Active Data Reconstruction Attacks in Vertical Federated Learning

Minh N. Vu*, Tre' R. Jeter†, Raed Alharbi‡ and My T. Thai§
Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611, USA *†§
College of Computing and Informatics, Saudi Electronic University, Riyadh, 11673, Saudi Arabia ‡
Email: *minhvu@ufl.edu, †t.jeter@ufl.edu, ‡ri.alharbi@seu.edu.sa, §mythai@cise.ufl.edu

2023 IEEE International Conference on Big Data (BigData) | 979-8-3503-2445-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/BigData59044.2023.10386594

***Abstract*—Vertical Federated Learning (VFL) stands out as a promising approach to safeguard privacy in collaborative machine learning, allowing multiple entities to jointly train models on vertically partitioned datasets without revealing private information. While recent years have seen substantial research on privacy vulnerabilities and defense strategies for VFL, the focus has primarily been on passive scenarios where attackers adhere to the protocol. This perspective undermines the practical threats since the attackers can deviate from the protocol to improve their inference capabilities. To address this gap, our study introduces two innovative data reconstruction attacks designed to compromise data privacy in an active setting. Essentially, both attacks modify the gradients computed during the training phase of VFL to breach privacy. Our first attack uses an *Active Inversion Network* exploiting a small portion of known data in the training set to coerce the passive participants into training an auto-encoder for the reconstruction of their private data. The second attack, *Active Generative Network*, utilizes the knowledge of the training data distribution to guide the system into training a conditional generative network (C-GAN) for feature inferences. Our experiments confirm the efficacy of both attacks in inferring private features from real-world datasets.**

***Index Terms*—Vertical Federated Learning, Data Privacy, Data Reconstruction Attacks**

I. Introduction

Vertical Federated Learning (VFL) is a form of Federated Learning in which multiple entities collaborate to develop machine learning models, utilizing distinct features from the same data samples [1], [2]. The goal of VFL is to collaboratively develop learning models that maximize data utilization without exposing raw sensitive data. The demand for VFL has increased significantly in recent years, especially by organizations with limited and fragmented data seeking robust privacy machine learning solutions [3], [4].

As the VFL protocol involves many parties and not all can be fully trusted, preventing malicious participants from stealing sensitive data is crucial. While previous research has demonstrated that the data is better protected when the distributed participants use their own dataset and refrain from data sharing, the risks that adversarial participants could steal sensitive information still remain. In particular, the VFL has been shown to be vulnerable to label inference [5], [6], data reconstruction [7], [8], and property inference attacks [8] during both its training phase and inference phase. In response, some ad-hoc defenses are proposed [9], [10]. Nevertheless, both existing attack and defense strategies fail to fully capture the privacy risks associated with VFL, as they inadequately examine the practical capabilities of adversarial participants. Specifically, all preceding works exclusively focus on the *passive* setting, wherein adversarial participants strictly adhere to the system protocol (Table I). This consideration undermines the practical privacy risks inherent in VFL. The reason is VFL participants have the capability to deviate from the protocol, becoming *active* adversaries, without being detected [11]–[13].

This paper presents the first examination of active privacy attacks on VFL, i.e., active attacks executed by a malicious *active participant*. Note that the term active participant refers to the user in VFL who owns the labels, which is not the active/passive property of the privacy attacks. We consider the open problem: *To what degree can an active participant in VFL reconstruct the private training data of passive participants when it can deviate from the protocol?* In contrast to previously studied passive privacy breaches in VFL, our data reconstruction attacks tamper with the training gradients transmitted back to the passive participants, i.e., other participants without the training labels. The goal of tampering is to exploit the local models trained on those gradients to recover the victims' data. The main contributions of the paper are:

- We introduce an active data reconstruction attack called *Active Inversion Network* (AIN). The method maliciously modifies the training gradients with the aid of some public training data. The goal is to force other participants into training a decoder to reconstruct the private features during inference.
- The second attack involves an *Active Generative Network* (AGN), designed for the case when the adversary does not know any private features. AGN operates under the assumption that the adversary knows the distribution of the training data. Its objective is to coerce all participants into training a generative network that can recover the private features.
- We conduct experiments on MNIST [14] and USCensus [15] datasets to show that the proposed attacks can leverage the active setting and significantly improve the reconstructed data compared to passive attacks in VFL.

