

# The applicability of the perturbation based privacy preserving data mining for real-world data

Li Liu <sup>\*</sup>, Murat Kantarcioglu, Bhavani Thuraisingham

*Computer Science Department, University of Texas at Dallas, Richardson, TX 75080, USA*

Available online 18 July 2007

---

## Abstract

The perturbation method has been extensively studied for privacy preserving data mining. In this method, random noise from a known distribution is added to the privacy sensitive data before the data is sent to the data miner. Subsequently, the data miner reconstructs an approximation to the original data distribution from the perturbed data and uses the reconstructed distribution for data mining purposes. Due to the addition of noise, loss of information versus preservation of privacy is always a trade off in the perturbation based approaches. The question is, to what extent are the users willing to compromise their privacy? This is a choice that changes from individual to individual. Different individuals may have different attitudes towards privacy based on customs and cultures. Unfortunately, current perturbation based privacy preserving data mining techniques do not allow the individuals to choose their desired privacy levels. This is a drawback as privacy is a personal choice. In this paper, we propose an individually adaptable perturbation model, which enables the individuals to choose their own privacy levels. The effectiveness of our new approach is demonstrated by various experiments conducted on both synthetic and real-world data sets. Based on our experiments, we suggest a simple but effective and yet efficient technique to build data mining models from perturbed data.

© 2007 Elsevier B.V. All rights reserved.

*Keywords:* Data mining; Privacy; Security

---

## 1. Introduction

Privacy preserving data mining has been studied extensively during the past several years. Several techniques ranging from perturbation to secure multi-party computation have been explored. In this paper, we focus primarily on the perturbation techniques. These techniques are usually used in scenarios where individuals can perturb their private data with some known random noise and report the perturbed data to the data miner. Since the distribution of the added noise is known, the data miner could reconstruct the original distribution using different statistical methods and mine the reconstructed data.

---

<sup>\*</sup> Corresponding author. Tel.: +1 972 345 1016; fax: +1 972 883 2349.

*E-mail addresses:* [liliu@utdallas.edu](mailto:liliu@utdallas.edu) (L. Liu), [muratk@utdallas.edu](mailto:muratk@utdallas.edu) (M. Kantarcioglu), [bhavani.thuraisingham@utdallas.edu](mailto:bhavani.thuraisingham@utdallas.edu) (B. Thuraisingham).

When we examine the details of the perturbation techniques, introducing noise and reconstructing the original distribution emerge as the two most important phases (see [1–3]). During the noise addition phase, random noise from a known distribution (e.g. Gaussian Noise with mean 0, variance  $\sigma^2$ ) is added to privacy sensitive data. Currently, all of the existing noise addition methods add the same amount of noise for all the individuals [1–3]. Such “one privacy level fits all” approach may not be realistic in practice. For example, according to an Internet user survey reported in [4], “it seems unlikely that a one-size-fits-all approach to online privacy is likely to succeed”. To address this issue, we suggest a new noise addition technique that allows each individual to choose his/her own privacy level according to their privacy choices. Our experimental results indicate that our noise addition approach can support various types of users with varying privacy needs without significant degradation in the quality of the data mining results.

In this paper, we also address the applicability of reconstruction techniques in the real world. Our experimental results indicate that the reconstruction techniques may not work well when applied to many real-world data sets. This implies that using the reconstruction methods, such as the one proposed in [1], may not give good data mining results in practice. Instead, for some data mining techniques such as the Naive Bayes classification, we suggest to skip the reconstruction phase and build data mining models from the perturbed data directly. We show the effectiveness of our proposed approach on different real-world data sets.

### *1.1. Our contributions*

Our main contributions in this paper are as follows:

- We present a novel two-phase perturbation method for numerical data that enables individually adaptable privacy protection.
- We carry out an extensive study of the perturbation methods on different data mining tasks using both synthetic and real-world data sets, and show that in many cases skipping the reconstruction phase can give better data mining results.

### *1.2. Organization of the paper*

The paper is organized as follows: In Section 2, we discuss the related work. Section 3 introduces a privacy metric used to measure privacy loss in our experiments. Section 4 describes our two-phase perturbation model and shows the reconstruction results of our model. Section 4.2 describes how the model described in Section 4 could be modified to create an individually adaptable perturbation model. In Section 5, we present our experimental results that are conducted by using various data mining techniques on both synthetic and real-world data sets. In Section 6, we discuss the applicability of the perturbation methods for the real-world data sets. In Section 7, we conclude our paper with our analysis of the interesting experimental results and the discussion of the future work.

## **2. Related work**

Previous work in privacy preserving data mining is based on two approaches. In the first one, the aim is to preserve customer privacy by perturbing the data values [1]. The main premise of this approach is that the perturbed data does not reveal private information, and thus is “safe” to use for data mining. The key result is that the distorted data together with the information on the distribution of the random data used to distort the data can be used to generate an approximation to the original data distribution without revealing the original data values. Later, refinements of this approach tightened the bounds on what private information is disclosed, by showing that the ability to reconstruct the distribution can be used to tighten estimates of the original values based on the distorted data [2]. Since then many good approaches have been proposed [3,5,6]. To the best of our knowledge, the existing perturbation based approaches apply random noise to the data sets without considering different privacy requirements of the different users.

Also one interesting reconstruction approach is proposed by Kargupta et al. in [3]. By using random matrix properties, Kargupta et al. [3] successfully separate the data from the random noise and subsequently reconstructs a good approximation to original data. Recently Huang et al. [6] analyzed the conditions under which the privacy of the underlying data used in perturbation method could be violated. Their results indicate that when the correlations between the data items are high, the original result can be constructed more easily. Some other researchers have proposed different noise addition techniques to protect the private data [7–9]. In [9], authors have proposed a random rotation based perturbation approach, in [7] the authors used multiplicative random projection matrices, and in [8] the authors investigated the distance preserving data perturbation technique from attackers' point of view. Only few of these works have explored the applicability of the perturbation based approaches on variety of real-world data sets.

The perturbation approach has been also applied to Boolean association rules [10,11]. Again, the idea here is to modify data values in such a way that it is difficult to reconstruct the values for any individual transaction, but at same time ensure that the rules learned on the distorted data are still valid. One interesting feature of this work is called randomized response technique [12], which is used to guess a value of "1" from the distorted data which can be considered "0". The same technique also can be used in decision tree classifier [13]. In all these cases the attributes have to be enumerable "0" or "1" values.

The other approach uses cryptographic tools to build data mining models. For example, in [14], the goal is to securely build an ID3 decision tree where the training set is distributed between two parties. Different solutions were given to address different data mining problems using cryptographic techniques (e.g., [15–18]). This approach treats privacy preserving data mining as a special case of secure multi-party computation and not only aims for preserving individual privacy but also tries to preserve leakage of any information other than the final result. Unfortunately, in this approach, the communication and computation cost grow significantly as the number of parties increases.

### 3. Privacy metrics

In [2], authors have proposed a privacy measure based on differential entropy. In this section we briefly review the related concepts that are used in this paper.

