Hindawi
Security and Communication Networks
Volume 2021, Article ID 5542283, 12 pages
https://doi.org/10.1155/2021/5542283

Research Article

Differential Privacy Principal Component Analysis for Support Vector Machines

Yuxian Huang,[1,2] **Geng Yang**,[1,2] **Yahong Xu**,[1,2] **and Hao Zhou**[1,2]

[1]*College of Computer Science and Software, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China*
[2]*Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing, Jiangsu 210023, China*

Correspondence should be addressed to Yuxian Huang; 1020041306@njupt.edu.cn

Received 23 January 2021; Accepted 5 July 2021; Published 31 July 2021

Academic Editor: A. Peinado

In big data era, massive and high-dimensional data is produced at all times, increasing the difficulty of analyzing and protecting data. In this paper, in order to realize dimensionality reduction and privacy protection of data, principal component analysis (PCA) and differential privacy (DP) are combined to handle these data. Moreover, support vector machine (SVM) is used to measure the availability of processed data in our paper. Specifically, we introduced differential privacy mechanisms at different stages of the algorithm PCA-SVM and obtained the algorithms DPPCA-SVM and PCADP-SVM. Both algorithms satisfy $(\varepsilon, 0)$-DP while achieving fast classification. In addition, we evaluate the performance of two algorithms in terms of noise expectation and classification accuracy from the perspective of theoretical proof and experimental verification. To verify the performance of DPPCA-SVM, we also compare our DPPCA-SVM with other algorithms. Results show that DPPCA-SVM provides excellent utility for different data sets despite guaranteeing stricter privacy.

1. Introduction

Data mining is a hot spot in the field of artificial intelligence and database research. In the past ten years, it has been widely used in our life, revolutionizing the face of the whole world. However, while providing convenience for people, data mining also brings about a series of hidden dangers. For example, in many modern information systems, the amount of data is very large, increasing the difficulty of data processing. Principal component analysis (PCA) [1] is a standard data analysis method that can reduce the dimension of data, make the data easier to process, and cut down the computational overhead of the algorithm. More specifically, it obtains low-dimensional data by projecting the original high-dimensional data into the principal component space composed of the eigenvectors of the data covariance matrix. What is more, these low-level data can represent most of the original data, revealing the nature of the data.

In addition, the privacy leakage of personal data is another drawback under the development of the data mining. There is no doubt that principal component analysis is an efficient method to reduce the dimensionality of data, but it also brings about some safety risks while dealing with private data. Financial systems and healthcare systems often contain private or sensitive information. If data processing algorithms like PCA are performed directly on the original data, the output of them will be likely to leak private information. As a result, how to protect sensitive information while dealing with data is one of the urgent problems in the field of the data mining. Differential privacy (DP) [2] is an effective privacy protection method which can protect individual information while ensuring basic statistics of original data through adding a proper amount of noise to the query results or analysis results [3]. Under the differential privacy protection model, the calculation results of data set are not sensitive to the changes of specific records, and the risk of privacy leakage when adding or deleting a record is controlled within a very small range [4]. In addition, the notion of differential privacy has two types: $(\varepsilon, 0)$-DP and (ε, δ)-DP [5]. $(\varepsilon, 0)$-DP is usually called pure differential privacy, while

(ε, δ)-DP with $\delta > 0$ is called approximate differential privacy. (ε, δ)-DP is a weaker version of $(\varepsilon, 0)$-DP as the former provides freedom to violate strict differential privacy for some low probability events.

The method that combines differential privacy and principal component analysis is mainly divided into input perturbation and output perturbation. Input perturbation adds noise to covariance matrix in the principal component analysis algorithm, while output perturbation adds noise to the output of the desired algorithm. There are several researches on principal component analysis of differential privacy. Blum et al. [6] first proposed the early input perturbation framework SuLQ. SuLQ (Sublinear Queries) guarantees (ε, δ)-DP through perturbing the covariance matrix A. It adds a matrix N of i.i.d. Gaussian noise and applies the PCA algorithm to matrix $A + N$. One drawback of this method is that matrix $A + N$ is not symmetric and the largest eigenvalue may not be real when the probability is 1. Therefore, Chaudhuri et al. [7] modify SuLQ by adding a symmetric noise matrix, so that eigenvalues are all real. The algorithm MOD-SuLQ also satisfies (ε, δ)-DP. Besides, Chaudhuri et al. proposed a new method, PPCA, which randomly samples a k-dimensional subspace from a distribution that ensures differential privacy and is biased towards high utility. Kapralov and Talwar [8] argued that the algorithm (Chaudhuri et al.) lacks convergence time guarantee, and they designed a complex algorithm using the exponential mechanism but is complicated to implement for high-dimensional data. Dwork et al. [9] provided the algorithms for (ε, δ)-DP, adding Gaussian noise to the original sample covariance matrix. Inspired by Dwork, Hafiz et al. [10, 11] and Wu-xuan Jiang et al. [12] designed their algorithms for $(\varepsilon, 0)$-DP. Both of them added Wishart noise and selected parameters with a better range of utility.

Obviously, previous algorithms were mostly based on the idea of input perturbation, and few people are involved in principal component analysis of differential privacy based on the output perturbation. At the same time, the privacy protection level and performance of algorithms on the basis of these perturbation methods are rarely compared and analyzed. What is more, most of the current principal component analyses of differential privacy can only guarantee (ε, δ)-DP and lack a method to measure data availability.

In this paper, support vector machine (SVM) [13] is added to measure the availability of processed data through comparing the accuracy of classification. On the basis of it, we combine principal component analysis, differential privacy, and support vector machines to propose two new algorithms: DPPCA-SVM based on input perturbation and PCA-DPSVM based on output perturbation. Both of these algorithms guarantee $(\varepsilon, 0)$-DP. In addition, for the purpose of analyzing the performance of algorithms based on different perturbations, we evaluate the classification accuracy of SVM, PCA-DPSVM, and DPPCA-SVM. Meanwhile, DPPCA-SVM is compared with other recent algorithms, such as MOD-SuLQ-SVM [7] and AG-SVM [9]. Through these experiments, we find that DPPCA-SVM and PCA-DPSVM provide better privacy protection while accomplishing the task of data processing. What is more, results show that DPPCA-SVM provides excellent utility for different data sets despite guaranteeing stricter privacy.

There is no doubt that DPPCA-SVM and PCA-DPSVM are valuable. On the one hand, it is efficient for them to cope with the two major difficulties in the current data processing field, dimensionality reduction, and privacy protection. On the other hand, they can also provide ideas for researchers in choosing the location of disturbance and the magnitude of noise to achieve differential privacy. Our main contributions are as follows:

(1) We propose DPPCA-SVM and PCA-DPSVM through applying Laplace mechanism to PCA-SVM algorithms. Furthermore, we give proof for $(\varepsilon, 0)$-DP of them.

(2) We contrast the performance of two algorithms in terms of noise expectation via theoretical analysis. Less noise means less error and better classification accuracy. Through theoretical verification, we ensure that DPPCA-SVM has better performance than PCA-DPSVM.

(3) We conduct the experiments to verify the performance of algorithms DPPCA-SVM and PCA-DPSVM in terms of classification accuracy on three real data sets. Then we compare our DPPCA-SVM with other recent algorithms AG-SVM and MOD-SuLQ-SVM, and the experimental results show that our algorithm can provide stronger privacy guarantee and excellent data utility. At last, we show that using principal component analysis before SVM can obviously save on computational complexity.

