

Privacy-Preserving SVM Classification on Vertically Partitioned Data*

Hwanjo Yu¹, Jaideep Vaidya², and Xiaoqian Jiang¹

¹ University of Iowa, Iowa City IA 08544, USA
{hwanjoyu, xjia}@cs.uiowa.edu

² Rutgers University, Newark NJ 07102, USA
jsvaidya@rbs.rutgers.edu

Abstract. Classical data mining algorithms implicitly assume complete access to all data, either in centralized or federated form. However, privacy and security concerns often prevent sharing of data, thus derailing data mining projects. Recently, there has been growing focus on finding solutions to this problem. Several algorithms have been proposed that do distributed knowledge discovery, while providing guarantees on the non-disclosure of data. Classification is an important data mining problem applicable in many diverse domains. The goal of classification is to build a model which can predict an attribute (binary attribute in this work) based on the rest of attributes. We propose an efficient and secure privacy-preserving algorithm for support vector machine (SVM) classification over vertically partitioned data.

1 Introduction

The goal of data mining is to efficiently analyze large quantities of data to find interesting patterns and/or summarize the data in novel ways. Classification is one of the most common applications found in the real world. The goal of classification is to build a model which can predict the value of one variable, based on the values of the other variables. For example, based on financial, criminal and travel data, one may want to classify passengers as security risks. In the financial sector, categorizing the credit risk of customers, as well as detecting fraudulent transactions are both classification problems. Numerous such problems abound.

There is considerable research on different classification algorithms. Indeed, several different solutions are commonly used in the real world. A basic assumption is that complete access to data is available, either in centralized or federated form. However, privacy and security concerns restrict access to data. Sharing of data may not be possible due to either legal or commercial reasons. For example, due to HIPAA laws [1], medical data cannot be released for any purpose without appropriate anonymization. Similar constraints arise in many applications. European Community legal restrictions apply to disclosure of any individual data. Customer data, process data, etc., is often a valuable business asset for corporations. For example, complete manufacturing processes are

* This research was supported in part by a Faculty Research Grant from Rutgers Business School - Newark and New Brunswick.

trade secrets (although individual techniques may be commonly known). All of these cases require distributed knowledge discovery, without the disclosure of data. (Section 5 discusses related work in this area of Privacy-Preserving Data Mining.)

We assume vertically partitioned data with at least three participating parties, *i.e.*, three or more parties that collect different information about the same set of entities. For instance, a bank, health insurance company and auto insurance company collect different information about the same people. A bank has customer information like average monthly deposit, account balance. The health insurance company has access to medical information and other policy information. The car insurance company has access to information such as car type, accident claims, etc. Together, they might evaluate if the person is a credit risk for life insurance.

Support Vector Machine (SVM) classification is one of the most actively developed methodologies in data mining. SVM has proven to be effective in many real-world applications [2]. Like other classifiers, the accuracy of an SVM classifier crucially depends on having access to the correct set of data. Data collected from different sites is useful in most cases, since it provides a better estimation of the population than the data collected at a single site.

In this paper, we propose a privacy-preserving SVM (support vector machine) classification method on vertically partitioned data, PP-SVMV for short, such that each party (e.g., bank, insurance company) need not disclose its data or general information to other parties while still acquiring the same SVM classification accuracy as when the data is centralized. Our algorithm is efficient and secure. We first overview SVM (Section 2) and develop our PP-SVM technique (Section 3). We empirically show the practicality of our method in Section 4. Finally, related work is discussed in Section 5.

