Hindawi
Security and Communication Networks
Volume 2022, Article ID 7963004, 13 pages
https://doi.org/10.1155/2022/7963004

Research Article

Horizontally Partitioned Data Publication with Differential Privacy

Zhen Gu,[1,2] **Guoyin Zhang**,[1] **and Chen Yang**[1]

[1]*College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*
[2]*The Department of Basic Education, East University of Heilongjiang, Harbin 150066, China*

Correspondence should be addressed to Guoyin Zhang; zhangguoyin@hrbeu.edu.cn

Received 13 April 2022; Revised 13 June 2022; Accepted 6 July 2022; Published 31 July 2022

Academic Editor: Debiao He

In this paper, we study the privacy-preserving data publishing problem in a distributed environment. The data contain sensitive information; hence, directly pooling and publishing the local data will lead to privacy leaks. To solve this problem, we propose a multiparty horizontally partitioned data publishing method under differential privacy (HPDP-DP). First, in order to make the noise level of the published data in the distributed scenario the same as in the centralized scenario, we use the infinite divisibility of the Laplace distribution to design a distributed noise addition scheme to perturb the locally shared data and use Paillier encryption to transmit the locally shared data to the semitrusted curator. Then, the semitrusted curator obtains the estimator of the covariance matrix of the aggregated data with Laplace noise and then obtains the principal components of the aggregated data and returns them to each data owner. Finally, the data owner utilizes the generative model of probabilistic principal component analysis to generate a synthetic data set for publication. We conducted experiments on different real data sets; the experimental results demonstrate that the synthetic data set released by the HPDP-DP method can maintain high utility.

1. Introduction

The ability of people to collect and analyze data is gradually improving with the development of the artificial intelligence. Sometimes the data are stored by different sites(data owners), and each site holds a smaller number of samples. For example, in Figure 1, there are three hospitals, the patients in each hospital are different from each other, but the data features of each patient are the same. In order to better mine the useful information behind the data, a large number of samples are needed. Pooling data in one central location enables efficient data analysis and mining, but data contain sensitive privacy; directly sharing or pooling the data will lead to privacy leakage [1, 2], which prevents people from sharing data. That is to say, data are facing serious privacy leakage risks in the process of data sharing, network transmission, and storage [3]. It is important to protect the privacy of shared data and weigh the security and availability of data [4, 5]. Therefore, it is desirable to propose an efficient distributed algorithm, which can provide the utility close to the centralized case and protect the privacy of data. In recent years, there have been some researches on privacy-preserving data publishing and sharing, for example, the *k*anonymity [6] technology, the encryption techniques, such as lattice-based cryptography [7] and quantum cryptography [8, 9]. The differential privacy [10] has been widely used for privacy-preserving data publishing; privacy-preserving data publishing based on differential privacy has become a research hot spot [11–15].

However, there are still some challenges when using the differential privacy technique to protect the privacy of the published data. One is that the data are stored by different data owners; directly pooling and publishing the data will lead to privacy leakage. When data are stored by multiple data owners, as the number of data owners increases, if differential privacy is used independently to add noise to the locally shared data, the utility of the published data will be reduced. In view of this, we propose a horizontally partitioned data publication approach with differential privacy. We make the following contributions:

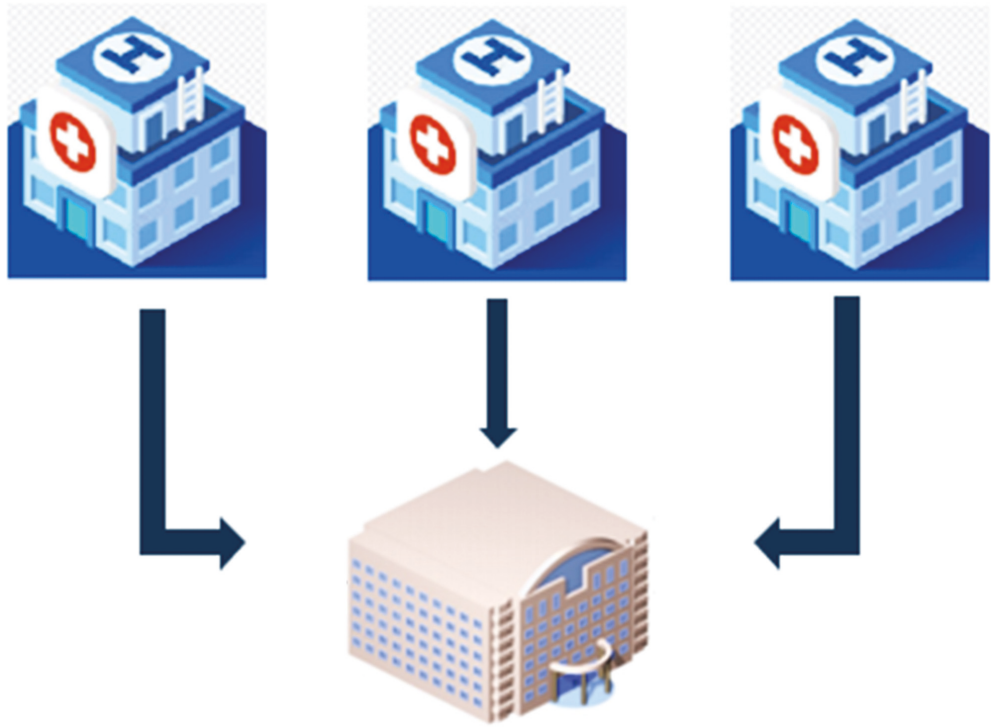

Figure 1: Aggregate data from different hospitals.

(1) We propose a method for horizontally partitioned data publication with differential privacy (HPDP-DP). In a distributed environment, data are owned by multiple parties. We use the weighted average of the noised covariance matrices of the local data to estimate the covariance matrix of the pooled data. The data owners and a semitrusted curator collaborate to get the principal components of the pooled data and generate a synthetic data set for publishing.

(2) In the distributed scenario, in order to make the noise level of the aggregated data the same as in the centralized scenario, the HPDP-DP method utilizes the infinite divisibility of the Laplace distribution and Paillier homomorphic encryption to alleviate the effects of noise and can achieve the same noise level as the centralized scenario.

(3) We evaluate the performance of HPDP-DP method through experiments on real data sets, and the experimental results show that HPDP-DP method can generate synthetic data with high efficiency.

2. Related Work

In this section, we introduce the research status of privacy-preserving data release based on differential privacy in the centralized and distributed scenarios, respectively.

2.1. Privacy-Preserving Data Publishing in Centralized Environment. In recent years, there are many researches on privacy-preserving data publishing based on differential privacy. Jiang et al. [16] proposed a method that adding Laplace noise to the covariance matrix and the projection matrix and then using the noisy projection matrix to restore and generate the synthetic data set for publishing. Zhang et al. proposed the PrivBayes method in [17]; they used the relationship between the features to build a Bayesian network. They added Laplace noise to the low-dimensional marginal distribution to make the Bayesian network satisfy differential privacy, and then they used the Bayesian network to generated a synthetic data set for publishing. Chen et al. proposed the Jtree method in [18]. First, they proposed a sampling-based testing framework that is used to explore pairwise dependencies while satisfying differential privacy. Then, they applied the connection tree algorithm to construct an inference mechanism to infer the joint data distribution. Finally, they efficiently generated a synthetic data set by using the noise margin table and inference model. Xu et al. [19] proposed DPPro scheme; they released high-dimensional data by using randomly projected. They projected the original high-dimensional data into a randomly selected low-dimensional subspace and added noise to the low-dimensional projected data. They theoretically demonstrated that the data published by the DPPro method have similar squared Euclidean distances to the original data. In order to solve the problem of dimensional disaster in high-dimensional data publishing, Zhang et al. [20] presented the PrivHD method with the junction tree. First, they used exponential mechanism to construct a Markov network; in order to reduce the candidate space, high-pass filtering technique is used in sampling. Then, they used the maximum spanning tree method to build a better joint tree. At last, a high-dimensional synthetic data set is generated for publication. Zhang et al. [21] presented the PrivMN method.

