Privacy-Preserving Outsourcing Support Vector Machines with Random Transformation

Keng-Pei Lin† Ming-Syan Chen†‡
†Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
‡Research Center of Information Technology Innovation, Academia Sinica, Taipei, Taiwan
kplin@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw

ABSTRACT
Outsourcing the training of support vector machines (SVM) to external service providers benefits the data owner who is not familiar with the techniques of the SVM or has limited computing resources. In outsourcing, the data privacy is a critical issue for some legal or commercial reasons since there may be sensitive information contained in the data. Existing privacy-preserving SVM works are either not applicable to outsourcing or weak in security. In this paper, we propose a scheme for privacy-preserving outsourcing the training of the SVM without disclosing the actual content of the data to the service provider. In the proposed scheme, the data sent to the service provider is perturbed by a random transformation, and the service provider trains the SVM for the data owner from the perturbed data. The proposed scheme is stronger in security than existing techniques, and incurs very little redundant communication and computation cost.

Categories and Subject Descriptors
H.2.7 [Database Management]: Database Administration—Security, integrity, and protection; H.2.8 [Database Management]: Database applications—Data mining

General Terms
Algorithms, Experimentation, Security

Keywords
Privacy-preserving data mining, classification, outsourcing, support vector machines

1. INTRODUCTION
Data mining is an important and widely used information technology. The organization which collects the data may not be expert in data mining techniques, or may not possess sufficient computing resources to perform data mining since the data mining algorithms are usually computationally intensive. With only limited computing resources, performing data mining tasks for large-scale data may consume a lot of time. Outsourcing the data mining tasks to third-party service providers who are professional in this area is a reasonable choice for the data owner. Major IT companies like Google and Microsoft are constructing infrastructures to run Software as a Service (SaaS). This benefits small companies to run larger applications in the cloud-computing environment. Compared with performing data mining tasks in-house, outsourcing will save much hardware, software and personnel investments of the data owner, and then the data owner can focus more on its core business.

Data privacy is a concern in outsourcing the data mining tasks since the external service providers may be malicious or compromised. The data is a valuable asset of the data owner. The interest of the data owner will be hurt if the data are leaked to its commercial competitors. Leaking the data may even violate the laws if the personal information is contained. Therefore, for legal or commercial reasons, the content of the data needs to be protected when outsourcing the data mining tasks.

The support vector machine (SVM) [28] is a classification algorithm which yields state-of-the-art performance. There have been studies for outsourcing the SVMs by geometrically transforming the data with rotation and/or translation [6,7], which transforms data to another vector space to perturb the content of instances but preserves the dot product or Euclidean distance relationships among all instances. Since the SVMs with common kernel functions depend only on the dot products or Euclidean distance among all instances, the SVMs can be derived from the rotated or translated data without hurting the performance. However, the preservation of the dot product or Euclidean distance relationships is a weakness in security. For $n$-dimensional data, if the attacker knows $n$ or more linearly independent instances from some external information sources, it can rebuild the mappings of the leaked instances and the transformed ones by distance inference, and then all transformed instances can be recovered by setting $n$ equations. The work of [7] defended this weakness by adding Gaussian noise to degrade the preserved distance, but this contradicts the objective of the rotational/translational transformation which aims to preserve the necessary utility of data for training the SVM.

Most existing privacy-preserving SVM works do not address the privacy issues for outsourcing the SVM training. The works of [13, 20, 21, 27, 31, 32] focused on how to train
The proposed scheme not only allows to solve an SVM problem from the perturbed training data, but the owner. The data owner can send the perturbed testing instances to the service provider for privately outsourcing the testing as shown in Figure 1(b).

In privacy-preserving outsourcing the data mining, the additional computational cost imposed on the data owner should be minimized and the redundant communication cost should also not be too much, or the data owner rather performs data mining by itself than outsourcing. In the proposed scheme for privacy-preserving outsourcing the SVM training, the redundant communication cost is about 10% of the original data size. The data sent to the service provider is perturbed by a random linear transformation. Since the linear transformation can be executed efficiently, it incurs very little computational load to the data owner.

The rest of this paper is organized as follows: In Section 2, we survey related works and review the SVM for preliminaries. Section 3 describes our scheme of training the SVM from randomly transformed data for privacy-preserving outsourcing the SVM training. Then in Section 4, we consider the outsourcing of the testing. Section 5 analyzes the complexity and security of the proposed scheme. Section 6 shows the experimental results. Finally, we conclude the paper in Section 7.

2. PRELIMINARY

In this section, we survey the related works of privacy-preserving data mining, and review the SVM to give the preliminaries of this work.

2.1 Related Work

Privacy-preserving data mining techniques can be roughly classified into two major categories: releasing modified data for data mining [1, 3, 8], and cooperative data mining on distributed private data [12,26].

The purpose of the techniques of data mining on distributed private data is for exchanging necessary information among parties to build data mining models on the whole data without revealing the actual content of each party’s own data to others. For example, the works of [12, 26] designed protocols for privacy-preserving association rule mining on the data partitioned among different parties, and the work of [19] considered for building decision trees. Most existing privacy-preserving SVM works [13,20,21,27,31,32] also belong to this category.

