VF-FD: Feature Deduplication for Vertical Federated Learning

Ziyi Li[1]*, Xiao Yan[2*✉], Yuanyuan Zhu[1], Ruixuan Zhang[1], Hao Huang[1], Qinbo Zhang[1], Guojia Wan[1], and Jiawei Jiang[1✉]

[1] School of Computer Science, Wuhan University
{lzy0323, yyzhu, zrx123, haohuang, qinbo_zhang, guojiawan, jiawei.jiang}@whu.edu.cn
[2] Centre for Perceptual and Interactive Intelligence (CPII)
yanxiaosunny@gmail.com

Abstract. Vertical federated learning (VFL) assumes that the features of data samples are scattered over the clients. In practice, the clients may hold overlapping and noisy features, and we conduct feature deduplication by choosing one value from these overlapping features for each sample. Existing feature selection methods cannot handle feature deduplication because they choose the same set of features for all samples while feature deduplication is more flexible and can choose different features for different samples. As such, we propose a method called VF-FD to conduct feature deduplication. VF-FD first identifies groups of possibly overlapping features and then conducts selection in each overlapping group. In particular, VF-FD uses the earth mover distance (EMD) to quantify the similarity of features and decides that two features are overlapping when their EMD is small. For selection, VF-FD trains a surrogate model with all features and deduces a loss sensitive score (LSS) to quantify the influence of each feature on the loss. For each overlapping feature group, a sample selects the feature component that minimizes its loss. We experiment VF-FD on 6 datasets and compare it with state-of-the-art feature selection methods. The results show that VF-FD can improve model accuracy by more than 10% over the best baseline.

Keywords: Vertical federated learning · Feature deduplication · Loss sensitive score.

1 Introduction

Vertical federated learning, denotes as VFL, aims at training model when the features of data samples are scattered over multiple participants (also called clients) [19, 21, 29, 34]. This usually happens when different institutions have different records for the same set of entities [20, 35]. For instance, in Fig 1, each person is an employee of the internet company and a customer of the bank; and the internet company and bank may collaborate via VFL to train a model to

* Equal contribution ✉ Corresponding author

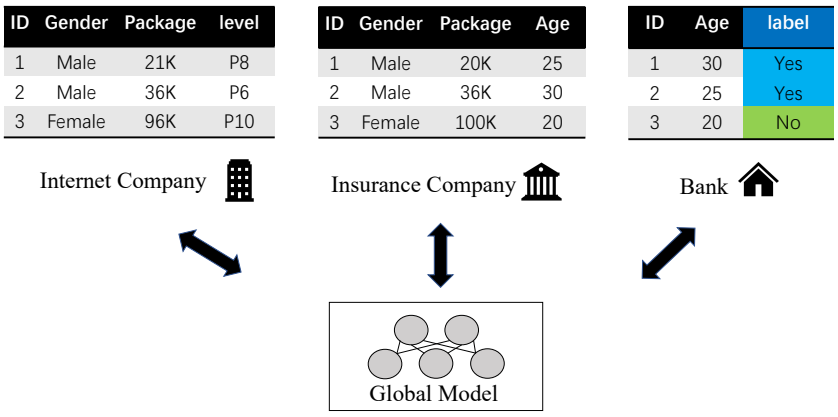


Fig. 1: An example of overlapping features in VFL.

predict whether a person has financial risks. Due to privacy concerns, VFL does not allow the clients to exchange raw data.

While existing VFL researches assume that the clients hold distinct features, we consider a more practical case where the clients may have overlapping and noisy features. Fig 1 provides such an example. Both the internet company and the insurance company have gender as a feature because they collect general information about the persons. Since their gender columns are the same, we can simply select an arbitrary column for model training. However, the situation is more challenging for the 'Age' feature. Both the insurance company and the bank record age but they have different values for the same person. For instance, the insurance company records an age of 25 for ID-1 while the bank records an age of 30. This can be caused by errors in the data collection process, which are common in practice. Moreover, different clients may have errors for different samples since errors are somewhat random. This means that for each set of overlapping features, we should select different features for different samples to conduct training. For instance, ID-1 should use Age 25 from the Insurance company while ID-2 should use Age 25 from the bank, and 'Package' in two clients is the same case. This motivates our feature deduplication problem, which selects a value from each set of overlapping features for each sample such that a model trained on the selected features yield good accuracy.

Some works tackle feature selection in VFL setting [4, 11, 15] but they are not suitable for feature deduplication. This is because they select the same set of features (i.e., feature-wise selection) for all samples while feature deduplication is more fine-grained and can select different features for different samples (i.e., sample-wise selection). For instance, FEAST [11] utilizes Conditional Mutual

Information (CMI) [32] to select features and eliminate similar features directly without addressing noises at the sample level. FedSDG-FS [15] selects features during training and utilize Gini-impurity for feature importance initialization to reduce the training overhead, yet without evaluation of feature components in each sample. LESS-VFL [4] assumes that each client learns a feature embedding and selects embedding features using group lasso model during training. However, it does not yet evaluate embedding feature components from each client.

