IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 22, 2019, accepted July 7, 2019, date of publication July 18, 2019, date of current version August 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929680

A Differential Privacy Support Vector Machine Classifier Based on Dual Variable Perturbation

YALING ZHANG[1], ZHIFENG HAO[1], AND SHANGPING WANG[2]
[1]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China
[2]School of Science, Xi'an University of Technology, Xi'an 710048, China

Corresponding author: Zhifeng Hao (zhifenghao123@gmail.com)

This work was supported by the Key Research and Development Program of Shaanxi Province of China under Grant No. 2019GY-028, and the Key Laboratory Research Project of Education Bureau of Shaanxi Province of China under Grant No. 16JS078.

ABSTRACT Data mining technology can be used to dig out potential and valuable information from massive data, and support vector machine (SVM) is one of the most widely used and most efficient methods in the field of data mining classification. However, the training set data often contains sensitive attributes, and the traditional training method of SVM reveals the individual privacy information. In view of the low prediction accuracy and poor versatility of the existing SVM classifiers with privacy protection, this paper proposed a new SVM training method for differential privacy protection. The algorithm first solved the dual problem of SVM by using SMO method and the difference E_i between the estimated value and the real value for each support vector was recorded. Then the ratio of the E_i of each support vector to the sum of the E_i of all the support vectors was calculated. Next, different levels of Laplace random noise were added to the corresponding dual variables α_i of each support vector to be released, according to the ratio of each support vector. According to the principle of differential privacy protection, the algorithm meets ϵ-differential privacy which can be used to effectively protect individual privacy. Experimental results on real datasets showed that the algorithm proposed in this paper could be used for classification prediction under a reasonable privacy budget.

INDEX TERMS Data mining, support vector machine, SMO, privacy protection, differential privacy.

I. INTRODUCTION

With the rapid development and wide application of the Internet technology, massive application data are accumulating at an "explosive" speed, and data mining technology can be used to find and extract the potential, regular and understandable patterns or knowledge in these massive data, and provide feedback and guidance for business and human life [1]. As an effective supervised learning model and data mining classification algorithm, the Support Vector Machine (SVM) was proposed by Vapnik et al. according to the structural risk minimization principle and the VC dimension theory in Statistical Learning Theory [2], and it provides a good theoretical guarantee for overfitting. The SVM performed well with an appropriate kernel function, and showed unique advantages in solving the problem of small samples, nonlinearity, and high dimension in data classification.

Data mining algorithms such as Support Vector Machine can be used to discover knowledge and patterns hidden in massive data, but at the same time put the individual privacy information at a risk of disclosure [3], therefore, privacy protection data mining technology has become a research focus in this field. Traditional privacy protection technologies include k-anonymity [4], l-diversity [5], m-invariance [6], t-closeness [7], and so on, which had been applied to various data mining methods by many scholars at home and abroad. However, these methods are based on the premise that the attackers had no background knowledge, which could not provide adequate security [8]. Differential Privacy (DP) was a privacy definition [9] proposed by Dwork in 2006 for the privacy disclosure of statistical database, and compared with traditional privacy protection models, the differential privacy model is defined on a solid mathematical basis and can be used to control the level of privacy protection.

In summary, the support vector machine classification algorithm based on differential privacy protection is a valuable research topic, of which the research goal is to ensure the high classification accuracy of support vector machines and to protect the individual privacy information of the training sets. The use of this algorithm is to be responsible for each

The associate editor coordinating the review of this manuscript and approving it for publication was Shenghong Li.

owner and contributor of the dataset, and makes it easier to collect more data. At the same time, the practical application of the algorithm can be promoted to obtain greater social value.

The rest of the paper is organized as follows. In Section II , related works are presented. Section III introduces some basic knowledge of differential privacy and support vector machine. Section IV presents the details design and related analysis of our scheme. In Section V, we experimentally evaluate the performance of the algorithm we proprosed. And the conclusion is made in Section VI.

II. RELATED WORK

Many works have been done for the research of the privacy problems for data mining algorithms, especially for SVM. Benjamin i. p. Rubinstein et al. proposed a support vector machine with differential privacy protection [10]. They first used the SVM algorithm, kernel function, and loss function to calculate the space vector , then calculated the result of $\boldsymbol{w} = \sum_{i=1}^{n} y_i \alpha_i \Phi(\boldsymbol{x}_i)$, with Φ as the random $2d$-dimensional feature map, and finally, obtained the corresponding vector $\boldsymbol{w} = \boldsymbol{w} + Lap(\lambda)$. Kamalika Chaudhuri et al. [11] proposed an algorithm of adding noise to the objective function *ObjectiveSVM*, which used Laplace Function to generate random noise b and added b to the objective function, and then the optimal hyperplane parameter $\boldsymbol{w}$ was solved for the objective function with noise. Haoran Li et al. proposed a mixed differential privacy protection support vector machine model [12]. The model assumed that part of the public data D_{public} was donated by users and did not need privacy protection, and there was another part of private data $D_{private}$ that needed to be protected. According to Fourier transform, D_{public} was used to calculate $\boldsymbol{\rho} = (\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \ldots, \boldsymbol{\rho}_d)^T$, and then $D_{private}$ was transformed from the original d-dimensional sample space to $2d$-dimensional feature map. After that, the dual variable $\boldsymbol{\alpha}$ was calculated in the transformed space of $D_{private}$ and finally, the results returned to $\boldsymbol{w}^* = \boldsymbol{w} + \boldsymbol{\mu}$ and $\boldsymbol{\rho}$. Prateek Jain et al. proposed a differential privacy machine learning method with kernel function [13]. This method provided three interactive models and deduced different differential privacy learning methods for each model. Finally, this method was extended to the SVM classification. Weilin Nie and Cheng Wang carried out a perturbation analysis on the algorithm of convex risk minimization [14], and applied the analysis to the differential privacy learning algorithm. Since the SVM itself is also a special risk minimization optimization problem, the authors used the method in the SVM, and gave the selection of noise parameters. A classification learning algorithm for SVM with limited training data samples was proposed in the literature [15]. In the case of limited labeled training set, the algorithm used the transductive support vector machine (TSVM) to learn from the unlabeled data, and then a label allocation pool was generated by minimizing the overall loss of labeled and unlabeled data. Out of consideration of privacy, each label allocation in the pool was evaluated and an uncertainty selection was made, and finally, an SVM classifier with differential privacy protection was generated. Han Wang et al. [16] proposed a privacy preserving support vector machine algorithm under differential privacy for multiple classification. The algorithm disturbed the kernel function in three different ways, including direct Laplace noise injection, Taylor formula replacement, and combination of previous two methods. The whole classification model was disturbed indirectly by the value of the normal vector obtained from disturbance. It was expected to protect the small sample data and not to interfere with the classification effect of the model to the whole dataset. Makhamisa Senekane [17] reported a scheme for privacy-preserving image classification using Support Vector Machine and DP. SVM was chosen as a classification algorithm because unlike variants of artificial neural networks, it converged to a global optimum. SVM kernels used were linear and Radial Basis Function (RBF), while ϵ-differential privacy was the used DP framework. The proposed scheme achieved an accuracy of up to 98%. The results underlined the utility of using SVM and DP for privacy-preserving image classification.

It can be seen from the analysis that the SVM classification algorithm based on the proposed differential privacy protection [10]–[17] has three types of problems: (1) When the training set was particularly large, the time consumption of the support vector machine prediction would be particularly large, and the noise would increase, and the accuracy would decrease. (2) The restriction on the objective function was overly strong, requiring it to remain convex and differentiable, so there was no universality. (3) The solution was limited to specific types of training set.

To solve the above problems, this paper proposed a construction scheme of differential privacy SVM classifier based on the dual variable perturbation. In the process of solving the dual problem of SVM by using the SMO [18] method, the difference E_i between the estimated value and the real value for each support vector was recorded. Then the ratio of the E_i of each support vector to the sum of the E_i of all the support vectors was calculated. Next, according to the ratio of each support vector, different levels of Laplace random noise were added to the corresponding dual variables α_i of each support vector that was calculated and to be released, and eventually, an SVM classifier with differential privacy protection was obtained. It is a novel idea to construct the differential privacy SVM classifier based on the method of dual variable perturbation.

The main contributions of this paper are as follows:

(1) We formalized the training process of SVM and analyzed the problem of individual privacy disclosure in the training of SVM classifier in the training set, and proposed a preliminary design of solving the problem by using differential privacy models.

(2) We proposed a differential privacy SVM classifier based on dual variable perturbation, and gave the specific pseudo-code of the scheme; meanwhile, we deduced the

global sensitivity when adding Laplace noise to the optimal dual variables.

(3) We analyzed theoretically that the algorithm proposed in this paper met the requirement of differential privacy protection, and proved the prediction accuracy of the algorithm through experiments. In other words, under the reasonable privacy budget setting, the algorithm proposed in this paper can maintain a high prediction accuracy.

III. PRELIMINARIES

A. DIFFERENTIAL PRIVACY

Differential privacy is a privacy protection technology based on data distortion. By adding random perturbations to the real data and calculation results, it ensured that the data privacy was under protection and meanwhile the data and calculation results kept a certain degree of usefulness.

