Privacy Attack Identification and Protection Strategy Analysis Based on Vertical Federation Clustering

Mingshan Fan[1,2] and Huijuan Guo[3,*]

[1]*College of Information Technology, Shanxi Finance and Taxation College, Taiyuan 030024, China*
[2]*College of Finance and Economics, Taiyuan University of Technology, Taiyuan 030024, China*
[3]*College of Information and Computer, Taiyuan University of Technology, Taiyuan 030600, China*
E-mail: hjljiaoxue@163.com
[*]*Corresponding Author*

Received 10 March 2025; Accepted 17 April 2025

Abstract

Although federated learning provides strong privacy guarantees when handling cross-device or cross-data center learning tasks, it still faces numerous challenges and potential security threats when applying it to real-world scenarios. This paper proposes a privacy attack identification and protection strategy based on vertical federation clustering, so as to improve privacy protection and data processing security in vertical federation clustering. By fusing parameters, it reduces multi-dimensional data to one-dimensional vector, thus reducing the amount of random disturbance in the subsequent random response process. Moreover, this paper proposes a method of independently setting the answer set for each parameter, which improves the probability of outputting the true value in the random response mechanism. In addition, it improves data utility and clustering precision while ensuring

Journal of Cyber Security and Mobility, Vol. 14_2, 475–504.
doi: 10.13052/jcsm2245-1439.1429

randomness. The comprehensive performance of the model proposed in this paper is excellent in the experiment. In particular, its privacy protection effect reaches 89.34% and 95.14% under ARP and Botnet attacks, respectively. At the same time, the identification rate and recall rate are generally high, showing good privacy protection ability and model robustness. Therefore, the model proposed in this paper improves the privacy protection degree of clustering algorithm in the face of various privacy attacks including data reconstruction attacks under federated learning architecture.

Keywords: Vertical federal clustering, privacy, attack identification, protection strategy.

1 Introduction

Federated learning is a cutting-edge machine learning technology that allows collaborative training of models among multiple data sources or clients without sharing data directly, effectively solving the data privacy and security issues existing in traditional centralized learning. This method is particularly suitable for dealing with distributed, heterogeneous and sensitive data sets, such as medical health records, financial transaction data, etc. This technology ensures that user data remains unshared locally, effectively dealing with privacy protection and data silos. By decentralizing model training to the client where the data is located, federated learning technology not only reduces the risk of privacy leakage caused by data transmission, but also optimizes data processing efficiency and realizes cross-domain data utilization without infringing personal privacy. In addition, different from traditional centralized machine learning methods, federated learning emphasizes training the model at the local client, and only needs to send the trained model update or gradient information to the central server for aggregation, thereby updating the global model. Because the original data never leaves the local device, this method effectively guarantees the privacy of the data. Furthermore, since federated learning is able to handle the heterogeneity of devices and data, it provides greater flexibility and adaptability for deploying machine learning models on different environments and devices [1].

However, although federated learning provides strong privacy guarantees when handling cross-device or cross-data center learning tasks, it still faces numerous challenges and potential security threats when applying it to real-world scenarios. First of all, although the model update can protect the data privacy of participants to a certain extent by aggregating local model

parameters, this process may still be exploited by malicious attackers. The attacker uses member inference attack technology [2] to infer the private data information of individual users from the aggregated model update. This security vulnerability needs to be solved by further improving encryption and privacy protection technology. Moreover, the existence of malicious clients also poses a severe challenge to federated learning systems. In poisoning attacks [3], malicious participants may affect or destroy the training process of the entire system by tampering with their own models to update data. Furthermore, poisoning attacks will cause the performance of the global model to degrade, or even completely deviate from the original training goal. Regarding the existence of malicious servers, some researchers have recently proposed preference attacks in federated learning [4]. Different from other types of inference attacks, the attacker analyzes the user's preference categories, such as products and common expressions, according to the gradient changes of the local user model, which seriously damages the privacy of the user's personal sensitive data. In order to deal with these security threats, researchers have proposed a variety of defense strategies. For privacy enhancement technologies, differential privacy (DP) technology, secure multi-party computing and homomorphic encryption technology are commonly used at present. Although these technologies have great advantages in privacy enhancement, DP disturbance will affect the precision of model query results and reduce the prediction performance of the model. In addition, homomorphic encryption and secure multi-party computing technology using cryptographic technology also face the problems of high computational complexity, high communication and computational overhead, which affect the performance of the system [5].

There is a risk of privacy leakage in existing research models, and there are communication and computational efficiency bottlenecks in the model construction process. Data heterogeneity can also affect the quality of the model, resulting in federated learning being unable to exert its privacy protection effect.

To solve this problem, this paper designs an attack method for the risk of privacy leakage in the process of parameter transfer, and introduces the random response mechanism that satisfies DP into the process of parameter transfer of vertical federation clustering. Moreover, this paper uses a variety of attack methods, including the attack methods proposed in this paper, to provide defense and improve the privacy protection of vertical federated clustering. The main work of this paper is to extend the clustering algorithm to the vertical federated learning architecture, and design an attack method

aiming at the privacy leakage risk. From the perspective of semi-trusted server, by analyzing the distance parameters passed in the clustering process, the data reconstruction attack on the participants' data sets is completed. Finally, this paper designs a privacy protection method based on DP to defend against various attacks including data reconstruction attacks proposed in this paper, thus solving the problem of insufficient privacy protection of vertical federated clustering algorithm.

2 Related Works

(1) Member inference attack

Member inference attack directly relates to important issues of data privacy and model transparency. When the model is overfitted, the overfitted model performs well on the training data, but its generalization ability to unseen new data is weak. At this point, an attacker can exploit this to infer whether specific data is being used for training by analyzing the model's response to that data. This type of attack is particularly destructive for applications involving sensitive information. Reference [6] studied the situation in online learning applications, where the ML image classifier was updated by new data samples, while the adversary can reconstruct the updated samples by using the information of the two versions before and after the update of the target ML model. Reference [7] showed similar results in natural language models and data deletion scenarios. These studies quantify information leakage during model updates.

For member inference attacks in data forgetting scenarios under federated learning, existing research mainly proposes three technical methods for data forgetting [8]: retraining method, approximate forgetting method and incremental update method. The retraining method is the most intuitive data forgetting method, which removes the deleted data from the training set and then retrains the model. However, while this approach is theoretically the most thorough, it is costly, especially if the data sets and models are extremely large. For the approximate forgetting method, in order to reduce the computational cost, researchers have proposed a variety of approximate forgetting techniques. These techniques do not need to retrain the model from scratch, but locally adjust the parameters of the model to reduce the influence of forgotten data. The incremental update method updates only the affected parts of the model instead of the entire model. This can be achieved by using an incremental learning algorithm, and it only modifies

the parameters affected by the data deletion. The attack model proposed in reference [9] adopts the SISA strategy with incremental update, aiming to provide a flexible and efficient data processing and forgetting mechanism for the machine learning field

(2) Defense efforts against vertical federated learning inference attacks

Because the concept of label inference attack in vertical federated learning has not been proposed for a long time, there is little research on the defense of label inference attack in vertical federated learning, especially to weigh the precision and defense ability of original federated learning. Reference [10] used several previously commonly used privacy protection methods in the paper, but the results were unsatisfactory. Reference [11] proposed a privacy protection combination method for machine learning. This defense framework includes DP, gradient compression and random selection of gradients. This framework is used to defend against label inference attacks, but according to the experimental results, it will interfere with the training of the original federated learning model. Reference [12] proposed the SignSGD method, which ensures the security of federated learning by reducing the communication of participants and reducing the shared gradient in horizontal federated learning. W When the neural network propagates forward/backward, the sign of the shared gradient can be varied. For example, the negative sign becomes a positive sign, the positive sign becomes a negative sign, etc. By using this method to modify the gradient, the safety is guaranteed. The experimental results also show the problem that the precision of attack and original federated learning cannot be weighed. In reference [13], Laplace noise was added to the gradient to interfere with the opponent's snooping on the gradient. The experimental results are slightly better than the first two, but the ideal effect has not been achieved. Reference [14] put forward the idea of gradient compression. When federated learning performs intermediate gradient transmission, only part of the gradient, such as 10%, is shared, which is feasible for horizontal federated learning. For 10% gradient, it does not affect the results of horizontal federated learning training, and it can still guarantee good precision and privacy security. In the experiment of longitudinal federated learning, it performs well in most data sets. However, for certain data sets, as the compression rates increase, the precision of the model tends to change. The larger the compression rate, the lower the precision of the model. In recent defense work, reference [15] proposed a new defense method, which uses obfuscation autoencoder to defense. This method is based on the regularization of autoencoder and entropy to obfuscate

real labels to fool the adversary, and introduces an enhanced obfuscation autoencoder method to defend against various label inference attacks.