To solve the feature deduplication problem in VFL setting, we propose a method called VF-FD, which first identifies groups of overlapping features and then conducts selection individually for each sample in each overlapping feature group. In particular, VF-FD addresses the following two technical challenges:

❶ *How to identify the overlapping features?* Different from Fig 1, in practical cases, the column names may be unavailable or mismatched, and thus we need to identify overlapping features according to data. To this end, we use the earth mover distance (EMD) to measure the similarity of two feature columns. EMD is widely used to quantify the similarity of distributions, and its value is small if one distribution can be easily transformed into the other. To compute the EMD, we normalize each feature and collect the number of samples that fall into bins to get a distribution. Two features are decided to be overlapping if their EMD is below a predefined threshold. As naively assigning the clients to communicate with each other to identify the overlapping features is expensive, we organize the clients into a ring, where each client forwards the distributions and identified overlapping groups to the subsequent client.

❷ *How to select proper feature component for each sample?* For each sample, we should select one feature component from each overlapping feature group. Since our goal is to achieve good model accuracy on the selected features, we train a surrogate model such that supervision signals can be exploited to guide selection. In particular, the surrogate model is a simple multi-layer perceptron (MLP); using the Taylor expansion, we deduce a loss sensitive score (LSS), which measures how deleting each feature component affects the loss of the sample. With the LSS, we choose the feature component that minimizes the sample-wise loss function in each overlapping feature group. The rationale is that by minimizing the loss, the selected components agree better with the labels, and thus improve subsequent model training.

To evaluate VF-FD, we experiment on 6 datasets with 4 state-of-the-art baselines. The results show that VF-FD consistently yields higher model accuracy than all baselines, and improves the accuracy of the best-performing baseline by 11.0% on average and 22.4% at the maximum. Moreover, ablation study shows that both the EMD-based overlapping feature identification and LSS-based feature component selection of VF-FD are crucial for accuracy. We also observe that VF-FD is comparable to the baselines in terms of end-to-end efficiency while protecting privacy.

To summarize, we make the following contributions:

- We now formulate the feature deduplication problem in VFL, which enables the clients to eliminate overlapping noisy features.

– We design a method based on earth mover distance (EMD) to identify overlapping features. The method is data driven and does not rely on the names of features.
– We deduce loss sensitive score (LSS) to conduct fine-grained sample-wise feature selection, using supervision from labels to select informative feature components.
– We provide extensive experiments for our proposed framework to demonstrate the superiority of VF-FD with other baselines.

2 Related Work

2.1 Feature Selection in Centralized Learning

In centralized setting, all the data are collected together on one party and accessed easily without consideration of privacy. Feature selection method in centralized settings can be divided into three categories, i.e., filter methods, wrapper methods, and embedded methods.

Filter methods using relevance score of each feature to select the best subset, such as pearson correlation coefficient [7] and mutual information (MI) [26] and its variants [8]. Wrapper method [10, 12, 24] aims to find the optimal subset of features by iteratively adding or removing features in subset of features and generally computationally expensive. mRMR algorithm is a typical wrapper method, aims at finding the features set that are most relevant to the final output while maintaining minimal redundancy among the features [25]. Embedded method aims to select feature while simultaneously learning the model. Recently most of the embedded method are AutoEncoders (AE)-based [16], SDAE [13] add a Selective Layer to AE, and Lasso regularization is applied to both the selective layer and the encoder to make weights in layer to be sparse, which is more interpretable for the relationship between features and weights. Agnostic feature selection (AgnoS) [9] adopts AE with different regularization, such as AgnoS-G, which uses $\ell_{2,1}$ regularization on gradient. These approach inspires us to explore the relationship between gradients, model weights and features.

Since these methods are designed for centralized learning setting, they are not applicable to VFL setting, which demands privacy protection for data separated over multiple clients, while still ensuring the accuracy and efficiency in training.

2.2 Feature Selection in VFL

In vertical federated learning setting, data are divided by feature spaces, which means more attention should be paid to communication and computation cost. [17] introduces a Secure Multiparty Computation (MPC) protocol for privacy feature selection based on filter method in VFL setting. [11] select informative features using filter method based on conditional mutual information (CMI) scores by iterative method and utilize statistical variable generation method to reduce the communication cost and protect raw data. Another approach FedSDG-FS [15] is

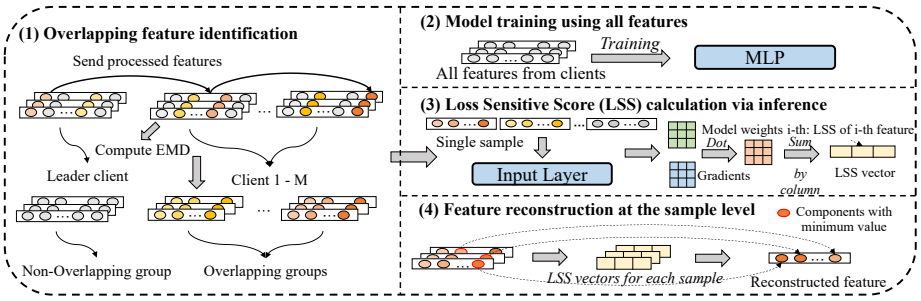


Fig. 2: The overview of VF-FD Framework. (1) Overlapping Feature Identification. (2) Training by whole feature space F. (3) Calculate Loss Sensitive Score (LSS) of each sample. (4) Feature Reconstruction with LSS at the sample level.

an embedded feature selection method, adopting Gaussian stochastic dual-gates for clients' inputs to efficiently approximate the probability of a feature being selected and using Gini impurity for feature importance initialization [5], which can reduce training overhead, but still suffers from features with only a subset of the samples have noisy features. The proposed framework VF-FD addressed these limitations of the state of the art methods.