1) DEFINITION OF DIFFERENTIAL PRIVACY

In the differential protection privacy model, it is assumed that the attacker has the largest background knowledge, that is, all the recorded information except the record he wants to attack, and he can not deduce the target record information from the published model, thus the sensitive information of individuals is not leaked.

Definition 1 (ϵ-Differential Privacy) [19]: Assumes there is a random algorithm $\boldsymbol{M}$ and $\boldsymbol{P_M}$ is the set of all possible output of $\boldsymbol{M}$. As for any two neighboring datasets $\boldsymbol{D}, \boldsymbol{D}'$ and $\boldsymbol{S_M}$, any subset of $\boldsymbol{P_M}$, if $\boldsymbol{M}$ fits the requirement below:

$$P_r[M(D) \in S_M] \leq exp(\epsilon) \times P_r[M(D') \in S_M] \tag{1}$$

$\boldsymbol{M}$ fitted the requirement of ϵ-Differential Privacy Protection.

$\boldsymbol{D}$ and $\boldsymbol{D}'$ are two neighboring datasets between which the difference is no more than one record. ϵ is called privacy protection budget. Under the same condition, the smaller the parameter ϵ is, the higher the degree of privacy protection will be. As long as the parameter ϵ is small enough, attackers can hardly distinguish dataset $\boldsymbol{D}$ and $\boldsymbol{D}'$ the query function acts on for the same output $\boldsymbol{S_M}$. But ϵ should be larger than 0.

Differential privacy protection is achieved by adding noise that obeys a specific distribution to the return value of the query function, and the amount of noise added is related to the sensitivity of the function. The sensitivity is divided into global sensitivity and local sensitivity. When adding noise to the query function, we often use the global sensitivity of the function.

Definition 2 (Global sensitivity) [20]: For any function $\boldsymbol{f}\ \boldsymbol{\mathcal{V}}\ \boldsymbol{D} \rightarrow \boldsymbol{R^d}$, the input is a dataset $\boldsymbol{D}$, and the output is a d-dimensional real vector. For any neighboring dataset $\boldsymbol{D}$ and $\boldsymbol{D}'$, the global sensitivity of function $\boldsymbol{f}$ is

$$GS_f = \max_{D,D'} \left\| f(D) - f(D') \right\|_k \tag{2}$$

where $\left\| f(D) - f(D') \right\|_k$ represent the k-norm distance between $f(D)$ and $f(D')$.

The global sensitivity of the function reflects the greatest change to the query result caused by deleting any record in the dataset, which is determined by the query function involved in the algorithm. Different algorithms and functions have different global sensitivity.

2) MECHANISM FOR IMPLEMENTING DIFFERENTIAL PRIVACY

Differential privacy model achieves privacy protection with the adding of noise. Laplace mechanism and exponential mechanism are common noise adding mechanisms.

Laplace mechanism is suitable for adding noise to numerical query results. It protects differential privacy by adding random noise following Laplace distribution to the exact results. When the location parameter of the Laplace distribution is 0 and the scale parameter of it is b, the Laplace distribution is recorded as $Lap(b)$, and the probability density function is

$$p(x) = \frac{1}{2b} exp(-\frac{|x|}{b}) \tag{3}$$

Definition 3 (Laplace mechanism) [21]: Given a dataset $\boldsymbol{D}$, suppose there is a function $\boldsymbol{f} : \boldsymbol{D} \rightarrow \boldsymbol{R^d}$ and the sensitivity is Δf, random algorithm $M(D) = f(D) + Y$ provides ϵ-Differential Privacy Protection. Where, $Y \sim Lap(\Delta f/\epsilon)$ is random noise and follows the Laplace distribution of scale parameter $\Delta f/\epsilon$.

In many practical applications, query results are entity objects (such as a scheme or a choice). In response to this situation, McSherry et al. proposed an exponential mechanism, which was suitable for adding noise to non-numerical query results.

Definition 4 (Exponential mechanism) [22]: Suppose that the input of the random algorithm M is dataset D, the output is an entity object $r \in Range$, $Range$ is the output domain of the query function, $q(D, r)$ is the availability function, and Δq is the sensitivity of the function $q(D, r)$. If the probability of selecting and outputting r from $Range$ is proportional to $exp(\frac{\epsilon q(D,r)}{2\Delta q})$, then the algorithm provides differential privacy protection.

$q(D, r) \rightarrow R$ is called the availability function of r, which is used to evaluate the quality of the output value r.

3) COMBINATION OF DIFFERENTIAL PRIVACY PROTECTION ALGORITHMS

Differential privacy protection technology has two important composition characteristics, namely sequence composition and parallel composition. Proper use of the composition characteristics in the designed algorithm can make the allocation of privacy budget more reasonable, and keep the level of privacy protection in the whole process within a given privacy protection budget ϵ.

Characteristic 1 (**Sequence composition**) [23]. Suppose there are algorithms $M_1, M_2, \ldots, M_n$, and their privacy budgets are $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$. As for the same dataset D, $M(M_1(D), M_2(D), \ldots, M_n(D))$, the combination algorithms of $M_1, M_2, \ldots, M_n$ on D, provides ϵ-differential privacy and $\epsilon = \sum_{i=1}^{n} \epsilon_i$.

Characteristic 2 (**Parallel combination**) [23]. Suppose there are random algorithms $M_1, M_2, \ldots, M_n$, and the privacy budgets are $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$. Dividing D into disjoint datasets $D_i, D_2, \ldots, D_n$, the combination algorithm $M(M_1(D_1), M_2(D_2), \ldots, M_n(D_n))$ provides ϵ-differential privacy and $\epsilon = \max(\epsilon_i)$.

B. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a binary classification model, which defines a classifier that maximizes the minimum margin in feature space. In 2-dimensional, 3-dimensional and higher-dimensional space, the classifier is correspondingly a straight line, a plane and a hyperplane. We determined the classifier by making the nearest sample as far as possible from the classifier. The problem of solving SVM classifier can be formalized as a convex quadratic programming problem, then the optimal solution of the problem can be solved by using the optimization algorithm. Finally a classifier can be obtained.

1) FORMAL DEFINITION OF SVM

Given the training set D, where n is the size of the dataset, $\boldsymbol{x}_i \in R^d$ is ith d-dimensional data sample, $y_i \in \{-1, 1\}$ is called the class label of $\boldsymbol{x}_i$. For each sample, there exists a mapping function $\Phi : \boldsymbol{x} \to \Phi(\boldsymbol{x})$, which maps $\boldsymbol{x}$ from the original input space R^d to the high-dimensional Hilbert space H, and there exists a kernel function k such that $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j)$. Suppose $\boldsymbol{w}^T \Phi(\boldsymbol{x}) + b = 0$ is the optimal partitioned hyperplane of SVM in space H.

According to the idea of maximizing classification hyperplane with minimum margin, we can construct a primitive constrained optimization problem of non-linear soft margin SVM, which is commonly used:

$$\min_{\boldsymbol{x}\in H, b\in R, \xi\in R^n} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{4}$$

$$\text{s.t. } y_i((\boldsymbol{w} * \boldsymbol{x}_i) + b) \geq 1 - \xi_i, \tag{5}$$

$$\xi_i \geq 0, \quad i = 1, \ldots n \tag{6}$$

where ξ_i is the slack variable corresponding to the ith training sample, and $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_n)^T$ describes the situation that the training set is allowed to be misclassified; $C > 0$ is the penalty factor.

After the optimal solutions $\boldsymbol{w}$ and b of the above problems were solved by an optimization algorithm, SVM classifier $f(\boldsymbol{x}) = \boldsymbol{w}^T \Phi(\boldsymbol{x}_i) + b$ was abtained.

In practice, it is very difficult to directly solve the original problems (4)-(6) of SVM. According to KKT conditions and Wolf duality theory, the original problems (4)-(6) can be transformed into the following dual problems:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{j=1}^{n}\alpha_j \tag{7}$$

$$\text{s.t. } \sum_{i=1}^{n} y_i \alpha_i = 0, \tag{8}$$

$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots n \tag{9}$$

where, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$ is the dual variable of the dual problem.

When the dual problems (7)-(9) of SVM have obtained the optimal solution, the decision function can be constructed by using support vector $SV = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$ (sample points where the dual variable satisfies $0 \leq \alpha_i \leq C$) and its dual variable $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$:

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \Phi(\boldsymbol{x}) + b = \sum_{i=1}^{l} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i)$$

When a new sample $\boldsymbol{x}_p \in R^d$ is given, its classification is predicted by the decision function $f(\boldsymbol{x}_p) = \sum_{i=1}^{l} \alpha_i y_i k(\boldsymbol{x}_p, \boldsymbol{x}_i)$. If $f(\boldsymbol{x}_p) > 0$, sample $\boldsymbol{x}_p$ belongs to class $+1$ and vice versa belongs to class -1.

2) SMO ALGORITHM

Many researchers have proposed different algorithms for solving dual problems (7)-(9), such as Chunking [24], Decomposing [25], and Sequential Minimal Optimization (SMO) [18]. SMO algorithm was proposed by John c. Platt of Microsoft Research in 1998 and became the fastest quadratic programming optimization algorithm, especially for linear SVM and sparse data.

