Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.055596

ARTICLE

SensFL: Privacy-Preserving Vertical Federated Learning with Sensitive Regularization

Chongzhen Zhang[1,2,*], Zhichen Liu[3], Xiangrui Xu[3], Fuqiang Hu[3], Jiao Dai[3], Baigen Cai[1] and Wei Wang[3]

[1]School of Automation and Intelligence, Beijing Jiaotong University, Beijing, 100044, China

[2]Shuohuang Railway Development Co., Ltd., National Energy Group, Cangzhou, 062350, China

[3]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, School of Computer Science and Technology, Beijing Jiaotong University, Beijing, 100044, China

*Corresponding Author: Chongzhen Zhang. Email: 21111092@bjtu.edu.cn

Received: 02 July 2024 Accepted: 10 October 2024 Published: 17 December 2024

ABSTRACT

In the realm of Intelligent Railway Transportation Systems, effective multi-party collaboration is crucial due to concerns over privacy and data silos. Vertical Federated Learning (VFL) has emerged as a promising approach to facilitate such collaboration, allowing diverse entities to collectively enhance machine learning models without the need to share sensitive training data. However, existing works have highlighted VFL's susceptibility to privacy inference attacks, where an honest but curious server could potentially reconstruct a client's raw data from embeddings uploaded by the client. This vulnerability poses a significant threat to VFL-based intelligent railway transportation systems. In this paper, we introduce *SensFL*, a novel privacy-enhancing method to against privacy inference attacks in VFL. Specifically, SensFL integrates regularization of the sensitivity of embeddings to the original data into the model training process, effectively limiting the information contained in shared embeddings. By reducing the sensitivity of embeddings to the original data, SensFL can effectively resist reverse privacy attacks and prevent the reconstruction of the original data from the embeddings. Extensive experiments were conducted on four distinct datasets and three different models to demonstrate the efficacy of SensFL. Experiment results show that SensFL can effectively mitigate privacy inference attacks while maintaining the accuracy of the primary learning task. These results underscore SensFL's potential to advance privacy protection technologies within VFL-based intelligent railway systems, addressing critical security concerns in collaborative learning environments.

KEYWORDS

Vertical federated learning; privacy; defenses

1 Introduction

In this digital age, various industries and people are extensively utilizing big data and artificial intelligence to improve their operations [1–4]. Although abundant data offers significant opportunities for AI applications [5–7], most of this data is inherently highly sensitive and exists in isolation. Traditional methods fail to effectively address the issues of training models across different locations and privacy concerns, which means organizations might have to risk data leakage to train models

[8,9]. Federated learning (FL) has emerged to tackle these issues. FL is a machine learning paradigm that collaboratively trains machine learning models involving multiple data repositories in a privacy-preserving manner, and this technology has been applied in multiple fields [10–14]. In Verdict Federated Learning (VFL), participants have overlapping data samples but differ in the feature space, which is quite similar to the data distribution in the railway industry. For instance, freight trains change their cargo at different stations. In addition, FL enables railway companies to grasp real-time information on trains, facilitating adjustments to train operations and cargo transportation plans [15].

The use of FL in the rail transit field is pervasive [16,17]. For example, FL allows railway companies to grasp real-time information about trains, which is convenient for adjusting train operations and cargo transportation plans. Furthermore, FL helps in real-time monitoring of track health, timely detection of issues such as cracks and deformations, and prediction of potential future failures, ensuring the safety and stability of railway lines. Sometimes, some information also fits the data distribution of VFL quite well, such as the replacement of cargo by freight trains at different stations. The schematic diagram of the scene is shown in Fig. 1.

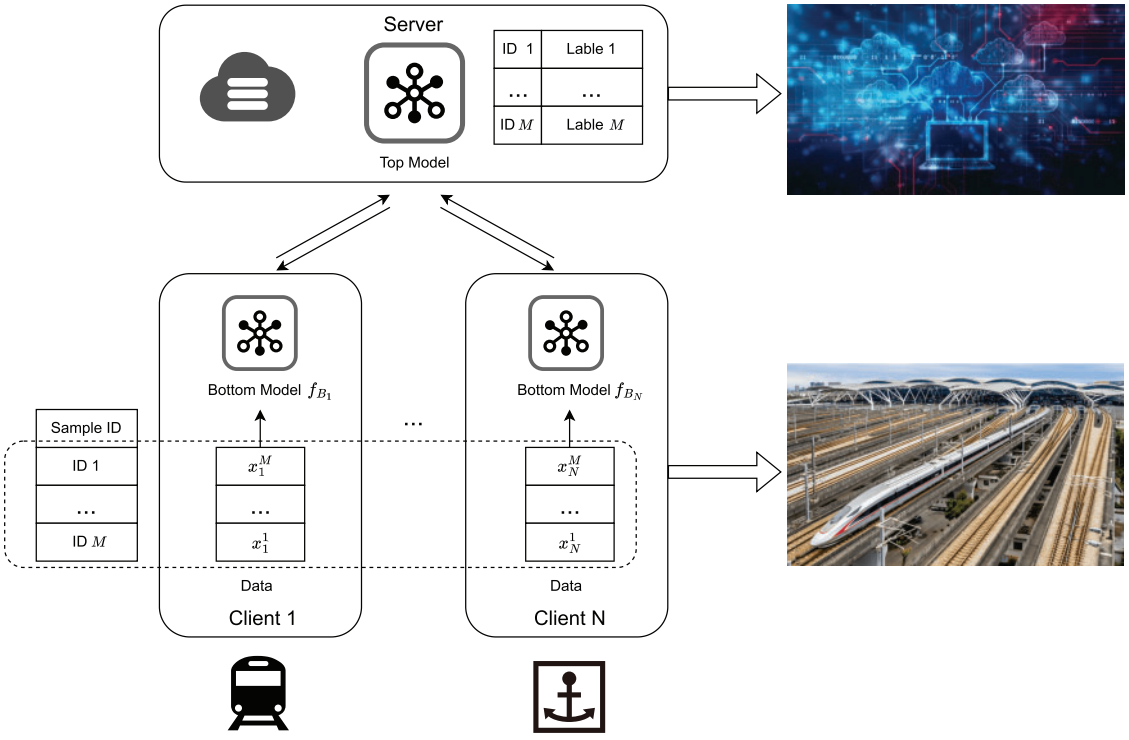

Figure 1: The structure of VFL in intelligent railway transportation systems

Although VFL has made gratifying progress in utilizing scattered data from different participants, its distributed nature makes it vulnerable to malicious attacks [18–22]. For example, some honest but curious servers might launch data reconstruction attacks on clients using the feature embeddings uploaded by local clients to grasp clients' private data, which is undoubtedly a massive challenge for VFL. Differential privacy is a standard defense method against privacy leakage, which mainly reduces the relationship between the embedding vectors and the original data by adding noise to alleviate privacy leakage. However, the added noise will inevitably reduce the accuracy of the model's primary

task. Therefore, defending against privacy leakage attacks without affecting the utility of VFL is still a considerable challenge.

In this work, we propose a new privacy-enhancing method. Its main task is to maintain the accuracy of the model's main task while resisting privacy theft attacks. Overall, we enhance the model's ability to defend against privacy theft attacks by limiting the maximum mutual information between the embedding vectors and the input samples. This method not only solves some common privacy leakage issues in the railway industry but also ensures the accuracy of the main task.

We evaluated the effectiveness of our method on four datasets and three models, considering the impact of different embedding dimensions. The experiments prove that our method effectively resists privacy inference attacks without sacrificing the accuracy of the main task.

Our research has made the following contributions:

- We have investigated the relationship between sensitivity and data privacy in the VFL and found that the sensitivity between the embedding vectors and the client's input information can directly affect the effectiveness of privacy attacks.
- Based on the above findings, we designed a defense method based on sensitivity regularization. This method resists privacy inference attacks by limiting the correlation between the embedding vectors and the sensitive input information. At the same time, this method does not affect the accuracy of the model's primary task.
- We conducted a comprehensive experiment and evaluation of the proposed method. The experimental results show that it can effectively resist privacy inference attacks without affecting the main task. These findings indicate that this method is essential in promoting the robust development of privacy protection technology in the railway transportation system.