The differential entropy  $h(A)$  of a random variable  $A$  is defined as follows:

$$h(A) = - \int_{\Omega_A} f_A(a) \log_2 f_A(a) da \quad (1)$$

where  $\Omega_A$  is the domain of  $A$ . Actually  $h(A)$  can be seen as a measure of uncertainty inherent in the value of  $A$ . Based on this observation, in [2], they proposed to measure the privacy inherent in the random variable  $A$  as  $\Pi(A) = 2^{h(A)}$ . We choose this privacy measure to quantify privacy in our experiments.

For example, using the above definition, a random variable  $U$  distributed uniformly between 0 and  $a$  has privacy  $\Pi(U) = 2^{\log_2(a)} = a$ . Thus if  $\Pi(A) = 1$ , then  $A$  has as much privacy as a random variable distributed uniformly in an interval of length 1. Furthermore if  $f_B(x) = 2f_A(2x)$ , then  $B$  offers half as much privacy as  $A$ . For example, a random variable uniformly distributed over  $[0, 1]$  has half as much privacy as a random variable uniformly distributed over  $[0, 2]$ . In [2] authors have also defined conditional privacy and information loss. For more details, please refer to [2].

### 4. Individually adaptable two-phase perturbation model

The perturbation method is based on introducing noise without significantly changing the distribution of the original data. Subsequently, different statistical techniques are applied to the perturbed data to reconstruct the original distribution. Information loss versus privacy preservation is always a trade off in this method. The extent to which we perturb the original data can dramatically affect the data mining results and subsequently contribute to the potential risk of privacy disclosure. At the same time, choosing the appropriate level of perturbation is not trivial. For example, consider the following common noise addition technique, first proposed in [1].

- Let  $x_1, x_2, \dots, x_n$  be the original values of a one-dimensional distribution as realization of  $n$  independent identically distributed (*iid*) random variables, each has the same distribution as the random variable  $X$ .
- Let  $y_1, y_2, \dots, y_n$  be the random values used to distort the original data,  $y_i$  is the realization of  $n$  independent identically distributed (*iid*) random variables, each has the same distribution as the random variable  $Y$ . Either uniform or gaussian distribution with the following properties is used to generate random variable  $Y$ :
  - Uniform distribution: The random variable has a uniform distribution over an interval  $[-\alpha, +\alpha]$ . The mean of the random variable is 0.
  - Gaussian distribution: The random variable has a normal distribution with  $\mu = 0$  and standard deviation  $\sigma$ .
- Given,  $x_1 + y_1, x_2 + y_2, \dots, x_n + y_n$  (perturbed data  $W$ ) and cumulative probability distribution  $F_Y$  (noise), estimate probability distribution  $F'_X$  (of original data).

We can see that the noise addition procedure described above is only one step; that is, we add the noise to the original data and then apply the reconstruction algorithm to estimate the original distribution. We call this model the one-phase perturbation model.

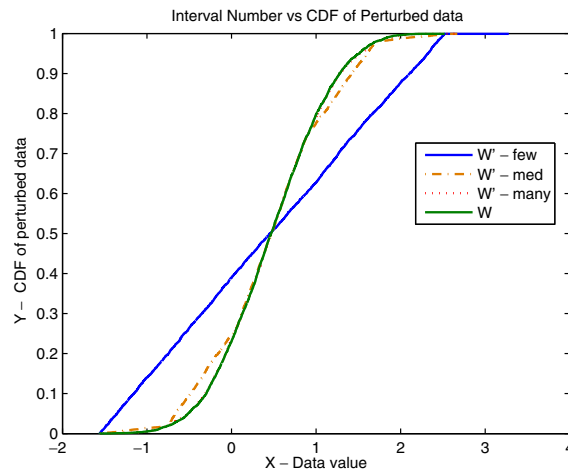


Fig. 1. Cumulative distribution function of perturbed data for various number of intervals.

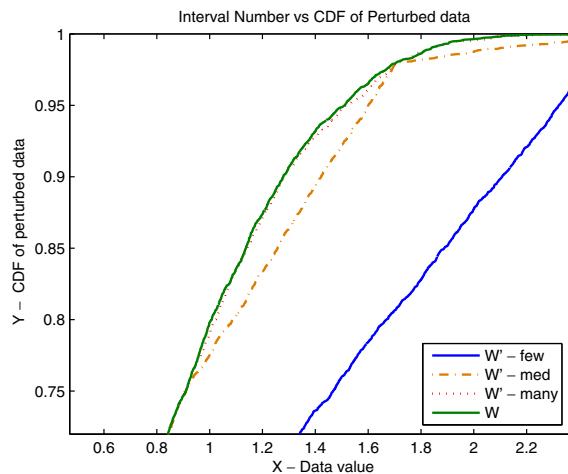


Fig. 2. Cumulative distribution function of perturbed data for various number of intervals, enlarged part.

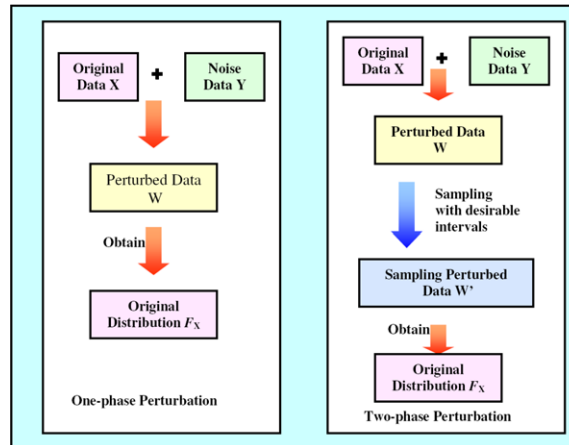


Fig. 3. Two-phase perturbation model.

In this paper we propose a two-phase perturbation model. In this new model, we first divide the domain of the  $W$  into predetermined intervals. After generating the  $w_i = x_i + y_i$ , we calculate the predetermined interval  $[l_k, l_{k+1})$  which  $w_i$  falls. Instead of using  $w_i$  during the reconstruction phase, we use a  $w'_i$  that is generated uniformly from  $[l_k, l_{k+1})$ . Clearly  $w'_i$ 's are created as *iid*. If the intervals used for sampling are chosen small enough, the second phase does not effect the cumulative distribution function (*c.d.f*) of  $W$ . To see the fact that  $W$  and  $W'$  have similar cumulative distribution functions for small intervals, consider the relationship between the *c.d.f* of  $W$  and  $W'$  shown in Figs. 1 and 2. In these figures,  $W'$ -few corresponds to the case where total three intervals are used to create the  $W'$  data set; similarly  $W'$ -med is created using six intervals;  $W'$ -many is created using 20 intervals. We can see that when the interval number is increasing, the *c.d.f* of  $W'$  is getting closer to the *c.d.f* of  $W$ . In practice, we set the number of intervals such that *c.d.f* of  $W'$  is close to  $W$ .