The SMO algorithm only selects two dual variables for optimization in each iteration. Assuming that the selected two variables are α_1 and α_2 and the other variables $\alpha_i (i = 3, 4, \ldots, n)$ are fixed, the SMO can transform the sub-problems of optimization problems (7)-(9) into optimization problems with only two variables α_1 and α_2:

$$\begin{aligned}\min\ W(\alpha_1, \alpha_2) = {} & \frac{1}{2}(k_{11}\alpha_1^2 + k_{22}\alpha_2^2 + 2y_1 y_2 k_{12}\alpha_1\alpha_2) \\ & -(\alpha_1 + \alpha_2) + y_1\alpha_1 \sum_{i=3}^{n} y_i \alpha_i k_{i1} \\ & + y_2\alpha_2 \sum_{i=3}^{n} y_i \alpha_i k_{i2} + const\end{aligned} \tag{10}$$

$$\text{s.t. } y_1\alpha_1 + y_2\alpha_2 = \sum_{i=3}^{n} y_i \alpha_i, \tag{11}$$

$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots n \tag{12}$$

where $k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j), i, j = 1, 2, \ldots, n$ and *const* is a constant term without α_1 and α_2.

For optimization problems (10)-(12), SMO algorithm gives the following α_1 and α_2 optimization steps through theoretical derivation [26]:

(1) Determine the clipping range of α_2. If $y_1 = y_2$ then $L = \max(0, \alpha_1 + \alpha_2 - C)$ and $H = \max(C, \alpha_1 + \alpha_2)$. if $y_1 = -y_2$, then $L = \max(0, \alpha_1 - \alpha_2)$ and $H = \max(C - \alpha_1 + \alpha_2, C)$.

(2) Calculate $\eta = k_{11} + k_{22} - k_{12}$.

(3) If $\eta > 0$, then update $\alpha_2^{new,unc} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$ along constraint direction, and then clip

$$\alpha_2^{new} = \begin{cases} H, & if H \le \alpha_2^{new,unc} \\ \alpha_2^{new,unc}, & if H \le \alpha_2^{new,unc} \le H \\ L, & if \alpha_2^{new,unc} \le L \end{cases}$$

If $\eta = 0$, then calculate the values of the objective function $W(\alpha_2) = -\frac{1}{2}\eta\alpha_2^2 - (y_2(E_1^{old} - E_2^{old}) - \eta\alpha_2^{old})\alpha_2 + const_2$ at L and H:$Lobj = W(L)$, $Hobj = W(H)$, let $\alpha_2 = Lobj < Hobj?L : H$.

(4) Update $\alpha_1^{new} = \alpha_1^{old} + (\alpha_1^{new}\alpha_2^{old})y_1\, y_2$.

At the same time, after updating the selected two variables α_1 and α_2 in each round, it is necessary to update b to update the deviation E_i corresponding to each training point, so as to select the next sample point to be updated. The update strategy of b is as follows:

(1) Calculate $b_1 = -E_1 - y_1(\alpha_1^{new} - \alpha_1^{old})K_{11} - y_2(\alpha_2^{new} - \alpha_2^{old})K_{21} + b^{old}$

(2) Calculate $b_2 = -E_2 - y_1(\alpha_1^{new} - \alpha_1^{old})K_{12} - y_2(\alpha_2^{new} - \alpha_2^{old})K_{22} + b^{old}$

(3) Let

$$b = \begin{cases} b_1, & if\ 0 < \alpha_1^{new} < C, \alpha_2^{new} = 0\ or\ C \\ b_2, & if\ 0 < \alpha_2^{new} < C, \alpha_1^{new} = 0\ or\ C \\ b_1\ or\ b_2, & if\ 0 < \alpha_1^{new} < C, 0 < \alpha_2^{new} < C \\ \frac{b_1 + b_2}{2}, & if\ \alpha_1^{new} = 0\ or\ C, \alpha_2^{new} = 0\ or\ C \end{cases}$$

When the whole SMO algorithm converges or the number of optimization iterations reaches the maximum number of the iterations set with the program, it means that the dual problem of SVM has obtained the optimal solution.

IV. DIFFERENTIAL PRIVACY SVM CLASSIFIER BASED ON DUAL VARIABLE PERTURBATION

Aiming at overcoming the shortcomings of the existing SVM methods based on privacy protection, such as low accuracy and overly strong restriction on the objective function, a differential privacy SVM based on dual variable perturbation (DPSVMDVP) was proposed. In this section, we first explained the construction scheme of DPSVMDVP proposed from a macro perspective, then analyzed the global sensitivity to be considered when adding Laplace noise to the dual variables corresponding to the finally released support vector, and finally gave the specific pseudo-code of DPSVMDVP and the corresponding explanation.

A. OUTLINE

Provided that the corresponding information of the SVM classifier is obtained from the training set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_{i-1}, y_{i-1}), (\boldsymbol{x}_i, y_i), (\boldsymbol{x}_{i+1}, y_{i+1}), \ldots, (\boldsymbol{x}_n, y_n)\}$ and released to the public: $SV = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_l, y_l)\}$, where l represents the number of support vectors, and its corresponding dual variable $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_l\}$ and the hyperplane parameter is b. While the attacker has acquired the maximum background knowledge $D' = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_{i-1}, y_{i-1}), (\boldsymbol{x}_{i+1}, y_{i+1}), \ldots, (\boldsymbol{x}_n, y_n)\}$ of the training set D, it could be seen that the two datasets D and D' only differed in the record of (x_i, y_i) that the attacker intended to get through attacking. Therefore, according to the training result released on D and the training result on D', the attacker could deduce the piece of sample information that differed between the two datasets, which raised the issue of individual privacy disclosure.

Now, we firstly used the core idea of the SMO algorithm to solve the dual problem of the original SVM problem. In this process, we save the difference E_i between the estimated value and the real value of each training data record in updating the optimization. Eventually, after the dual variables of all training samples were optimized, the dual variable $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_l\}$ corresponding to the support vector was perturbed, namely adding corresponding Laplace noise to the corresponding α_i of each support vector $(\boldsymbol{x_i}, y_i)$. In this way, even though attackers acquired the training result information of D and D', they could not know whether the finally released $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_l\}$ was on D or D'. Therefore, changing any record in the dataset, the change of classifier parameters obtained by the algorithm would not disclose the privacy information of the dataset sample, thus met the requirement of individual privacy protection.

B. GLOBAL SENSIBILITY OF DPSVMDVP ALGORITHM

Based on the theory of differential privacy protection, this paper proposed that the information protection of the DPSVMDVP algorithm was realized by adding Laplace noise to $\boldsymbol{\alpha}$ that was released. Laplace noise is actually a series of random values that follow the Laplace distribution $Lap(\Delta f/\epsilon)$. It is closely related to the sensitivity of the function, so here we got the function sensitivity of the DPSVMDVP algorithm.

Firstly, for the following original constrained optimization problem of the support vector machines:

$$\begin{aligned} &\min_{\boldsymbol{w},b,\xi} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t. } y_i((\boldsymbol{w} * \boldsymbol{x}_i) + b) \ge 1 - \xi_i, \\ &\qquad \xi_i \ge 0, \quad i = 1, \ldots n \end{aligned}$$

We used the method of Lagrange multipliers to construct the following Lagrangian function:

$$\begin{aligned} &L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\gamma}) \\ &\quad = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\qquad - \sum_{i=1}^{n}\alpha_i(y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b) - 1 + \xi_1) - \sum_{i=1}^{n}\gamma_i\xi_i \end{aligned}$$

where $\alpha_i \ge 0, \gamma_i \ge 0$. According to Wolfe duality theorem, we solved L for the minimum value on

$\boldsymbol{w}$, b and $\boldsymbol{\xi}$.

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i x_i = 0, \quad namely\ \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0, \quad namely\ \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \boldsymbol{\xi}} = C - \alpha_i - \gamma_i = 0, \quad namely\ \alpha_i + \gamma_i = C$$

and we further obtained $0 \le \alpha_i \le C$.

Supposing that the solutions to the SVM dual problem on the two adjacent datasets D and D', namely the optimal dual variables were $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_l\}$ and $\boldsymbol{\alpha}' = \{\alpha'_1, \alpha'_2, \ldots, \alpha'_i, \ldots, \alpha'_l\}$ respectively, where l represents the number of support vectors(sample points satisfying $0 < \alpha_i < C$).

Here we constructed a vector $\boldsymbol{\alpha} - \boldsymbol{\alpha}' = (\alpha_1 - \alpha'_1, \alpha_2 - \alpha'_2, \ldots, \alpha_i - \alpha'_i, \ldots, \alpha_l - \alpha'_l)$, and the L_2 norm distance of the vector was :

$$\left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right\|_2 = \sqrt{\sum_{i=1}^{l} (\alpha_i - \alpha'_i)^2} \tag{13}$$

As $0 \le \alpha_i \le C, 0 \le \alpha'_i \le C$,$|\alpha_i - \alpha'_i| \le C$, and therefore

$$\left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right\|_2 \le \sqrt{\sum_{i=1}^{l} C^2} \tag{14}$$

Which means that the least upper bound of $\left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right\|_2$ was $sup\left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right\|_2 = \sqrt{lC^2}$.