When combing the research on vertical federation clustering, it is found that the current related research on vertical federation clustering mainly focuses on optimizing parameter initialization, improving clustering precision and accelerating model convergence [16]. Aiming at the risk of privacy leakage in the parameter transfer process of vertical federated learning, no relevant research has pointed out the way of privacy leakage. Meanwhile, at present, the existing vertical federated clustering privacy protection method does not consider the threat of semi-trusted servers to the privacy security of participants [17], and the sensitive information related to data contained in the parameters uploaded by participants. At the same time, no study has pointed out the relationship between parameters and data [18].

Federated learning may expose data feature distributions or user behavior patterns during parameter transmission, and attackers can reverse deduce the original data through inversion attacks, member inference attacks, and other means. For example, gradient updates may carry sensitive information, leading to indirect leakage risks in both horizontal and vertical scenarios Secondly, frequent transmission of model parameters can lead to high communication overhead, especially in vertical federated scenarios where cross institutional data feature alignment and encryption calculations significantly increase system complexity. In vertical federated scenarios, the overlap of participant data features is low and the distribution differences are large, making global model convergence difficult. Local model updates may introduce biases and reduce clustering accuracy In terms of security, there are new types of attacks such as model poisoning attacks (malicious participants upload and tamper with parameters to destroy the global model) and man in the middle attacks (stealing transport layer parameters), and existing encryption methods are difficult to fully defend against.

The privacy attack recognition model of vertical federated clustering introduces a real-time monitoring module in the parameter aggregation stage. By comparing historical parameter updates to identify abnormal gradients, potential attack behaviors are marked. Based on the sample ID alignment characteristics of vertical federated clustering, a feature cross validation mechanism is constructed to detect attribute inference attacks initiated by non collaborators through public ID associations Partial homomorphic encryption is used instead of fully homomorphic encryption to selectively encrypt core parameters such as cluster center vectors, balancing security and computational efficiency. A layered perturbation mechanism is designed: differential

privacy is used to add Gaussian noise to low sensitivity features, and secure multi-party computation (MPC) is used for joint computation of high sensitivity features.

3 Privacy Preservation Algorithm for Vertical Federated Clustering Based on Localized DP

In the clustering algorithm under the vertical federated learning architecture, replacing the sharing of training data with distance parameters cannot guarantee the privacy and security of training data, and the semi-trusted server can perform inference according to the parameters uploaded by the participants.

In this paper, DP technology is used to add disturbance to the distance parameters, so as to improve the privacy protection degree of vertical federated clustering algorithm. Moreover, a discretization method is proposed, in which the participants fuse and cut the distance parameters before sharing them, only retain the valuable part in the parameter aggregation stage, and reduce the influence of random disturbance on the clustering results. Then, the parameters are discretized, and the continuous numerical parameters are converted into discrete numerical parameters. Furthermore, a random response strategy satisfying ε-DP is proposed, which adds random perturbation to discrete numerical parameters to improve the privacy protection degree of clustering algorithm under the vertical federated learning architecture.

Based on practical needs, a general framework for vertical federated clustering privacy protection algorithm is proposed. In this model, a localized DP protection algorithm based on random responses is used for system security privacy protection, mainly including distance parameter fusion and discretization processing, Gap RR localization random perturbation algorithm and privacy protection analysis three algorithms are used to improve the reliability of the model.

3.1 Overall Framework of Privacy Preservation Algorithm for Vertical Federated Clustering

A common federated learning model architecture is the client-server architecture, which contains roles divided into client and server. As shown in Figure 1, the client, as the data holder, and the server jointly participate in training the model in a specified way. In the system initialization phase, a portion of clients extracted by the server downloads the shared global model

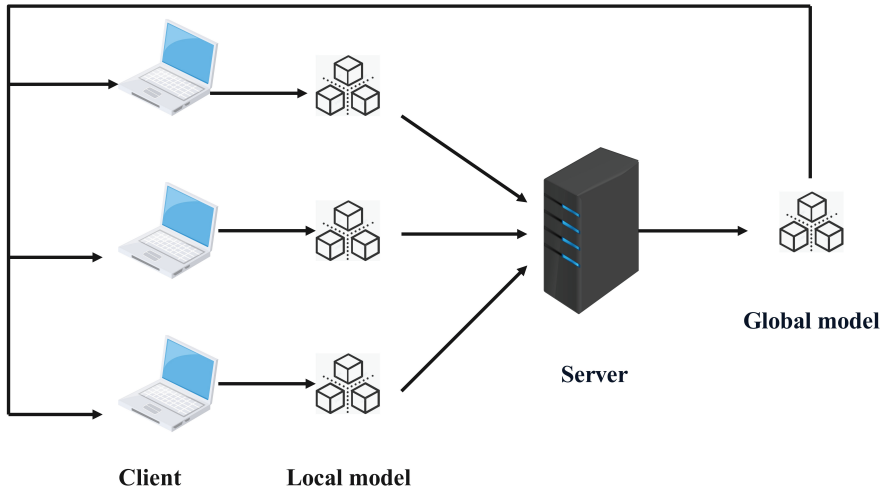


Figure 1 Federated learning model architecture.

from the server, and each participant (client) independently trains the model on its local dataset [19].

Vertical Federated Learning (VFL) is an important branch of federated learning. It is a feature-based distributed machine learning designed for data sets with similar or identical sample spaces but different feature spaces. Under this framework, different entities (such as various institutions or organizations) have different feature spaces of the same sample, and they jointly build machine learning models while ensuring that the privacy of their own data is protected [20].

In longitudinal federated learning (Figure 2), different participants hold datasets containing the same sample space but different feature spaces. In this case, the data contributed by each participant has the same sample identifier (such as user ID), but the attributes or characteristics of each sample are different. The key challenge of vertical federated learning lies in how to effectively integrate feature information from different sources and maintain data security and privacy throughout the process.

In this paper, a pseudo-distributed federated learning architecture is used to unite the participants with different attribute data with the same sample ID to cluster the whole data. Before the discretization of distance parameters, a parameter processing method is designed to remove useless information in distance parameters, so as to reduce the amount of disturbance added and improve the clustering effect.

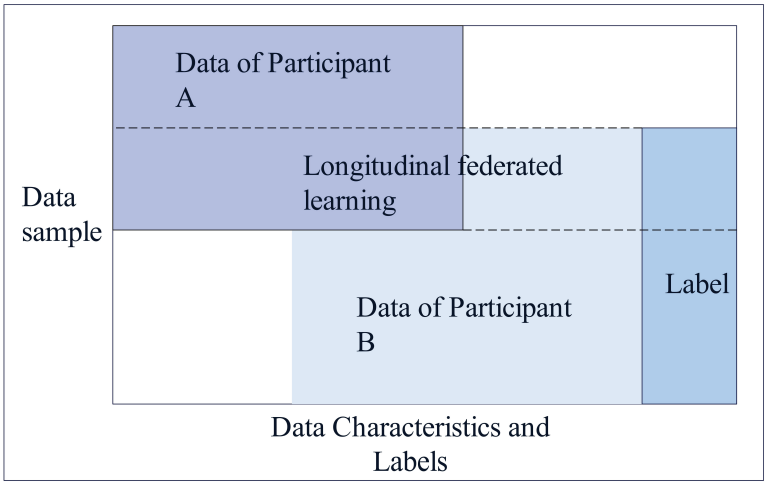


Figure 2 Vertical federated learning.

