A New Noise Generating Method Based on Gaussian Sampling for Privacy Preservation

Bo Ma¹(✉), Wei Qi Yan¹, Edmund Lai¹, and Jingsong Wu²,³

¹ School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand
{bo.ma,weiqi.yan,edmund.lai}@aut.ac.nz
² Department of Electrical Engineering, Universidad de Chile, Santiago, Chile
jswu@ieee.org
³ Department of Artificial Intelligence, Guilin University of Electronic Technology, Guilin, China
http://www.aut.ac.nz

Abstract. Centralised machine learning brings in side effect pertaining to privacy preservation, most of machine learning methods prone to using the frameworks without privacy protection, as current methods for privacy preservation will slow down model training and testing. In order to resolve this problem, we develop a new noise generating method based on information entropy by using differential privacy for betterment the privacy protection which owns the architecture of federated machine learning. Our experiments unveil that this solution effectively preserves privacy in the vein of centralized federated learning. The gained accuracy is promising which has a room to be uplifted.

Keywords: Differential privacy · Noise generating · Gaussian subsampling

1 Introduction

Privacy leakage is defined in [16] as “the accidental or unintentional distribution of private or sensitive data to an unauthorized entity”. The methods for tackling the private sensitive data include:

- randomization [23];
- k-anonymity [18] and l-diversity [13];
- distributed privacy preservation [12];
- downgrading the effectiveness of data mining [3]

Randomization by deliberately introducing noises into a datasets, is a simple and effective way to provide differential privacy (DP). It has been applied to adaptive boosting (AdaBoost) [14], principal component analysis (PCA) [21],
linear and logistic regression [8], support vector machines (SVM) [5], risk minimization [19], and continuous data processing [22]. A general mechanism to control the amount of added Gaussian noises was introduced in [6], in conjunction with the idea of $\varepsilon$-differential privacy, it provides the same amount of privacy to individuals so that the statistical results based on the dataset are roughly independent on the data of any individuals.

In this paper, we explore and exploit an image-based privacy preserving algorithm in the dataset where the noises of the data are identified. It is true that data sampling process is able to assist to identify information from the sensitive attributes. Furthermore, data sampling is a key determinant in measuring the quality of privacy preservation because the excellent methods for privacy preservation offer users to prove their original data. The challenges are how to retrieve and connect those links together as evidences, so as to identify the levels of a user’s privacy leakage, and trace unspecific users with a high accuracy. In this paper, we address the problem of data provenance from the viewpoint of collaborative learning for privacy preservation.

Our contribution in this paper is to present noise generating methods with a machine learning framework which outperforms the existing solutions. The remaining parts of this paper will be organized as follows. We present the existing methods for privacy preservation in Sect. 2.1, our proposed methods for privacy presentation will be explicated in Sect. 3. We will demonstrate our resultant evaluations in Sect. 4, our conclusion will be drawn in Sect. 5.

2 Related Work

2.1 Gaussian Noise Generating

Noise sampling and generating are the vital methods for protecting privacy in differential privacy. An approach by using Gaussian noise sampling $\{\phi_n\}$ was proffered [15] for privacy preservation. This approach applies probability density function (PDF) with noisy samples as $p_\phi(\phi_n)$ having zero mean, if the standard deviation and mean are denoted as $\sigma$ and $\mu$ respectively, the PDF is

$$p_\phi(\phi_n) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(\phi_n - \mu)^2}{2\sigma^2}}. \quad (1)$$

In Eq. (1), the Gaussian PDF is obtained from

$$\phi = \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{\infty} e^{j\phi_n - \frac{\phi_n^2}{2\sigma^2}} d\phi_n \quad (2)$$

The Gaussian noise is generated from the PDF of samples,

$$E\{z\} = A^2[\sum_{n=1}^{N} |a_n|^2 + \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}] \quad (3)$$

where $A^2$ is the power of samples, $|a_n|^2$ is the deterministic weights, $z$ is the discrete Fourier transform of the given samples.
However, this method relies on the PDF of input samples, the deviation of weighting coefficients \( \{a_n\} \) depends on the PDF, the deviation of noises cannot be employed to protect the privacy efficiency.