In the end when we added the differential privacy noise, it was performed on each of the optimal dual variables, which meant it was added to each dimension of $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_l\}$, therefore, under the concept of differential privacy, for the DPSVMDVP algorithm, the sensitivity of the function was the variation of a single dimension of the vector $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_l\}$ when adding differential privacy noise. Therefore the sensitivity of the function was:

$$\Delta f = \frac{sup\left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right\|_2}{l} = \frac{\sqrt{lC^2}}{l} = \sqrt{\frac{C^2}{l}} \tag{15}$$

C. DESIGN OF DPSVMDVP ALGORITHM

In the implementation of the DPSVMDVP algorithm, the process of solving the SVM dual problem used the most efficient SMO algorithm for reference to optimize and update the dual variables of the training samples, and two dual variables were selected for the optimization of each round. At the end of the algorithm iteration, we added Laplace noise to the values of the dual variables corresponding to all the support vectors.

In each iteration optimization of the DPSVMDVP algorithm, it was necessary to find a dual variable α_i that needed to be optimized, and then find another variable α_j that is paired with α_i for optimization by the principle of maximizing the change after optimization. The process of finding the corresponding α_j according to α_i is shown in Algorithm 1 below.

Algorithm 1 innerLoop(i)

Require: i.

Ensure: flag(Represents whether α_j optimized for pairing with α_i has been found).

Calculate $E_i = \sum_{i=1}^{t} y_t \alpha_t k(\boldsymbol{x}_i, \boldsymbol{x}_t) + b - y_i$;
Loop over all training examples to find α_j that maximizes $|E_i - E_j|$;
if $|W(\alpha_i) - W(\alpha_j)| > tolerance$ **then**
    Update α_i, α_j and b with the SMO update strategy described in section III;
    return 1;
end if
for all α_j such that $\alpha_j = 0$ **do**
    if $|W(\alpha_i) - W(\alpha_j)| > tolerance$ **then**
        Update α_i, α_j and b with the SMO update strategy described in section III;
        return 1;
    end if
end for;
for $j = 1$ to n **do**
    if $|W(\alpha_i) - W(\alpha_j)| > tolerance$ **then**
        Update α_i, α_j and b with the SMO update strategy described in section III;
        return 1;
    end if
end for
return 0;

It can be seen from Algorithm 1 that to select a dual variable α_i that needed updating, we had to go through all the data samples to find a α_j that maximized $|E_i - E_j|$. If the α_j that had been found could make the objective function $W(\alpha_j) = -\frac{1}{2}\eta\alpha_j^2 - (y_j(E_i^{old} - E_j^{old}) - \eta\alpha_j^{old})\alpha_j + const_2$ decline to a certain degree, which led to $|W(\alpha_i) - W(\alpha_j)| > tolerance$, then the selected point was used as the second point to update α_i, α_j and b according to the SMO update policy described in Section III, and meanwhile, the returned value was 1. Otherwise, we had to go through all the non-boundary samples and all the training sets. If the α_j that could make the objective function $W(\alpha_j)$ decline to a certain degree was found, then α_i, α_j and b were updated according to the SMO update policy, and the returned value was 1; or else, the returned value was 0.

After solving the dual problem of SVM, adding Laplace noise to the dual variable of the corresponding support vector was the core of the DPSVMDVP algorithm. The complete pseudo-code description of the DPSVMDVP algorithm was shown in Algorithm 2 below:

It could be seen from Algorithm 2 that the solving process of the DPSVMDVP algorithm could be divided into four stages:

Algorithm 2 DPSVMDVP Algorithm

Require: $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots (\boldsymbol{x}_n, y_n)\}$, privacy budget ϵ.

Ensure: support vector $SV = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots (\boldsymbol{x}_l, y_l)\}$ and its dual variable $\boldsymbol{\alpha}^* = \{\alpha_1^*, \alpha_2^*, \ldots, \alpha_i^*, \ldots, \alpha_l^*\}$, classification hyperplane parameter b.

1: Initialize:$\alpha_1 = \alpha_2 = \cdots = \alpha_n = 0$, $b = 0$, $C = 1$, $iter = 0$, $tolerance = 0.001$, $maxIter = \max(10000000, n)$, $numChanged = 0$, $examineAll = 1$.
2: **while** $(iter < maxIter)$ and $(numChanged > 0 | examineAll == 1)$ **do**
3: $numChanged = 0$;
4: **if** $examineAll == 1$ **then**
5: **for** $i = 1$ to n **do**
6: Calculate $E_i = \sum_{i=1}^{t} y_t \alpha_t k(\boldsymbol{x}_i, \boldsymbol{x}_t) + b - y_i$;
7: **if** ($y_i E_i < -tolerance$ and α_i<C) or($y_i E_i > tolerance$ and $\alpha_i > C$) **then**
8: $numChanged +=$ innerLoop (i);
9: **end if**
10: **end for**
11: **else**
12: **for all** α_i such that $0 < \alpha_i < C$ **do**
13: **if** $|y_i E_i| < tolerance$ **then**
14: $numChanged +=$ innerLoop (i);
15: **end if**
16: **end for**;
17: **end if**
18: **if** $examineAll == 1$ **then**
19: $examineAll = 0$;
20: **else**
21: $examineAll = 1$;
22: **end if**
23: **end while**
24: **for all** α_i such that $0 < \alpha_i < C$ **do**
25: $E_i' = \frac{E_i}{\sum_{j=1}^{l} E_j}, i = 1, 2, \cdots l, l$ is the number of support vectors.
26: $\alpha_i^* = \alpha_i + Lap(\frac{\Delta f}{\epsilon_i}), where \Delta f = \sqrt{\frac{C^2}{l}}, \epsilon_i = E_i' \cdot \epsilon$;
27: **end for**;
28: Output support vector $SV = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots (\boldsymbol{x}_l, y_l)\}$ and its dual variable $\boldsymbol{\alpha}^* = \{\alpha_1{}^*, \alpha_2{}^*, \ldots, \alpha_l{}^*\}$, classification hyperplane parameter b.

1) INITIALIZATION STAGE

Step (1) was the exact initialization stage of the DPSVMDVP algorithm. In this stage, the dual variable values of all variables needed to be initialized: $\alpha_1 = \alpha_2 = \cdots = \alpha_n = 0$, the hyperplane parameter $b = 0$, the current number of iteration $iter = 0$; meanwhile, a penalty factor $C = 1$ was set with the loose scope of KKT conditional stop criterion $tolerance = 0.001$, and the maximum of iteration $maxIter = \max(10000000, n)$.

2) ITERATIVE OPTIMIZATION STAGE

this stage was from Step (2) to Step (23) of the DPSVMDVP algorithm, and the whole iterative optimization and update process were based on the idea of the SMO algorithm. Step (2) controlled the number of iterations. When the optimization had not reached convergence and the number of iterations iter had reached the maximum $maxIter$, the algorithm would terminate the optimization process. Steps (4) to (17) endlessly switched between going through the entire dataset and going through the training points corresponding to the support vectors within the boundary, so as to select a sample point $(\boldsymbol{x}_i, y_i)$ that violated the KKT conditions, and then the algorithm called innerLoop(i) of Algorithm 1 found another sample point $(\boldsymbol{x}_j, y_j)$ that paired with it for optimization and updating. Steps (18) to (22) controlled the search for the first sample point that violated KKT conditions on that part of the dataset.

3) PERTURBATION STAGE

Steps (24) to (27) were the perturbation stage. Step (25) was to calculate the ratio E_i' of the difference E_i corresponding to each support vector sample point in the optimization process to the sum of the difference E_i of all support vector points; Step (26) was to perturb the dual variable α_i of each support vector sample point $(\boldsymbol{x}_i, y_i)$, that was, to add noise subject to Laplace distribution $Lap(\Delta/\epsilon)$. Obviously, the privacy budget allocated to each dual variable α_i was different here, and its value was related to the value of E_i'.

4) OUTPUT STAGE

Step (28) was to output the support vector points set $SV = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots (\boldsymbol{x}_l, y_l)\}$, their perturbed dual variables $\boldsymbol{\alpha}^* = \{\alpha_1{}^*, \alpha_2{}^*, \ldots, \alpha_i{}^*, \ldots, \alpha_l{}^*\}$, and their hyperplane parameters b.

At this point, the training stage was over. Actually, the following decision function could be obtained:$f(\boldsymbol{x}) = \sum_{i=1}^{l} \alpha_i{}^* y_i k(\boldsymbol{x}, \boldsymbol{x}_i)$. This decision function, also known as a hyperplane, could well classify the prediction samples and protect the individual privacy information of the training set.

For a sample that required a new prediction, we put its corresponding value into the decision function to make the prediction.

D. ALGORITHM PRIVACY ANALYSIS

According to preliminary of differential privacy mentioned in Section III, as long as the noise adding method used in the algorithm accorded with differential privacy, and the privacy budget allocation satisfied the characteristics of differential privacy when adding noise to all calculation results, the whole algorithm would satisfy differential privacy. This section proved that DPSVMDVP algorithm satisfied differential privacy strictly by theorem 1.