However, the method achieves deduplication only at the feature level, without considering the feature components in each sample.

3 Problem Formulation

We now formulate the feature deduplication problem for vertical federated learning. We consider a set of M clients $C = \{c_1, c_2, \ldots, c_M\}$, and each client c_k owns a dataset $D^k \in \mathbb{R}^{N \times d_k}$, where $k \in \{1, ..., M\}$, N and d_k denote the number of samples and features held by client c_k, note that N is same for all the clients and we assume that all the clients's sample are aligned before [6, 11, 15]. Among all clients M, there is a leader client who holds the whole instance labels Y and has the duty to communicate with server. Without loss of generality, we assume client c_1 as the leader. We denote the feature set in c_k as $F_k = \{f_1, ..., f_{d_k}\}$. The whole feature space is $F = \{F_1, ..., F_M\}$, utilized for downstream training task.

We consider a scenario where clients possess overlapping features, implying redundant in training task, and some samples of them may exhibit noise. The goal of feature deduplication in VFL is to reduce overlapping features, which may be duplicated features, similar noisy features or high correlated features. And then construct a subset of features to train a better global model $\hat{\theta}$, We formulate the training objective by minimizing the risk as follows:

$$R(\theta) = \mathbb{E}_{X,Y}\mathcal{L}(g(F_1, F_2, ..., F_M); y_n) \tag{1}$$

where $g(\cdot)$ is the feature deduplication strategy, which identifies and deduplicates overlapping features. And $\mathcal{L}(\cdot)$ is the loss function. $g(\cdot)$. It is evident that

Algorithm 1 Overlapping Feature Identification

Input: Clients $C=\{c_1, c_2, ..., c_M\}$, similarity threshold τ
Output: List of overlapping feature groups L.

Initialization: $L \leftarrow \emptyset$
for $i = 1$ **to** $M - 1$ **do**
  Client c_i send nomarlized and binned F_i to c_{i+1}
  for all $f \in F_i$ and $f' \in F_{i+1}$ **do**
    Compute EMD (f, f')
    if $EMD < \tau$ **then**
      Add $\{f, f'\}$ to overlapping group G_{match}
      if $f' \in$ existing overlapping group G **then**
        Update group G_{match} with $G \cup G_{match}$
        Remove old group from L
      end if
    end if
    if $G_{match} \neq$ NULL **then**
      Add G_{match} to L
    end if
  end for
end for
return L

the design of $g(\cdot)$ is the most crucial part of feature deduplication. A desired VFL framework should be able to judge the noise in features by their effects to the objective of the global model and enable all the clients to jointly train a simple global model by a small number of high quality features while eliminating overlapping noisy features. Accordingly, this paper will develop an efficient framework for feature deduplication.

4 The VF-FD Framework

In this section, we will present the formulation and architecture of VF-FD.

4.1 Overview

Fig 2 illustrates an overview of VF-FD for feature deduplication, which consists of two core steps, an overlapping feature identification step and a sample-wise feature reconstruction step. In the former step, overlapping features are grouped, with similar features being placed in the same group. In the latter step, we select feature component from each overlapping group and reconstruct features by using selected components to obtain high quality features. In the following, we will explain the workflow and each step in detail.

4.2 Overlapping Feature Identification

To perform feature deduplication in original feature space from all the clients, our first goal is to identify which features are considered as overlapping features. For

the purpose of quantifying the correlation of features, we utilize Earth Mover's Distance (EMD) [27] as the criterion for similarity evaluation, which estimates the distance between two distributions.

EMD is designed to solve a special case of transportation by linear optimization, which has been widely applied in the fields of image classification [18, 33], recommendation systems [23] and other areas. We employ EMD as criterion for similarity evaluation between features because it does not suffer from arbitrary quantization problems due to rigid binning strategies resulting its robustness to errors in transformation that take raw data into feature space [2].

Considering that grouping among all the clients is inefficient in VFL, we present an iterative overlapping feature identification method. Algorithm 1 details the overlapping feature identification method. The process begins with leader client by sending the local features to the next client in line 3, as each client hold a subset of whole feature space in VFL with each feature has different ranges of values, features should be normalized and binned beforehand to assess EMD between features. Another motivation for this pre-processing is for privacy concern, features after binning and normalization do not retain the original value, which can not be reconstructed. After receiving the processed features from the last client, the similarity of features between two clients is assessed by EMD. Assuming that processed features i in client c is $\hat{f}_c$, EMD between two features $\hat{f}_c$ and $\hat{f}'_c$ can be assessed as follows:

$$\begin{aligned} \mathrm{EMD}(\hat{f}_c, \hat{f}'_c) &= \sum_{j=1}^{n} \sum_{i=1}^{n} |p_{c,i} - p_{c',j}| \cdot w_{ij} \\ \sum_{j=1}^{n} w_{ij} = 1, \sum_{i=1}^{n} w_{ij} &= 1, w_{ij} = \{0, 1\} \end{aligned} \tag{2}$$

where w_{ij} is the weights in i-th bin and j-th bin between feature $\hat{f}_c$ and $\hat{f}'_c$, $p_{c,i}$ is the corresponding probability distribution in i-th bin in client c, c' similarly.