The techniques of releasing modified data for data mining is to let others build data mining models without revealing actual data. A popular approach for releasing modified data for data mining is perturbing the data by adding random noise. The data are individually perturbed by adding noise randomly drawn from a known distribution, and data mining algorithms are performed from the reconstructed aggregate distributions of the noisy data [3, 8]. The work of [3] addressed learning decision trees, and the work of [8] addressed mining association rules. A condensation-based approach is proposed in [1], where data are first clustered into groups, and then pseudo-data are generated from those clustered groups. Data mining tasks are then done on the generated synthetic data instead of the original data. The anonymous data publishing technique like k-anonymity [22, 25], which makes the quasi-identifier values be able to indistinguishably map into at least k-records by generalizing or suppressing the values in quasi-identifier attributes, has also been successfully used in data mining. For example, the work
of [11] studied the performance of the SVM built upon the anonymized data and the anonymized data with additional statistics of the generalized attributes.

The techniques of releasing modified data for data mining can partly fulfill the objective of privacy-preserving outsourcing data mining tasks, which aims to let external service providers build data mining models for the data owner without revealing actual data. An issue of using such techniques for outsourcing is the data privacy since the modified data still disclose the content in degraded precision or anonymized form. The other issue is that the built data mining models are also possessed by the service provider if such techniques are adopted. The main difference is that the privacy-preserving outsourcing data mining also requires protecting the access of the built data mining models. These issues of outsourcing the data mining are seldom addressed in the literature of privacy-preserving data mining.

The work of [29] designed a scheme for privacy-preserving outsourcing the association rule mining, in which the items in the database are substituted by ciphers. The cipher-substituted items protect the actual content of data but preserve the counts of itemsets for performing association rule mining. The association rules obtained by the service provider are also in ciphered form, which can only be utilized by the data owner itself.

The works of [6, 7] considered the privacy-preserving outsourcing of the SVM by utilizing geometric transformations to hide the actual content but preserve the dot product or Euclidean distance relationships of data for outsourcing the SVM. The built SVM classifiers from the transformed data are also in transformed form.

Except the works of geometric perturbation [6, 7], existing privacy-preserving SVM works do not address the outsourcing of the SVM. The work of [18] considered the problem of releasing a built SVM classifier without revealing the support vectors, which is a subset of the training data. The methods proposed in [13, 20, 21, 27, 31, 32] designed protocols to train the SVM on the training data separately held by different parties without revealing data to each other. In the works of [27, 31, 32], training data owners cooperatively compute the Gram matrix to build the SVM problem without revealing their own data to others by utilizing the secure multi-party integer sum. The work of [13] utilizes cryptographic techniques to design a private two-party kernel adatron algorithm for training the SVM. The works of [20, 21] adopt the reduced SVM [15] with random reduced set for cooperatively building the kernel matrix among parties. The above-mentioned privacy-preserving SVM works are not applicable for a data owner to send the whole dataset to a single service provider to train the SVM with privacy-preservation.

To the best of our knowledge, currently only the geometric transformation-based schemes of [6, 7] are applicable to privacy-preserving outsourcing the SVM. However, transforming the data by geometric transformation is weak in security as we have mentioned in Section 1.

Our scheme also utilizes the RSVM with random vectors as the reduced set, which has been used in [20, 21] for privately computing the kernel matrix on the data partitioned among different parties. However, the way the RSVM is employed with random vectors in our work is different from [20, 21]. In our privacy-preserving SVM outsourcing scheme, we capitalize on random vectors to prevent the inherent weakness in the full kernel matrix and enable the computation of the kernel matrix from randomly transformed training data.

There have been many works considering for outsourcing the database queries with privacy-preservation. For example, the works of [2, 9] proposed schemes for performing comparison operations and SQL queries on encrypted databases, and the work of [30] proposed a scheme for performing \( k \)-nearest neighbor queries on randomly transformed database. Outsourcing data mining with privacy-preservation is less discussed since there are usually complex operations involved in data mining algorithms.

### 2.2 Review of the SVM

The SVM is a statistically robust learning method based on the structural risk minimization of the statistical learning theory [28]. The objective of the SVM is to find an optimal separating hyperplane which maximizes the margin between two classes of data in the kernel induced feature space.

Without loss of generality, suppose there are \( m \) instances of training data. Each instance consists of a \((x_i, y_i)\) pair where \( x_i \in \mathbb{R}^n \) contains \( n \) attributes of the \( i\)-th instance and \( y_i \in \{+1, -1\} \) is its class label. The SVM finds the optimal separating hyperplane \( w \cdot x + b = 0 \) to obtain the decision function \( f(x) = w \cdot x + b \) by solving the following quadratic programming optimization problem:

\[
\begin{align*}
\min_{w, b} & \quad \frac{1}{2} ||w||^2 + C \sum_{i=1}^{m} \xi_i \\
\text{subject to} & \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, m
\end{align*}
\]

Minimizing \( \frac{1}{2} ||w||^2 \) in the objective function means maximizing the margin between the two classes of data. The slack variable \( \xi_i \) denotes the extent of \( x_i \) falling in the erroneous region, for \( i = 1, \ldots, m \), and \( C > 0 \) is the parameter which denotes the cost of slacks. The objective function is a trade-off between maximizing the margin and minimizing the slacks. The resulted classifier is \( \text{sgn}(f(x)) \) for determining which side of the optimal separating hyperplane the testing instance \( x \) falls into.