The rest of the paper is organized as follows: Section 2 introduces principal component analysis, support vector machines, and differential privacy. Section 3 first describes the two algorithms we proposed and then analyzes the privacy and utility of them. Section 4 shows the performance of algorithms on three real data sets compared with other algorithms. Section 5 concludes the paper.

2. Preliminaries

2.1. Principal Components Analysis (PCA). Given a data set $X = [x_1, x_2, \cdots, x_n]^T$, where x_i is the i-th sample, matrix $X \in R^{n \times d}$ contains information about d attributes of n samples (generally $d \ll n$), and we assume that the norm of each sample satisfies $\|x_i\|_2 \leq 1$. Define the $d \times d$ symmetric covariance matrix of the original data as

$$A = \frac{1}{n} X^T X = \frac{1}{n} \sum_{i=1}^{n} x_i^T x_i. \tag{1}$$

The principal components are obtained by computing eigenvalues and corresponding eigenvectors of the covariance matrix A:

$$A v_i = \lambda_i v_i, \tag{2}$$

where $\lambda_i (1 \leq i \leq d)$ is one of the eigenvalues of covariance matrix A, illustrating the proportion of information that

corresponding component contains. Larger λ_i means that the component is more important. We assume that λ_i are sorted in the descending order, that is, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_d \geq 0$. v_i is the corresponding eigenvector.

In order to reduce data dimension, a target dimension k is needed. We want to select first k eigenvectors, which correspond to the top k eigenvalues. For selecting the number k, threshold $\alpha\,(0 \leq \alpha \leq 1)$ is introduced to denote accumulative contribution rate of principal components. Given a parameter α, target dimension k can be decided by

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq \alpha. \tag{3}$$

Suppose that $V_k = (v_1, v_2, \ldots, v_k)$ are the first k eigenvectors of A, and v_i and $v_j\,(1 \leq i, j \leq k)$ are orthonormal. We project the original data X to V_k to get low-dimensional data:

$$Y = XV_k, \tag{4}$$

where $Y \in R^{n \times k}$; we substitute $Y = [y_1, y_2, \ldots y_n]^T$ for $X = [x_1, x_2, \ldots, x_n]^T$, so the data dimension is reduced and the computational complexity of algorithm can be saved.

2.2. Support Vector Machine (SVM). For a data set $D = \{(x_i, y_i)\}_{i=1}^{n}$, data x_i is the i-th record, and label $y_i \in \{-1, 1\}$. The classification decision function is defined as

$$f(x) = \operatorname{sgn}(wx + b) \tag{5}$$

where $sgn(\cdot)$ represents the symbol function, w denotes vector orthogonal to optimal hyperplane, and b is a constant.

To obtain the estimations of w and b, the following minimization problem should be solved:

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2, \\ &s.t. \quad \begin{aligned} &y_i(wx_i + b) \geq 1, \\ &i = 1, 2, \cdots n. \end{aligned} \end{aligned} \tag{6}$$

The objective function is obviously a convex function. The Lagrangian dual function can be introduced to convert the constrained original objective function into an unconstrained Lagrangian objective function [14]. The formula is transformed as follows:

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(wx_i + b) - 1\right), \\ &s.t. \quad \alpha_i \geq 0, \end{aligned} \tag{7}$$

where α_i is a Lagrangian multiplier, $0 \leq \alpha_i \leq C$, and C is a constant, denoting penalty factor.

Let $L(w, b, \alpha) = 1/2\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(wx_i + b) - 1)$; compute the gradient of $\mathscr{W}$ and b in function L:

$$\begin{cases} \dfrac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i, \\ \dfrac{\partial L}{\partial w} = \dfrac{1}{2} \times 2 \times w - \sum_{i=1}^{n} \alpha_i x_i y_i. \end{cases} \tag{8}$$

For $\partial L/\partial b = \partial L/\partial w = 0$, we have

$$\begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0, \\ w = \sum_{i=1}^{n} \alpha_i x_i y_i. \end{cases} \tag{9}$$

So, the classification decision function can be simplified as follows:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i y_i x_i x_j\right). \tag{10}$$

2.3. Differential Privacy

Definition 1 (differential privacy) (see [15]). A randomized mechanism M satisfies ε-differential privacy, if, for any two neighbouring data sets D and $D\prime$ (with at most one different sample) and for all outputs $O\,(O \subseteq \operatorname{range}(M))$,

$$\Pr[M(D) \in O] \leq e^{\varepsilon} \times \Pr[M(D\prime) \in O], \tag{11}$$

where $\varepsilon > 0$ is the privacy budget controlling the strength of privacy guarantee. Lower ε ensures more privacy guarantee.

Definition 2 (sensitivity) (see [16]). For a function $f\colon D \longrightarrow R^d$ and any two neighbouring data sets D and D', the sensitivity of function f is defined as

$$\Delta f = \max_{D, D'} \|f(D) - f(D')\|_1. \tag{12}$$

The sensitivity describes the largest change due to a sample replacement. Sensitivity Δf is only related to the function f.

The Laplace mechanism adds independent noise to data; we use *lap(b)* to represent the noise sampled from Laplace distribution with a scaling of b.

Definition 3 (Laplace mechanism) (see [17]). Given a data set D, for a function $f\colon D \longrightarrow R^d$, with sensitivity Δf, the mechanism M provides ε–differential privacy satisfying

$$M(D) = f(D) + \operatorname{lap}\left(\frac{\Delta f}{\varepsilon}\right), \tag{13}$$

where $\operatorname{lap}(\cdot)$ is a random variable. Its probability density function is

$$p(x) = \frac{1}{2b} e^{-(|x|/b)}. \tag{14}$$

3. Proposed Algorithms and Analysis

In this section, we propose two algorithms: DPPCA-SVM and PCA-DPSVM. Both algorithms can achieve fast classification while reducing the risk of sample leakage in data set. Through theoretical analysis, we prove that they satisfy $(\varepsilon, 0)$-DP differential privacy. Meanwhile, the utility of two proposed algorithms is investigated in this section. Table 1 shows the notations that will be used in this paper.

3.1. Algorithm Description. The algorithm DPPCA-SVM takes the low-dimensional data with privacy protection as the input of support vector machines. It can simplify data and reduce complexity of computation at the same time. Meanwhile, it provides adequate protection for private data. In DPPCA-SVM, we compute covariance matrix of original data X and then add symmetric noise matrix to it. Each element in noise matrix is sampled from Laplace distribution. Afterwards, we follow standard PCA to calculate first k eigenvectors to make up principal components space. Then original high-dimensional data is projected to principal components space to obtain low-dimensional data. The low-dimensional data now is under privacy protection and can be directly applied to support vector machines to train classification function. Algorithm DPPCA-SVM is described in Algorithm 1.

Algorithm DPPCA-SVM introduces differential privacy into principal component analysis to achieve privacy protection of training data. We also can add noise to the real classification hyperplane (w, b) computed by support vector machines so that attackers are unable to obtain real classification hyperplane and training data by simulating testing data.

The idea of algorithm PCA-DPSVM is to take the original low-dimensional data computed by principal component analysis as the input of support vector machines. Then we add noise to real classification hyperplane. Since parameter b does not contain information about training data and will not leak privacy, the perturbation of classification hyperplane is concentrated on the perturbation of parameter w. The algorithm PCA-DPSVM is described in Algorithm 2.