Organization. The paper is structured as follows: Section II provides the relevant background, introduces the notations employed in our paper, and reviews relevant literature. Section III outlines our threat models in the active setting along with the associated protocols. The details of our proposed AIN and AGN data reconstruction attacks are presented in Section IV. A comprehensive examination of our experiments is provided in Section V. Finally, Section VI concludes the paper.

II. Background, Notations and Related Works

This section provides the background of FL, our technical notations, and related works.

Federated Learning. The concept was first introduced by Google [16] for a cross-device scenario where millions of mobile devices collaboratively train a global machine learning model with the aid of a centralized server. At the first iteration, the central server randomly initializes the global model parameters. In each following training iteration, a subset of clients is chosen to join the training. Each selected participant will then receive the global parameters from the server and compute the model's gradients using their local dataset. These gradients are then aggregated to update the global model. The training continues iteratively until the global model converges.

Vertical Federated Learning. Depending on how data is partitioned among users, FL has two main forms: horizontal or vertical. In the horizontal setting, all users optimize on the same architecture and the same feature space while holding different data samples. On the other hand, this work studies the vertical setting in which participants possess disjoint sets of features while sharing the same set of samples [2]. The label set can be considered as a special feature and is owned by a special participant called the *active participant*. Other participants are referred to as *passive participants*.

The training process for VFL typically consists of two steps: Entity Alignment and Privacy-preserving training. While the Entity Alignment matches features from the same samples together for collaborative training, the Privacy-preserving training updates the global model via gradient descent [17]. The training procedure will be described in more detail later in Section III.

Data Reconstruction Attacks in VFL. The core of privacy protection in VFL is the users' private data. Various feature and label inference attacks have been developed to breach privacy in this context recently. These attacks typically consider the scenarios where the active participant aims to recover features of passive participants. The attacker mainly operates under two settings: *white-box*, where the attacker possesses knowledge of the passive participant's model architecture and parameters, and *black-box*, where such information is unavailable. Model inversion (MI) and gradient inversion (GI) are two primary white-box attacks. White-box MI methods [7], [8], [18], typically search for features that make the model's predicted outputs resemble the observed outputs. The GI attack, CAFE [9], seeks to identify features whose gradients align with the public gradients. For the black-box setting, the Binary Feature Inference attack (BFI) [10] is introduced to recover private binary features when the local models of passive participants consist of only one fully connected layer. Black-box MI attacks [7], [18] involve the adversarial training a shadow model $\hat{fi}$ to mimic the local model fi. Subsequently, the adversary replaces f_i by $\hat{f}_i$ and executes the privacy breach as in the white-box settings. In scenarios where the adversary is allowed to probe passive participants, previous research [7] showed that a direct inversion model g' can be used to recover input features $x_{n,k}$ from intermediate activations $z_{n,k}$. The highlight of these techniques are reported in Table I.

TABLE I: Characteristics of Data Reconstruction Attacks on Neural Networks in VFL. Aux. data, Bin. feat. and Data Dist. abbreviates for the needs of knowledge on auxiliary data, binary features and data distribution.

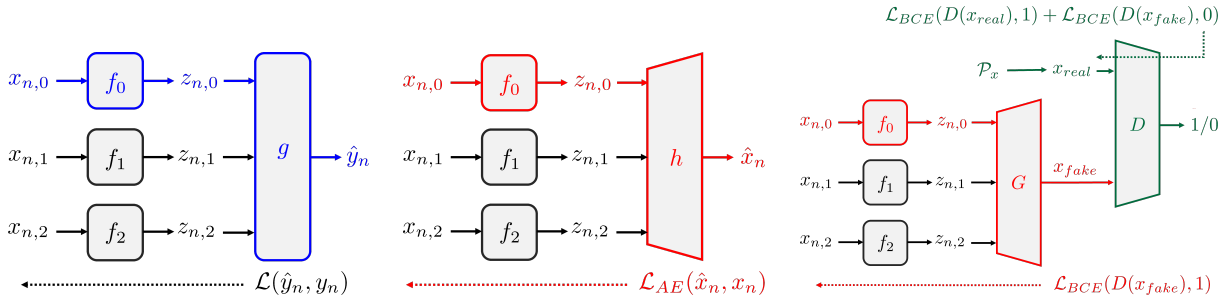
Attack	Setting	Type	Aux. Req.
CAFE [9]	White-box	Passive	–
White-Box MI [7], [8], [18]	White-box	Passive	–
BFI [10]	Black-box	Passive	Bin. feat.
Black-Box MI [7]	Black-box	Passive	Aux. data
AIN (Ours)	Black-box	Active	Aux. data
AGN (Ours)	Black-box	Active	Data Dist.