They first constructed a Markov model to express the relationship of features. Then, they used the Laplace mechanism to add noise to the marginal distribution to generate the noisy marginal distribution table. Finally, they used the noisy marginal distribution to generate a synthetic data set for publishing. Gu et al. [22] proposed the PPCA-DP method; they first used the principal component analysis to reduce the dimensionality of high-dimensional data and then added Laplace noise to the low-dimensional projection data; finally, they used the generative model of probabilistic principal component analysis to generate a synthetic data set for publishing. The above are all studies on privacy-preserving data publishing in centralized scenarios.

2.2. Privacy-Preserving Data Publishing in Distributed Environment. At present, most of the existing privacy-preserving data publishing works focus on the centralized scenario; there are fewer studies on privacy-preserving data publishing in distributed scenario. The multiparty data release scenario studied in this paper is that each data owner owns a data set and uses the differential privacy technology to protect the privacy of the local data set rather than the scenario that multiple individuals keep their data locally. The latter typically utilize the local differential privacy [23] techniques to protect the privacy of individual data [24, 25]. In the following, we will introduce the research status of privacy-preserving data release in multiparty data release, where each data owner owns a data set.

Alhadidi et al. [26] proposed the first noninteractive two-party horizontally partitioned data publication method that satisfies differential privacy and secure multiparty computation. The data set published by this method is suitable for classification tasks. Hong et al. [27] constructed the framework (CELS protocol) that enables distributed parties to securely generate outputs while satisfying differential privacy. The security and differential privacy guarantees of the protocol are proved. Ge et al. [28] presented the DPS-PCA algorithm. Data owners collaborated to compute the principal components while protecting the privacy of data. The DPS-PCA algorithm can trade off the relationship between the accuracy of estimating principal components and the degree of privacy protection, but this method only outputs a low-dimensional subspace of high-dimensional sparse data. An efficient and scalable distributed PCA protocol is proposed by Wang et al. [29] for the computation of principal components of split horizon data in a distributed environment. First, the shared data are encrypted and sent to a semitrusted third party. Second, the shared data are aggregated by a semitrusted third party, and the aggregated result is sent to the data consumer. Finally, the data consumer performed a principal component analysis and obtained the principal components of the pooled data. Cheng et al. [30] presented the DP-SUBN^3 approach; the data owners built a Bayesian network with the assistance of a semitrusted curator, and then the Bayesian network is used to generate a synthetic data set. In DP-SUBN^3 approach, the four stages of correlation quantification, structure initialization, structure update, and parameter learning all need to access the local data set, and each stage satisfies differential privacy, which in turn makes the DP-SUBN^3 approach satisfy differential privacy. For the privacy protection of data publishing in arbitrary partitions between two parties, Wang et al. [31] presented the first distributed algorithm, which generates anonymous data from two parties. In order to prevent both parties from leaking private information, the anonymization process satisfies both differential privacy and secure two-party computation. Gu et al. [32] presented the PPCA-DP-MH approach. The data owners collaborate with a semitrusted curator to reduce the dimensionality of the data, and then the data owners used the probabilistic generative model of principal component analysis to generate a published data set. In the PPCA-DP-MH method, since multiple data owners add noise to the data locally and independently, the utility of publishing data gradually decreases as the number of data owners increases. In response to this challenge, we propose the HPDP-DP method in this paper. We design the generation and addition scheme of correlated noise, so that the utility of publishing data will not decrease with the increase of data owners, and even the utility of publishing data will gradually increase with the increase of data owners.

3. Preliminaries

3.1. Probabilistic Principal Component Analysis (PPCA). Principal component analysis is one of the commonly used dimensionality reduction methods. Principal component analysis is a statistical analysis method that converts multiple variables into a few hidden variables through dimensionality reduction techniques. These fewer low-dimensional and not correlated hidden variables are also called principal components. The principal components can reflect most of the information of the original variables. Next, the main process of finding principal components is introduced. First, computing the covariance matrix Σ of the data. Then perform eigenvalue decomposition on the covariance matrix Σ, $\Sigma = U\Lambda U^T$, where Λ is a diagonal matrix and the elements on the diagonal are the eigenvalues of the matrix Σ, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$. The corresponding eigenvectors are as follows: $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p$ which are called the principal components. U is an orthogonal matrix consisting of the eigenvectors. Usually, the top k principal components retained are determined by the cumulative contribution rate $c = \sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{p} \lambda_i$.

However, Michael et al. [33] proposed that the principal component analysis (PCA) is a nongenerative model, they presented that the principal component analysis (PCA) also has a generative model called probabilistic principal component analysis (PPCA). The most common model to associate low-dimensional latent variables with high-dimensional observable variables is the factor analysis model, i.e. $\mathbf{x} = W\mathbf{s} + \mu + \xi$, where $\mathbf{x}$ is p -dimensional observation vector consisting of the p original variables, $\mathbf{s}$ is a k -dimensional vector consisting of k latent variables, $\xi \sim N(\mathbf{0}, \Psi)$, the matrix W associates the vector $\mathbf{x}$ with the vector $\mathbf{s}$. The vector μ allows the model to have a nonzero mean vector.

Theorem 1 [33]. *From Figure 2 and the latent variable model* $\mathbf{x} = W\mathbf{s} + \mu + \xi$, *when* $\xi \sim N(\mathbf{0}, \sigma^2 I)$, $\mathbf{s} \sim N(\mathbf{0}, I_k)$, *then* $\mathbf{x}|\mathbf{s} \sim N(W\mathbf{s} + \mu, \sigma^2 I_p), \sigma > 0, W \in R^{p\times k}$, *where the maximum likelihood estimation of* μ, σ^2 , *and* W *are*

$$\begin{aligned} \widehat{\boldsymbol{\mu}} &= \tilde{\boldsymbol{\mu}}, \\ \widehat{\sigma}^2 &= \frac{1}{p-k}\sum_{i=k+1}^{p}\lambda_i, \\ \widehat{W} &= U_k\left(\Lambda_k - \widehat{\sigma}^2 I\right)\frac{1}{2}, \end{aligned} \tag{1}$$

where $\tilde{\mu}$ *is the mean vector, the column vectors in* U_k *is the eigenvectors corresponding to the top* k *eigenvalues of the covariance matrix.*

3.2. Differential Privacy. Differential privacy is a strong privacy protection model independent of background knowledge. If the output of a privacy-preserving algorithm is insensitive to small changes in the input, the algorithm satisfies differential privacy. The essence of differential privacy is to randomly perturb the query results, so that people cannot infer the original input information based on the query results.

Definition 1 (Differential Privacy) [10]. A random algorithm $\mathcal{M}$ satisfies ϵ differential privacy, if for any two neighboring data sets $D, \widehat{D}$ (only one record differs between the two data sets) and for any $S(S \in Rang(M))$ there is

$$\left|\ln\frac{P_r\{\mathcal{M}(D) \in S\}}{P_r\left\{\mathcal{M}(\widehat{D}) \in S\right\}}\right| \leq \varepsilon, \tag{2}$$

ε is a small positive real number, which is also called privacy budget.

In the Definition 1, ε is used for controlling the probability ratio of the random algorithm $\mathcal{M}$ to obtain the same output on the two neighboring data sets D and $\widehat{D}$; it reflects the level of privacy protection that the algorithm $\mathcal{M}$ can provide.

Definition 2 (Sensitivity). [10]. Let f be a function that maps a data set into a fixed size vector of real numbers, $f: D \longrightarrow R^d$, for any neighboring data sets D and $\widehat{D}$, the sensitivity of f is defined as follows:

$$\Delta f = \max_{D,\widehat{D}} \|f(D) - f(\widehat{D})\|_1, \tag{3}$$

where $\|\cdot\|_1$ denotes the L_1 norm.

Definition 3 (Laplace mechanism). [34]. For any function $f: D \longrightarrow R^d$, if the random algorithm $\mathcal{M}$ satisfies the equation:

$$\mathcal{M}(D) = f(D) + \left(Lap_1\left(\frac{\Delta f}{\varepsilon}\right), \ldots, Lap_d\left(\frac{\Delta f}{\varepsilon}\right)\right), \tag{4}$$

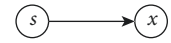

Figure 2: Graphical model for principal component analysis.

then the algorithm $\mathcal{M}$ satisfies ε differential privacy, $\text{Lap}_1(\Delta f/\varepsilon), \ldots, \text{Lap}_d(\Delta f/\varepsilon)$ are independent Laplace random variables.