3.2. Privacy Analysis. Before proving that algorithm DPPCA-SVM satisfies $(\varepsilon, 0)$−differential privacy, we should analyze its sensitivity. Suppose that there are two neighbouring data sets $X = [x_1, \ldots, x_i, \ldots, x_n]^T \in R^{n\times d}$ and $X\prime = [x_1, \ldots, x_i', \ldots, x_n]^T \in R^{n\times d}$, where $x_i \neq x_i'$, and we assume the normalized data vector $\|x_i\|_2 \leq 1$.

Lemma 1. *In algorithm DPPCA-SVM, for all the input data, denote $f(X) = 1/nX^TX$; then the sensitivity of function $f(X)$ is equal to $2\ d/n$.*

Proof. Suppose that A_1 and A_2 are the covariance matrices of X and X', respectively.

$$\begin{cases} A_1 = \dfrac{1}{n}X^TX, \\ \\ A_2 = \dfrac{1}{n}X'^TX\prime. \end{cases} \tag{15}$$

According to Definition 2, the sensitivity of function $f(X)$ is equal to $\max\|A_1 - A_2\|_1$. Then we have

$$\begin{aligned} \left\|A_1 - A_2\right\|_1 &= \left\|\frac{1}{n}X^TX - \frac{1}{n}X'^TX'\right\|_1 \\ &= \frac{1}{n}\left\|x_i^Tx_i - x_i'^Tx_i'\right\|_1, \end{aligned} \tag{16}$$

where $\|\cdot\|_1$ denotes the l_1 norm; for a matrix $C \in R^{m\times n}$, $\|C\|_1 = \max_j \sum_{i=1}^m |c_{ij}|\ (1 \leq j \leq n)$.

For normalized $\|x_i\|_2 \leq 1$, we have

$$\begin{aligned} \left\|A_1 - A_2\right\|_1 &= \frac{1}{n}\left\|x_i^Tx_i - x_i'^Tx_i'\right\|_1 \\ &\leq \frac{1}{n}\left(\left\|x_i^Tx_i\right\|_1 + \left\|x_i'^Tx_i'\right\|_1\right) \leq \frac{2\ d}{n} \end{aligned} \tag{17}$$

□

Theorem 1. *Algorithm DPPCA-SVM satisfies $(\varepsilon, 0)$−differential privacy.*

Proof. For $A\prime$ derived from algorithm DPPCA-SVM on X and $X\prime$, we obtain $A\prime = A_1 + N_1$ and $A\prime = A_2 + N_2$, where N_1 and N_2 are the corresponding noise matrices.

$$\begin{aligned} \frac{f(A'|X)}{f(A'|X')} &= \frac{p(N_1)}{p(N_2)} \\ &= e^{(n\varepsilon/2\ d)\left(\|N_2\|_1 - \|N_1\|_1\right)}. \end{aligned} \tag{18}$$

$p(N_1)$ and $p(N_2)$ are the density functions of output functions at neighbouring data sets X and X', respectively.

According to Lemma 1, we have

$$\begin{aligned} \left\|N_2\right\|_1 - \left\|N_1\right\|_1 &\leq \left\|N_2 - N_1\right\|_1 \\ &= \left\|A_1 - A_2\right\|_1 \leq \frac{2\ d}{n}, \end{aligned} \tag{19}$$

and we can obtain

$$\frac{f(A'|X)}{f(A'|X')} \leq e^{\varepsilon}. \tag{20}$$

Therefore, algorithm DPPCA-SVM satisfies $(\varepsilon, 0)$−differential privacy.

Before proving that algorithm PCA-DPSVM satisfies $(\varepsilon, 0)$−differential privacy, we should also analyze its sensitivity. Suppose that there are two neighbouring classification data sets $D_1 = \{(x_1, y_1) \cdots (x_i, y_i) \cdots (x_n, y_n)\}$ and $D_2 = \{(x_1, y_1) \cdots (x_i, y_i) \cdots (x_n', y_n')\}$, where $x_i \in R^k$ and $x_n \neq x_n'$, and we assume the normalized data vector $\|x_i\|_2 \leq 1$. □

Table 1: The notations used in this paper.

Notations	Meaning
X	A set of samples
x_i	i-th piece of sample
n	The number of samples
d	The number of attributes
$R^{n\times d}$	The value space of samples
A	The covariance matrix of X
E	A noise matrix that obeys the Laplace distribution
A'	The covariance matrix after adding noise
λ_i	The i-th eigenvalue of the matrix
v_i	The i-th eigenvector of the matrix
k	The number of principal components
V_k	The principal component space, composed of eigenvectors corresponding to the first k eigenvalues
Y	The set of low-dimensional samples
w	The normal vector of classification hyperplane
B	The intercept of classification hyperplane
C	The penalty factor
ε	The privacy budget
δ	The privacy parameter

Input: data $X \in R^{n\times d}$, samples n, attributes d, privacy budget ε;
Output: classification decision function $w^T x + b$
(1) Compute covariance matrix of input data $A = 1/nX^T X$;
(2) Compute symmetric noise matrix E, each element in E is sampled from *Laplace* distribution with a scaling of $\text{lap}(2\ d/n\varepsilon)$;
(3) Add noise to the original covariance matrix $A' = A + E$;
(4) Compute eigenvalues λ_i and corresponding eigenvectors v_i of the noise covariance matrix $A' v_i = \lambda_i v_i$;
(5) Select first k eigenvectors V_k to determine the low-dimensional data $Y = XV_k$;
(6) Compute classification function $w, b = SVM(Y)$;

Algorithm 1: Differential privacy principal component analysis-support vector machine (DPPCA-SVM).

Input: data $X \in R^{n\times d}$, samples n, attributes d, privacy budget ε;
Output: noised classification decision function $w'^T x + b$
(1) Compute covariance matrix of input data $A = 1/nX^T X$;
(2) Compute eigenvalues λ_i and corresponding eigenvectors v_i of the covariance matrix $Av_i = \lambda_i v_i$;
(3) Select first k eigenvectors V_k to determine the low-dimensional data $Y = XV_k$;
(4) Compute classification function $w, b = SVM(Y)$;
(5) Add noise $w\prime = w + \text{lap}(2Cn\ d/\varepsilon)$

Algorithm 2: Principal component analysis-differential privacy support vector machine (PCA-DPSVM).

Lemma 2. *In algorithm PCA-DPSVM, for all the input data, denote* $f(x, y) = \sum_{i=1}^{n} \alpha_i x_i y_i$, *and then the sensitivity of function* $f(x, y)$ *is equal to* $2Cn\ d$.

Proof. In Section 3.2, we get the classification hyperplane normal vector expression $w = \sum_{i=1}^{n} \alpha_i x_i y_i$; suppose that w_1 and w_2 are the classification hyperplane normal vectors of D_1 and D_2, respectively. According to Definition 2, the sensitivity of function $f(x, y)$ is $\max\|w_1 - w_2\|_1$. Then we have

$$\|w_1 - w_2\|_1 = \left\|\sum_{i=1}^{n-1} \alpha_i x_i y_i - \sum_{i=1}^{n-1} \alpha_i' x_i y_i + \alpha_n x_n y_n - \alpha_n' x_n' y_n'\right\|_1. \tag{21}$$

According to $\|A + B\|_1 \le \|A\|_1 + \|B\|_1$, we have

$$\|w_1 - w_2\|_1 \le \left\|\sum_{i=1}^{n-1} \alpha_i x_i y_i - \sum_{i=1}^{n-1} \alpha_i' x_i y_i\right\|_1 + \left\|\alpha_n x_n y_n - \alpha_n' x_n' y_n'\right\|_1 \le \left\|\sum_{i=1}^{n-1} \alpha_i x_i y_i\right\|_1 + \left\|\sum_{i=1}^{n-1} \alpha_i' x_i y_i\right\|_1 + \left\|\alpha_n x_n y_n\right\|_1 + \left\|\alpha_n' x_n' y_n'\right\|_1. \tag{22}$$