### 2.2 Whittle’s Noise Estimator

In 1954, Whittle [20] proposed a method to estimate the noise level. In this method, it is assumed that \( x_t \) is the sample of \( X = \{x_t, t = 1, 2, \cdots, n\} \) in the self-fitting process. For this process, all other parameters except the variable \( \text{var}(X) \) and the parameter \( H \) are known. Let \( S(A : H) \) denote the power spectrum of \( X \) if it is normalized to the variance 1.0, let \( I(A) \) denote the power spectrum of \( x_t \), and the power spectrum is estimated by using Fourier transform.

In order to estimate \( H \), we need to find the minimum of \( H \)

\[
g(\hat{H}) = \int_{-\pi}^{\pi} \frac{I(\lambda)}{f(\lambda : \hat{H})} d\lambda. \tag{4}
\]

If the length of \( X = \{x_t, t = 1, 2, \cdots, n\} \) is \( n \), then under the frequencies, the number is converted into a discrete summation

\[
\lambda = \frac{2\pi}{n}, \frac{4\pi}{n}, \ldots, \frac{2(n-1)\pi}{n}. \tag{5}
\]

This estimator reflects the truth that \( I(A) \) is independent, the mean is subject to the distribution \( f(A; H) \) as an exponential function, the variance of the estimator is

\[
\sigma^2_H = 4\pi \left[ \int_{-\pi}^{\pi} (\log f(\omega)) e^{\lambda H} d\omega \right]^{-1} \tag{6}
\]

By examining the sample paths, Whittle’s estimator is applied to compare with \( \sigma^2_H \), which determines whether \( H \) is within the acceptable range of \( H \). The Whittle’s estimator is not a test regarding whether a sample is consistent with long-range dependence. Rather than, it is an estimator of \( H \), given the assumption that the power spectrum of the underlying process corresponds to \( f(\lambda : H) \).

### 2.3 The Method Based on Fourier Transform

A method [10] is proposed for synthesizing fractional Gaussian noises by using discrete time Fourier transform (DTFT), where \( f(\lambda : H) \) is known as a variance, the fractional Gaussian noises have been generated via power spectrum. In addition, it constructs a sequence of complex numbers \( z \) which is akin to power spectrum. Besides, \( z_i \) is the sampling path for frequency-domain and an inverse discrete time Fourier transform is employed to find the best output of \( x_i \) related to \( z_i \). Because \( x_i \) is generated from power spectrum of fractional Gaussian noises (FGN), FGN is automatically corrected via the power spectrum by utilizing the Fourier pair, hence, \( x_i \) is guaranteed to enable the automatic correction, this is also the salient attribute of FGN process.
However, this approach has a problem, because $f(\lambda : H)$ must be accurately computed, $z_i$ is connected to the power spectrum of FGN. Hence, $z_i$ is independent and $f(\lambda : H)$ may not be so accurate due to the changes of $z_i$.

Because DTFT and its inverse are rapidly computed by using fast Fourier transform (FFT), we prefer to using our method as an FFT method for synthesizing the fractional Gaussian noises. We will not prove that the method results in true FGN due to various approximations whilst the method is being developed. But we will assert that the method effectively produces FGN noises. By the way, the sample paths generated by using this method are indistinguishable with the current FGN tests for the purposes such as simulations, the sample paths are with a high degree of confidence.

2.4 Distributed SGD for Differential Privacy

Stochastic gradient descent (SGD) provides an effectively iterative solution to minimize a function so as to reach the local minimum, which has been applied to artificial neural networks, Bayesian networks, genetic algorithms, and simulated annealing. A SGD method for privacy preservation was proffered in 2016 [1]. In 2018, an efficient approach was developed and improved for privacy-preservation [2] so as to approach the local minimum. Thus, through a mapping $M : X \rightarrow O$, the privacy cost is defined as

$$Pr(M(f(D_1)) \in S) \leq e^{\varepsilon} Pr(M(f(D_2)) \in S) + \delta$$

(7)

where $f(D)$ is an iteration for SGD, the probability $\delta$ means $\varepsilon$ -differential privacy is breached, especially, the noises are generated by using [1].

$$M(f(D)) \triangleq f(D) + \mathcal{N}(0, S^2_f \cdot \sigma^2)$$

(8)

where $\mathcal{N}(0, S^2_f \cdot \sigma^2)$ is the distribution of samples and $S^2_f$ is sensitivity of $f(\cdot)$.

