Privacy Preserving Outlier Detection through Random Nonlinear Data Distortion

Kanishka Bhaduri, *Member, IEEE*, Mark D. Stefanski, and Ashok N. Srivastava, *Senior Member, IEEE*

***Abstract*— Consider a scenario in which the data owner has some private/sensitive data and wants a data miner to access it for studying *important* patterns without revealing the sensitive information. Privacy preserving data mining aims to solve this problem by randomly transforming the data prior to its release to data miners. Previous work only considered the case of linear data perturbations — additive, multiplicative or a combination of both for studying the usefulness of the perturbed output. In this paper, we discuss nonlinear data *distortion* using potentially nonlinear random data transformation and show how it can be useful for privacy preserving anomaly detection from sensitive datasets. We develop bounds on the expected accuracy of the nonlinear distortion and also quantify privacy by using standard definitions. The highlight of this approach is to allow a user to control the amount of privacy by varying the degree of nonlinearity. We show how our general transformation can be used for anomaly detection in practice for two specific problem instances: a linear model and a popular nonlinear model using the sigmoid function. We also analyze the proposed nonlinear transformation in full generality and then show that for specific cases it is distance preserving. A main contribution of this paper is the discussion between the invertibility of a transformation and privacy preservation and the application of these techniques to outlier detection. Experiments conducted on real-life datasets demonstrate the effectiveness of the approach.**

I. Introduction

Privacy preservation is a critical need for a variety of data mining applications where there exists a repository of data which needs to be analyzed without the analyst obtaining the data directly. To solve this problem, researchers have developed many techniques to mask or anonymize the data in order to allow for the analysis to occur. In the simplest case, de-identification (or anonymization) of the data is performed, whereby sensitive information is either obfuscated, redacted, or eliminated from the data records, while only transmitting those attributes of the data that are nonsensitive. However, anonymization techniques can be defeated using the fact that idiosyncratic data can lead to unexpected re-identification of data [1][2][3]. Approaches based on anonymization techniques [4] have been employed in the field by Netflix and various government agencies such as HIPAA[1].

Another approach that can be taken is to allow sensitive data to be analyzed where the data is obfuscated through additive or multiplicative noise. These approaches rely on the fact that a given dataset $\mathcal{D}$ can be passed through an operation (or set of operations) defined by a function $\mathcal{T}$. The mapping is often chosen to be linear an affine transformation. The output of the system, $\mathcal{T}(\mathcal{D})$, is then transmitted with the hope that the original data cannot be reconstructed using the image of $\mathcal{T}(\mathcal{D})$ alone. Many researchers have shown that under certain situations these operations can be reverse engineered, thereby revealing the original data without any information about the nature of the operations or any additional information [2][3]. Essentially, each attack strategy attempts to find an inverse mapping $\mathcal{T}^{-1}$ such that, when applied to $\mathcal{T}(\mathcal{D})$, the original data (within a trivial translation or rotation) can be re-identified, *viz.*, $\mathcal{D} \approx \mathcal{T}^{-1}(\mathcal{T}(\mathcal{D}))$.

In this paper we show a third technique for preserving privacy using functions which cannot be inverted. Specifically, we discuss the situation where $\mathcal{T}$ is a nonlinear mapping parameterized by a set of weights θ. We discuss the situation where the distribution of the weights is known and also study situations where the properties of $\mathcal{D}$ can be observed. We show a method to quantify the probability that a mapping $\mathcal{T}$ can be inverted and show a situation where it cannot be inverted. We refer to this method of data obfuscation as *nonlinear distortion*.

We demonstrate our techniques of nonlinear distortion on the problem of anomaly detection, which is prevalent in a variety of application domains where privacy must be preserved. We discuss the application of these techniques to the realm of aviation safety, where data from multiple air carriers must be kept private to the airline to protect proprietary information. In this situation, it is not possible for the data to be disclosed to the public for analysis or anomaly detection. Moreover, anomalies often tend to provide unique characteristics, thereby identifying a specific airline. However, with an appropriate privacy-preserving data mining approach, it may be possible to apply anomaly detection methods to the data after it has been nonlinearly distorted. For this approach to work, the nonlinear distortion method must preserve the important statistical properties of the data. Thus, if the anomaly detection method is based on Euclidean distance, or the inner product distance, those distances must be preserved through the nonlinear distortion. This paper quantifies the degree of distortion injected by the nonlinear transformation and shows how it affects the ability of the algorithms to detect anomalies using Euclidean distance as the measure of an anomaly. In this paper we:

- Present a new technique which we call the data *distortion*

Manuscript received ...; revised ....

Kanishka Bhaduri is with Mission Critical Technologies Inc at NASA Ames Research Center, MS 269-1, Moffett Field CA 94035. Email:Kanishka.Bhaduri-1@nasa.gov. Mark D. Stefanski is with the Electronic and Computer Engineering Department at Ngee Ann Polytechnic, 535, Clementi Road, Singapore 599489. Email: mark.d.stefanski@gmail.com. Ashok N. Srivastava is with the NASA Ames Research Center, MS 269-3, Moffett Field, CA 94035. Email:Ashok.N.Srivastava@nasa.gov. This work was done when the second author was an intern at NASA Ames Research Center.

[1] http://www.hhs.gov/ocr/privacy/index.html

scheme for preserving data privacy. The framework uses non-invertible non-linear functions for mapping the data to a different space. Mathematically, we show that this transformation cannot be reverse engineered and thereby the original data cannot be recovered due to the condition of non-invertibility.

- Analyze the transformation in its full generality and show that, for specific cases, the transformation is distance preserving, thereby proving useful to the data mining algorithm. Our results generalize all of the previous work on perturbation-based data privacy such as in [5][2][3][6].
- Finally, we show how our technique is particularly useful for a specific data mining technique *viz.* anomaly detection.

The rest of this paper is organized as follows. Section II discusses the motivation for this research. Section III presents the related work. Section IV introduces the notations and discusses the formal problem definition followed by the non-linear distortion technique in Section V. Bounds on the quality of the distortion are discussed in Section VI while some special cases of the distortion are presented in Section VII. A discussion of privacy of the technique follows in Section VIII. Section IX demonstrates the performance of the technique on real-world data for a commercial air carrier. Finally the paper is concluded in Section X with future research plans.

II. Motivation and Background

Outlier or anomaly detection [7] refers to the technique of finding patterns from a dataset that are inconsistent or considerably dissimilar from the rest of the dataset. Outlier detection has been studied in the statistics community for a long time [8][9]. Data mining researchers have developed a number of solutions for outlier detection in various domains: fraud detection, network intrusion detection, climate and ocean current change modeling using wireless sensor networks, engineering systems, and more. Since in most of these domains the data is not sensitive, privacy is not an issue for these applications. For a more detailed literature on anomaly detection and its different application areas, interested readers are referred to a recent survey by Chandola *et al.* [10].

The problem that we aim to solve in this paper can be informally stated as follows: consider a number of different airline companies each having their own aircrafts' systems health and flight operation data commonly referred to as a Flight Operational Quality Assurance (FOQA) archive. In order to analyze operational characteristics and safety issues from a large set of data encompassing multiple air carriers, the Distributed National FOQA Archive (DNFA) [11] has been developed jointly by NASA and FAA with collaboration by different air carriers. Figure 1 shows the architecture. Note that the connections between different FOQA archives and a central node use dedicated and secure T1 lines. As shown in Figure 1, when an analyst executes a query about the data, it is disseminated across the FOQA archive of each air carrier. The computations are done locally at each site and an anonymized and de-identified result set of the query is sent back through the secure lines to the central node. In this architecture, there is no way of accessing the raw data for a more in-depth analysis due to its proprietary nature. Also, anonymization may not be an effective method of privacy preservation since it can be broken with sufficient background information [12][13]. We aim to develop a privacy preserving technique which will enable us to detect distance-based outliers from such global datasets while preserving their privacy in a strict sense since outliers often contain uniquely identifiable information linking a data point to a data repository. We assume that the privacy requirements of the normal operating points are less, because most of the airlines have similar operational characteristics.

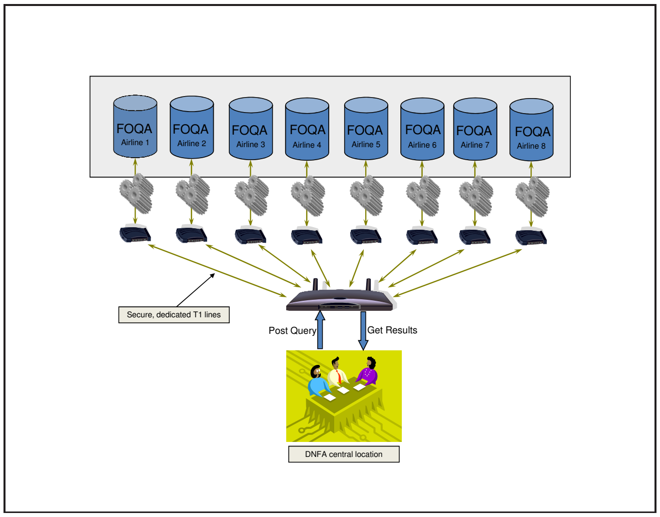


Fig. 1. Distributed National FOQA Archive architecture showing how the analysts can post query and get results for further analysis. Image source: www.faa.gov/library/reports/medical/oamtechreports/2000s/media/200707.pdf

The privacy technique proposed in this paper essentially uses a random non-linear map to transform the input data. The mapping or the function satisfies two properties: (1) for all points in the normal operating region, the mapping *approximately* preserves the distance between those points in the transformed space, and (2) it maps all outliers to a finite set of discrete values. We show that if this transformation is non-invertible, then it is virtually impossible to break this transformation and uncover the original data. As a result of this transformation, most of the outliers will remain such even after transformation. Also note that the privacy of the non-outlier points are also protected since we apply a combination of additive and multiplicative perturbation to these points as done in [5][3][14]. However, as stated before, our main aim is to protect the privacy of the outliers. There are several other places where our technique can be applied such as detecting fraud across multiple financial institutions and finding unusual patterns in medical records.