After that, the fusion parameters are discretized. According to the minimum and maximum values of the fusion parameters, Q discrete values with the same spacing are set. At the same time, each fusion parameter selects the nearest discrete value as the result of discretization processing [21].

In this paper, random response mechanism is used to add random perturbation to discrete parameters, and the discrete parameters with random perturbation are uploaded to the server. Then, the number of the cluster center to which each sample belongs is calculated by the server side and the result is issued to all participants. After that, the participants calculate the new clustering center and recalculate the distance between the sample and the clustering center, and repeat the above steps until the clustering center is stable or reaches the maximum iteration round, and the global clustering training ends. The overall framework of the vertical federated clustering privacy protection algorithm is shown in Figure 3.

The specific steps are as follows:

(1) Each participant P_h initializes P_h cluster centers $C = \{c_1, c_2, \ldots, c_j, \ldots, c_k\}$ on its local attributes and calculates the Euclidean distance from the sample to c_k.

(2) Participant P_h adds the distances of different attributes to merge multiple attributes into one attribute, and cuts the part that will not affect the clustering result from the parameters to obtain the fusion distance parameter m_h^{sum}, and m_h^{sum} is a one-dimensional matrix with a dimension of $1 \times k$.

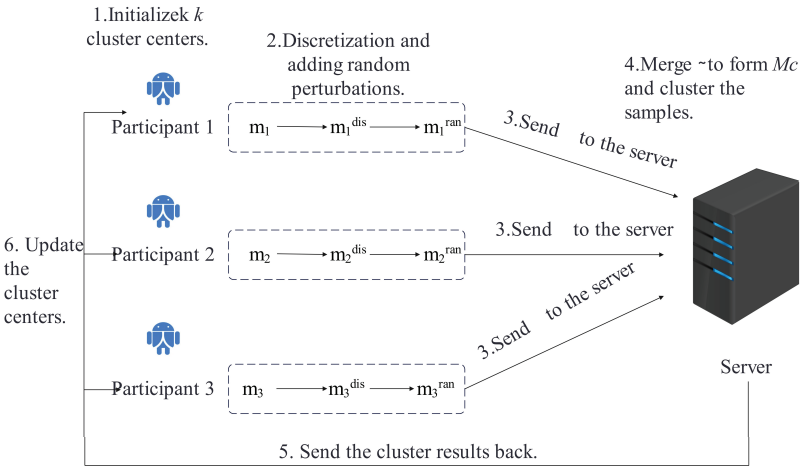


Figure 3 Overall framework diagram of vertical federated clustering privacy protection algorithm based on localized DP.

(3) The participants discretize the fusion distance parameters and use the random response mechanism to add random perturbations that satisfy the localized ε-DP to the discretized fusion distance parameters.
(4) The server side merges the upload distance matrix from the participants and obtains the global distance in each iteration.
(5) The algorithm returns the clustering results and updates each cluster center c'_j.
(6) The algorithm repeats steps (2) to (5) until convergence or the maximum number of iterations is reached.

3.2 Localized DP Protection Algorithm Based on Random Response

3.2.1 Distance parameter fusion and discretization processing

This section introduces the discretization process of distance parameters. First, the distance parameters are fused to eliminate the parts unrelated to clustering, and then the fusion parameters are discretized by using the idea of equal width binning. This process takes the discretization process of a single sample s_i held by a participant P_h as an example. The specific steps are as follows:

(1) Participant P_h randomly generates k initial cluster centers $C = \{c_1, c_2, \cdots, c_j, \ldots, c_k\}$ for the d_h attributes it holds, calculates the

distance between each attribute and the cluster center for each sample s_i, and each participant holds a matrix of dimension$d_h \times k$ [22]:

$$m_h = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1d_h} \\ m_{21} & m_{22} & \cdots & m_{2d_h} \\ \vdots & \vdots & \ddots & \vdots \\ m_{k1} & m_{k2} & \cdots & m_{kd_h} \end{bmatrix} \tag{1}$$

(2) The distance parameters of different attributes are summed. For sample s_i, each participant has a component of the distance corresponding to k clusters, which is equivalent to all participants sharing a matrix of dimension $k \times H$. H participants each hold vectors m_h of k elements [23].

$$p_1 \; has \; m_1 = \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{k1} \end{bmatrix}$$

$$p_2 \; has \; m_2 = \begin{bmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{k2} \end{bmatrix}$$

$$p_H \; has \; m_H = \begin{bmatrix} x_{1H} \\ x_{2H} \\ \vdots \\ x_{kH} \end{bmatrix} \tag{2}$$

(3) Each participant P_h calculates the distance between the cluster center closest to the sample and the sample, and subtracts this value from all distances.

$$m_h = m_h - \min_j x_{jh} \tag{3}$$

Among them, $\min_j x_{jh}$ represents the distance between the sample and the nearest cluster center.

(4) The maximum value $\max_h$ and the minimum value $\min_h$ of all distances are calculated. According to $\max_h$ and $\min_h$, a discrete value set $A = \{a_1, a_2, \ldots, a_Q\}$ containing Q discrete values is set, and the

difference between two adjacent discrete values is $\frac{\max_h - \min_h}{Q-1}$. Each parameter selects the nearest discrete value to obtain the discretized fusion parameter m_h^{dis} [24].

$$x_{jh} = \arg\min_{a_k \in A} |a_q - x_{jh}| \tag{4}$$

$$m_h^{dis} = [x_{1h}, x_{2h}, \ldots, x_{kh}] \tag{5}$$

3.2.2 Gap-RR localized random perturbation algorithm

The important problem of the random response mechanism is the setting of the answer set. If all parameters adopt the same answer set, when the set contains more elements, the probability of the output value being the true value will be small, the probability of the output random value will be high, and the addition of too many random disturbances will lead to the decline of data utility. To solve this problem, this section proposes a random response algorithm Gap-RR (Gap Randomized Response) with variable response interval, which makes the number of elements in the response interval of each parameter fixed to 3. Moreover, it changes the amount of disturbance added by adjusting the size of the difference between the false value and the true value of the random response set, thereby improving the data utility while ensuring randomness.

Figure 4 shows the setting of the random response set. The discretization process converts the continuous numerical data into discrete values to obtain

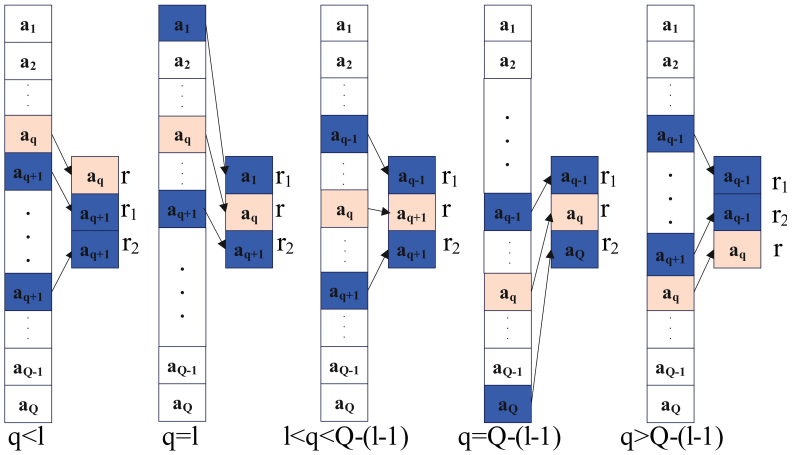


Figure 4 Setting of random response set.