Theorem 2 [35]. *Let* $Y \sim$ *Laplace* (λ), *then, the distribution of* Y *is infinitely divisible. Furthermore, for every integer* $M \geq 1$, $Y = \sum_{m=1}^{M}(Y_{1m} - Y_{2m})$, *where* Y_{1m} *and* Y_{2m} *are i.i.d. with the Gamma density* $f(x) = ((1/\lambda)^{(1/n)}/\Gamma(1/n))x^{(1/n)-1}e^{-(x/\lambda)}$, $x \geq 0$.

Theorem 3 (Sequential Composition). [34]. *Let* $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n$ *be a series of privacy algorithms, and their privacy budgets are* $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$, *for the same data set* D, *the combined algorithm* $\mathcal{M}(\mathcal{M}_1(D), \mathcal{M}_2(D), \ldots, \mathcal{M}_n(D))$ *provides* $\sum_{i=1}^{n}\varepsilon_i$ *differential privacy.*

Theorem 4 (Parallel Composition). [34]. *Let* $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n$ *be a series of privacy algorithms, which privacy budgets are* $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$, $D_1, D_2, \ldots, D_n$ *are disjoint data sets, the combined algorithm* $\mathcal{M}(\mathcal{M}_1(D_1), \mathcal{M}_2(D_2), \ldots, \mathcal{M}_n(D_n))$ *provides* $\max_{1\leq i\leq n}\varepsilon_i$ *differential privacy.*

3.3. Paillier Encryption and Decryption. In this paper, we use Paillier encryption scheme [36] to encrypt the local shared data before being aggregated. The Paillier encryption scheme is described as follows:

(1) Key generation: $n = pq$, where p and q are large primes, $\lambda = \text{lcm}(p-1, q-1)$. Euler function $\Phi(n) = (p-1)(q-1)$, $g \in Z_{n^2}^*$, the (n, g) is public key and λ is private key.

(2) Encryption: plaintext $m < n$, randomly select $r < n$, ciphertext $c = g^m \cdot r^n \bmod n^2$.

(3) Decryption: ciphertext $c < n^2$, plaintext $m = (L(c^\lambda \bmod n^2)/L(g^\lambda \bmod n^2)) \bmod n$, where $L(u) = (u-1)/n$.

Paillier encryption is additively homomorphic. We use $[[m]]$ to represent the encrypted ciphertext of m. Then, $\forall m_1, m_2 \in Z_n, k \in N$, $[[m_1]] \cdot [[m_2]] = [[m_1 + m_2]]$ and $[[m]]^k = [[k \cdot m]]$.

4. The HPDP-DP Method

4.1. Problem Statement. There exist $M(M \geq 2)$ data owners, the m-th data owner P_m holds a local data set denoted as $X_m = \left\{\mathbf{x}_1^m, \mathbf{x}_2^m, \ldots, \mathbf{x}_{N_m}^m\right\}$, N_m is the number of individuals owned by data owner $P_m, m = 1, \ldots, M$, $N = \sum_{m=1}^{M} N_m$. Each individual is a p-dimensional vector. The data sets $X_1, X_2, \ldots, X_M$ can be viewed as horizontally split the integrated data set $X = \cup_{m=1}^{M} X_m$ by M data owners. That is all the local data sets have the same attributes and do not intersect with each other. Our goal is to design an algorithm that can publish these horizontally partitioned data sets privately; specifically, it is that with the assistance of a

Input: Data sets X_m, $m = 1, 2, \ldots, M$. Private key λ, public key (n, g). θ_m $(m = 0, 1, 2, \ldots, M)$, where $\theta_0 \cdot \theta_1 \cdot \cdots \cdot \theta_M = 1$. Privacy budget ε and cumulative contribution rate c

Output: Synthetic data set $\widetilde{X} = \cup_{m=1}^{M} \widetilde{X}_m$

(1) **for** $m = 1$ to M **do**
(2) Data owner generates $p \times p$ noise matrices $B_{m1} = (b_{ij}^{m1})_{p\times p}$ and $B_{m2} = (b_{ij}^{m2})_{p\times p}$, let B_{m1} and B_{m2} be the symmetric matrix with the upper triangle (including the diagonal) entries are sampled from Gamma $(1/M, p + p^2/M\varepsilon)$, and set $b_{ji}^{m1} = b_{ij}^{m1}, b_{ji}^{m2} = b_{ij}^{m2}, \forall i < j$.
(3) Compute: $L_m = (l_{ij}^m)_{p\times p} = \sum_{k=1}^{N_m} (\mathbf{x}_k^m - \mu^m)(\mathbf{x}_k^m - \mu^m)^T$
(4) Compute: $\widetilde{L}_m = (\widetilde{l}_{ij}^m)_{p\times p} = (l_{ij}^m + b_{ij}^{m1} - b_{ij}^{m2})_{p\times p}$
(5) **for** $i = 1$ to p **do**
(6) **for** $j = 1$ to p **do**
(7) Compute: $\theta_m \cdot [[\widetilde{l}_{ij}^m]] \leftarrow \theta_m \cdot g^{l_{ij}^m + b_{ij}^{m1} - b_{ij}^{m2}} \cdot r^n \bmod n^2$
(8) **end for**
(9) **end for**
(10) **end for**
(11) **return** $C_m = (\theta_m \cdot [[\widetilde{l}_{ij}^m]])_{p\times p}, m = 1, 2, \ldots, M$
(12) Compute the Hadamard product: $C \leftarrow (\theta_0)_{p\times p} \circ C_1 \circ C_2 \circ \cdots \circ C_M$
(13) Decrypt C: $\widetilde{L} = (\sum_{m=1}^{M} (l_{ij}^m + b_{ij}^{m1} - b_{ij}^{m2}))_{p\times p} \leftarrow C$
(14) Compute: $\Sigma = (1/N)\widetilde{L}$
(15) Eigenvalue decomposition of matrix Σ, return eigenvalues in descending order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$, and corresponding eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p$
(16) **for** $k = 1$ to p **do**
(17) **if** $\sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{p} \lambda_i \geq c$ **then**
(18) $\Lambda_k = (\lambda_1, \lambda_2, \ldots, \lambda_k)$
(19) $U_k = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k)$
(20) **end if**
(21) **end for**
(22) **return** Λ_k, U_k
(23) **for** $m = 1$ to M **do**
(24) Compute $S_m = X_m \times U_k$
(25) Use the model defined in Theorem 1 to generate a synthetic data set $\widetilde{X}_m$
(26) **end for**
(27) **return** $\widetilde{X} = \cup_{m=1}^{M} \widetilde{X}_m$

Algorithm 1: HPDP-DP algorithm.

semitrusted curator, the M data owners and the curator collaborate to publish a synthetic data set $\widetilde{X} = \cup_{m=1}^{M} \widetilde{X}_m$, which has the same scale and statistical properties as the data set $X = \cup_{m=1}^{M} X_m$. Typically, we assume that the data owners and the curator are honest-but-curious, that is, they will follow the protocol but try to find out as much secret information as possible.

In view of the above scenario, we propose a horizontally partitioned data publishing method with differential privacy (HPDP-DP). The Algorithm 1 depicts the HPDP-DP algorithm. First, the data owner perturbs the local scatter matrix with random noise that obeys the Gamma distribution and sends it to the semitrusted curator. Then the semitrusted curator aggregates all the local scatter matrices to get the noisy estimator of the covariance matrix of the pooled data. The semitrusted curator performs eigenvalue decomposition on the covariance matrix to get the principal components and then the top k principal components are sent to each data owner. At last, each data owner uses the top k principal components and the generative model of probabilistic principal component analysis to generate a synthetic data set.