Theorem 1: DPSVMDVP algorithm satisfied differential privacy.

Proof: As shown in algorithm 2 in section IV, DPSVMDVP allocated a privacy budget of ϵ, and the final addition of Laplacian noise was performed by $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_l\}$ on each dual variable in the vector space composed of all support vector point dual variables. Therefore, the dual variable α_i corresponding to the ith support vector point $(\boldsymbol{x}_i, y_i)$ would be analyzed.

As described in Section IV, the difference between the estimated value corresponded to the ith support vector point $(\boldsymbol{x}_i, y_i)$ and the true value was E_i, and the ratio of the difference value E_i of the support vector point to the sum of the difference value of all the support vector points was:

$$E_i' = \frac{E_i}{\sum_{j=1}^{l} E_j} \tag{16}$$

The privacy budget assigned to the ith support vector point was:

$$\epsilon_i = E_i' \cdot \epsilon \tag{17}$$

At output α_i, a random noise of Laplace distribution $Lap(\frac{\Delta f}{\epsilon_i})$ was added to it, where Δf was the global sensitivity $\Delta f = \sqrt{\frac{C^2}{l}}$ in Section IV. According to the Laplace mechanism described in Definition 3 of Section III, the output value $\boldsymbol{\alpha}^* = \{\alpha_1{}^*, \alpha_2{}^*, \ldots, \alpha_i{}^*, \ldots, \alpha_l{}^*\}$ satisfied the ϵ_i-differential privacy.

For the DPSVMDVP algorithm, when the Laplacian noise was added to the dual variable corresponding to all the support vector points, the true privacy budget allocation satisfied:

$$\begin{aligned}\sum_{i=1}^{l} \epsilon_i &= \left(\frac{E_1}{\sum_{j=1}^{l} E_j} + \frac{E_1}{\sum_{j=1}^{2} E_j} + \cdots + \frac{E_l}{\sum_{j=1}^{l} E_j}\right) \cdot \epsilon \\ &= \left(\frac{\sum_{i=1}^{l} E_i}{\sum_{j=1}^{l} E_j}\right) \cdot \epsilon \\ &= \epsilon \end{aligned} \tag{18}$$

Among them, $i = 1, 2, \ldots, l$, output value $\alpha_1{}^*, \alpha_2{}^*, \ldots, \alpha_l{}^*$ satisfied ϵ_1-differential privacy, ϵ_2-differential privacy ,..., ϵ_l-differential privacy respectively.

For the entire DPSVMDVP algorithm, $\boldsymbol{\alpha}^* = (\alpha_1{}^*, \alpha_2{}^*, \ldots, \alpha_l{}^*)$ was the final output vector space; meanwhile, DPSVMDVP was performed on the entire dataset D when solving each of the optimal dual variables, but the noise added to $\alpha_1, \alpha_2, \ldots, \alpha_l$ varied with the parameters, these different noise-adding processes could be regarded as different sub-algorithms satisfying differential privacy. It could be seen that the training process of each SVM sub-algorithms in the DPSVMDVP algorithm strictly satisfied the Characteristic 1 described in Section III, that is, the sequence combination of differential privacy; combined with the formula (18), it could be seen as that the DPSVMDVP algorithm satisfied the differential privacy.

The proof was completed.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the DPSVMDVP algorithm proposed in this paper was evaluated experimentally and compared with the SVM that did not have differential privacy protection and the PrivateSVM. The usability and operational efficiency of the DPSVMDVP algorithm proposed in this paper were verified experimentally and analyzed accordingly. To carry out the experiment efficiently, we borrowed the SVM code implementation method LIBSVM [27] of Professor Lin Chih-Jen of Taiwan University .

A. EXPERIMENTAL ENVIRONMENT

1) HARDWARE ENVIRONMENT

The processor of the main machine was Intel (R) Core (TM) i5-4590 CPU @3.30GHz, and the RAM capacity was 4.00 GB.

2) SOFTWARE ENVIRONMENT

The operating system of the experimental platform was Windows 7 64-bit Operating System, the integrated development environment for the program was Eclipse4.3+Jdk 1.7.0_2 5, and the algorithm was implemented in Java language.

B. EXPERIMENTAL DATASET

The datasets selected for the experiment were the ''cod-rna'' dataset and ''splice'' dataset, both of which were the preprocessed datasets from the homepage of Professor Lin Chih-Jen of Taiwan University (https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html). Among them, the ''cod-rna'' dataset was the processed dataset from Detection of Non-coding RNAs on the Basis of Predicted Secondary Structure Formation Free Energy Change by Andrew V Uzilov, Joshua M Keegan, and David H Mathews. The ''splice'' dataset was from the ''splice'' dataset under the classified dataset on the dataset website of Delve (http://www.cs.toronto.edu/ delve/data/datasets.html). The dataset used to identify two types of splicing junctions in DNA sequences was from the UCI machine learning database repository. The basic information of the two datasets is shown in Table 1 below.

TABLE 1. Basic information of the two datasets.

DataSet	Number of features	Number of classification	Samples in training set	Samples in test set
"cod-rna"	8	2	59535	271617
"splice"	123	2	2175	1000

C. ALGORITHM PERFORMANCE EXPERIMENT

In this section, we respectively studied the influence of the three factors, namely, the size of privacy budget, the size of training set and the feature number of training set, on the performance of the DPSVMDVP algorithm. The performance indexes of DVPDPSVM included prediction accuracy, training time and prediction time. At the same time, we also

experimentally implemented two algorithms, standard SVM and Private SVM, and made an intuitive comparison and theoretical analysis by using line chart and histograms.

Accuracy was an important indicator to evaluate the usability of the classification algorithm. It was the ratio of the number of samples correctly predicted to the total number of samples in the test set, which could be formally defined as:

$$AccuracyRate = \frac{Accurate(TestData)}{Total(TestData)} \times 100\% \quad (19)$$

where $Accurate(TestData)$ is the number of samples correctly predicted in the test set and $Total(TestData)$ is the total number of samples in the test set. The higher the accuracy, the better the usability of the classification algorithm.

In this section, we respectively studied the influence of the three factors, namely, the size of privacy budget, the size of training set and the feature number of training set, on the usability of the DPSVMDVP algorithm, and conduct experimental comparison and theoretical analysis against the standard SVM and Private algorithms.

Training time referred to the time required to run the algorithm and generate the classification models based on the training set. The shorter the training time, the shorter the training model time.

Prediction time is the time required to predict all samples in the testing dataset according to the classification model generated in the training stage. The shorter the prediction time, the shorter the prediction time.

Since the Laplace noise values added to the SVM classification parameters were a series of random values subject to a specific distribution under the differential privacy protection mechanism, to get a stable result, we conducted three experiments on SVM, PrivateSVM and DPSVMDVP algorithms with the same parameter under each experiment type, and took the average accuracy of the three experiments as the final value.

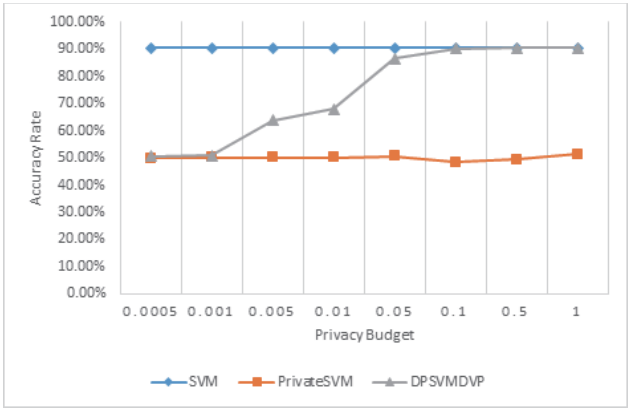
1) INFLUENCE OF PRIVACY BUDGET ON ALGORITHM PERFORMANCE

To evaluate the influence of privacy budget on algorithm usability, we fixed the number of samples and the number of features of the training set here. The privacy budget value were set at 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 and 1 successively. For each privacy budget value, we conducted three experiments by using the SVM and PrivateSVM algorithms respectively, and took the mean values of the three experiments. The results of running in the two datasets are shown in FIGURE.1 and FIGURE.2 below.

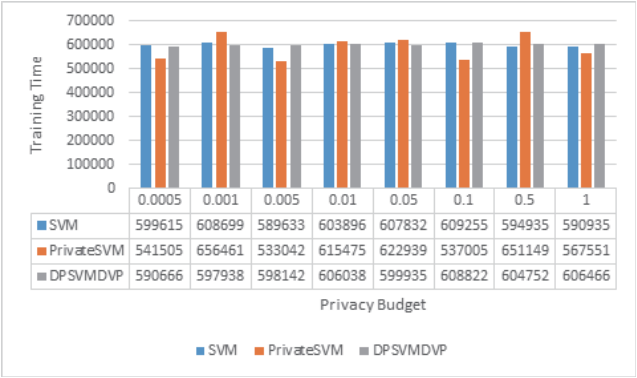
As could be seen in FIGURE.1 and FIGURE.2:

(1) The larger the privacy budget, the higher the accuracy of the DPSVMDVP algorithm. This was mainly due to the fact that the larger the privacy budget ϵ was, the smaller the Laplace noise perturbation needed to be added, which would have smaller influence on the accuracy of the SVM classifier model. Meanwhile, it could be seen in FIGURE. 1(a) that when the privacy budget of the "cod-rna" dataset was lower than 0.001, the accuracy rate of the DPSVMDVP algorithm remained at 50%, which was roughly same to that of the PrivateSVM. However, when the privacy budget ϵ increased from 0.001 to 0.05, the accuracy became higher and higher; when it reached 0.05, its accuracy could reach to the accuracy level of the standard SVM. The same result could be obtained in FIGURE. 2(a), but the difference was that in the "splice" dataset, the two inflection points of privacy budget that made the accuracy change were 0.01 and 0.5, respectively.