The SVM’s optimization problem is usually solved in its dual form, which makes available the application of the kernel trick:

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} Q\alpha - \sum_{i=1}^{m} \alpha_i \\
\text{subject to} & \quad \sum_{i=1}^{m} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m
\end{align*}
\]

where \( Q \) is called kernel matrix with \( Q_{i,j} = y_i y_j k(x_i, x_j) \), \( i = 1, \ldots, m \), \( j = 1, \ldots, m \). The decision function in the dual form is \( f(x) = \sum_{i=1}^{m} \alpha_i y_i k(x_i, x) + b \). The function \( k(x, y) \) is called kernel function, which implicitly maps \( x \) and \( y \) into a high-dimensional space and computes their dot product there. By applying the kernel trick, the SVM maps data into the kernel induced high-dimensional space to find an optimal separating hyperplane. The original dot product is called linear kernel \( k(x, y) = x \cdot y \). Commonly used kernel functions include Gaussian kernel \( k(x, y) = \exp(-g||x - y||^2) \) with \( g > 0 \), polynomial kernel \( k(x, y) = (g x \cdot y + r)^d \) with \( g > 0 \), and the neural network kernel \( k(x, y) = \tanh(g x \cdot y + r) \), where \( g \), \( r \), and \( d \) are kernel parameters.
Without appropriate choices on the cost and kernel parameters, the SVM will not achieve good classification performance. A process of parameter search (also called model selection) is required to determine a suitable parameter combination of the cost and kernel parameters for training the SVM. Practitioners usually evaluate the performance of a parameter combination by measuring its average accuracy of cross-validation [10, 23], and the search of parameter combinations is often done in a brute-force way. For example, with the Gaussian kernel, the guide of LIBSVM [5, 10] suggests performing a grid-search with exponential growth on the combinations of \((C, g)\). The parameter combination which results in the highest average cross-validation accuracy will be selected to train an SVM classifier on the full training data.

The parameter search process can be very time-consuming since there are usually hundreds of parameter combinations to try, and for each parameter combination, with \(k\)-fold cross-validation, there are \(k\) SVMs to be trained on \(\frac{n-1}{k}\) of the training data. For example, the default search range of LIBSVM’s parameter search tool for Gaussian kernel is \(C = 2^{-5}\) to \(2^{15}\) and \(g = 2^{-15}\) to \(2^3\), both stepped in \(2^2\), where 5-fold cross-validation is used. There are \(11 \times 10^2 = 110\) parameter combinations to be tested in its default setting, and hence there are totally \(5 \times 110 = 550\) SVMs to be trained in the parameter search process.

If the training dataset is large, training an SVM is already costly, and the parameter search process even involves training hundreds of SVMs along with hundreds of testings. Due to the heavy computational load, for a data owner who has only limited computation resource, it is reasonable to outsource the SVM training to an external service provider. It is noted that the SVM problems in the parameter search process are independent, and hence the parameter search process can be easily parallelized in cloud computing environment. Since the service provider may be untrusted or malicious, the actual content of the data needs appropriate protections to preserve the data privacy.

3. TRAINING SVM ON RANDOMLY TRANSFORMED DATA

In this section, we introduce the scheme of training SVMs from random transformation-perturbed data for outsourcing the SVM training to a potentially malicious external service provider. Our work overcomes the security weakness of the geometric transformation by perturbing data with random transformation, which does not preserve the Euclidean distance and dot product relationships among the training instances, and hence provides stronger security in data privacy.

3.1 The Reduced SVM with Secure Kernel Matrix

Note that the full kernel matrix \(K\) of the conventional SVM problem (2) contains dot products or Euclidean distance among all training instances for common kernel functions. If the service provider needs to solve SVM problems from the full kernel matrix, there will be similar security weakness like the rotationally/translationally transformed training data.

We tackle this problem by adopting the reduced SVM (RSVM) with random reduced set [14, 15, 20, 21] for training SVMs from a secure kernel matrix. The RSVM is a SVM scaling up method, which utilizes a reduced kernel matrix. Each element of the reduced kernel matrix is computed from an instance in the training dataset and an instance in the reduced set. The number of instances in the reduced set is typically less than 10% of the training dataset [14, 15]. Hence the reduced kernel matrix is much smaller than a full kernel matrix and can easily fit into the main memory. Let \(x_i \in \mathbb{R}^n, i = 1, \ldots, m\) stand for the instances of the training dataset, and \(y_i \in \{1, -1\}, i = 1, \ldots, m\) are their corresponding labels. Let \(\mathcal{R} = \{r_j | r_j \in \mathbb{R}^n, j = 1, \ldots, \tilde{m}\} \) denotes the reduced set, where \(\tilde{m} < m\). The original RSVM paper adopted a subset of the training dataset as the reduced set [15]. The reduced kernel matrix \(K\) is an \(m \times \tilde{m}\) matrix where

\[
K_{ij} = k(x_i, r_j), \quad i = 1, \ldots, m, \quad j = 1, \ldots, \tilde{m}
\]

An interesting property of the RSVM is that the reduced set \(\mathcal{R}\) is not necessary to be a subset of the training dataset [14]; completely random vectors can act as the instances of the reduced set [20, 21].

If random vectors are adopted, then an element in the reduced kernel matrix is the kernel evaluation between an instance in the training dataset and a random vector in the reduced set but not the kernel evaluation between the training instances like the kernel matrix of a conventional SVM. Hence the reduced kernel matrix does not contain the dot product or distance relationships of the training instances. This forms the secure kernel matrix, which avoids the security weakness from disclosing the kernel matrix of a conventional SVM. As long as the random vectors are kept secret, even if part of training instances are leaked, other instances still cannot be derived from the secure kernel matrix.

