. It has an encoder-decoder structure that maps the original labels into the fake label space via a confusional autoencoder and recovers the labels from the fake labels via a decoder. Zou *et al.* [2022] devised a specific loss function for this structure, which incorporates the following objectives: 1) the fake labels should be less relevant to the original labels; 2) the recovered labels should be close to the original labels; and 3) the fake labels entropy should be large since high entropy implies low confidence in the prediction.

5 Experimental Datasets and Metrics

To guide the evaluation of future LIA approaches, we provide a summary of the experimental benchmark datasets and evaluation metrics in this section.

Models	Tasks	Datasets	Features	# Training Samples	# Testing Samples	Cited Literature
Neural Network	Classification & Recognition	MNIST [LeCun *et al.*, 1998]	28×28	60,000	10,000	[Wainakh *et al.*, 2021], [Wainakh *et al.*, 2022], [Liu and Lyu, 2022], [Zhang *et al.*, 2022], [Zou *et al.*, 2022], [Zheng *et al.*, 2022], [Erdoğan *et al.*, 2022], [Bai *et al.*, 2023]
		Fashion-MNIST [Xiao *et al.*, 2017]	28×28	60,000	10,000	[Liu and Lyu, 2022], [Zheng *et al.*, 2022], [Erdoğan *et al.*, 2022], [Kariyappa and Qureshi, 2023]
		CIFAR-10 [Krizhevsky, 2009]	$32 \times 32 \times 3$	50,000	10,000	[Liu and Lyu, 2022], [Zou *et al.*, 2022], [Fu *et al.*, 2022a], [Zheng *et al.*, 2022], [Erdoğan *et al.*, 2022], [Bai *et al.*, 2023], [Kariyappa and Qureshi, 2023]
		CIFAR-100 [Krizhevsky, 2009]	$32 \times 32 \times 3$	50,000	10,000	[Wainakh *et al.*, 2021], [Wainakh *et al.*, 2022], [Liu and Lyu, 2022], [Zhang *et al.*, 2022], [Zou *et al.*, 2022], [Fu *et al.*, 2022a], [Kariyappa and Qureshi, 2023]
	Recommendation	Criteo[1]	$13 + 26$	4×10^7	6×10^6	[Li *et al.*, 2022], [Fu *et al.*, 2022a], [Sun *et al.*, 2022], [Kariyappa and Qureshi, 2023]
Regression	Prediction	Boston Housing[2]	13	506		[Xie *et al.*, 2023], [Qiu *et al.*, 2023]
		California Housing[3]	8	20,640		[Xie *et al.*, 2023], [Qiu *et al.*, 2023]

[1] Criteo, Criteo dataset, https://labs.criteo.com/2014/02/download-kaggle-display-advertising-challenge-dataset
[2] StatLib, Boston Housing dataset, http://lib.stat.cmu.edu/datasets/boston
[3] StatLib, California Housing dataset, https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html

Table 1: Summary of benchmark datasets used in LIA.

5.1 Benchmark Datasets

Similar to other attacks in the VFL scenario, LIA focuses on various tasks across multiple models, with a particular emphasis on classification and recognition tasks. In this section, we summarize the benchmark datasets commonly used for different models and tasks in Table 1.

Specifically, these benchmark datasets are typically utilized in the context of neural networks and regression models. Under the neural networks, the benchmark datasets are further categorized into two parts *classification and recognition* and *recommendation* based on the tasks, while the main task of the regression model is *prediction*.

We recommend that future LIA studies should be evaluated on these benchmark datasets to ensure fairness and facilitate meaningful comparisons.

5.2 Evaluation Metrics

To evaluate the performance of LIA, the majority of approaches [Wainakh *et al.*, 2021; Wainakh *et al.*, 2022; Liu and Lyu, 2022; Fu *et al.*, 2022a; Zheng *et al.*, 2022; Arazzi *et al.*, 2023; Kariyappa and Qureshi, 2023] use a classical metric *Attack Accuracy*, which denotes the percentage of all attack samples whose labels are inferred correctly. A higher attack accuracy typically indicates a more successful attack. Additionally, this metric often considers the accuracy of the first k inferences like Top-1 and Top-5 accuracy.