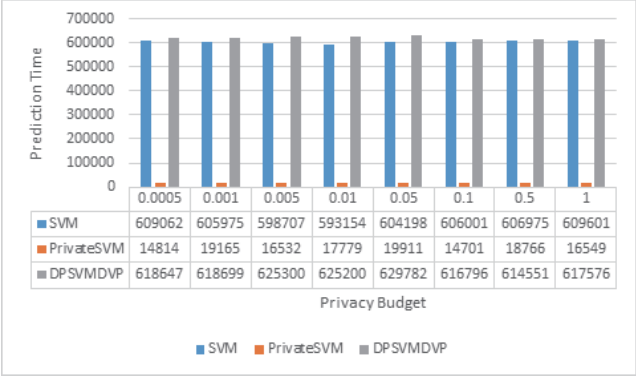
(2) Privacy budget ϵ had little effect on the training time and prediction time of DVPDPSVM algorithm.



(a) Accuracy rate of the algorithm with different privacy budget

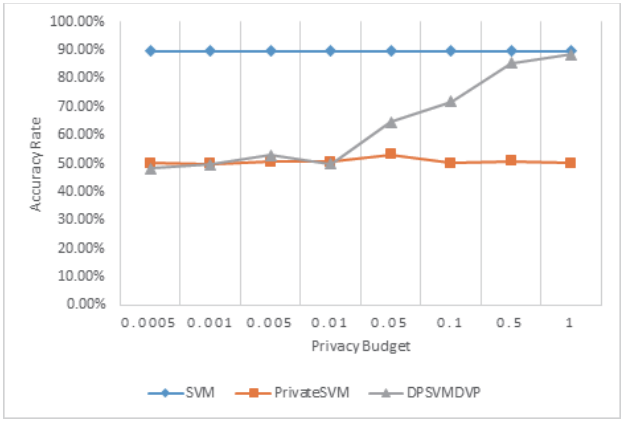


(b) Training time of the algorithm with different privacy budget

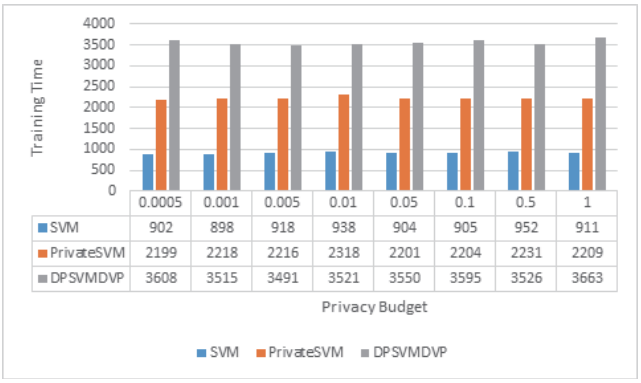


(c) Prediction time of the algorithm with different privacy budget

FIGURE 1. Performance of the algorithm with different privacy budget in the "cod-rna" dataset.

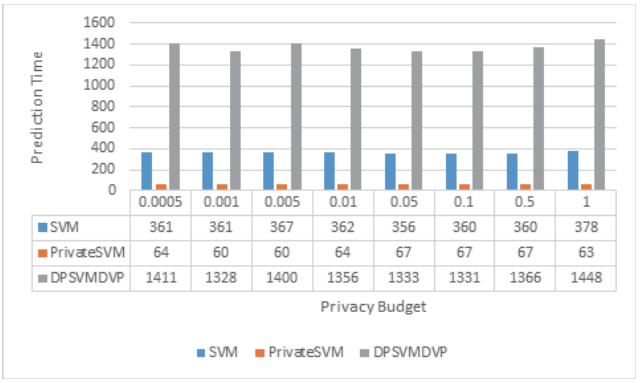


(a) Accuracy rate of the algorithm with different privacy budget



	0.0005	0.001	0.005	0.01	0.05	0.1	0.5	1
SVM	902	898	918	938	904	905	952	911
PrivateSVM	2199	2218	2216	2318	2201	2204	2231	2209
DPSVMDVP	3608	3515	3491	3521	3550	3595	3526	3663

(b) Training time of the algorithm with different privacy budget



	0.0005	0.001	0.005	0.01	0.05	0.1	0.5	1
SVM	361	361	367	362	356	360	360	378
PrivateSVM	64	60	60	64	67	67	67	63
DPSVMDVP	1411	1328	1400	1356	1333	1331	1366	1448

(c) Prediction time of the algorithm with different privacy budget

FIGURE 2. Performance of the algorithm with different privacy budget in the "splice" dataset.

From FIGURE. 1(b), FIGURE. 1(c) and FIGURE. 2(b) and FIGURE. 2(c), it could be seen that on "cod-rna" dataset, the training time and prediction time of DVPDPSVM algorithm were about 600000ms and 620000ms respectively, and on "splice" dataset, they fluctuated around 3600ms and 1400ms respectively. At the same time, it could be seen that on the two datasets, the training time and prediction time of DVPDPSVM algorithm were slightly higher than that of the standard SVM. This was because DVPDPSVM algorithm also needed a certain time when Laplace noise was added. The time in the prediction stage of PrivateSVM was very low, because PrivateSVM only needed to calculate with hyperplane parameters $\boldsymbol{w}$ and b when predicting samples, while for DVPDPSVM algorithm and standard SVM algorithm, the original feature data and dual variables of all support vector points were required for calculation.

2) INFLUENCE OF TRAINING SET SIZE ON ALGORITHM PERFORMANCE

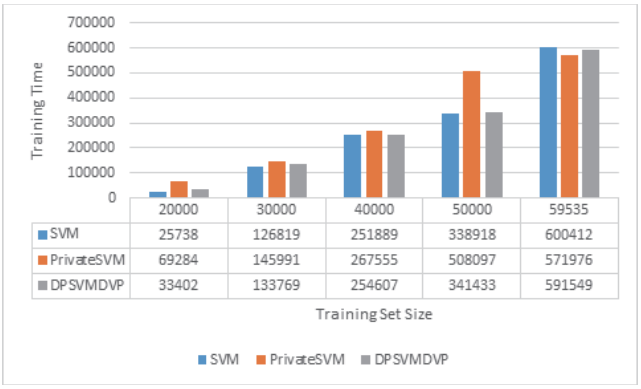
The size of the training set is an important index affecting algorithm performance. Therefore, we fixed the feature number of the training set, set the privacy budget value at 0.1 in the "cod-rna" dataset and "splice" dataset respectively according to the experimental results in Influence of Privacy Budget on Algorithm Performance, and gradually increased the number of the training set samples, namely, took numbers of 20,000, 30,000, 40,000, 50,000, and 59,535 successively in the "cod-rna" dataset and numbers of 1,300, 1,600, 1,900, and 2,175 successively in the "splice" dataset. For each experiment with a fixed sample size, we conducted three experiments by using the SVM, PrivateSVM and DPSVMDVP algorithms respectively, and took the mean value of the three experiments. The results of running in the two datasets are shown in FIGURE. 3 and FIGURE. 4 below.

As could be seen in FIGURE. 3 and FIGURE. 4:

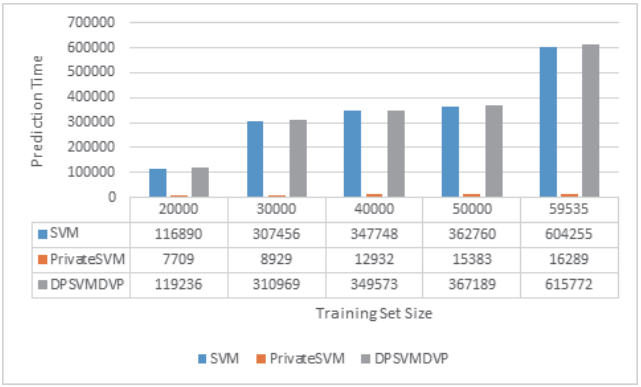
(1) With the increase in the number of training set samples, the accuracy of the DPSVMDVP algorithm had been steadily improved. The larger the sample size of the training set, the richer the data features of the entire dataset, and the more accurate the final classification model. As could be seen in FIGURE. 3(a), in the "cod-rna" dataset, when the sample size of the training set increased from 20,000 to 50,000, the prediction accuracy also increased from 72% to about 80%. Furthermore, when the sample size was greater than 50,000, the accuracy rate of the algorithm could be as high as 90%, which was basically consistent with the accuracy and growth trend of the SVM (The line graphs of the two algorithms basically overlapped in the figure). In contrast, the accuracy of the PrivateSVM changed particularly slightly with the number of training set samples, and its accuracy remained within the range of 50% to 60%. However, in the "splice" dataset, the prediction accuracy increased from 74% to about 78% when the size of the training set samples increased from 1,300 to 2,175.

