Privacy-Preserving Sharing of Horizontally-Distributed Private Data for Constructing Accurate Classifiers

Vincent Y. F. Tan
Department of Electrical Engineering and Computer Science (EECS)
Massachusetts Institute of Technology (MIT)
Cambridge, MA 02139
vtan@mit.edu

See-Kiong Ng
Data Mining Department
Institute for Infocomm Research (I²R)
Singapore 119613
sknq@i2r.a-star.edu.sg

ABSTRACT

Data mining tasks such as supervised classification can often benefit from a large training dataset. However, in many application domains, privacy concerns can hinder the construction of an accurate classifier by combining datasets from multiple sites. In this work, we propose a novel privacy-preserving distributed data sanitization algorithm that randomizes the private data at each site independently before the data is pooled to form a classifier at a centralized site. Distance-preserving perturbation approaches have been proposed by other researchers but we show that they can be susceptible to security risks. To enhance security, we require a unique non-distance-preserving approach. We use Kernel Density Estimation (KDE) Resampling, where samples are drawn independently from a distribution that is approximately equal to the original data's distribution. KDE Resampling provides *consistent* density estimates with randomized samples that are *asymptotically independent* of the original samples. This ensures high accuracy, especially when a large number of samples is available, with low privacy loss. We evaluated our approach on five standard datasets in a distributed setting using three different classifiers. The classification errors only deteriorated by 3% (in the worst case) when we used the randomized data instead of the original private data. With a large number of samples, KDE Resampling effectively preserves privacy (due to the asymptotic independence property) and also maintains the necessary data integrity for constructing accurate classifiers (due to consistency).

1. INTRODUCTION

Consider the following scenario: A group of hospitals are seeking to construct an accurate *global* classifier to predict new patients' susceptibility to illnesses. It would be useful for these hospitals to pool their data, since data mining tasks such as supervised classification can often benefit from a large training dataset. However, by law, the hospitals cannot release private/sensitive patient data (e.g. blood pressure, heart rate, EKG signal, X-ray images). Instead, some form of sanitized data has to be provided to a centralized server for training and classification purposes. It is thus imperative to discover means to protect private information, while at the same time, be able to perform data mining tasks with a masked version of the raw data. Can privacy and accuracy co-exist?

In fact, in many application domains, privacy concerns hinder the combining of datasets generated from multiple sources despite the growing need to share sensitive data. For example, military organizations may now need to share sensitive security information for anti-terrorist operations, financial institutions may need to share private customer data for anti-money laundering operations, and so on. In all these applications, the setting is a Distributed Data Mining (DDM) scenario [20] in which the private data sources are distributed across $L \geq 2$ multiple sites. The L sites each contain private information that should be shared or combined as they are probably inadequate on their own. To protect privacy, the data at each site must undergo randomization locally to give sanitized data for sharing. The sanitized data are pooled as a large training data set to construct an accurate global classifier, as shown in Fig. 1. Note that unlike other previous works [37], in our formulation, there is only a one-way communication to the centralized server required. This further minimizes potential security risks when dealing with large number of sensitive datasets at distributed sites.

In this work, we consider a privacy-preserving distributed data sanitization approach [3] for the purpose of constructing accurate classifiers at the centralized site. Our work is very closely related to privacy-preserving classification [22, 23]. Here we focus on randomizing the data at each site independently before transmitting the data for constructing a global classifier, which is similar to the horizontally partitioned scenario presented in Du et al. [11]. Also, Lindell and Pinkas [22] used Secure Multi-Party (or 2-party) Computation techniques to compute a global decision tree with a (secure) ID3 algorithm. Here, we seek a generic data sanitization approach that can be applied to *any* classification algorithms on numerical data. More recently, Liu et al. [23] and Olivera et al. [27] discussed how random projection-based multiplicative data perturbation can be used for the privacy-preserving DDM scenario. This data perturbation method has several nice properties, including being distance-preserving, which ensures high accuracy in classification and clustering. However, in section 4, we will show that the distance-preserving property can present potential compromises on security of the data. As such, in this work, we will employ a *non-distance-preserving* randomization algorithm for (i) randomizing (sanitizing) the data at the distributed sites (ii) constructing an accurate classifier centrally.

We thus suggest Kernel Density Estimation (KDE) [28,

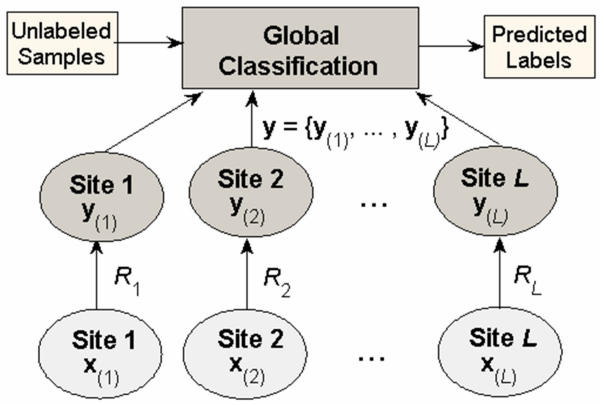

Figure 1: Privacy-Preserving Distributed Data Mining (DDM) Scenario with $L \geq 2$ sites. $\mathbf{x}_{(l)}$ and $\mathbf{y}_{(l)}$ contain the original and randomized data vectors respectively. The $R_l(\cdot)$'s are the (KDE Resampling) nonlinear randomization operators such that $\mathbf{y}_{(l)} = R_l(\mathbf{x}_{(l)})$. The collection $\mathbf{y} = \{\mathbf{y}_{(l)}\}_{l=1}^{L}$ is to be used as the training data for a *global* classifier. Testing data samples are used for cross-validation of the global classifier at the one centralized site. The non-shaded and shaded cells contain private and randomized data respectively.

31, 32] Resampling. KDE Resampling is not new [8] but it has hitherto not been applied to privacy-preserving data mining, to the best of the authors' knowledge. This method possesses some very desirable properties, including *asymptotic independence* and *consistency*, which we will discuss later. Other randomization methods in the literature [1, 2, 7, 23, 27, 37] do not possess these appealing properties. We will exploit these properties to preserve privacy of the distributed data and ensure that the sanitized training samples are still adequate for the construction of accurate classifiers. Note that in our proposed approach, we do not publish the data's distribution/density, since the distribution is fully parameterized by the data records themselves and publishing it would be akin to releasing the private data. Instead, we only transmit the sanitized feature vectors to the centralized site. As shown in Fig. 1, the shaded cells contain the randomized data to be transmitted to the centralized site for the construction of a classifier.

The rest of the paper is structured as follows. In section 2, we discuss in further detail some of the relevant work in sanitization, privacy-preserving DDM and privacy-preserving classification. In section 3, the problem will be formally stated and mathematical notations defined. In section 4, we play the role of a malicious intruder to illustrate the potential security risk in using distance-preserving perturbation methods such as the random projection-based multiplicative data perturbation method [23, 27]. KDE Resampling will then be described in section 5. In the same section, we will also discuss the two elegant properties of the samples produced by KDE Resampling. Following that in section 6, we will define two performance metrics and explain the validity. Section 7 details the evaluation experiments and summarizes the main results. Finally, Section 8 concludes our discussion and suggests directions for future work.

2. RELATED WORK

Atallah et al. [3] first considered data sanitization but the work had mainly been applied to association rule mining. Optimal sanitization is NP-hard [3]. We consider classification in this work and a particular randomization method that is computationally tractable.