III. Related Work

The research in privacy preserving data mining spans many areas: data perturbation techniques [15][5], cryptographic (secure multi-party) techniques [16][17][18] and output perturbation techniques [19]. In this paper we only discuss the data perturbation techniques since they are most closely related to this area of research.

Data perturbation-based privacy preserving techniques perturb data elements or attributes directly by additive noise, multiplicative noise or a combination of both. They all rely on the fundamental property that the randomized dataset may not reveal private data while still allowing data analysis to be performed on them. We discuss each of the techniques in more detail in this section.

Given a data set $\mathcal{D}$, Agrawal and Srikant [15] proposed a technique of generating a perturbed dataset $\mathcal{D}^*$ by using additive noise *i.e.* $\mathcal{D}^* = \mathcal{D}+\mathcal{R}$, where the entries of $\mathcal{R}$ are i.i.d. samples from a zero mean unit variance Gaussian distribution. Kargupta *et al.* [20] questioned the use of random additive noise and pointed out that additive noise can be easily filtered out using spectral filtering techniques causing a privacy breach of the data.

Due to the potential drawback of additive perturbations, several types of multiplicative perturbation techniques have been proposed. Kim and Winkler [14] proposed one such perturbation technique which multiplies a random number generated from a truncated Gaussian distribution of mean one and small variance to each data point *i.e.* $\mathcal{D}^* = \mathcal{D} \times \mathcal{R}$, where the matrix multiplication is the Hadamard product, which means that it is carried out element-wise. An appropriate attack strategy would be to estimate the matrix $\mathcal{R}$ given the data. One such attack technique has been discussed by Liu *et al.* [2] which uses a sample of the input and output to derive approximations on the estimate of the matrix $\mathcal{R}$.

A closely related but different technique uses random data projection to preserve privacy. In this technique, the data is projected into a random subspace using either orthogonal matrices (*e.g.* DCT/DFT as done by Mukherjee *et al.* [6]) or pseudo-random matrices (as done by Liu *et al.* [5], Teoh and Yuang [21]). It can be shown that using such transformations, the Euclidean distance among any pairs of tuples is preserved and thus, many distance-based data mining techniques can be applied. Moreover, the privacy of the projection scheme can be quantified using the number of columns of the projection matrix. Figure 2 shows the distribution of the error as a function of the output dimension for simulated data with the hyperbolic tangent $(\tanh)$ nonlinearity. In the graph, the input dataset $\mathcal{D}$ consists of two column vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ each of dimension 50. The output is generated according to $\mathbf{y_1} = f(\mathcal{R}\mathbf{x_1})$ and $\mathbf{y_2} = f(\mathcal{R}\mathbf{x_2})$, where $\mathcal{R}$ is a random projection matrix $(m \times 50)$ with m varying from 5 to 100, and f refers to the $\tanh$ function. In the graph, $\left|\mathbf{x_1}^{\mathrm{T}}\mathbf{x_1} - \mathbf{y_1}^{\mathrm{T}}\mathbf{y_1}\right|$ is plotted in the y-axis for different values of m. As expected, increasing m reduces the error due to projection in a larger subspace. More about this nonlinearity and its role in privacy preservation will be discussed in subsequent sections.

In a more recent study, Chen *et al.* [3] proposed a combination of these techniques: $\mathcal{D}^* = \mathcal{A} + \mathcal{R} \times \mathcal{D} + \mathcal{N}$, where $\mathcal{A}$ is a random translation matrix, $\mathcal{R}$ is a random rotation matrix and $\mathcal{N}$ is a noise matrix. The paper further shows how to break this transformation in practice using a linear regression technique when the attacker knows a set of input-output pairs. However, the success of this attack depends on the variance of the matrices. The paper further defines a privacy measure known as *variance of difference* (*VoD*) which measures the difference of the covariance matrix between each column of $\mathcal{D}^*$ and $\mathcal{D}$. We discuss this in more detail later.

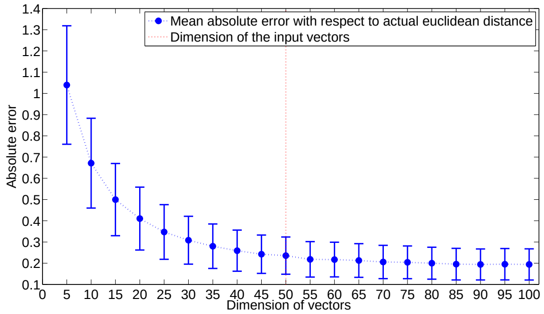


Fig. 2. This graph shows the variation of error in estimating the inner product between two arbitrary vectors vs. the dimension of the output vector. The output is generated by first randomly projecting the input in the subspace shown by points on the x-axis and then transforming it by a hyperbolic tangent $(\tanh)$ function. The dimension of the input vectors are 50 as shown by dotted line. The y-axis refers to the error. The squares to the left of this line refers to dimensionality reduction and to the right refers to dimensionality inflation. Each point in the graph is an average of 100 independent trials.

Data perturbation techniques for categorical attributes have also been proposed by Warner [22] and [23]. Evfimevski *et al.* proposed the γ-amplification model [24] to bound the amount of privacy breach in categorical datasets.

In this section and the next we introduce the notations and discuss in detail about the non-linear data distortion scheme for privacy preserving outlier detection.

IV. Notations and Problem Definition

A. Notations

Let $\mathbf{x} = [x_1 x_2 \dots x_n]^{\mathrm{T}}$ be an n-dimensional input data vector where each $x_i \in \mathbb{R}$. Let $\mathbf{x}^* = [x_1^* x_2^* \dots x_p^*]^{\mathrm{T}}$ be the corresponding output generated according to some transformation $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^p$, where again $x_i^* \in \mathbb{R}$. In this paper we study a very general form of $\mathcal{T}$:

$$\mathbf{x}^* = \mathcal{T}(\mathbf{x}) = \mathrm{B} + \mathrm{Q} \times f\left(\mathrm{A} + \mathrm{W}\mathbf{x}\right) \tag{1}$$

where $f : \mathbb{R}^m \to \mathbb{R}^m$ is a function which[2]

1) acts element-wise on its argument,
2) is continuous over the real line $\mathbb{R}$,
3) bounded on all bounded intervals on $\mathbb{R}$, and
4) $f(x) = O\left(e^{|x|^\alpha}\right)$ as $|x| \to \infty$ where $\alpha \in \mathbb{R}$ is a constant and $\alpha < 2$.

$[\mathrm{B}]_{p\times 1}$, $[\mathrm{Q}]_{p\times m}$, $[\mathrm{A}]_{m\times 1}$, and $[\mathrm{W}]_{m\times n}$ are matrices (with dimensions shown) whose entries b_{ij}, q_{ij}, a_{ij}, and w_{ij} are each independently drawn from normal distributions with mean zero and standard deviations σ_b, σ_q, σ_a, and σ_w respectively *e.g.* $w_{ij} \sim N(0, \sigma_w)$. The normal distribution assumption for generating random matrices is not new and has been proposed by several authors [3][5]. The transformation $\mathcal{T}$ was chosen for three principal reasons: (i) the transformation is flexible in that one can choose f from a large class of functions, (ii) one can set the variances of the Gaussian-distributed matrix entries to

[2] These are sufficient but by no means necessary conditions, which are in place to ensure the existence of the improper integrals that we later derive.

any value, and eliminate the bias matrices B and A by setting $\sigma_b = 0$ and $\sigma_a = 0$, respectively; (iii) and, intuitively, this randomized and potentially non-linear transformation should perturb data better than the simple projection- or rotation-based transformations considered so far in the literature and should thus be less susceptible to attack for wise choices of f and parameter values. Special cases of $\mathcal{T}$ can be instantiated by choosing specific instances of f, two of which we discuss in Section VII. $E(\cdot)$ denotes the mean of a random variable and $\sigma^2(\cdot)$ denotes its variance. The inner product between two vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted by $\mathbf{x} \cdot \mathbf{y}$.

B. Problem Definition

In this paper we analyze the relationship between the input data vectors and their corresponding outputs under the transformation $\mathcal{T}$. While such a relationship can be studied in many different ways, we focus on the *inner product* between the input and the output. Inner product is an important primitive which can be used for many advanced data mining tasks such as distance computation, clustering, classification and more. Specifically, we try to gain insight into the following problem.

Given two vectors $\mathbf{x} = [x_1 x_2 \ldots x_n]^{\mathrm{T}}$ and $\mathbf{y} = [y_1 y_2 \ldots y_n]^{\mathrm{T}}$, let $\mathbf{x}^* = \mathcal{T}(\mathbf{x}) = [x_1^* x_2^* \ldots x_p^*]^{\mathrm{T}}$ and $\mathbf{y}^* = \mathcal{T}(\mathbf{y}) = [y_1^* y_2^* \ldots y_p^*]^{\mathrm{T}}$ be the corresponding output vectors. Since $\mathbf{x}^*$ and $\mathbf{y}^*$ are random transformations of their parent vectors, we analyze the relationship between $\mathbf{x} \cdot \mathbf{y}$ and $\mathbf{x}^* \cdot \mathbf{y}^*$. Our study in this paper focuses on:

1) understanding the **accuracy** of $\mathcal{T}$ in preserving distances *i.e.* studying the properties of $E\left[\mathbf{x}^* \cdot \mathbf{y}^*\right]$, and
2) analyzing the **privacy-preserving** properties of $\mathcal{T}$ *i.e.* under what conditions is $\mathcal{T}^{-1}(\mathcal{T}(D)) \neq D$ in the absence of auxiliary information?