If the service provider has the secure kernel matrix, it can solve the following RSVM problem [14, 15]:

\[
\arg \min_{\mathbf{v}, b, \xi} \frac{1}{2} \left( \|\mathbf{v}\|^2 + b^2 \right) + C \sum_{i=1}^{m} \xi_i^2
\]

subject to \(y_i (\mathbf{v}^T x_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m\)

where \(\xi_i, i = 1, \ldots, m\) are slack variables, and \(C\) is the cost parameter. The solutions \(\mathbf{v} = (v_1, \ldots, v_{\tilde{m}})^T\), \(b\), and the vectors of the reduced set constitute the decision function

\[
f(x) = \sum_{j=1}^{\tilde{m}} v_j k(x, r_j) + b
\]

The optimization problem of the RSVM can be solved by a normal linear SVM solver [17] or the smooth SVM [16] used in the original RSVM paper [15]. Empirical studies of [14, 15, 17] showed that the RSVM can achieve similar classification performance to a conventional SVM.

Note that simply sharing the secure kernel matrix is not appropriate for outsourcing the SVM training since a reduced kernel matrix is computed from a fixed kernel parameter, while there are various SVMs with different kernel parameters to be trained in the parameter search process. The data owner will be imposed much computation load as well as much communication cost to build and send many secure kernel matrices with different kernel parameters to the service provider. Our goal is to outsource the SVM training which must minimize as much load of the data owner as possible.
In the following, we show a random transformation-based data perturbation scheme which makes available the computation of the secure kernel matrix from perturbed training data. The computational cost of the random transformation is similar to geometric perturbation methods [6, 7], but the random transformation does not preserve the dot product or distance relationships between training instances and hence is stronger in security.

### 3.2 Building the Secure Kernel Matrix from Randomly Transformed Data

The data owner will send the perturbed training instances as well as the perturbed random vectors of the reduced set to the service provider for computing secure kernel matrices. A secure kernel matrix $K$ is computed from training instances and random vectors of the reduced set by $K_{i,j} = k(x_i, r_j)$, $i = 1, \ldots, m$, $j = 1, \ldots, \tilde{m}$. Our objective is to let the service provider compute the same $K$ from perturbed training instances and random vectors, while the perturbation scheme should not allow the security weakness of the geometric perturbation schemes, i.e., the dot product and Euclidean distance among training instances should not be preserved. The perturbation scheme needs to preserve the kernel evaluations between a training instance and a random vector for computing secure kernel matrices. We utilize the random transformation perturbation for computing the dot product of two differently transformed instances [30]. Note that the attribute vectors $x_i$'s of training instances are usually considered sensitive, but the class labels $y_i$'s are usually not.

Let $M$ be a nonsingular $n \times n$ matrix composed of random values. We perturb the instances of the training dataset by a random linear transformation $L : \mathbb{R}^n \rightarrow \mathbb{R}^n$, where the matrix $M$ acts as the random linear operator. All training instances are perturbed by the random transformation $^1$

$$c_i = L(x_i) = Mx_i \quad \text{for } i = 1, \ldots, m \quad (6)$$

Unlike the geometric perturbation, the random transformation does not preserve the Euclidean distance and dot products between training instances since the vector space is randomly transformed. Hence the security weakness of the rotational or translational transformation does not exist in the data perturbed by random transformation. The random vectors $r_j$, $j = 1, \ldots, \tilde{m}$ of the reduced set are also perturbed by another random linear transformation $L' : \mathbb{R}^n \rightarrow \mathbb{R}^n$ with $(M^T)^{-1}$ as the random linear operator:

$$s_j = L'(r_j) = (M^T)^{-1}r_j \quad \text{for } j = 1, \ldots, \tilde{m} \quad (7)$$

The perturbed training instances $c_i$, $i = 1, \ldots, m$ and perturbed random vectors of the reduced set $s_j$, $j = 1, \ldots, \tilde{m}$ are then sent to the service provider for building secure kernel matrices.

The dot product between an instance $x_i$ and a random vector $r_j$ can be equivalently computed from the dot product of $c_i$ and $s_j$ by $c_i^T s_j = (Mx_i)^T (M^T)^{-1}r_j = x_i^T M^T (M^T)^{-1}r_j = x_i^T r_j$. Therefore, for the dot product-based kernel functions including the linear kernel $k(x_i, r_j) = x_i \cdot r_j$, polynomial kernel $k(x_i, r_j) = (g(x_i, r_j))^d$, and neural network kernel $k(x_i, r_j) = \tanh(g(x_i, r_j))$, the kernel evaluations between an instance and a random vector can be equivalently derived from the perturbed training instances and random vectors.

For Gaussian kernel $k(x_i, r_j) = \exp(-\gamma ||x_i - r_j||^2)$ which is based on the Euclidean distance, a slight modification is needed to add another two dimensions to the original instances $x_i \in \mathbb{R}^n$ as $x_i' = (x_i, 1, \ldots, 1, -\frac{1}{2}||x_i||^2)^T$ before applying the transformation. The random vectors $r_j$'s of the reduced set are also added by another two dimensions as $r_j' = (r_j, 1, \ldots, 1, -\frac{1}{2}||r_j||^2, 1)^T$. Then the corresponding random matrix for random transformation is a nonsingular $(n+2) \times (n+2)$ matrix $M$. Similarly, the data are perturbed by $c_i = Mx_i$, and the random vectors are perturbed by $s_j = (M^T)^{-1}r_j$. The Euclidean distance between $x_i$ and $r_j$ in the Gaussian kernel can be equivalently computed from $c_i$ and $s_j$ by $-2c_i^T s_j = -2x_i^T M^T (M^T)^{-1}r_j = -2x_i^T r_j' = -2x_i^T (x_i - r_j)^T = ||x_i||^2 - 2x_i^T r_j + ||r_j||^2 = ||x_i - r_j||^2$.