If the EMD between two features is below the similarity threshold τ, a new overlapping group is formed and updated with group beforehand, as detailed in line 6-12. Through this process, overlapping groups between clients are generated and subsequently sent back to leader with non-overlapping features in one group.

4.3 Loss Sensitive Score

After obtaining the overlapping feature groups, We need to deduplicate features in each group. Previous approaches adopt feature selection strategy to model noisy features [12, 15] or eliminate overlapping features [11] at the feature level. However, we address this problem at the sample level by assessing the sensitivity to loss function of each single sample's feature components, which involves inferring one single sample at a time. And we assess the loss sensitivity of the feature components by combining the model weights of the input layer with the gradients, which we called LSS (Loss Sensitive Score). Assuming sample

Algorithm 2 Sample-Wise Feature Reconstruction

Input: List of overlapping feature groups $L = \{G_1, G_2, ...G_N\}$, dataset $X = \{x_1, x_2, ..., x_n\}$
Output: Reconstructed feature group G_{result}
1: **Initialization**: $W \in R^n, s.t. w_{ij} \in [0, 1]$.
2: **Training**: Train a MLP with and Update W.
3: **for all** x_i in X **do**
4: Compute the gradient of the input layer $\nabla f(W)^T$.
5: Compute $LSS(i) = \sum_{j=1}^{q} \left(W \cdot \nabla f(W)^T\right)_{j,i}$
6: Select the feature component a_i with the minimum score from each overlapping feature group $G_k \in L$.
7: **end for**
8: Reconstruct feature f_i from G_k using $\{a_1, a_2, ..., a_n\}$.
9: Add f_i to G_{result}.
10: **return** G_{result}

$x = \{a_1, a_2, ..., a_{dk}\}$, where a_i is the i-th component of sample x, the feature components of sample x is evaluated as follows:

$$LSS(i) = \sum_{j=1}^{q} \left(W \cdot \nabla f(W)^T\right)_{j,i} \tag{3}$$

where q is the number of neurons in the input layer. W is the model weights and $\nabla f(W)^T$ is the gradient vector of input layer for single sample x. As d_k is the number of features in client c_k, W and $\nabla f(W)^T$ are both $q \times d_k$ matrix, Similarly, $W \cdot \nabla f(W)^T$ is also a $q \times d_k$ matrix, which we called loss sensitive score matrix. After the element-wise product of two matrices and sum according to the dimension of neurons in the input layer, a $d_k$vector is obtained, which represents the sum of loss sensitivities of each feature to the neurons in the input layer. LSS (i) is the i-th component of the vector. Let $W \in R^n$ denote the vector of weights, each weight is constrained within the range of 0 to 1 during training to depict the relevancy of features and make LSS more interpretable [30, 31]. In the following part, we will give an explanation of how this metric applies to our sample-wise feature reconstruction and its feasibility.

4.4 Sample-wise Feature Reconstruction

During the Feature Reconstruction process, a simple MLP is built on the leader client, using all processed features from clients. The simple architecture of model makes it more explainable. With inference phase utilizing each single sample as input, we can obtain the gradient of the model's input layer for single samples. The entire process uses the processed features for privacy concern. After selecting the feature component of each sample in each overlapping feature group with the minimum LSS, a new set of reconstructed features is ultimately obtained using features of overlapping feature groups. Algorithm 2 details this process.

Table 1: Statistics of the datasets.

Dataset	**Sonar**	**Phishing**	**PhysioNet**	**Android**	**Nomao**	**Mushroom**
# Samples	208	10,000	12,000	29,332	34,465	61,069
# Features	100	83	70	141	195	36

The motivation of our strategy is that feature component with smaller LSS has more positive impact on model's performance, the theory behind this is:

Theorem 1. *Let $f : \mathbb{R} \to \mathbb{R}$ be a continuously differentiable function that accepts a d-dimensional vector as input and outputs as a scalar. Suppose f represents the loss function linked to with the model weights, As ΔW denotes a very small increment of model weights, f can be approximated by the first-order of Taylor Expansion:*

$$f(W + \Delta W) \approx f(W) + \Delta W \cdot \nabla f(W)^T$$

During the feature deduplication process, the elimination of the feature implies setting the corresponding weight of that feature to zero. Consequently, as the majority of feature components are removed, the following relationship holds:

$$W + \Delta W \to 0 \tag{4}$$

which implies $\Delta W \to -W$. And the loss function of feature deduplication process can be represented approximately by the following expression:

$$f(W + \Delta W) \to f(W) - W \cdot \nabla f(W)^T \tag{5}$$

According to Equation 5, to minimize the loss during feature deduplication process, the matrix $W * \nabla f(W)^T$ should be maximized, Consequently, feature components with larger LSS should be eliminated to reduce greater losses. This strategy encourage us to select the feature components with smallest LSS in each overlapping feature group.

5 Experiments

In this section, we will evaluate the performance of VF-FD with state-of-art baselines on various datasets. And we will also perform extensive experiments to evaluate the efficiency of VF-FD and the effectiveness of each step.