a set $A = \{a_1, a_2, \ldots, a_Q\}$ containing Q discrete values. By setting the set of random responses with discrete values and using the random response mechanism for each distance parameter, random perturbation is added to the parameters. The specific steps are as follows:

(1) The distance between the false value and the true value is defined by l, which is determined based on l_{ratio} [25].

$$l = \begin{cases} \dfrac{Q \times l_{ratio} - 1}{2} & \textit{When Q is an odd number} \\ \dfrac{Q \times l_{ratio}}{2} - 1 & \textit{When Q is an even number} \end{cases} \tag{6}$$

(2) All elements in the discrete value set of each participant are determined to be random response sets $R_q = r, r_1, r_2$, R_q contain f as the true value and two other discrete values r_1 and r_2 belonging to set A and regarded as false values. The setting of the random response set includes 5 different cases to ensure that the output value of the random response mechanism meets the definition of ε-localized DP [26].

$$R_q = \begin{cases} \{a_q, a_{q+1}, a_{q+l}\} & q < l \\ \{a_q, a_1, a_{q+l}\} & q = 1 \\ \{a_q, a_{q-l}, a_{q+l}\} & l < q < Q - (l-1) \\ \{a_q, a_{q-l}, a_{q-1}\} & q = Q - (l-1) \\ \{a_q, a_{q-1}, a_{q-l}\} & q > Q - (l-1) \end{cases} \tag{7}$$

(3) The GRR local perturbation method is used to add random perturbations to each parameter of the participant P_h. $a_i, a_j \in A$ is set, and the real value or random value is output according to a certain probability, so that the output result conforms to the output probability of the GRR algorithm.

$$Pr[GRR(a_i) = a_j] = \begin{cases} p = \dfrac{e^{\varepsilon}}{e^{\varepsilon} + d - 1} & a_i = a_j \\ q = \dfrac{1-p}{d-1} & a_i \neq a_j \end{cases} \tag{8}$$

3.2.3 Privacy protection degree analysis

In this paper, it is considered that the privacy protection degree of Gap-RR localization perturbation algorithm meets the ε-localized DP. The following will demonstrate this.

In the response set of all discrete values, each discrete value has a certain probability of outputting the true value, and a certain probability of outputting the other two values. Gap-RR algorithm has different random response sets for each discrete value, but the probability of outputting untrue values is the same, and the number of untrue values contained in the random response set is the same.

For each discrete value $r \in M_h^{dis}$ of the Gap-RR input, there are three possible output values, r, r_1, and r_2. When $r_i, r_j \in \{r, r_1, r_2\}$ is set, the Gap-RR algorithm is [27]:

$$Pr[Gap - RR(r_i) = r_j] = \begin{cases} p = \dfrac{e^{\varepsilon}}{e^{\varepsilon} + d - 1} = \dfrac{e^{\varepsilon}}{e^{\varepsilon} + 1} & r_i = r_j \\ q = \dfrac{1-p}{d-1} = \dfrac{1}{e^{\varepsilon} + 1} & r_i \neq r_j \end{cases} \tag{9}$$

For any r_i and r_j, if the output of Gap-RR is r, then the inequality holds [28]:

$$\frac{Pr[Gap - RR(r_i) = r]}{Pr[Gap - RR(r_i) = r]} \leq \frac{e^{\varepsilon}/e^{\varepsilon} + 1}{1/e^{\varepsilon} + 1} = e^{\varepsilon} \tag{10}$$

According to the definition of ε-localized DP, it can be seen that the Gap-RR algorithm meets the ε-localized DP, so the proof is complete.

4 Experimental Analyses

4.1 Experimental Methods

(1) Experimental data

Six publicly available datasets are selected as training data for the experiment. They are forge, wave, MNIST, ImageNet, REDSTONE-Web1, UCI datasets, respectively. The number of data selected from forge, wave, and MNIST is 500, and the number of data selected from ImageNet, REDSTONE-Web1, and UCI is 5000, 10000, and 30000, respectively. These datasets cover different domains and data types, provide a wide range of testing and validation scenarios, and can effectively evaluate the performance of privacy-preserving algorithms for vertical federated clustering [29].

(2) Experimental setup

This experiment simulates the process of federated learning on a computer. Using Python language and based on Pytorch open source framework, a

federated learning simulation platform is constructed, which simulates the training of participants and the federated communication process with the server through multi-threading. In the experimental setting, the number of participants is 2, and each participant holds a part of the attributes of the data set, so as to divide the number of attributes equally as much as possible. The initialization of cluster centers is randomly generated. Each data set generates a set of cluster centers, and different parameters are clustered with the same cluster center, so as to avoid the difference of results caused by the randomness of cluster center initialization. The maximum iteration rounds are set to 20 rounds.

The experiment is divided into two aspects: one is to verify the influence of different discrete values on the clustering results, and the other is to verify the influence of different privacy budgets on the clustering results.

(1) The first set of experiments verifies the influence of discretization processing distance parameters on clustering results. The number of discrete values is from 5 to 50. Ten groups of experiments are set, and the experimental interval of each group is 5. Taking the clustering results of continuous clustering as the benchmark, a group of experiments with random noise are set, and the parameters are set as $\varepsilon = 0.5, l_{ratio} = 0.5$. The experiments with each parameter are trained for 10 times, and the average value is taken as the final result.

(2) The second set of experiments verifies the influence of different probabilities and sizes of adding noise on the clustering results by adjusting different ε and l_{ratio}. ε is set from 0.1 to 1.0 in 10 sets of parameters, and the experimental interval of each set is 0.1, and l_{ratio} is set from 0.2 to 1.0 in 5 sets of parameters. Taking the clustering based on discretization distance without adding noise as the benchmark, the experiments of each parameter are trained for 10 times, and the average value is taken as the final result.

(3) Evaluation Index

According to the clustering results of clustering algorithm in vertical federated learning environment, F-measure is used to measure the clustering precision. F-measure is often used to evaluate classification tasks, and it is often used as an external evaluation index of clustering. It is a comprehensive index of Precision and Recall. Considering the precision and recall, it is calculated by harmonic average method.

In order to apply F value to the clustering in this paper, it needs to be extended into a form suitable for clustering results. The specific calculation

steps of F value in clustering are as follows [30]:

(1) First, it is necessary to calculate the precision rate, the recall rate, and the F-value for each category (cluster).
Precision: For category i, precision is defined as the proportion of the number of samples correctly classified into that category in the clustering results to the total number of samples in that category.

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{11}$$

Among them, TP_i is the number of samples correctly classified into category i, and FP_i is the number of samples incorrectly classified into category i.
Recall: For category i, the recall rate is defined as the proportion of the number of samples correctly classified into the category to the total number of true samples in the category.

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{12}$$

Among them, FN_i is the number of samples that are not correctly classified into category i.
F-value: For category i, the F-value is the harmonic mean of precision and recall.

$$F_i = \frac{2 \times P_i \times R_i}{P_i + R_i} \tag{13}$$

(2) The weight of each category is calculated
The weights can be calculated based on the actual number of samples for each category to reflect the importance of the category. The weight ω_i of category i is defined as the proportion of the number of samples in this category to the total number of samples.

$$\omega_i = \frac{n_i}{N} \tag{14}$$

Among them, n_i is the number of samples of category i and N is the total number of all samples.

(3) The weighted F value is calculated
The total F-value (weighted F-value) is the sum of the F-values of each category weighted by their weights.

$$F = \sum_{i=1}^{k} \omega_i \times F_i \tag{15}$$

Among them, k is the total number of categories.