For normalized $\|x_i\|_2 \le 1$ and $0 \le \alpha_i \le C$, we have

$$\|w_1 - w_2\|_1 \le \left\|\sum_{i=1}^{n-1} \alpha_i x_i y_i\right\|_1 + \left\|\sum_{i=1}^{n-1} \alpha_i' x_i y_i\right\|_1 + \|\alpha_n x_n y_n\|_1 + \|\alpha_n' x_n' y_n'\|_1 \le (n-1)C\ d + (n-1)C\ d + C\ d + C\ d \le 2Cn\ d. \tag{23}$$

Theorem 2. *Algorithm PCA-DPSVM satisfies* $(\varepsilon, 0)$*–differential privacy.*

Proof. For w' derived from algorithm PCA-DPSVM on D_1 and D_2, we obtain $w' = w_1 + N_1, w' = w_2 + N_2$, where N_1 and N_2 are the corresponding noise vectors.

$$\frac{f\left(w\prime|D_1\right)}{f\left(w\prime|D_2\right)} = \frac{p(N_1)}{p(N_2)} = e^{(\varepsilon/2Cn\ d)\left(\|N_2\|_1 - \|N_1\|_1\right)}. \tag{24}$$

$p(N_1)$ and $p(N_2)$ are the density functions of output functions at neighbouring data sets D_1 and D_2, respectively. According to Lemma 2, we have

$$\|N_2\|_1 - \|N_1\|_1 \le \|N_2 - N_1\|_1 = \|w_1 - w_2\|_1 \le 2Cn\ d, \tag{25}$$

and we can obtain

$$\frac{f\left(w\prime|D_1\right)}{f\left(w\prime|D_2\right)} \le e^{\varepsilon}. \tag{26}$$

Therefore, algorithm PCA-DPSVM satisfies $(\varepsilon, 0)$–differential privacy. □

3.3. Utility Analysis. In Section 3.2, we prove that algorithms DPPCA-SVM and PCA-DPSVM both satisfy $(\varepsilon, 0)$–differential privacy; next we evaluate the performance of the two algorithms. It is obvious that adding noise can protect data privacy but at the same time it will have negative effect on data utility. Noise magnitude directly determines effect magnitude. So we evaluate the performance of the two algorithms in terms of noise magnitude.

Theorem 3. *For a given privacy parameter ε, algorithm DPPCA-SVM adds less noise compared to PCA-DPSVM.*

Proof. In algorithm DPPCA-SVM, the noise covariance matrix $A\prime = A + \mathrm{lap}(2\ d/n\varepsilon)$; let $Z = A\prime - A = \mathrm{lap}(2\ d/n\varepsilon)$; now we calculate noise expectation $E_1(|Z|)$; smaller $E_1(|Z|)$ guarantees less noise and stronger utility. Let $\Delta = 2\ d/n$, so

$$E_1(|Z|) = \int_{-\infty}^{+\infty} |Z| f(|Z|)\mathrm{d}z = \int_{-\infty}^{+\infty} |Z| \frac{\varepsilon}{2\Delta} e^{-(|Z|\varepsilon/\Delta)}\mathrm{d}z. \tag{27}$$

□

According to formula $E_1(|Z|) = E_1(|Z|) + E_1(-Z)$, we have

$$\begin{aligned} E_1(|Z|) &= \int_{-\infty}^{+\infty} |Z| \frac{\varepsilon}{2\Delta} e^{-(|Z|\varepsilon/\Delta)}\mathrm{d}z \\ &= \int_{-\infty}^{0} (-Z) \frac{\varepsilon}{2\Delta} e^{(Z\varepsilon/\Delta)}\mathrm{d}z + \int_{0}^{+\infty} Z \frac{\varepsilon}{2\Delta} e^{-(Z\varepsilon/\Delta)}\mathrm{d}z \\ &= -\frac{\Delta}{2\varepsilon}\int_{-\infty}^{0} \frac{Z\varepsilon}{\Delta} e^{(Z\varepsilon/\Delta)}\mathrm{d}\left(\frac{Z\varepsilon}{\Delta}\right) + \frac{\Delta}{2\varepsilon}\int_{0}^{+\infty} -\frac{Z\varepsilon}{\Delta} e^{-(Z\varepsilon/\Delta)}\mathrm{d}\left(-\frac{Z\varepsilon}{\Delta}\right). \end{aligned} \tag{28}$$

Let $t = (Z\varepsilon/\Delta)$, so

$$\begin{aligned} E_1(|Z|) &= -\frac{\Delta}{2\varepsilon}\int_{-\infty}^{0} te^{t}\mathrm{d}t + \frac{\Delta}{2\varepsilon}\int_{0}^{+\infty} -te^{-t}\mathrm{d}(-t) \\ &= \frac{\Delta}{2\varepsilon} + \frac{\Delta}{2\varepsilon} = \frac{\Delta}{\varepsilon} = \frac{2\ d}{n\varepsilon}. \end{aligned} \tag{29}$$

The noise expectation of algorithm DPPCA-SVM is $E_1(|Z|) = 2\ d/n\varepsilon$.

In algorithm PCA-DPSVM, we add noise to classification hyperplane normal vector $w\prime = w + \mathrm{lap}(2Cn\ d/\varepsilon)$; let $Z = w\prime - w = \mathrm{lap}(2Cn\ d/\varepsilon)$ to compute noise expectation $E_2(|Z|)$, and let $\Delta = 2Cn\ d$, so

$$\begin{aligned} E_2(|Z|) &= \int_{-\infty}^{+\infty} |Z| \frac{\varepsilon}{2\Delta} e^{-(|Z|\varepsilon/\Delta)}\mathrm{d}z \\ &= \int_{-\infty}^{0} (-Z) \frac{\varepsilon}{2\Delta} e^{(Z\varepsilon/\Delta)}\mathrm{d}z + \int_{0}^{+\infty} Z \frac{\varepsilon}{2\Delta} e^{-(Z\varepsilon/\Delta)}\mathrm{d}z \\ &= -\frac{\Delta}{2\varepsilon}\int_{-\infty}^{0} \frac{Z\varepsilon}{\Delta} e^{(Z\varepsilon/\Delta)}\mathrm{d}\left(\frac{Z\varepsilon}{\Delta}\right) + \frac{\Delta}{2\varepsilon}\int_{0}^{+\infty} -\frac{Z\varepsilon}{\Delta} e^{-(Z\varepsilon/\Delta)}\mathrm{d}\left(-\frac{Z\varepsilon}{\Delta}\right). \end{aligned} \tag{30}$$

Let $t = Z\varepsilon/\Delta$; we have

$$\begin{aligned} E_2(|Z|) &= -\frac{\Delta}{2\varepsilon}\int_{-\infty}^{0} te^{t}\mathrm{d}t + \frac{\Delta}{2\varepsilon}\int_{0}^{+\infty} -te^{-t}\mathrm{d}(-t) \\ &= \frac{\Delta}{2\varepsilon} + \frac{\Delta}{2\varepsilon} = \frac{\Delta}{\varepsilon} = \frac{2Cn\ d}{\varepsilon}. \end{aligned} \tag{31}$$

The noise expectation of algorithm PCA-DPSVM is $E_2(|Z|) = 2Cn\ d/\varepsilon$.