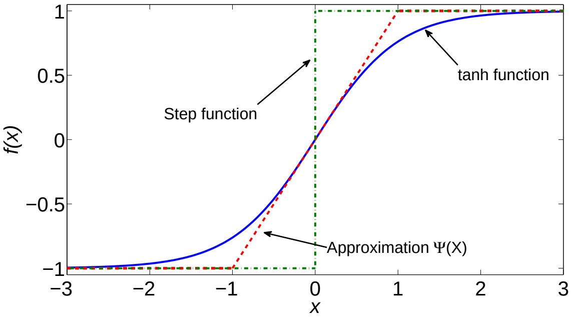


Fig. 3. This figure shows an example nonlinearity, the hyperbolic tangent (tanh) function shown in bold. As the slope of the nonlinearity increases, the function becomes less invertible. In the limit, as the function's slope becomes infinite, it becomes a non-invertible step function (dotted line). $\Psi(x)$ is an approximation to $\tanh(x)$ that we use to bound the expected distortion due to this nonlinearity.

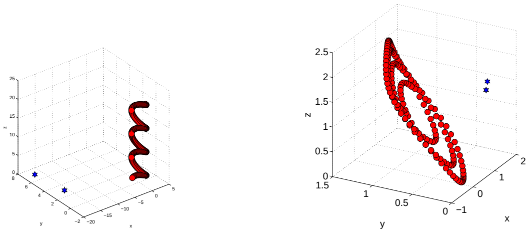


Fig. 4. This synthetic data set is used to show the effect of the nonlinear transformation. The helical coil (left) represents nominal data and the two outlying points represent off-nominal or anomalous data points. The right graph shows the output after non-linear transformation as described in Equation (2) using $f = \tanh$. Notice that the outlying points are far away from the majority of the data, thus validating the distance preservation property of this nonlinear distortion scheme.

C. Overview of Approach

In order to illustrate the idea behind our approach, consider a situation where a single scalar variable x is passed through a nonlinear function $\mathcal{T}$. Figure 3 shows the hyperbolic function as an example nonlinearity. In this figure, the slope is parameterized by a single number θ which sets the slope of the function near the origin. Notice that for moderate values of θ the function is invertible. Thus, a value of x outside of the neighborhood of the origin will be mapped to a number close to -1 or 1 depending on its sign. As the slope becomes steeper, corresponding to a larger value of θ, the invertibility of the function diminishes because the range of the function becomes binary, thus producing a many-to-one mapping. As the function converges to a step function (with infinite slope at the origin), the values of x get mapped directly to 0 or 1 depending on the sign of the variable. In this situation, the function is no longer invertible because given an image of the input, it is impossible to determine the input itself even if the non-invertible function is known.

Figure 4 (left) shows a synthetic data set in which the input space is a helical coil with two outliers. This data set is transformed via the $\tanh$ nonlinear mapping. The output is shown in the right subplot and indicates that under this transformation, the outliers in the input space are still outliers in the output space of the system.

The following sections derive the quantity $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ which is the expected value of the distance between output vectors of the system using Gaussian assumptions about the input distribution. We compute rigorous bounds on this quantity as well as the second moment of the output distribution. These bounds demonstrate that under certain conditions, the nonlinear mapping is distance preserving for all the data points which are close to the origin and highly private for all outliers (since they all get mapped to the same output value). However, as the system becomes more nonlinear, the bounds increase to unity. This reduces the probability of inverting the mapping, and increases the privacy of the overall system, even for points which are non-outliers. The degree to which distances are preserved decreases as a consequence. It is important to note that the example of a single $\tanh$ function is given only as an example. For real-world applications, a full neural network based architecture can be used with multiple weights and nonlinearities, thus providing a more complex nonlinear mapping. Even in this significantly more complex case, however, our derivation of $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ is valid.

V. Nonlinear Data Distortion

In this section we present our data distortion method using a potentially nonlinear transformation. Later we will analyze two special cases of this method: (1) the $f = \tanh$ function that corresponds to the nonlinear function used in neural networks, and (2) f as the identity function. We study the second case in order to demonstrate that our results lead to those obtained by other authors that have studied random projections for privacy preservation.

In the next subsection we introduce the mechanism of this transformation and then show its distance-preserving properties.

A. Mechanism

Let $[\mathrm{D}]_{n \times m}$ be a data set owned by Alice in which there are m instances (columns) each of dimensionality (rows) n. Alice wants to grant Mark (a data miner) access to this dataset. However, she does not want Mark to look at the raw data. So for every vector $\mathbf{x} \in \mathbb{R}^n$, Alice generates a new tuple $\mathbf{x}^* \in \mathbb{R}^p$ according to the following transformation:

$$\mathbf{x}^* = \mathrm{B} + \mathrm{Q} \times f(\mathrm{A} + \mathrm{W}\mathbf{x}) \tag{2}$$

where B, Q, A and W are all mean zero and constant variance Gaussian i.i.d. random matrices as defined in Section IV-A. Figure 4 shows sample input data and the perturbation achieved by the transformation $f = \tanh$.

In the next subsection we discuss how the inner product between two input vectors is related to their transformed counterpart.

B. Derivation of $E[\boldsymbol{x}^* \cdot \boldsymbol{y}^*]$

In this section we show how $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ can be evaluated. Note that,

$$\begin{aligned} E[\mathbf{x}^* \cdot \mathbf{y}^*] &= E[x_1^* y_1^* + x_2^* y_2^* + \ldots + x_p^* y_p^*] \\ &= E\left[x_1^* y_1^*\right] + E\left[x_2^* y_2^*\right] + \ldots + E\left[x_p^* y_p^*\right] \\ &= pE\left[x_i^* y_i^*\right] \end{aligned} \tag{3}$$

where i is arbitrary. The last equality follows from the fact that the entries of each of the matrices are i.i.d. Gaussian variables. Further, letting $\mathbf{w_i} \in \mathbb{R}^n$ denote the i-th row of W, we have

$$x_i^* y_i^* = \left[b_i + \sum_{\ell=1}^{m} q_{i\ell} f(a_\ell + \mathbf{w}_\ell \cdot \mathbf{x})\right] \cdot \left[b_i + \sum_{\ell=1}^{m} q_{i\ell} f(a_\ell + \mathbf{w}_\ell \cdot \mathbf{y})\right]$$

In taking the expected value of the above expression, one need only consider those terms that are not linear in both $q_{i\ell}$ and b_i. All other terms evaluate to zero under the expected value operator due to the independence of the random variables concerned and their property of having a mean of zero. Thus,

$$\begin{aligned} E\left[x_i^* y_i^*\right] &= E\left[b_i^2 + \sum_{\ell=1}^{m} q_{i\ell}^2 f(a_\ell + \mathbf{w}_\ell \cdot \mathbf{x}) f(a_\ell + \mathbf{w}_\ell \cdot \mathbf{y})\right] \\ &= E\left[b_i^2\right] + mE\left[q_{i\ell}^2\right] E\left[f(a_\ell + \mathbf{w}_\ell \cdot \mathbf{x}) f(a_\ell + \mathbf{w}_\ell \cdot \mathbf{y})\right] \\ &= \sigma_b^2 + m\sigma_q^2 E[f(a_i + \mathbf{w_i} \cdot \mathbf{x}) f(a_i + \mathbf{w_i} \cdot \mathbf{y})] \end{aligned} \tag{4}$$

where i and ℓ are interchangeable. So it suffices to find $E[f(a_i + \mathbf{w_i} \cdot \mathbf{x}) f(a_i + \mathbf{w_i} \cdot \mathbf{y})]$ where i is arbitrary. Below we define two vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ which aid in finding the expected value.

Definition 5.1: [***Linear Combination of random variables***] Let $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ be $(n+1)$-dimensional vectors defined as follows:

$$\hat{\mathbf{x}} = [\sigma_w \mathbf{x} \quad \sigma_a]^{\mathrm{T}} = [\sigma_w x_1 \quad \ldots \quad \sigma_w x_n \quad \sigma_a]^{\mathrm{T}} \tag{5}$$

$$\hat{\mathbf{y}} = [\sigma_w \mathbf{y} \quad \sigma_a]^{\mathrm{T}} = [\sigma_w y_1 \quad \ldots \quad \sigma_w y_n \quad \sigma_a]^{\mathrm{T}} \tag{6}$$

where σ_w and σ_a are the variances of W and A respectively and, $\mathbf{x}$ and $\mathbf{y}$ are the n-dimensional inputs.
Now let

$$X = a_i + \mathbf{w_i} \cdot \mathbf{x} \tag{7}$$

$$Y = a_i + \mathbf{w_i} \cdot \mathbf{y} \tag{8}$$

be two random variables. The following lemma shows the distribution of X and Y.

