Vertical federated participant sample generation based on Semi-supervised learning.

# Introduction

# Related Work

Vertical Federated Learning (VFL) has gained significant attention as a privacy-preserving framework that enables collaborative learning between parties holding different features of the same dataset without sharing raw data. In VFL, it is essential to align the samples from different parties, as they may hold complementary features. One of the key challenges in VFL is the limited number of aligned samples, which may affect the model’s performance when used for training.

A considerable body of work has focused on improving the efficiency and accuracy of VFL models by addressing the issue of missing aligned samples. Existing approaches largely rely on using only aligned data, which limits the potential of the model, especially when the number of aligned samples is small. Methods like Federated Learning with Missing Data [1] have proposed handling missing data through imputation techniques. However, these methods do not fully exploit the unlabeled data available across the parties, which could lead to significant performance gains.

Semi-supervised learning (SSL) has been employed in VFL to enhance model performance by making use of both labeled and unlabeled data. In Vertical Federated Semi-Supervised Learning [2], the authors proposed a model that enables learning from unlabeled data while maintaining data privacy. This approach, while promising, still faces the challenge of how to integrate the unlabeled data from multiple parties in a way that respects privacy and maximizes the utilization of all available data.

In the realm of missing feature imputation, Data Completion Techniques in Federated Learning [3] have explored various imputation methods, such as using generative models to predict missing features in federated settings. These methods generally focus on imputing the missing values within the aligned dataset but fail to incorporate the non-aligned data in a meaningful way.

In contrast, our proposed method, VFPU-M-Syn, introduces a novel approach that combines vertical federated semi-supervised learning with synthetic tabular data generation to address the challenge of missing or non-aligned samples. By treating non-aligned samples as unlabeled data and leveraging semi-supervised learning techniques, VFPU-M-Syn is able to enhance the model’s generalization ability and improve data utilization. Moreover, the introduction of tabular data generation technology allows for the more informed filling of missing values in the feature set, which optimizes the data completion process in a vertical federated setting.

In summary, while existing works have made strides in handling missing data and unlabeled samples in federated learning, our approach is unique in its use of both semi-supervised learning and data generation techniques to improve the performance of VFL models, particularly when dealing with limited aligned samples. The results from our proposed method demonstrate significant improvements in terms of both accuracy and generalization capability.

# Proposed Method

## Problem Setup

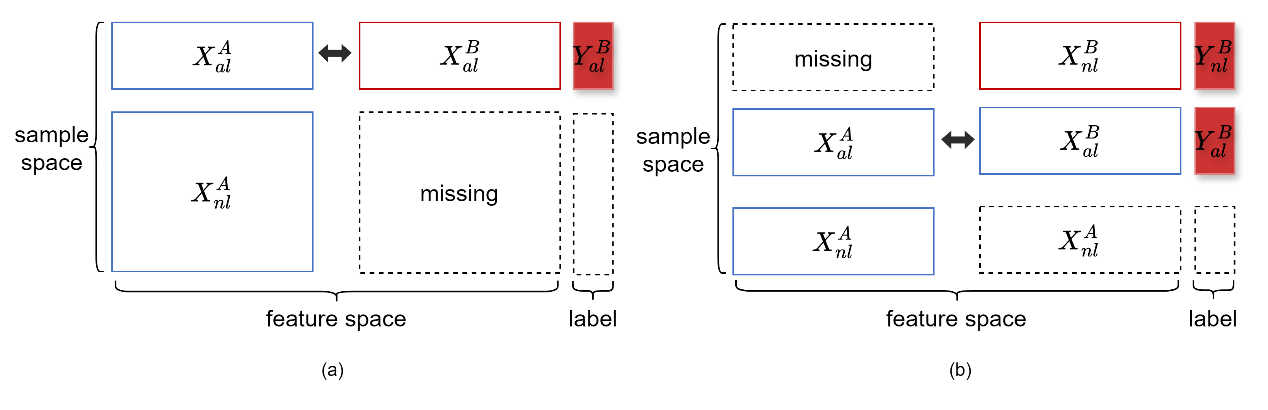


Fig.1 Vertically federated missing aligned samples and missing samples.

Consider a scenario of two - party Vertical Federated Learning (VFL), which is a typical VFL setting defined in Reference [40]. The participating parties include Party A and Party B, and only one of them has the labels.

First, Party A has a dataset:

where X^A\_i is the feature vector of the i-th sample, and n^A is the number of samples. Party A's data only contains features and no labels.