For common kernel functions, the kernel evaluations between an instance in the training set and an instance in the reduced set can be equivalently computed from their perturbed versions. Hence the service provider can derive the same secure kernel matrix $K$ of (4) from the perturbed data $c_i$'s and $s_j$'s, i.e., the RSVM optimization problem derived from the perturbed data is the same to the one derived from original data. Therefore, the service provider can obtain the same solutions $v_j$, $j = 1, \ldots, \tilde{m}$ and $b$ of the decision function (5).

### 3.3 Incorporating the Parameter Search Process

The parameter search by cross-validation involves training many SVMs on subsets of training instances with different parameter combinations. The service provider can perform this by building secure kernel matrices on subsets of the perturbed training instances with different kernel parameters, and then forming the RSVM problems with different parameter combinations to train intermediate classifiers for evaluating the performance by cross-validation.

An essential part in the cross-validation is testing for measuring the accuracy of classifying remaining part of the perturbed training instances by the intermediate classifiers. Note that the elements of the reduced set $r_j$, $j = 1, \ldots, \tilde{m}$ work as support vectors in the decision function (5). Since the service provider only has perturbed reduced set $s_j = (M^T)^{-1}r_j$, $j = 1, \ldots, \tilde{m}$, the classifier obtained by the service provider is also in a perturbed form:

$$f'(x) = \sum_{j=1}^{\tilde{m}} v_j k(x, s_j) + b \quad (8)$$

To use this classifier for testing, the testing instances also need to be perturbed by the same random transformation of the training instances, which implies that the intermediate classifiers can classify the perturbed training instances possessed by the service provider. For example, with the linear kernel $k(x_i, r_j) = x_i^T r_j$, the perturbed decision function $f'(x)$ classifies a perturbed training instance $c_i = Mx_i$ as

$$f'(c_i) = \sum_{j=1}^{\tilde{m}} v_j k(c_i, s_j) + b = \sum_{j=1}^{\tilde{m}} v_j ((Mx_i)^T s_j) + b$$

$$= \sum_{j=1}^{\tilde{m}} v_j (x_i^T M^T (M^T)^{-1}r_j) + b = \sum_{j=1}^{\tilde{m}} v_j (x_i^T r_j) + b$$

$^1$It is not necessary to put the whole matrix $M$ in the main memory. The computation can be decomposed to $Mx = x_1M_{1,i} + \cdots + x_nM_{n,i}$, where $M_{i,j}$ is the $i$-th column of $M$. 

For common kernel functions, the kernel evaluations between an instance and a random vector can be equivalently computed from the perturbed training instances and random vectors.

For Gaussian kernel $k(x_i, r_j) = \exp(-\gamma ||x_i - r_j||^2)$ which is based on the Euclidean distance, a slight modification is needed to add another two dimensions to the original instances $x_i \in \mathbb{R}^n$ as $x_i' = (x_i, 1, \ldots, 1, -\frac{1}{2}||x_i||^2)^T$ before applying the transformation. The random vectors $r_j$'s of the reduced set are also added by another two dimensions as $r_j' = (r_j, 1, \ldots, 1, -\frac{1}{2}||r_j||^2, 1)^T$. Then the corresponding random matrix for random transformation is a nonsingular $(n+2) \times (n+2)$ matrix $M$. Similarly, the data are perturbed by $c_i = Mx_i$, and the random vectors are perturbed by $s_j = (M^T)^{-1}r_j$. The Euclidean distance between $x_i$ and $r_j$ in the Gaussian kernel can be equivalently computed from $c_i$ and $s_j$ by $-2c_i^T s_j = -2x_i^T M^T (M^T)^{-1}r_j = -2x_i^T r_j' = -2x_i^T (x_i - r_j)^T = ||x_i||^2 - 2x_i^T r_j + ||r_j||^2 = ||x_i - r_j||^2$.

For common kernel functions, the kernel evaluations between an instance in the training set and an instance in the reduced set can be equivalently computed from their perturbed versions. Hence the service provider can derive the same secure kernel matrix $K$ of (4) from the perturbed data $c_i$'s and $s_j$'s, i.e., the RSVM optimization problem derived from the perturbed data is the same to the one derived from original data. Therefore, the service provider can obtain the same solutions $v_j$, $j = 1, \ldots, \tilde{m}$ and $b$ of the decision function (5).
The Gaussian kernel, polynomial kernel, and neural network kernel can be tackled in similar ways like the step of deriving secure kernel matrices from perturbed data. Classifying the perturbed training instance \( c_i = Mx_i \), with the perturbed decision function \( f'(x) \) is equivalent to classifying the original training instance \( x_i \) by the unperturbed decision function \( f(x) \). Hence the service provider can perform cross-validations to evaluate the performance of intermediate classifiers by training and testing on perturbed training instances.

It is noted that the service provider cannot use the perturbed decision function to classify the data other than the ones sent from the data owner. Since the perturbed decision function \( f'(x) \) requires the testing instance being transformed to \( Mx \) to annihilate the effect of \((M^T)^{-1}\) multiplying to \( r_j \)'s, if the testing instance is not perturbed by \( M \), the perturbed decision function \( f'(x) \) will result in an unmeaningful value. Hence for the service provider, the applicability of the perturbed decision function is restricted to classify only the data sent from the data owner for performing cross-validation. This protects the classifier built from the data of the data owner.