(2) The more samples in the training set, the longer it took for the training and prediction phases of DVPDPSVM algorithm. The linear growth was observed. The more samples the training set had, the higher the number of times that the dual variable values of each sample needed to be updated and iterated in the algorithm, which resulted in the longer time needed to determine the support vector sample points, therefore, longer the training time was needed. With the increase of the number of samples in the training set, the more support vector sample points that ultimately supported the SVM classifier leading to the longer time needed to predict the test set samples.

(a) Accuracy rate of the algorithm with different training set size

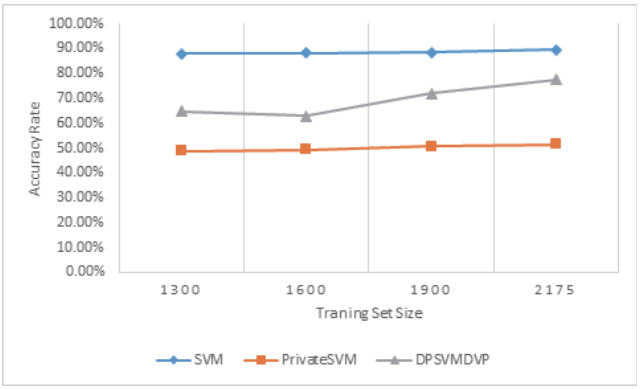


(b) Training time of the algorithm with different training set size

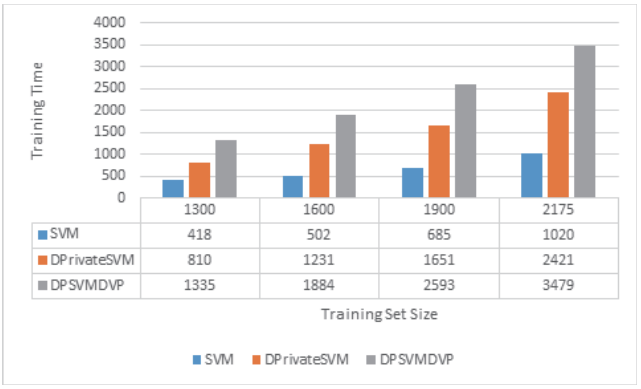


(c) Prediction time of the algorithm with different training set size

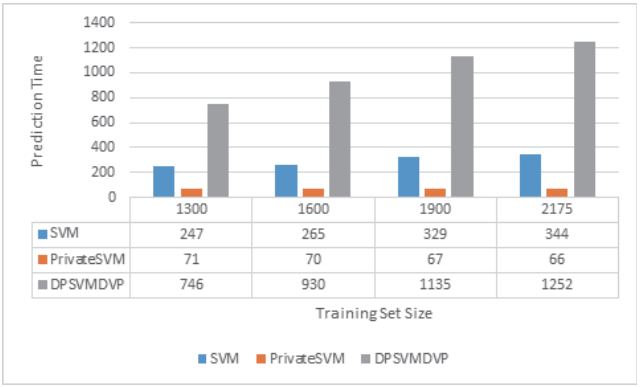
FIGURE 3. Performance of the algorithm with different training set size in the "cod-rna" dataset.



(a) Accuracy rate of the algorithm with different training set size



(b) Training time of the algorithm with different training set size



(c) Prediction time of the algorithm with different training set size

FIGURE 4. Performance of the algorithm with different training set size in the "splice" dataset.

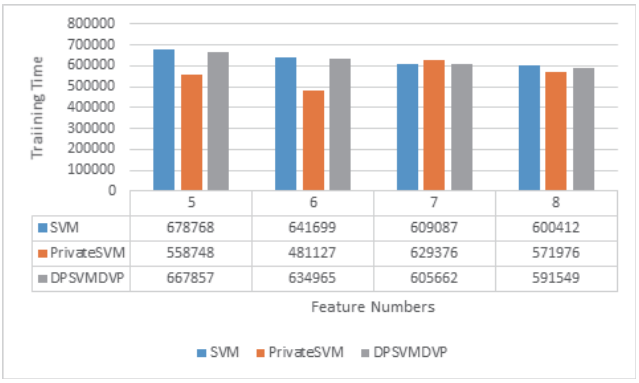
3) INFLUENCE OF TRAINING SET FEATURE NUMBER ON ALGORITHM PERFORMANCE

The number of features of a data sample represents the richness of the data sample, so in this section, we investigated the influence of feature number on the algorithm performance. First, we fixed the size of the training set as the total number of samples, set the sizes of the "cod-rna" dataset and "splice" dataset at 59,535 and 2,175 respectively, and fixed the privacy budget value at 0.1 respectively. Then we gradually increased the number of features of the training set, namely, taking 5, 6, 7, and 8 successively as the feature numbers in the "cod-rna" dataset, and 55, 56, 57, 58, 59, and 60 successively as the feature numbers in the "splice" dataset. For each experiment with a fixed sample size, we conducted three experiments by using the SVM, PrivateSVM and DPSVMDVP algorithms respectively, and took the mean value of the three experiments. The results of running in the two datasets are shown in FIGURE. 5 and FIGURE. 6 below.

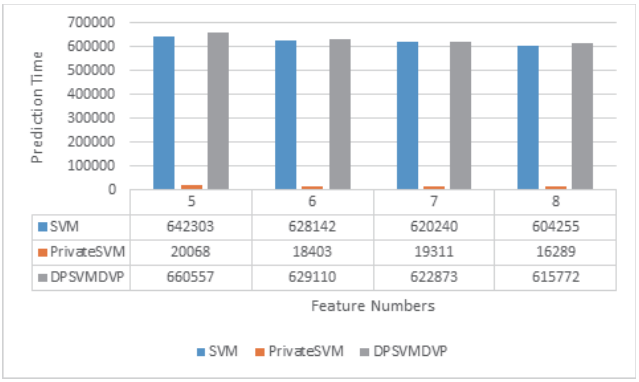
It could be seen in FIGURE. 5 and FIGURE. 6 that:

(1) With the increase in the number of training set sample features, the accuracy of the DPSVMDVP algorithm had

(a) Accuracy rate of the algorithm with different feature numbers



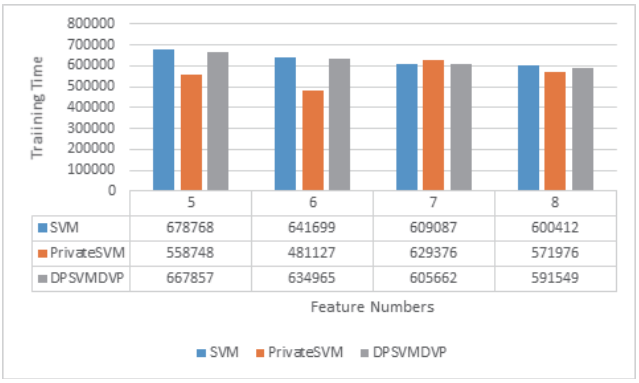
(b) Training time of the algorithm with different feature numbers



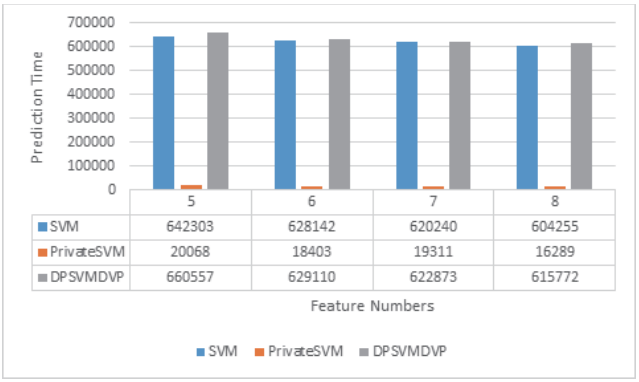
(c) Prediction time of the algorithm with different feature numbers

FIGURE 5. Performance of the algorithm with different feature numbers in the "cod-rna" dataset.

(a) Accuracy rate of the algorithm with different feature numbers



(b) Training time of the algorithm with different feature numbers



(c) Prediction time of the algorithm with different feature numbers

FIGURE 6. Performance of the algorithm with different feature numbers in the "splice" dataset.

been steadily improved. As seen in FIGURE. 5(a), when the sample size of the training set increased from 5 to 8 in the "cod-rna" dataset, the prediction accuracy rate increased from 90.22% to 90.48%, which was weak and not obvious. However, as seen in FIGURE. 6(a), when the number of sample features increased from 55 to 60 in the "splice" dataset, the prediction accuracy rate of the algorithm obviously increased from 64.13% to 77.57%.