Next, Party B has a dataset:

where is the feature vector of the -th sample, is the corresponding one - hot encoded true label, C represents the number of classes, and is the number of samples. Party B has the labels, which is crucial in VFL because labels are usually used for supervised learning tasks. However, Party B lacks sufficient features to build an accurate model alone, so it needs to utilize the supplementary features provided by Party A.

It should be emphasized that and are privately stored by Party A and Party B respectively, and neither party can expose their datasets to the other.

In VFL, the datasets and of Party A and Party B contain features of different samples. To perform joint learning, the samples with the same identity need to be aligned. Assume that through the privacy - preserving encrypted entity matching technology [30], the two

parties have completed sample alignment, and the aligned sample set is obtained:

where n\_{al} is the number of aligned samples. Party A has the features of the aligned samples:

Party B has the features and labels of the aligned samples:

If we concatenate and and align the samples with the same identity, we will get a single dataset as shown in Figure 1.b. This dataset is vertically partitioned, and each party has a vertical partition (or partial view) of this dataset, which is exactly the origin of the term "Vertical Federated Learning". However, there are usually only a limited number of aligned samples between the two parties.

In addition to the aligned samples, each party also has some non - aligned samples, that is, data without corresponding samples from the other party. For Party A, the non - aligned samples are represented as:

For Party B, the non - aligned samples are represented as:

From the perspective of a single tabular dataset (see Figure 1.b), each party has no corresponding features (or labels) for the non - aligned samples of the other party. We regard these features (or labels) as "missing".

Traditional VFL methods only use the aligned samples to build a federated machine - learning model, while discarding the non - aligned samples and . This approach may limit the performance of the model when the number of aligned samples is small, because a large amount of potentially useful data is ignored.

This paper proposes a new method called VFPU - M - Syn, aiming to make full use of the non - aligned samples and to improve the performance of the Vertical Federated Learning (VFL) model. This method combines vertical federated semi - supervised learning and tabular data generation technology. By regarding the aligned samples \mathcal{D}\_{al} as labeled data (where the "labels" of can be regarded as the feature values of X^B\_{al}) and the non - aligned samples as unlabeled data, it uses semi - supervised learning to learn from the aligned samples to enhance the generalization ability of the model. At the same time, it uses tabular data generation technology to fill in the missing values of features that are less relevant to Party A and combines it with vertical federated learning to optimize data completion. Compared with traditional VFL methods, VFPU - M - Syn not only uses the aligned samples but also makes full use of the non - aligned samples and , significantly improving data utilization. Through vertical federated semi - supervised learning, the model can extract useful information from unlabeled data to further improve the generalization ability. The introduction of tabular data generation technology makes the filling of missing features more reasonable, thus optimizing the data filling strategy and improving the overall performance of the model. In summary, VFPU - M - Syn introduces innovative technologies on the basis of the traditional VFL framework, makes full use of non - aligned samples, and significantly improves the accuracy and generalization ability of the VFL model when the number of aligned samples is limited. The experimental results also verify its superiority.