To avoid the deviation problem in Eq. (3), we put forward a privacy preservation method to refrain this issue, set forth a privacy measurement to detect the deviation between training samples and added noises, the latter can be added during the training process. We compared the existing method in Eq. (8), the test result will be given in evolution phase.

3 Our Methods

3.1 Contribution

Our contribution in this paper is to present noise generating methods fit into the exists machine learning framework which outperforms the existing solutions. The remaining parts of this paper will be organized as follows. We present the existing methods for privacy preservation in Sect. 2.1, our proposed methods for privacy presentation will be explicated in Sect. 3. We will demonstrate our resultant evaluations in Sect. 4, our conclusion will be drawn in Sect. 5.
3.2 The Process of Our Methods

In order to protect the privacy, in this paper, we develop the method based on noise analysis [15] and link the existing approach with differential privacy [2]. Figure 1 shows the process to tackle the problem of privacy preservation by using federated machine learning. The first step is to run a sanitizer [4] for differential privacy. The sanitizer usually is denoted as an algorithm to protect $\varepsilon$-differential privacy. In privacy preservation, the dataset does not affect the sanitizer, the data entropy is measured by using estimators, the effective noisy pattern is identified.

The second step is a machine learning stage. In this step, the noises generated in the first step are injected into the data by using SGD in federated machine learning. The final step is related to generalization. The function is assigned for model validation which is related to privacy of the output data. In this step, generative adversarial network (GAN) is taken into account to detect data leakage. The last step is to generate similar data from the original input data so as to measure its privacy.

The fake data is employed to reproduce the privacy if the proposed privacy preserving method fails to protect privacy. The step 1, 2, and 3 are formed as a dynamic process, the noises need to be rechecked so as to ensure the original data is safe enough to defend privacy leakage.

3.3 Noise Variant in Stochastic Gradient Descent

For privacy preservation, we assume the distance between two datasets is $d(D, D') = ||D - D'||$, an attacker is expected to infer whether the dataset is $D$ or $D'$, according to the distance between local sensitivity $f(D)$ and the output in local differential privacy or $\varepsilon$-differential privacy [7].
In order to protect the privacy in model training, recent results divulge that the variant of stochastic gradient descent achieves optimal error for minimizing Lipschitz convex functions over 2-bounded sets, the randomized ‘dropout’ is applied to prevent overfitting, which also strengthens the privacy and guaranteed to find the solution.

In previous work, this problem [17] is identified through collaborative deep learning under multiple participants. However, the existing algorithms [17] cannot be executed in complex environment, because most of data are stored in the cloud. The new challenge of collaborative deep learning is how to cooperate with the cloud and local terminals. Thus, the second problem is how to preserve the privacy during and after the model training process, how to balance the costs of privacy preservation and the utility of data output or training efficiency.

In order to balance the privacy cost and the utility of the resultant estimations in statistics, the boundary of mutual information has to be found and the noise subsampling has to be determined. These two steps will influence the amount of privacy preserved. The subsampling based on Gaussian distribution will be presented in the next section.

### 3.4 Gaussian Distribution for Subsampling

In the proposed solution, we mingle (Gaussian) distribution estimator for differential privacy, and machine learning algorithms together. We create this solution to speed up the model training process for privacy preservation, we also provide feedback mechanism to test results, and amend the generated noises in preparing stage. Our new approach leverages the impact between privacy and training. In addition, we use GAN-based generation method to measure and adjust the privacy level of output data so as to give feedback to the process of noise generating. In addition, differential privacy sanitizer is synthesized with traditional statistical methods like support vector machine or multilayer perceptron. Moreover, the kernel for pattern classification has been analysed via the sanitizer, the input data and generated noisy data are hashed in the learning process. In order to estimate the probability of privacy leakage from training output, the existing approach needs to test samples through our entropy analysis model. The entropy analysis is associated with data privacy protection.