As part of LIA, the quality and the quantity of auxiliary data have a direct impact on the success of the attack [Wainakh *et al.*, 2021; Fu *et al.*, 2022a; Xie *et al.*, 2023]. Even a small amount of auxiliary data can significantly enhance the attack accuracy [Fu *et al.*, 2022a]. We recommend that the impact of auxiliary data on attacks should be evaluated in all LIA approaches that utilize auxiliary data. Furthermore, it is important to keep the auxiliary data consistent when comparing the different LIA approaches.

In addition, some classification-based LIA approaches also plot the receiver operating characteristic (ROC) curve and calculate the area under the curve (AUC) to evaluate the accuracy of the classifier [Li *et al.*, 2022; Sun *et al.*, 2022]. In some cluster-based approaches, the performance of the clusters formed by gradients or embeddings directly impacts the effectiveness of the attack. To visually evaluate it, these approaches often plot the clusters using t-SNE [Fu *et al.*, 2022a] or principal component analysis (PCA) [Liu and Lyu, 2022].

In LIA based on model reconstruction, the accuracy of the attack is often positively correlated with the accuracy of the model. Therefore, the model accuracy is also considered an important evaluation metric in such cases [Erdoğan *et al.*, 2022; Arazzi *et al.*, 2023; Kariyappa and Qureshi, 2023].

6 Conclusion

Focusing on the VFL-specific label leakage threat, this survey aims to provide a comprehensive summary of existing research from both attack and defense perspectives. It proposes two new taxonomies to categorize these attack and defense approaches, while also highlighting the importance of experimental benchmark datasets and evaluation metrics. Although many attack approaches have been proposed, there is a notable deficiency in studying the impact of auxiliary datasets on LIA and detecting malicious local optimizers. It is recommended that future research investigates these aspects to further enhance the understanding of LIA. As a rapidly evolving field, LIA holds great potential for future advancements. This survey serves to consolidate existing knowledge and provide a foundation for future work in this area. We hope that researchers in VFL will recognize the threats posed by LIA and take appropriate measures to mitigate them.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2021YFF1201102, the National Natural Science Foundation of China under Grants 62172016, 61932001 and 61972005, and the Tsinghua University (AIR)-Asiainfo Technologies (China) Inc. Joint Research Center.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*, pages 308–318, 2016.

[Abuadbba *et al.*, 2020] Sharif Abuadbba, Kyuyeon Kim, Minki Kim, Chandra Thapa, Seyit A Camtepe, Yansong Gao, Hyoungshick Kim, and Surya Nepal. Can we use split learning on 1d cnn models for privacy preserving training? In *AsiaCCS*, pages 305–318, 2020.

[Aggarwal *et al.*, 2021] Abhinav Aggarwal, Shiva Kasiviswanathan, Zekun Xu, Oluwaseyi Feyisetan, and Nathanael Teissier. Label inference attacks from log-loss scores. In *ICML*, pages 120–129, 2021.

[Almanifi *et al.*, 2023] Omair Rashed Abdulwareth Almanifi, Chee-Onn Chow, Mau-Luen Tham, Joon Huang Chuah, and Jeevan Kanesan. Communication and computation efficiency in federated learning: A survey. *Internet of Things*, 22:100742, 2023.

[Arazzi *et al.*, 2023] Marco Arazzi, Mauro Conti, Stefanos Koffas, Marina Krcek, Antonino Nocera, Stjepan Picek, and Jing Xu. BlindSage: Label inference attacks against node-level vertical federated graph neural networks. *arXiv preprint arXiv:2308.02465*, 2023.

[Bai *et al.*, 2023] Yijie Bai, Yanjiao Chen, Hanlei Zhang, Wenyuan Xu, Haiqin Weng, and Dou Goodman. VILLAIN: Backdoor attacks against vertical split learning. In *USENIX Security*, pages 2743–2760, 2023.

[Busa-Fekete *et al.*, 2021] Robert Istvan Busa-Fekete, Umar Syed, and Sergei Vassilvitskii. On the pitfalls of label differential privacy. In *NeurIPS Workshop*, 2021.

[Cheng *et al.*, 2021] 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.

[Cheu *et al.*, 2019] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *EUROCRYPT*, pages 375–403, 2019.

[Dwork, 2006] Cynthia Dwork. Differential privacy. In *ICALP*, 2006.

[Erdoğan *et al.*, 2022] Ege Erdoğan, Alptekin Küpçü, and A Ercüment Çiçek. UnSplit: Data-oblivious model inversion, model stealing, and label inference attacks against split learning. In *WPES*, pages 115–124, 2022.