2 Background

2.1 Federated Learning

FL is a distributed machine learning paradigm. It allows multiple clients, such as mobile devices, servers, and databases, to co-train a model and ensure data is only stored locally. This method has significant advantages in data privacy and security because it does not require centralized storage of data in the cloud or on a single server. Google proposed the FedAvg [23] algorithm in 2016, which is considered the beginning of FL.

FL can be divided into three categories [24]: Horizontal Federated Learning (HFL), VFL, and Federated Transfer Learning (FTL). The difference between them lies in how data is divided in the sample and feature spaces.

- HFL: HFL refers to the situation where participants share the same feature space but have different data samples. It is the most commonly used FL model. For example, Google has applied HFL to train language models on mobile devices [23].
- VFL: In VFL, participants have overlapping data samples, but their feature space is different. For example, Webank [25] uses the VFL to build financial risk models for their enterprise customers.
- FTL: FTL refers to an FL model where the datasets have differences in both the feature space and the sample space, with limited overlap. For example, one study [26] suggested that it can run Anomaly Detection (AD) models on edge devices to ensure the safety of agricultural equipment.

We now consider an easy FL setting: N clients exist, and each client i has its local dataset D_i. w is the model parameter. In each round of training, the client i uses the gradient descent to update local model parameters on the local dataset.

$$w_i^{(t+1)} = w^{(t)} - \eta \nabla \mathscr{L}_i(w^{(t)}) \tag{1}$$

where the η is learning rate and $\nabla \mathscr{L}_i(w^{(t)})$ is the gradient of local loss function. Next, the central server aggregates the model parameters from all clients.

$$w^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} w_i^{(t+1)} \tag{2}$$

The execution of FL can be divided into three stages [27], including Data and Behavior Auditing, Training, and Predicting. The model faces different security and privacy threats at each phase of FL execution. Because of the distributed characteristics of FL, it is susceptible to various security threats, such as poisoning attacks, backdoor attacks, privacy inference attacks, etc. For instance, distributed backdoor attacks [28] exploit the distributed nature of FL by breaking down the backdoor trigger into multiple local triggers and embedding them into the training data of different malicious clients. To enhance the effectiveness of backdoor attacks, Bagdasaryan and others proposed a model replacement attack [29] that replaces the original global model with a malicious local model designed by the attacker.

2.2 Privacy Inference Attacks in VFL

In VFL, features are private because they contain sensitive information. Therefore, people have already proposed various methods used to obtain the features of the data. The attackers can attack the model at any state, including the training phase and inference phase.

Training phase attacks: In the training phase, the attacker may have more information about the model and data. Ye et al. [30] proposed a Binary Feature Inference attack (BFI). It can reconstruct the sensitive binary features from the passive user's local output, and this research has proven that under the assumption of binary features, this attack is a Non-deterministic Polynomial-time hard (NP-hard) problem. Weng et al. [31] proposed two privacy attacks: reverse multiplication attack for the logistic regression VFL protocol and reverse sum attack for the XGBoost VFL protocol. The attacker uses encrypted intermediate multiplication results to infer the passive party's original training data in a reverse multiplication attack and uses unique. Magic Number to reveal partial sequences of the passive party's features. Jin et al. [32] extended gradient inversion to the white-box VFL, and they proposed Catastrophic Data Leakage in Vertical Federated Learning (CAFE), which exploits the shared aggregated gradients in VFL to recover batch data efficiently. Compared to traditional data leakage attacks, CAFE demonstrates higher efficiency and recovery quality when handling large-scale data recovery.

Inference phase attacks: Luo et al. [33] proposed three feature inference attacks: Equality Solving Attack (ESA), Path Restriction Attack (PRA), and Generative Regression Network (GRN) to attack logistic regression, decision tree, and neural network. When the number of the passive party's features is small, the feature values of the passive party can be accurately inferred by using ESA. In PRA, the attacker can restrict the path in the decision tree to infer the passive party's features. GRN infers the private features by analyzing the model's prediction outputs. He et al. [34] proposed a black-box model inversion attack, it can learn the passive party's features by training a shadow model to mimic the local model using auxiliary data in SplintNN.

2.3 Defense against Privacy Inference Attacks in VFL

In response to these attacks, researchers have proposed a variety of defense mechanisms aimed at protecting privacy and security in VFL. Mostly, the defense strategies can be classified based on whether cryptographic encryption is used.

Cryptographic defense: These strategies employ secure computation methods to evaluate the functions of multiple participants while exposing only the necessary information to the intended parties to prevent potential attackers from inferring private data [35–37]. Hardy et al. [38] proposed HardyLR, which is a privacy-preserving FL method on vertically partitioned data based on entity resolution and homomorphic encryption (HE). Secure Logistic Regression for Vertical Federated Learning (SecureLR) [39] is HardyLR's follow-up work, it removes the coordinator from the training and inference procedure by relaxing either the efficiency or privacy constraint. SecureBoost [40] trains a high-quality tree-boosted model (XGBoost) for each party and exploits additive HE to maintain the confidentiality of the training data among multiple parties. Chamani et al. [41] found the weakness in SecureBoost and proposed a leakage-abuse attack based on its leakage profile. This research also proposed two countermeasures based on a Trusted Execution Environment (TEE) to mitigate feature leakage.

Non-cryptographic defense: In essence, non-cryptographic defense methods reduce the correlation between private data and leaked data, for example, adding noise, differential privacy [42] and knowledge distillation [43]. A hybrid differentially private VFL methods [44] was proposed to ensure the data confidentiality of VFL participants. This strategy adds Gaussian noise to all parties' intermediate results. Dryden et al. [45] proposed Gradient Discretization (GD), which explores quantizing gradient updates before communication, encoding originally continuous gradients into discrete ones to reduce the leakage of private information. Gradient Sparsification (GS) [46] indicates that most of the gradient updates are close to zero, so setting the original gradients with smaller absolute values to zero and exchanging them with a sparse matrix can alleviate privacy leakage without affecting the convergence of the original VFL task. Beyond the common defense methods that detect suspicious local gradients based on plaintext, some approaches have also been proposed that can resist poisoning attacks without sacrificing accuracy [47]. In the field of Intelligent Transportation and Next Generation Internet-of-Things (NG-IoT), individual equipment has become computing platforms, which also brings the risk of privacy leakage. Yazdinejad's research [48] also proposes a hybrid privacy-preserving federated model based on a combination of synchronous and asynchronous methods, which improves issues related to user dropout and low-quality data.

3 Problem Setup

3.1 Vertical Federated Learning

The core idea of VFL is the vertical partitioning of data, where each participant possesses different dimensions or features of the data but shares the same data subjects [33]. We now consider a classical VFL setting: there are N clients and a central server S collaboratively trains a VFL model with parameters θ on a dataset $D = \{(\mathbf{x}^j, y^j)\}_{j=1}^{M}$ with M samples, $\mathbf{x} \in \mathscr{R}^P$. Each training data $\mathbf{x}$ is split by N unique and distinct subset for all clients, i.e., $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$. Each client trains its bottom model f_{B_i} to extract high-level feature embeddings for local data, i.e., $\mathbf{e}_i = f_{B_i}(\mathbf{x}_i)$, $e \in \mathscr{R}^K$. Then, the generated feature embeddings $\mathbf{e}_i$ will be sent to the server. The server concatenate the embeddings $\mathbf{e}_{\text{cat}} = \text{Concat}[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_N]$ and ground-truth labels y to train a top model F_T. Therefore, the training

of VFL is formulated as follows:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\in D}[\ell_{main}(\mathbf{x}, y; \theta)], \tag{3}$$

where the θ is the collection of all bottom model and top model parameters, i.e., $\theta = \{\theta_{B_1}, \theta_{B_2}, \cdots, \theta_{B_N}, \theta_T\}$. The ℓ_{main} is the loss function and it can be represented as:

$$\ell_{main}(F_T(\text{Concat}(f_{B_1}(\mathbf{x}_1), f_{B_2}(\mathbf{x}_2), \ldots, f_{B_N}(\mathbf{x}_N)), y; \theta) \tag{4}$$

The server needs to update the top model by optimizing the objective function Eq. (3). After updating, it will dispatch the gradients of this function relative to the feature embeddings of each local client to themselves. Every client receives these gradients and utilizes them to perform backpropagation, updating their local bottom models.