Lemma 5.1: X and Y, as defined above, are distributed as

$$X \sim N(0, \|\hat{\mathbf{x}}\|^2)$$

$$Y \sim N(0, \|\hat{\mathbf{y}}\|^2)$$

Proof: X and Y are linear combinations of normally distributed independent random variables; hence they themselves are Gaussian random vectors. ■

Combining Equations (3), (4), (7), and (8), we can write:

$$E[\mathbf{x}^* \cdot \mathbf{y}^*] = p\left\{\sigma_b^2 + m\sigma_q^2 E\left[f(X)f(Y)\right]\right\} \tag{9}$$

The last equation shows that the expected inner product can be evaluated using the joint probability distribution between X and Y. Further, since X and Y are Gaussian random variables, the joint probability distribution is a bivariate Gaussian distribution $g_{X,Y}(x,y)$:

$$g_{X,Y}(x,y) = \frac{1}{2\pi \|\hat{\mathbf{x}}\| \|\hat{\mathbf{y}}\| \sqrt{1 - \rho_{X,Y}^2}} \tag{10}$$

$$\exp\left(-\frac{1}{2(1-\rho_{X,Y}^2)}\left(\frac{x^2}{\|\hat{\mathbf{x}}\|^2} + \frac{y^2}{\|\hat{\mathbf{y}}\|^2} - \frac{2\rho_{X,Y}xy}{\|\hat{\mathbf{x}}\| \|\hat{\mathbf{y}}\|}\right)\right)$$

where for this form to be valid $\|\hat{\mathbf{x}}\|$ and $\|\hat{\mathbf{y}}\|$ must be nonzero and $\rho_{X,Y}$, the correlation coefficient of X and Y, must not be ± 1. Unless otherwise stated, from now on we will assume that

- $\|\hat{\mathbf{x}}\| > 0$, $\|\hat{\mathbf{y}}\| > 0$, and
- $\rho_{X,Y} \neq \pm 1$

Note that these conditions are equivalent to $|\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}| < \|\hat{\mathbf{x}}\| \|\hat{\mathbf{y}}\|$. We make these assumptions so that $g_{X,Y}$ has a consistent, explicit, bivariate Gaussian expression. When these assumptions are not satisfied $g_{X,Y}$ is degenerate, so these assumptions leave us with the most general form of the problem. $\rho_{X,Y}$ can be defined in terms of $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ as:

$$\rho_{X,Y} = \frac{\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}}{\|\hat{\mathbf{x}}\| \|\hat{\mathbf{y}}\|} \tag{11}$$

Finally, we can write,

$$E[f(X)f(Y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x)f(y)g_{X,Y}(x,y)dxdy$$

Note that $E[f(X)f(Y)]$ can be difficult if not impossible to solve explicitly and in full generality, depending on the choice of f because the anti-derivative might be impossible or extremely difficult to evaluate. However, given f, the above integrals can be approximated numerically [25] for instances of $\mathbf{x}$ and $\mathbf{y}$ in such a way that scales very well computationally with the input dimension, n, which enters into the (trivial) computations of $\|\hat{\mathbf{x}}\|$, $\|\hat{\mathbf{y}}\|$, and $\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}$ alone. Using such an approximation, $E[f(X)f(Y)]_{\text{approx}}$, one can obtain a numerical approximation of $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ (refer to Equation (9)). However, the approximation becomes less accurate the larger p, m, and σ_q are. The conditions we imposed on f in Section IV-A ensure the existence of the improper integrals. Putting it all together, we can write:

$$E[\mathbf{x}^* \cdot \mathbf{y}^*] \quad = \quad p\sigma_b^2 + pm\sigma_q^2 E[f(X)f(Y)] \tag{12}$$

Next, we state some interesting properties of $E[f(X)f(Y)]$.

C. *Properties of* $E[f(X)f(Y)]$

- **Case 1**: if $\hat{\mathbf{x}} \cdot \hat{\mathbf{y}} = 0$:
  - This implies that X and Y are independent (since X and Y are Gaussian vectors). Hence $E[f(X)f(Y)] = E[f(X)]E[f(Y)]$.
- **Case 2**: if f is an odd function and $\hat{\mathbf{x}} \cdot \hat{\mathbf{y}} > 0$ or $\hat{\mathbf{x}} \cdot \hat{\mathbf{y}} < 0$:
  - It can be shown using the expression for $g_{X,Y}(x,y)$ that:
    Lemma 5.2: $\hat{\mathbf{x}} \cdot \hat{\mathbf{y}} > 0 \Rightarrow E[f(X)f(Y)] > 0$
    Lemma 5.3: $\hat{\mathbf{x}} \cdot \hat{\mathbf{y}} < 0 \Rightarrow E[f(X)f(Y)] < 0$
    The proofs follow from the symmetry of the Gaussian distribution.

Since computation $E[f(X)f(Y)]$ is difficult in full generality, in the next section we develop a bound on $E[f(X)f(Y)]$ and analyze its properties.

VI. Bounds on $E[f(X)f(Y)]$

In order to develop a bound on $E[f(X)f(Y)]$, we use the following lemma.

Lemma 6.1: $|E[f(X)f(Y)]| \leq \sqrt{E[f^2(X)]E[f^2(Y)]}$

Proof: For any $\lambda \in \mathbb{R}$,

$$\begin{aligned} 0 &\leq E[(\lambda f(X) - f(Y))^2] \\ &= \lambda^2 E[f(X)^2] - 2\lambda E[f(X)f(Y)] + E[f(Y)^2] \end{aligned}$$

The above is quadratic in λ and because it is always non-negative, it has one root or imaginary roots. Thus, the discriminant

$$(-2E[f(X)f(Y)])^2 - 4E[f(X)^2]E[f(Y)^2] \leq 0$$

which upon rearranging terms and taking the (positive) square root of both sides becomes

$$|E[f(X)f(Y)]| \leq \sqrt{E[f(X)^2]E[f(Y)^2]}$$

■

The following lemma (Lemma 6.2) shows the bound on $E[f(X)f(Y)]$.

Lemma 6.2: Let X, Y, $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ be as defined in the previous sections. It can be shown that,

$$\begin{aligned} |E[f(X)f(Y)]| \quad &\leq \sqrt{\left(\int_{-\infty}^{\infty} f^2(x) \cdot \frac{e^{-x^2/(2\|\hat{\mathbf{x}}\|^2)}}{\sqrt{2\pi}\,\|\hat{\mathbf{x}}\|} dx\right)} \\ &\sqrt{\left(\int_{-\infty}^{\infty} f^2(y) \cdot \frac{e^{-y^2/(2\|\hat{\mathbf{y}}\|^2)}}{\sqrt{2\pi}\,\|\hat{\mathbf{y}}\|} dy\right)} \end{aligned}$$

Proof: This can be easily proved using the definitions of $E[f^2(X)]$, $E[f^2(Y)]$ and Lemma 6.1. ■

A. *Variance Analysis*

In practice, given two input vectors, it is difficult to run the transformation for many independent trials and then take the average inner products of the output vectors. In this section we derive bounds on the variance of the estimated inner product, in order to quantify the error injected for a single run of the transformation.

Lemma 6.3: Let X and Y be two random variables as defined earlier. The variance of the inner product between the output vectors $\mathbf{x}^*$ and $\mathbf{y}^*$ can be written as:

$$\begin{aligned} \sigma^2_{(\mathbf{x}^* \cdot \mathbf{y}^*)} \quad = \quad & 2p\sigma_b^4 + pm\sigma_b^2\sigma_q^2(E[f(Y)^2] + E[f(X)^2]) \\ & + pm\sigma_q^4\{3pE[f(X)^2 f(Y)^2] - pE[f(X)f(Y)]^2 \\ & + (m-1)E[f(X)^2]E[f(Y)^2]\} \end{aligned}$$

Proof: The proof is algebra intensive, so we omit it here. We plan to put it as a supplementary material. ■

The expression for the variance of the inner product between the two output vectors $\mathbf{x}^*$ and $\mathbf{y}^*$ has several interesting properties. It is an increasing function of the dimensionality of the input space and the number of hidden units (m) for a neural network implementation. These quantities are user-defined and thus can be changed depending on the application. In many situations it may be advantageous to choose $p > m$ thus increasing the expected variance in the distribution. Situations where $m = 1$ or $p = m$ may be suited for applications where the expected variance needs to be reduced. These parameters provide a mechanism to tune the degree of distortion in the output signal while maintaining control over the bound on $|E(f(X)f(Y)|$. We discuss these tradeoffs more in the next section.

VII. Special cases

In this section we study two special cases of the general transformation $\mathcal{T}$, when: (1)f is a sigmoid or $\tanh$ function (a popular choice for nonlinear mapping), and (2)f is an identity function making the resulting $\mathcal{T}$ linear.

A. $f = \tanh$ *function*

In this section we analyze the properties of $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ when f is a sigmoid or hyperbolic tangent ($\tanh$) function. Our choice of $f = \tanh$ is not arbitrary; it makes transformation $\mathcal{T}$ resemble that of a two-layer neural network, a tool widely used in data mining and machine learning for learning nonlinear relationships from the data. With such a substitution, $\mathcal{T}$ takes the following form:

$$\begin{aligned} \mathrm{H}(\mathbf{x}) \quad &= \quad \tanh(\mathrm{A} + \mathrm{W}\mathbf{x}) \\ \mathbf{x}^* \quad &= \quad \mathcal{T}(\mathbf{x}) = \mathrm{B} + \mathrm{Q}\mathrm{H}(\mathbf{x}) \end{aligned}$$

However, for the results here to describe such a trained neural network, one must assume that the weights are indeed independent and normally distributed with a mean of zero. Weights are assumed to be normal in much research in this area as shown in [26] and [27]. Other researchers have shown empirically that learning neural networks in high noise situations can lead to nearly linear networks [28].