Fig. 3 shows the processes of both one-phase and two-phase perturbation models. Below, we show that the reconstruction method proposed in [1] can be successfully used to reconstruct original distribution in our two-phase perturbation model.

#### 4.1. Original distribution reconstruction for two-phase perturbation model

As we mentioned above, the second important step of the perturbation based approaches is to reconstruct the original data distribution. Since data mining models is built using the reconstructed data, the techniques of reconstructing the original data distribution are very important in learning useful models.

To show that our two-phase noise addition approach described in this section could be used in practice, we run experiments using the pioneering Bayesian inference base reconstruction method proposed by Agrawal et al. [1]. (Please see the Appendix for more details.)

For accurate comparison, we use the same experimental set up that is also used in [1]. Let  $X$  be the original distribution. We create the original data set by creating 10K records from the triangle shape distribution from  $[0, 1]$ . Let  $Y$  be the noise distribution where we use the following parameter values:

- Gaussian distribution: We use a normal distribution with mean  $\mu = 0$  and standard deviation  $std = 0.25$ .

We keep the original data set and the first step perturbed data set  $W$ . We create the data  $W'$  as follow:

- (1) We divide  $[-0.5, 1.5]$  (the domain of  $W$ ) into 40 intervals.
- (2) Given the each perturbed data point  $w_i$ , we sample uniformly from the corresponding interval where  $w_i \in [l_k, l_{k+1})$ .

The experimental results shown in Fig. 4 indicate that Agrawal et al.'s Bayesian inference based reconstruction technique can still successfully reconstruct the original data distribution. This is not surprising, because

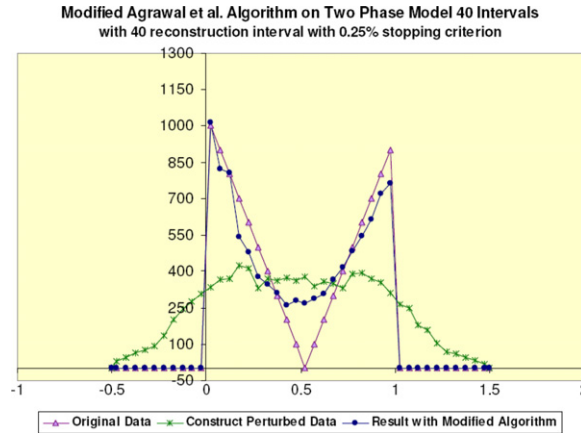


Fig. 4. Reconstructing the original data distribution in two-phase perturbation model using the Agrawal et al. algorithm.

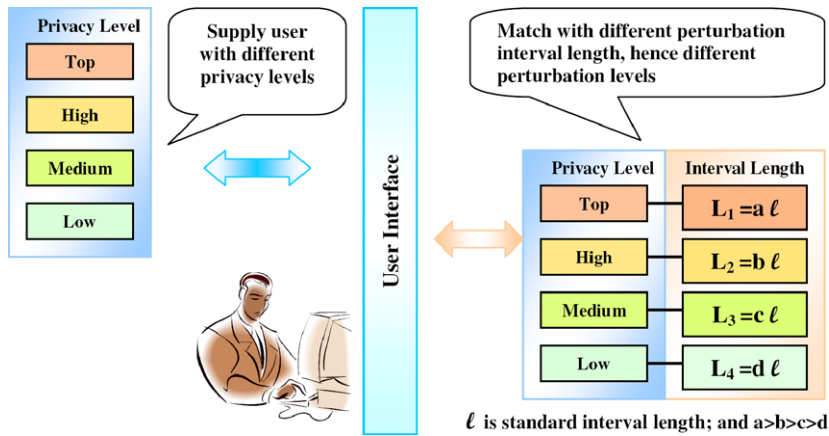


Fig. 5. Privacy level vs perturbation levels (i.e. interval length).

Bayesian inference based approach uses the *c.d.f.* of the perturbed data, and clearly, our two-phase algorithm does not significantly change the underlying cumulative distribution of the  $W$ . Therefore the Bayesian inference based reconstruction approach could still achieve good reconstruction results, as shown in Fig. 4. For more experimental results, please refer to our earlier work [19].

#### 4.2. Individually adaptable perturbation model using two-phase perturbation

Our two-phase perturbation model described in Section 4 could be easily modified to incorporate different individual privacy preferences. Note that for each  $w_i$ , we are sampling a random point from the interval where  $w_i \in [l_k, l_{k+1})$ . Clearly this sampling could be done using different intervals for different users. For example, a user who is more cautious could use the interval  $[l_{k-1}, l_{k+2})$  for generating  $w'_i$  (i.e., twice the original size). Using this fact, we can describe individually adaptable perturbation method as follows:

- (1) The system first adds a random noise  $Y = (y_1, y_2, \dots, y_i, \dots, y_n)$  to the original data values  $X = (x_1, x_2, \dots, x_i, \dots, x_n)$  to get  $W = (w_1, w_2, \dots, w_i, \dots, w_n)$ .
- (2) User  $i$  chooses his/her privacy level among various privacy levels shown in Fig. 5.
- (3) Based on the user's privacy level choice, the system applies an interval length  $[l_i, l_j)$  that correspondent to the chosen privacy level. Later on,  $w'_i$  is created by sampling uniformly from the interval  $[l_i, l_j)$
- (4)  $w'_i$  value is sent to the data miner.

Table 1  
Description of the synthetic data set attributes

Attribute	Description
Salary	Uniformly distributed from 20K to 150K
Commission	Salary $\geq 75K \Rightarrow commission = 0$ else uniformly distributed from 10K to 75K
Age	Uniformly distributed from 20 to 80
Elevel	Uniformly chosen from 0 to 4
Car	Uniformly chosen from 1 to 20
Zipcode	Uniformly chosen from 9 zipcodes
Hvalue	Uniformly distributed from $k \times 50K$ to $k \times 150K$ , where $k \in \{0 \dots 9\}$ depends on zipcode
Hyears	Uniformly distributed from 1 to 30
Loan	Uniformly distributed from 0 to 500K

The relationship between different privacy preference and perturbation level is shown in Fig. 5. Each privacy level described in Fig. 5 will have a corresponding interval length, and this different interval length for different individuals will enable the individual adaptability.

Once the data miner receives the  $w'_i$  values, Bayesian based reconstruction approach described in Section 4 could be used to reconstruct the original data distribution. Clearly if the length of the chosen intervals is large, the given data will look more like a random sample. If the length of the chosen interval is small  $w'_i$  will be very closed to the  $w_i$ . At the same time, if the number of users who are choosing to have a large interval to sample from is small, cumulative distribution function  $F_W$  and  $F_{W'}$  will not be too much different. In our model, customers can choose different interval lengths to modify their privacy levels. This privacy level could be measured using the metrics described in Section 3. To test the effectiveness of our individually adaptable perturbation method, we have conducted extensive experiments as described in Section 6.