5.1 Experiment Settings

Datasets. We use six datasets for evaluation, with all of the datasets from real-world scenarios, as shown in Table 1. For each dataset, we randomly select 20% of

Table 2: Accuracy comparison for the evaluated methods (mean std). The best and runner-up are marked with bold and underlined, respectively.

Datasets	All Features	Random	MI	Local Lasso	Global Lasso	FEAST	VF-FD
Sonar	<u>72.86 ± 0.04</u>	48.57 ± 0.05	60.48 ± 0.08	49.52 ± 0.06	71.90 ± 0.08	60.50 ± 0.08	**82.86 ± 0.04**
Phishing	<u>95.19 ± 0.01</u>	50.10 ± 0.10	63.86 ± 0.01	49.54 ± 0.01	94.69 ± 0.01	59.25 ± 0.11	**96.62 ± 0.02**
PhysionNet	69.63 ± 0.19	70.69 ± 0.19	73.66 ± 0.24	74.22 ± 0.17	68.42 ± 0.14	<u>86.23 ± 0.01</u>	**86.44 ± 0.00**
Android	<u>74.89 ± 0.11</u>	50.00 ± 0.00	69.83 ± 0.00	50.07 ± 0.00	65.83 ± 0.00	58.07 ± 0.08	**96.30 ± 0.01**
Nomao	80.46 ± 0.05	56.09 ± 0.19	86.50 ± 0.00	54.37 ± 0.18	69.46 ± 0.04	<u>89.87 ± 0.03</u>	**95.98 ± 0.00**
Mushroom	55.40 ± 0.02	51.38 ± 0.05	<u>78.36 ± 0.01</u>	50.29 ± 0.04	57.62 ± 0.02	54.98 ± 0.01	**95.95 ± 0.00**

the features as base features. For each base feature, we generate overlapping features by adding different sizes of random noise to sample. Both the type and the size are completely random. This setting reflects the real-world scenarios, where it is generally impossible to know whether it contains unknown noise for a given dataset. Our noise generation refers to the noise settings in [12, 22], by randomly adding noises to base features with 20%-$U(-\alpha, \alpha)$, 40%- $U(-0.1 * \alpha, 0.1 * \alpha)$, 20%-$N(0, \alpha)$, and the remaining 20% sample untreated, where $\alpha \in [1, 2, 3, 4]$, U is a uniform distribution and N is a normal distribution. These processed features are used to replace base features. After processing, we randomly split the dataset vertically into four parts for VFL setting.

Baselines. We compare VF-FD with six baselines by training classifiers and evaluate the test accuracy of the classifier. Each baseline adopt a different strategy for feature deduplication. We describe these baselines as follows.

- **All Features.** Baseline based on dataset without any deduplication.
- **Random.** Randomly select one feature from each client.
- **MI.** Each client select top- k features correlated to label with higher Mutual Information (MI) scores [26].
- **Local Lasso.** Each client run lasso model locally for feature deduplication.
- **Global Lasso.** Clients collaborately train a global lasso model in [20].
- **FEAST.** State-of-art vertical federated feature deduplication framework [11] based on conditional mutual information (CMI) scores [32].

Evaluation models. To evaluate the performance of feature deduplication, we utilize logistic regression model for the sonar dataset with fewer samples and a 3-layer dense neural network for other datasets.

Environment. We implement our framework in Pytorch, All the experiments are conducted on Ubuntu 20.04 device equipped with 12-core i7 intel CPU, 32G of RAM and 1 NVIDIA GeForce RTX 4090 GPU. Our code is available at https://github.com/lizzy-0323/VF-FD.

5.2 Main Results

Accuracy. The classifier's classification accuracy with features is reported on Table 2. According to the result, we can make four key observations as follows:

VF-FD achieves the highest test accuracy in all five datasets over six baselines. Taking Mushroom dataset as an example, the accuracy of VF-FD is 39.50%, 43.57%, 16.69%, 44.66%, 37.33%, 39.97% higher than the six baselines, which demonstrates the superiority of our framework. Through the process of effective feature deduplication, we not only reduce the existing redundant features but also form the features at the sample level by selecting feature components in the overlapping feature groups, which is more beneficial for model training.

VF-FD outperforms Lasso-based baselines significantly. The reason is that Lasso regression only uses the coefficients of the correlation matrix to represent the relevance of features to the label, without considering the correlation between features, resulting in repeat selections and a large number of redundant features. and VF-FD uses an overlapping feature identification step to group overlapping features. Since the result of Lasso-based baselines is highly related to the threshold of correlation coefficient, we sample several common settings from [28] library, and adopt the optimal settings for each dataset to ensure fairness.

VF-FD is comparable to FEAST. Although FEAST is also a specifically designed framework for feature deduplication in VFL, it only addresses feature deduplication at the feature level by directly eliminating overlapping features when they are identified. However, VF-FD employs feature deduplication at the sample level, resulting in features of high quality after reconstruction from overlapping features groups. Although FEAST also performs well on two datasets, when the data size is small (Sonar) or differences are not significant at the feature level (Phishing, Mushroom, Android), directly removing features and selecting at the feature level significantly impacts the quality of features and leads to poor performance. In contrast, VF-FD employs a unified deduplication strategy, the number of features ultimately obtained is regulated only by a hyper parameter τ, which can minimize the impact of the dataset itself, during the reconstruction step, the most beneficial features for the model are selected through all the multiple feature components, which can effectively decrease the influence of noise at the sample level. Furthermore, VF-FD employs EMD rather than CMI to measure overlapping features, which places more emphasis on the distribution difference between features, preventing features from being misidentified.