Even with the substitution $f(x) = \tanh(x)$ in Equation (12), evaluation of $E[\tanh(X)\tanh(Y)]$ in closed form is still intractable due to the absence of anti-derivatives. Hence we use the bound presented in Lemma 6.2 to gain insight into $E[\tanh(X)\tanh(Y)]$. Let us first evaluate $E[\tanh^2(X)]$. By definition,

$$E[\tanh^2(X)] = \int_{-\infty}^{\infty} \tanh^2(x) \cdot \frac{e^{-x^2/(2\|\hat{\mathbf{x}}\|^2)}}{\sqrt{2\pi}\,\|\hat{\mathbf{x}}\|} dx$$

Unfortunately, an anti-derivative does not exist even for this function. We approximate the $\tanh$ function with a linear function that takes on the values -1 and 1 far to the left and right of the origin, respectively, and has a slope of constant, positive value in between. For simplicity we make this slope tangent to the slope of the f function at the origin, which means the slope of our approximation to be 1 over $[-1, 1]$ and zero otherwise. Letting $\Psi(X)$ denote the approximating function,

$$\tanh(X) \approx \Psi(X) = -1 \cdot \chi_{(-\infty,-1)} + x \cdot \chi_{[-1,1]} + 1 \cdot \chi_{(1,\infty)}$$

where χ is the indicator function. Figure 3 shows the original $\tanh$ function, the approximation to it and the step function. It is easy to see that,

$$\Psi(X)^2 = 1 \cdot \chi_{(-\infty,-1)} + x^2 \cdot \chi_{[-1,1]} + 1 \cdot \chi_{(1,\infty)}$$

Denoting $g_X(x)$ as the marginal distribution of X we get,

$$\begin{aligned} E[\tanh^2(X)] &= \int_{-\infty}^{\infty} \tanh^2(x) \cdot g_X(x) dx < \int_{-\infty}^{\infty} \Psi(X)^2 \cdot g_X(x) dx \\ &= 2\int_{-\infty}^{-1} g_X(x) dx + \int_{-1}^{1} x^2 \cdot g_X(x) dx \end{aligned}$$

$$\text{Term } 1 = 2\int_{-\infty}^{-1} \frac{e^{-x^2/(2\|\hat{\mathbf{x}}\|^2)}}{\sqrt{2\pi}\,\|\hat{\mathbf{x}}\|} dx = 2\Phi\left(-\frac{1}{\|\hat{\mathbf{x}}\|}\right)$$

where $\Phi(\cdot)$ is the CDF of a standard normal distribution. For evaluating Term 2, we evaluate the following integral.

$$\int x e^{-x^2/(2\|\hat{\mathbf{x}}\|^2)} dx = -\|\hat{\mathbf{x}}\|^2 e^{-x^2/(2\|\hat{\mathbf{x}}\|^2)} + c$$

$$\begin{aligned} \text{Term } 2 &= \frac{1}{\sqrt{2\pi}\,\|\hat{\mathbf{x}}\|} \left[\int_{-1}^{1} x^2 \cdot e^{-x^2/(2\|\hat{\mathbf{x}}\|^2)} dx\right] \\ &= \frac{-\|\hat{\mathbf{x}}\|}{\sqrt{2\pi}} \left(e^{-1/(2\|\hat{\mathbf{x}}\|^2)} + e^{-1/(2\|\hat{\mathbf{x}}\|^2)}\right) \\ &+ \|\hat{\mathbf{x}}\|^2 \left[\Phi\left(\frac{1}{\|\hat{\mathbf{x}}\|}\right) - \Phi\left(-\frac{1}{\|\hat{\mathbf{x}}\|}\right)\right] \end{aligned}$$

Combining the results,

$$E[\tanh^2(X)] < 2\Phi\left(-\frac{1}{\|\hat{\mathbf{x}}\|}\right) + \|\hat{\mathbf{x}}\|^2 \left[\Phi\left(\frac{1}{\|\hat{\mathbf{x}}\|}\right) - \Phi\left(-\frac{1}{\|\hat{\mathbf{x}}\|}\right)\right]$$

Using a similar argument, it can be shown that,

$$E[\tanh^2(Y)] < 2\Phi\left(-\frac{1}{\|\hat{\mathbf{y}}\|}\right) + \|\hat{\mathbf{y}}\|^2 \left[\Phi\left(\frac{1}{\|\hat{\mathbf{y}}\|}\right) - \Phi\left(-\frac{1}{\|\hat{\mathbf{y}}\|}\right)\right]$$

These results can now be combined to get the final bound of $|E[\mathbf{x}^*\mathbf{y}^*]| = |E[\tanh(X)\tanh(Y)]| < E[\tanh^2(X)]E[\tanh^2(Y)]$ using Lemma 6.2 and the expressions for $E[\tanh^2(X)]$ and $E[\tanh^2(Y)]$.

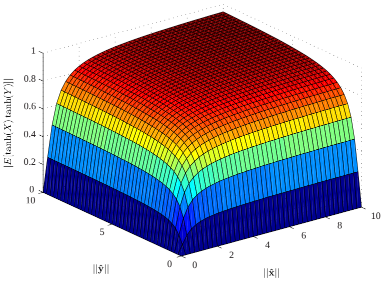


Fig. 5. Plot of the bound on $|E[\tanh(X)\tanh(Y)]|$ vs. $\|\hat{\mathbf{x}}\|$ and $\|\hat{\mathbf{y}}\|$.

Figure 5 shows the bound on $|E[\tanh(X)\tanh(Y)]|$ with variation of $\|\hat{\mathbf{x}}\|$ and $\|\hat{\mathbf{y}}\|$. By taking appropriate limits, it can be shown that the bound lies between 0 and 1. When both $\|\hat{\mathbf{x}}\|$ and $\|\hat{\mathbf{y}}\|$ are small *i.e.* close to the origin, we know that the expected inner product of their output should be close to 0 as well. The bound is a good approximation when we are close to the origin but becomes crude as we move further away from the origin. This bound gives a quantitative measure of privacy and is related to the probability of a successful attack given the data with no additional information. When we operate in a region far from the origin, the bound tells us that the maximum expected value of the output distribution is close to 1. This situation is the generalized version of the intuition described in Section IV-C and Figure 3. In that simplified example, the higher the slope, the less invertible the function, and therefore the higher degree of privacy. Note that with a finite (but large slope) with enough samples of inputs and corresponding outputs and under low-noise conditions, it will be possible to invert the map. However, the complexity of this inversion increases dramatically with the use of a full neural network architecture as discussed here. We therefore take the probability of a successful attack given the data to be proportional to $|E[\tanh(X)\tanh(Y)]|$.

B. Linear Transformation

The second transformation that we study is a linear transformation. Linear transformations have been widely studied in the form of random projection, multiplicative perturbation [5][20][3] where the output is linearly dependent on the input:

$$\mathbf{x}^* = \mathrm{T} + \mathrm{R}\mathbf{x}$$

where T and R are random translation and rotation matrices. In order for our transformation $\mathcal{T}$ to be linear, we assume that f is an identity function *i.e.* $f(x) = x, \forall x \in \mathbb{R}$. Unlike the previous section, in this section we show how a closed form expression for $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ can be developed for such a transformation.

Using the definition of X and Y, it is easy to show that,

$$E[f(X)f(Y)] = E[XY] = \hat{\mathbf{x}} \cdot \hat{\mathbf{y}}$$

Since $\hat{\mathbf{x}} = [\sigma_w \mathbf{x} \quad \sigma_a]^{\mathrm{T}}$ and $\hat{\mathbf{y}} = [\sigma_w \mathbf{y} \quad \sigma_a]^{\mathrm{T}}$,

$$\hat{\mathbf{x}} \cdot \hat{\mathbf{y}} = \sigma_w^2(\mathbf{x} \cdot \mathbf{y}) + \sigma_a^2$$

Combining these results, we have:

$$\begin{aligned} E[\mathbf{x}^* \cdot \mathbf{y}^*] &= p\sigma_b^2 + pm\sigma_q^2 E[XY] \\ &= p\sigma_b^2 + pm\sigma_q^2 \left(\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}\right) \\ &= p\sigma_b^2 + pm\sigma_a^2\sigma_q^2 + pm\sigma_q^2\sigma_w^2 \left(\mathbf{x} \cdot \mathbf{y}\right) \end{aligned}$$

This equation shows that for a linear transformation, the inner product of the output vectors is proportional to the inner product of the input vectors. In other words, the distances are preserved on average (up to scaling and translation). This result is in-line with what some other authors have reported elsewhere [3][5].