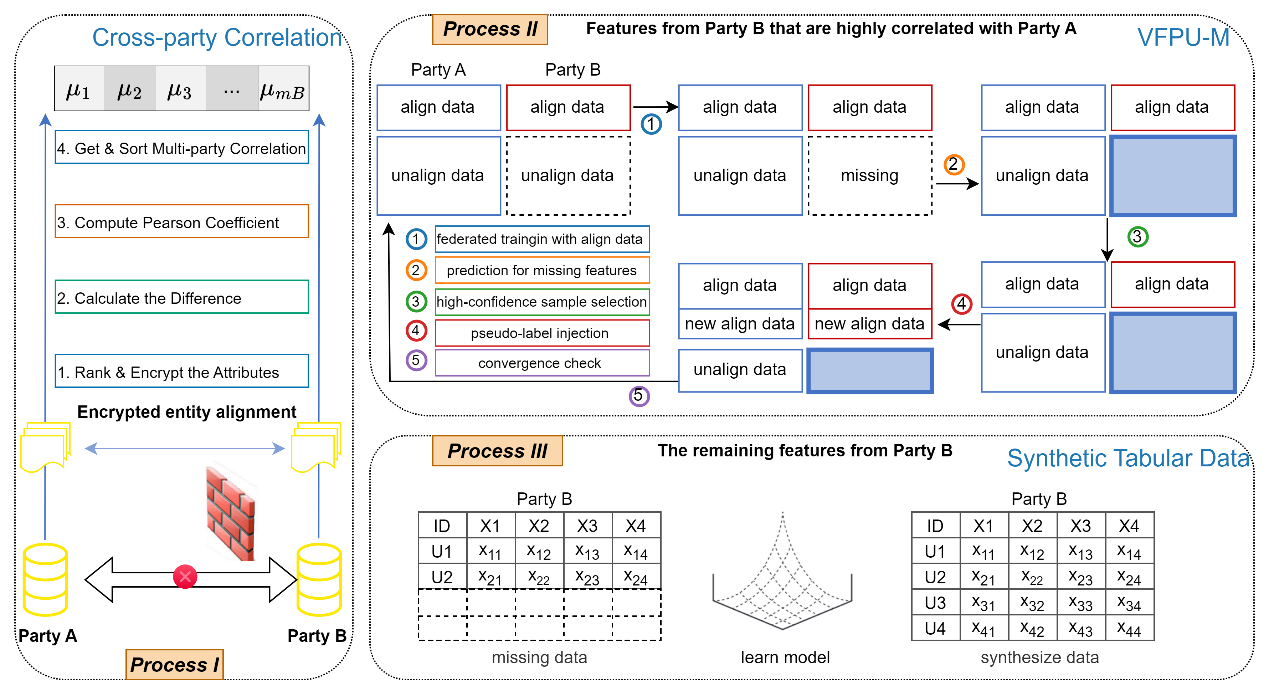


Fig.2 The Proposed Method

## Process I: Computing Cross - Party Feature Correlation

In the Vertical Federated Learning (VFL) framework, different parties possess heterogeneous data with the same samples but different features. To effectively utilize the feature information of the aligned samples, it is necessary to quantify the statistical associations between cross - party features. This section proposes a privacy - preserving Spearman rank correlation analysis method to construct a cross - party feature correlation ranking system.

Let the coordinator act as a trusted third - party, responsible for generating a homomorphic encryption (HE) key pair , where:

- is the public key, used for encrypting data;

- is the secret key, used for decrypting data.

The coordinator distributes the public key to Party A and Party B so that they can perform encrypted computations on the data without directly exposing the original data. The specific computation process is as follows:

### Step 1: Feature Column Rank Vector

Let the feature space of Party A be , where represents the feature dimension of Party A, i.e., Party A has m\_A features. \varphi^A\_p\in\mathbb{R}^{n\_{al}} represents the observation vector of the p - th feature of Party A on the aligned sample set , and represents the values of this feature on all aligned samples, where is the number of aligned samples.

Similarly, the feature space of Party B is , where represents the feature dimension of Party B, i.e., Party B has features. represents the observation vector of the q - th feature of Party B on the aligned sample set . represents the values of this feature on all aligned samples, where is the number of aligned samples.

For any feature column , calculate its rank vector:

where r^A\_{pi} represents the rank of sample i on feature \varphi^A\_p, i.e., the sorting position of this sample in this feature column. If there are identical values, the average rank is used. Similarly, the corresponding rank vector can be calculated for the feature column \varphi^B\_q of Party B:

### Step 1: Encrypted Rank Transmission

Party A uses the public key \text{pk} to homomorphically encrypt the rank vector R^A\_p and obtains:

and sends the encrypted rank vector to Party B. Similarly, Party B also homomorphically encrypts its own rank vector R^B\_q and obtains:

### Step 2: Rank Difference Calculation

For any feature pair , Party B calculates the encrypted rank difference vector:

where represents the rank difference of sample between Party A's feature and Party B's feature . Since homomorphic encryption supports addition operations, Party B can directly calculate the rank difference in the encrypted state without decryption. Party B sends the encrypted rank difference vector to the coordinator .

### Step 3: Spearman Correlation Calculation

The coordinator C decrypts [D\_{pq}] and obtains:

Then calculate the Spearman correlation coefficient:

\rho\_{pq} represents the Spearman correlation coefficient between Party A's feature f^A\_p and Party B's feature f^B\_q. Finally, construct the cross - party correlation matrix:

stores the Spearman correlation coefficient between the - th feature column of Party A and the - th feature column of Party B.

### Step 4: Feature Association Strength

For each feature f^B\_q of Party B, calculate its average association strength with all features of Party A:

represents the comprehensive dependence of Party B's feature on Party A's feature space.

### Step 5: Generate a Sorted List

Construct the feature importance sequence:

Sort it in descending order of \mu\_q to obtain the sorted feature list

This list guides the feature completion of Party B's aligned samples using federated semi - supervised learning. Dimensions with strong associations with Party A's features are preferentially supplemented using the federated semi - supervised learning method.