Random performs worst in most cases. This is because it adopt a random strategy; in other words, it has no consideration between features and labels, or between features themselves from clients.

Efficiency. In this section, we evaluate the efficiency of VF-FD by computing the end-to-end communication cost during deduplication and transmission. We use the same dataset setup beforehand, calculating the communication overhead for FEAST, All Features, and VF-FD within the Nomao and Sonar datasets across 100 samples. As the communication cost is controlled by iteration round in FEAST, for fairness concern, we set the iteration round equals three to ensure can high accuracy and low communication cost. We report our result on Fig 3.

From the results, we can see that the communication cost of VF-FD is lower than the other baselines on both two datasets. And especially 30% lower than

Table 3: Accuracy comparison with different strategies for grouping. (mean std). The best and runner-up are marked with bold and underlined, respectively. 5-5 random denotes five overlapping groups with each group consists of five features.

Datasets	5-5 Random	10-5 Random	5-10 Random	Cosine similarity	VF-FD
Sonar	71.43 ± 0.07	70.06 ± 0.06	67.14 ± 0.03	$\underline{74.76 \pm 0.04}$	$\mathbf{77.14 \pm 0.05}$
PhysionNet	80.05 ± 0.01	80.48 ± 0.00	80.10 ± 0.01	$\underline{80.69 \pm 0.01}$	$\mathbf{86.44 \pm 0.01}$
Android	96.23 ± 0.02	$\underline{96.62 \pm 0.00}$	92.93 ± 0.01	79.16 ± 0.03	$\mathbf{96.83 \pm 0.00}$
Nomao	89.98 ± 0.00	90.45 ± 0.00	90.53 ± 0.00	$\underline{94.10 \pm 0.00}$	$\mathbf{96.03 \pm 0.00}$

Table 4: Accuracy comparison with different strategies for deduplication. (mean std). The best and runner-up are marked with bold and underlined, respectively.

Datasets	Random Feature Selection	Random Feature Reconstruction	VF-FD
Sonar	70.95 ± 0.09	$\underline{75.71 \pm 0.06}$	$\mathbf{76.67 \pm 0.08}$
PhysionNet	82.42 ± 0.02	$\underline{82.71 \pm 0.01}$	$\mathbf{86.97 \pm 0.01}$
Android	58.13 ± 0.02	$\underline{90.84 \pm 0.00}$	$\mathbf{95.84 \pm 0.01}$
Nomao	79.05 ± 0.11	$\underline{90.35 \pm 0.00}$	$\mathbf{95.46 \pm 0.00}$

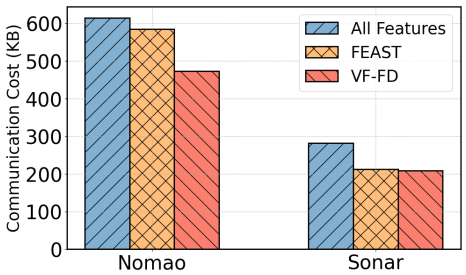


Fig. 3: Comparison of our VF-FD, All features, FEAST in terms of communication cost for one hundred samples.

All features without deduplication. These results clearly demonstrate VF-FD can effectively reduce the amount of data transmitted to the server and decrease the communication cost, thereby improving efficiency in model training.

5.3 Ablation Study and Parameters

In this section, we conduct two ablation studies on Sonar, PhysionNet, Android, Nomao dataset to validate the effectiveness of overlapping feature identification and feature reconstruction step.

Overlapping feature identification. Our first step in the VF-FD framework is the overlapping feature identification method, which utilizes EMD as metric for similarity calculation. To evaluate the effectiveness of the step, in this section,

we compare our overlapping feature identification strategy with random grouping strategy and cosine similarity grouping strategy. For random strategies, a m-n Random grouping strategy means that the number of groups is n, and the number of features in each group is m, during iterations between clients, features from the previous client are selected and placed into group randomly. After the iterations are completed, an group list of (m, n) will be obtained. For Cosine Similarity group strategy, we employ cosine distance as metric. In this experiment setup, we use 5-1, 5-5, and 5-10 random grouping and as random baselines, and set cosine distance threshold equal to 0.1. For dataset setup, we add 40% overlapping features to base features with noise setup in former section, and the following step of VF-FD remains the same. The results are reported on Table 3.