## 5. Data mining experiment results

Although, we wish to protect individual privacy by carrying out privacy preserving data mining, our ultimate goal is to obtain accurate data mining results. Especially, compared to the results obtained from the original data set, we do not want the perturbation technique to have any significant effect on the data mining results. To test the effect of privacy preserving data mining on accuracy, we build decision trees and naive Bayes classifiers on both synthetic and real-world data sets.

Before, we discuss the results, we give an overview of the data sets used in our experiments.

### 5.1. Overview of the data sets

For comparison purposes, we used the synthetic data which was used by Agrawal et al. in [1]. The data set has nine attributes, e.g. age, salary, education level, house value, loan and so on. We describe these attributes in Table 1. There are five functions used as data mining classification functions, we have listed them in the Appendix for completeness.

We have chosen three real-world data sets for our experiments from University of California, Irvine, machine learning database repository,<sup>1</sup> We briefly summarize the properties of these data sets below.

- *Income data*

The data set was extracted from the census database 1994, and it is used to predict whether the income exceeds 50K annually. The data set is also cited as the ‘‘Adult’’ database. The original data set has fourteen attributes, six are continuous attributes and eight are nominal attributes. We only use the six continuous attributes in our experiments. We used 32,561 instances for training and 16,281 for testing purposes.

<sup>1</sup> <http://www.ics.uci.edu/mllearn/MLSummary.html>.

- *Haberman survival data*

The data set contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. It has 306 instances, and three numerical attributes. The class label indicates whether the patients survive more than five years after the operations.

- *Liver data*

The data is cited as liver-disorders database, and created by BUPA Medical Research Ltd. It has 345 instance and five numerical attributes which are all blood test results that may be sensitive to liver disorders caused by excessive alcohol consumption. The class label is whether the study subjects have drunk more than 2.5 half-pint equivalents of alcoholic beverages per day.

### 5.2. Reconstruction of the original distribution

As described above, in each data set there are more than one attribute, and for training data, each instance has its class label. Clearly, when and how to reconstruct the original distribution can effect the data mining results. In [1], authors have listed three ways, Global, ByClass and Local. We repeat it here for the sake of completeness.

- *Global*

Reconstruct the original distribution for each attribute using the complete perturbed training data.

- *ByClass*

For each attribute, first split the training data by class, then reconstruct the distributions separately for each class.

- *Local*

In the Agrawal et al.'s work [1], they build a decision tree classifier, so they repeat the ByClass at each node.

In [1], the authors mentioned that the global method does not give good results, and local method's computation cost is high due to repeated the reconstruction process. For these reasons, we choose to use ByClass method as a reasonable compromise between efficiency and accuracy to reconstruct the original distribution.

### 5.3. Data mining experimental results on synthetic data

For the synthetic data experiments, we have generated 100,000 records, and randomly choose 66,000 of those records are used for training, and 34,000 of those records are used for testing purposes. We used WEKA [20] data mining software to run decision tree and Naive Bayes classifiers on our reconstructed data, and reported the experiment results on the test data. We have compared the performance of these two classifiers on various scenarios with the results reported in [1]. In our experiments, we assumed that there are four different user types with different privacy requirements. Table 2 shows the four different cases with different percentage of people with different attitudes towards privacy. Compared to the normal people, cautious people may want to preserve more privacy. So when we are sampling the perturbed data, we use a smaller interval length for normal people; and we use twice the interval length for cautious people; and so on. Based on the four different groups, we generate four different data sets, named case 1 to

Table 2  
Privacy measure of different data sets used for data mining

Attitude	Interval	Case 1	Case 2	Case 3	Case 4	One phase
Paranoid	25	0%	0%	5%	5%	N/A
Extra cautious	50	0%	0%	5%	5%	N/A
Cautious	100	0%	10%	10%	20%	N/A
Normal	200	100%	90%	80%	70%	N/A
Privacy loss	N/A	0.2902	0.2809	0.2788	0.2705	0.2816

case 4, and with different percentages are assigned to different categories. Since different attributes have different lengths in our data set, we set the intervals for each type by calculating domain-size divided by the number of intervals. For example, in case 4, 100K perturbed record data set is created as follows: we sample 70K perturbed data records using 200 intervals, then we sample 20K records using 100 intervals, and then we sample 5K records using 50 intervals, the last 5K records are sampled using 25 intervals.

We apply the privacy metrics described in Section 3 to four different cases. We used the mutual information estimation technique given in [21] to calculate the privacy loss (shown in Table 2). We can see case 1 has the most privacy loss, case 2 has less, and so on, then case 4 has the least privacy loss.

Although it looks like that the privacy enhancement between case 1 and case 4 shown in Table 2 is small, the main difference occurs due to increased privacy provided for paranoid, cautious and extra cautious user types. As it can be seen from Table 2, both case 1 and case 4 have almost 70% normal users.

Fig. 6 shows our data mining results. First of all, our results indicate that the differences between the outcomes of our two-phase perturbation model compared to those from one-phase perturbation model are not significant. Among these five functions we tested, function 2 and 3 are not easy to learn. (Please see the discussion in [1].) Although the models built using the reconstructed data have good prediction accuracy, there are not as good compared to the models built from the original data. This is especially true for the decision tree classifier built by using the Gaussian noise added data set. Clearly privacy has a price. Based on our experimental results, using this synthetic data set, the performance of decision tree classifiers is better than the Naive Bayes classifiers.