4.2 Experimental Results

The influence of different numbers of discrete values on the experimental results is shown in Figure 5. Each subgraph contains a baseline and two polylines, and the baseline is the clustering result based on continuous distance parameters. The two polylines are the clustering results based on discrete distance parameters and the clustering results based on discrete distance parameters with random perturbation [31].

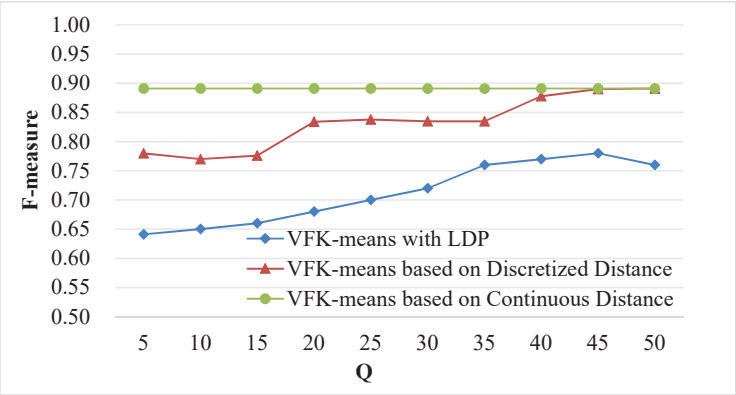
The effects of different privacy budgets and noise levels on clustering precision are shown in Figure 6. Each subgraph contains 1 baseline and 5 polylines. The baseline is the clustering precision based on discrete distance when no noise is added, and the line chart shows the change of F value as the value of ε increases from 0.1 to 1.0 when $l_{ratio} = 0.2, 0.4, 0.6, 0.8$ and 1.0 are respectively.

Table 1 shows the influence of random noise on clustering precision under four different parameter settings.

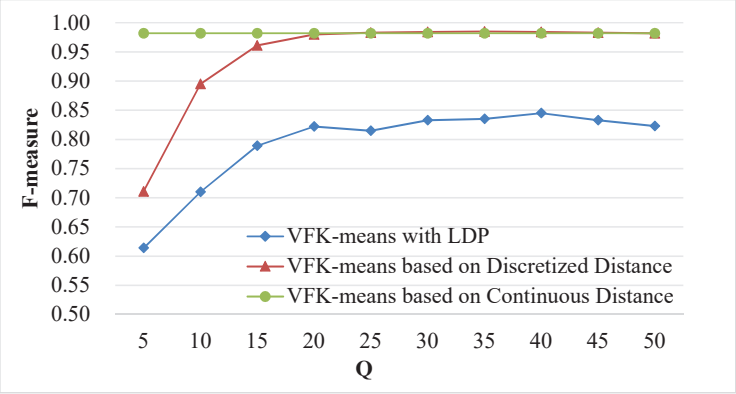
In order to further verify the protection effect of the model in this paper, this paper takes the UCI data set as the research object. Moreover, this paper integrates PPA (Proactive Password Auditor) attacks, ARP (Address Resolution Protocol) spoofing attacks, man-in-the-middle attacks (MITM), data Sniffing (Sniffing), and Botnet (Botnet) concentrated attacks into the test to explore the privacy attack identification rate and protection effect under different attack types, in which the protection effect is quantitatively analyzed through expert subjective evaluation methods. The algorithm proposed in this paper is named RSM-DP (Random response mechanism-Differential privacy), and the model proposed in this paper is compared with the DP

Table 1 Influence of random noise on clustering precision under different parameter settings

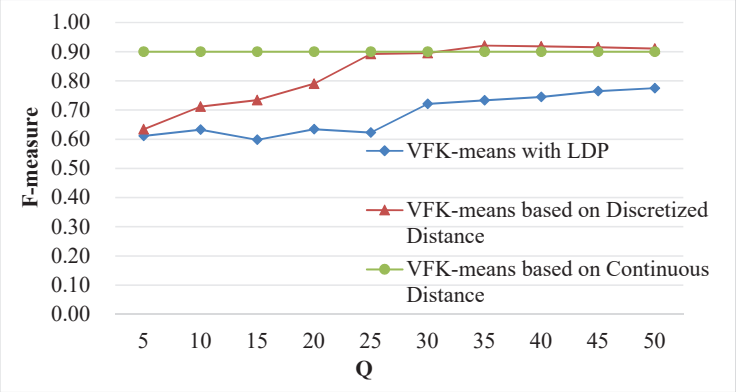
	lratio	ε	lratio	ε	lratio	ε	lratio	ε	lratio	ε
Data Set	1.0	0.1	1.0	1.0	0.2	0.1	0.2	1.0	/	/
Forge	0.6237		0.7227		0.8514		0.8613		**0.8811**	
Wave	0.7227		0.792		0.9306		0.9405		**0.9603**	
MNIST	0.6039		0.7128		0.8316		0.8712		**0.8811**	
ImageNet	0.7623		0.8019		0.891		0.9009		**0.9108**	
REDSTONE-Web	0.8019		0.8316		0.9504		0.9603		**0.9702**	
UCI	0.3663		0.5247		0.9009		0.9009		**0.9009**	