## Process II: Vertical federated semi-supervised prediction missing feature

In the context of vertical federated learning, the generation and processing of data is a complex and crucial process. Algorithm 1 and Algorithm 2 together form a complete semi - supervised learning framework, which aims to fully utilize unlabeled data to improve model performance while protecting data privacy.

The core objective of Algorithm 1 is to generate pseudo - labeled data through the vertical federated semi - supervised learning method to enhance the model's training set. This algorithm obtains data from two participating parties (Party A and Party B) and an external label data source (\mathcal{L}\_B). Specifically, Party A provides the labeled dataset X\_{al}^A and the unlabeled dataset X\_{nl}^A, while Party B provides the corresponding label dataset \mathcal{L}\_B. The algorithm sets a threshold \tau to filter out the pseudo - label pairs (\mu\_q, f^B\_q) with high confidence in \mathcal{L}\_B, and these data will be used to generate new training data.

In the initialization stage, the algorithm sets the generated dataset X^B to an empty set and filters out the eligible pseudo - label pairs from \mathcal{L}\_B. Subsequently, the algorithm calls the VFPU - M algorithm (i.e., Algorithm 2) to perform joint processing between the data of Party A and Party B to generate pseudo - labeled data with high confidence. In this way, the algorithm gradually adds the generated pseudo - labeled data to X^B, ultimately forming a complete pseudo - labeled dataset.

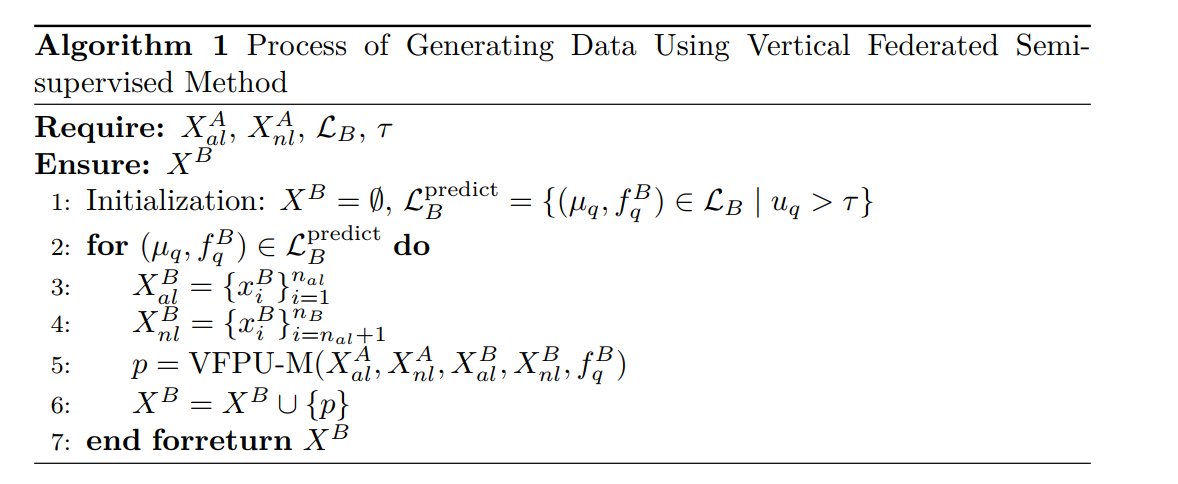
The VFPU - M algorithm (Vertical Federated Pseudo - Labeling with Uncertainty Management) is a core component for generating pseudo - labels and selecting high - confidence samples. This algorithm receives the datasets (X\_{al}^A, X\_{nl}^A, X\_{al}^B, X\_{nl}^B) from two participating parties and the relevant label information, and optimizes the model performance through a series of iterative processes.