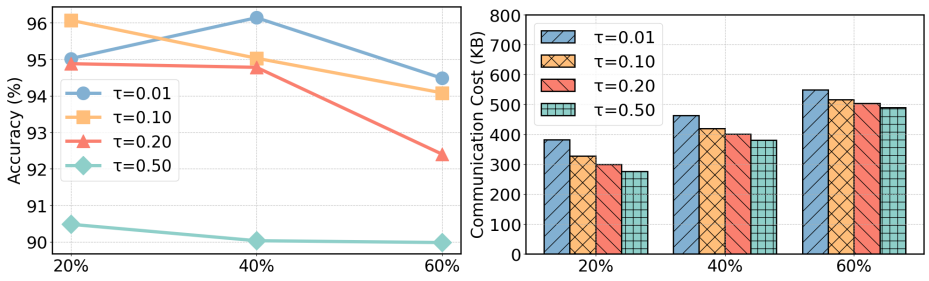
We can see that our VF-FD framework outperforms the other four grouping strategies over all datasets, which shows the superiority of our overlapping feature identification method. In the last three datasets, our method has achieved an average accuracy improvement of 2.4% over the runner-up grouping strategy. The reason is that our overlapping feature identification method generate overlapping groups by the features itself, and EMD can measure the distribution differences between processed features, with cosine distance only considers the directional differences. The random strategies, in contrast, lacks any consideration of the relationships between features.

Feature reconstruction. After obtaining overlapping feature groups, we compare our LSS metric designed for feature reconstruction with other two deduplication strategies over three datasets. 1) Random select features in each overlapping feature group. 2) Random reconstruction features in each overlapping feature group. To demonstrate the superiority of our method, we adopt the noise setup beforehand but with $\alpha = [0.1, 0.2, 0.3, 0.4]$, which make noise difference among samples is not significant. For evaluation, We use the accuracy of classifier. The results are reported on Table 4.

From the results, we can observe that VF-FD with sample-wise strategy outperforms feature-wise strategies at an average 9.95% higher over the three datasets. This results show that sample-wise strategy for feature brings more combinations for better performance. And VF-FD also outperforms random feature reconstruction strategy over the three datasets, which shows that the LSS can measure the sensitivity to the loss function of feature components in each sample with relatively simple MLP. By employing LSS to reconstruct feature components, features that are more conducive to enhance the model's performance can be obtained.

Similarity threshold τ for EMD. In this experiment, we analyze the impacts of different similarity threshold τ of EMD with different overlapping rate in dataset setup. We set client number $M = 4$ by default, vary overlapping rate from $[20\%, 40\%, 60\%]$, and τ from $[0.01, 0.10, 0.20, 0.50]$. We analyze the influence of τ from two aspects, efficiency and accuracy. The results are shown on Fig 4.

In terms of accuracy, we can see from Fig 4(a) that accuracies are stable at an average of 90% under each τ and overlapping rate. This is because our framework overlapping feature identification method decreases the number of



(a) influence on accuracy (b) influence on efficiency

Fig. 4: The influence of different similarity threshold τ and overlapping rate in dataset on accuracy (a) and efficiency (b).

feature essentially without considering the dataset size, and use LSS for feature reconstruction to obtain high-quality features from overlapping groups. Thus, the way we deduplicate the features has little impact on the classifiers' accuracy.

Regarding the efficiency, we can see from Fig 4(b) that with the τ increases, the communication cost decreases. The reason is that by increasing the τ, there are more features in the same overlapping group. As a result, the number of features transmitted to the server decreases. In addition, we also notice that as τ increases, the accuracy is also decrease. This is expected and imply to make a trade-off for better performance in practical model training scenarios.

6 Privacy Analysis

In this work, we assume clients and server are honest-but-curious model, a widely recognized threat model in federated learning research [3, 14]. During the overlapping identification process, each client sends the processed features to the next client. It is clear that the features after binning and normalization can not be used to restore the original features. And the features are also used in feature reconstruction process. This ensures neither the server nor the client can restore original features from the entire process.

However, a curious server or client might still deduce the approximate range of the original data from the preprocessed features. For stricter privacy requirements, the process can be encrypted using Homomorphic Encryption [1].

7 Conclusion

In this paper, we study feature deduplication in VFL settings. We present an efficient framework called VF-FD. First, We design an federated overlapping feature identification mechanism utilizing EMD while protecting privacy simultaneously. Then, we train a model and combine weights and gradients for selecting feature components and reconstruct features in overlapping feature groups, which we called LSS. Extensive experiments show that VF-FD outperforms

state-of-the-art baselines by up to 22.4% in terms of accuracy, which demonstrates the superiority of our VF-FD over other baseline methods.

8 Acknowledgments

This work was sponsored by National Natural Science Foundation of China (62472327, 62272353) and Key R&D Program of Hubei Province (2023BAB077).