3.2 Privacy Inference Attack

The goal of the privacy attack [32] is to divulge some private training pieces of information about the clients and their bottom models. In this paper, we attempt to reconstruct the data within the image dataset and infer the features of the tabular data. Because the structures and attributes of these two types of data are different, we need to use different processing methods to perform attacks. When the dataset is images, the features can be some compositional information, such as pixel values and channel count, etc. We consider using the Data Reconstruction (DR) attack, and the attackers need to obtain various features of the images to reconstruct the content of the original images. However, since the server does not come into direct contact with the bottom model, it cannot obtain the corresponding model knowledge. Therefore, before conducting DR, Model Stealing (MS) should be used first to learn a surrogate model that can be used to replace the original true bottom model. In attacks targeting tables, the main purpose is to capture the correlations between the feature embeddings and corresponding target property values. Therefore, attackers need to use auxiliary data to mock the corresponding and infer real data's features [33].

3.2.1 Attacks to Image Data

In this attack method, the attacker first needs to clone the client's bottom model to obtain the corresponding feature embeddings. More specifically, we use $\mathbf{x}^{aux}$ to denote the auxiliary data created by the attacker, the target bottom model f_{B_t} will generate the embeddings of $\mathbf{x}^{aux}$ and denote it as $f_{B_t}(\mathbf{x}^{aux})$. The surrogate model $\hat{f}_{B_t}$ with its parameters $\hat{\omega}_{B_t}$ can be trained by minimizing the loss function ℓ_{MS} between it and bottom model. The process can be represented as:

$$\arg\min_{\hat{\omega}_B} \ell_{MS}(\hat{f}_{B_t}(\mathbf{x}^{aux}; \hat{\omega}_B), f_{B_t}(\mathbf{x}^{aux})) \tag{5}$$

By optimizing Eq. (5), if the attacker has the same input as clients, the surrogate model $\hat{f}_{B_t}$ can produce approximately the same embeddings as the true bottom model f_{B_t}.

After learning the surrogate model, the goal of the attack is to find an estimate of the features of the target training data. The attacker first obtains the feature embedding of the surrogate model and then optimizes the embedding results to approach the embedding results of the true bottom model. Afterward, through reverse optimization, an attempt is made to recover the real data features. Given a random piece of noise $\hat{\mathbf{x}}_t$, surrogate model $\hat{f}_{B_t}$ can output its specific feature embedding results $\hat{f}_{B_t}(\hat{\mathbf{x}}_t)$. Similarly, the true bottom model f_{B_t} will also output its corresponding feature embedding $f_{B_t}(\mathbf{x}_t)$ when

the input is true data $\mathbf{x}_t$. This task can be represented as follows:

$$\arg\min_{\hat{\mathbf{x}}_t} \ell_{DR}(\hat{f}_{B_t}(\hat{\mathbf{x}}_t), f_{B_t}(\mathbf{x}_t)) \tag{6}$$

where ℓ_{DR} is the Mean Squared Error (MSE)-based loss function of the embedding matching function. The logic behind executing this optimization function is: by performing feature embedding matching optimization, $\hat{f}_{B_t}(\hat{\mathbf{x}}_t)$ can be made to approximate $f_{B_t}(\mathbf{x}_t)$. The attacker can then derive the hidden feature information from $\mathbf{x}_t$, successfully achieving the goal of stealing private information.

3.2.2 Attacks to Tabular Data

Privacy inference attacks on tabular data are equally serious, as some studies have shown that racial prediction models can be misused to predict gender [49], posing a serious threat to privacy. The core idea of this kind of attack on tabular data is to capture the correlation between feature embeddings and the corresponding target attribute values and we can call it the feature inference attacks [32]. It can be defined as a multi-class classifier, with each unique attribute category having a class label. Specifically, during the training phase, the attacker first records the feature embeddings of the target data, and during the inference phase, the attacker introduces an auxiliary dataset $\mathbf{x}^{aux}$ and continuously queries the original true bottom model f_{B_t} to obtain its attribute embeddings. Subsequently, let the $S_p = (S_1, S_2, \ldots, S_p)$ denotes the corresponding p features. The attacker uses different attribute embedding values C and the corresponding target attribute values to train a classifier f_{C_p} to infer the attributes of any training data [30]. The task can be represented as:

$$\arg\min_{\omega_{C_p}} \ell_{C_p}(f_{C_p}(f_{B_t}(\mathbf{x}^{aux}); \omega_{C_p}), S_p) \tag{7}$$

where ℓ_{C_p} is the cross-entropy loss and ω_{C_p} is the parameter of classifier. After getting the f_{C_p}, attackers can infer the features of the target data in the training set.

3.3 Defense Goals, Knowledge and Capability

In this section, we will introduce the defense goals, knowledge, and capabilities of our defense method in railway information systems.

Defense goals: The goal of this paper is to design a defense mechanism against privacy inference attacks. In specific terms, the client has added an appropriate regularization term to its embedding, making its sensitivity to the original data as minimal as possible. This defense method should satisfy the two following goals:

- Effectiveness: To defend against privacy leaks, the embedding of local data at the bottom model should be designed to minimize the inclusion of original sensitive information in the embedding results as much as possible.
- Fidelity: To avoid affecting the main task performance, the regularization coefficient should be appropriately selected based on performance feedback.

Defense knowledge and capability: For the clients, they want to prevent their local sensitive information from being leaked. Clients can control their own features and the embedding from their bottom model. Before uploading, add a regularization term to minimize the amount of original data in the embedding as much as possible.

4 SensFL

In this section, we introduce the privacy inference attacks that are currently being faced and the necessity to defend against them. Based on this, we propose a sensitivity regularization protection strategy SensFL, and present detailed computational methods to illustrate how to implement the defense way.

4.1 Motivation

As introduced in Section 3, attackers conduct privacy inference attacks using the feature embedding vectors uploaded by clients to the local bottom model. This is because these embedding vectors are mappings of the local original data and contain a wealth of information. Privacy inference attacks have become a serious threat, where attackers analyze publicly or semi-publicly available datasets to deduce undisclosed sensitive information within the datasets. Defending against privacy inference attacks is crucial, as it relates not only to the protection of individual privacy but also to the security and trustworthiness of data. Although there are many methods currently available to address privacy inference attacks, achieving a balance between privacy protection and model performance remains challenging. Our goal is to discover an efficient defense method that does not affect the accuracy of the main task.

The defense method introduced in this paper is designed to reduce the amount of original data information contained in the embedded vectors, and a low sensitivity will significantly increase the difficulty for attackers to reconstruct the data, thereby preventing the original data from being attacked and ensuring the regular progress of VFL training. As shown in Fig. 2, the normal VFL process involves optimizing the bottom and top models by comparing the predicted labels from the top model with the true labels. However, attackers might exploit the embedding vectors of the bottom model to steal sensitive information. Therefore, we have added a regularization term when updating the bottom model using the embedding vectors and the original data, in order to enhance privacy.