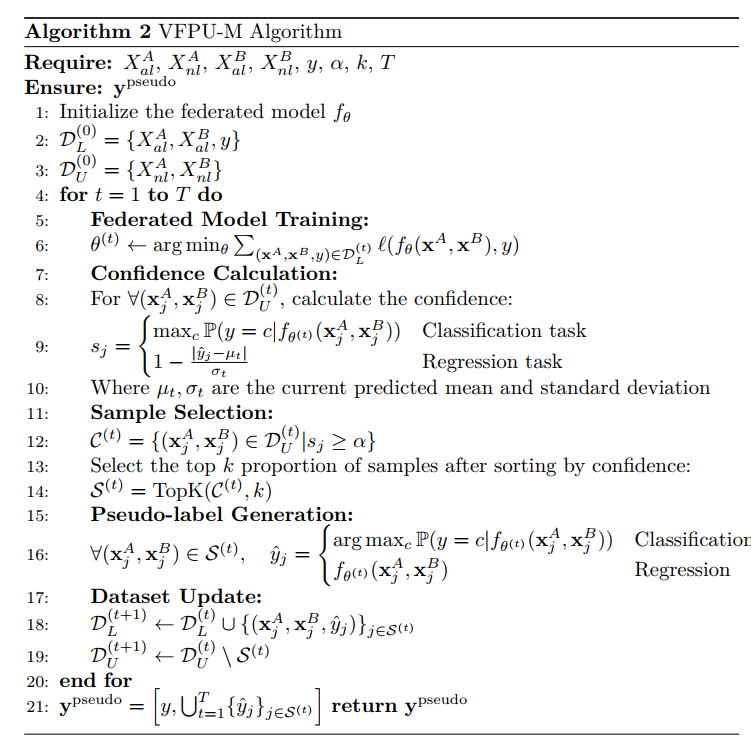
In each iteration, the VFPU - M algorithm first trains a global model f\_\theta through the federated learning framework. This model learns each sample (\mathbf{x}^A, \mathbf{x}^B, y) by minimizing the loss function, thereby optimizing the model parameters \theta. After the training is completed, the algorithm calculates the confidence of each unlabeled sample. For classification tasks, the confidence is determined by the maximum class probability of the sample predicted by the model; for regression tasks, it is determined by calculating the deviation between the predicted value and the actual value.

Next, the algorithm filters out the eligible samples according to the confidence and selects the top k proportion of samples with the highest confidence to generate pseudo - labels. For classification tasks, the pseudo - label is the class with the maximum probability; for regression tasks, it is the value predicted by the model. The generated pseudo - labeled data is added to the training set, and these samples are removed from the unlabeled dataset.

Through multiple rounds of training and sample selection, the VFPU - M algorithm continuously updates and expands the pseudo - labeled dataset, and finally returns a complete dataset \mathbf{y}^{\text{pseudo}} containing pseudo - labeled data. This process not only improves the generalization ability of the model but also fully exploits the potential of unlabeled data while protecting data privacy.

The above is a detailed description of Section 2.1, which describes the working principles of Algorithm 1 and Algorithm 2 and their applications in vertical federated semi - supervised learning.





## Process II: Synthetic Tabular Data

# Experiment

## Dataset

In this study, we utilize three widely recognized datasets to evaluate the performance of the proposed method. Each dataset presents unique challenges and is selected for its relevance to different domains of machine learning.

**Covertype Dataset：**The Covertype dataset, available from the UCI Machine Learning Repository, is used in the task of classifying forest cover types based on cartographic variables. This dataset consists of 581,012 instances, each described by 54 features. The goal is to predict the cover type (one of seven possible categories) based on attributes such as elevation, aspect, slope, and soil type. It is a typical classification problem and is chosen for its large scale and high dimensionality, making it suitable for testing the scalability and efficiency of machine learning algorithms in handling large and complex datasets.

**INTRUSION Dataset：**The INTRUSION dataset, sourced from Kaggle, is a well-known dataset in the domain of network intrusion detection. It contains 125,973 instances and 41 features, including both continuous and categorical variables. The task is to classify network traffic into either normal activity or one of several types of attacks. The dataset provides a challenging environment for model evaluation, as it involves both class imbalance and high-dimensional features, making it ideal for exploring methods that can handle such complexities effectively.

**Loan Dataset：**The Loan dataset, accessible through Kaggle, focuses on predicting loan approval decisions based on various customer attributes. It contains 5000 records with 13 features, such as income, loan amount, and credit score, and aims to classify whether a loan is approved or not. This dataset is a typical binary classification problem that allows for the investigation of algorithms that handle tabular data and categorical variables, providing insight into financial decision-making processes.

## Experiment Setup

大论文第四章先写起，

实验分析:

1. 方法内部的：不同的基分类器，不同生成模型、不同和置信度、相关性阈值，不同缺失率

2、

首先，基分类器、生成模型

## 实验分析1：比较不同基分类器的结果

## 实验分析2：比较不同生成模型的结果

比较不同的生成模型，有几个层次：

1. 相关性高的生成
2. 相关性系数，在多少程度上生成

超参数：相关性系数，置信度、k

## 实验分析3：基线对比实验

全部生成

# Conclusion