Now we compare $E_1(|Z|)$ and $E_2(|Z|)$ to measure the noise magnitude of two algorithms. Let $\theta = E_1(|Z|)/E_2(|Z|)$; θ is noise ratio.

$$\theta = \frac{E_1(|Z|)}{E_2(|Z|)} = \frac{2\ d/n\varepsilon}{2Cn\ d/\varepsilon} = \frac{1}{Cn^2} < 1. \tag{32}$$

We observe that $\theta < 1$, which means that, for a given privacy parameter ε, algorithm DPPCA-SVM adds less noise compared to PCA-DPSVM. □

3.4. Algorithm Comparison. In Section 3.3, we prove that DPPCA-SVM based on input disturbance has better data availability compared to PCA-DPSVM based on output disturbance. In this section, two other differential privacy principal component analysis algorithms based on input disturbance are introduced. Besides, we compare DPPCA-SVM with them theoretically.

The algorithm AG was proposed by Dwork. It provides $(\varepsilon, \delta) - DP$ privacy protection through adding a symmetric Gaussian noise matrix to the algorithm PCA. The goal of the algorithm is to output a subspace that can protect privacy and preserve data matrix variance as much as possible. The upper triangular element of the matrix is an independent and identically distributed value satisfying $N(0, \Gamma^2)$, where $\Gamma = \Delta f \sqrt{2\ln(1.25/\delta)}/\varepsilon$.

The algorithm MOD-SuLQ proposed by Chaudhuri is an improvement of SuLQ. SuLQ is the only algorithm that provides principal component analysis of differential privacy at the beginning, and it was proposed by Blum in 2005. Blum adds Gaussian noise to covariance matrix A to ensure differential privacy and publishes the first k eigenvectors of the perturbed covariance matrix. But this method has a fatal problem. When the perturbed covariance $A + N$ is an asymmetric matrix, the maximum eigenvalue may not be real, and the corresponding eigenvector will be very complicated. Therefore, instead of adding an asymmetric Gaussian matrix, Chaudhuri adds a symmetric matrix with i.i.d. Gaussian entries N. As a result, the perturbed covariance matrix is symmetric but not necessarily semipositive definite, so some eigenvalues may be negative. However, the eigenvectors are all real, which does not affect subsequent calculations.

Obviously, both of the above algorithms, like DPPCA-SVM, implement privacy protection via input disturbance. However, the mechanism and scale of the noise they add are quite different. Therefore, it is necessary to compare these three algorithms theoretically and experimentally. For input perturbation methods, the magnitude of the noise is usually used to reflect availability of data. As a result, noise scale and privacy level of these algorithms are presented in Table 2. We find that the noise scale added by AG is the smallest among the three algorithms, followed by DPPCA-SVM, and that added by MOD-SuLQ is the largest. In addition, both AG and MOD-SuLQ just meet $(\varepsilon, \delta) - DP$, while DPPCA-SVM satisfies $(\varepsilon, 0) - DP$. Therefore, we can infer that DPPCA-SVM achieves stricter privacy protection while adding noise of a suitable scale. Furthermore, in order to intuitively compare the utility of these algorithms in experiments, we apply support vector machine on the basis of AG and MOD-SuLQ, resulting in AG-SVM and MOD-SuLQ-SVM.

4. Experimental Results and Analysis

In this section, we will first give some experimental results to verify that algorithm DPPCA-SVM outperforms PCA-DPSVM in data utility. Next, in order to verify the effectiveness of the algorithm DPPCA-SVM, we compare it with the existing algorithms SVM, AGPCA-SVM, and MOD-SuLQ-SVM for classification accuracy. Three UCI data sets are used in our experiments: Sensorless [18], Covtype [19], and Musk [20]. The data set information is shown in Table 3; the target dimension k is the value when the principal component contribution rate $\alpha > 85\%$.

Table 2: Display of the noise scale and privacy level.

Algorithm	Noise scale	Privacy level
AG	$O(\sqrt{d}/n\varepsilon)$	(ε, δ)
MOD-SuLQ	$O(d\sqrt{d\log d}/n\varepsilon)$	(ε, δ)
DPPCA	$O(d/n\varepsilon)$	$(\varepsilon, 0)$

4.1. Experiments for Classification Accuracy of Algorithms DPPCA-SVM and PCA-DPSVM. We compare algorithms DPPCA-SVM, PCA-DPSVM, and SVM in terms of classification accuracy; higher classification accuracy indicates higher data utility. We set privacy budget $\varepsilon \in [0.005, 1.5]$. Figure 1 shows that, with the increase of ε, the classification accuracy of three algorithms on all data sets is continuously improved. Classification accuracy is SVM > DPPCA-SVM > PCA-DPSVM.

Algorithm SVM does not involve privacy protection, and its classification accuracy is not affected by privacy budget. Algorithm DPPCA-SVM adds noise to protect private information of data, making classification accuracy slightly lower than that of SVM. However, as the privacy budget increases, the noise added decreases. At the same time, classification accuracy of DPPCA-SVM gradually increases and approaches that of SVM. Figure 1 also visually shows the comparison of algorithms DPPCA-SVM and PCA-DPSVM. When privacy budget is large, the accuracy of algorithm PCA-DPSVM is also very high, which is close to that of DPPCA-SVM, but when privacy budget is small, the strength of privacy protection is large, and the classification accuracy is not satisfactory and is worse than that of DPPCA-SVM.

According to the above experiment, the following conclusions can be achieved. On the one hand, when a certain amount of noise is added to meet the requirements of differential privacy, the availability of results will be inevitably affected. Although the impact will decrease as the privacy budget increases, there is still a slight gap in data availability between the noise-added algorithm and the original algorithm. On the other hand, we found that the method of perturbing the covariance matrix in principal component analysis (DPPCA-SVM) is better than the method of perturbing the normal vector of classification hyperplane in support vector machine (PCA-DPSVM). In other words, if an algorithm is asked to achieve the same level of differential privacy protection, the input disturbance method is more effective than the output disturbance, and the data availability is higher.

4.2. Experiments for Classification Accuracy of Algorithm DPPCA-SVM and Other Algorithms. From Section 4.1, we know that algorithm DPPCA-SVM outperforms

Table 3: Data set information.

Data set	Samples n	Attributes d	Target dimension k
Sensorless	58509	49	14
Covtype	580000	54	6
Musk	6598	166	20