The following summarizes the whole scheme of privacy-preserving outsourcing the SVM training based on random transformations. The data owner generates random vectors as the reduced set for the RSVM, perturbs the training instances by a random matrix \( M \), perturbs the random vectors by \((M^T)^{-1}\), and then sends the perturbed training instances and perturbed random vectors to the service provider. Upon receiving the perturbed data, the service provider performs the cross-validation-based parameter search process by computing secure kernel matrices from part of perturbed training instances to train intermediate classifiers and using the intermediate classifiers to test on remaining perturbed training instances for evaluating the performance of various parameter combinations. The parameter combination which achieves the best average cross-validation accuracy will be adopted to form a final RSVM problem on complete perturbed training instances, and then the solutions \( v_j, j = 1, \ldots, m \) and \( b \) of the final problem along with the adopted kernel parameter(s) will be returned to the data owner for composing a decision function with the original random vectors.

The random transformations are utilized to protect the data privacy in the initial stage before forming the SVM problem, and the RSVM with random vectors as the reduced set is utilized to prevent the security weakness in the intermediate computations. This two-layer protection overcomes the security weakness existing in the conventional SVM formulations and the geometric perturbation-based schemes [6, 7] for outsourcing the SVM training with privacy-preservation.

4. PRIVACY-PRESERVING OUTSOURCING THE TESTING

The perturbed decision function \( f'(x) \) in (8) also enables privacy-preserving outsourcing of the testing tasks. Since classifying the perturbed instance \( Mx \) by the perturbed decision function \( f'(x) \) is equivalent to classifying the original instance \( x \) by the original decision function \( f(x) \), the data owner can perturb the testing instances by the random matrix \( M \) the same with the one to perturb training instances, and then send the perturbed testing instances to the service provider for outsourcing the testing tasks.

Outsourcing the testing tasks is not limited to outsourcing to the service provider who generates the SVM classifier from perturbed training data. The testing task can be outsourced to another party by sending to whom the perturbed decision function, including the perturbed reduced set, and the perturbed testing instances for performing testings. When outsourcing the testing task, the random matrix for perturbation can be changed for higher security concerns. It is done by perturbing the testing instances by another random matrix \( M' \) and perturbing the random vectors of the reduced set by corresponded \((M'^T)^{-1}\), i.e., the random matrix \( M \) for perturbation in outsourcing the testing can be arbitrarily changed by providing that the testing instances and random vectors are perturbed by the pair \( M \) and \((M^T)^{-1}\) respectively.

Although the decision function of the SVM is very simple and can be computed fast, since there are slower exponent computations involved in common kernel functions, randomly transforming the testing data for outsourcing the testing is still able to save some computational load of the data owner. We will validate this claim in the experiment.

5. COMPLEXITY AND SECURITY ANALYSIS

Compared to simply outsourcing the SVM training without protecting the data, the additional computational cost imposed on the data owner for protecting data privacy includes: (1) Generating an \( n \times n \) random matrix for perturbation, which costs \( O(n^2) \). (2) Perturbing \( m\) \( n\)-dimensional training instances by matrix multiplications, which costs \( O(m\cdot n^2) \). (3) Generating and perturbing \( m \) random vectors, which costs \( O(m\cdot n^2) \). Hence the totally additional computational cost imposed on the data owner is \( O((m+m+1)n^2) \approx O(mn^2) \). For outsourcing the testing, the additional computational cost is to perturb the testing instances, which costs \( O(n^2) \) for each instance.

Compared to the geometric transformation-based scheme, applying the rotational transformation costs \( O(mn^2) \), and applying the translational transformation vector costs \( O(mn) \). Hence the additional computational cost of using the geometric transformation is about \( O(mn^2) \), which is similar to using our random transformation-based scheme. The communication cost of sending the transformation-perturbed instances is same to sending the original instances, both are \( O(mn) \). The additional communication cost of using our random transformation-based scheme is to send the perturbed random vectors, which incurs \( O(mn) \) redundant communication cost.

In the following, we consider the security issues. The discussions consider the cases of dot product-based kernels. The Euclidean distance-based kernels can be applied in a similar way. Without loss of generality, consider the situation that the service provider has known the content of \( m+1 \) training instances from external information sources.

(i) The service provider cannot recover the random matrix \( M \) from the leaked \( m+1 \) training instances. If the service provider can successfully determine the correspondence between original...
instances \( x_i \)'s and perturbed ones \( c_i \)'s of \( n \) or more linearly independent \( x_i \)'s, the \( n \times n \) random matrix \( M \) for perturbation can be derived by solving \( n \) equations. However, since the distance and dot product between the training instances are not preserved in the random transformation-based perturbation, the service provider is not able to determine the correspondence between the original instances and the perturbed ones. Hence the service provider cannot setup equations to derive the random matrix \( M \).

(ii) The service provider cannot recover the content of the perturbed reduced set. The service provider knows the dot product between any pair of \( c_i \) and \( s_j \), \( i = 1, \ldots, m \), \( j = 1, \ldots, m \). If the service provider knows \( m \) linearly independent \( x_i \)'s and their corresponding \( c_i \)'s, all the \( r_j \)'s can be recovered from \( s_j \)'s by solving \( m \) equations. But the service provider cannot not know the correspondence between any \( c_i \) and \( x_i \), as we have discussed in (i). Hence the service provider cannot setup equations to recover the original content of the reduced set \( r_j \), \( j = 1, \ldots, m \). Also, since the service provider cannot recover the reduced set, it cannot recover the random matrix \( M \) from the perturbed reduced set.