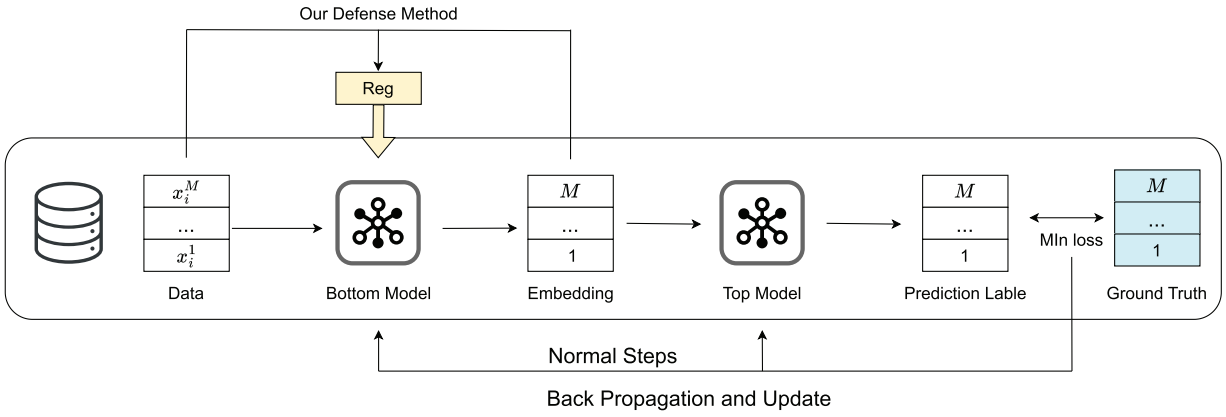

Figure 2: Overview of the defense method

Algorithm 1: Algorithm of SensFL

Require: Model parameters of top model θ_{top} and bottom models $\theta_1, \theta_2, \cdots, \theta_N$, local data of clients $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$, learning rate η, ground truth label y.

Server:

Initialize θ_{top} and $\theta_1, \theta_2, \cdots, \theta_N$
for each training round **do**
  for $n = 1$ to N **do**
    $\mathbf{e}_n \leftarrow f_{B_n}(\mathbf{x}_n)$
  end for
  $\mathbf{e}_{top} \leftarrow F_T(\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_N; \theta_{top})$
  $\mathscr{L} \leftarrow$ **LossFunction**$(\mathbf{e}_{top}, y)$
  $g_{top} \leftarrow \dfrac{\partial \mathscr{L}}{\partial \theta_{top}}$
  $\theta_{top} \leftarrow \theta_{top} - \eta \cdot g_{top}$
  for $n = 1$ to N **do**
    $J_n = ||\dfrac{\partial \mathbf{e}_n}{\partial \mathbf{x}_n}||_2$
    $g_n \leftarrow g_n \cdot \dfrac{\partial \mathbf{e}_n}{\partial \theta_n} + \dfrac{\partial(\epsilon * J_n)}{\partial \theta_n}$
    $\theta_n \leftarrow \theta_n - \eta \cdot g_n$
  end for
end for

4.2 Overview of SensFL

In privacy inference attacks, the mutual information between the original data and the embedded information is often exploited. Inspired by this, we propose SensFL, whose core idea is to reduce the sensitivity between sensitive information and embeddings as much as possible without affecting the main task. Specifically, this method effectively reduces the sensitivity of feature embeddings to the original data by introducing the L2 norm of the Jacobian matrix during the optimization process. Not only does this approach enhance the privacy security of the model, but experiments have also shown that it can ensure that the model's performance on the main task is not significantly affected. The defense method is shown in Fig. 2 and the algorithm is illustrated in Algorithm 1.

To achieve the effectiveness and fidelity goals set in Section 3.3, SensFL considers the following scenarios: there are N clients have their own input $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ and bottom model $\{f_{B_1}, f_{B_2}, \cdots, f_{B_N}\}$, the bottom model is parameterized by θ. Each clinet respectively have the local embeddings $\mathbf{e}_n$ and it can be represented as $\mathbf{e}_n = f_{B_n}(\mathbf{x}_n)$. The information about $\mathbf{x}$ in $\mathbf{e}$ should be as minimal as possible, so the loss function should be:

$$\ell = \ell_{main} + \varepsilon \cdot J \tag{8}$$

where the ℓ_{main} is the loss of the main task, and the J is the L2 norm of the Jacobian matrix for each feature embedding concerning the original data. J can be seen as the sensitivity of the feature embedding vectors for the original data. ε is the coefficient of J, primarily used to control the magnitude of the added regularization term. For each client n, the J_n for $\mathbf{x}_n$ and $\mathbf{e}_n$ can be calculated

as the Eq. (9).

$$J_n = \left\| \frac{\partial \mathbf{e}_n}{\partial \mathbf{x}_n} \right\|_2 \tag{9}$$

To better evaluate SensFL and gain a deeper understanding of its practical applicability in real-world scenarios, we have considered its computational complexity. The core of SensFL calculates the Jacobian matrix for feature embedding $\mathbf{e} \in \mathscr{R}^K$ with data $\mathbf{x} \in \mathscr{R}^P$, so the complexity is $O(K \times P)$, K and P are the dimensions of $\mathbf{e}$ and $\mathbf{x}$.

In summary, our defense method involves calculating the derivative of the sensitivity J_n, which is issued by the clients to the local bottom model parameters during backpropagation, and adding it to the regular backpropagation updates to incorporate regularization into the model. By computing the sensitivity gradient, clients can continuously update their local models and minimize the sensitivity, thereby finding the minimum defense cost.

5 Empirical Evaluation

5.1 Experimental Setup

Datasets: We used four datasets in our experiment: UTKFace, CelebA, Credit, and Bank Marketing. UTKFace and CelebA feature 32 × 32 pixel RGB images, Credit and Bank Marketing is tabular datasets.

Models: For the image datasets UTKFace and CelebA, we constructed the target bottom models based on the residual network structure. Residual networks are widely recognized for their ability to mitigate the vanishing gradient problem in deep networks. We designed three models of varying complexities, each with 1, 2, and 3 residual blocks, to explore the impact of model complexity on privacy protection and primary task performance. The embedding vectors outputted by these bottom models were fed into a four-layer fully connected neural network (FCNN) as the top model. For the tabular datasets Credit and Bank Marketing, we defined the top model as a nonlinear, fully connected network, while the bottom model was constructed with fully connected layers of four different depths. We selected four different network depths to build the bottom models to assess the impact of network depth on feature extraction capabilities. For simplicity and clarity in our presentation, we refer to these three distinct deep target models as M1, M2, and M3. For the image datasets, we set the embedding dimensions to 1000, and for the tabular datasets, to 200. These dimensions were chosen based on their effectiveness demonstrated in previous related research, as well as considering the computational efficiency of the models. During training, we employed mini-batches of size 128 and utilized the cross-entropy loss function to optimize the model's classification performance. We chose the Adam optimizer as our optimization protocol with a learning rate of 1e−4, due to its efficiency and robustness shown across a variety of tasks. In terms of the selection of the regularization coefficient, we based our decisions on empirical experimental results. For the image datasets, we set $\varepsilon \in 0.1$, as experiments revealed that this coefficient effectively balances privacy protection with model performance. For the tabular datasets, we set $\varepsilon \in 0.01$, taking into account the intrinsic characteristics of tabular data and the higher demand for privacy protection.

In our VFL setup, we consider two clients interacting with one server. It is worth noting that our method focuses on reducing the sensitivity between the embeddings and the original data. Each benign client only needs to be concerned with its own sensitivity, so this defense method is independent of the number of clients. However, to assess the impact of different clients on the experimental results, we

conducted validation experiments with three clients for the image dataset. In the experiments across the three clients, we set $\varepsilon \in 1$ and the noise scale to 0.01.

Metrics: For the main task of the VFL experiment, we adopt the standard classification metric accuracy (ACC) as the metric. For the DR attack and the corresponding defense methods, we use different metrics to evaluate the result. When the datasets are images, we adopt the Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM) as evaluation metrics. For tabular datasets, we use Accuracy (ACC), F1 Score, Precision, and Recall as evaluation metrics. This is because metrics such as MSE, PSNR, and SSIM focus on quantifying image quality, reflecting the visual effect and structural similarity from different perspectives, which is crucial for image reconstruction and detecting the effectiveness of defense methods. For tabular data, metrics like ACC, F1 Score, Precision, and Recall are used for assessing the quality of reconstructed tubular attributes. The inference of table attributes can be regarded as a classification task, hence we select standard classification metrics. This is consistent with ML-Doctor [49].

Baseline Methods: We use Differentially-Private Stochastic Gradient Descent (DP-SGD) as our baseline defense method to compare with our approach.