2 SVM Overview

We first describe the notation to overview SVM. All vectors are column vectors unless transposed to a row vector by a prime superscript $'$. The scalar (inner) product of two vectors x and y in the n -dimensional real space R^n is denoted by $x'y$ and the 2-norm of x is denoted by $\|x\|$. An $m \times n$ matrix \mathcal{A} represents m data points in a n -dimensional input space. An $m \times m$ diagonal matrix \mathcal{D} contains the corresponding labels (*i.e.*, +1 or -1) of the data points in \mathcal{A} . (A class label \mathcal{D}_{ii} , or d_i for short, corresponds to the i -th data point x_i in \mathcal{A} .) A column vector of ones of arbitrary dimension is denoted by e . The identity matrix of arbitrary dimension is denoted by \mathcal{I} .

First, consider a linear binary classification task, as depicted in Figure 1. For this problem, SVM finds the separating hyperplane ($w \cdot x = \gamma$) that maximizes the *margin*, denoting the distance between the hyperplane and closest data points (*i.e.*, support vectors). In practice, we use the “soft” margin to deal with noise, in which the distance from the boundary to each support vector could be different. The “hard” margin is formulated as $\frac{1}{\|w\|}$, as illustrated in Figure 1. To maximize the margin while minimizing the error, the standard SVM solution is formulated into the following primal program [2, 3]:

$$\min_{w,y} \quad \frac{1}{2}w'w + \nu e'y \quad (1)$$

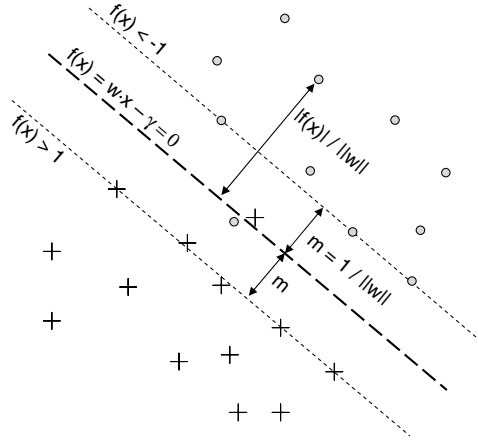


Fig. 1. The separating hyperplane that maximizes the margin. ('+' is a positive data point, *i.e.*, $f('+) > 0$, and 'o' is a negative data point, *i.e.*, $f('o') < 0$.)

$$s. t. \mathcal{D}(Aw - e\gamma) + y \geq e \text{ and } y \geq 0 \tag{2}$$

which minimizes the reciprocal of the margin (*i.e.*, $w'w$) and the error (*i.e.*, $e'y$). By having the slack variable y in the constraint (2), SVM allows error or the soft margin. The slack or error is minimized in the objective function (1) and it will be larger than zero when the point is on the wrong side or within the margin area. The soft margin parameter ν (a user parameter) is tuned to balance the margin size and the error. The weight vector w and the bias γ will be computed by this optimization problem. Once w and γ are computed, we can determine the class of a new data object x by $f(x) = w'x - \gamma$, where the class is *positive* if $f(x) > 0$, or else *negative*.

In order to reduce the number of variables in the objective function and also be able to apply the kernel trick, we transform the primal problem to the following dual problem by applying the Lagrange multipliers:

$$\min_{\alpha} \quad \frac{1}{2} \alpha' Q \alpha - e' \alpha \tag{3}$$

$$s.t. \quad 0 \leq \alpha_i \leq \nu \text{ and } \sum_i d_i \alpha_i = 0, \quad i = 0, \dots, m \tag{4}$$

where d_i (*i.e.*, \mathcal{D}_{ii}) and α_i are the class label and the coefficient respectively for a data vector x_i . The coefficients α are to be computed from this dual problem. An $m \times m$ matrix Q is computed by the scalar product of every data pair, *i.e.*, $Q_{ij} = K(x_i, x_j) d_i d_j$ where $K(x_i, x_j) = x_i \cdot x_j$ for linear SVM. The support vectors are the data vectors $\{x_i\}$ such that the corresponding coefficients $\alpha_i > 0$. The weight vector $w = \sum \alpha_i d_i x_i$ and thus the classification function $f(x) = \sum \alpha_i d_i x_i \cdot x - \gamma$ for linear SVM. For nonlinear SVMs, $f(x) = \sum \alpha_i d_i K(x_i, x) - \gamma$, where we can apply a nonlinear kernel for $K(x_i, x)$ (e.g., $K(x_i, x) = \exp(-\frac{\|x_i - x\|^2}{q})$ for RBF kernel, $K(x_i, x) = (x_i \cdot x + 1)^p$ for polynomial kernel,). [2] provides further details on SVM.