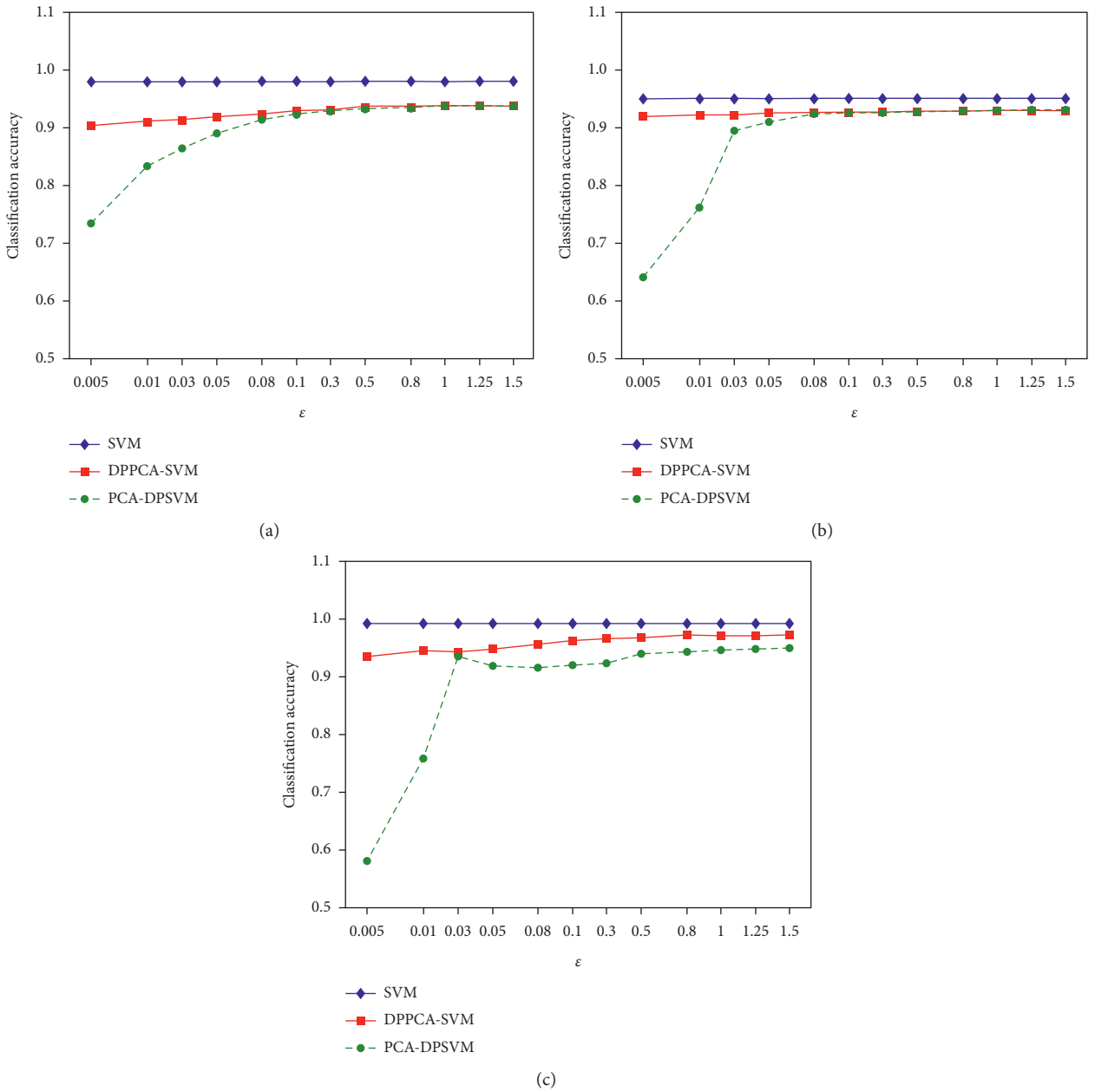

Figure 1: Variation of classification accuracy with different value of privacy budget (SVM, DPPCA-SVM, and PCA-DPSVM). (a) Sensorless, (b) Covtype, and (c) Musk.

PCA-DPSVM in data utility. That is to say, from the perspective of classification accuracy and privacy protection, input perturbation which provides privacy protection for low-dimensional data is better than output perturbation acting on the classification hyperplane. Furthermore, in order to verify the effectiveness of algorithm DPPCA-SVM, we compare it with MOD-SuLQ-SVM [7] and AG-SVM [9] in terms of classification accuracy.

In our experiment, we take privacy budget $\varepsilon \in [0.000005, 1]$. Another privacy parameter δ is set to $1/n^2$. Then, the classification accuracy variation curves of the four classification methods under different privacy budgets are

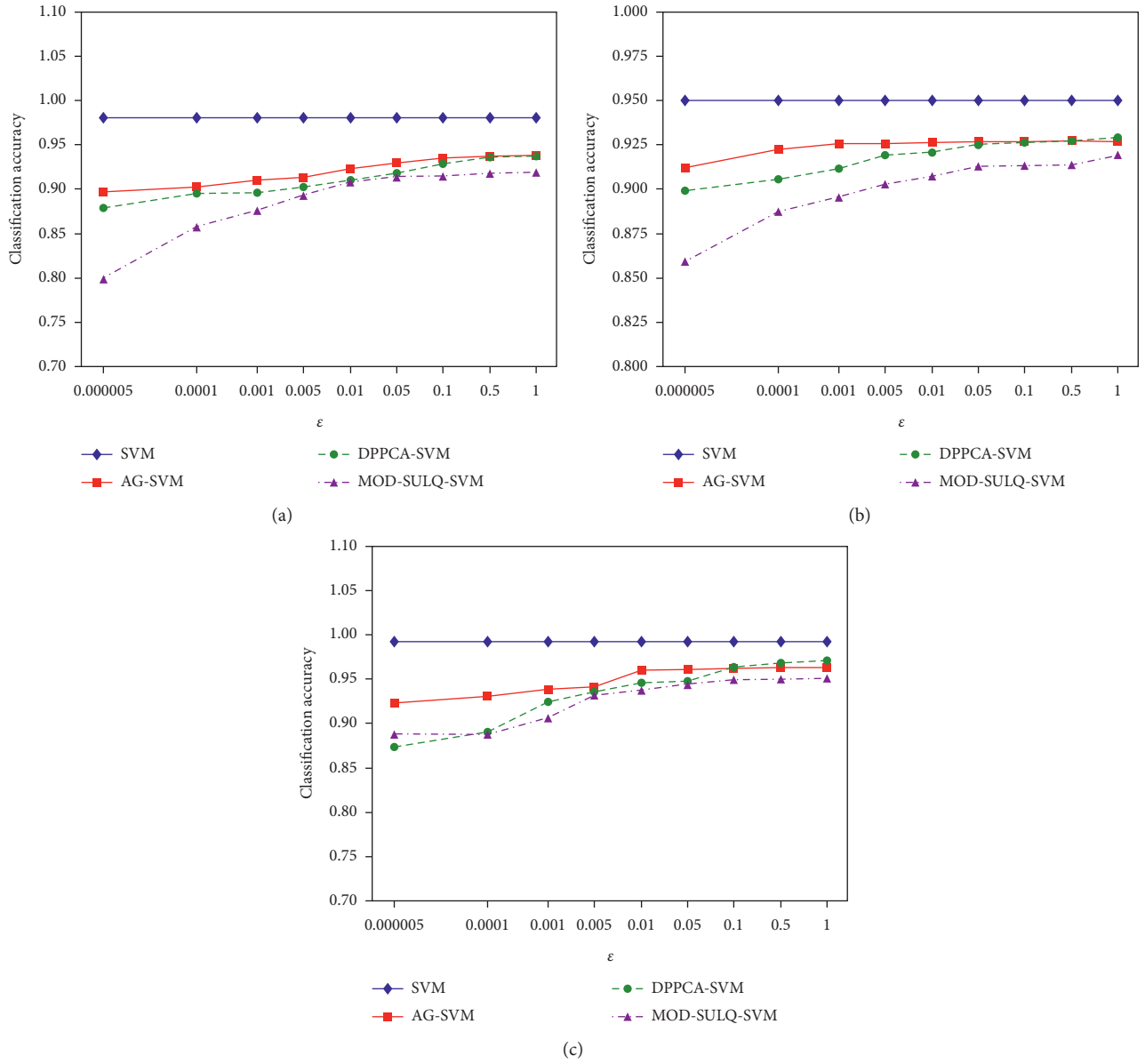

Figure 2: Variation of classification accuracy with different value of privacy budget (SVM, AG-SVM, DPPCA-SVM, and MOD-SuLQ-SVM). (a) Sensorless, (b) Covtype, and (c) Musk.

obtained. The result is shown in Figure 2. The higher the classification accuracy is, the better the usability of the algorithm is.