[Erlingsson *et al.*, 2019] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *SODA*, pages 2468–2479, 2019.

[Fu *et al.*, 2022a] 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 *USENIX Security*, pages 1397–1414, 2022.

[Fu *et al.*, 2022b] Fangcheng Fu, Huanran Xue, Yong Cheng, Yangyu Tao, and Bin Cui. BlindFL: Vertical federated machine learning without peeking into your data. In *SIGMOD*, pages 1316–1330, 2022.

[Ghazi *et al.*, 2021] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. In *NeurIPS*, pages 27131–27145, 2021.

[Gu *et al.*, 2023] Hanlin Gu, Jiahuan Luo, Yan Kang, Lixin Fan, and Qiang Yang. Fedpass: Privacy-preserving vertical federated deep learning with adaptive obfuscation. In *IJCAI*, pages 3759–3767, 2023.

[Kairouz *et al.*, 2021] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawit, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[Kariyappa and Qureshi, 2023] Sanjay Kariyappa and Moinuddin K Qureshi. ExPLoit: Extracting private labels in split learning. In *SaTML*, pages 165–175, 2023.

[Knott *et al.*, 2021] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. CrypTen: Secure multi-party computation meets machine learning. In *NeurIPS*, volume 34, pages 4961–4973, 2021.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2022] 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 *ICLR*, 2022.

[Li *et al.*, 2023] Songze Li, Duanyi Yao, and Jin Liu. FedVS: Straggler-resilient and privacy-preserving vertical federated learning for split models. In *ICML*, pages 20296–20311, 2023.

[Liu and Lyu, 2022] Junlin Liu and Xinchen Lyu. Clustering label inference attack against practical split learning. *arXiv preprint arXiv:2203.05222*, 2022.

[Liu *et al.*, 2020] Yang Liu, Zhihao Yi, and Tianjian Chen. Backdoor attacks and defenses in feature-partitioned collaborative learning. *arXiv preprint arXiv:2007.03608*, 2020.

[Liu *et al.*, 2024] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances and challenges. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20, 2024.

[McMahan *et al.*, 2016] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016.

[Mo *et al.*, 2021] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. PPFL: privacy-preserving federated learning with trusted execution environments. In *MobiSys*, pages 94–108, 2021.

[Mohassel and Zhang, 2017] Payman Mohassel and Yupeng Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *S&P*, pages 19–38, 2017.

[Naseri *et al.*, 2023] Mohammad Naseri, Yufei Han, and Emiliano De Cristofaro. BadVFL: Backdoor attacks in vertical federated learning. *arXiv preprint arXiv:2304.08847*, 2023.

[Pasquini *et al.*, 2021] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. Unleashing the tiger: Inference attacks on split learning. In *CCS*, pages 2113–2129, 2021.

[Qiu *et al.*, 2023] Haoze Qiu, Fei Zheng, Chaochao Chen, and Xiaolin Zheng. Defending label inference attacks in split learning under regression setting. *arXiv preprint arXiv:2308.09448*, 2023.

[Rodríguez-Barroso *et al.*, 2023] Nuria Rodríguez-Barroso, Daniel Jiménez-López, M Victoria Luzón, Francisco Herrera, and Eugenio Martínez-Cámara. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90:148–173, 2023.

[Shokri and Shmatikov, 2015] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *CCS*, pages 1310–1321, 2015.

[Soltani *et al.*, 2023] Behnaz Soltani, Yipeng Zhou, Venus Haghighi, and John Lui. A survey of federated evaluation in federated learning. In *IJCAI*, pages 6769–6777, 2023.

[Sun *et al.*, 2021] Jiankai Sun, Yuanshun Yao, Weihao Gao, Junyuan Xie, and Chong Wang. Defending against reconstruction attack in vertical federated learning. *arXiv preprint arXiv:2107.09898*, 2021.

[Sun *et al.*, 2022] Jiankai Sun, Xin Yang, Yuanshun Yao, and Chong Wang. Label leakage and protection from forward embedding in vertical federated learning. *arXiv preprint arXiv:2203.01451*, 2022.

[Székely *et al.*, 2007] Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

[Takahashi *et al.*, 2023] Hideaki Takahashi, Jingjing Liu, and Yang Liu. Eliminating label leakage in tree-based vertical federated learning. *arXiv preprint arXiv:2307.10318*, 2023.