As for the DP, It is a robust statistical method designed to provide privacy protection when querying statistical databases [50]. The core concept of DP is to ensure that for any two adjacent datasets D and D' (which differ by only one record), the output distributions $M(D)$ and $M(D')$ of any algorithm M run on the datasets are statistically indistinguishable. An algorithm M satisfies φ-differential privacy if for all output sets S:

$$\Pr[M(D) \in Q] \leq e^{\varphi} \cdot \Pr[M(D') \in Q] \tag{10}$$

φ is the privacy parameter which quantifies the level of privacy protection, this inequality ensures that the probability of the algorithm output being in a particular set Q does not change significantly when switching between the neighboring datasets D and D'.

DP-SGD is one of the most representative DP mechanisms for protecting ML models. Generally speaking, DP-SGD adds Gaussian noise to the gradient g during the training process of the target ML model.

$$\tilde{g} = g + \mathcal{N}\left(0, \Delta_g^2 \sigma^2 \mathbf{I}\right) \tag{11}$$

$\mathcal{N}\left(0, \Delta_g^2 \sigma^2 \mathbf{I}\right)$ denotes a multi-dimensional random variable sampled from the normal distribution with mean 0 and standard deviation $\Delta_g \sigma$. The Δ_g is the sensitivity of g, and it can not be computed directly because there is no prior knowledge to determine the influence of a single training sample on the gradient g. In DP-SGD, it trims g to $g/\max\{1, ||g||_2/C\}$ to limit the 2-norm of the gradient to C. When $||g||_2 \leq C$, retain g; otherwise, scale down proportionally to the norm of C.

We also identified some newer defensive methods [51,52], but they still have some limitations. For example, some study [51] used RÃ©nyi differential privacy to provide a tighter privacy analysis for the composite Gaussian mechanism, but the price of improving efficiency is the loss of utility. However, our study focuses on discussing the fidelity and effectiveness of defense strategies. Therefore, our method has an advantage while ensuring utility. In some other studies [52], the focus is on the field of text dataset, which does not apply to the scenarios we are discussing.

5.2 The Effectiveness of SensFL

To assess our defense method, we evaluate it using all the datasets and models mentioned above and compare it with the standard defense strategy DPSGD. We present the experimental results for the image dataset and tabular dataset in Tables 1 and 2. In these tables, we denote the results without using any defense methods as 'clean' and set the DPSGD defense's noise scale to 0.1 and 0.01, denoting 'n−0.1' and 'n−0.01'. In Table 2, S1 and S2 represent the two data attributes we selected for the experiment.

Table 1: Performance comparison for privacy inference attacks under clean, DPSGD and our method of UTKFace and CelebA

Dataset	Defense	M1			M2			M3		
		MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
UTKFace	Clean	3.18e−3	29.81	0.94	4.18e−3	28.43	0.94	1.21e−2	24.05	0.87
	n−0.1	4.87e−3	28.11	0.88	3.42e−3	30.45	0.93	1.33e−2	24.18	0.81
	n−0.01	4.58e−3	28.32	0.90	4.05e−3	29.73	0.93	1.34e−2	24.42	0.85
	Our	**4.37e−2**	**16.19**	**0.34**	**4.93e−2**	**15.37**	**0.31**	**5.26e−2**	**14.88**	**0.29**
CelebA	Clean	3.51e−3	29.28	0.94	3.89e−3	27.31	0.92	2.18e−2	21.58	0.79
	n−0.1	4.97e−3	28.81	0.91	4.84e−3	28.26	0.90	2.77e−3	31.19	0.94
	n−0.01	3.18e−3	30.74	0.94	1.24e−2	24.71	0.81	1.03e−2	24.42	0.81
	Our	**6.37e−2**	**15.31**	**0.14**	**7.03e−2**	**14.42**	**0.15**	**5.16e−2**	**15.31**	**0.15**

Table 2: Performance comparison of privacy inference attacks on different features under the clean, DBSGD, and our method in credit and bank marketing

(a) Credit

Feature	Defense	M1				M2				M3			
		ACC	F1	Precision	Recall	ACC	F1	Precision	Recall	ACC	F1	Precision	Recall
S1	Clean	0.99	0.99	0.99	0.99	0.98	0.98	0.98	0.98	0.97	0.97	0.96	0.97
	n−0.1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00
	n−0.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00
	Our	**0.73**	**0.72**	**0.72**	**0.71**	**0.68**	**0.68**	**0.67**	**0.68**	**0.82**	**0.81**	**0.81**	**0.81**
S2	Clean	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.95	0.90	1.00	1.00
	n−0.1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	n−0.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Our	**0.62**	**0.52**	**0.69**	**0.51**	**0.58**	**0.47**	**0.55**	**0.47**	**0.84**	**0.47**	**0.46**	**0.46**

(Continued)

Table 2 (continued)

(b) Bank marketing

Feature	Defense	M1				M2				M3			
		ACC	F1	Precision	Recall	ACC	F1	Precision	Recall	ACC	F1	Precision	Recall
S1	Clean	0.99	0.99	0.99	0.99	0.98	0.96	0.98	0.94	0.95	0.90	0.96	0.85
	n−0.1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	n−0.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Our	**0.87**	**0.67**	**0.72**	**0.71**	**0.85**	**0.46**	**0.67**	**0.68**	**0.84**	**0.47**	**0.81**	**0.81**
S2	Clean	0.97	0.76	0.83	0.75	0.96	0.69	0.82	0.71	0.78	0.54	0.53	0.55
	n−0.1	0.98	0.81	0.84	0.79	0.98	0.81	0.84	0.78	0.98	0.81	0.84	0.79
	n−0.01	0.98	0.80	0.84331	0.785	0.97	0.81	0.83	0.79	0.97	0.80	0.83	0.78
	Our	**0.57**	**0.36**	**0.69**	**0.51**	**0.52**	**0.30**	**0.55**	**0.47**	**0.35**	**0.14**	**0.46**	**0.46**

5.3 The Fidelity of SensFL

Observing the results in Tables 1 and 2, we can draw the following conclusions. First, in the absence of effective defense methods, the more complex the bottom models, the better their defensive capability against DR attack. For example, in CelebA, the SSIM of the complex model (M3) is 0.79, which is lower than that of the simple model (M1) at 0.94. For the Credit dataset, the F1 of attribute S2 in M3 is 0.90, but it is 1 in M1. This result indicates that simpler models possess a more direct functional mapping from private features to the generated feature embeddings, which implies that attackers can more easily reconstruct the data by matching feature embeddings.

Second, the experiment demonstrates that using DPSGD to defend against DR attacks in VFL is ineffective. The reason for this phenomenon is that adding noise to the gradients does not change the relationship between the local private data and its embedding vectors, their connection remains unbroken and fixed. Therefore, attackers can still exploit the embedding vectors for DR attacks. But Sometimes experiments may reveal an interesting phenomenon, where the effect of DR attacks improves after the application of DPSGD. This could be due to the introduction of DPSGD increasing the deviation between the clusters, promoting the classifier to be more generalizable.

Third, our defense method possesses strong cross-model and cross-data defense capabilities. For instance, after employing our defense method, even in the simplest model M1, the SSIM of the results reconstructed from a DR attack on the UTKFace dataset is only 0.34, and in the more complex model M3, it is only 0.29, significantly lower than the situation without added defense measures. In the tabular dataset Credit and M1, the F1 metric for the attribute S2 has dropped to 0.52, and in M3, it has decreased to 0.46. We believe that this defensive capability is due to the regularization term we added, which restricts the embedding vectors from containing features of the original data, significantly increasing the difficulty for attackers to perform reverse engineering.

To test the fidelity of our defense methods on the primary task, we conducted experiments using all the datasets and models mentioned above. We adopted the same experimental setup as in the test for effectiveness, marking the model without defense as ‘Clean’ and the models with noise ratios of 0.1 and 0.01 in DPSGD as ‘n−0.1’ and ‘n−0.01’, respectively.

As the results in the Fig. 3 show, our defense method hardly affects the ACC of the primary task. For example, after applying SensFL, the drop of the main task's ACC is controlled within 0.05. An interesting phenomenon is that when applied to the Credit dataset, the ACC increased slightly. Upon analysis, the reason for this phenomenon may be that the addition of regularization is quite common in the training of machine learning models. It is not only an effective method to prevent model overfitting but also a suitable regularization term that can limit the complexity of the model, thereby improving the model's generalization on different datasets. In contrast, although DPSGD provides privacy protection, it leads to a decrease in the ACC of the main task. For instance, when training on the CelebA dataset with model M1, the ACC without any defense is around 0.85. However, after applying the DPSGD defense method with coefficients of 0.1 and 0.01, the ACC dropped to approximately 0.50 and 0.70, respectively.