(iii) The service provider cannot recover the remaining one training instance. Since the correspondence between the original training instances and the perturbed ones cannot be determined, as the reasons given in (i), and the dot product relationships are not preserved, the remaining one training instance cannot be recovered by setting up equations. Although the service provider knows the dot products between the remaining one training instance and the random vectors of the reduced set, it does not know the original content of the random vectors. Hence the remaining one training instance also cannot be recovered from the perturbed random vectors.

The security of our scheme depends on the property of the random transformation-based perturbation, which does not preserve the dot product and Euclidean distance relationships among the training instances and hence is resistant to the distance inference attack. For the geometric perturbation-based schemes [6, 7], if \( n \) or more linearly independent instances are leaked, the correspondence between the original instances and the geometric-perturbed ones can be determined from the distance and dot product relationships, and then all other instances can be recovered by setting up equations.

The security of outsourcing the testing is similar. Since the service provider does not know the original content of the perturbed reduced set, it cannot recover the content of the perturbed testing instances from the dot product or Euclidean distance between the perturbed testing instances and the perturbed reduced set.

6. EXPERIMENTAL ANALYSIS

In the experiments, we first compare the the classification performance between a conventional SVM and the RSVM with random vectors as the reduced set to evaluate the effectiveness of the proposed scheme on classification. Then we measure the computational time imposed on the data owner of using our privacy-preserving outsourcing SVM scheme, and compare with the computational time of training the SVM locally to demonstrate the computational load saved from using the outsourcing scheme. Finally, we compare the classification performance with the SVM trained from the anonymized data since the anonymous data publishing technique [25] is suitable for revealing the datasets where only the identities of instances are concerned.

Since training SVMs on large datasets is very time consuming, for the ease of experiments, we choose the datasets with moderate size for performing experiments. The difference in the scale of consuming time between outsourcing and local training is clear to demonstrate the efficacy of our scheme. The datasets used in the experiments are available at the UCI machine learning repository [4]. We select some medical datasets and bank credit datasets, which have stronger privacy concerns, to evaluate the effectiveness of the scheme. The medical datasets include Wisconsin breast cancer, Pima Indian diabetes, Liver disorder, Statlog heart disease, which contain medical records of patients. The bank credit datasets are Australian credit and German credit numeric version, in which personal information of bank customers is contained. The statistics of the datasets are shown in Table 1. The programs of random transformation and the RSVM are written in Matlab. The experimental platform is a PC featured with an Intel Core 2 Q6600 CPU and 4GB RAM, running Windows XP.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Heart</th>
<th>Breast</th>
<th>Aus. credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>instances</td>
<td>270</td>
<td>683</td>
<td>690</td>
</tr>
<tr>
<td>attributes</td>
<td>13</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Liver</th>
<th>Ger. credit</th>
<th>Diabetes</th>
</tr>
</thead>
<tbody>
<tr>
<td>instances</td>
<td>345</td>
<td>1000</td>
<td>768</td>
</tr>
<tr>
<td>attributes</td>
<td>6</td>
<td>24</td>
<td>8</td>
</tr>
</tbody>
</table>

6.1 Utility of Classification

In this section, we compare the classification performance between a conventional SVM implementation, the LIBSVM [5], and the RSVM with random reduced set to show the effectiveness of our scheme on classification.

We solve the L2-norm RSVM problem (the (4) of Section 3.1) by the smooth SVM method [15,16,17]. Randomly generated vectors are adopted as the reduced set for training the RSVM. The size of the reduced set is set to 10% of the size of the training dataset. The adopted kernel function in both the RSVM and the LIBSVM is the Gaussian kernel function. The cost/kernel parameters for training the RSVM and LIBSVM are respectively determined by applying the grid search using cross-validation, where the search range is the default of LIBSVM’s parameter search tool [5,10].

Figure 2 shows the experimental results of comparing the classification performance. The reported accuracy is the 5-fold cross-validation average. It is seen that the classification accuracy of the RSVM with random vectors as the reduced set is similar to a conventional SVM, which validates that our scheme is effective for classification.

6.2 Efficiency of Outsourcing

To demonstrate the benefits of outsourcing, we measure the computational overhead imposed on the data owner with
using the privacy-preserving outsourcing scheme, and compare it with the computational time of training the SVM locally by the data owner itself to show how much computational cost can be saved from utilizing the privacy-preserving outsourcing scheme.

Table 2 shows the comparison of the required computing time of the data owner with and without utilizing the outsourcing. The SVM training includes the parameter search process and training the final classifier by the selected parameter combination. The search range adopted here is also the default of the LIBSVM’s parameter search tool [5, 10]. The training time of both the RSVM and the LIBSVM is listed for reference. Note that we do not aim to compare the training time between the two training methods since they are different implementations of the SVM. When using the outsourcing scheme for the dataset with $m$ instances in $n$-attribute, to perturb the data, the data owner needs to generate an $n \times m$ random matrix for transformation, generate $\lceil m/10 \rceil$ random vectors in $n$-dimensional for the reduced set of the RSVM, and transform the $m$ training instances and $\lceil m/10 \rceil$ random vectors by matrix multiplication. It is seen that these computations can be executed very fast. On all of the datasets, they are done within 0.5 millisecond. However, locally training the SVM takes at least several seconds to complete, which costs the data owner more than 10,000 times of computing time than the outsourcing.

The difference in the scale of computing time between outsourcing and locally training is very large. This validates the claim that the proposed privacy-preserving outsourcing scheme incurs merely little computational overhead to the data owner. The computational load of the data owner can be significantly reduced, which clearly justify the efficacy of the privacy-preserving outsourcing scheme.