In order to reduce the impact of noise on the availability of published data, the HPDP-DP algorithm employs a distributed Laplace mechanism to add noise to the local scatter matrix. According to Theorem 2, the infinite additivity of Laplace distribution, we perturb the local scatter matrix with the noise follows a Gamma distribution, which makes the estimator of the covariance matrix of the pooled data contain the same level of noise as the centralized scene. Inspired by [37], since the step of perturbing the local scatter matrix with gamma-distributed noise does not satisfy differential privacy, we will use the Paillier encryption scheme to encrypt the perturbed scatter matrix to protect the privacy of local data. The HPDP-DP algorithm mainly consists of the following stages.

Initialization phase: in the initialization phase, the Paillier cryptographic system generates the public key (n, g) and the private key λ. The system also generates $M + 1$ factors $\theta_0, \theta_1, \ldots, \theta_M$, where $\theta_m \in Z_{n2}, m = 0, 1, 2, \ldots, M$ and $\theta_0 \cdot \theta_1 \cdot \cdots \cdot \theta_M = 1$. The factor θ_0 and the private key λ are secretly sent to the curator. The public key (n, g) and θ_m are secretly sent to the data owner P_m, $m = 1, 2, \ldots, M$.

Perturbation and encryption phase: each data owner randomly perturbs the local scatter matrix. The scatter matrix of the data owner P_m is given by

$$L_m = \left(l_{ij}^m\right)_{p\times p} = \sum_{k=1}^{N_m} \left(\mathbf{x}_k^m - \mu^m\right)\left(\mathbf{x}_k^m - \mu^m\right)^T = \sum_{k=1}^{N_m} \mathbf{x}_k^m \left(\mathbf{x}_k^m\right)^T - N_m \mu^m \left(\mu^m\right)^T. \tag{5}$$

where $\mu^m = (1/N_m)\sum_{k=1}^{N_m} \mathbf{x}_k^m$.

The data owner P_m generates two $p \times p$ symmetric random matrices $B_{m1} = (b_{ij}^{m1})_{p\times p}$ and $B_{m2} = (b_{ij}^{m2})_{p\times p}$; b_{ij}^{m1} and b_{ij}^{m2} are sampled from Gamma$((1/M), (p + p^2)/M\varepsilon)$, $1 \le i \le j \le p$. Then, the local noisy scatter matrix is $\widetilde{L}_m = (\widetilde{l}_{ij}^m)_{p\times p} = (l_{ij}^m + b_{ij}^{m1} - b_{ij}^{m2})_{p\times p}$. Using the public key (n, g) and θ_m to encrypt each element of $\widetilde{L}_m$ to get the encrypted matrix $C_m = (\theta_m[[\widetilde{l}_{ij}^m]])_{p\times p} = (\theta_m \cdot g^{l_{ij}^m + b_{ij}^{m1} - b_{ij}^{m2}} \cdot r^n \cdot \mathrm{mod} n^2)_{p\times p}$ which will be sent to the curator, $m = 1, 2, \ldots, M$.

Aggregation and decryption phase: After receiving these encrypted matrices $C_1, C_2, \ldots, C_M$, the curator performs the Hadamard product on these encrypted matrices. We use the symbol $\circ$ as the Hadamard product of matrices.

$$\begin{aligned}
(\theta_0)_{p\times p} \circ C_1 \circ C_2 \circ \cdots \circ C_M &= \left(\theta_0 \prod_{m=1}^{M} \theta_m \cdot g^{l_{ij}^m + b_{ij}^{m1} - b_{ij}^{m2}} \cdot r^n \cdot \mathrm{mod} n^2\right)_{p\times p} \\
&= \left(\prod_{m=1}^{M} \cdot g^{l_{ij}^m + b_{ij}^{m1} - b_{ij}^{m2}} \cdot r^n \cdot \mathrm{mod} n^2\right)_{p\times p} \\
&= \left(g^{\sum_{m=1} M l_{ij}^m + \sum_{m=1}^{M}\left(b_{ij}^{m1} - b_{ij}^{m2}\right)} \cdot r^{Mn} \cdot \mathrm{mod} n^2\right)_{p\times p} \\
&= \left(g^{\sum_{m=1} M l_{ij}^m + \mathrm{Lap}\left(p + p^2/\varepsilon\right)} \cdot r^{Mn} \cdot \mathrm{mod} n^2\right)_{p\times p} \\
&= \left(\left[\left[\sum_{m=1}^{M} l_{ij}^m + \mathrm{Lap}\left(\frac{p + p^2}{\varepsilon}\right)\right]\right]\right)_{p\times p},
\end{aligned} \tag{6}$$

where $\sum_{m=1}^{M}(b_{ij}^{m1} - b_{ij}^{m2}) \sim \mathrm{Lap}((p + p^2)/\varepsilon)$ holds due to Theorem 2. The curator decrypts the above results to get the sum of local scatter matrices with Laplace noise $\widetilde{L} = \sum_{m=1}^{M} \widetilde{L}_m = \sum_{m=1}^{M} L_m + (\mathrm{Lap}((p + p^2)/\varepsilon))_{p\times p}$, which is used as an estimation of the scatter matrix of the pooled data, and then the estimation of the covariance matrix of the pooled data is $\Sigma = (1/N)\widetilde{L}$.

In this stage, our idea is to use the weighted average of the local covariance matrices to estimate the covariance matrix of the pooled data. Assuming that the covariance matrix of data owner P_m is $\widetilde{\Sigma}_m$, the relationship with the scatter matrix is $\widetilde{\Sigma}_m = (\widetilde{L}_m/N_m)$, and then the estimation of the covariance matrix of the pooled data is $\Sigma = \sum_{m=1}^{M}(N_m/N)\widetilde{\Sigma}_m = (1/N)\sum_{m=1}^{M}\widetilde{L}_m = (1/N)\widetilde{L}$.

Principal component analysis phase: the curator performs eigenvalue decomposition on matrix Σ . The curator gets the eigenvectors (the top k principal components) $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ and then sends them to each data owner.

Generate synthetic data set phase: Each data owner uses the returned top k principal components and the generative model of probabilistic principal component analysis in Theorem 1 to generate a synthetic data set.

4.2. Analysis

4.2.1. Security Analysis

Theorem 5. *The data set owned by P_m is X_m and its corresponding scatter matrix is $L_m = (l_{ij}^m)_{p\times p}$, $m = 1, 2, \ldots, M$. Defining the query function,*

$$f(X_1, X_2, \ldots, X_M) = \sum_{m=1}^{M} L_m, \tag{7}$$

the output result intended to be protected. $B_{m1} = (b_{ij}^{m1})_{p\times p}$ and $B_{m2} = (b_{ij}^{m2})_{p\times p}$ are symmetric random matrices will be added to L_m, b_{ij}^{m1} and b_{ij}^{m2} are sampled from Gamma$((1/M), (p + p^2/M\varepsilon))$, $1 \le i \le j \le p$. If the random algorithm $\mathscr{M}$ holds

$$\mathscr{M}(X_1, X_2, \ldots, X_M) = f(X_1, X_2, \ldots, X_M) + \sum_{m=1}^{M}(B_{m1} - B_{m2}), \tag{8}$$

then the algorithm $\mathscr{M}$ satisfies ε differential privacy.

Proof. According to Theorem 2, it can be known each element of $\sum_{t=1}^{M}(B_{m1} - B_{m2})$ obeys $\mathrm{Lap}(p(1+p)/\varepsilon)$. So, next we will prove if algorithm $\mathscr{M}$ holds

$$\mathscr{M}(X_1, X_2, \ldots, X_M) = f(X_1, X_2, \ldots, X_M) + B, \tag{9}$$

$B = (b_{ij})_{p\times p}$ is a symmetric random matrix and b_{ij} is sampled from $\mathrm{Lap}(p(1+p)/\varepsilon)$, $1 \le i \le j \le p$, then the algorithm $\mathscr{M}$ satisfies ε differential privacy.