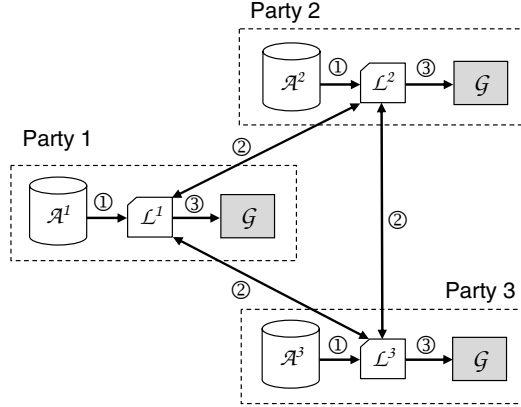


Fig. 2. PP-SVM: Framework for privacy-preserving SVM

3 Privacy-Preserving SVM

To generate the global SVM model (*i.e.*, the SVM model constructed from the data from multiple parties) without sharing any data among the parties, (1) the framework must be able to generate the global model only from models locally constructed by parties on their own data, without seeing others' data. We call this requirement *data privacy*. To prevent disclosing the general classification information on each party, (2) the local model must not be disclosed when jointly generating the global model. We call this requirement *model privacy*.

These two requirements lead to design our PP-SVM framework illustrated in Figure 2. (Figure 2 involves only three parties but can be generalized to more.) Each party builds a local model \mathcal{L} from its own data \mathcal{A} (①), and each party *securely* merges its model with others (②), in order to generate the global model \mathcal{G} (③). The global model \mathcal{G} will be the same for every party, which will be used for classifying new data objects. Assuming that the merge of the local models is done securely, this framework keeps private the local models (*i.e.*, $\mathcal{L}^1, \mathcal{L}^2, \mathcal{L}^3$) as well as the data of each party (*i.e.*, $\mathcal{A}^1, \mathcal{A}^2, \mathcal{A}^3$). This section presents techniques that implement the framework. First, we discuss the choice for the local model \mathcal{L} . Then, we present a method to securely merge the local models.

3.1 Local Model

As we see from the last paragraph of Section 2, an SVM model is represented by the bias γ , and a list of support vectors, their labels, and coefficients $\{(x_i, d_i, \alpha_i)\}$ such that $\alpha_i > 0$. That is, the global model \mathcal{G} is composed of γ and $\{(x_i, d_i, \alpha_i)\}$ which are computed from the dual problem in Section 2.

Given vertically partitioned data over multiple parties, we cannot use a local SVM model (*i.e.*, computed only over local data) for our local model \mathcal{L} in the framework (Figure 2), because the global SVM model \mathcal{G} cannot be built only from local SVM models;

The globally optimal coefficients (computed by the dual problem) will be different from the locally optimal coefficients computed on local data. Since each party has the data of an attribute subset, the dual problem on the attribute subset will not generate the globally optimal coefficients. Thus, in our framework, the local model \mathcal{L} needs to go beyond the standard SVM model.

To solve the dual problem globally, we need the $m \times m$ matrix $\mathcal{Q} = K(x_i, x_j)d_i d_j$ in Eq.(3) which is computed over the data of all the attributes. The diagonal matrix \mathcal{D} for d_i is given as class labels, thus we only need to compute the global kernel matrix $\mathcal{K} = K(x_i, x_j)$. For linear kernel where $K(x_i, x_j) = x_i \cdot x_j$, the global matrix \mathcal{K} can be directly computed from local matrices because \mathcal{K} is a gram matrix and a gram matrix can be merged from gram matrices of vertically partitioned data, as Lemma 1 proves.