### Table 2: Time comparison of training SVMs with/without outsourcing

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Privacy-preserving Outsourcing</th>
<th>Locally Training (RSVM)</th>
<th>Locally Training (LIBSVM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart Disease</td>
<td>0.12 ms</td>
<td>2.9 s</td>
<td>6.5 s</td>
</tr>
<tr>
<td>Aus. credit</td>
<td>0.17 ms</td>
<td>9.7 s</td>
<td>40.4 s</td>
</tr>
<tr>
<td>Ger. credit</td>
<td>0.35 ms</td>
<td>19.7 s</td>
<td>141.9 s</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>0.14 ms</td>
<td>9.8 s</td>
<td>12.4 s</td>
</tr>
<tr>
<td>Diabetes</td>
<td>0.13 ms</td>
<td>10.8 s</td>
<td>123.4 s</td>
</tr>
<tr>
<td>Liver disorder</td>
<td>0.07 ms</td>
<td>2.6 s</td>
<td>32.3 s</td>
</tr>
</tbody>
</table>

### 6.2.1 Scalability of Random Transformation

We measure the computing time of the perturbation with random transformation on large-scale synthetic datasets to evaluate the scalability. The number of the instances of the synthetic datasets ranges from 10000 to 50000, where the dimensionality of those datasets are in 500 and 1000-dimensional, respectively. The computing time of the random transformation is shown in Figure 3. It is seen that the random transformation scales well with the increase of instances. Randomly transforming 50000 instances in 1000-dimensional takes less than 5 seconds to complete.

![Figure 3: Computing time of perturbing data by random transformation.](image)

### 6.2.2 Efficiency of Outsourcing the Testing

The overhead imposed on the data owner for outsourcing the testing is randomly transforming the testing instances. We randomly generate 10000 testing instances for each dataset to compare the time of outsourcing testing and local testing, where classifiers of each dataset are the ones trained above. The results are reported in Table 3. It is seen that the outsourcing scheme can save tens to hundreds of times of computational load for the data owner. Since the SVM testing is already efficient, the difference between outsourcing the testing and the local testing is not as significant as the cases of training.

### Table 3: Time comparison of testing 10,000 instances with/without outsourcing

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Privacy-preserving Outsourcing</th>
<th>Locally Testing (RSVM)</th>
<th>Locally Testing (LIBSVM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heart Disease</td>
<td>1.40 ms</td>
<td>43.29 ms</td>
<td>233.20 ms</td>
</tr>
<tr>
<td>Aus. credit</td>
<td>1.46 ms</td>
<td>111.35 ms</td>
<td>1380.08 ms</td>
</tr>
<tr>
<td>Ger. credit</td>
<td>3.13 ms</td>
<td>195.74 ms</td>
<td>1606.66 ms</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>1.23 ms</td>
<td>103.21 ms</td>
<td>113.17 ms</td>
</tr>
<tr>
<td>Diabetes</td>
<td>0.96 ms</td>
<td>109.69 ms</td>
<td>816.05 ms</td>
</tr>
<tr>
<td>Liver disorder</td>
<td>0.39 ms</td>
<td>46.79 ms</td>
<td>430.28 ms</td>
</tr>
</tbody>
</table>

### 6.3 Utility Comparison with $k$-Anonymity

In this section, we compare the classification performance between the RSVM with random reduced set with the SVM classifiers trained from anonymous data anonymized by the $k$-anonymity technique [11, 24]. If only the identities of data are concerned, the anonymous data publishing technique can be adopted for sending the anonymized data to service provider for outsourcing the SVM.
There are three of the above datasets containing quasi-identifier attributes: Statlog heart has \{age, sex\}, Pima Indian diabetes has \{age, number of pregnant, body mass index\}, and German credit has \{purpose, credit amount, personal status and sex, present residence since, age, job\}. Value generalization hierarchies are first built on the quasi-identifiers of each dataset, and then the Datally algorithm \cite{24} is performed to achieve $k$-anonymity. Since the SVM is a value-based algorithm, for numerical attributes, each generalized range is represented by the mean value, and for categorical data, the generalized category is represented by exhibiting all children categories \cite{11}. The cost/kernel parameters to train the SVMs from anonymized data are determined by grid-search using cross-validation.

The performance comparison between the RSVM with random reduced set and the SVMs trained from $k$-anonymized data with $k = 32$ and $k = 128$ is shown in Figure 4. The reported accuracy is 5-fold cross-validation average. On German credit dataset, the accuracy of applying the $k$-anonymity technique with $k = 32$ is similar to the RSVM with random reduced set, but it falls down when $k = 128$ due to the severer distortion of the quasi-identifier values. On the Heart and Diabetes datasets, $k = 32$ is enough to significantly distort their quasi-identifier values and thus results in lower accuracy.

![Figure 4: Classification performance comparison between the RSVM with random reduced set and the SVMs trained from $k$-anonymized data.](image)

It is seen that the distortion of quasi-identifiers to achieve $k$-anonymity will hurt the performance of the SVM, and the performance may get worse when a large $k$ is applied for better identity protection. Compared with outsourcing the SVM by $k$-anonymity, our scheme hardly hurts the performance of the SVM, and provides better protection to the data privacy since all attributes are perturbed by the random transformation.

7. CONCLUSION

We propose a scheme for training the SVM from random transformation-perturbed data, which makes available the outsourcing of the SVM with privacy-preservation. The scheme overcomes the security weakness of existing works, and incurs very little overhead on the data owner.

8. REFERENCES