We denote the two neighboring data sets as $X = \cup_{m=1}^{M} X_m$ and $\widehat{X} = \cup_{m=1}^{M} \widehat{X}_m$; there is only one individual is different, without losing general assumption, suppose the different individuals are in X_M and $\widehat{X}_M$. We denote the only two different individuals as $\mathbf{x}_{N_M}^{M} \in X_M$ and $\widehat{\mathbf{x}}_{N_M}^{M} \in \widehat{X}_M$. Assume that all individual data have been normalized to the [0,1] interval. The estimation of the scatter matrices of X and $\widehat{X}$ are as follows:

$$L = \sum_{m=1}^{M} L_m = \sum_{m=1}^{M} \left(l_{ij}^{m}\right)_{p\times p}, \tag{10}$$

and

$$\widehat{L} = \sum_{m=1}^{M-1} L_m + \widehat{L}_M = \sum_{m=1}^{M-1} \left(l_{ij}^{m}\right)_{p\times p} + \left(\widehat{l}_{ij}^{M}\right)_{p\times p}. \tag{11}$$

Let $B = (b_{ij})_{p\times p}$ and $\widehat{B} = (\widehat{b}_{ij})_{p\times p}$ be two independent symmetric random matrices, where b_{ij} and $\widehat{b}_{ij}$ are sampled from $\mathrm{Lap}(p(1+p)/\varepsilon)$, $1 \le i \le j \le p$.

Let $S = L + B$ and $\widehat{S} = \widehat{L} + \widehat{B}$, then the log ratio of the probabilities of S and $\widehat{S}$ at a point H is given by

$$\left|\ln \frac{P\{H|X\}}{P\{H|\widehat{X}\}}\right| = \left|\ln \frac{P\{H - L|X\}}{P\{H - \widehat{L}|\widehat{X}\}}\right|. \tag{12}$$

According to the definition of differential privacy (Definition 1), we need to prove that the following inequalities holds:

$$\left|\ln \frac{P\{H|X\}}{P\{H|\widehat{X}\}}\right| = \left|\ln \frac{P\{H - L|X\}}{P\{H - \widehat{L}|\widehat{X}\}}\right| \le \varepsilon. \tag{13}$$

The mean vectors of X_M and $\widehat{X}_M$ are as follows:

$$\boldsymbol{\mu}^{M} = \frac{1}{N_M} \sum_{k=1}^{N_M} \mathbf{x}_k^{M}, \tag{14}$$

and

$$\widehat{\boldsymbol{\mu}}^{M} = \frac{1}{N_M} \left(\sum_{k=1}^{N_M - 1} \mathbf{x}_k^{M} + \widehat{\mathbf{x}}_{N_M}^{M} \right), \tag{15}$$

so $\widehat{\mu}^{M} = \mu^{M} + (1/N_M)(\widehat{x}_{N_M}^{M} - \mathbf{x}_{N_M}^{M})$. Hence, we have the following:

$$\begin{aligned}
\left|l_{ij}^{M} - \widehat{l}_{ij}^{M}\right| &= \left|\sum_{k=1}^{N_M} x_{ik}^{M} x_{jk}^{M} - N_M \mu_i^{M} \mu_j^{M} - \left(\sum_{k=1}^{N_M - 1} x_{ik}^{M} x_{jk}^{M} + \widehat{x}_{iN_M}^{M} \widehat{x}_{jN_M}^{M} - N_M \widehat{\mu}_i^{M} \widehat{\mu}_j^{M} \right)\right| \\
&= \left| x_{iN_M}^{M} x_{jN_M}^{M} - \widehat{x}_{iN_M}^{M} \widehat{x}_{jN_M}^{M} + N_M \left(\widehat{\mu}_i^{M} \widehat{\mu}_j^{M} - \mu_i^{M} \mu_j^{M}\right)\right| \\
&= \left| x_{iN_M}^{M} x_{jN_M}^{M} - \widehat{x}_{iN_M}^{M} \widehat{x}_{jN_M}^{M} + \mu_i^{M} \left(\widehat{x}_{jN_M}^{M} - x_{jN_M}^{M}\right) + \mu_j^{M} \left(\widehat{x}_{iN_M}^{M} - x_{iN_M}^{M}\right) + \frac{1}{N_M} \left(\widehat{x}_{iN_M}^{M} - x_{iN_M}^{M}\right)\left(\widehat{x}_{jN_M}^{M} - x_{jN_M}^{M}\right)\right| \\
&= \left|\left(x_{iN_M}^{M} - \widehat{x}_{iN_M}^{M}\right)\left(x_{jN_M}^{M} - \mu_j^{M}\right) + \left(x_{jN_M}^{M} - \widehat{x}_{jN_M}^{M}\right)\left(\widehat{x}_{iN_M}^{M} - \mu_i^{M}\right)\right| \\
&\le \left|\left(x_{iN_M}^{M} - \widehat{x}_{iN_M}^{M}\right)\left(x_{jN_M}^{M} - \mu_j^{M}\right)\right| + \left|\left(x_{jN_M}^{M} - \widehat{x}_{jN_M}^{M}\right)\left(\widehat{x}_{iN_M}^{M} - \mu_i^{M}\right)\right| \le 2.
\end{aligned} \tag{16}$$

Therefore, the following formula holds:

$$\begin{aligned}
\left|\ln \frac{P\{H|X\}}{P\{H|\widehat{X}\}}\right| &= \left|\ln \frac{P\{H - L|X\}}{P\{H - \widehat{L}|\widehat{X}\}}\right| \\
&= \frac{\varepsilon}{p(1+p)} \sum_{1\le i\le j\le p} \left(\left|h_{ij} - \widehat{l}_{ij}^{M}\right| - \left|h_{ij} - l_{ij}^{M}\right| \right) \\
&\le \frac{\varepsilon}{p(1+p)} \sum_{1\le i\le j\le p} \left|l_{ij}^{M} - \widehat{l}_{ij}^{M}\right| \\
&\le \frac{\varepsilon}{p(1+p)} p(1+p) = \varepsilon.
\end{aligned} \tag{17}$$

So the conclusion of Theorem 5 holds.

Security against external attacks: external attacker will eavesdrop on data sent by local data owners to the curator. According to the semantic security of Paillier encryption against plaintext attacks, external attacker unable to decrypt data $(\theta_m \cdot g^{l_{ij}^{m} + b_{ij}^{m1} - b_{ij}^{m2}} \cdot r^{n} \cdot \mathrm{mod} n^2)_{p\times p}$ without knowing private key λ and θ_m, $1 \le m \le M$. External attacker may also eavesdrop on the aggregated value of the data owners $(g^{\sum_{m=1}^{M} l_{ij}^{m} + b_{ij}^{m1} - b_{ij}^{m2}} \cdot r^{Mn} \cdot \mathrm{mod}\, n^2)_{p\times p}$, external attacker unable to decrypt data without knowing private key λ. Even though the external attacker get the sum of scatter matrices with noise $(\sum_{m=1}^{M} l_{ij}^{m} + b_{ij}^{m1} - b_{ij}^{m2})_{p\times p}$, because it contains Laplace noise, so the local data are still safe according to Theorem 5. *Security against internal attacks*: internal adversaries are data owners and the curator. The data owner P_m holds θ_m

secretly, the rest of the data owners and the curator cannot decrypt $(\theta_m \cdot g^{l_{ij}^m + b_{ij}^{m1} - b_{ij}^{m2}} \cdot r^n \cdot \mathrm{mod} n^2)_{p\times p}$ without private key λ and θ_m unless the curator colluded with the $M-1$ data owners. The curator can use private key λ and θ_0 to decrypt the aggregated value $(\prod_{m=1}^{M} (\theta_m \cdot g^{l_{ij}^m + b_{ij}^{m1} - b_{ij}^{m2}} \cdot r^n \cdot \mathrm{mod} n^2)_{p\times p}$, but the curator can only get the aggregated value with Laplace noise, so the local data are safe according to Theorem 5. □

4.2.2. Complexity Analysis. Computation time cost analysis: the total time complexity of Algorithm 1 is $O(Mp^2 + Mn)$, where M is the number of data owners, p is the number of attributes, $n = n_1 + n_2 + \cdots + n_M$, n_m is the number of samples owned by data owner P_m, $m = 1, 2, \cdot, M$. It is due to the following facts. In Algorithm 1, the major computational cost of Algorithm 1 is reflected in lines 1–11, lines 16–21, and lines 23–26. The lines $1-11$ are to perturb and encrypt the scatter matrix of the local data of the M data owners, and the time complexity is $O(Mp^2)$. The lines $16-21$ are to perform principal component analysis on the aggregated scatter matrix, and its time complexity is $O(K)$, where K is the number of retained principal components, which is proportional to p, so the complexity is $O(p)$. The lines $23-26$ are that each data owner uses Theorem 1 to generate a published data set, and the time complexity is $O(Mn_1 + Mn_2 + \cdot + Mn_M) = O(Mn)$. In summary, the time complexity of Algorithm 1 is $O(Mp^2 + p + Mn)$, which is $O(Mp^2 + Mn)$.