III. Active Threat Models

Previous research mainly focused on situations where the attacker breaches privacy while still following the learning protocol. It is commonly known as the honest-but-curious or semi-honest threat model. Nevertheless, that consideration fails to fully capture the potential vulnerability of VFL since the participants can deviate from the protocol to achieve stronger privacy attacks [11]–[13]. On the other hand, the focus of this work is on the training of neural networks in VFL. We investigate the concerns related to an active participant that may modify the training gradients. We later show that, by altering the gradients, the malicious active participant can exploit other participants into training models that can effectively infer the private features owned by those participants and, therefore, enhance the attack's capability compared to the passive cases. Furthermore, this work considers the black-box setting in which the attacker does not know the local models of passive participants. We choose to study the black-box settings to demonstrate better the gain that the active attackers have over those in the passive settings.

Notations. We study the VFL scenario in which $K+1$ users collaboratively train a neural network for the classification task. We denote the training dataset by $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$, with n representing the sample index. This dataset is partitioned and stored without sharing among $K+1$ participants: a sample $x_n \in \mathcal{D}$ is be divided into $[x_{n,0}, \cdots, x_{n,K+1}]$, where $x_{n,k}$ is the k-th partition of the n-th sample.

The model under consideration is a global neural network, as depicted in Fig. 1a. Since only the active participant possesses the label, it manages the aggregation of intermediate activations $z_{n,k}$ transmitted from passive participants $k, k \in \{1, \cdots, K+1\}$. We write the intermediate activations as $z_{n,k} = f_k(x_{n,k})$, where f_k denotes the encoder of the



(a) Normal VFL (b) Adversarial Decoder (c) Adversarial GAN

Figure 1: Illustrations of VFL in normal operation and under our proposed methods. The components controlled by the active participant are highlighted in colors. Solid arrows represent the forwarding direction, while dashed arrows indicate the back-propagation direction. In Adversarial GAN, green and red denote two distinct training phases of the attacker.

participant k. Additionally, by denoting the aggregation model as g, the final output of the global neural network in a forwarding computation is given by:

$$\hat{y}_n = g\left([f_0(x_{n,0}), \cdots, f_{K+1}(x_{n,K+1})]\right) \tag{1}$$

During normal VFL training iteration, the loss function on the data $\mathcal{D}$ is computed:

$$\mathcal{L}(\mathbf{\Theta}; \mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N} l\left(\mathbf{\Theta}; x_n, y_n\right) + \lambda \sum_{k=0}^{K} \gamma(\mathbf{\Theta}) \tag{2}$$

where $\mathbf{\Theta}, l, \gamma$ and λ are the trainable parameters, the loss function, the regularizer, and the controlling hyper-parameter for the strength of the regularization, respectively. The active participant then updates the weights and biases of its encoder f_0 and the aggregation model g, referred by $\mathbf{\Theta}_0$ and $\mathbf{\Theta}_g$, using the gradients $\nabla_{\mathbf{\Theta}_0}\mathcal{L}(\mathbf{\Theta}; \mathcal{D})$ and $\nabla_{\mathbf{\Theta}_g}\mathcal{L}(\mathbf{\Theta}; \mathcal{D})$, respectively. When the context is clear, we use $\nabla_{\mathbf{\Theta}_0}\mathcal{L}$ and $\nabla_{\mathbf{\Theta}_g}\mathcal{L}$ for brevity. After that, the active participant transmits $\nabla_{\mathbf{\Theta}_k}\mathcal{L}$ to the passive participant k to allow the local update of $\mathbf{\Theta}_k$.

Active attacks' settings. In an active attack, the active participant computes and transmits the tampered gradients to other participants to manipulate their models' update. For instance, our proposed methods exploit an auto-encoder reconstruction loss $\mathcal{L}_{AE}$ and a generative loss $\mathcal{L}_{GAN}$ instead of the normal loss to compute the gradients as illustrated in Fig. 1b and 1c. It is noteworthy to point out that active adversaries further have the capability to adjust the later layers of the global models to enhance privacy inference. Since the passive participants have no knowledge of the encoder f_0 and the later layers owned by the adversary, there is no trivial way for them to detect if the gradients have been tampered with.