(2) For different datasets, the number of sample features had different effects on the training and prediction phases of DVPDPSVM algorithm. It could be seen from FIGURE. 5(b) and FIGURE. 5(c) that the training time of "cod-rna" dataset decreased to a certain extent with the increase of the feature number of the sample set. This may be because the convergence of the algorithm was accelerated with the increase of the feature number of the sample set in "cod-rna" dataset, and the reasonable data sample points were judged to be support vector points, so the training time and prediction time were needed. The intervals were reduced. However, as is shown in FIGURE. 6(b) and FIGURE. 6(c), the training time and prediction time for "splice" dataset increased to a certain extent when the number of sample features increased from 55 to 60, which may be just the opposite of that for "cod-rna" dataset. With the increase of the number of sample features,

more time was needed for the algorithm to converge and more sample points of support vector were determined. Gradually, the time it ultimately took was longer and longer.

By analyzing the above three sets of comparative experiments, we could draw such a conclusion: The DPSVMDVP algorithm proposed in this paper can be used to achieve higher prediction accuracy under reasonable privacy budget. When the privacy budget reached a certain value, it could show an accuracy that was particularly close to the standard SVM algorithm, which was much higher than that of the PrivateSVM algorithm. The training time and prediction time of the DVPDPSVM algorithm were slightly higher than those of the standard SVM algorithm, and the training time and prediction time of the PrivateSVM were the shortest. This is because the algorithm mechanism of PrivateSVM is different from that of the standard SVM and the algorithm in this paper. In the training stage, the PrivateSVM algorithm directly mapped the original data into 2D space with the use of Fourier transformation, and then conducted the training solution in the mapped 2D space. The hyperplane parameters $\boldsymbol{w}$ and b of the support vector machine were then obtained through the operation of the calculated support vector sample points and the corresponding optimal dual variable values, and finally only these two hyperplane parameters were saved to the model file. However, the standard SVM and DPSVMDVP algorithms made full use of the kernel techniques and used the selected kernel functions in the original space for the training solution, and they finally needed to save all the support vector sample points and their corresponding dual variable value outputs into the model file. Since the size of the two datasets was large, and there were many support vector sample points, it took more time to save the model parameters into the model file. Similarly, compared with the PrivateSVM algorithm, which only reads two hyperplane parameters in the prediction stage, the standard SVM and DPSVMDVP algorithms needed to read more support vector sample point information for the prediction calculation, so they also took more time.

To sum up, Under a reasonable privacy budget, the classification accuracy of DPSVMDVP algorithm in this paper is close to that of the standard SVM algorithm, which is much higher than that of PrivateSVM algorithm. Although the training time and prediction time of DPSVMDVP algorithm are slightly higher than that of the standard SVM algorithm, the difference is only about 20 seconds, which is acceptable. Therefore, in consideration of the privacy protection of the training set, in order to obtain better classification availability, the DPSVMDVP algorithm can be used as a better SVM classification algorithm with privacy protection.

VI. CONCLUSION

In this paper, the DPSVMDVP method is proposed. In this scheme, the dual problem of the original SVM problem was firstly solved by using the core idea of the SMO algorithm. After the iteration was completed, the corresponding Laplace noise was injected into the dual variable α_i of each support vector sample point $(\boldsymbol{x}_i, y_i)$. According to the privacy analysis, the algorithm met the definition of differential privacy. Therefore, the SVM information released by this algorithm could be used not only to conduct classification prediction, but also to achieve the purpose of individual privacy protection. Finally, experiments on the two datasets of ''cod-rna'' and ''splice,'' showed that the proposed scheme still had a high classification accuracy rate when the privacy budget was reasonably set, namely when the privacy protection level remained at a certain level, the proposed scheme still had a high classification accuracy.

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, ''Introduction,'' in *Data Mining* (The Morgan Kaufmann Series in Data Management Systems), J. Han, M. Kamber, and J. Pei, Eds., 3rd ed. Boston, MA, USA: Morgan Kaufmann, 2012, pp. 1–38. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123814791000010

[2] H. Li, S. Wang, and F. Qi, ''SVM model selection with the VC bound,'' in *Computational and Information Science* (Lecture Notes in Computer Science), vol. 3314. Berlin, Germany: Springer, 2004, pp. 1067–1071.

[3] R. Mendes and J. P. Vilela, ''Privacy-preserving data mining: Methods, metrics, and applications,'' *IEEE Access*, vol. 5, pp. 10562–10582, 2017.

[4] L. Sweeney, ''K-anonymity: A model for protecting privacy,'' *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[5] G. Yang, J. Li, S. Zhang, and Y. Li, ''An enhanced *l*-diversity privacy preservation,'' in *Proc. Int. Conf. Fuzzy Syst. Knowl. Discovery*, 2014, pp. 1115–1120.

[6] M. Hochman, ''Geometric rigidity of times-m invariant measures,'' *J. Eur. Math. Soc.*, vol. 14, no. 5, pp. 1539–1563, 2012.

[7] J. Soria-Comas and J. Domingo-Ferrert, ''Differential privacy via t-closeness in data publishing,'' in *Proc. IEEE 11th Annu. Conf. Privacy, Secur. Trust*, Jul. 2013, pp. 27–35.

[8] C. Dwork, ''Differential privacy: A survey of results,'' in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Berlin, Germany: Springer, 2008, pp. 1–19.

[9] C. Dwork, ''Differential privacy,'' *Lect. Notes Comput. Sci.*, vol. 26, no. 2, pp. 1–12, 2006.

[10] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, ''Learning in a large function space: Privacy-preserving mechanisms for SVM learning,'' 2012, *arXiv:0911.5708*. [Online]. Available: https://arxiv.org/abs/0911.5708

[11] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, ''Differentially private empirical risk minimization,'' *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Mar. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1953048.2021036

[12] H. Li, L. Xiong, L. Ohno-Machado, and X. Jiang, ''Privacy preserving RBF kernel support vector machine,'' *BioMed Res. Int.*, vol. 2014, Jun. 2014, Art. no. 827371.

[13] P. Jain and A. Thakurta, ''Differentially private learning with kernels,'' in *Proc. ICML*, vol. 28, Jan. 2013, pp. 118–126.

[14] W. Nie and C. Wang, ''Perturbation of convex risk minimization and its application in differential private learning algorithms,'' *J. Inequal. Appl.*, vol. 2017, no. 1, p. 9, 2017.

[15] X. Liu, Q. Li, and T. Li, ''Private classification with limited labeled data,'' *Knowl. Based Syst.*, vol. 133, pp. 197–207, Oct. 2017.

[16] H. Wang and S. Li, ''Differential private multiple classification algorithm for SVM,'' in *Proc. 5th IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Nov. 2019, pp. 604–609.

[17] M. Senekane, ''Differentially private image classification using support vector machine and differential privacy,'' *Mach. Learn. Knowl. Extraction*, vol. 1, pp. 483–491, Mar. 2019.

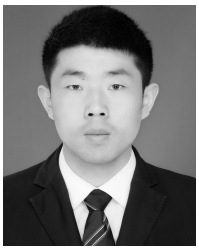
[18] J. Platt, ''Sequential minimal optimization: A fast algorithm for training support vector machines,'' Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-98-14, 1998.

[19] C. Dwork, ''A firm foundation for private data analysis,'' *Commun. ACM*, vol. 54, no. 1, pp. 86–95, Jan. 2011.

[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith, ''Calibrating noise to sensitivity in private data analysis,'' in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Germany: Springer, 2006, pp. 265–284.

[21] N. Phan, X. Wu, H. Hu, and D. Dou, "Adaptive Laplace mechanism: Differential privacy preservation in deep learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Los Alamitos, CA, USA, Nov. 2017, pp. 385–394. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICDM.2017.48
[22] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2007, pp. 94–103.
[23] F. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," *Commun. ACM*, vol. 53, no. 9, pp. 89–97, Sep. 2010.
[24] T. Kudo, "Chunking with support vector machines," in *Proc. NAACL*, 2001, pp. 1–8. [Online]. Available: https://ci.nii.ac.jp/naid/10014960494/en/
[25] J.-X. Dong, A. Krzyzak, and C. Y. Suen, "Fast SVM training algorithm with decomposition on very large data sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 603–618, Apr. 2005.
[26] T. Ramakrishnan and B. Sankaragomathi, "A professional estimate on the computed tomography brain tumor images using SVM-SMO for classification and MRG-GWO for segmentation," *Pattern Recognit. Lett.*, vol. 94, pp. 163–171, Jul. 2017.
[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27-1–27-27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

YALING ZHANG received the B.S. degree in computer science from Northwest University, Xi'an, China, in 1988, and the M.S. degree in computer science, in 2001, and the Ph.D. degree in mechanism electron engineering, in 2008, both from the Xi'an University of Technology, Xi'an, China, respectively, where she is currently a Professor. Her current research interests include data mining and privacy protection.



ZHIFENG HAO received the B.S. degree from the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China, in 2016. He is currently pursuing the M.S. degree from the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His research interests include data mining and privacy protection.

SHANGPING WANG received the B.S. degree in mathematics from the Xi'an University of Technology, Xi'an, China, in 1982, and the M.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 1989, and the Ph.D. degree in cryptology from Xidian University, Xi'an, China, in 2003. He is currently a Professor with the Xi'an University of Technology. His current research interests include cryptography and information security.