Communication cost analysis. There exist three stages that incur communication costs. The first stage is the M data owners send the local scatter matrix to the curator, the size of the message sent by each data owner is p^2, the total size of the message sent in this stage is Mp^2. The second stage is the curator sends the top K eigenvalues and their corresponding eigenvectors to each data owner; the total size of the message sent in this stage is MpK^2. The third stage is each data owner sends the synthetic data set to the curator; the size of the message sent by data owner P_m is $n_m p, m = 1, 2, \ldots, M$; the total size of the message sent during this stage is $np = (n_1 + n_2 + \cdots + n_M)p$.

5. Experiment

In this section, we experimentally evaluate the performance of HPDP-DP algorithm by comparing with the DP-SUBN^3 algorithm [30]. We conduct experiments on different real data sets that are NLTCS [38] and Adult [39] data sets. NLTCS data set contains 21574 individuals, each individual has 16 attributes. Adult data set contains 45222 individuals, each individual has 15 attributes. We use the method in [30] to preprocess the Adult data set. After processing, the number of attributes in the Adult data set is 52. We use SVM classification accuracy to evaluate the performance of HPDP-DP algorithm. We train multiple classifiers on published synthetic data sets. For NLTCS data set, predicting whether a person is unable to go outside and whether a person is unable to manage money. For Adult data set, predicting whether a person holds a postsecondary degree and whether a person earns more than $50K$. In each classification task, we use 20% of the individuals as the test set and 80% of the individuals as the training set. Each experiment is run five times, and the average results are reported. The number of retained principal components is determined by the cumulative contribution rate c. The cumulative contribution rate c is set to 0.8 for NLTCS data set and 0.95 for Adult data set. In order to measure the performance of the HPDP-DP algorithm more clearly, the same SVM classifier are trained on the original data set; we label the SVM classification accuracy on the original data set with "No Privacy."

5.1. The Impact of the Number of Principal Components Retained on the SVM Classification Accuracy. In this section, we train multiple classifiers to study the influence of the number of principal components retained on the SVM classification accuracy. In this set of experiments, the number of data owners is set to 3; the privacy budget ε is set to 0.5.

For the Adult data set, Figures 3(a) and 3(c) show the cumulative contribution rate and individual contribution rate of the principal components. Because there are more attributes after preprocessing the Adult data set, so we only marked the corresponding SVM classification accuracy when the number of retained principal components k are $5, 10, 15, 20, 25, 30, 35$, and 40 in Figures 3(b) and 3(d). For the NLTCS data set, it can be seen from Figures 3(e) and 3(g) that the contribution rate of only the first principal component has reached more than 30%. The cumulative contribution rate of the top seven principal components can reach 80%, and it can be seen from Figures 3(f) and 3(h) that the corresponding SVM classification accuracy can reach more than 80%.

The common conclusion is that when the cumulative contribution rate increases (the number of principal components retained increases), the SVM classification accuracy increases accordingly. This phenomenon is consistent with the principle of principal component analysis. The principal components are not correlated with each other and contain the information of the original data. The more principal components retained, the more information of the original data contained in the published data, and the better the performance of the published data set.

5.2. Performance Comparison of HPDP-DP and DP-SUBN^3 with Different Privacy Budgets. In this part of the experiments, we fixed the number of data owners to three while making the privacy budget ε take different values. Figure 4 shows the impact of privacy budgets on HPDP-DP and DP-SUBN^3 algorithms. Figures 4(a) and 4(b) show the SVM classification accuracy of the HPDP-DP and DP-SUBN^3 algorithms on Adult data set. Figures 4(c) and 4(d) show the SVM classification accuracy of the HPDP-DP and DP-SUBN^3 algorithms on NLTCS data set. From Figure 4, except for the salary classifier of the Adult data set, the performance of HPDP-DP algorithm is

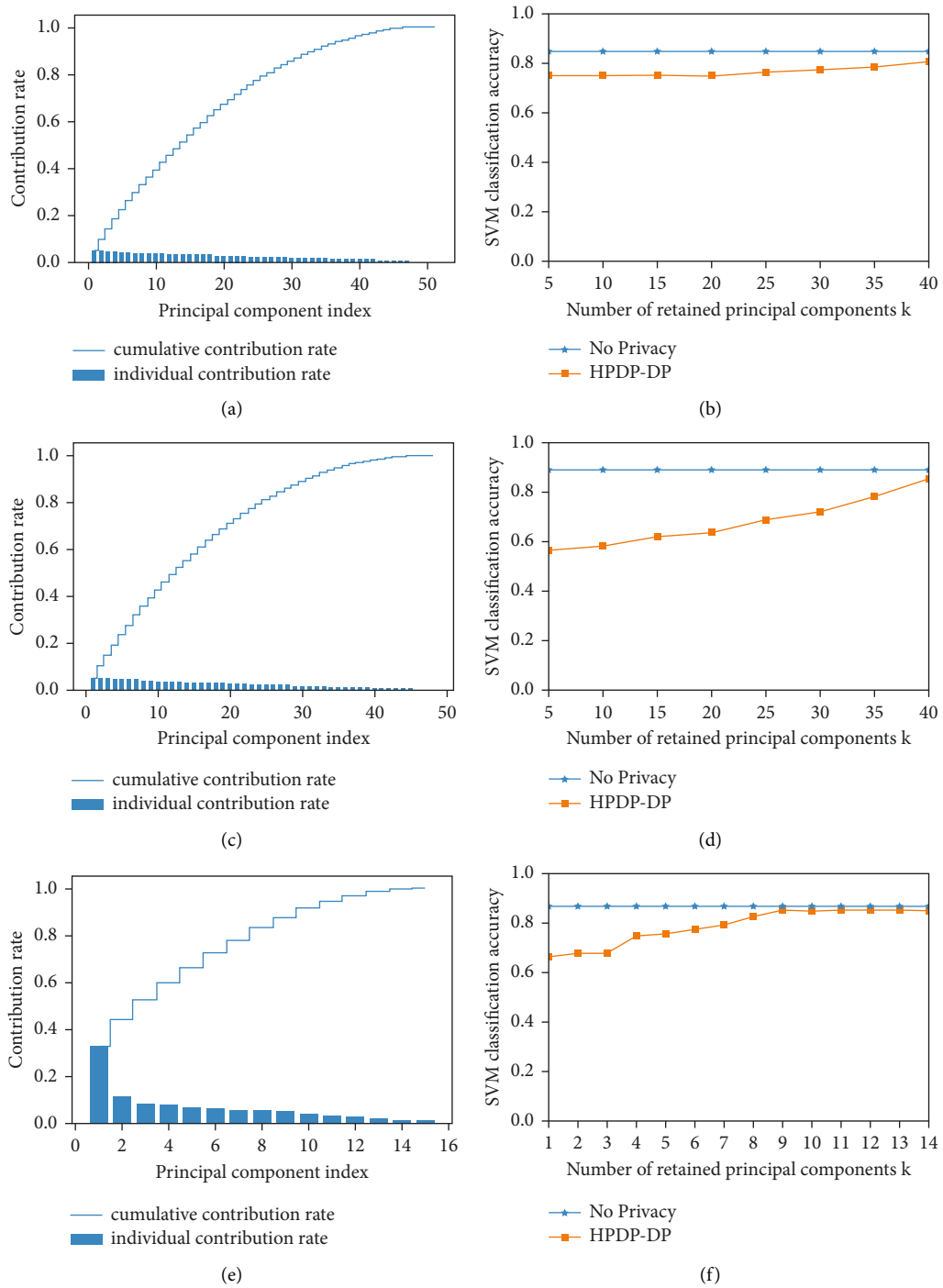

Figure 3: Continued.