The addition of randomly generated Independent and Identically Distributed (IID) noise to the original data was then proposed [1, 2] for masking the private data. The authors reconstructed the probability density function (PDF) of the data for distribution-based data mining. In addition, they constructed decision-trees based on the noisy data and found that the classification results were similar to that using the original data. Muralidhar et al. [25] comprehensively examined the statistical properties of noise addition.

However, such noise addition has since been shown to be insecure [17, 19] and other methods have been proposed. Chen et al. [7] proposed using a rotation-based perturbation technique that ensures low accuracy loss for most classifiers. This perturbation technique was further extended in two papers [23, 27] where the authors used a random projection-based multiplicative data perturbation method to perturb the data, while maintaining its utility. These two papers described a randomization method that is distance-preserving. However, it was shown by Caetano [5] that there is data disclosure vulnerability in adopting these distance-preserving approaches. We will further augment Caetano's argument in section 4 by showing that there can be other security risks with distance-preserving approaches. Thus, we will adopt a non-distance-preserving randomization scheme in this paper.

In Zhang et al. [37], an algebraic-based randomization approach was suggested but it involves multiple communication from the server to the sites. This makes it infeasible for extremely large datasets and in scenarios where the communication channels may not be robust (e.g. military scenarios). In our formulation, there is only a one-way communication to the centralized server (Fig. 1). We also do not assume an underlying probability distribution that is parameterized, in contrast to Liew et al. [21]. In addition, we generalize Liew et al. [21] to multiple dependent confidential attributes by using multivariate densities.

Non-randomization approaches have been suggested as well. In Sweeney's papers [30, 34], k-anonymization was proposed to generalize databases for preserving privacy. Du et al. [11] approaches the privacy-preserving classification problem from yet another perspective. Using Secure Multi-Party Computation (SMC) techniques [4, 33, 36], parties can collaborate to deduce a global classifier or regression function or just a general function, like the sum. We will not deal with SMC techniques in this paper as SMC is not as efficient as randomization approaches [29]. However, the obtained results are more accurate than sanitization methods. SMC solutions [4] send and receive input from each of the participating sites thus it is obvious that this method will incur higher communication cost than randomization. For SMC techniques to be collusion-resistant, significant communication is required between the many sites, which make this technique non-practical. Moreover, in SMC the number of participating sites are typically small, which is often not the case in the distributed mining context where number of sites could be few hundreds to several thousands (e.g. surveying, consumer browsing patterns etc.). For a detailed

statistical analysis of computation overhead of SMC, the reader is referred to Subramaniam et al. [33].

Yet another method in the literature concerns distributed clustering (unsupervised classification) in which the authors chose local models before combining them to give a global model via optimization of information theoretic quantities [24]. We focus on supervised classification here, but our method can be extended for clustering applications.

As mentioned, we will be adopting a technique known as KDE Resampling [8]. This method has many appealing properties, including *asymptotic independence* and *consistency*, which will be fully explained in section 5. Besides these two appealing properties, Indyk and Woodruff [18] also demonstrated that sampling achieves perfect privacy in 2-party polylog-communication $\mathcal{L}_2$ distance approximations. Motivated by promising nature of sampling, we explore its properties when applied to a distributed scenario and the subsequent construction of classifiers.

In terms of evaluation metrics, privacy has typically been measured using *mutual information* [1] as well as *privacy breaches* [13]. Mutual information is an average measure of disclosure while privacy breaches examine the worst-case scenario. Because of our Distributed Data Mining (DDM) setting, we will measure privacy in this paper using a new metric – the Distributed Aggregate Privacy Loss, $\mathcal{DAPL}$, which is related to the mutual information. Our measure is advantageous because it explicitly takes into account the *distributed* nature of the data mining scenario. Moreover, the privacy breach measure was primarily used in the context of association rule mining [14] while in this paper, we are concerned with supervised classification using the sanitized data from the L independent data sites.

3. PROBLEM DEFINITION AND NOTATION

We represent the storage of private information in the form of d-dimensional real row vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$, where N is the number of individuals (subjects) and d is the number of attributes. These row vectors can be vertically concatenated into a $N \times d$ matrix $\mathbf{x}$ such that

$$\mathbf{x} \stackrel{\triangle}{=} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} x_{11} & \ldots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{N1} & \ldots & x_{Nd} \end{bmatrix}. \quad (1)$$

These N individuals are associated with N targets (class labels) $t_1, \ldots t_N$. The class labels are typically not regarded as sensitive/private data [2, 37] and thus they do not have to be randomized.

We then assume that there are L (for $2 \leq L \leq N$) distributed data sites (private) and 1 centralized (untrusted) server (Fig. 1), where the sanitized data are sent to for constructing an accurate classifier using the combined training data. Each data site possesses the private information of N_l individuals, with $\sum_{l=1}^{L} N_l = N$. As in Fig. 1, we use the notation $\mathbf{x}_{(l)}$ for the $N_l \times d$ matrix that contains the N_l data vectors at site l. Thus,

$$\mathbf{x}_{(l)} \stackrel{\triangle}{=} \begin{bmatrix} \mathbf{x}_{(l,1)}^T & \ldots & \mathbf{x}_{(l,N_l)}^T \end{bmatrix}^T, \quad 1 \leq l \leq L, \quad (2)$$

where $\mathbf{x}_{(l,j)}$ for $1 \leq j \leq N_l$ is a sample vector at site l. Thus, $\mathbf{x}$ can alternatively be written as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{(1)}^T & \ldots & \mathbf{x}_{(L)}^T \end{bmatrix}^T. \quad (3)$$

Furthermore, we assume that the row vectors in $\mathbf{x}_{(l)}$ are drawn from IID random vectors with PDF $f_{\mathbf{x}_{(l)}}\left(\mathbf{x}_{(l)}\right)$. We seek to find a randomization scheme for site l, R_l such that

$$\mathbf{y}_{(l)} = R_l(\mathbf{x}_{(l)}), \qquad 1 \leq l \leq L. \quad (4)$$

and $R_l : \mathbb{R}^{N_l \times d} \rightarrow \mathbb{R}^{M_l \times d}$ is the nonlinear randomization operator that maps N_l row vectors in $\mathbf{x}_{(l)}$ to M_l row vectors in $\mathbf{y}_{(l)}$. $\mathbf{y} = \{\mathbf{y}_{(l)}\}_{l=1}^{L}$ is then sent to the centralized server, along with the N associated with targets $t_1, \ldots, t_N$, where classification can then be done using randomized data as training samples[1]. The centralized server will use the pooled randomized/sanitized data as training samples to build a classifier. We will show that the classification results using these randomized data as training samples are compatible to the classification results using the original private data as training samples. Before that, let us first examine why distance-preserving approaches may be vulnerable to attacks by malicious intruders.

4. RISK OF DISTANCE-PRESERVING RANDOMIZATION

In this section, we play the role of a malicious attacker and attempt to deduce information such as the bounds on private data sanitized with a distance-preserving perturbation method such as the random projection-based multiplication method [7, 23, 27]. Caetano [5] had showed previously that the randomized data can be vulnerable to disclosure. We will further augment his argument with two lemmas here.