Let us investigate the quality of the bound for this transformation. Substituting $f(X) = X$ and $f(Y) = Y$, in Lemma 6.2, we see that the integrals are $E[X^2]$ and $E[Y^2]$ respectively. Now, since $X \sim N(0, \|\hat{\mathbf{x}}\|^2)$ and $Y \sim N(0, \|\hat{\mathbf{y}}\|^2)$, $E[X^2] = \|\hat{\mathbf{x}}\|^2$ and $E[Y^2] = \|\hat{\mathbf{y}}\|^2$. Thus,

$$E_{est}[XY] \leq \|\hat{\mathbf{x}}\| \, \|\hat{\mathbf{y}}\|$$

where E_{est} denotes the estimated value of the expectation. Therefore we can write the following expression for the bound:

$$E[\mathbf{x}^* \cdot \mathbf{y}^*] \quad \leq \quad p\sigma_b^2 + pm\sigma_q^2 \, \|\hat{\mathbf{x}}\| \, \|\hat{\mathbf{y}}\|$$

where,

$$\|\hat{\mathbf{x}}\| = \sqrt{\sigma_w^2(\|\mathbf{x}\|^2) + \sigma_a^2}$$
$$\|\hat{\mathbf{y}}\| = \sqrt{\sigma_w^2(\|\mathbf{y}\|^2) + \sigma_a^2}$$

Note that the true value of $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ and the estimated value differ only in θ, the angle between $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$. Figure 6 shows a plot of $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ as θ varies. For all the figures, the circles show the true variation of $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ vs. θ. The squares represent the bound. Note that for all the figures, the bound correctly represent the inner-product only when $\theta = 0, \pm 2\pi, \pm 4\pi, \ldots$. The three figures demonstrate the effect on the output for three values of $\|\hat{\mathbf{x}}\|$ and $\|\hat{\mathbf{y}}\|$. As can be seen, the bound is a good approximation of the true value when $\|\hat{\mathbf{x}}\|$ and $\|\hat{\mathbf{y}}\|$ are small.

VIII. Privacy Analysis and Distance Preservation for Anomaly Detection

The essence of perturbation-based privacy preservation in the context of data mining is that if a transformed dataset or query result is provided to a user it should be difficult or impossible to reconstruct the original, untransformed dataset. While several methods have been used to address this issue, the notion of function invertibility has not been used in this context in the past. Essentially, if one produces a set of N operations $O_1, O_2, ..., O_N$, and passes a data set through those operations, privacy will be preserved if the chain $O_N(O_{N-1}...(O_1))$ is not invertible either functionally, due to randomization of the output, or because of prohibitively high computation cost. In the past, researchers have analyzed the effects of randomization as a means of privacy protection and developed several sophisticated schemes to undo the randomization thereby recovering either the original data or a distribution. We present a general methodology explaining why randomization is breakable and propose a stronger functional privacy guarantee based on non-invertibility of functions. Note that the privacy guarantees of any linear orthogonal transformation (such as in [5] and [6]) holds true for our transformation as well.

Since our privacy model is related to the concept of function invertibility, we first define an invertible function.

Definition 8.1 (Invertible function): A function $f : \mathcal{D} \rightarrow \mathcal{R}$ is ***invertible*** iff (i) it is *one-to-one* (injective) *i.e.* $\forall (d_1, d_2) \in \mathcal{D}$, $f(d_1) = f(d_2) \Rightarrow d_1 = d_2$, and (ii) it is *onto* (surjective) *i.e.* $\forall r \in \mathcal{R}$, $\exists d \in \mathcal{D}$, such that $r = f(d)$.

In order to diminish the probability of inverting a function and thus attack a privacy preservation scheme, the function must be such that there exists a many-to-one mapping from the domain of the function to the range of the function. In this situation, given the output, it would be difficult or impossible to map back to the original data space. In the event that only the outputs are provided without the inputs this reverse mapping would be made more difficult. Below we formally define this notion of privacy.

Definition 8.2 (Privacy preserving transformation): A transformation (or a function) $\mathcal{T}$ is ***privacy preserving*** if, for any dataset D, the composition transformation $\mathcal{T}^{-1}(\mathcal{T}(D))$ does not give D back *i.e.* $\mathcal{T}^{-1}(\mathcal{T}(D)) \neq D$.

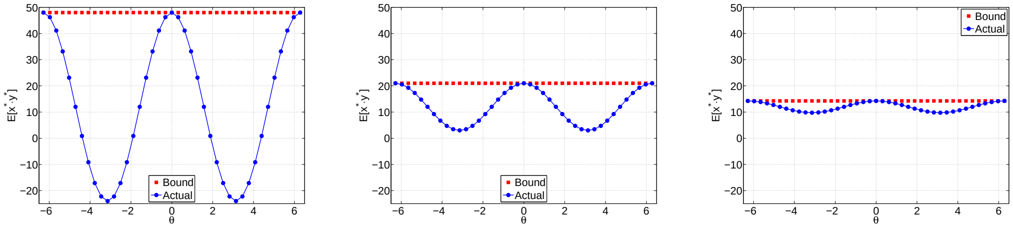
Therefore given the output $\mathcal{T}(D)$ and $\mathcal{T}$ it is impossible to get D back.

The idea of using non-invertible functions for privacy preservation is not new; it has been used successfully thus far in the field of security and cryptography [17][29]. The Hash functions such as SHA and MD-5 were developed with the basic idea that no polynomial time algorithm exists for finding the reverse mapping which will break the encryption. To the best of our knowledge, this concept has not yet been explored in the context of privacy preserving data mining. In the past, researchers have only analyzed the situations in which the transformation $\mathcal{T}$ is either random multiplicative or additive noise or both. Mathematically, both of these transformations are invertible and thus are not privacy preserving. This claim has been bolstered in recent years by the development of sophisticated techniques for thwarting these transformations such as in [2] and [3]. It is fairly straightforward to show that our non-linear non-invertible distortion technique is resilient to such attacks. Of course, privacy comes at a price — higher privacy decreases the accuracy.

Data privacy usually comes at a price. Utility or usefulness of the data is often lost during privacy preservation using perturbation or distortion schemes. For example, consider the transformation:

$$f : \mathbb{R} \rightarrow \{0, 1\}.$$

By Definition 8.2, this transformation is privacy preserving. However, since all the data is mapped to a single bit, it is



(a) $\|\hat{\mathbf{x}}\| = \|\hat{\mathbf{y}}\| = 1$

(b) $\|\hat{\mathbf{x}}\| = \|\hat{\mathbf{y}}\| = 0.5$

(c) $\|\hat{\mathbf{x}}\| = \|\hat{\mathbf{y}}\| = 0.25$

Fig. 6. Variation of the output $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ with respect to θ (in radians), the angle between $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$. Circles represent the true output and squares represent the bound. For all figures, the bound is independent of θ. For a fixed $\|\hat{\mathbf{x}}\|$ and $\|\hat{\mathbf{y}}\|$, actual output oscillates and equals the bound only at $\theta = 0, \pm 2\pi, \ldots,$. As $\|\hat{\mathbf{x}}\| \to 0$ and $\|\hat{\mathbf{y}}\| \to 0$, the actual value approaches the estimated value. The bound is very tight when $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are close to the origin.

not directly clear how important the data will be for data mining purposes. This tradeoff can be controlled easily in our framework by changing the slope (θ) of the non-linear function used. In the remainder of this section we discuss how our data distortion scheme offers data utility in the case of outlier detection.

In this paper we have used the definition of outliers as in [30] and [31]. By definition, distance based outliers are those for which:

- there are fewer than p other points at a distance of d
- the distance (or the average distance) to the k nearest neighbors are the greatest.

Note that the crux of all these computations use a distance metric defined on the input space. Specifically, let

$$\mathbf{dist} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \quad \mathbf{dist}^* : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$$

be a distance measure on the input and output space respectively, which computes the Euclidean distance between two vectors $\mathbf{x}$ and $\mathbf{y}$. Now three cases can occur after the transformation (using $\tanh$ as the non-linearity):

1) $\mathbf{x}$ and $\mathbf{y}$ are not outliers
   - In this case,

$$\mathbf{dist}^*(\mathcal{T}(\mathbf{x}), \mathcal{T}(\mathbf{y})) \approx \mathbf{dist}(\mathbf{x}, \mathbf{y})$$

   assuming $\mathbf{x}$ and $\mathbf{y}$ lie close to the origin and the $\tanh$ function is linear in this region. In this case, the distances are approximately preserved. The privacy protection is typically based on linear randomization (rotation and translation) and therefore less. In our scenario this is acceptable since the normal operating conditions are similar for many airline companies and hence the lesser privacy guarantee for these data points may be acceptable.

2) $\mathbf{x}$ is an outlier while $\mathbf{y}$ is not
   - In this case,

$$\mathbf{dist}^*(\mathcal{T}(\mathbf{x}), \mathcal{T}(\mathbf{y})) \approx \mathbf{dist}(c, \mathbf{y}).$$

   where c is a constant. Note that the distances are not preserved. However, with a proper choice of threshold, we can distinguish between c and $\mathbf{y}$. In this case, given c, it is impossible to find $\mathbf{x}$. This is because the transformation is non-invertible since, $\mathbf{x}$, being an outlier, is far away from the origin. Thus the privacy guarantee is high for all outliers. This is important since outliers may be specific to an airline company, and mapping all outliers to a single entity may preserve privacy while still allowing their detection as long as they are away from the non-outlier data points.

3) $\mathbf{x}$ and $\mathbf{y}$ are outliers
   - In this case,

$$\mathbf{dist}^*(\mathcal{T}(\mathbf{x}), \mathcal{T}(\mathbf{y})) \approx \mathbf{dist}(c, c) \approx 0$$

   which implies that all outliers approximately get mapped to the same points. Since we are not interested in distinguishing the outliers, this mapping is acceptable. Moreover, this ensures that given c, it is impossible for an attacker to figure out if it came from $\mathbf{x}$ or $\mathbf{y}$ (one-to-many mapping).

Referring back to Figure 6 we see that for a linear transformation, the quantity $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ easily bounds the relative positions of the original input vectors, particularly when they are close to the origin. This implies, regardless of the nature of the linear transformation, it will always be possible to reidentify some important properties of the data set if those vectors lie close to the origin. However, as they move away from the origin, the actual variation in the expectation sinusoidally oscillate under the bound. Because the integral needed to compute $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ is intractable for nonlinear transformations, we can only analyze the bound given in Figure 5 and see that the transformation becomes highly nonlinear and therefore highly private in the situation where $E[\mathbf{x}^* \cdot \mathbf{y}^*]$ is close to unity.