We examine two attacking scenarios. The first is when the adversary knows some samples of the training data called the auxiliary data $\mathcal{D}_{\text{AUX}} \subset \mathcal{D}$. The assumption implies that the attacker knows $x_{n,k} \forall k$ in the auxiliary data, but only knows $x_{n,0}$ for x_n in $\mathcal{D} \setminus \mathcal{D}_{\text{AUX}}$. This scenario captures a situation in which the adversary engages in a collaborative VFL model training with local data owners who exclusively possess data for some specific features. However, the active participant may know those features for some samples used in the VFL training. A practical example is when a central hospital (the active participant) collaborates with a specialized healthcare center (the passive participant). The model's inputs consist of patient records, including features that are only known by the specialized healthcare center. The attack applies when the central hospital knows those features for some records used for the training.

The second setting investigate the scenarios that the adversary only knows the training data distribution, i.e., $|\mathcal{D}_{\text{AUX}}| = 0$. This threat model describes the situations when the active participant owns a massive amount of data to cover the features distribution [19] or when the data can easily be sampled by the adversary. For example, it can depict the example mentioned in the previous setting when the central hospital has a big database of records with all features. As long as the training data comes from the same distribution, the methods introduced in this setting can be executed by that central hospital.

IV. Active Data Reconstruction Attacks

We now discuss our black-box data reconstruction attacks. Our methods leverage an under-exploited capability of the active participants in VFL, which is its control over the gradients transmitted toward the passive participants. As outlined in the threat models, each of our attacks operates with minimal assumptions about the private training data. A fundamental distinction in our active adversaries, as opposed to prior passive approaches, lies in the engagement of the passive local models. Unlike passive methods that take the local models as fixed, our approach involves the exploitation on their gradients to breach privacy.

A. Active Inversion Network

Our first attack consists of an *Active Inversion Network* (AIN) that can recover the private features from the intermediate activations. The model is denoted by the h function in Fig. 1b. By tampering the returned gradients, the active attacker forces the passive participants into the joint training of the AIN. Particularly, the adversary uses a set of auxiliary data that is known prior to the training and a reconstruction loss, e.g., Mean-Squared-Error (MSE) loss, to calculate the

tampered gradients and transmit them to the passive users. When the passive users update their local parameters using that gradients, their local models are actually trained for the adversarial reconstruction task. Consequently, the intermediate signals $z_{n,k}$ resulting from those models after the adversarial training can be used to reconstruct the private input features.

Training of AIN. Upon receiving the intermediate activations z from passive participants during the training, the AIN adversary executes three activities: filtering the data for computing of the adversarial loss, updating its weights and biases, and transmitting the tampered gradients to the passive participants. These activities enforce the training loss of the inverse network h to be the reconstruction MSE loss:

$$\mathcal{L}_{AE}(\hat{X}, X) = \frac{1}{|X|}\sum_{n=1}^{|X|} \|\hat{x}_n - x_n\|^2 \tag{3}$$

where X and $\hat{X}$ are the set of inputs that are in both the auxiliary data and the training batch, and the output of h on the intermediate activations, respectively. As such, the AIN training can be described by the following optimizations:

$$\text{Active user: } \arg\min_{\boldsymbol{\Theta_0}, \boldsymbol{h}} \mathcal{L}_{AE}(\hat{X}, X) \tag{4}$$

$$\text{Passive user } k\text{: } \arg\min_{\boldsymbol{\Theta_k}} \mathcal{L}_{AE}(\hat{X}, X) \tag{5}$$

Data reconstruction. When the training of the AIN h converges, the active participant can reconstruct any private input data from its intermediate activations: $x_n \approx h(z_n)$.