In Figure 2, we observe that classification accuracy is generally SVM > AG-SVM > DPPCA-SVM > MOD-SuLQ-SVM. Since no noise is added, there is no doubt that SVM has the highest classification accuracy. In addition, other algorithms achieve differential privacy by adding different noise, making classification accuracy slightly lower than that of SVM. Among them, AG-SVM and MOD-SuLQ-SVM both provide (ε, δ)−differential privacy, while DPPCA-SVM provides $(\varepsilon, 0)$−differential privacy. As we all know, $(\varepsilon, 0)$−differential privacy usually provides stronger privacy guarantee and weaker data utility than (ε, δ)−differential privacy. However, the accuracy of our DPPCA-SVM is only slightly lower than that of AG-SVM and higher than that of MOD-SuLQ-SVM in general. Moreover, when the value of privacy budget is large enough $(\varepsilon > 0.5)$, the classification accuracy of DPPCA-SVM is higher than that of AG-SVM. Therefore, compared with other algorithms, DPPCA-SVM not only provides superior privacy protection but also has relatively high data availability.

In the next experiment, we evaluate the classification accuracy of algorithms SVM, AG-SVM, DPPCA-SVM, and MOD-SuLQ-SVM when privacy budget is fixed and the number of samples changes. The privacy parameters ε and δ are set to 0.005 and $1/n^2$, respectively. The experimental results are shown in Figure 3. Above all, as the sample size increases, the classification accuracy is generally on the rise but sometimes slightly decreases due to some inappropriate samples. What is more, Figure 3 also intuitively shows the

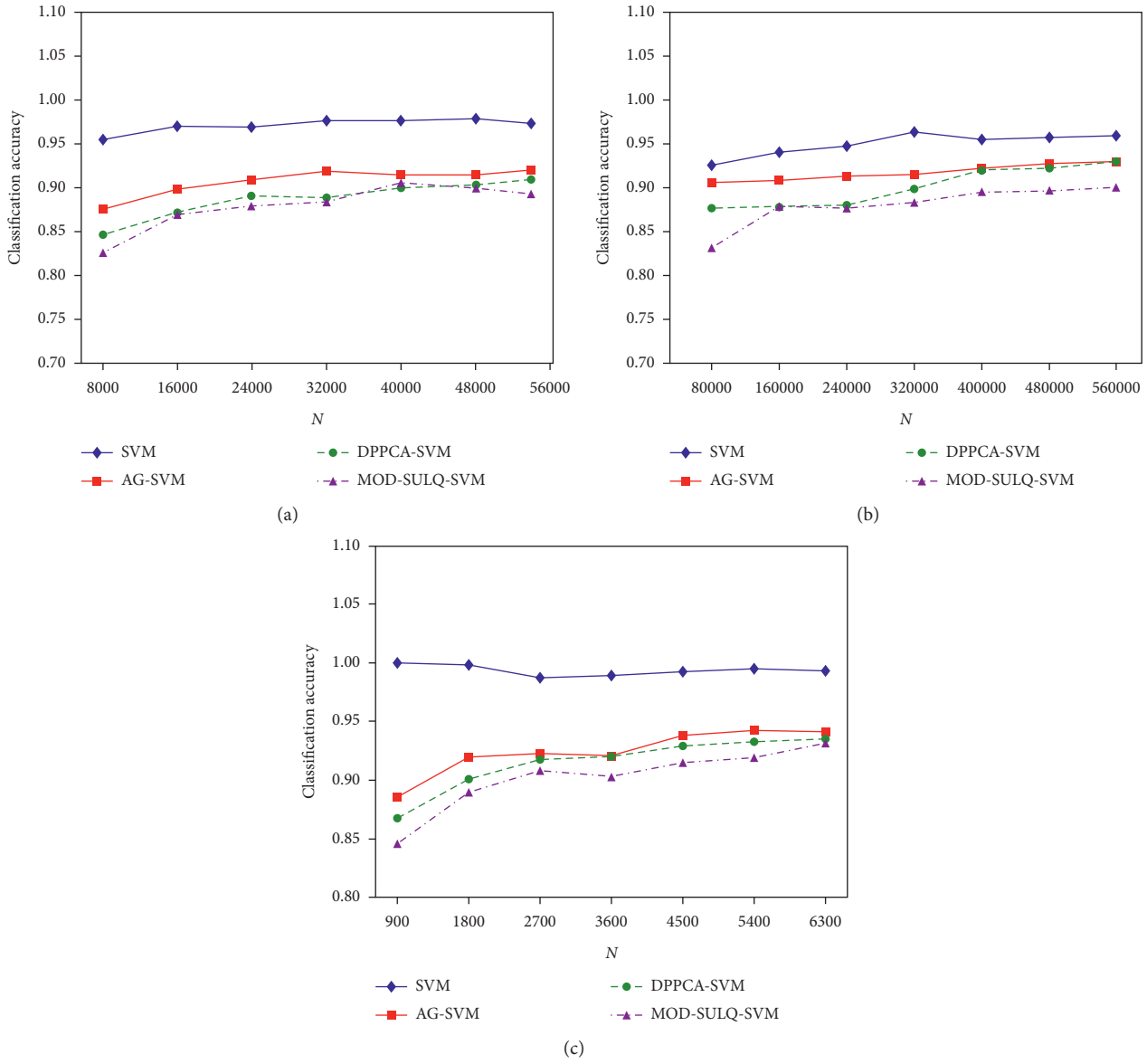

Figure 3: Variation of classification accuracy with different value of samples (SVM, AG-SVM, DPPCA-SVM, and MOD-SuLQ-SVM). (a) Sensorless, (b) Covtype, and (c) Musk.

comparison of the accuracy of DPPCA-SVM, AG-SVM, and MOD-SuLQ-SVM when the number of samples N changes. In the case of the same privacy budget, the DPPCA-SVM classification method has a high classification accuracy when the number of samples is large, which is close to the AG-SVM method. However, when the samples are relatively small, due to the stronger privacy protection, the classification effect is slightly worse than that of AG-SVM. For MOD-SuLQ-SVM, DPPCA-SVM is superior to it in terms of privacy protection and classification accuracy.

In summary, DPPCA-SVM that we proposed is compared with AG-SVM and MOD-SuLQ-SVM. These algorithms all realize differential privacy through input disturbance. However, the noise they add is quite different, which leads to a distinguishment in the usability of the results. As a result, in order to measure the performance of these algorithms, we designed the above two experiments. Among them, privacy budget and the size of samples are treated as independent variables; that is, we control one of these two parameters and adjust the other to observe changes in the data availability of these algorithms. The results show that when the number of samples is fixed and the privacy budget is small, there is still a certain gap between the classification accuracies of DPPCA-SVM and AG-SVM. However, if the privacy budget is large, the classification accuracies of them will be close to each other. For MOD-SuLQ-SVM, DPPCA-SVM is always better than it overall. In addition, when the privacy budget is fixed, as the sample size increases, the classification accuracies of these algorithms are on the rise. The classification accuracy of DPPCA-SVM is still between those of AG-SVM and MOD-SuLQ-SVM. What is more, when the number of samples is large enough,