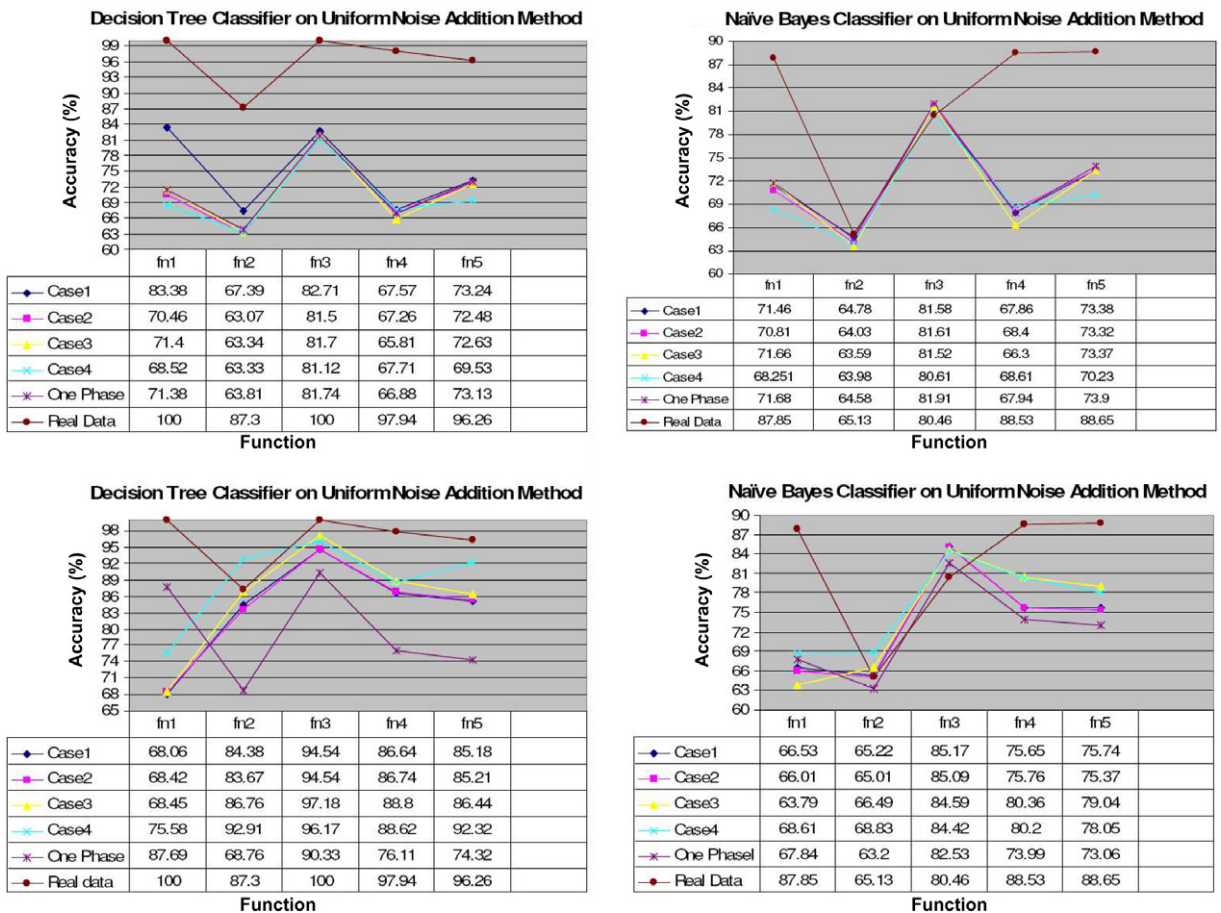


Fig. 6. Data mining accuracy of individually adaptable model on synthetic data.

Table 3  
Data mining accuracy for real-world data sets

Data	Origin	Perturbed	Recon. test 1	Recon. test 2	Recon. test 3
<i>Decision tree C4.5 classifier accuracy (%)</i>					
Income	83.77	78.39	91.43	26.30	40.16
Haberm	71.89	77.78	99.67	75.49	75.49
Liver	79.42	77.10	97.68	25.22	23.48
<i>Naive Bayes classifier accuracy (%)</i>					
Income	79.72	77.89	88.74	24.06	28.12
Haberm	74.84	76.14	99.67	75.49	75.49
Liver	79.97	78.26	99.71	24.35	24.35

#### 5.4. Experimental results for real-world data

We use the three real-world data sets, Income, Liver and Haberman, which we have described in the previous section in our experiments. We added gaussian noise to the original data with signal to noise ratio (SNR) is equal to 1.0.<sup>2</sup> In all of our experiments, we divided each data into training and testing sets. For Income data set, we used the default partition, training set has 32,561 instances and test set has 16,281 instances. Since the income data set is quite large, using cross-validation or this one fix partition of training and testing sets does not affect the predication accuracy significantly. For Liver and Haberman data sets, we randomly choose 2/3 of the instances as the training set, and the remaining 1/3 instances as the test set. For both Haberman and Liver data sets, we have tried different random partition, the predication accuracy vary slightly among different partitions. In this paper, we report the predication accuracy obtained by this experiment set up unless specified otherwise.

The interesting experimental results are shown in Table 3. The first column is the data mining accuracy obtained from original data set. We use this column as the base line. The second column is the data mining accuracy obtained from perturbed data sets. We can see that accuracy is lower for the perturbed data, except for the Haberman data set. We also build classifiers from reconstructed data, and perform three different tests to compare data mining accuracy. In the first test, we run the classifier on the reconstructed test data; in the second test, we run the classifier on original test data; and in the third test, we run the classifier on the perturbed test data.

In the first test, all the results have higher accuracy than the ones obtained from the original data sets. In the second and third tests, two data sets get very low data mining accuracies. Only the Haberman data set achieves an accuracy similar to what we obtain from the perturbed data. This result raises the following question: Is the reconstruction method applicable for all the real-world data sets? Please note that for the synthetic data used in [1], every attribute is uniformly distributed in a given range. In this case, after reconstruction, the instances of the reconstructed data set are similar to the original data set due to the fact that every attribute is independently and uniformly distributed. This is not realistic in many situations.

In this paper, we have conducted experiments on three real-world data sets. These data sets may have their limitations, but still our experimental results indicate that perturbation methods do not work well when applied to some real-world data sets. This implies that for some data distributions, we should not try to solve the hard reconstruction problem as an intermediate step. Although, at this point we can not simply say that the reconstruction method is not applicable for real-world data sets. Clearly, caution needs to be exercised before applying reconstruction phase in practice.

## 6. Applicability of reconstruction method in PPDM for real-world data

As shown in the previous section, reconstruction phase of the perturbation based privacy preserving data mining may not be applied in some cases. The question is what can we do if the reconstruction phase fails for the data set we are interested in (e.g. Income data and Liver data in our case)?

<sup>2</sup> By default, SNR 1.0 is used for all experiments. Please see the Appendix for the definition of SNR.

Our solution is rather simple. We suggest a direct approach for such hard reconstruction cases. Our approach is influenced by the Vapnik's following comments [22]: "If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information sufficient off a direct solution but is insufficient for solving a more general intermediate problem."

In the spirit of the above observation, for the data sets where reconstructing the original data distribution is hard, instead of using the reconstruction as an intermediate step, we can try to build data mining models from the perturbed data. By learning data mining models from perturbed data directly, we intend to solve the original problem (e.g. building data mining models) without trying to solve a hard intermediate problem (e.g. reconstruction of the original data distribution). Below, we show that this is not only feasible but it may be preferable in the case of learning Naive Bayes Classifiers from perturbed data directly.

Also Agrawal et al. [1] observed the phenomenon that mining directly from the perturbed data sets can obtain fair data mining results for synthetic data sets. Compared to their work, we have performed extensive experiments using different data mining techniques on several different real-world data sets. We believe that our results provide more evidence to support this phenomenon.

### 6.1. Naive Bayes classifier construction over perturbed data

We do not need to reconstruct the original data distribution to build a Naive Bayes classifier from the perturbed data. Instead, we can directly use the Naive Bayes algorithm on the perturbed data. Before showing why such a direct approach is feasible, we give an overview of the Naive Bayes classifier construction, then we explain why the Naive Bayes algorithm can be applied directly on the perturbed data.

Using the Bayesian approach, the Naive Bayes classifier labels a new instance by assigning the most probable class value. Besides, it assumes that attribute values are conditionally independent given the class value to simplify the estimation of the required probabilities. Using the above assumptions, Naive Bayes classifier selects the most likely classification  $C_{nb}$  as [23]

$$C_{nb} = \operatorname{argmax}_{C_j \in C} P(C_j) \prod_i P(X_i | C_j) \quad (2)$$

where  $X = X_1, X_2, \dots, X_n$  denote the set of attributes,  $C = C_1, C_2, \dots, C_d$  denote the finite set of possible class labels, and  $C_{nb}$  denote the class label output by the Naive Bayes classifier.