B. Active Generative Network

Our second approach, consisting of an *Active Generative Network* (AGN), is designed for cases when the adversary does not have access to the auxiliary data. AGN is based on the idea of Generative Adversarial Networks (GAN) [20] which consists of a *generator* and a *discriminator*. The generator is responsible for generating new data samples that mimic those in the distribution while the discriminator tries to differentiate between the *real* data and the *fake* samples, i.e., the samples generated by the generator. As the output of the GAN's generator is typically originated from random noise, it is not designed for reconstruction/inference tasks. To tackle this issue, AGN exploits the information that the active participant knows about the target sample, i.e., $x_{n,0}$ and potentially y_n to encourage the generator to return x_n instead of an arbitrary data point from the true distribution. From this aspect, AGN shares some similarities with the Conditional GAN [21], which is a modified GAN to condition the data generation on side-channel information.

Training of AGN. During one VFL training iteration, the AGN adversary conducts 4 main activities. The first activity generates the data batch to train the discriminator. As the true distribution is known, the real data labeled as 1 is drawn from that distribution while the fake data labeled as 0 is created using the intermediate activations. Note that, to incorporate the information that the adversary has on the sample, $x_{n,0}$ is additionally feed to the GAN generator. The second activity involves updating the discriminator, following the conventional GAN training protocol. Specifically, the discriminator undergoes an update using the Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}_{BCE}(Y, 1) = \frac{1}{|X|}\sum_{n} \log(y_n) \tag{6}$$

$$\mathcal{L}_{BCE}(Y, 0) = \frac{1}{|X|}\sum_{n} \log(1 - y_n) \tag{7}$$

The third activity involves updating the generator. Unlike the discriminator, which is trained to identify the generated data using the loss $\mathcal{L}_{BCE}(D(X_{fake}), 0)$, the generator is encouraged to produce data from the true distribution using the loss $\mathcal{L}_{BCE}(D(X_{fake}), 1)$.

Similar to AIN, the final activity of AGN is the transmission of tampered gradients $\nabla_{z_{n,k}}\mathcal{L}_{BCE}(D(X_{fake}), 1)$ to the passive participants. The activity coerces other users into the generator's training.

Data reconstruction. During inference, the adversary uses the generator and calculates the reconstructed signal as:

$$\hat{x}_n = G([z_{n,0}, \cdots, z_{n,K+1}]), \quad \hat{x}_{n,0} = x_{n,0}$$

where $z_{n,0}, \cdots, z_{n,K+1}$ are the intermediate activation received by the active participant. Since the generator is designed to generate an $\hat{x}_n$ that is similar to the training data and its partition $\hat{x}_{n,0}$ matches that of the original sample x_n, the generator is able to recover $\hat{x} \approx x$ since it also helps defeat the discriminator during GAN training. The intuition is that, the more $x_{n,0}$ are known, the higher the likelihood that $\hat{x}_n$ will resemble x_n. We provide an experiment to demonstrate this intuition in Section V (see Table II).

The advantage of AGN lies in its independence from any private features used during training. Given the rapid increase of real-world public data, the assumption that the active adversary has access to or can easily sample data from the training distribution is becoming more plausible. Consequently, the AGN attack effectively emphasizes real-world privacy threats.

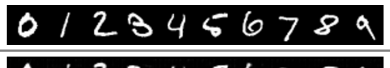
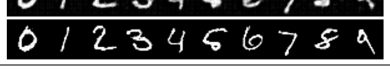
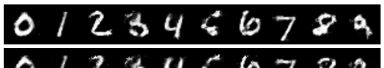
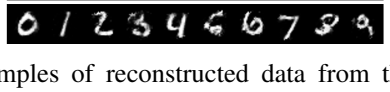
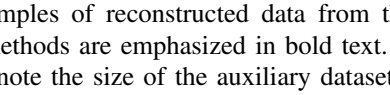
Inputs		PSNR
AIN(2K)		42.62dB
AIN(1K)		40.08dB
AGN		33.12dB
Benchmark(2K)		35.14dB
Benchmark(1K)		32.92dB

Figure 2: Examples of reconstructed data from the MNIST dataset. Our methods are emphasized in bold text. The labels 1K and 2K denote the size of the auxiliary dataset.

V. Experiments

This section provides experiments demonstrating the effectiveness of our proposed methods in reconstructing private real-world data and illustrates the potential privacy threat that an active attack can achieve by deviating from the VFL protocol.