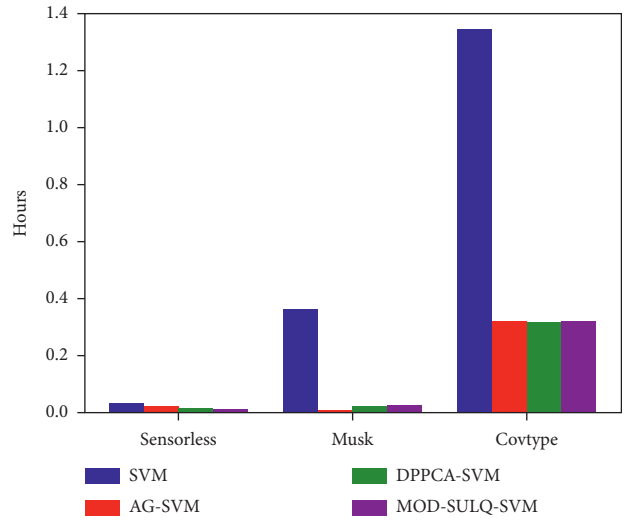

Figure 4: Running time of algorithms.

the classification accuracy of DPPCA-SVM is close to or even surpasses that of AG-SVM. Moreover, we know that DPPCA-SVM provides the privacy protection of $(\varepsilon, 0)$, while both AG-SVM and MOD-SuLQ-SVM only satisfy the differential privacy of (ε, δ). Therefore, DPPCA-SVM can achieve higher data availability while providing a higher level of privacy protection.

4.3. Experiments for Running Time of Algorithm DPPCA-SVM and Other Algorithms. In the above experiments, we obtained the data availability of these algorithms by comparing the classification accuracies of them. There is no doubt that classification accuracy is a valuable measure of algorithm performance. Among them, the SVM algorithm always maintains extremely high classification accuracy because it does not implement dimensionality reduction and noise processing. But, in practical applications, the execution efficiency of the algorithm also needs to be considered. Therefore, we compared the running times of SVM, DPPCA-SVM, AG-SVM, and MOD-SuLQ-SVM to highlight the necessity of DPPCA-SVM that we proposed. As is shown in Figure 4, in the three data sets, SVM's running time is much longer than those of the other three algorithms, especially when the data set is large. Therefore, using principal component analysis before classification can obviously save on computational complexity.

All in all, considering the three aspects of privacy protection level, data availability, and execution efficiency, DPPCA-SVM has a relatively excellent performance, which means that it will have a broad space in practical applications.

5. Conclusions

Nowadays, privacy protection algorithms have been widely applied to the field of data processing and achieved remarkable achievements. However, few researchers consider differential privacy, principal component analysis, and support vector machines at the same time and combine them together. In this paper, for fast classification and data privacy protection, we propose two algorithms that satisfy $(\varepsilon, 0)$-DP, namely, DPPCA-SVM based on input perturbation and PCA-DPSVM based on output perturbation.

In this paper, we have designed three experiments to measure the performance of DPPCA-SVM and PCA-DPSVM that we proposed. In the first experiment, we mainly demonstrated that DPPCA-SVM based on input disturbance outperforms PCA-DPSVM based on output disturbance when providing the same privacy protection. In the second experiment, DPPCA-SVM is compared with two other algorithms which are both based on input disturbance, namely, AG-SVM and MOD-SuLQ-SVM. For the data sets Sensorless, Covtype, and Musk, when the privacy budget $\varepsilon > 0.005$, the classification accuracy of DPPCA-SVM is always greater than 0.90, which is slightly lower than that of AG-SVM but higher than that of MOD-SuLQ-SVM. However, DPPCA-SVM provides $(\varepsilon, 0)$-DP privacy protection, while AG-SVM only provides (ε, δ)-DP privacy protection. In the last experiment, we verified that DPPCA-SVM can effectively reduce the data dimension and greatly shorten the processing time of data.

Therefore, considering the intensity of privacy protection, classification accuracy, and the speed of data processing, there is no doubt that DPPCA-SVM has a broader prospect in the field of data processing and machine learning. For example, when the data of user have complex attributes and urgent need for privacy protection, the concept of DPPCA-SVM has a high reference value. In addition, the DPPCA-SVM algorithm also has some limitations. For instance, in this model, the classification accuracy of the SVM algorithm is used to measure the availability of data, which may not be an optimal method. In the future, we can design a scoring function to evaluate the performance of the algorithm in many aspects.

Data Availability

The data set Sensorless used to support the findings of this study is available through visiting http://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis/. The data set Covtype used to support the findings of this study is available through visiting https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/. The data set Musk used to support the findings of this study is available through visiting http://archive.ics.uci.edu/ml/machine-learning-databases/musk/.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61872197 and 61972209), the Postgraduate Research and Practice Innovation Program of

Jiangsu Province (KYCX180891), the Natural Science Foundation of Jiangsu Province (BK20161516 and BK20160916), and the Postdoctoral Science Foundation Project of China (2016M601859).

References

[1] A. Wiesel and A. O. Hero, "Decomposable principal component analysis," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4369–4377, 2009.

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Theory of Cryptography Conference*, pp. 265–284, Springer, New York, NY, USA, March 2006.

[3] J. S. Comas, J. D. Ferrer, D. Sánchez, and D. Megías, "Individual differential privacy: a utility-preserving formulation of differential privacy guarantees," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1418–1429, 2017.

[4] F. Liu, "Generalized Gaussian mechanism for differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 747–756, 2019.

[5] A. Beimel, K. Nissim, and U. Stemmer, Edited by P. Raghavendra, Ed., Edited by S. Raskhodnikova, Ed., Edited by K. Jansen, Ed., "Private learning and sanitization: pure vs. approximate differential privacy," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, J.D.P. Rolim, Ed., Springer, Berlin, Germany, pp. 363–378, 2013.

[6] A. Blum, C. Dwork, F. McSherry et al., "Practical privacy: the SuLQ framework," in *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 128–138, ACM, New York, NY, USA, June 2005.

[7] K. Chaudhuri, A. D. Sarwate, and S. Kaushik, "A near-optimal algorithm for differentially-private principal components," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2905–2943, 2005.

[8] M. Kapralov and K. Talwar, "On differentially private low rank approximation," in *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1395–1414, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, January 2013.

[9] C. Dwork, K. Talwar, A. Thakurta et al., "Analyze gauss: optimal bounds for privacy-preserving principal component analysis," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 11–20, ACM, New York, NY, USA, May 2014.

[10] H. Imtiaz and A. D. Sarwate, "Symmetric matrix perturbation for differentially-private principal component analysis," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2339–2343, ICASSP, Shanghai, China, March 2016.

[11] H. Imtiaz and A. D. Sarwate, "Differentially private distributed principal component analysis," in *Proceediongs of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2206–2210, IEEE, Calgary, AB, Canada, April 2018.

[12] W. Jiang, C. Xie, and Z. Zhang, "Wishart mechanism for differentially private principal components analysis," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI Press, Phoenix, AZ, USA, February 2016.

[13] F. Farokhi, "Privacy-preserving public release of datasets for support vector machine classification," *IEEE Transactions on Big Data*, vol. 1, 2020.

[14] P. Bouboulis, S. Theodoridis, C. Mavroforakis, and L. E. Dalla, "Complex support vector machines for regression and quaternary classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1260–1274, 2015.

[15] 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.

[16] C. Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, no. 1, pp. 86–95, 2011.

[17] Y. Xu, G. Yang, and S. Bai, "Laplace input and output perturbation for differentially private principal components analysis," *Security and Communication Networks*, vol. 2019, no. 1, 10 pages, Article ID 9169802, 2019.

[18] http://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis/.

[19] https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/.

[20] http://archive.ics.uci.edu/ml/machine-learning-databases/musk/.