LEMMA 4.1. *Assume a Distributed Data Mining (DDM) scenario with $L = 2$ sites which contain private data matrices $\mathbf{x}_{(1)}$ and $\mathbf{x}_{(2)}$ respectively. Upon randomization using the random projection-based multiplicative data perturbation method[2], we get $\mathbf{y}_{(1)} = \mathbf{R}\mathbf{x}_{(1)}$ and $\mathbf{y}_{(2)} = \mathbf{R}\mathbf{x}_{(2)}$ respectively. Let the matrix $\mathbf{x}_{(1)}$ have the structure as follows:*

$$\mathbf{x}_{(1)} \stackrel{\triangle}{=} \begin{bmatrix} \tilde{\mathbf{x}}_{(1,1)} & \ldots & \tilde{\mathbf{x}}_{(1,d)} \end{bmatrix} \quad (5)$$

and its columns $\tilde{\mathbf{x}}_{(1,i_1)}$ to be defined as

$$\tilde{\mathbf{x}}_{(1,i_1)} = \begin{bmatrix} \tilde{x}_{(1,i_1,1)} & \ldots & \tilde{x}_{(1,i_1,N_1)} \end{bmatrix}^T, \quad 1 \leq i_1 \leq d \quad (6)$$

Let the other matrices $\mathbf{x}_{(2)}$, $\mathbf{y}_{(1)}$ and $\mathbf{y}_{(2)}$ have similar structures. Further, suppose we have $\widehat{\|\tilde{\mathbf{x}}_{(2,i_2)}\|}$, an estimate of the norm[3] of the $i_2{}^{th}$ column of $\mathbf{x}_{(2)}$ for any $1 \leq i_2 \leq d$, then

$$\|\tilde{\mathbf{x}}_{(1,i_1)}\| \geq \gamma_1, \quad (7)$$

for all $1 \leq i_1 \leq d$, where $\gamma_1 > 0$ is a constant.

All proofs can be found in the Appendix. Lemma 4.1, gives us a lower bound for the norm of *all* the columns of the matrix $\mathbf{x}_{(1)}$, given an estimate of *just one* column of the matrix $\mathbf{x}_{(2)}$. Clearly, there is an obvious security risk, especially if the private values are susceptible to being leaked. Lemma 4.2 builds on this to infer a lower bound on any private data value given other data values.

[1]Note that random vectors are denoted in boldface upper case and the realization is denoted is boldface lower case. For e.g. , $\mathbf{X}_{(l)}$ is a random vector and its realization is $\mathbf{x}_{(l)}$.
[2]In [23], $\mathbf{R} \in \mathbb{R}^{K \times N}$, a random matrix was used to perturb the data via a linear transformation to a lower-dimensional subspace i.e. $K < N$.
[3]Any valid l_p (for $p \geq 1$) norm can be used.

LEMMA 4.2. *Assume exactly the same DDM scenario as in Lemma 4.1 and that we have estimates for all the elements of $\tilde{\mathbf{x}}_{(1,i_1)}$ except the q^{th} element $\tilde{x}_{(1,i_1,q)}$ i.e. we are given the set*

$$\mathcal{A}_{i_1,\backslash q} = \{\tilde{x}_{(1,i_1,k)} | \tilde{x}_{(1,i_1,k)} \in \tilde{\mathbf{x}}_{(1,i_1)}, k \neq q\}. \quad (8)$$

Then,

$$\left|\tilde{x}_{(1,i_1,q)}\right| \geq \gamma_2, \quad (9)$$

for all $1 \leq q \leq N_1$, where $\gamma_2 > 0$ is a constant.

Lemma 4.2 shows that if a malicious attacker were to obtain estimates of data values except the q^{th} element for the data vectors in any of the d dimensions, he or she will be able to infer lower bounds on the private data value he does not possess i.e. $|\tilde{x}_{(1,i_1,q)}|$. This is a potential security breach. Intuitively, there is such a breach because along with the preservation of distances, the 'ordering' of the samples is also preserved. This reasoning (and lemmas) can be extended to the case where $L > 2$. Together with Caetano's argument [5], there is clearly a need for a new randomization method for privacy-preserving classification that is not distance-preserving. In light of the limitations of the additive method [1, 2, 25] and the random projection perturbation method [7, 23, 27], in this work, we will use KDE Resampling, which is a non-distance-preserving randomization algorithm, for data sanitization.

5. KDE RESAMPLING FOR DATA SANITIZATION

In this section, we will detail KDE Resampling and discuss some elegant and useful properties of the randomized samples. We will also comment on its computational tractability and compare it to the more inefficient SMC methods [22].

5.1 Resampling from reconstructed PDF

For each of the L data sites (refer to Fig. 1), we will generate M_l independent vectors in $\mathbf{y}_{(l)}$ with approximately the same density as the original N_l vectors in $\mathbf{x}_{(l)}$. M_l and N_l do not necessarily have to be equal. The algorithm takes place in two steps. Firstly, we will approximate the PDF of the vector in $\mathbf{x}_{(l)}$ using Parzen-Windows Estimation [28] also known as KDE [10, 31, 32]. Then we will sample M_l vectors from this PDF, which we denote $\mathbf{y}_{(l)}$.

5.1.1 Kernel Density Estimation

As mentioned, for data site l, we will construct the multivariate PDF using the N_l vectors in the $N_l \times d$ matrix $\mathbf{x}_{(l)}$, which we denote $\mathbf{x}_{(l,1)}, \ldots, \mathbf{x}_{(l,N_l)}$. This is given by

$$\hat{f}_{\mathbf{x}_{(l)}}\left(\mathbf{x}_{(l)}; \mathbf{x}_{(l,1)}, \ldots, \mathbf{x}_{(l,N_l)}\right) = \frac{1}{N_l} \sum_{j=1}^{N_l} K\left(\mathbf{x}_{(l)} - \mathbf{x}_{(l,j)}; \mathbf{h}_l\right), \quad (10)$$

where $K\left(\mathbf{x}_{(l)} - \mathbf{x}_{(l,j)}; \mathbf{h}_l\right)$ is the Epanechnikov[4] (a truncated quadratic) kernel parameterized by $\mathbf{h}_l$, the vector of bandwidths. In one dimension, K is given by

$$K_1(x; h) = c\, h^{-1} \left(1 - \left(\frac{x}{h}\right)^2\right) \mathbb{I}\{|x| \leq h\}, \quad (11)$$

where c is the normalizing constant. The multivariate version of the Epanechnikov kernel is a straightforward generalization by taking products of the univariate kernel in

[4]The Epanechnikov kernel is optimal in the l_2 sense [8].

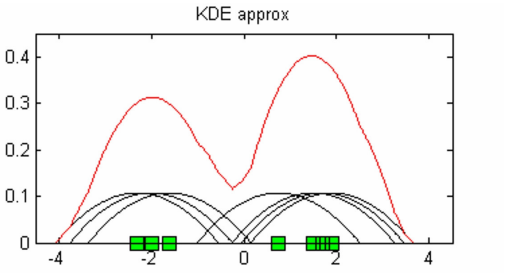

Figure 2: Illustration of KDE approximation for estimation of the multimodal PDF. The *boxes* show the $N = 7$ independent realizations of the multimodal random variable. The individual Epanechnikov kernels ($h = 1.75$) are centered at the realizations. Their *sum*, as detailed in Eq (10), is the Kernel Density Estimate (KDE), which is the sum of the Epanechnikov kernels.