Clearly, we need to calculate the probabilities  $P(X_i = x | C_j)$  used in Eq. (2) based on the training data. In practice, for numeric attributes,  $P(X_i = x | C_j)$  is estimated by using Gaussian distribution  $N(\mu_{ij}, \sigma_{ij}^2)$ . The required parameters,  $\mu_{ij} = E(X_i | C_j)$  and  $\sigma_{ij}^2 = \operatorname{Var}(X_i | C_j)$  are estimated by using the training data.

In our case, we need to estimate  $\mu_{ij}$  and  $\sigma_{ij}^2$  for each attribute  $X_i$  and for each class label  $C_j$  using the perturbed numeric data to construct a Naive Bayes classifier. In the perturbed data case, instead of the original attribute value  $X_i$ , we only see the  $W_i = X_i + R$  values. Let  $w_{ij}^t$  be the  $i$ th attribute value of the  $t$ th training data instance with class label  $C_j$ . In addition, we assume that there are  $n$  instances with class label  $C_j$ .

We also know that  $w_{ij}^t = x_{ij}^t + r_{ij}^t$  where  $r_{ij}^t$  is the randomly generated noise with mean zero and known variance  $\sigma_R^2$ . Using the above facts, we can show that the expected value of  $\bar{w}_{ij} = \frac{1}{n} \cdot \sum_{t=1}^n (w_{ij}^t)$  is equal to  $\mu_{ij}$ .

Since the sample variance  $S^2 = \frac{1}{n-1} \cdot \sum_{t=1}^n (w_{ij}^t - \bar{w}_{ij})^2$  has an expected value  $\sigma_{ij}^2 + \sigma_R^2$ , we can use  $S^2$  and the known  $\sigma_R^2$  to estimate the  $\sigma_{ij}^2$  (i.e. use  $S^2 - \sigma_R^2$  to estimate  $\sigma_{ij}^2$ ).

As a result, as long as we do not change the class labels, we can directly construct Naive Bayes classifier from the perturbed data. Even more, since the parameter estimations done by using the perturbed data and the original data have the same expected values, we should be able to get similar classification accuracy in both cases. In addition, since estimating mean and variances is an easier problem than original distribution reconstruction, we may get a better accuracy.

The test results reported in Table 3 verifies the above intuition. The reported results using the Naive Bayes classifier from the WEKA machine learning toolkit [20] indicate that there is very little difference in terms of accuracy when we mine the Naive Bayes model directly from the perturbed data.

## 6.2. Directly apply data mining technique to perturbed data sets

From the above analysis of Naive Bayes classifier, we can see that when we only introduce white noise, directly applying data mining techniques to perturbed data is an effective and efficient approach. Of course our analysis are only valid for Naive Bayes classification. Unfortunately, it is not trivial to extend such analytical analysis for more complex data mining models. Instead, to investigate the effect of directly mining perturbed data in general, we have performed experiments using different data mining techniques.

### 6.2.1. Experimental results of data mining accuracy vs data set size

We used three different classifiers (e.g. decision tree C4.5, Naive Bayes (NB) and neural network, multilayer perceptron (MLP) learning) on different size perturbed data sets randomly selected from Income data. In these experiments, we use Gaussian noise with signal to noise ration set to 1.0. In our experiments, we used different size, (e.g. 1k, 3k, ..., 11k), of perturbed data as training data set, and then test the model accuracy either on original data set or perturbed data set. The experimental results have been reported in Table 4. The data mining accuracy fall in the rage from 76.22 to 79.76, compared with the data mining accuracy obtained from original 3k data set, which are 83.29 for tree C4.5, 79.66 for Naive Bayes, and 83.43 for neural network. Especially for Naive Bayes classifier, the accuracy difference is less than 1%. This confirms our analysis in the previous section. Also it is interesting to see that for income data set, number of instances bigger than 1K did not have any significant effect in terms of accuracy.

### 6.2.2. Experimental results of data mining accuracy vs. SNR values

As we mentioned before, *SNR* denotes to signal to noise ratio. This is an important feature when we use noise to disguise the original data. High *SNR* means that the variance of added noise data is low, and the effect on original data is low. This definitely effects the data mining accuracy when mining is done using the perturbed data. The Fig. 7 shows the data mining accuracy vs. the *SNR* values. We can see that for all three classifiers as the *SNR* value decreases, the data mining accuracy decreases as well.

### 6.2.3. Experimental results of data mining accuracy vs. different real-world data sets

The results reported in Tables 5 and 6 indicate that directly mining perturbed data for the cases where reconstruction phase do not work is a viable alternative even for some other data mining tasks.

### 6.2.4. Individually adaptable perturbation model on perturbed data sets

We also applied our two-phase individually adaptable perturbation model on these three real-world data sets. Similar to the experiments of synthetic data in Section 5.3, we assumed that there are four different user types (shown in Section 5.3). We applied different data mining techniques on these four different cases and compared to the results obtained using one-phase perturbation model. The data mining result accuracy has

Table 4  
Different data mining classifier accuracy on different size perturbed data sets

	Data size					
	1K	3K	5K	7K	9K	11K
TreeC4.5 on original	79.69	79.21	79.09	78.29	78.47	78.85
TreeC4.5 on perturbed	76.65	76.22	77.13	77.16	77.67	77.8
NB on original	78.29	78.43	78.24	78.21	78.18	78.24
NB on perturbed	77.34	76.33	77.42	77.37	77.46	77.46
MLP on original	79.76	77.45	78.5	78.39	78.79	77.86
MLP on perturbed	76.96	77.44	77.75	77.98	78.12	77.78

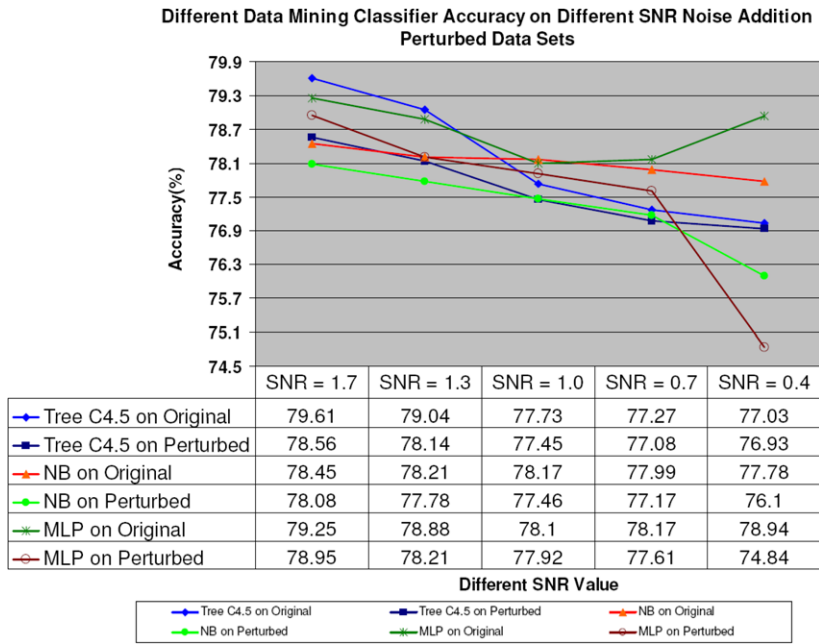


Fig. 7. Data Mining accuracy of applying data mining techniques directly on 10k perturbed training data set with different SNR values.