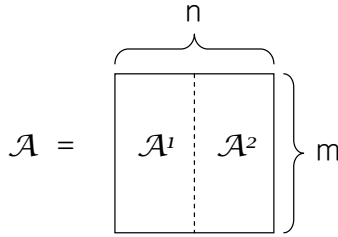


Fig. 3. Vertically partitioned matrix \mathcal{A}

Lemma 1. Suppose the $m \times n$ data matrix \mathcal{A} is vertically partitioned into \mathcal{A}^1 and \mathcal{A}^2 as Figure 3 illustrates. Let \mathcal{K}^1 and \mathcal{K}^2 be the $m \times m$ gram matrices of matrices \mathcal{A}^1 and \mathcal{A}^2 respectively. That is, $\mathcal{K}^1 = \mathcal{A}^1 \mathcal{A}^{1'}$ and $\mathcal{K}^2 = \mathcal{A}^2 \mathcal{A}^{2'}$. Then, \mathcal{K} , the gram matrix of \mathcal{A} , can be computed as follows:

$$\mathcal{K} = \mathcal{K}^1 + \mathcal{K}^2 = \mathcal{A}^1 \mathcal{A}^{1'} + \mathcal{A}^2 \mathcal{A}^{2'} \tag{5}$$

Proof. An $(i, j)^{th}$ element of \mathcal{K} is $x_i \cdot x_j$, where x_i and x_j are i^{th} and j^{th} data vectors in \mathcal{A} . Let x_i^1 and x_i^2 be vertically partitioned vectors of x_i , which are the parts from \mathcal{A}^1 and \mathcal{A}^2 respectively. Then,

$$x_i \cdot x_j = x_i^1 \cdot x_j^1 + x_i^2 \cdot x_j^2 \tag{6}$$

From Eq.(6), each element in \mathcal{K} is equal to the sum of the elements in \mathcal{K}^1 and \mathcal{K}^2 . Thus $\mathcal{K} = \mathcal{K}^1 + \mathcal{K}^2$.

Lemma 1 proves that local gram matrices are sufficient to build the global gram matrix which is the kernel matrix \mathcal{K} for linear kernel. Some popular nonlinear kernel matrices can also be computed from the gram matrix: The polynomial kernel is represented by a dot product of data vectors (i.e., $K(x_i, x_j) = (x_i \cdot x_j + 1)^p$). The RBF kernel can also be represented by dot products (i.e., $K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{g}) = \exp(-\frac{|x_i \cdot x_i - 2x_i \cdot x_j + x_j \cdot x_j|}{g})$). Thus, the local gram matrix from each party is sufficient

to construct the global kernel matrix \mathcal{K} for nonlinear kernels such as polynomial and RBF which can be represented by dot products.

Thus, we use the local gram matrix as the local model \mathcal{L} in our framework. Section 3.2 discusses how to merge \mathcal{L} securely from each party to securely build the global gram matrix. Once the global gram matrix is built, each party can run a quadratic programming solver to compute the global SVM model \mathcal{G} , which will be the same for every party.

3.2 Secure Merge of Local Models

To keep both data and model privacy, it is necessary to securely merge the local models which are the $m \times m$ local gram matrices. A *secure addition* mechanism for $m \times m$ matrices is required. For $k \geq 3$ parties, we developed such a method based on simple secure addition of scalars.

We first describe a simple method to securely calculate the sum of integers from individual sites under the assumption that there are at least three parties and the parties do not collude. We then extend the method so as to seamlessly merge the local models with high efficiency and privacy.

Secure Sum of Integers: Formally, we assume $k \geq 3$ parties, P_0, \dots, P_{k-1} , with party P_i holding value v_i . Together they want to compute the sum $v = \sum_{i=0}^{k-1} v_i$. Assume that the sum v is known to lie in a field \mathcal{F} .