We present the randomized method as $M : \mathbb{G} \rightarrow \mathbb{R}$ with Gaussian distribution $\mathbb{G}$ and range $\mathbb{R}$ which satisfies $(\varepsilon, \delta)$-differential privacy. We assume the adjacent inputs as $i \in \mathbb{G}$, the random noise is $n' \in \mathbb{G}$ for any subset of outputs $D \subseteq \mathbb{R}$ and it has $\Pr[M(d) \in \mathbb{R}] \leq \Pr[M(d') \in \mathbb{P}]$. The output of $\varepsilon$-differential privacy is obtained according to $\delta \leq \frac{\Delta p \log \frac{1+\Delta p}{\varepsilon}}{2}$. We apply the noise generating model to sample the subset $D_t \leq K(x, y)$. The noise generating model $n_t$ updates itself by using $\Delta n^k = n^k - n_t$, where $n^k$ is the generated noise sample, $n_t$ is the noise $n$ at time $t$. We expand the Gaussian distribution-based noise generating model as

$$n_{t+1} \leftarrow n_t + \frac{1}{m} \left( \sum_{k=0}^{t} \frac{\Delta n^k}{\max \langle \Phi(x), \Phi(y) \rangle, \| \Delta n^k \|_2} \right) + \mathcal{N}(0, \varepsilon^2 P^2) \quad (9)$$
A New Noise Generating Method Based on Gaussian Sampling

Equation (9) suits for input noisy data, we update the noises by using \( n_{t+1} \), \( \sum_{k=0}^{m_t} \frac{\Delta n^k}{\max(1, \frac{\|\Delta n^k\|_2}{P})} + N(0, \varepsilon^2 P^2) \) is the sum of the updated noises, the sum of updated noises is denoted as \( \sum_{k=0}^{m_t} \frac{\Delta n^k}{\max(<\Phi(x), \Phi(y)>, \frac{\|\Delta n^k\|_2}{P})} \), \( <\Phi(x), \Phi(y)\> = K(x, y) \), \( N(\mathcal{N}, \varepsilon^2 P^2) \) is the noise scaled to \( P \).

Based on the sanitizer, we tackle the problem of privacy preservation by using the RKHS smoothing, RKHS means the reproducing kernel Hilbert spaces.

\[
\mathcal{H}(x, y) = \exp\left\{-\frac{|x - y|^2}{\delta}\right\} \tag{10}
\]

\[
X_i(t) = \mu(t) + \sum_{j=1}^{m} j^{-\frac{5}{2}} U_{ij} v_j(t) \tag{11}
\]

In Eq. (10) and Eq. (11), \( x \) and \( y \) represent input dataset in the Hilbert space \( \mathcal{H} \), \( \delta \) means the risk is fixed by choosing \( \varepsilon \) in the definition of differential privacy. All comparisons encompass privacy restrictions \( \varepsilon \) and other factors occurring on a grid of equal distances, in the RKHS kernel, \( \delta = 0.1 \). The parameter of Kernel \( \mathcal{H} \) is used to define \( \rho = 0.001 \) and the parameter \( x_i(t) \) is set at \( p = 4 \). The parameters of median function, sample size, and differential privacy are set to \( \mu(t) = 0.1 \sin(\pi t) \), \( N = 25 \), \( \varepsilon = 1.0 \), \( \delta = 0.1 \), respectively.

Algorithm 1 shows the procedure of our solution. The inputs include the size of sample data, loss function, and other parameters, such as learning rate, noise scale, and gradient bound. The process is split into three parts: From line 1 to line 7, the task of this part is for sampling input data and generating noises. The second part is for conducting SGD, the local minimum is attained and the noises are added during the time of seeking the local minimum by using the gradient descent. The last part is the process for outputting the results. The evaluation of the proposed solution is presented at the end of this algorithm.

4 Evaluations

The effectiveness of the noise generating method presented in Sect. 3 is evaluated through using the CIFAR-10 and CIFAR-100 datasets [9]. CIFAR-10 consists of a total of 60,000 small images with the size of 32 \( \times \) 32 pixels distributed in 10 classes. Each class contains 5,000 training and 1,000 test images. CIFAR-100 has the same size of images as CIFAR-10, the images are allocated in 100 classes, with 600 images belonging to each class. Therefore, there are 500 training and 100 test images per class.