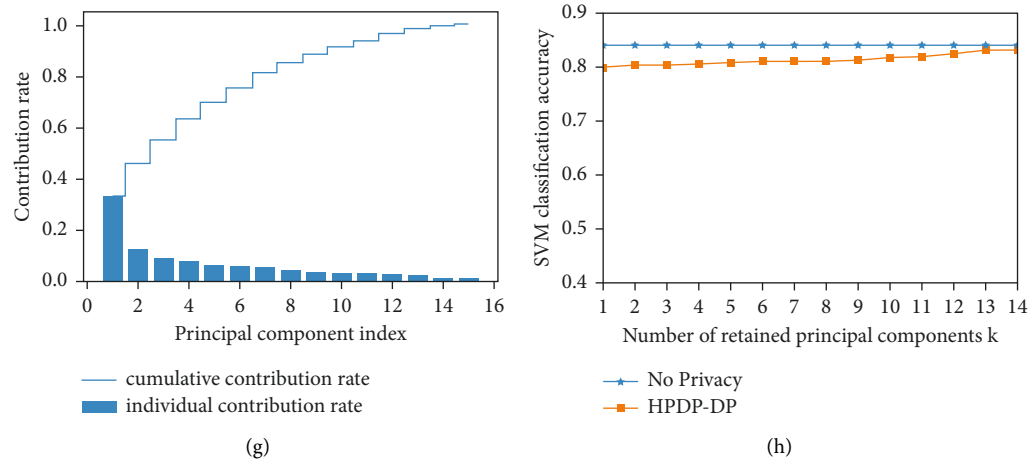

Figure 3: The impact of the number of principal components retained on the SVM classification accuracy. (a) Adult, Y = salary. (b) Adult, Y = salary. (c) Adult, Y = education. (d) Adult, Y = education. (e) NLTCS, Y = money. (f) NLTCS, Y = money. (g) NLTCS, Y = outside. (h) NLTCS, Y = outside.

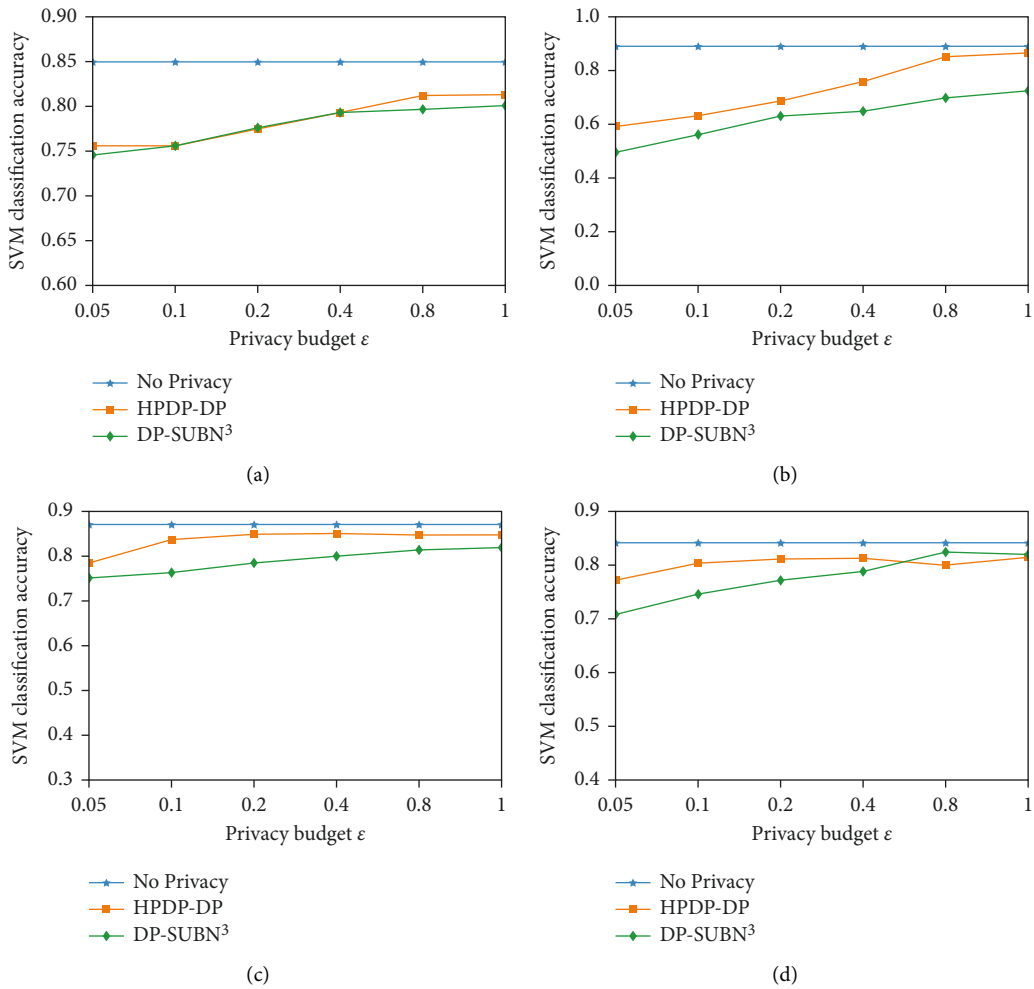

Figure 4: Performance comparison of HPDP-DP and DP-SUBN ^{3} with different privacy budgets. (a) Adult, Y = salary. (b) Adult, Y = education. (c) NLTCS, Y = money. (d) NLTCS, Y = outside.

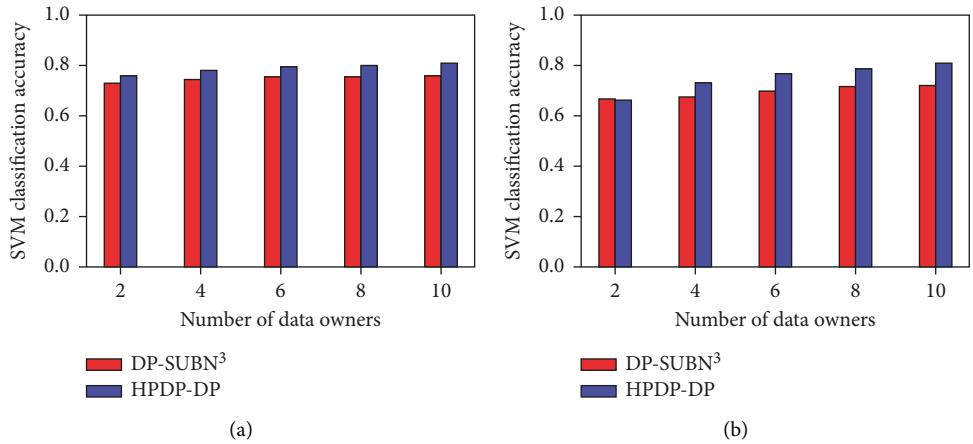

Figure 5: The impact of the number of data owners on the SVM classification accuracy. (a) Adult, Y = salary. (b) Adult, Y = education.

significantly better than DP-SUBN^3 algorithm. Even for the salary classifier of the Adult data set, the SVM classification accuracy of HPDP-DP algorithm is still not lower than DP-SUBN^3 algorithm. From Figure 4, the experimental results show that the SVM classification accuracy of both synthetic data sets released by HPDP-DP and DP-SUBN^3 algorithms increases with the increase of the privacy budget. This is because, according to the definition of differential privacy, when the privacy budget ε increases, the degree of privacy protection decreases and the availability of the released data increases.