The parties also randomly order themselves into a ring. The ordering can be selected by one of the parties, or by a third party. If the parties cannot decide on a suitable order and no third party can be found, then a protocol developed by Sweeney and Shamos can be used [4] to fix upon a random ordering. The protocol developed by Sweeney and Shamos is quite efficient and requires only $O(k)$ communication. For this paper, to simplify the presentation, without loss of generality, we assume that this order is the canonical order P_0, \dots, P_{k-1} . In general, any order can be decided on. The protocol proceeds as follows:

P_0 randomly chooses a number R , from a uniform distribution over \mathcal{F} . P_0 adds this to its local value v_0 , and sends the sum $R + v_0 \bmod |\mathcal{F}|$ to site P_1 . For the remaining sites $P_i, i = 1, \dots, k - 1$, the algorithm proceeds as follows:

P_i receives

$$V = R + \sum_{j=0}^{i-1} v_j \bmod |\mathcal{F}|.$$

P_i then computes

$$R + \sum_{j=1}^i v_j \bmod |\mathcal{F}| = (v_i + V) \bmod |\mathcal{F}|$$

and passes it to site $P_{i+1} \pmod{k}$. Finally, P_0 , subtracts R from the final message it gets (i.e., adds $-R \pmod{|\mathcal{F}|}$) to compute the actual result.

Clearly, the above protocol correctly calculates the required sum. In order to evaluate the security of the protocol, it is necessary to have a definition of *what* is meant by security. The area of Secure Multi-Party Computation (SMC) provides a theoretical

framework for defining and evaluating secure computation. This protocol can be proven to be completely secure under our assumptions in the SMC framework. A complete proof of security is presented in our technical report [5].

Secure Sum of Matrices: We can extend the secure addition of scalars to securely adding matrices. The key idea is as follows. Suppose a master party wants to merge (*i.e.*, add) its local matrix with those in other slave parties. We assume that the parties have arranged themselves in some sequence and the master initiates the protocol.

1. The master party creates a random matrix of the same size as its local matrix. (The random matrix is hidden from the other parties.)
2. The master party merges (adds) the random matrix with its local matrix, sends the merged matrix to the following slave party.
3. Each slave party, receives the perturbed matrix, merges it with its local matrix and passes it to the following party (the last slave party sends the matrix back to the master).
4. The master subtracts the random matrix from the received matrix, which results in the matrix that adds the matrices of all the parties, without disclosing their local matrices to each other.

All addition is done in a closed field, and subtraction refers to addition of the complement. This secure addition mechanism is proven to be secure and efficient [5]. The extra computation required by the first party is the generation of the random matrix and the final subtraction. In terms of communication overhead, k rounds are required for every party to acquire the summed matrix, where k is the number of participating parties. One problem with the matrix summation method is that it is vulnerable to collusion. The parties preceding and following a party, can collude to recover its local matrix. However, the technique can easily be made collusion resistant to q parties by splitting up the local matrices into q random parts and carrying out the addition protocol q times. The sum of the final matrices from all q rounds gives the real global matrix. As long as the parties are ordered differently in each run, recovery of a local matrix is only possible if collusion occurs in all q rounds. Further details can be found in [5].

3.3 Security

Our method preserves “data privacy”, since only the original party gets to exactly see the data; The local model is directly computed from the local data. However, to ensure “model privacy,” we need at least three participating parties; Each party gets the final global model, which is simply the sum of local models. Thus, with only two parties participating, the other party’s local matrix could be found simply by subtracting the local model from the global model. What is revealed, is the sum of the local models of the other parties. Since the SVM requires knowing the global matrix, this is always possible from the final result as well, and so is unavoidable.