Eq (11). $K(\cdot; \mathbf{h}_l)$, a scalar kernel function, has to satisfy the following properties for Eq (10) to be a valid PDF [15].

$$K(\mathbf{x}; \mathbf{h}_l) \geq 0, \ \forall \mathbf{x} \in \mathbb{R}^d, \quad \int_{\mathbb{R}^d} K(\boldsymbol{\xi}; \mathbf{h}_l)\, d\boldsymbol{\xi} = 1. \quad (12)$$

Example An illustration of how the univariate KDE works for $N = 7$ is shown in Figure 2. The kernels are centered on the realizations of the multi-modal random variable and the sum is an approximation to the true PDF. Notice that, consistent with intuition, more probability mass is placed in areas where there are more realizations of the random variable.

The selection of the bandwidth vector $\mathbf{h}_l \in \mathbb{R}^d$ is a very important consideration in any KDE and will be discussed in section 5.2.1. For optimal performance and accuracy of the KDE, $\mathbf{h}_l$ is to be a function of the number of samples N_l. We note that the Kernel Density Estimate in Eq (10) is a function of $\mathbf{x}_{(l)}$ and it is parameterized by the realizations of IID random vectors $\mathbf{x}_{(l,1)}, \ldots, \mathbf{x}_{(l,N_l)}$ present at site l. Thus, the distribution cannot be published. Instead, we will transmit the M_l randomized data vectors from site l to the centralized site for the construction of a classifier.

Remark For the sake of convenience, we chose M_l and N_l to be equal. However, in practice, they do not have to be equal. In fact, one can sample fewer data vectors than N_l, for example to choose $M_l = N_l/2$. From our experiments, the classification results do not change significantly when $M_l = N_l/2$. We refer the reader to Devroye's book [9, Chapter 14] for a more rigorous treatment on the selection of M_l.

We will subsequently abbreviate the estimate of the true PDF by $\hat{f}_l \triangleq \hat{f}_{\mathbf{x}_{(l)}}\left(\mathbf{x}_{(l)}; \mathbf{x}_{(l,1)}, \ldots, \mathbf{x}_{(l,N_l)}\right)$ and the true PDF by $f_l \triangleq f_{\mathbf{x}_{(l)}}\left(\mathbf{x}_{(l)}\right)$.

5.1.2 Resampling

Equipped with the non-parametric estimate of the true PDF $\hat{f}_l$, we will then sample from this PDF to obtain M_l independent samples $\mathbf{y}_{(l,1)}, \ldots, \mathbf{y}_{(l,M_l)}$. Noting that the random vector $\mathbf{X}_{(l)} = (1/N_l) \sum_{j=1}^{N_l} \mathbf{X}_{(l,j)}$ is a mixture density – it does not have to be constructed explicitly before random

samples are taken. Instead we will sample for a random integer r from 1 to N_l. Following that we will sample a random vector from the r^{th} kernel $K\left(\mathbf{x}_{(l)} - \mathbf{x}_{(l,r)}; \mathbf{h}_l\right)$. The resampling algorithm is summarized in Algorithm 1.

KDE Resampling Algorithm
Data : $\mathbf{x}_{(l,1)}, \ldots, \mathbf{x}_{(l,N_l)}$ for all $1 \leq l \leq L$
Result: $\mathbf{y}_{(l,1)}, \ldots, \mathbf{y}_{(l,M_l)}$ for all $1 \leq l \leq L$
for $l \leftarrow$ *1 to* L **do**
  for $i \leftarrow$ *1 to* d **do**
    $\hat{\sigma}_{l,i}$ = Standard deviation in dimension i;
    $h_{l,i}$ = Bandwidth in dimension i (c.f Eq (16));
  endFor
  for $j \leftarrow$ *1 to* M_l **do**
    r = Random integer from 1 to N_l inclusive;
    $\mathbf{y}_{(l,j)}$ = Random sample vector from r^{th} Epanechnikov kernel $K\left(\mathbf{y}_{(l,j)} - \mathbf{x}_{(l,r)}; \mathbf{h}_l\right)$;
  endFor
endFor

Algorithm 1: KDE Resampling

5.2 Discussion

In any privacy-preserving data mining research, the two key questions are: Has privacy been preserved? Can the randomized vectors be used for data mining purposes? In this section, we will state some very important and salient results from [10]. These results show that the randomized samples $\mathbf{y}_{(l)}$ at each of the L sites are *asymptotically independent* of the original samples $\mathbf{x}_{(l)}$ at the respective L sites. Also, the KDE is *consistent*. We will explain why these two properties are desirable in subsequent sections. We will explain that privacy can indeed be preserved while the randomized samples can be employed for data mining.

5.2.1 Asymptotic Independence

Asymptotic independence implies that the randomized samples are independent of the original samples as the number of samples N_l tends to infinity. If the joint density of $\mathbf{X}_{(l)}$ and $\mathbf{Y}_{(l)}$ is denoted as $f_{\mathbf{X}_{(l)},\mathbf{Y}_{(l)}}\left(\mathbf{x}_{(l)}, \mathbf{y}_{(l)}\right)$ and the marginals as $f_{\mathbf{X}_{(l)}}\left(\mathbf{x}_{(l)}\right)$ and $f_{\mathbf{Y}_{(l)}}\left(\mathbf{y}_{(l)}\right)$, then asymptotic independence can be expressed mathematically as

$$\limsup_{N_l \to \infty} |\Delta_{N_l}| = 0, \tag{13}$$

where the difference between the joint and product of the marginals is defined as

$$\Delta_{N_l} \stackrel{\triangle}{=} f_{\mathbf{X}_{(l)},\mathbf{Y}_{(l)}}\left(\mathbf{x}_{(l)}, \mathbf{y}_{(l)}\right) - f_{\mathbf{X}_{(l)}}\left(\mathbf{x}_{(l)}\right) f_{\mathbf{Y}_{(l)}}\left(\mathbf{y}_{(l)}\right), \tag{14}$$

and the supremum in Eq (13) is over all possible realizations of $\mathbf{x}_{(l)}$ and $\mathbf{y}_{(l)}$.

Another important point is that asymptotic independence is dependent on how we select the bandwidth vector $\mathbf{h}_l$ in Eq (10). If $\mathbf{h}_l$, a function of N_l, satisfies

$$h_{l,i} \xrightarrow{\mathcal{P}} 0, \quad \text{and} \quad N_l h_{l,i}^d \xrightarrow{\mathcal{P}} \infty, \tag{15}$$

as $N_l \to \infty$ then *asymptotic independence* will be achieved [10]. Note that $h_{l,i}$ is the i^{th} element of the bandwidth vector $\mathbf{h}_l$.

In our experiments, we are going to use the Scott's 'rule-of-thumb' [31] to select $\mathbf{h}_l$. Thus,

$$h_{l,i} = \left(\frac{4}{d+2}\right)^{1/(d+4)} N_l^{-1/(d+4)}\, \hat{\sigma}_{l,i}, \tag{16}$$

where $\hat{\sigma}_{l,i}$ is the unbiased estimate of the standard deviation in the i^{th} dimension at the l^{th} site. Scott's 'rule-of-thumb' satisfies both the asymptotic conditions and thus, we have asymptotically independent samples. Since the samples are asymptotically independent, probabilistic inference cannot be performed based on the randomized samples $\mathbf{y}_{(l)}$ if N_l is sufficiently large. This is very often the case in practical data mining scenarios, where datasets are extremely large. Privacy will thus be preserved.