Table 5

Accuracy of data mining models built from perturbed data directly and tested both on original and perturbed data for different real-world data sets

	TreeC4.5 original	TreeC4.5 perturbed	NB original	NB perturbed	MLP original	MLP perturbed
Income	77.73	77.45	78.17	77.46	78.1	77.92
Haberman	75.49	75.49	75.49	77.45	78.43	74.51
Liver	76.52	73.91	79.13	79.13	84.35	74.78

Table 6

Data mining accuracy on original data with different real-world data sets

	TreeC4.5	NB	MLP
Income	83.29	79.66	82.43
Haberman	76.19	75.82	74.84
Liver	77.39	78.84	77.10

Table 7

Different data mining classifier accuracy with individually adaptable perturbation model on Income perturbed data set

	Case 1	Case 2	Case 3	Case 4	One phase
Tree C4.5 on original	78.69	78.32	78.41	78.79	78.28
Tree C4.5 on perturbed	77.73	77.61	77.57	77.46	77.68
NB on original	78.21	78.19	78.16	77.95	78.2
NB on perturbed	77.48	78.19	77.51	77.87	77.4
MLP on original	77.15	77.16	77.2	77.46	77.17
MLP on perturbed	77.53	77.56	77.59	77.49	77.57
Privacy loss	0.1592	0.1590	0.1577	0.1513	0.1597

Table 8  
Different data mining classifier accuracy with individually adaptable perturbation model on Liver perturbed data set

	Case 1	Case 2	Case 3	Case 4	One phase
Tree C4.5 on original	80	80	80	80	80
Tree C4.5 on perturbed	80.87	80	80	74.78	80
NB on original	78.26	78.26	78.26	78.26	78.26
NB on perturbed	79.13	78.26	79.13	79.13	78.26
MLP on original	80	80.87	80.87	77.39	79.13
MLP on perturbed	78.26	75.65	78.26	74.78	74.78
Privacy loss	0.1292	0.1289	0.1286	0.1205	0.1306

Table 9  
Different data mining classifier accuracy with individually adaptable perturbation model on Haberman Survival perturbed data set

	Case 1	Case 2	Case 3	Case 4	One phase
Tree C4.5 on original	70.59	71.57	71.57	71.57	71.57
Tree C4.5 on perturbed	73.53	71.57	71.57	71.57	71.57
NB on original	70.59	70.59	70.59	72.55	70.39
NB on perturbed	66.67	66.67	66.67	68.63	71.57
MLP on original	73.53	73.53	74.51	72.55	71.57
MLP on perturbed	68.63	67.65	68.63	67.65	68.63
Privacy loss	0.1398	0.1393	0.1387	0.1366	0.1442

shown in Tables 7–9. We can see that our two-phase adaptable model enables users to have more privacy choices without reducing the data mining accuracy.

## 7. Conclusions and future work

Due to the varying privacy needs of different individuals, the one-size-fits-all approach is not realistic in many privacy preserving data mining tasks. In order to address this problem, we have proposed a new perturbation method for privacy preserving data mining that can provide individually adaptable privacy protection. Our method enables users to choose different privacy levels without significant data mining performance degradation.

In order to confirm the effectiveness of our method, we have carried out extensive experiments under different data mining scenarios. Our results indicate that if only a small number of people choose to have high levels of privacy (i.e., more perturbation), we can still find useful data mining results.

Similar to the previous work in this area, our method does not address the multiple attribute case. The obvious solution of adding independent random noise to each attribute may not offer good privacy protection for high dimensional data, since outliers can be easily detected. Although existing methods could be used by adding a random noise using a multivariate distribution, construction process could require large amounts of data. As part of our future work, we plan to investigate a different approach. Instead of trying to come up with a noise addition method that can be used for general data mining tasks, we plan to develop a noise addition method specific to different data mining techniques.

Reconstruction is a very important step for the perturbation based PPDM approaches. We have found that when applied to real-world data sets reconstruction could be a problem. To address this problem, we proposed PPDM methods which skip this reconstruction step and compute the data mining results directly. We prove the viability of directly constructing data mining models on real-world data sets. In the future, we plan to investigate different data mining methods in this direction.

## Appendix A. Agrawal et al.'s algorithm – Bayes estimation based approach

In Agrawal et al.'s work [1], the Bayes theorem based algorithm is described as follow: Given the noisy data probability distribution  $F_Y$ , and the random values of perturbed data ( $x_i + y_i = w_i$ ), the density function of the original data set can be estimated as below:

$$f'_x(a) = \frac{1}{n} \sum_{i=1}^n \frac{f_Y(w_i - a)f_X(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z)f_X(z)dz}$$

### Algorithm 1. Agrawal and Srikant Bayes theorem reconstruction algorithm

---

Initial  $f_X^0 = \text{Uniform distribution}$

Iteration number  $j := 0$

$$f_X^{j+1}(a) = \frac{1}{n} \sum_{i=1}^n \frac{f_Y(w_i - a)f_X^j(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z)f_X^j(z)dz}$$

$j := j + 1$ ;

Perform this iteration until the stopping criterion is met.

---

Running the algorithm on a large data set, and set the stop criterion to a small number, e.g. 0.25%, the algorithm would calculate an estimated density function that is very close to the real density function.

## Appendix B. Signal to Noise Ratio

Signal to noise ratio (SNR) is a very important index in the noise addition techniques. When using Gaussian noise, the variance  $\sigma^2$  can dramatically affect the results. This was used by Kargupta et al.'s in their work [3]. SNR is the term to quantify the relative amount of noise added to actual data.

$$\text{SNR} = \frac{\text{Variance of Actual Data}}{\text{NoiseVariance}}$$

## Appendix C. Data mining functions used for synthetic data

There are five functions used as data mining classification functions in mining synthetic data. These functions are well described in the work [1], shown in Table C.1.