(a) Forge dataset

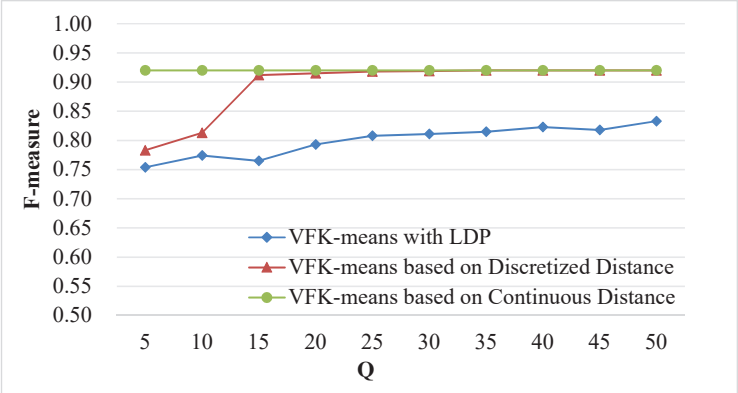


(b) Wave dataset

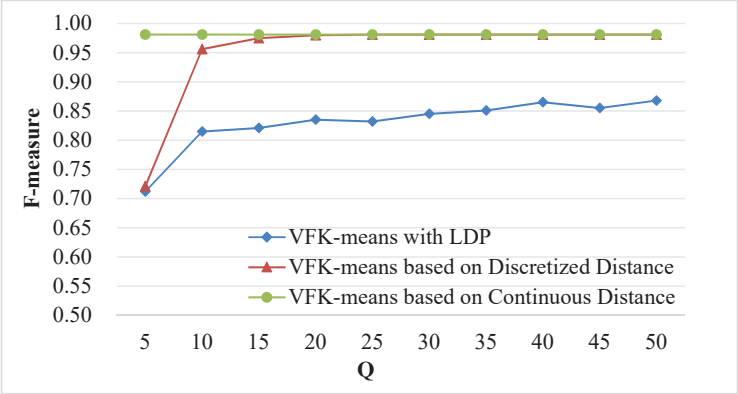


(c) MNIST dataset

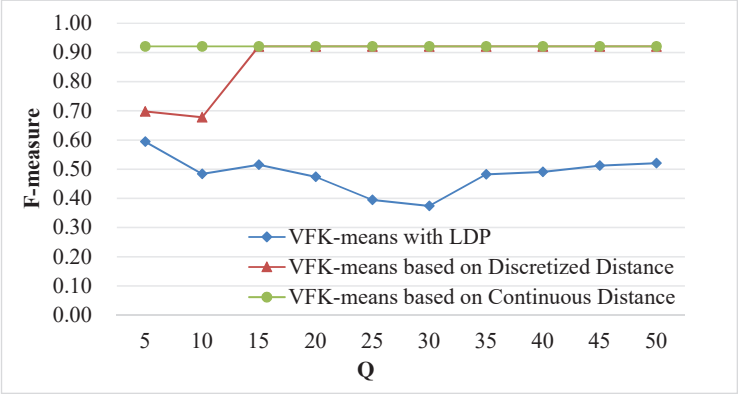
Figure 5 Continued



(d) Imagenet dataset

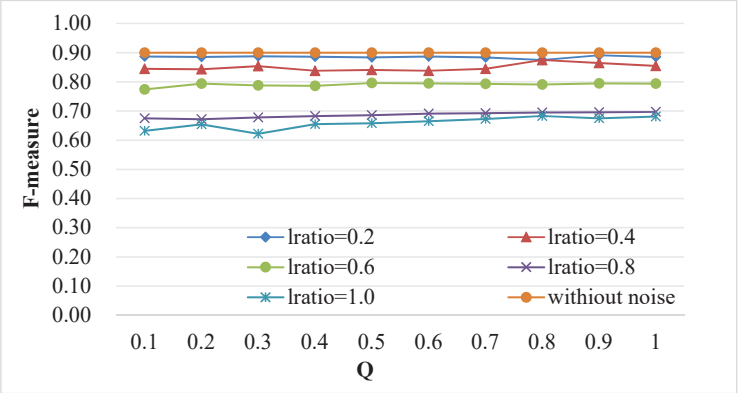


(e) Redstone-web1 dataset

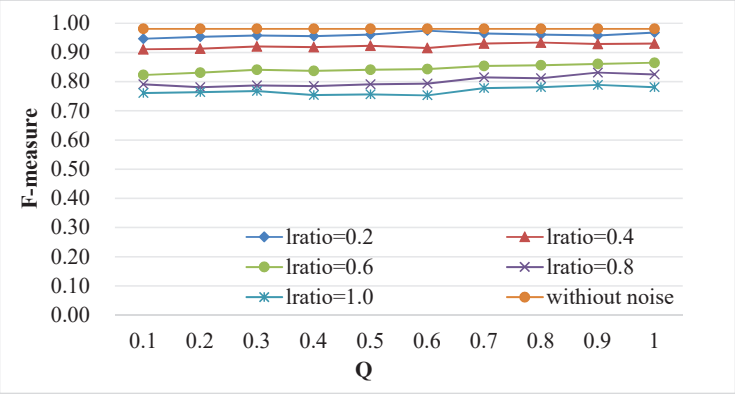


(f) UCI dataset

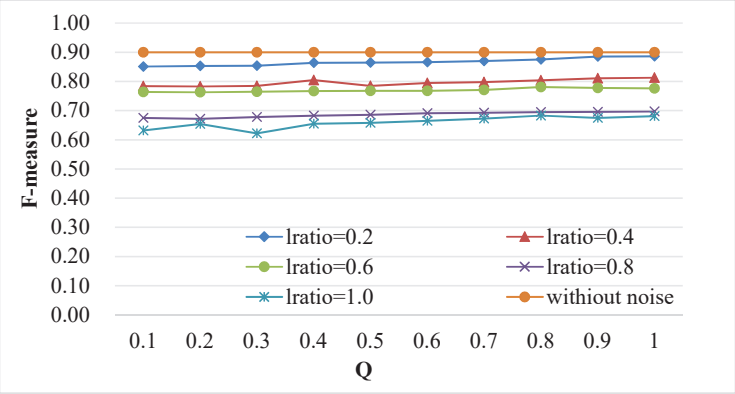
Figure 5 Influence of the number Q of discrete values on the F value.



(a) Forge dataset

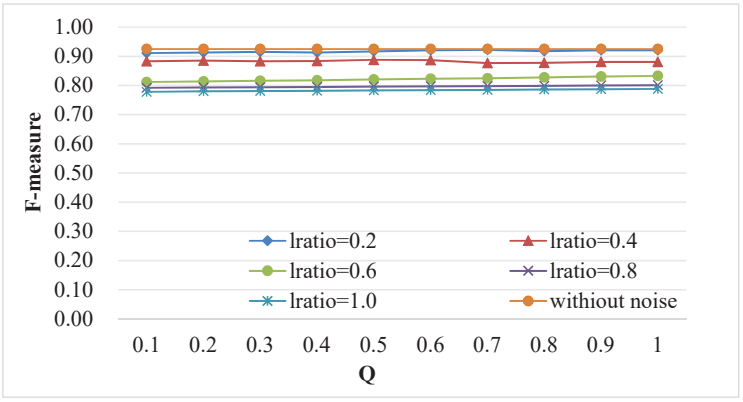


(b) Wave dataset

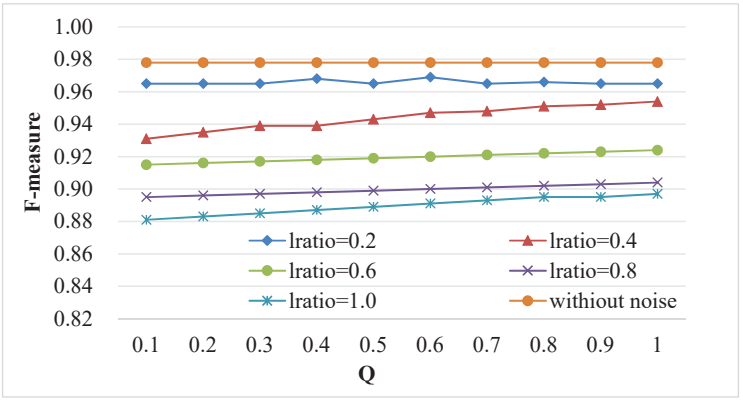


(c) MNIST dataset

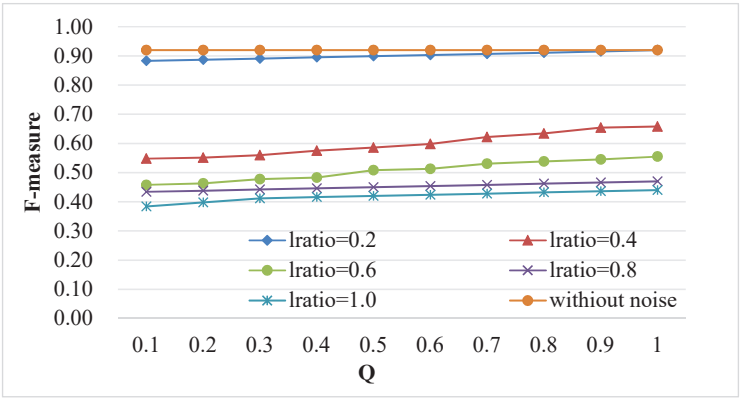
Figure 6 Continued



(d) Imagenet dataset



(e) Redstone-web1 dataset



(f) UCI dataset

Figure 6 Influence of privacy budget ε on the F value.

Table 2 Comparison of privacy attack identification and protection effects of different models

Attack Mode	Protection Model	Recognition Rate (%)	Privacy Protection Effect	Recall Rate (%)	F Value
PPA	DP	87.05	87.76	89.57	0.88
	HE	80.92	84.54	85.29	0.85
	ZKP	85.17	86.38	86.81	0.86
	RSM-DP	**88.21**	**91.28**	**92.74**	**0.91**
ARP	DP	83.85	86.86	91.16	0.90
	HE	75.17	75.18	76.08	0.75
	ZKP	80.85	83.96	84.35	0.84
	RSM-DP	87.33	89.34	92.33	0.91
MITM	DP	88.32	92.65	96.77	0.96
	HE	75.90	76.61	80.44	0.81
	ZKP	82.19	83.89	87.98	0.86
	RSM-DP	**91.29**	**92.86**	**93.45**	**0.92**
Sniffing	DP	87.10	88.59	92.79	0.92
	HE	75.14	78.21	78.94	0.79
	ZKP	89.89	89.93	93.43	0.92
	RSM-DP	**93.93**	**95.40**	**99.46**	**0.99**
Botnet	DP	85.19	89.42	93.13	0.94
	HE	83.44	84.41	88.43	0.89
	ZKP	92.05	92.44	92.70	0.94
	RSM-DP	**93.67**	**95.14**	**94.45**	**0.96**

model (Differential privacy), the HE model (Homomorphic Encryption) and ZKP model (Zero-Knowledge Proof), and the results of Table 2 are obtained.