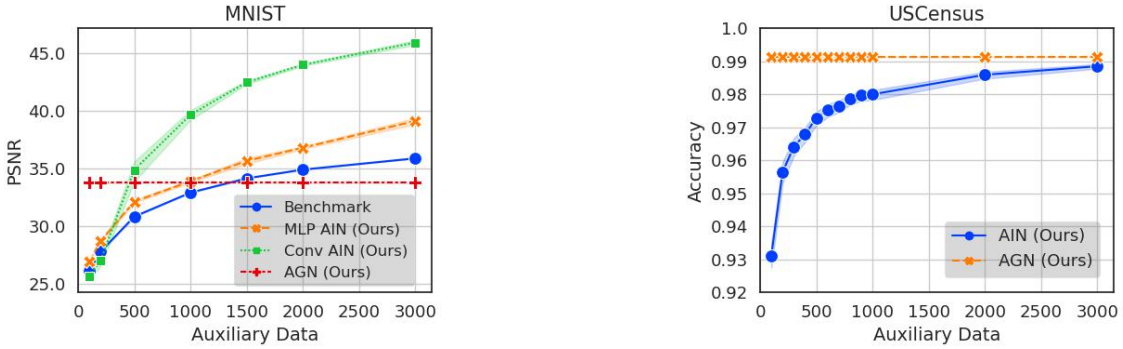


Figure 3: PSNR(dB) for the reconstructed data in the image dataset and the accuracy of the reconstructed features in the tabular dataset. As AGN does not necessitate auxiliary data $\mathcal{D}_{\text{AUX}}$, its results are reported in a straight line for easy comparison.

TABLE II: The accuracy and PSNR of the AGN attack on the USCensus dataset with varying numbers of known features.

No. known features	50/68	46/68	38/68	30/68	22/68	14/68
Accuracy	99.73 ± 0.10	99.36 ± 0.06	99.28 ± 0.24	98.69 ± 0.40	97.65 ± 0.55	95.90 ± 0.34
PSNR(dB)	52.55 ± 3.05	44.18 ± 0.70	43.52 ± 2.51	38.20 ± 2.31	33.00 ± 1.93	27.98 ± 0.65

A. Experimental settings

Our evaluations are conducted on MNIST [14] and USCensus [15] datasets. The MNIST dataset is widely used in vision tasks, while USCensus consists of a one percent sample with 68 features from Public Use Microdata Samples records.

Simulations of the passive participants. In practical scenarios, the model's architectures for the passive users are selected according to the specific applications. In our experiments, we opt for common neural network architectures tailored to the respective datasets. Specifically, we employ Multi-layer Perceptron/Fully-connected layers (MLP) for the tabular dataset and Convolutional layers (Conv.) for the image dataset. The hyper-parameters of the examined models are set based on the typical tasks associated with the data: image classifications for MNIST [22], and data clustering for USCensus.

Configurations of the attackers. As the components of the active participant are controlled by the attacker, they have the flexibility to choose architectures that enhance the reconstruction of signals. For our AIN attack on the image dataset, we explore two architectures: MLP AIN and Conv. AIN, where the decoders employ Multi-layer Perceptron (MLP) and Convolutional layers, respectively. For all other results, MLPs are employed.

Benchmark. While no existing work specifically addresses active data reconstruction attacks in VFL, black-box inference attacks are also limited (Table I). Consequently, we benchmark our method against the black-box MI method [7], referred by *benchmark* in our reports. Our methods differ from the benchmark in two key aspects. Firstly, our methods operate during the training phase of VFL, while the benchmark is conducted during the inference phase. Secondly, for the same reason, the benchmark does not involve the choice of the neural network's architecture. The architecture used for the benchmark follows the configuration outlined in the original paper, which is MLP.

B. Experimental Results

General performance. The results of data reconstruction attacks on MNIST are shown in Fig. 2. The first row displays the target of inference taken from the test set, while the subsequent rows showcase the reconstructed samples obtained by different attacks. In our two-participant system, consisting of one active and one passive participant, each participant possesses half of the image samples. Specifically, the active attacker in the experiments has the bottom half of the images and its objective is to reconstruct the top hafts. Consequently, the computed Peak Signal-to-Noise Ratios (PSNRs) are reported solely for the top half. Nevertheless, the figures presented depict the entire reconstructed images for a more comprehensive and intuitive visualization.

The results of our attacks are referred to in bold text in the figure. The results demonstrate that our adversaries can reconstruct the input samples with high quality. Notably, with an equivalent quantity of auxiliary data, our AIN attacks surpass the benchmark by a significant margin. These examples clearly illustrate the advantage of active adversaries over their passive counterparts.