Another way to illustrate this is using the privacy loss measure based on mutual information [1]. Indeed, if N_l is sufficiently large (like in most practical data mining applications), the mutual information $I(\mathbf{x}_{(l)}; \mathbf{y}_{(l)})$ will be close to zero (because of asymptotic independence) and thus, the privacy loss $\mathcal{P}\left(\mathbf{x}_{(l)}; \mathbf{y}_{(l)}\right) = 1 - 2^{-I(\mathbf{x}_{(l)};\mathbf{y}_{(l)})}$ will also be low. In section 6, we will define a new privacy metric, $\mathcal{DAPL}$, and argue that the asymptotic independence of the randomized samples will result in low privacy loss when a large number of samples are available. This property ensures that KDE Resampling is especially effective for preserving the privacy of large datasets i.e. large N_l's.

5.2.2 Consistency of KDE

It is well known [10, 32] that the KDE $\hat{f}_l$, as defined in Eq (10) is consistent i.e.

$$\lim_{N_l \to \infty} \mathbb{E}\left[\int \left|\hat{f}_l - f_l\right|\right] = 0, \quad 1 \leq l \leq L, \tag{17}$$

if the asymptotic conditions in Eq (15) are satisfied. This means that as the number of samples at each site N_l becomes large, the KDE $\hat{f}_l(\cdot)$ becomes increasingly accurate. This property is important and useful because we can treat the collection of randomized samples at all the L sites $\{\mathbf{y}_{(l)}\}_{l=1}^{L}$ as the training data for supervised classification purposes since the distribution it is drawn from is consistent.

Remark We note that because of resampling, our randomization algorithm does not suffer from the problems of [7, 23, 27] that were highlighted in section 4 – namely that of being able to derive bounds on private data given other (relevant) private information. This is one of the key advantages of our novel randomization technique as it removes the inherent ordering of the feature vectors by resampling randomly.

5.2.3 Low Computational Complexity

The random vector $\mathbf{X}_{(l)} = (1/N_l) \sum_{j=1}^{N_l} \mathbf{X}_{(l,j)}$ is a mixture density with N_l components. Thus, we do not need to construct the full KDE. This is typically the bottleneck for any algorithm that uses the Kernel Density Estimate (KDE). Thus the randomized vectors can be obtained simply by:

1. First, estimating the kernel bandwidths $h_{l,i}$, $\forall (l,i) \in \{1, \ldots, L\} \times \{1, \ldots, d\}$ using Eq (16).
2. Generating a random (integer) index r from 1 to N_l.
3. Then drawing a random sample vector from the r^{th} multivariate Epanechnikov kernel.

This is detailed more precisely in Algorithm 1. Each step in the algorithm is tractable. There is no multi-way communication between the parties, unlike in SMC [33, 36]. In conclusion, the KDE Resampling algorithm is computationally feasible.

5.2.4 Possible Application to Horizontally or Vertically Partitioned Data

We have presented a randomization algorithm for the purpose of randomizing *horizontally partitioned* data over L sites. The extension to the *vertically partitioned* scenario, where different sites hold different attributes, is not trivial unless attributes are assumed to be independent as in [35]. KDE Resampling requires multiple full data vectors to be most effective and accurate.

6. PERFORMANCE METRICS

For evaluation, the two performance metrics that we will use to quantify privacy and accuracy are the *Distributed Aggregate Privacy Loss* $\mathcal{DAPL}$ and the *Deterioration of Classification* ϕ respectively.

6.1 Distributed Aggregate Privacy Loss $\mathcal{DAPL}$

The privacy loss is a function of *mutual information* [1], which depends on the *degree of independence* between the randomized samples and the original samples. We have decided to design our privacy metric based on mutual information because the task we are handling is supervised classification. In Evfimievski [13], the notion of security breach was raised. In this work, we focus more on privacy loss, which is an average measure of privacy disclosure. Moreover, the privacy measures proposed by the same paper were more applicable to association rule mining [14]. Thus, in this paper, we use $\mathcal{DAPL}$, which is intimately related to mutual information. Mutual information measures the average amount of information disclosed when the randomized data is revealed. Indeed, [8] also mentioned that

> "...for the sake of asymptotic sample independence, it suffices that the expected l_1 distance between $[\hat{f}_l]$ and $[f_l]$ tends to zero with $[N_l]$."

Because expected l_1 distances provide us with the degree of independence, we will define our privacy loss as a weighted average of expected l_1 distances.

Definition The *Distributed Aggregate Privacy Loss* $\mathcal{DAPL}$ is defined as half of the weighted average of the expected l_1 distance between the estimate $\hat{f}_l$ and f_l, over the L sites. That is,

$$\mathcal{DAPL} \triangleq \frac{1}{2}\left(\sum_{l=1}^{L} c_l \, \mathbb{E}\left[\int \left|\hat{f}_l - f_l\right|\right]\right), \tag{18}$$

where $c_l \triangleq N_l/N$ for $1 \leq l \leq L$ is the proportion of samples at site l. Clearly, $0 \leq \mathcal{DAPL} \leq 1$.

The $\mathcal{DAPL}$ is low (≈ 0) when privacy loss is low and vice versa. Finally, we emphasize that our privacy loss metric $\mathcal{DAPL}$ is related to, but not exactly identical to, the privacy loss metric defined in [1]. The difference is in our considering of the distributed scenario here. Furthermore, we also measure the degree of independence using the expected l_1 distance between $\hat{f}_l$ and f_l as opposed to using mutual information. Since the l_1 distance $\to 0$ as $N_l \to \infty$ [32], the asymptotic independence property ensures low $\mathcal{DAPL}$ when N_l is large.

We emphasize that Eq (18) is a reasonable *privacy measure* because independence of the original and randomized data samples is measured in terms of expected l_1 distances between the original density and the KDE [8].

Example For an *intuitive* feel of Eq (18), let us consider a single site with N_l samples. Suppose N_l is large, then an accurate KDE will be constructed. Subsequently, because we sample from a randomly chosen kernel (out of the N_l kernels whose means are the original data vectors), the resulting randomized data vectors will be approximately independent of the original samples. On the other hand, suppose N_l is small, say only two samples, then the resulting randomized data vectors will be strongly dependent on the positions, in $\mathbb{R}^d$, of the two original samples, resulting in less 'randomness' and greater privacy loss. Another relevant paper by Dwork [12] applies the definition of *differential privacy* to the case of distributed computations, much like our paper.

6.2 Deterioration of Classification ϕ

In any supervised classification algorithm, the usual performance metric is the probability of error, which is also known as the classification error and is defined as [16]

$$P(err) \triangleq 1 - \sum_{i=1}^{|\mathcal{C}|} \int_{\Omega_i} p\left(\boldsymbol{\xi}|\mathcal{C}_i\right) P(\mathcal{C}_i) \, d\boldsymbol{\xi}, \tag{19}$$

where $P(\mathcal{C}_i)$ is the prior probability (known *a priori*) of class $\mathcal{C}_i$ and $\int_{\Omega_i} p\left(\boldsymbol{\xi}|\mathcal{C}_i\right) P(\mathcal{C}_i) \, d\boldsymbol{\xi}$ is the conditional probability of correct classification[5] for given the sample is from $\mathcal{C}_i$. $|\mathcal{C}|$ denotes the total number of classes.

Definition The *Deterioration of Classification* ϕ is defined as:

$$\phi \triangleq P_{rand}(err) - P_{ori}(err), \tag{20}$$

where $P_{ori}(err)$ (resp. $P_{rand}(err)$) is the classification error using the original (randomized) samples as training data.

Clearly, the closer ϕ is to zero, the greater the utility of the randomized samples and the higher the accuracy. So, we want ϕ to be as small as possible.