5.3. The Impact of the Number of Data Owners on the SVM Classification Accuracy. In order to study the effect of the number of data owners on the performance of the HPDP-DP algorithm, in this section, we set the number of data owners to 2, 4, 6, 8, and 10. We fix the privacy budget ε to 0.2. The results in Figure 5 show that the performance of HPDP-DP algorithm is better than that of DP-SUBN^3 algorithm. We can observe that when the number of data owners increases, the SVM classification accuracy of the synthetic data sets released by HPDP-DP and DP-SUBN^3 algorithms increases accordingly. For DP-SUBN^3 algorithm, the reason is that when the number of data owners increases, the number of update iterations in DP-SUBN^3 algorithm increases, which helps to get better Bayesian network. For HPDP-DP algorithm, we use the weighted average of the local covariance matrices as an estimate of the covariance matrix of the pooled data, and the estimation effect will get better as the number of data owners increases. At the same time, we use the distributed Laplace mechanism to add noise to the shared data, so even when the number of data owners increases, the aggregated result still contain only one share of random noise (the same level as the centralized scene). The scale of random noise is determined only by the privacy budget and the sensitivity. Therefore, the SVM classification accuracy of the synthetic data set released by HPDP-DP algorithm increases as the number of data owners increases.

6. Conclusion

In this paper, in order to privately publish the horizontally partitioned data owned by multiple parties, we present a multiparty horizontally partitioned data publishing method with differential privacy. We use the weighted average of the covariance matrices of the local data to estimate the covariance matrix of the pooled data and then obtain the principal components of the pooled data. In order to protect the privacy of the local data and improve the utility of the published data, we exploit the infinite divisibility of the Laplace distribution to add noise to the locally shared data to improve the utility of the published data. The experimental results show that the synthetic data set released by the HPDP-DP algorithm can maintain high utility. However, this paper also has limitations. (1) The principal component analysis is only suitable for linear dimensionality reduction and not for nonlinear dimensionality reduction. (2) The HPDP-DP algorithm is only suitable for horizontally partitioned data publishing, not for vertically partitioned data publishing. We will conduct research on these aspects in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," *International Conference on Artificial Intelligence and Statistics*, vol. 04, pp. 1472–1482, 2012.

[2] R. Lu, X. Liang, L. Xu, X. Lin, and X. Shen, "Eppa: an efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on*

Parallel and Distributed Systems, vol. 23, no. 9, pp. 1621–1631, 2012.

[3] C. Wang, D. Wang, G. Xu, and D. He, "Efficient privacy-preserving user authentication scheme with forward secrecy for industry 4.0," *Science China Information Sciences*, vol. 65, no. 1, pp. 767–784, 2020.

[4] Y. T. Tsou, "PPDCA: privacy-preserving crowdsourcing data collection and analysis with randomized response," *IEEE Access*, vol. 6, pp. 76970–76983, 2018.

[5] X. Ren, C. M. Yu, W. Yu et al., "High-dimensional crowdsourced data publication with local differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.

[6] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[7] Z. Li and D. Wang, "Achieving one-round password-based authenticated key exchange over lattices," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 308–321, 2022.

[8] Z. Li, D. Wang, and E. Morais, "Quantum-safe round-optimal password authentication for mobile devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1885–1899, 2020.

[9] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2fa: Efficient quantum-resistant two-factor authentication scheme for mobile devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 24, p. 1, 2021.

[10] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Journal of Privacy and Confidentiality*, vol. 7, no. 3, pp. 17–51, 2017.

[11] C. Han and K. Wang, "Sensitive disclosures under differential privacy guarantees," *IEEE International Congress on Big Data*, vol. 25, pp. 110–117, 2015.

[12] Q. Wang, Y. Zhang, L. Xiao, Z. Wang, and K. Ren, "Rescuedp: real-time spatio-temporal crowd-sourced data publishing with differential privacy," in *Proceedings of the IEEE Infocom -the IEEE International Conference on Computer Communications*, 10-14 April 2016.

[13] W. Hao and Z. XuCts-dp, "Publishing correlated time-series data via differential privacy," *Knowledge-Based Systems*, vol. 122, pp. 167–179, 2017.

[14] H. Wang and H. Wang, "Correlated tuple data release via differential privacy," *Information Sciences*, vol. 560, pp. 347–369, 2021.

[15] S. Chen, A. Fu, S. Yu, H. Ke, and M. Su. Dp-qic, "A differential privacy scheme based on quasi-identifier classification for big data publication," *Soft Computing*, vol. 25, no. 3, p. 2021, 2021.

[16] X. Jiang, Z. Ji, S. Wang, N. Mohammed, S. Cheng, and L. Ohno-Machado, "Differential-private data publishing through component analysis," *Transactions on data privacy*, vol. 6, no. 1, pp. 19–34, 2013.

[17] J. Zhang, G. Cormode, C. M. Procopiuc, and D. Srivastava, "PrivBayes," *ACM Transactions on Database Systems*, vol. 42, no. 4, pp. 1–41, 2017.

[18] C. Rui, X. Qian, Z. Yu, and J. Xu, "Differentially private high-dimensional data publication via sampling-based inference," in *Proceedings of the 21th ACM SIGKDD International Conference*, 2015.

[19] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren, "Dppro: differentially private high-dimensional data release via random projection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3081–3093, 2017.

[20] X. Zhang, L. Chen, K. Jin, and X. Meng, "Private high-dimensional data publication with junction tree," *Journal of Computer Research and Development*, vol. 55, no. 12, pp. 2794–2809, 2018.

[21] W. Zhang, J. Zhao, F. Wei, and Y. Chen, "Differentially private high-dimensional data publication via Markov network," *ICST Transactions on Security and Safety*, vol. 6, no. 19, p. 159626, 2019.

[22] Z. Gu, G. Zhang, C. Ma, and L. Song, "Differential privacy data publishing method based on the probabilistic principal component analysis," *Journal of Harbin Engineering University*, vol. 42, no. 8, pp. 1217–1223, 2021.

[23] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 26-29 October 2013.

[24] T. T. Nguyên, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin, "Collecting and analyzing data from smart device users with local differential privacy," p. 11, 2016, http://arxiv.org/abs/1606.05053.

[25] Y. Sei, J. Andrew, H. Okumura, and A. Ohsuga, "Privacy-preserving collaborative data collection and analysis with many missing values," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2022.

[26] D. Alhadidi, N. Mohammed, B. Fung, and M. Debbabi, "Secure distributed framework for achieving-differential privacy," *Springer, Berlin, Heidelberg*, vol. 15, no. 4, pp. 316–333, 2012.

[27] Y. Hong, J. Vaidya, H. Lu, P. Karras, and S. Goel, "Collaborative search log sanitization: toward differential privacy and boosted utility," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 504–518, 2015.

[28] J. Ge, Z. Wang, M. Wang, and L. Han, "Minimax-optimal privacy-preserving sparse pca in distributed systems," in *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, pp. 1589–1598, Playa Blanca, Lanzarote, Canary Islands, April 9 - 11, 2018.

[29] S. Wang and J. M. Chang, "Differentially private principal component analysis over horizontally partitioned data," in *Proceedings of the 2018 IEEE Conference on Dependable and Secure Computing*, 10-13 December 2018.

[30] X. Cheng, P. Tang, S. Su, R. Chen, Z. Wu, and B. Zhu, "Multi-party high-dimensional data publishing under differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1557–1571, 2020.

[31] R. Wang, B. Fung, Y. Zhu, and Q. Peng, "Differentially private data publishing for arbitrarily partitioned data," *Information Sciences*, vol. 553, no. 10, pp. 247–265, 2021.

[32] Z. Gu, G. Zhang, and C. Yang, "Multi-party high-dimensional related data publishing via probabilistic principal component analysis and differential privacy," in *Security and Privacy in New Computing Environments*, W. Shi, X. Chen, and K. K. R. Choo, Eds., , pp. 117–131, Springer International Publishing, 2022.

[33] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B*, vol. 61, no. 3, pp. 611–622, 1999.

[34] R. Cynthia and A. Dwork, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.

[35] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace Distribution and Generalizations*, p. 01, Birkhäuser, Boston, MA, 2001.
[36] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *Proc. EUROCRYPT'99, Czech Republic, May*, vol. 34, pp. 223–238, 1999.
[37] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private naive bayes learning over multiple data sources," *Information Sciences*, vol. 444, pp. 89–104, 2018.
[38] Lib, "StatLib---Datasets Archive," Available at: http://lib.stat.cmu.edu/datasets/, September 8.
[39] D. Dua and C. Graff, *Uci Machine Learning Repository*, University of california, school of information and computer science, irvine, ca, 2019, http://archive.ics.uci.edu/ml.