Impact of the size of auxiliary data. Fig. 3 reports more comprehensive assessments of the methods when the sizes of the auxiliary data vary. It can be observed from the experimental results on MNIST that, by optimizing the architecture of the later layers in VFL training, the attacker can substantially enhance the attack.

In the case of tabular data from USCensus, the objective of the attacks is to reconstruct 34 out of 68 private features in the training samples. AGN consistently achieves approximately 99% accuracy in inference, whereas the AIN approach requires approximately 3000 samples to achieve a similar PSNR. As the passive counterparts for tabular data are not yet available, the passive benchmark is not included. Nevertheless, the results demonstrate that our active attacks exhibit competitive

performance, not just in the context of image data but also in the case of tabular data.

Impact of the number of known features in AGN. Intuitively, the more features $x_{n,0}$ that the adversary possesses, the more straightforward for it to recover the remaining private features. The intuition is illustrated in Table II presenting the accuracy and PSNRs of our generative approach AGN in the USCensus dataset. Notably, even when there are only 14 features in $x_{n,0}$, our AGN attack can reconstruct the other 54 private features with an accuracy of nearly 96%. This result suggests that merely limiting the number of features accessed by active adversaries may not serve as an effective defense against this kind of attack.

A significant observation is that even with only 14 known features, our AGN attack can infer the remaining 54 features with nearly 96% accuracy. This implies that simply restricting the number of features known to the attacker might not be an effective defense against active attacks.

VI. Conclusion

This paper examines the feasibility of inferring sensitive private data used for the training of VFL. We have discovered that the VFL active user can diverge from the protocol and successfully recover the local data of passive users by tampering with the training gradients. Specifically, we introduce two data reconstruction attacks exploiting the information about either a limited training dataset or the overall data distribution, showcasing their strong inference capability in practical situations. Our research emphasizes the necessity of preserving data privacy throughout the VFL training process.

Acknowledgement

This material is based upon work supported by the National Science Foundation under grants CNS-1935923 and CNS-2140477.

References

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, jan 2019.

[2] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, *Vertical Federated Learning*. Cham: Springer International Publishing, 2020, pp. 69–81.

[3] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui, "Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, may 2022.

[4] D. Cha, M. Sung, and Y.-R. Park, "Implementing vertical federated learning using autoencoders: Practical application, generalizability, and utility study," *JMIR Medical Informatics*, vol. 9, p. e26598, 06 2021.

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

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

[7] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Proceedings of the 35th Annual Computer Security Applications Conference*, ser. ACSAC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 148–162.

[8] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi, "Feature inference attack on model predictions in vertical federated learning," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2021.

[9] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, "Cafe: Catastrophic data leakage in vertical federated learning," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 994–1006.

[10] P. Ye, Z. Jiang, W. Wang, B. Li, and B. Li, "Feature reconstruction attacks and countermeasures of dnn training in vertical federated learning," *arXiv preprint arXiv:2210.06771*, 2022.

[11] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot, "When the curious abandon honesty: Federated learning is not private," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2023, pp. 175–199.

[12] T. Nguyen, P. Lai, K. Tran, N. Phan, and M. T. Thai, "Active membership inference attack under local differential privacy in federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 5714–5730.

[13] L. H. Fowl, J. Geiping, W. Czaja, M. Goldblum, and T. Goldstein, "Robbing the fed: Directly obtaining private data in federated learning with modified models," in *International Conference on Learning Representations*, 2021.

[14] Y. LeCun and C. Cortes, "MNIST handwritten digit database," http://yann.lecun.com/exdb/mnist/, 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[15] M. Meek, T. Thiesson, and H. Heckerman, "US Census Data (1990)," UCI Machine Learning Repository. [Online]. Available: https://doi.org/10.24432/C5VP42

[16] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[17] L. Wan, W. K. Ng, S. Han, and V. C. S. Lee, "Privacy-preservation for gradient descent methods," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '07. NY, USA: Association for Computing Machinery, 2007.

[18] X. Jiang, X. Zhou, and J. Grossklags, "Comprehensive analysis of privacy leakage in vertical federated learning during prediction." *Proc. Priv. Enhancing Technol.*, vol. 2022, no. 2, pp. 263–281, 2022.

[19] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[21] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8026–8037.