7. SIMULATION RESULTS

In this section, we will detail the classification experiments on five diverse datasets with continuous, numerical values using three different classifiers. We will consider the distributed setting in Fig. 1. We will empirically show that the classification error is invariant to original and randomized data being used as training examples.

7.1 Experimental Setup of Distributed Setting

In all our experiments, we consider a *distributed* scenario (like in Fig. 1) with L sites, where L is taken over a range of integers. Hence, suppose there are N data points (and N is a multiple of L), then each site will contain $N_l = N/L$ points. If N is not a multiple of L, minor adjustments are made. Each of the N_l data points at the L sites are randomized

[5] Also known as a 'hit' in the detection theory literature.

Dataset	#Class	#Dim(d)	#Trg(N)	#Test
Iris	2	4	120	30
SVMGuide1	2	4	3089	4000
Diabetes	2	8	576	192
Breast-Cancer	2	10	512	171
Ionosphere	2	34	263	88

Table 1: Our five datasets from LIBSVM and the UCI Machine Learning Repository.

using the algorithm detailed section 5. The data is then pooled to the centralized site for the construction of various classifiers. The classification accuracy is compared to the baseline – the result when the data is not randomized.

7.2 Datasets

We obtained five numerical datasets from LIBSVM [6] and the UCI Machine Learning Repository [26]. These are summarized in Table 1. These include the common Iris Dataset and the more difficult Pima Indians Diabetes Dataset.

We pre-process the data by normalizing the values in each dimension to the interval $[-1, 1]$ before the randomization and classification processes. As mentioned in the above section, for each dataset, we performed the randomization followed by classification using a different number of sites L. For example, for the Iris dataset[6] (see Figure 3), L was chosen to be from 1 to 4.

For the purpose of validating the classification accuracy, the raw data (except for the SVMGuide1 dataset[7]) was randomly split into a training set (≈75%) and a testing set (≈25%). This is also commonly-known in the literature as random subsampling *4-fold cross-validation*. For consistent results, we averaged the classification errors over 100 independent random seeds.

Finally, the number of vectors we resampled M_l is the same as the number of original vectors N_l at all L sites. However, as argued in section 5.1, M_l does not have to be the same as N_l. For brevity, we only report the case where $M_l = N_l$ in this section. However, an obvious advantage of using a fewer number of samples is reduction in complexity.

7.3 Classification Techniques

We used three standard classification techniques on the combined randomized data from the L sites $\mathbf{y}_{(1)}, \ldots, \mathbf{y}_{(L)}$ and the original data in $\mathbf{x}$. These techniques include:

1. Artificial Neural Networks (ANN) by trained by error backpropagation.
2. k-Nearest Neighbors classifier (kNN) with $k = 11$.
3. Naïve Bayes classifier (NB) with each attribute or dimension (d) assumed to follow a Gaussian distribution.

The details of these classification techniques can be found in any standard pattern classification text. See for instance [16].

[6]For the Iris dataset, we merged the Setosa and the Versicolour classes into one single class so that we have a binary classification problem. Even though all of our analysis can be extended to the multi-class scenario, it seems not too relevant for the questions addressed here.

[7]SVMGuide1 had already been partitioned into training and testing data *a priori* [6] and thus we use the given partitioning to test the classifiers constructed.

Dataset	ANN	kNN	NB
Iris	0.0352	0.0346	0.1230
SVMGuide1	0.0389	0.0328	0.0695
Diabetes	0.2441	0.2611	0.2263
Breast-Cancer	0.0312	0.0214	0.0396
Ionosphere	0.1255	0.1522	0.0000

Table 2: $P_{ori}(err)$ using original data x as training samples for various classification techniques. Refer to Figs. 3 to 7 for $\phi = P_{rand}(err) - P_{ori}(err)$.

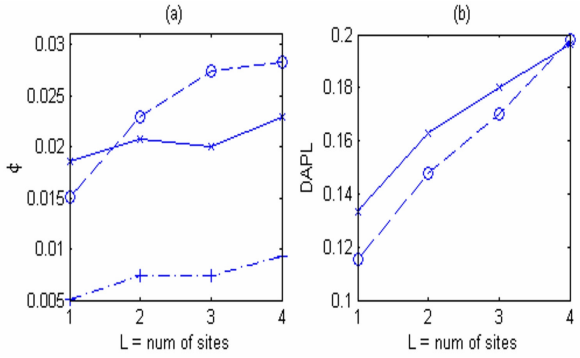

Figure 3: Iris; (a) Deterioration of Classification ϕ (Key: x - ANN, o - kNN, + - NB); (b) Distributed Aggregate Privacy Loss $\mathcal{DAPL}$ (Key: x - Class $\mathcal{C}_1$, o - Class $\mathcal{C}_2$); L was chosen to be from 1 to 4 for the commonly-encountered Iris dataset. Note from (a) that ϕ increases as L is increased, as the number of data records N_l at each site is reduced. However, the Deterioration of Classification ϕ is less than 3% and in this case, the Naïve Bayes classifier (+) performs the best (least deterioration). $\mathcal{DAPL}$ increases as the number of sites L increases because there are fewer samples at each site (cf. example in section 6.1).

First, we used the above three classification techniques to obtain initial classification results on the original data samples. These are shown in Table 2. These results, denoted $P_{ori}(err)$, will be compared to $P_{rand}(err)$, the classification results on the randomized samples in a distributed setting with L sites. The basis for comparison is their difference $\phi \triangleq P_{rand}(err) - P_{ori}(err)$ (cf. section 6.2).

7.4 Results of Distributed Experiments

The results are shown in Figures 3 to 7. Sub-figures (a) show the values of the Deterioration of Classification ϕ, which is defined in Eq (20). The three lines show ϕ for different classification techniques ANN (in crosses – x), kNN (in circles – o) and NB (in plusses – +). We plot the Distributed Aggregate Privacy Loss $\mathcal{DAPL}$ for the two classes in sub-figures (b) (in crosses – x and circles – o). From the plots, we made the following observations.

The classification errors $P_{rand}(err)$ and $P_{ori}(err)$ are close. This can be seen from sub-figures (b) for each of the five datasets, where ϕ deviates from zero by at most only 3%. In general, Naïve Bayes (NB) and Artificial Neural Networks (ANN) perform better as compared to k-Nearest Neighbors (kNN) as the Deterioration of Classification ϕ is closest to zero for all the datasets for NB and ANN.

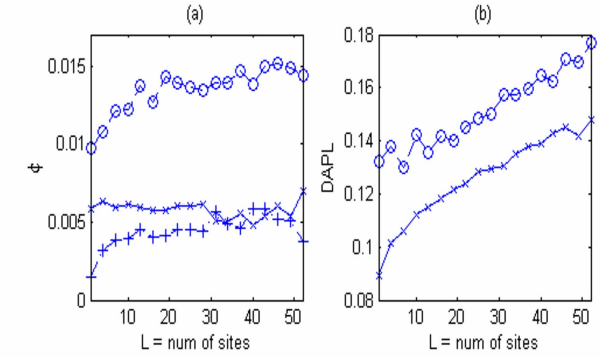

Figure 4: SVMGuide1; (a) Deterioration of Classification ϕ (Key: x - ANN, o - kNN, + - NB); (b) Distributed Aggregate Privacy Loss $\mathcal{DAPL}$ (Key: x - Class $\mathcal{C}_1$, o - Class $\mathcal{C}_2$); The SVMGuide1 dataset describes an astroparticle application. For this dataset, we observe that the ϕ does not increase significantly across L. There is little correlation between the number of sites L and the classification errors $P_{rand}(err)$ or the Deterioration of Classification ϕ. Comparing the results in (a) with Table 2, we observe that the deterioration is not too severe. However, as expected, Privacy Loss $\mathcal{DAPL}$ increases as the number of sites L increases for the same reason as stated in the caption for the Iris dataset.