4.3 Analysis and Discussion

The experimental results presented in Figure 5 indicate that the discretization of distance parameters results in a decline in clustering precision. Nevertheless, as the number of discrete values increases, the clustering precision gradually draws near to that based on continuous distance parameters. With the increase of discrete values, the final clustering precision of Forge, ImageNet, REDSTONE-Web1 and UCI data sets is the same as clustering precision based on continuous distance parameters. With the increase of the number of discrete values in Wave and MNIST datasets, the clustering precision based on discrete distance exceeds that based on continuous distance parameters. The reason is that the sample size of the dataset is small and the clustering results of individual samples are different due to the discretization

of distances. When the number of discrete values continues to increase, the clustering precision of discrete-based distance and continuous-based distance of Wave and MNIST datasets is the same. In addition, the clustering precision of different discrete values with noise added decreases, but it shows an upward trend with the increase of discrete values. Among them, the clustering precision of UCI data set with added noise is poor. The reason is that the amount of data of different categories in this dataset is very different, and the influence of noise on clustering precision is more obvious.

The experimental results in Figure 6 show that the curves of Forge, Wave, and MNIST data sets with smaller sample sizes fluctuate more obviously. The reason is that the small sample dataset receives the influence of randomness, but the experimental results of three datasets with larger sample sizes, ImageNet, REDSTONE-Web1, and UCI, are smoother. It shows that when the sample size is large, the clustering precision is less affected by randomness. Both ε and l_{ratio} affect clustering precision. As l_{ratio} increases, the size of the noise gradually increases, and the clustering results of the same privacy budget gradually decrease. Under different l_{ratio} settings, the clustering precision is positively correlated with the size of the privacy budget. As the privacy budget j increases, the probability of parameter perturbation decreases, and the F value continues to increase. According to the definition of DP, the larger the privacy budget, the lower the privacy protection degree. The actual results are in line with expectations. The experimental results of the UCI dataset show that as l_{ratio} increases, the clustering precision decreases significantly, indicating that datasets with unbalanced category distribution are more susceptible to random noise.

In Table 1, when $l_{ratio} = 1.0, \varepsilon = 0.1$, the added noise is the largest, the probability of output noise is the largest, and the clustering precision decreases relatively significantly. Among them, the clustering precision of the UCI dataset decreases the most, indicating that datasets with unbalanced category distribution are more susceptible to noise. However, when $l_{ratio} = 0.2, \varepsilon = 1.0$, the added noise is the smallest, and the probability of the output being noise is the smallest, and the clustering precision is closest to that without adding noise. The settings of the two parameters $l_{ratio} = 1.0, \varepsilon = 1.0$ and $l_{ratio} = 0.2, \varepsilon = 0.1$ represent the two situations of maximum noise, minimum output probability and minimum noise, maximum output probability, respectively. The experimental results show that when $l_{ratio} = 0.2, \varepsilon = 0.1$, the clustering precision is higher and almost the same as when no noise is added, indicating that the influence of l_{ratio} on clustering precision is more obvious than that of ε.

By comparing the performance of different privacy protection models in Table 2 under various attack modes, we can analyze from the aspects of identification rate, privacy protection effect, recall rate and F value:

Performance of DP (DP) model: The identification rate and privacy protection effect are generally high. Especially under Botnet attacks, the privacy protection effect reaches 95.14%. The recall rate and F value are stable under different attack methods, and most of them are above 90%, showing good comprehensive performance.

Performance of HE (homomorphic encryption) model: The identification rate is relatively low, especially under Sniffing and Botnet attacks. The identification rate under Sniffing attack is only 75.14%. The privacy protection effect is also relatively low, and the highest value is 76.61% under MITM attack.

Moreover, its recall rate and F-value are generally low. Especially under Sniffing and Botnet attacks, the F values are 0.78 and 0.87, respectively, indicating that their privacy protection effect is weak.

Performance of ZKP (Zero Knowledge Proof) model: It performs well under PPA, ARP and Sniffing attacks. Especially under Sniffing attacks, the privacy protection effect reaches 89.93%. Identification rate and recall rate perform better under some specific attack modes, but the overall fluctuation is large, which is not as stable as DP model.

Performance of RSM-DP model: The overall performance is good, and especially under ARP and Botnet attacks, the privacy protection effect reaches 89.34% and 95.14% respectively. The identification rate and recall rate are generally high, showing better privacy protection ability and model robustness.

On the whole, the HE model generally performs poorly in various attack modes, indicating that its ability to resist attacks is weak. DP and RSM-DP models perform more stably under different attack modes, showing strong privacy protection capabilities. The ZKP model performs well in some specific attack modes, but the overall fluctuation is large. The RSM-DP model has the best performance in terms of overall privacy protection effect and stability, followed by the DP model. However, the performance of HE model is generally poor in resisting different attack modes, while the performance of ZKP model fluctuates greatly under different attack modes. DP's mathematically provable privacy protection mechanism, dynamic parameter adjustment capability and multi-technology collaborative compatibility have become the core solutions that take into account privacy security and data value mining in current data-driven services (such as AI training and cross-institutional data

analysis). However, there will be some data redundancy and overfitting in actual clustering. Moreover, although the comprehensive capability is strong, there are still some problems. In addition, HE and ZKP models are protection models in specific scenarios, and their comprehensive performance is insufficient. RSM-DP model is an improvement on DP model. By designing a discrete processing method, the algorithm converts the continuous distance parameters generated in the clustering process into discrete parameters, and uses the random response mechanism to add random noise in accordance with DP to these discrete parameters. Subsequently, the distance parameters with random noise are uploaded to the server, which calculates them and returns the results to the participants. After receiving the results, the participants calculate a new cluster center and update the distance between the sample and the cluster center until the cluster center converges or reaches the maximum number of iterations. Therefore, it has a better data processing effect than the DP model.

The privacy protection model based on vertical federated clustering reduces multidimensional data to one-dimensional vectors through parameter fusion, which may result in the loss of key features, especially in high-dimensional sparse data scenarios. The reduced vectors are difficult to fully preserve the distribution characteristics of the original data, affecting clustering accuracy. The overlapping degree of data features among all participants is low, and the reduced one-dimensional vectors may amplify local data distribution differences, leading to global clustering center shift To address the above limitations, improvements can be made in terms of hierarchical dimensionality reduction, feature selection optimization, and dynamic privacy budget allocation mechanisms.

5 Conclusion

The main work of this paper is to expand the clustering algorithm to the vertical federated learning architecture, and design attack methods to address the privacy leakage risks therein. From the perspective of semi-trusted server, by analyzing the distance parameters passed in the clustering process, the data reconstruction attack on the participants' data sets is completed. At the same time, this paper designs a privacy protection method based on DP to defend against various attack methods including the data reconstruction attack proposed in this paper, thereby solving the problem of insufficient privacy protection of the vertical federated clustering algorithm. Combined with the experimental analysis, it can be seen that the algorithm proposed

in this paper has a good overall performance, and especially under ARP and Botnet attacks, the privacy protection effect reaches 89.34% and 95.14% respectively. Furthermore, the identification rate and recall rate are generally high, showing good privacy protection ability and model robustness.