Table C.1  
Description of five data mining functions [1]

	Group A	Group B
Function 1	$(\text{age} < 40) \vee (60 \leq \text{age})$	Otherwise
Function 2	$((\text{age} < 40) \wedge (50\text{K} \leq \text{salary} \leq 100\text{K})) \vee$ $((40 \leq \text{age} < 60) \wedge (75\text{K} \leq \text{salary} \geq 125\text{K})) \vee$ $((\text{age} \geq 60) \wedge (25\text{K} \leq \text{salary} \leq 75\text{K}))$	Otherwise
Function 3	$((\text{age} < 40) \wedge (((\text{elevel} \in [0, \dots, 1]) \wedge (25\text{K} \leq \text{salary} \leq 75\text{K})) \vee$ $((\text{elevel} \in [2, \dots, 3]) \wedge (50\text{K} \leq \text{salary} \leq 100\text{K})))) \vee$ $((40 \leq \text{age} < 60) \wedge (((\text{elevel} \in [1, \dots, 3]) \wedge (50\text{K} \leq \text{salary} \leq 100\text{K})) \vee$ $((\text{elevel} = 4) \wedge (75\text{K} \leq \text{salary} \leq 125\text{K})))) \vee$ $((\text{age} \geq 60) \wedge (((\text{elevel} \in [2..4]) \wedge (50\text{K} \leq \text{salary} \leq 100\text{K})) \vee$ $((\text{elevel} = 1) \wedge (25\text{K} \leq \text{salary} \leq 75\text{K}))))$	Otherwise
Function 4	$(0.67 \times (\text{salary} + \text{commission}) - 0.2 \times \text{loan} - 10\text{K}) > 0$	Otherwise
Function 5	$(0.67 \times (\text{salary} + \text{commission}) - 0.2 \times \text{loan} + 0.2 \times \text{equity} - 10\text{K}) > 0$ where $\text{equity} = 0.1 \times \text{hvalue} \times \max(\text{hyears} - 20, 0)$	Otherwise

## References

- [1] R. Agrawal, R. Srikant, Privacy-preserving data mining, in: SIGMOD Conference, 2000, pp. 439–450.
- [2] D. Agrawal, C.C. Aggarwal, On the design and quantification of privacy preserving data mining algorithms, in: PODS, ACM, 2001.
- [3] H. Kargupta, S. Datta, Q. Wang, K. Sivakumar, On the privacy preserving properties of random data perturbation techniques, in: ICDM, IEEE Computer Society, 2003, pp. 99–106.
- [4] L.F. Cranor, J. Reagle, M.S. Ackerman, Beyond concern: Understanding net users' attitudes about online privacy, CoRR cs.CY/9904010.
- [5] A. Evfimovski, J. Gehrke, R. Srikant, Limiting privacy breaches in privacy preserving data mining, in: Proceedings of the ACM SIGMOD/PODS Conference, San Diego, CA, 2003, pp. 211–222.
- [6] Z. Huang, W. Du, B. Chen, Deriving private information from randomized data, in: SIGMOD Conference, 2005, pp. 37–48.
- [7] K. Liu, H. Kargupta, J. Ryan, Random projection-based multiplicative data perturbation for privacy preserving distributed data mining, IEEE Transactions on Knowledge and Data Engineering (TKDE) 18 (1) (2006) 92–106. <http://doi.ieeecomputersociety.org/10.1109/TKDE.2006.14>.
- [8] K. Liu, C. Giannella, H. Kargupta, An attacker's view of distance preserving maps for privacy preserving data mining, in: Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'06), Berlin, Germany, 2006.
- [9] K. Chen, L. Liu, Privacy preserving data classification with rotation perturbation, in: ICDM, 2005, pp. 589–592.
- [10] A.V. Evfimovski, R. Srikant, R. Agrawal, J. Gehrke, Privacy preserving mining of association rules, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 217–228.
- [11] S. Rizvi, J.R. Haritsa, Maintaining data privacy in association rule mining, in: VLDB, Morgan Kaufmann, 2002 pp. 682–693.
- [12] A.C. Tamhane, Randomized response techniques for multiple sensitive attributes, The American Statistical Association 76 (376) (1981) 916–923.
- [13] W. Du, Z. Zhan, Using randomized response techniques for privacy-preserving data mining, in: KDD, 2003, pp. 505–510.
- [14] Y. Lindell, B. Pinkas, Privacy preserving data mining, in: M. Bellare (Ed.), CRYPTO, Lecture Notes in Computer Science, vol. 1880, Springer, 2000, pp. 36–54.
- [15] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, M.Y. Zhu, Tools for privacy preserving data mining, SIGKDD Explorations 4 (2) (2002) 28–34.
- [16] M. Kantarcioglu, C. Clifton, Privacy-preserving distributed mining of association rules on horizontally partitioned data, in: DMKD, 2002.
- [17] M. Kantarcioglu, C. Clifton, Privately computing a distributed  $k$ -nn classifier, in: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), PKDD, Lecture Notes in Computer Science, Vol. 3202, Springer, 2004, pp. 279–290.
- [18] J. Vaidya, C. Clifton, Privacy-preserving-means clustering over vertically partitioned data, in: KDD, 2003, pp. 206–215.
- [19] L. Liu, B. Thuraisingham, M. Kantarcioglu, L. Khan, An adaptable perturbation model of privacy preserving data mining, in: ICDM Workshop on Privacy and Security Aspects of Data Mining, Huston, TX, US, 2005.
- [20] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufmann, San Francisco, 2005.
- [21] R. Moddemeijer, On estimation of entropy and mutual information of continuous distributions, Signal Processing 16 (3) (1989) 233–246.
- [22] V.N. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
- [23] T.M. Mitchell, Machine Learning, Mcgraw-hill, 1997. <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/mlbook.html>.



**Li Liu** is currently a PhD candidate in the Department of Computer Science at University of Texas at Dallas. She received her Master degree in Computer Science from university of Texas at Dallas in 2003. She received her Bachelor degree in computer science at Northern Jiaotong University, Beijing, China. Her research interests include privacy preserving data mining, anomaly detection, and security issues in data and data application.



**Dr. Murat Kantarcioglu** is currently an assistant professor of computer science at University of Texas at Dallas. He had a Ph.D. degree from Purdue University in 2005. He received his master's in Computer Science from Purdue University in 2002 and his bachelor degree in computer engineering from METU, Ankara, Turkey in 2000. During his graduate years, he worked as a summer intern at IBM Almaden Research Center and at NEC Labs.

His research interests lie at the intersection of Privacy, Security, Data Mining and Databases: Security and Privacy issues raised by data mining; Distributed Data Mining techniques; Security issues in Databases; Applied Cryptography and Secure Multi-Party Computation techniques; Use of data mining for intrusion and fraud detection.



**Prof. Bhavani Thuraisingham** joined The University of Texas at Dallas (UTD) in October 2004 as a Professor of Computer Science and Director of the Cyber Security Research Center in the Erik Jonsson School of Engineering and Computer Science. She is an elected Fellow of three professional organizations: the IEEE (Institute for Electrical and Electronics Engineers), the AAAS (American Association for the Advancement of Science) and the BCS (British Computer Society) for her work in data security. She received the IEEE Computer Society’s prestigious 1997 Technical Achievement Award for “outstanding and innovative contributions to secure data management.” Prior to joining UTD, she worked for the MITRE Corporation for 16 years which included an IPA (Intergovernmental Personnel Act) at the National Science Foundation as Program Director for Data and Applications Security. Her work in information security and information management has resulted in over 80 journal articles, over 200 refereed conference papers three US patents. She is the author of eight books in data management, data mining and data security.