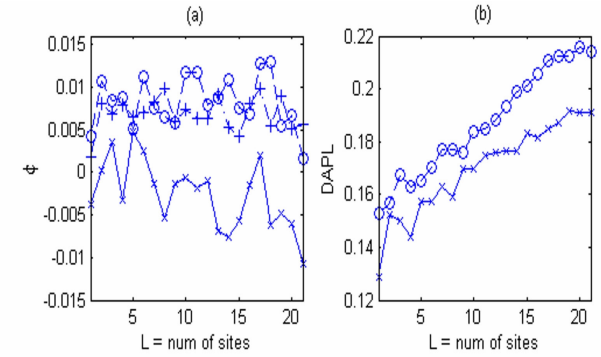

Figure 5: Diabetes; (a) Deterioration of Classification ϕ (Key: x - ANN, o - kNN, + - NB); (b) Distributed Aggregate Privacy Loss $\mathcal{DAPL}$ (Key: x - Class $\mathcal{C}_1$, o - Class $\mathcal{C}_2$); The diabetes dataset exhibits the same characteristics as the SVMGuide1 dataset. However, in this case, it is somewhat surprising to note that in most cases, the ANN classifier constructed based on the randomized samples results in a lower classification error as compared to the one constructed based on the original samples.

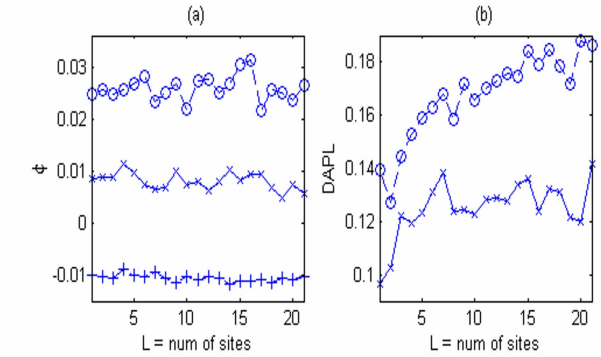

Figure 6: Breast-Cancer; (a) Deterioration of Classification ϕ (Key: x - ANN, o - kNN, + - NB); (b) Distributed Aggregate Privacy Loss $\mathcal{DAPL}$ (Key: x - Class $\mathcal{C}_1$, o - Class $\mathcal{C}_2$); For the Breast-Cancer dataset, the Deterioration of Classification ϕ stays fairly constant across all L. Indeed, the NB classifier (+) constructed based on the randomized samples is better (improves by 1%) than the classifier constructed using the original samples. Again, we observe that, as expected, Privacy Loss $\mathcal{DAPL}$ increases as L increases because the number of samples at each site N_l decreases.

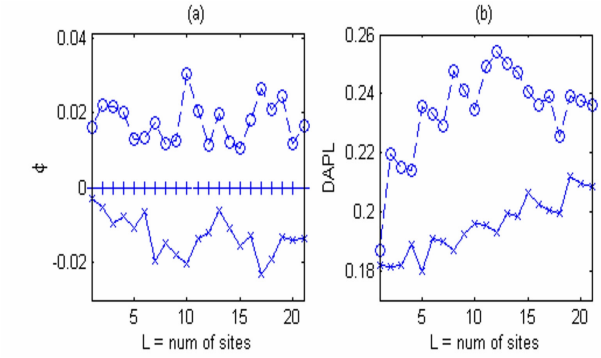

Figure 7: Ionosphere; (a) Deterioration of Classification ϕ (Key: x - ANN, o - kNN, + - NB); (b) Distributed Aggregate Privacy Loss $\mathcal{DAPL}$ (Key: x - Class $\mathcal{C}_1$, o - Class $\mathcal{C}_2$); In this dataset, we observe that the ANN classifier constructed based on the pooled randomized data samples from L sites results in a smaller classification error. The kNN classification technique does results in a worse classification error but, in the worst case, the performance does not deteriorate by more than 3%. Compared with the baseline in Table 2, we see that this deterioration is not too severe.

Except for the Iris dataset, there is no correlation between the number of sites L and the classification errors $P_{rand}(err)$ or the Deterioration of Classification ϕ. Consequently, the randomized data is still amenable to data mining tasks irregardless of L. This is because of the consistency of the KDE as discussed in section 5.2.2.

Finally, from sub-figures (b), we observe that there is a general increasing trend for the $\mathcal{DAPL}$. This is because as the number of sites L increases, the number of individuals at each site N_l decreases. Consequently, the expected l_1 difference between the the original and reconstructed PDFs increases and the privacy loss $\mathcal{DAPL}$ also increases. Thus, with the use of KDE Resampling, there is a *compromise* between L and $\mathcal{DAPL}$. The $\mathcal{DAPL}$ can be further reduced by improving the simple sampling algorithm suggested in Algorithm 1. This can be done by improving selection of the optimal of $\mathbf{h}_l$ [31, 32] by possibly optimizing over non-diagonal bandwidth matrices, which enhances generality but increases complexity.

8. CONCLUSIONS

In this paper, we have suggested a novel method for data *sanitization* for the purpose of sharing private data at distributed data sites for constructing a classifier. In our setup, we are provided with N_l data records at each site and we apply the randomization algorithm at each site independent of other sites. Then the data is pooled together for classification at the centralized site. As mentioned in the introduction, this problem has ramifications in a variety of settings, including the sharing of patients' private records and for collaborations across military or financial organizations for various security operations.

We employed Kernel Density Estimation (KDE) Resampling to sample for new, representative data vectors given the original private data. Our experiments on five datasets conducted in a distributed data setting illustrate that resampling provides sanitized/randomized data samples that can be adequately employed for a particular data mining task – supervised classification. In summary, our data sanitization algorithm has the following advantages over some existing approaches for privacy-preserving classification.

KDE Resampling provides samples that are *asymptotically independent* and the KDE is *consistent*. We have explained, that the former ensures low privacy loss, while the latter preserves the data's integrity and hence, its utility.

We have shown that the classification errors using the *randomized data* collated from the distributed sites as training samples differs from that using the *original data* as training samples by less than 3% for all the datasets. We have also shown that various data mining algorithms can be applied on the randomized training data.

In contrast to random projection-based multiplicative data perturbation methods [7, 23, 27], a malicious intruder cannot establish bounds on the private data using KDE Resampling. In fact, Caetano [5] also argued that random projection-based randomization may be susceptible to disclosure. KDE Resampling thus ensures greater security as it involves an element of random *swapping*, which enhances privacy by losing the ordering of the feature vectors.

In contrast to [37], our framework as shown in Figure 1 does not involve multi-way communication from the centralized server to the individual sites and vice versa. Since our technique involves only a *one-way* communication from the sites to the server, it is feasible for large datasets. Besides, one-way communication reduces the risk of inadvertent disclosure of private data.

Although SMC techniques may provide better privacy protection and accuracy as compared to randomization methods, they suffer from inefficiency [22, 29, 33]. Our algorithm generates samples in an efficient fashion, because each step of the algorithm is tractable and there is no need for multi-way communication.