IX. Experimental Results

In this section we demonstrate the quality of our non-linear transformation in preserving the inner-product among the feature vectors. We provide experimental results on a publicly available high-fidelity aircraft engine simulation dataset (C-MAPSS) and a proprietary aviation dataset (CarrierX).

A. Simulation Environment and Dataset

Our experimental setup uses a distance-based outlier detection algorithm Orca developed by Bay and Schwabacher [32] to test the quality of distance-preservation of our transformation. Orca assigns an anomaly score (between 0 and 1) to each point in the dataset based on its distance to its nearest neighbors. The higher the distance, the higher the score. Our data distortion technique preserves distances if the data is close to the origin, and distorts them otherwise. Therefore a distance-based outlier detection technique should be able to detect outliers under our potential non-linear transformation. Orca is written in C++ with a wrapper written in Matlab. The default value for the distance computation was chosen as the average distance to five nearest neighbors. All our simulations were run on a 64-bit 2.33 GHz quad core dell precision 690 desktop running Red Hat Enterprise Linux version 5.4 having 2GB of physical memory.

In our experiments we report the *detection rate*. By detection rate we mean the percentage of outliers which are preserved even after the transformation. We repeat this experiment several times and report the mean and the standard deviation of the detection rate.

The first dataset is simulated commercial aircraft engine data. This data has been generated using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) [33]. The dataset contains 6,875 full flight recordings sampled at 1 Hz with 29 engine and flight condition parameters recorded over a 90 minute flight that includes ascent to cruise at 35000 feet and descent back to sea level. This dataset has 32,640,967 tuples. Interested readers can refer to this dataset at DASHlink[3].

The second dataset is a real life commercial aviation dataset of a US regional carrier (CarrierX) consisting of 3,573 flights[4]. Each flight contains 47 variables. Out of these 39 are real-valued (continuous) attributes while the remaining 7 are discrete (binary). In our previous study (not reported in this paper) we have seen that there are several anomalies in this dataset detectable by Orca. We hope to detect a high percentage of those outliers even after our non-linear distortion. Unlike C-MAPSS dataset which is public, the CarrierX dataset is proprietary and hence there is strong motivation to protect data privacy. Note that our technique only distorts the real-valued attributes. However the code works even if we include discrete attributes.

B. Performance Results

In this section we show the quality of outlier detection before and after the transformation. For all the experiments, we have preprocessed the datasets by transforming each variable independently to lie between 0 and 1.

1) CMAPSS Dataset: Figure 7 shows the effect of linear distortion on the outcome of the anomaly scores. For this experiment, we ran Orca with the default parameters on the CMAPSS dataset. The output of the algorithm is a set of anomaly scores for each point. We then sort these points and select the top 500 among them. The stars in Figure 7 show the scores output by Orca on the original dataset after they have been normalized between 0 and 1. In order to distort the dataset, we used the following transformation:

$$\mathcal{T}(\mathbf{x}) = \mathrm{B} + \mathrm{Q} \times (\mathrm{A} + \mathrm{W}\mathbf{x})$$

Using this transformation, we again run Orca on this distorted dataset. The diamond markers in Figure 7 show the normalized anomaly scores of the same 500 outliers in the distorted dataset. As can be seen in the figure, there is a high degree of correlation among the two scores. Since linear transformation preserves distances, for any outlier point, distance to its k-nearest neighbors are also preserved. This is why we see very similar anomaly scores for the two experiments. Notice

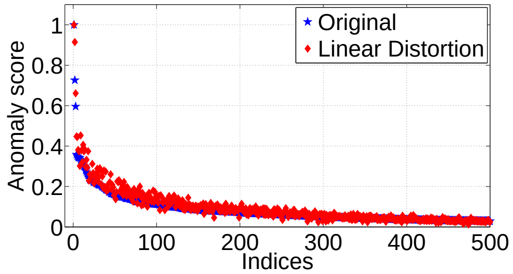


Fig. 7. Plot of anomaly scores of original CMAPSS dataset (star) and transformed datasets using linear (diamond) transformation as produced by a distance-based outlier detection technique Orca[32]. The x-axis shows the indices of the top 500 anomalies as found by Orca. The diamond markers show the anomaly scores of the same 500 indices after the transformation.

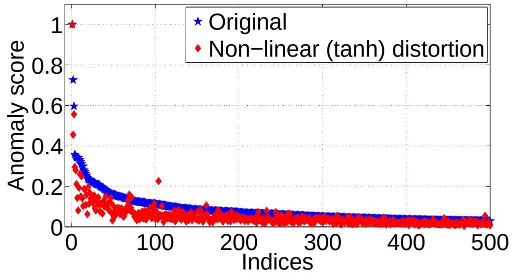


Fig. 8. Plot of anomaly scores of original CMAPSS dataset (star) and transformed datasets using tanh (diamond) transformation as produced by a distance-based outlier detection technique Orca[32]. The x-axis shows the indices of the top 500 anomalies as found by Orca. The diamond markers show the anomaly scores of the same 500 indices after the transformation.

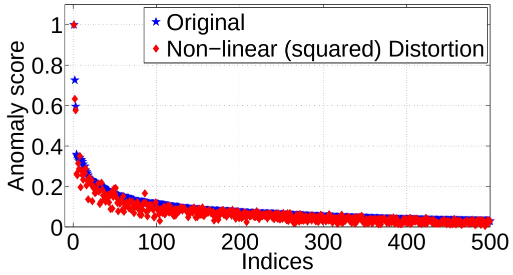
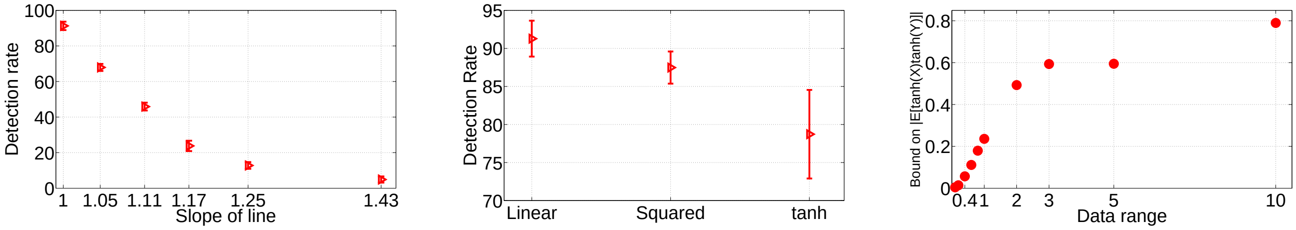


Fig. 9. Plot of anomaly scores of original CMAPSS dataset (star) and transformed datasets using squared (diamond) transformation as produced by a distance-based outlier detection technique Orca[32]. The x-axis shows the indices of the top 500 anomalies as found by Orca. The diamond markers show the anomaly scores of the same 500 indices after the transformation.

[3] https://dashlink.arc.nasa.gov/data/c-mapss-aircraft-engine-simulator-data/

[4] We cannot release the name of the carrier due to the data sharing agreement



(a) Detection rate vs. slope

(b) Detection rate vs. non-linearity

(c) Detection rate vs. data range

Fig. 10. Plot of detection rate of CMAPSS vs. different parameters. The reference set is the top 500 outliers assigned by Orca. We refer to detection rate as the percentage of outliers in that list which are correctly identified after the transformation. The results are an average of 50 independent trials.

that the variation in the anomaly scores is higher than that of the original data due to the random linear projection. These variations become more emphasized under nonlinear transformations.

Figure 8 shows the effect of nonlinear distortion on the CMAPSS dataset using the $\tanh$ function. As before, the star markers represent the outlier scores of the top 500 anomalies on the original dataset. For the distortion, we have used the following transformation:

$$\mathcal{T}(\mathbf{x}) = \mathrm{B} + \mathrm{Q} \times \tanh(\mathrm{A} + \mathrm{W}\mathbf{x})$$

The diamond markers show the anomaly scores of the same 500 outliers after distortion. In this case, there is more deviation in the anomaly scores compared to the linear distortion case. Notice that although the transformation provides a high degree of privacy compared to the linear transformation, the highest scoring anomalies are still discovered by the anomaly detection algorithm. This result supports the intuition and the derivations shown earlier: *nonlinear transformation can allow anomalies to pass through a privacy preserving transformation*.

We have also tested a quadratic nonlinearity: *i.e.* $f(x) = x^2$: $\mathcal{T}(\mathbf{x}) = \mathrm{B} + \mathrm{Q} \times (\mathrm{A} + \mathrm{W}\mathbf{x})^2$. Figure 9 shows the effect of this transformation. In this case as well, there is a good correlation among the true and transformed outliers. Notice that the overall variation is lower than that of the $\tanh$ transformation. In this case the privacy preservation is high compared to linear distortion due to the fact that the nonlinear function is non-invertible.

Our next experiments analyze the variation of the detection rate and privacy preservation using this dataset and the $\tanh$ function. First, we have experimented with an increasing slope of the transformation (similar to Figure 3). As shown in Figure 10(a), the detection rate is very sensitive to the slope — it drops to approximately 4% for a slope of 1.43. This is as expected since with increasing slope, more of the data gets mapped to the constant regions, making it extremely difficult for the outlier detection algorithm to extract the anomalous patterns. The privacy using such high slope transformation is expected to be very high.