We still need to analyze the effects of knowing the sum of gram matrices computed over the attributes of other parties. In general, the number and type of attributes of the other parties are still assumed to be unknown. As such, the summed matrix does not disclose any attribute values. If the exact number and types of attributes of the other

parties are known, a number of quadratic equations will be revealed in the attribute values; Every cell of the gram matrix corresponds to a dot product – thus the quadratic equation. Since the matrix is symmetric, there are a total of $m(m + 1)/2$ distinct equations (where m is the number of data objects). If the number of total variables (i.e., the sum of all the attributes of other parties) is larger than $m(m + 1)/2$, it is impossible to recover the exact attribute values. Knowing that the matrix is symmetric and positive semidefinite does not disclose further information. While this does reveal more information than strictly necessary, this is a trade off in the favor of efficiency. If complete security is required, the summed matrix could be kept randomly split between two of the parties, and an oblivious protocol run to compute the global model using the generic circuit evaluation technique developed for secure multiparty computation [6, 7].

4 Experiment

The goal of the experiments is simply to demonstrate the scalability of our PP-SVMV. The accuracy will be exactly the same as that of SVM when the data is centralized. We revised the sequential minimal optimization (SMO) source¹ to implement the PP-SVMV. We used the Tic-Tac-Toe data set included in the SMO package for our experiment. We sampled around 958 data objects (m) and extracted around 27 features (n). PP-SVMV generates above 99% with an RBF kernel which is the same as that of the original SVM when the data is centralized.

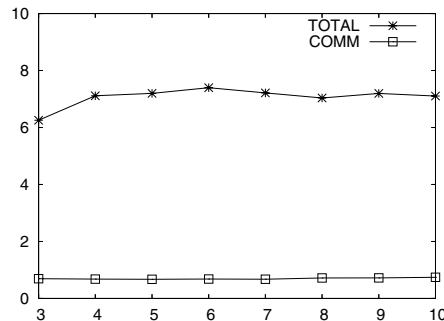


Fig. 4. X-axis:# parties; Y-axis: time (sec.); COMM: communication time; TOTAL: total training time

To check the scalability of the PP-SVMV on an increasing number of participating parties, we vary the number of parties from three to ten. We divide the 27 features about equally between the participating parties. For instance, when ten parties participate, three parties have two features, and the other seven parties have three features. Figure 4 shows results of our experiments: The total training time (including the parallel local computations) hardly changes; SVM is sensitive to the number of data objects more than the features, and the change on the number of features are not visibly influential to

¹ <http://www.datalab.uci.edu/people/xge/svm>

the total training time. The difference of the communication time is also not visible due to the dominant computation time. The results are averaged over ten runs.

5 Related Work

Recently, there has been significant interest in the area of Privacy-Preserving Data Mining. We briefly cover some of the relevant work. Several solution approaches have been suggested. One approach is to perturb the local data (by adding “noise”) before the data mining process, and mitigate the impact of the noise from the data mining results by using reconstruction techniques [8]. However, there is some debate about the security properties of such algorithms [9, 10]. The alternative approach of using cryptographic techniques to protect privacy was first utilized for the construction of decision trees [11]. Our work follows the same approach. A good overview of prior work in this area can be found in [12]. Recently, some alternative techniques such as condensation [13] and transformation [14] have also been proposed.

In terms of data mining problems, work addressed includes association rule mining [15], clustering [16, 17], classification [18], and regression [19, 20]. All of the cryptographic work falls under the theoretical framework of Secure Multiparty Computation. Yao first postulated the two-party comparison problem (Yao’s Millionaire Protocol) and developed a provably secure solution [6]. This was extended to multiparty computations by Goldreich et al. [7]. The key result in this field is that *any* function can be computed securely. Thus, the generic circuit evaluation technique can be used to solve our current problem. However, the key issue in privacy-preserving data mining is one of efficiency. The generic technique is simply not efficient enough for large quantities of data. This paper proposes an efficient technique to solve the problem.