Finally, we hope to re-examine the issue of the privacy metric. Since $\mathcal{DAPL}$ only looks at the l_1 distances between the two distributions the point-wise distance can be rather significant. Thus, the privacy of any individual may be compromised, without the knowledge of whether the data vector happens to belong to the set with unusually high distance between the two distributions. Though the probability of this event may be low, it is precisely the privacy of outliers that we ought to protect. Another point worth noting is the following – our assumption that the data records is generated from IID random variables is may not be entirely realistic in some practical applications. We hope to relax this assumption in our future research.

9. ACKNOWLEDGEMENTS

Vincent Tan is supported by A*STAR, Singapore. This work was performed at the Institute for Infocomm Research (I^2R). This paper also has benefited from discussions with Dr. Mafruzzaman Ashrafi from I^2R.

10. REFERENCES

[1] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proc. of Symposium on Principles of Database Systems*, pages 247–255, 2001.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of ACM SIGMOD Conf. on Management of Data*, pages 439–450, 2000.

[3] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios. Disclosure limitation of sensitive rules. In *Proc. of Knowledge and Data Engineering Exchange, 1999 (KDEX '99)*, pages 45–52, 1999.

[4] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for cryptographic fault-tolerant distributed computation. In *Proc. of 20th ACM Symposium on the Theory of Computation (STOC)*, pages 1–10, 1988.

[5] T. Caetano. *Graphical Models and Point Set Matching*. PhD thesis, Universidade Federal do Rio Grande do Sul (UFRGS), 2004.

[6] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines, 2001. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] K. Chen and L. Liu. Privacy preserving data classification with rotation perturbation. In *Proc. of 5th IEEE Int. Conf. on Data Mining (ICDM '05), Houston, TX*, pages 589–592, 2005.

[8] L. Devroye. Sample-based non-uniform random variate generation. In *18th conference on Winter simulation*, 1985.

[9] L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, 1986.

[10] L. Devroye and L. Gyorfi. *Non-parametric Density Estimation. The L1 View*. Wiley, 1955.

[11] W. Du, Y. S. Han, and S. Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proc. of SIAM Int. Conf. on Data Mining (SDM '04)*, 2004.

[12] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our Data, Ourselves: Privacy via Distributed Noise Generation. In *EUROCRYPT*, 2006.

[13] A. Evfimievski. Randomization in privacy preserving data mining. In *ACM SIGKDD Explorations Newsletter*, volume 4, pages 43–48, 2002.

[14] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proc. of 8th ACM SIGKDD Int. Conf. on Knowledge Discovery in Databases and Data Mining*, pages 217–228, 2002.

[15] K. Fukunaga and L. D. Hostetler. The estimation of gradient of a density function with applications to pattern recognition. *IEEE Transactions on Information Theory*, 1975:32–40, 21.

[16] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2000.

[17] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *Proc. of ACM SIGMOD Conf., Baltimore, MD*, pages 37–48, 2005.

[18] P. Indyk and D. Woodruff. Polylogarithmic private approximations and efficient matching. In *Proc. of Theory of Cryptography Conf., NY*, 2006.

[19] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Proc. of 3rd IEEE Int. Conf. on Data Mining, Washington, DC, USA,*, pages 99–106, 2003.

[20] H. Kargupta, B. Park, D. Hershbereger, and E. Johnson. Collective data mining: A new perspective toward distributed data mining. *Advances in distributed data mining*, pages 133–184, 1999.

[21] C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM Trans. Database Systems (TODS)*, 10:395–411, 1985.

[22] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Proc. of Advances in Cryptology (CRYPTO '00) Springer-Verlag*, pages 36–53, 2000.

[23] K. Liu, H. Kargupta, and J. Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 18:92–106, 2006.

[24] S. Merugu and J. Ghosh. A privacy-sensitive approach to distributed clustering. *Special issue: Advances in pattern recognition*, 26(4):399–410, 2005.

[25] K. Muralidhar, R. Parsa, and R. Sarathy. A general additive data perturbation method for database security. *Management Science*, 19:1399–1415, 1999.

[26] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI Repository of Machine Learning Databases, University of California, Irvine, Dept. of Information and Computer Sciences, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[27] S. R. Oliveira and O. R. Zaiane. A privacy-preserving clustering approach toward secure and effective data analysis for business collaboration. *Computers & Security*, 26(1):81–93, 2007.

[28] E. Parzen. On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[29] B. Pinkas. Cryptographic techniques for privacy preserving data mining. *SIGKDD Explorations*, 4:12–19, 2002.

[30] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. In *Proc. of the IEEE Symposium on Research in Security and Privacy, Oakland, CA*, May 1998.

[31] D. W. Scott. *Multivariate Density Estimation. Theory, Practice and Visualization*. Wiley, 1992.

[32] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[33] H. Subramaniam, R. N. Wright, and Z. Yang. Experimental analysis of privacy-preserving statistics computation. In *Proc. of the Workshop on Secure Data Management (in conjunction with VLDB '04)*, 2004.

[34] L. Sweeney. k-anonymity: A model for protecting privacy. *Int. Journal of Uncertainty Fuzziness Knowledge Based Systems*, 10:557–570, 2002.

[35] J. Vaidya and C. Clifton. Privacy preserving Naïve Bayes classifier for vertically partitioned data. In *Proc. of SDM'04*, pages 330–334, 2004.

[36] A. Yao. How to generate and exchange secrets. In *Proc. 27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.

[37] N. Zhang, S. Wang, and W. Zhao. A new scheme on privacy-preserving data classification. In *Proc. of 11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining*, pages 374–383, 2005.

APPENDIX

A. PROOFS OF LEMMAS 4.1 AND 4.2

PROOF. Since the random projection-based multiplicative data perturbation method is orthogonal on expectation [23], $\mathbb{E}\left[\mathbf{Y}_{(1)}^T\mathbf{Y}_{(2)}\right] = \mathbf{x}_{(1)}^T\mathbf{x}_{(2)}$, the columns are also orthogonal on expectation i.e. $\mathbb{E}\left[\tilde{\mathbf{Y}}_{(1,i_1)}^T\tilde{\mathbf{Y}}_{(2,i_2)}\right] = \tilde{\mathbf{x}}_{(1,i_1)}^T\tilde{\mathbf{x}}_{(2,i_2)}$ for all $1 \leq i_1, i_2 \leq d$. Using the Cauchy-Schwarz Inequality, we have $\tilde{\mathbf{x}}_{(1,i_1)}^T\tilde{\mathbf{x}}_{(2,i_2)} \leq \|\tilde{\mathbf{x}}_{(1,i_1)}\|\|\tilde{\mathbf{x}}_{(2,i_2)}\|$. Since we are also given $\widehat{\|\tilde{\mathbf{x}}_{(2,i_2)}\|}$, we can bound $\|\tilde{\mathbf{x}}_{(2,i_2)}\|$,

$$\|\tilde{\mathbf{x}}_{(1,i_1)}\| \geq \frac{\mathbb{E}\left[\tilde{\mathbf{Y}}_{(1,i_1)}^T\tilde{\mathbf{Y}}_{(2,i_2)}\right]}{\widehat{\|\tilde{\mathbf{x}}_{(2,i_2)}\|}} \triangleq \gamma_1, \tag{21}$$

This yields Eq (7). Lemma 4.2 follows directly from Lemma 4.1 by subtracting elements contained in the set $\mathcal{A}_{i_1,\setminus q}$ as defined in Eqn (8). □