References

1. Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes: theory and implementation. ACM Computing Surveys **51**(4), 1–35 (2018)
2. Applegate, D., Dasu, T., Krishnan, S., Urbanek, S.: Unsupervised clustering of multidimensional distributions using earth mover distance. In: SIGKDD. pp. 636–644 (2011)
3. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In: SIGSAC. pp. 1175–1191 (2017)
4. Castiglia, T., Zhou, Y., Wang, S., Kadhe, S., Baracaldo, N., Patterson, S.: Less-vfl: communication-efficient feature selection for vertical federated learning. In: ICML (2023)
5. Chen, J., Stern, M., Wainwright, M.J., Jordan, M.I.: Kernel feature selection via conditional covariance minimization. NeurIPS **30** (2017)
6. Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Papadopoulos, D., Yang, Q.: Secureboost: a lossless federated learning framework. IEEE Intelligent Systems **36**(6), 87–98 (2021)
7. Cohen, I., Huang, Y., Chen, J., Benesty, J.: Pearson correlation coefficient. Noise reduction in speech processing pp. 1–4 (2009)
8. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found. Trends Comput. Graph. Vis. **7**, 81–227 (2012)
9. Doquet, G., Sebag, M.: Agnostic feature selection. In: Machine Learning and Knowledge Discovery in Databases: European Conference. pp. 343–358. Springer (2020)
10. El Aboudi, N., Benhlima, L.: Review on wrapper feature selection approaches. In: International Conference on Engineering and MIS. pp. 1–5 (2016)
11. Fu, R., Wu, Y., Xu, Q., Zhang, M.: Feast: a communication-efficient federated feature selection framework for relational data. Proceedings of the ACM SIGMOD International Conference on Management of Data **1**(1) (2023)
12. Guo, Y., Wang, W., Wang, X.: A robust linear regression feature selection method for data sets with unknown noise. IEEE TKDE **35**(1), 31–44 (2023)
13. Hassanieh, W., Chehade, A.: Selective deep autoencoder for unsupervised feature selection. In: AAAI. vol. 38, pp. 12322–12330 (2024)
14. Le, J., Zhang, D., Lei, X., Jiao, L., Zeng, K., Liao, X.: Privacy-preserving federated learning with malicious clients and honest-but-curious servers. IEEE Transactions on Information Forensics and Security (2023)
15. Li, A., Peng, H., Zhang, L., Huang, J., Guo, Q.W., Yu, H., Liu, Y.: Fedsdg-fs: efficient and secure feature selection for vertical federated learning. IEEE INFOCOM pp. 1–10 (2023)

16. Li, P., Pei, Y., Li, J.: A comprehensive survey on design and application of autoencoder in deep learning. Applied Soft Computing **138**, 110176 (2023)
17. Li, X., Dowsley, R., De Cock, M.: Privacy-preserving feature selection with secure multiparty computation. In: ICML. pp. 6326–6336. PMLR (2021)
18. Ling, Y., Zhong, Z., Luo, Z., Yang, F., Cao, D., Lin, Y., Li, S., Sebe, N.: Cross-modality earth mover's distance for visible thermal person re-identification. In: AAAI. vol. 37, pp. 1631–1639 (2023)
19. Liu, B., Lv, N., Guo, Y., Li, Y.: Recent advances on federated learning: A systematic survey. Neurocomputing p. 128019 (2024)
20. Liu, Y., Fan, T., Chen, T., Xu, Q., Yang, Q.: Fate: An industrial grade platform for collaborative learning with data protection. Journal of Machine Learning Research **22**(226), 1–6 (2021)
21. Liu, Y., Kang, Y., Zou, T., Pu, Y., He, Y., Ye, X., Ouyang, Y., Zhang, Y.Q., Yang, Q.: Vertical federated learning: Concepts, advances, and challenges. IEEE TKDE (2024)
22. Meng, D., De la Torre, F.: Robust matrix factorization with unknown noise. In: IEEE ICCV. pp. 1337–1344 (2013)
23. Meng, Y., Dai, X., Yan, X., Cheng, J., Liu, W., Guo, J., Liao, B., Chen, G.: Pmd: an optimal transportation-based user distance for recommender systems. In: ECIR. pp. 272–280. Springer (2020)
24. Nikolic, M., Zhang, H., Kara, A., Olteanu, D.: F-ivm: learning over fast-evolving relational data. In: SIGMOD. pp. 2773–2776 (2020)
25. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE TPAMI **27**(8), 1226–1238 (2005)
26. Ross, B.C.: Mutual information between discrete and continuous data sets. PloS one **9**(2) (2014)
27. Rubner, Y., Tomasi, C., Guibas, L.: The earth mover's distance as a metric for image retrieval. IJCV **40**, 99–121 (2000)
28. Sklearn: (2024), https://scikit-learn.org
29. Wang, G., Gu, B., Zhang, Q., Li, X., Wang, B., Ling, C.X.: A unified solution for privacy and communication efficiency in vertical federated learning. NeurIPS **36** (2024)
30. Wu, X., Cheng, Q.S.: Fractal autoencoders for feature selection. AAAI **2021**, 10370–10378 (2020)
31. Xu, J., Yu, M., Shao, L., Zuo, W., Meng, D., Zhang, L., Zhang, D.: Scaled simplex representation for subspace clustering. IEEE Transactions on Cybernetics **51**, 1493–1505 (2021)
32. Yang, H., Moody, J.: Feature selection based on joint mutual information. In: Proceedings of International ICSC symposium on advances in intelligent data analysis. vol. 23. Citeseer (1999)
33. Zhang, C., Cai, Y., Lin, G., Shen, C.: Deepemd: differentiable earth mover's distance for few-shot learning. IEEE TPAMI **45**(5), 5632–5648 (2022)
34. Zhang, Q., Yan, X., Ding, Y., Xu, Q., Hu, C., Zhou, X., Jiang, J.: Treecss: An efficient framework fornbsp;vertical federated learning. In: Database Systems for Advanced Applications. p. 425–441. Springer (2024)
35. Zhou, X., Yan, X., Li, X., Huang, H., Xu, Q., Zhang, Q., Jerome, Y., Cai, Z., Jiang, J.: VFDV-IM: an efficient and securely vertical federated data valuation. In: Database Systems for Advanced Applications. vol. 14850, pp. 409–424. Springer (2024)