Yu and Vaidya [21] developed a privacy-preserving SVM classification on *horizontally partitioned* data. Since their method is based on the trick of the proximal SVM [3], it is limited to linear classification. Our PP-SVMV is the first one proposing a secure SVM classification on *vertically partitioned* data, which uses the techniques of the secure matrix addition [5] and distributed SVM [22].

6 Conclusion

We propose a scalable solution for privacy-preserving SVM classification on vertically partitioned data (PP-SVMV). With three or more participating parties, our method PP-SVMV securely computes the global SVM model, without disclosing the data or classification information of each party to the others (*i.e.*, keeping the *model privacy* as well as the *data privacy*). Future work may address the idea of efficiently achieving complete security by keeping the global model split between parties as well.

References

1. “Standard for privacy of individually identifiable health information,” *Federal Register*, vol. 66, no. 40, Feb. 28 2001.
2. V. N. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, 1998.

3. G. Fung and O. L. Mangasarian, "Proximal support vector machine classifiers," in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'01)*, 2001.
4. L. Sweeney and M. Shamos, "A multiparty computation for randomly ordering players and making random selections," Tech. Rep. CMU-ISRI-04-126, Carnegie Mellon University, 2004.
5. H. Yu and J. Vaidya, "Secure matrix addition," Tech. Rep., UIOWA Technical Report UIOWA-CS-04-04, <http://hwanjoyu.org/paper/techreport04-04.pdf>, 2004.
6. A. C. Yao, "How to generate and exchange secrets," in *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*. IEEE, 1986, pp. 162–167.
7. O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game - a completeness theorem for protocols with honest majority," in *ACM Symp. on the Theory of Computing*, 1987.
8. R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD Conference on Management of Data*, 2000.
9. H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)*, 2003.
10. Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proc. of ACM SIGMOD Int. Conf. Management of data*, 2005.
11. Yehuda Lindell and Benny Pinkas, "Privacy preserving data mining," *Journal of Cryptology*, vol. 15, no. 3, pp. 177–206, 2002.
12. V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, and Y. Saygin, "State-of-the-art in privacy preserving data mining," *SIGMOD Record*, vol. 33, no. 1, pp. 50–57, Mar. 2004.
13. Charu C. Aggarwal and Philip S. Yu, "A condensation approach to privacy preserving data mining," in *EDBT*, 2004, pp. 183–199.
14. Stanley R. M. Oliveira and Osmar R. Zaiane, "Privacy preserving clustering by data transformation," in *SBBD*, 2004.
15. Jaideep Vaidya and Chris Clifton, "Secure set intersection cardinality with application to association rule mining," *Journal of Computer Security*, to appear.
16. Xiaodong Lin, Chris Clifton, and Michael Zhu, "Privacy preserving clustering with distributed EM mixture modeling," *Knowledge and Information Systems*, to appear 2004.
17. J. Vaidya and C. Clifton, "Privacy-preserving k -means clustering over vertically partitioned data," in *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2003.
18. J. Vaidya and C. Clifton, "Privacy preserving naïve bayes classifier for vertically partitioned data," in *2004 SIAM International Conference on Data Mining*, 2004.
19. A. F. Karr, X. Lin, A. P. Sanil, and Jerry P. Reiter, "Secure regressions on distributed databases," *Journal of Computational and Graphical Statistics*, 2005.
20. A. P. Sanil, A. F. Karr, X. Lin, and J. P. Reiter, "Privacy preserving regression modelling via distributed computation," in *ACM SIGKDD Int. Conf. Knowledge discovery and data mining*, 2004.
21. H. Yu, X. Jiang, and J. Vaidya, "Privacy-preserving svm using nonlinear kernels on horizontally partitioned data," in *Proc. ACM SAC Conf. Data Mining Track*, 2006.
22. F. Poulet, "Multi-way distributed SVM," in *Proc. European Conf. Machine Learning (ECML'03)*, 2003.