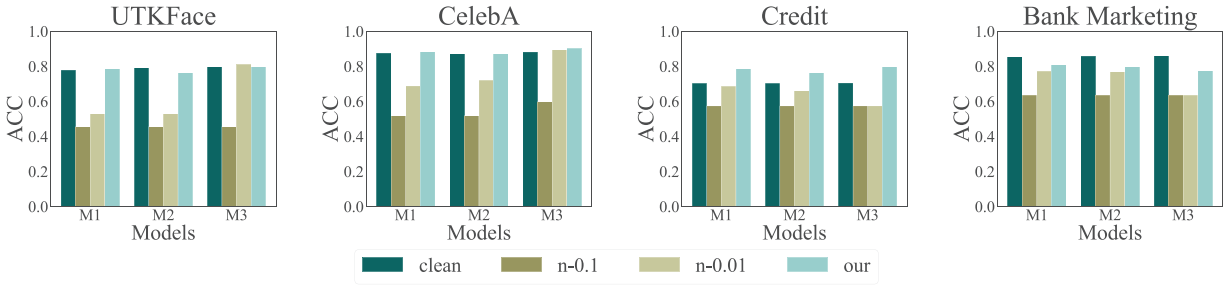

Figure 3: The main task ACC of VFL model with/without defense mechanisms

This may be because the noise introduced interferes with the model's learning process, preventing it from accurately capturing the information in the data.

5.4 The Impact of Different Embedding Dimensions

Embedding dimensions can affect the information encapsulated in the data, which in turn can impact the effectiveness of defenses. To explore this impact, we conducted tests on the aforementioned datasets and models. In UTKFace and CelebA, we set the embedding dimensions to 500, 800, and 1000 while using MSE, PSNR, and SSIM as evaluation metrics. In Credit and Bank Marketing, the embedding dimensions were set to 100, 150, and 200, and evaluations were conducted using ACC, F1, Precision and Recall. Through observation of the results shown in Figs. 4 and 5, we have drawn the following conclusions. First, simply modifying the dimensions does not effectively mitigate the DR attack and can only have a minor impact on the model. This may be because changing the embedding dimensions affects the complexity of the underlying model, which not only may not mitigate the attack but could even inadvertently exacerbate the attack when the complexity is reduced. Second, whether in the image or tabular datasets, our defense method has demonstrated strong defensive capabilities against DR attacks, and this defensive capability is not strongly related to the embedding dimensions.

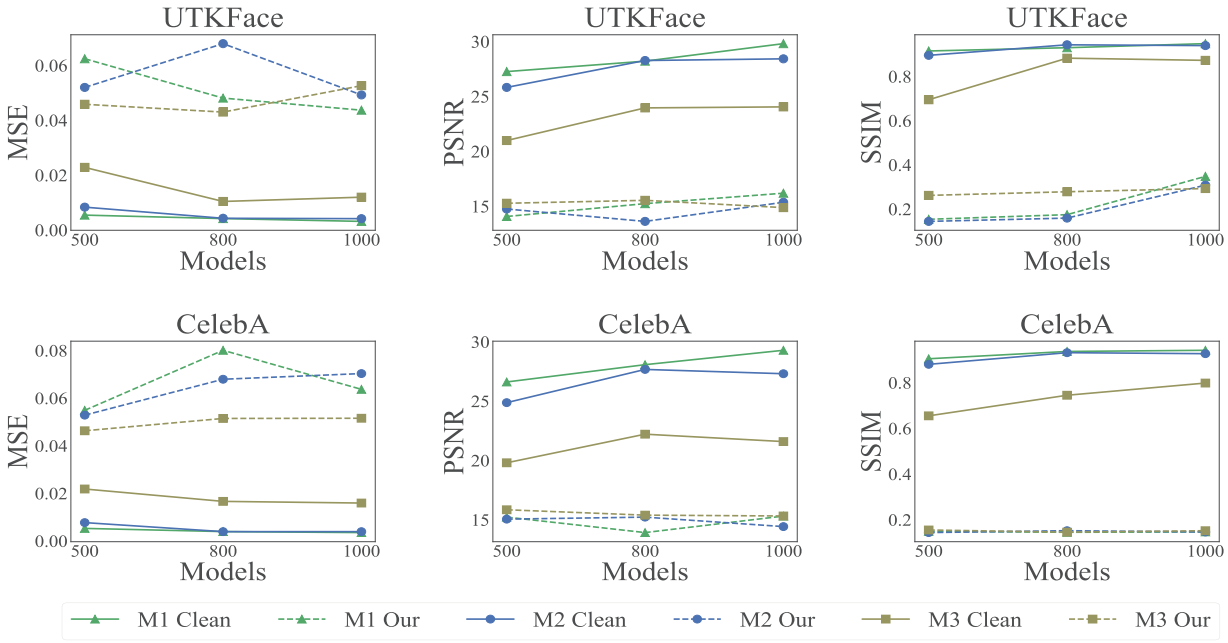

Figure 4: Attack performance under different dimensions of VFL embedding with/without defense in image dataset

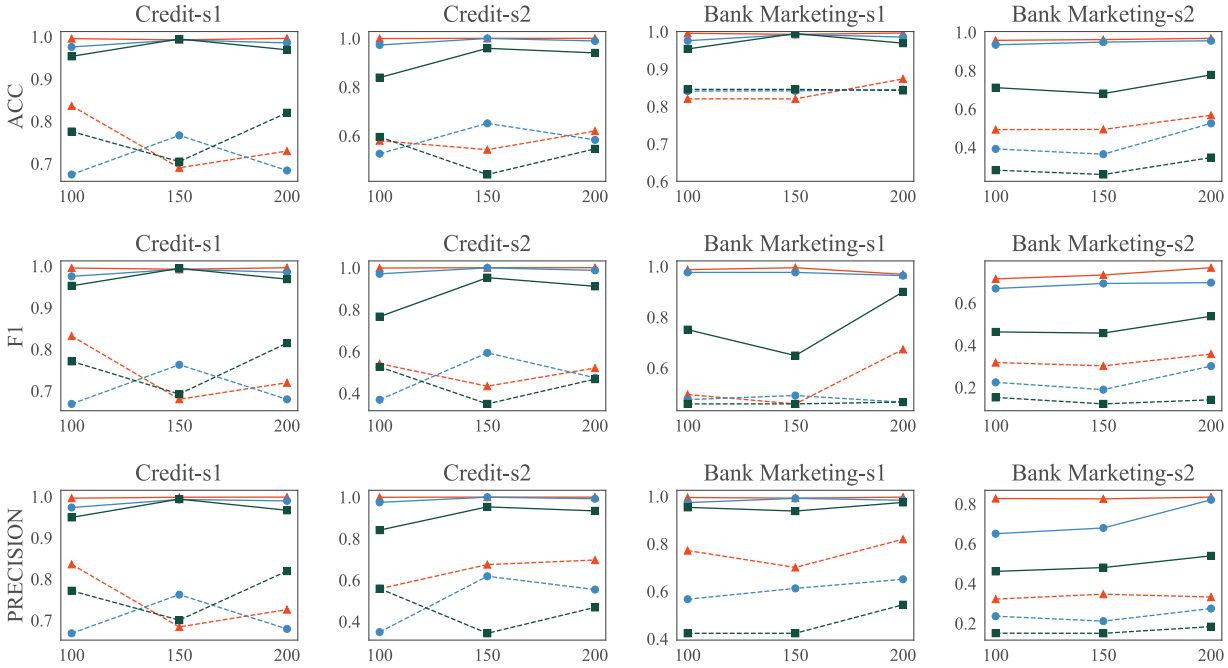

Figure 5: (Continued)