[Tan *et al.*, 2022] Juntao Tan, Lan Zhang, Yang Liu, Anran Li, and Ye Wu. Residue-based label protection mechanisms in vertical logistic regression. In *BigCom*, pages 356–364, 2022.

[Tran *et al.*, 2018] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, pages 8011–8021, 2018.

[Vepakomma *et al.*, 2018] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

[Vepakomma *et al.*, 2019] Praneeth Vepakomma, Otkrist Gupta, Abhimanyu Dubey, and Ramesh Raskar. Reducing leakage in distributed deep learning for sensitive health data. In *ICLR Workshop*, 2019.

[Wainakh *et al.*, 2021] Aidmar Wainakh, Till Müßig, Tim Grube, and Max Mühlhäuser. Label leakage from gradients in distributed machine learning. In *CCNC*, pages 1–4, 2021.

[Wainakh *et al.*, 2022] Aidmar Wainakh, Fabrizio Ventola, Till Müßig, Jens Keim, Carlos Garcia Cordero, Ephraim Zimmer, Tim Grube, Kristian Kersting, and Max Mühlhäuser. User-level label leakage from gradients in federated learning. In *PETS*, pages 227–244, 2022.

[Wen *et al.*, 2023] Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, 14(2):513–535, 2023.

[Wu *et al.*, 2023a] Ruihan Wu, Jin Peng Zhou, et al. Does label differential privacy prevent label inference attacks? In *AISTATS*, pages 4336–4347, 2023.

[Wu *et al.*, 2023b] Yuncheng Wu, Naili Xing, Gang Chen, Tien Tuan Anh Dinh, Zhaojing Luo, Beng Chin Ooi, Xiaokui Xiao, and Meihui Zhang. Falcon: A privacy-preserving and interpretable vertical federated learning system. *Proceedings of the VLDB Endowment*, 16(10):2471–2484, 2023.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Xie *et al.*, 2023] Shangyu Xie, Xin Yang, Yuanshun Yao, Tianyi Liu, Taiqing Wang, and Jiankai Sun. Label inference attack against split learning under regression setting. *arXiv preprint arXiv:2301.07284*, 2023.

[Xue *et al.*, 2024] Rui Xue, Kaiping Xue, Bin Zhu, Xinyi Luo, Tianwei Zhang, Qibin Sun, and Jun Lu. Differentially private federated learning with an adaptive noise mechanism. *IEEE Transactions on Information Forensics and Security*, 19:74–87, 2024.

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

[Zeng *et al.*, 2023] Dun Zeng, Siqi Liang, Xiangjing Hu, Hui Wang, and Zenglin Xu. Fedlab: A flexible federated learning framework. *Journal of Machine Learning Research*, 24(100):1–7, 2023.

[Zhang *et al.*, 2022] Xiaoxue Zhang, Xiuhua Zhou, and Kongyang Chen. Data leakage with label reconstruction in distributed learning environments. In *ML4CS*, pages 185–197, 2022.

[Zhao *et al.*, 2020] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. iDLG: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[Zheng *et al.*, 2022] Fei Zheng, Chaochao Chen, Binhui Yao, and Xiaolin Zheng. Making split learning resilient to label leakage by potential energy loss. *arXiv preprint arXiv:2210.09617*, 2022.

[Zhou *et al.*, 2022] Jun Zhou, Longfei Zheng, Chaochao Chen, Yan Wang, Xiaolin Zheng, Bingzhe Wu, Cen Chen, Li Wang, and Jianwei Yin. Toward scalable and privacy-preserving deep neural network via algorithmic-cryptographic co-design. *ACM Transactions on Intelligent Systems and Technology*, 13(4):1–21, 2022.

[Zhu *et al.*, 2019] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *NeurIPS*, pages 14774–14784, 2019.

[Zou *et al.*, 2022] Tianyuan Zou, Yang Liu, Yan Kang, Wenhan Liu, Yuanqin He, Zhihao Yi, and Qiang Yang. Defending batch-level label inference and replacement attacks in vertical federated learning. *IEEE Transactions on Big Data*, pages 1–12, 2022.

[Zou *et al.*, 2023] Tianyuan Zou, Yang Liu, and Ya-Qin Zhang. Mutual information regularization for vertical federated learning. *arXiv preprint arXiv:2301.01142*, 2023.