For this dataset we also show the detection rate when different types of distortion are used. As shown in Figure 10(b), for linear distortion, the mean detection rate is 91.28% with a standard deviation of 2.36%. Similar results for square distortion, are 87.48% and 2.11%. Finally, using $\tanh$ function, we get a mean detection rate of 78.72% and with 5.82% variation. Figure 10(b) gives a plot of the mean and one standard deviation estimate of the variation in detection rate.

Finally, in Figure 10(c) we give an idea of the amount of privacy that is preserved as the range of the data is varied. Using our bound in Lemma 6.2, we see that if the data lies close to the origin (range of 0 to 0.1), the privacy is very low. As the range of the data is increased, the privacy is increased. This explains our hypothesis that the nearer the data is to the origin, the lower the data privacy and vice-versa. Therefore, in order to have more privacy, one might map the data to a large range in which case, as argued, non-invertibility preserves data privacy.

2) CarrierX Dataset: We applied two types of transformation on this dataset. Figure 11 shows the outlier detection results using a linear transformation. As before, the blue stars refer to the actual top 500 anomalies while the red diamonds refer to the scores of the same 500 points after transformation. We noticed, that on average the detection rate is 88% with a standard deviation of 1.3% for this linear transformation. Similarly, Figure 12 shows the anomalies detected when $\tanh$ non-linearity is used. In this case, we have observed a mean detection rate of 68% with a standard deviation of 1.7%.

Therefore for all these experiments we see that our distortion technique provides a good detection rate for different types of non-linearity used.

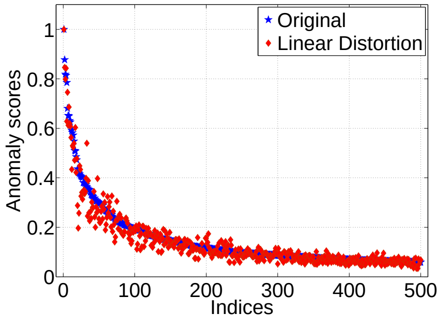


Fig. 11. Plot of anomaly scores of original CarrierX dataset (star) and transformed datasets using linear (diamond) transformation as produced by a distance-based outlier detection technique Orca[32]. The x-axis shows the indices of the top 500 anomalies as found by Orca. The diamond markers show the anomaly scores of the same 500 indices after the transformation.

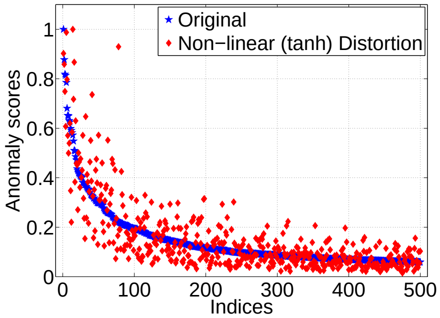


Fig. 12. Plot of anomaly scores of original CarrierX dataset (star) and transformed datasets using $\tanh$ (diamond) transformation as produced by a distance-based outlier detection technique Orca[32]. The x-axis shows the indices of the top 500 anomalies as found by Orca. The diamond markers show the anomaly scores of the same 500 indices after the transformation.

X. Conclusion

We have shown a general method for computing the bounds on a nonlinear privacy preserving data mining technique with applications to anomaly detection. We have also shown the connection between the invertibility of a function and privacy preservation, and have computed rigorous bounds on the relationship between the distances of input vectors and the expected distances of the output vectors. These nontrivial bounds show that privacy preservation increases as the input vectors move further from the origin. We have also demonstrated that for real-world applications, such as engine health monitoring, the nonlinear transformation approach allows anomalies to pass through the transformation while maintaining a high degree of privacy. We have given a novel method for quantifying privacy due to a general nonlinear transformation and have shown that this quantity can be treated as being proportional to probability of a successful attack. We have made all the source codes of this work and the supplemental information available at dashlink [34].

Acknowledgments

This work was supported by the NASA Aviation Safety Program, Integrated Vehicle Health Management Project. We would also like to thank the anonymous reviewers for their excellent comments and suggestions.

References

[1] A. Narayanan and V. Shmatikov, “Robust De-Anonymization of Large Sparse Datasets,” *Proc. of IEEE SSP’08*, pp. 111–125, 2008.

[2] K. Liu, C. Giannella, and H. Kargupta, “An Attacker’s View of Distance Preserving Maps for Privacy Preserving Data Mining,” in *Proceedings of PKDD’06*, Berlin, Germany, 2006, pp. 297–308.

[3] K. Chen, G. Sun, and L. Liu, “Towards Attack-Resilient Geometric Data Perturbation,” in *Proceedings of SDM’08*, 2008, pp. 78–89.

[4] L. Sweeney, “k-anonymity: A Model for Protecting Privacy,” *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[5] K. Liu, H. Kargupta, and J. Ryan, “Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining,” *IEEE TKDE*, vol. 18, no. 1, pp. 92–106, January 2006.

[6] S. Mukherjee, Z. Chen, and A. Gangopadhyay, “A Privacy-preserving Technique for Euclidean Distance-based Mining Algorithms using Fourier-related Transforms,” *VLDB J.*, vol. 15, no. 4, pp. 293–315, 2006.

[7] S. T. Sarasamma, Q. A. Zhu, and J. Huff, “Hierarchical Kohonenen net for anomaly detection in network security,” *IEEE SMC Part B*, vol. 35, no. 2, pp. 302–312, 2005.

[8] V. Barnett and T. Lewis, *Outliers in Statistical Data*. John Wiley, 1994.

[9] D. Hawkins, *Identification of Outliers*. Chapman and Hall, 1980.

[10] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection : A Survey,” *ACM Computing Surveys (to appear)*, 2009.

[11] “Voluntary Aviation Safety Information-Sharing Process.” [Online]. Available: www.faa.gov/library/reports/medical/oamtechreports/2000s/media/200707.pdf

[12] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “ℓ-diversity: Privacy beyond k-anonymity,” *TKDD*, vol. 1, no. 1, 2007.

[13] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond k-anonymity and ℓ-diversity,” in *Proceedings of ICDE’07*, 2007, pp. 106–115.

[14] J. J. Kim and W. E. Winkler, “Multiplicative Noise for Masking Continuous Data,” Statistical Research Division, U.S. Bureau of the Census, Washington D.C., Tech. Rep. Statistics #2003-01, April 2003.

[15] R. Agrawal and R. Srikant, “Privacy-preserving Data Mining,” in *Proceedings of SIGMOD’00*, May 2000, pp. 439–450.

[16] J. Vaidya, C. Clifton, and M. Zhu, *Privacy Preserving Data Mining*, ser. Series: Advances in Information Security. Springer, 2006, vol. 19.

[17] A. C. Yao, “How to Generate and Exchange Secrets,” in *Proceedings of FOCS’86*, Canada, October 1986, pp. 162–167.

[18] J. C. Silva and M. Klusch, “Privacy-Preserving Discovery of Frequent Patterns in Time Series,” in *Proceedings of Industrial Conf. on Data Min. ’07*, 2007, pp. 318–328.

[19] C. Dwork, “Differential Privacy,” in *Proceedings of ICALP’06*, vol. 4052. Springer, 2006, pp. 1–12.

[20] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, “On the Privacy Preserving Properties of Random Data Perturbation Techniques,” in *Proceedings of ICDM’03*, Melbourne, Florida, November 2003, p. 99.

[21] A. Teoh and C. T. Yuang, “Cancelable Biometrics Realization With Multispace Random Projections,” *IEEE SMC, Part B*, vol. 37, no. 5, pp. 1096–1106, 2007.

[22] S. Warner, “Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias,” *Journal of American Statistical Association*, vol. 65, no. 63–69, 1965.

[23] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy Preserving Mining of Association Rules,” in *Proc. of KDD’02*, 2002, pp. 217–228.

[24] A. Evfimievski, J. Gehrke, and R. Srikant, “Limiting Privacy Breaches in Privacy Preserving Data Mining,” in *Proc. of PODS’03*, 2003, pp. 211–222.

[25] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C (2nd ed.): The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 1992.

[26] I. Bellido and E. Fiesler, “Do Backpropagation Trained Neural Networks Have Normal Weight Distributions?” in *Proceedings of ICANN’93*, Amsterdam, Netherlands, September 1993, pp. 772–775.

[27] T. Szabó, L. Antoni, G. Horváth, and B. Fehér, “A Full-Parallel Digital Implementation for Pre-Trained NNs,” in *Proceedings of IJCNN’00-Volume 2*, Como, Italy, July 2000, p. 2049.

[28] B. Lebaron and A. S. Weigend, “Evaluating Neural Network Predictors by Bootstrapping,” Computer Science Department, University of Colorado at Boulder, Tech. Rep., 1994.

[29] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-key Cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[30] E. M. Knorr, R. T. Ng, and V. Tucakov, “Distance-based Outliers: Algorithms and Applications,” *VLDB*, vol. 8, no. 3-4, pp. 237–253, 2000.

[31] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient Algorithms for Mining Outliers from Large Data Sets,” *SIGMOD Rec.*, vol. 29, no. 2, pp. 427–438, 2000.

[32] S. D. Bay and M. Schwabacher, “Mining Distance-based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule,” in *Proceedings of KDD’03*. New York, NY, USA: ACM, 2003, pp. 29–38.

[33] D. K. Frederick, J. A. DeCastro, and J. S. Litt, “User’s Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS),” *NASA Technical Manuscript*, vol. 2007-215026, 2007.

[34] “Dashlink resources.” [Online]. Available: https://dashlink.arc.nasa.gov/topic/privacy-preserving-outlier-detection-through-random-nonlinear-da/