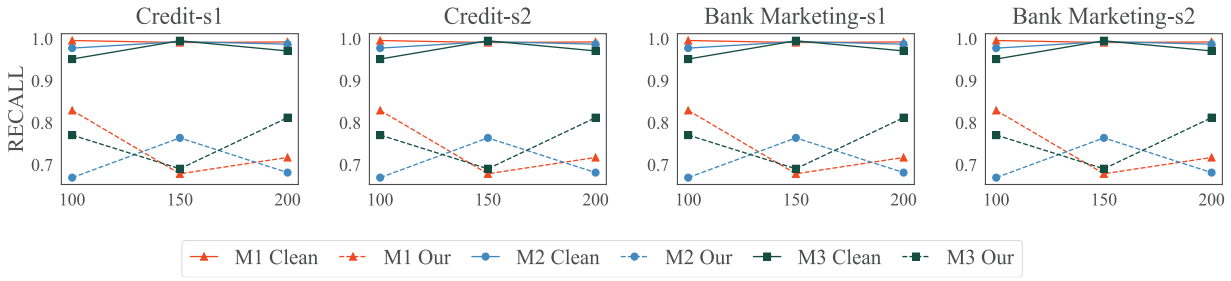

Figure 5: Attack performance under different dimensions of VFL embedding with/without defense in tabular dataset

5.5 The Impact of Different Numbers of Clients

If the number of clients increases, the data characteristics available to each client will correspondingly decrease. This may lead to attributes concentrating around the same patterns, thereby affecting the relationship between the original data and this feature, which in turn may impact both the primary and attack tasks. To conduct a general assessment, we compared the performance of the primary and attack tasks when there were two and three clients. Our evaluation results are presented in Table 3, showing the changes in ACC for the main task and the privacy inference attack evaluation metric when the number of clients is 2 and 3.

Table 3: Performance comparison of the main task ACC and privacy inference attack under different numbers of clients

Dataset	Client numbers	Main task			Attack performence								
		ACC			PSNR			MSE			SSIM		
		M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3
UTKFace	2	0.80	0.79	0.76	16.19	15.37	14.88	4.37e−2	4.93e−2	5.26e−2	0.34	0.31	0.29
	3	0.77	0.78	0.78	17.74	17.32	17.03	4.33e−2	4.31e−2	3.89e−2	0.45	0.42	0.52
CelebA	2	0.85	0.86	0.88	15.31	14.42	15.31	6.37e−2	7.03e−2	5.16e−2	0.14	0.15	0.15
	3	0.88	0.89	0.87	15.29	13.44	16.52	7.70e−2	9.62e−2	5.91e−2	0.24	0.17	0.33

As shown in the Table 3, the accuracy of the main task remains almost unchanged when the number of clients increases. This is because although each client gets fewer features, the total number of features remains the same, so the impact on the primary task accuracy is minimal. For example, in the case of 3 clients, when the dataset is Celeba, the main task ACC for M1, M2, and M3 are 0.88, 0.89, and 0.87, respectively, which are negligible compared to the case with 2 clients.

At the same time, as we expected, the effectiveness of the defense has slightly decreased. This is because when the features of the clients become fewer, the relationship between the original data and the embedding becomes weaker, so the defense effect is slightly worse, but the defense is still successful. For example, in the case of 3 clients, when the dataset is UTKFace, the SSIM values for M1 and M2 are 0.45 and 0.42, respectively, indicating that we have successfully conducted the defense.

6 Conclusion

In this work, we propose a privacy-enhancing method based on sensitivity regularization, which can defend against privacy inference attacks in the VFL model. The main principle is to reduce the connection between the embedding vectors and the input data while retaining the necessary information about the target labels. We tested our method on four datasets and three models, confirming that it can achieve defensive effects without affecting the accuracy of the main task. These results demonstrate its potential to protect sensitive information in the field of Intelligent Railway Transportation Systems, providing support for the advancement of related privacy protection technologies.

Notably, SensFL is not without limitations. The fundamental mechanism of SensFL hinges on the computation of the Jacobian matrix during training to regulate sensitivity. Although this is crucial for mitigating information leakage, it introduces moderate computational overhead, particularly for high-dimensional datasets or large-scale models.

Acknowledgement: The authors are thankful to the anonymous reviewers for improving this article.

Funding Statement: This work was supported by Systematic Major Project of Shuohuang Railway Development Co., Ltd., National Energy Group (Grant Number: SHTL-23-31), and in part by Beijing Natural Science Foundation (U22B2027).

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design, data collection, analysis, and interpretation of results: Chongzhen Zhang, Zhichen Liu and Wei Wang; draft manuscript preparation, and editing: Chongzhen Zhang, Zhichen Liu and Xiangrui Xu; validation: Fuqiang Hu, Jiao Dai and Baigen Cai. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Based upon reasonable request, data can collect from the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Peng JL, Cheng S, Diau E, Shih YY, Chen PH, Lin YT, et al. A survey of useful LLM evaluation. arXiv:2406.00936. 2024.
2. Wang W, Shang Y, He Y, Li Y, Liu J. BotMark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. Inform Sci. 2020;511(3):284–96. doi:10.1016/j.ins.2019.09.024.
3. Maqsood M, Yasmin S, Gillani S, Bukhari M, Rho S, Yeo SS. An efficient deep learning-assisted person re-identification solution for intelligent video surveillance in smart cities. Front Comput Sci. 2023;17(4):174329. doi:10.1007/s11704-022-2050-4.
4. Mamvong J, Goteng G, Gao Y. Low-cost client-side encryption and secure Internet of Things (IoT) provisioning. Front Comput Sci. 2022;16(6):166824. doi:10.1007/s11704-022-1256-9.
5. Munim ZH, Dushenko M, Jimenez VJ, Shakil MH, Imset M. Big data and artificial intelligence in the maritime industry: a bibliometric review and future research directions. Marit Pol Manag. 2020;47(5):577–97. doi:10.1080/03088839.2020.1788731.

6. Jagatheesaperumal SK, Rahouti M, Ahmad K, Al-Fuqaha A, Guizani M. The duo of artificial intelligence and big data for Industry 4.0: applications, techniques, challenges, and future research directions. IEEE Internet Things J. 2022;9(15):12861–85. doi:10.1109/JIOT.2021.3139827.
7. Jan Z, Ahamed F, Mayer W, Patel N, Grossmann G, Stumptner M, et al. Artificial intelligence for industry 4.0: systematic review of applications, challenges, and opportunities. Expert Syst Appl. 2023;216:119456. doi:10.1016/j.eswa.2022.119456.
8. Wang W, Suo X, Wei X, Wang B, Wang H, Dai HN, et al. HGATE: heterogeneous graph attention auto-encoders. IEEE Trans Knowl Data Eng. 2023;35(4):3938–51. doi:10.1109/TKDE.2021.3138788.
9. Xu X, Liu P, Wang W, Ma HL, Wang B, Han Z, et al. CGIR: conditional generative instance reconstruction attacks against federated learning. IEEE Trans Depend Secur Comput. 2023;20(6):4551–63. doi:10.1109/TDSC.2022.3228302.
10. Bochicchio M, Zeleke SN. Personalized federated learning in edge-cloud continuum for privacy-preserving health informatics: opportunities and challenges. In: Barolli L, editor. Advanced information networking and applications. Cham: Springer Nature Switzerland; 2024. p. 368–78. doi:10.1007/978-3-031-57931-8_36.
11. Wang R, Liu X, Xie L, Liu Y, Su Z, Liu D, et al. Privacy-preserving incentive scheme design for UAV-enabled federated learning. In: 2024 IEEE Wireless Communications and Networking Conference (WCNC), 2024; Dubai, United Arab Emirates; p. 1–6. doi:10.1109/WCNC57260.2024.10571180.
12. Mandal S. A privacy preserving federated learning (PPFL) based cognitive digital twin (CDT) framework for smart cities. Proc AAAI Conf Artif Intell. 2024 Mar;38(21):23399–400. doi:10.1109/WCNC57260.2024.10571180.
13. Liu X, Li J, Chen S, Jiang X, Yang F, Yang J. Privacy-preservation robust federated learning with blockchain-based hierarchical framework. In: Proceedings of the International Conference on Computing, Machine Learning and Data Science (CMLDS '24), 2024; New York, NY, USA: Association for Computing Machinery. doi:10.1145/3661725.3661726.
14. Yu S, Munoz JP, Jannesari A. Federated foundation models: privacy-preserving and collaborative learning for large models. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 2024; Torino, Italia: ELRA and ICCL.
15. Xu X, Liu P, Zhao Y, Han L, Wang W, Zhu Y, et al. Enhancing privacy in distributed intelligent vehicles with information bottleneck theory. IEEE Internet Things J. 2024. doi:10.1109/JIOT.2024.3434627.
16. Zhang S, Li J, Shi L, Ding M, Nguyen DC, Tan W, et al. Federated learning in intelligent transportation systems: recent applications and open problems. IEEE Trans Intell Trans Syst. 2024;25(5):3259–85. doi:10.1109/TITS.2023.3324962.
17. Zhang R, Wang H, Li B, Cheng X, Yang L. A survey on federated learning in intelligent transportation systems. arXiv:2403.07444. 2024.
18. Liu Y, Yi Z, Chen T. Backdoor attacks and defenses in feature-partitioned collaborative learning. arXiv:2007.03608. 2020.
19. Liu Y, Kang Y, Zou T, Pu Y, He Y, Ye X, et al. Vertical federated learning: concepts, advances, and challenges. IEEE Trans Knowl Data Eng. 2024;36(7):3615–34. doi:10.1109/TKDE.2024.3352628.
20. Fu C, Zhang X, Ji S, Chen J, Wu J, Guo S, et al. Label inference attacks against vertical federated learning. In: 31st USENIX Security Symposium (USENIX Security 22), 2022; Boston, MA: USENIX Association; p. 1397–414.
21. Naseri M, Han Y, Cristofaro ED. BadVFL: backdoor attacks in vertical federated learning. arXiv:2304.08847. 2023.
22. Xu X, Wang W, Chen Z, Wang B, Li C, Duan L, et al. Finding the PISTE: towards understanding privacy leaks in vertical federated learning systems. IEEE Trans Dependable Secur Comput. 2024;1–14. doi:10.1109/TDSC.2024.3445600.