Further improvements can be made to the model in terms of hierarchical dimensionality reduction, feature selection optimization, and dynamic privacy budget allocation mechanism Adopting a hierarchical feature fusion strategy to preserve key high-dimensional features (such as filtering important parameters through attention mechanisms), avoiding information loss caused by global dimensionality reduction, and dynamically adjusting disturbance intensity based on feature sensitivity: allocating higher privacy budget to high correlation parameters, and reducing disturbance to low sensitivity parameters to improve data utility.

References

[1] Liu, Y., Kang, Y., Zou, T., Pu, Y., He, Y., Ye, X., ... and Yang, Q. (2024). Vertical federated learning: Concepts, advances, and challenges. IEEE Transactions on Knowledge and Data Engineering, 36(7), 3615–3634.

[2] Gu, B., Xu, A., Huo, Z., Deng, C., and Huang, H. (2021). Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning. IEEE transactions on neural networks and learning systems, 33(11), 6103–6115.

[3] Novikova, E., Doynikova, E., and Golubev, S. (2022). Federated learning for intrusion detection in the critical infrastructures: Vertically partitioned data use case. Algorithms, 15(4), 104–115.

[4] Jia, B., Zhang, X., Liu, J., Zhang, Y., Huang, K., and Liang, Y. (2021). Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT. IEEE Transactions on Industrial Informatics, 18(6), 4049–4058.

[5] Wu, J. M. T., Teng, Q., Huda, S., Chen, Y. C., and Chen, C. M. (2023). A privacy frequent itemsets mining framework for collaboration in IoT using federated learning. ACM Transactions on Sensor Networks, 19(2), 1–15.

[6] Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., ... and He, B. (2021). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Transactions on Knowledge and Data Engineering, 35(4), 3347–3366.

[7] Zhou, X., Ye, X., Kevin, I., Wang, K., Liang, W., Nair, N. K. C., ... and Q. (2023). Hierarchical federated learning with social context clustering-based participant selection for internet of medical things applications. IEEE Transactions on Computational Social Systems, 10(4), 1742–1751.

[8] Li, D., Luo, Z., and Cao, B. (2022). Blockchain-based federated learning methodologies in smart environments. Cluster Computing, 25(4), 2585–2599.

[9] Ouyang, L., Wang, F. Y., Tian, Y., Jia, X., Qi, H., and Wang, G. (2023). Artificial identification: A novel privacy framework for federated learning based on blockchain. IEEE Transactions on Computational Social Systems, 10(6), 3576–3585.

[10] Tian, Y., Zhang, Z., Xiong, J., Chen, L., Ma, J., and Peng, C. (2021). Achieving graph clustering privacy preservation based on structure entropy in social IoT. IEEE Internet of Things Journal, 9(4), 2761–2777.

[11] Kreso, I., Kapo, A., and Turulja, L. (2021). Data mining privacy preserving: Research agenda. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 11(1), e1392–e1403.

[12] Wang, J., Pal, A., Yang, Q., Kant, K., Zhu, K., and Guo, S. (2022). Collaborative machine learning: Schemes, robustness, and privacy. IEEE Transactions on Neural Networks and Learning Systems, 34(12), 9625–9642.

[13] Sadilek, A., Liu, L., Nguyen, D., Kamruzzaman, M., Serghiou, S., Rader, B., ... and Hernandez, J. (2021). Privacy-first health research with federated learning. NPJ digital medicine, 4(1), 132–144.

[14] Domadiya, N., and Rao, U. P. (2021). Improving healthcare services using source anonymous scheme with privacy preserving distributed healthcare data collection and mining. Computing, 103(1), 155–177.

[15] Beltrán, E. T. M., Pérez, M. Q., Sánchez, P. M. S., Bernal, S. L., Bovet, G., Pérez, M. G., ... and Celdrán, A. H. (2023). Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. IEEE Communications Surveys & Tutorials, 25(4), 2983–3013.

[16] Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., and Zhang, W. (2023). A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics, 14(2), 513–535.

[17] Han, M., Xu, K., Ma, S., Li, A., and Jiang, H. (2022). Federated learning-based trajectory prediction model with privacy preserving for

intelligent vehicle. International journal of intelligent systems, 37(12), 10861–10879.

[18] Jie, Z., Chen, S., Lai, J., Arif, M., and He, Z. (2023). Personalized federated recommendation system with historical parameter clustering. Journal of Ambient Intelligence and Humanized Computing, 14(8), 10555–10565.

[19] Singh, S., Rathore, S., Alfarraj, O., Tolba, A., and Yoon, B. (2022). A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology. Future Generation Computer Systems, 129(2), 380–388.

[20] Zhou, X., Yang, Q., Zheng, X., Liang, W., Kevin, I., Wang, K., ... and Jin, Q. (2024). Personalized federated learning with model-contrastive learning for multi-modal user modeling in human-centric metaverse. IEEE Journal on Selected Areas in Communications, 42(4), 817–831.

[21] Shiau, W. L., Wang, X., and Zheng, F. (2023). What are the trend and core knowledge of information security? A citation and co-citation analysis. Information & Management, 60(3), 103774–103788.

[22] Wang, R., and Tsai, W. T. (2022). Asynchronous federated learning system based on permissioned blockchains. Sensors, 22(4), 1672–1684.

[23] Alzubi, J. A., Alzubi, O. A., Singh, A., and Ramachandran, M. (2022). Cloud-IIoT-based electronic health record privacy-preserving by CNN and blockchain-enabled federated learning. IEEE Transactions on Industrial Informatics, 19(1), 1080–1087.

[24] Menaga, D., and Saravanan, S. (2022). GA-PPARM: constraint-based objective function and genetic algorithm for privacy preserved association rule mining. Evolutionary Intelligence, 15(2), 1487–1498.

[25] Lee, J., Solat, F., Kim, T. Y., and Poor, H. V. (2024). Federated learning-empowered mobile network management for 5G and beyond networks: From access to core. IEEE Communications Surveys & Tutorials, 26(3), 2176–2212.

[26] Xenakis, A., Chen, Z., and Karabatis, G. (2024). A cluster-based approach for distributed anonymisation of vertically partitioned data. International Journal of Web Engineering and Technology, 19(4), 397–420.

[27] Zhu, X., Wang, D., Pedrycz, W., and Li, Z. (2023). Privacy-preserving realization of fuzzy clustering and fuzzy modeling through vertical federated learning. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 54(2), 915–924.

[28] Manzoor, H. U., Shabbir, A., Chen, A., Flynn, D., and Zoha, A. (2024). A survey of security strategies in federated learning: Defending models, data, and privacy. Future Internet, 16(10), 374.

[29] Islam, T. U., Mohammed, N., and Alhadidi, D. (2024). Privacy preserving vertical distributed learning for health data. Journal of Surveillance, Security and Safety, 5(1), 1–18.

[30] Xia, F., and Cheng, W. (2024). A survey on privacy-preserving federated learning against poisoning attacks. Cluster Computing, 27(10), 13565–13582.

[31] Wang, Y., Zheng, W., Liu, Z., Wang, J., Shi, H., Gu, M., and Di, Y. (2023). A federated network intrusion detection system with multi-branch network and vertical blocking aggregation. Electronics, 12(19), 4049–4060.

Biographies

Mingshan Fan was born in Shanxi, China, in 1979. From 1998 to 2002, he studied at Taiyuan Normal University and obtained his bachelor's degree in 2002. From 2002 to 2023, he worked at No. 3 Middle School, Datong. Currently, he works at the College of Finance and Economics of Taiyuan University of Technology and Shanxi Finance & Taxation College. He has published five papers, and his research interests is information technology.

Huijuan Guo was born in Shanxi, China, in 1979. From 1998 to 2002, she studied at Taiyuan Normal University and obtained her bachelor's degree in 2002. From 2002 to 2006, she studied at Xihua University and received her master's degree in 2006. She worked at Taiyuan Normal University from 2006 to 2019. Currently, she is studying for a doctoral degree at Taiyuan University of Technology. She has published several papers, and her research interests include image processing, machine learning, and deep learning.