23. McMahan B, Moore E, Ramage D, Hampson S, BAy Arcas. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 2017; Fort Lauderdale, FL, USA: PMLR; vol. 54, p. 1273–82.
24. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol. 2019 Jan;10(2):1–19. doi:10.1145/3298981.
25. Cheng Y, Liu Y, Chen T, Yang Q. Federated learning for privacy-preserving AI. Commun ACM. 2020 Nov;63(12):33–6. doi:10.1145/3387107.
26. Praharaj L, Gupta M, Gupta D. Hierarchical federated transfer learning and digital twin enhanced secure cooperative smart farming. In: 2023 IEEE International Conference on Big Data (BigData), 2023; Sorrento, Italy; p. 3304–13. doi:10.1109/BigData59044.2023.10386345.
27. Liu P, Xu X, Wang W. Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. Cybersecurity. 2022;5(1):1–19. doi:10.1186/s42400-021-00105-6.
28. Xie C, Huang K, Chen PY, Li B. DBA: distributed backdoor attacks against federated learning. In: International Conference on Learning Representations, 2020.
29. Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning. In: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, 2020; Cambridge, MA, USA: PMLR; vol. 108, p. 2938–48.
30. Ye P, Jiang Z, Wang W, Li B, Li B. Feature reconstruction attacks and countermeasures of DNN training in vertical federated learning. arXiv:2210.06771. 2022.
31. Weng H, Zhang J, Ma X, Xue F, Wei T, Ji S, et al. Practical privacy attacks on vertical federated learning. arXiv:2011.09290. 2020.
32. Jin X, Chen PY, Hsu CY, Yu CM, Chen T. CAFE: catastrophic data leakage in vertical federated learning. arXiv:2110.15122. 2021.
33. Luo X, Wu Y, Xiao X, Ooi BC. Feature inference attack on model predictions in vertical federated learning. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), 2021; Chania, Greece; p. 181–92. doi:10.1109/ICDE51399.2021.00023.
34. He Z, Zhang T, Lee RB. Model inversion attacks against collaborative inference. In: Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC '19), 2019; New York, NY, USA: Association for Computing Machinery; p. 148–62. doi:10.1145/3359789.3359824.
35. Li L, Liu J, Cheng L, Qiu S, Wang W, Zhang X, et al. CreditCoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles. IEEE Trans Intell Trans Syst. 2018;19(7):2204–20. doi:10.1109/TITS.2017.2777990.
36. Yang Y, Zhang G, Li S, Liu Z. Offline/online attribute-based searchable encryption scheme from ideal lattices for IoT. Front Comput Sci. 2023;18(3):1–3. doi:10.1007/s11704-023-3128-3.
37. Yang Y, Dong X, Cao Z, Shen J, Dou S. IXT: improved searchable encryption for multi-word queries based on PSI. Front Comput Sci. 2023;17(5):175811. doi:10.1007/s11704-022-2236-9.
38. Hardy S, Henecka W, Ivey-Law H, Nock R, Patrini G, Smith G, et al. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. arXiv:1711.10677. 2017.
39. He D, Du R, Zhu S, Zhang M, Liang K, Chan S. Secure logistic regression for vertical federated learning. IEEE Internet Comput. 2022;26(2):61–8. doi:10.1109/MIC.2021.3138853.
40. Cheng K, Fan T, Jin Y, Liu Y, Chen T, Papadopoulos D, et al. SecureBoost: a lossless federated learning framework. IEEE Intell Syst. 2021;36(6):87–98. doi:10.1109/MIS.2021.3082561.
41. Chamani JG, Papadopoulos D. Mitigating leakage in federated learning with trusted hardware. arXiv:2011.04948. 2020.
42. Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Halevi S, Rabin T, editors. Theory of cryptography. Berlin, Heidelberg, Berlin Heidelberg: Springer; 2006. p. 265–84. doi:10.1007/11681878_14.

43. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv:1503.02531. 2015.
44. Wang C, Liang J, Huang M, Bai B, Bai K, Li H. Hybrid differentially private federated learning on vertically partitioned data. arXiv:2009.02763. 2020.
45. Dryden N, Moon T, Jacobs SA, Van Essen B. Communication quantization for data-parallel training of deep neural networks. In: 2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC), 2016; Salt Lake City, UT, USA, IEEE; p. 1–8. doi:10.1109/MLHPC.2016.004.
46. Aji AF, Heafield K. Sparse communication for distributed gradient descent. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017; Copenhagen, Denmark: Association for Computational Linguistics. doi:10.18653/v1/D17-1045.
47. Yazdinejad A, Dehghantanha A, Karimipour H, Srivastava G, Parizi RM. A robust privacy-preserving federated learning model against model poisoning attacks. IEEE Trans Inf Forens Secur. 2024;19: 6693–708. doi:10.1109/TIFS.2024.3420126.
48. Yazdinejad A, Dehghantanha A, Srivastava G, Karimipour H, Parizi RM. Hybrid privacy preserving federated learning against irregular users in next-generation internet of things. J Syst Archit. 2024;148:103088. doi:10.1016/j.sysarc.2024.103088.
49. Liu Y, Wen R, He X, Salem A, Zhang Z, Backes M, et al. ML-Doctor: holistic risk assessment of inference attacks against machine learning models. In: 31st USENIX Security Symposium (USENIX Security 22), 2022; Boston, MA, USA: USENIX Association; p. 4525–42.
50. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16), 2016; New York, NY, USA: Association for Computing Machinery; p. 308–18. doi:10.1145/2976749.2978318.
51. Xu J, Zhang W, Wang F. A$(DP)^2$2SGD: asynchronous decentralized parallel stochastic gradient descent with differential privacy. IEEE Trans Pattern Anal Mach Intell. 2022;44(11):8036–47. doi:10.1109/TPAMI.2021.3107796.
52. Dupuy C, Arava R, Gupta R, Rumshisky A. An efficient DP-SGD mechanism for large scale NLU models. In: ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022; Singapore; p. 4118–22. doi:10.1109/ICASSP43922.2022.9746975.