In order to compare the new approach with SGD, we compare the learning rate with multiple training epochs. In the experiment, we take use of CNN with 60D PCA algorithm and CIFAR-10 dataset, 1,000 hidden units. The noise levels \( (\varepsilon, \delta) \) for \( (\varepsilon, \delta) \)-differential privacy in the neural network for PCA projection is set as \( (4, 7) \).
Algorithm 1. Gaussian Noise generating algorithm under the federated SGD

**Input:** Data source $D_1, D_2, \ldots, D_N$ and data size $N$, the parameters are: Loss function $L(\theta) = \frac{1}{N} \sum_i L(\theta, d_n)$, learning rate $\eta$, noise scale $\tau$ and gradient bound $G$.

**Output:** Training data $\Theta$, privacy cost $(\rho, \omega)$

1: procedure GENERATE NOISE(Obtain stochastic sample $D_n$ and parameter group $P_{new}(\eta, \tau, G)$ and replace previous parameters $P_{prev}(\eta, \tau, S, G)$)
2:    for $n \in [N]$ do
3:        Obtain parameter group $P_{global}(\eta, \tau, G)$ from $P_{prev}$
4:        Replace previous parameters and generate $P_n(\eta, \tau, G)$
5:        Fetch stochastic sample $D_n$ and probability $\frac{D_n}{N}$
6:        Test distribution of sample $D_n$
7:        Generate noise samples with the same distribution $D_n$
8:        for each $i \in L_t$ compute $g_t(d_i) \leftarrow \nabla_{\theta_t} L(\theta_t, d_n)$ do
9:            Clip gradients
10:              $\hat{g}_t(d_i) \leftarrow \frac{g_t(d_i)}{\max(1, \|g_t(d_i)\|_2)}$
11:            Add Gaussian Noises
12:              $\hat{g}_t \leftarrow \frac{1}{\sqrt{N}}(\Sigma_i \hat{g}_t(d_i) + \mathcal{N}(0, \eta^2 G^2 I))$
13:            Descent and output $\hat{g}_t$
14:              $\Theta_{t+1} \leftarrow \Theta_t - \eta \hat{g}_t$
15:        end for
16:    end for
17:    Sort gradients in $\hat{g}_t(d_i)$ and output largest $\Theta_{t+1}$
18:    Choose gradient sample $D_{n+1}$ inferior to the bound $G$
19: end procedure

4.1 Our Experiments for Comparing Learning Rates

Figure 2 (a) shows the comparison of accuracy rates, the $x$-axis indicates Root Mean Square Error (RMSE) and the $y$-axis represents accuracy rate. Compared with our proposed approach: Federated privacy-preserving SGD, the original SGD has better learning rate, but the advantage is not obvious, which reveals that the learning rate with the new federated SGD method is able to approach the original SGD if the number of training epochs grows, the approach asserts that the learning performance is acceptable. It also spots that the new approach is able to leverage the balance between privacy and learning, the deviation between our new approach and the original method is acceptable.

Figure 2(b) shows the experimental results related to learning rate during model training, we take use of the same environment, i.e., under CelebA [11] dataset and parameter setting. We compare the accuracy and learning rates from 0.05 to 0.14. In Fig. 2(b), $x$-axis shows learning rate and $y$-axis represents accuracy rate, the accuracy meets the peak point if the learning rate reaches 0.06. The trend declines smoothly.

Besides, the proposed privacy-preserving noise generating scheme with federated SGD (PPNGFSGD) tends to have better accuracy with original non-privacy
SGD approach if the learning rate is between 0.06 and 0.1. If the learning rate gets up to 0.1, the accuracy rate by using the new approach slashes faster than original SGD approach. All two drops are out of 0.9 if the learning rate is up to 0.14. The experiment shows the learning rate between 0.05 to 0.07 is the best one for our proposed approach. Compared with original SGD, the gap between the
new and former federated SGD is not clear. The new approach replaces original one and the deviation is controlled under an acceptable way.

In next experiment, we take use of CNN having 60D PCA projection and 5,000 hidden units with CelebA [11] dataset, which was trained by using the size 2,400 and clipping threshold 8.

4.2 Experiments for Gradient Clipping and Noise Levels

In Fig. 3, we have compared two noise generating models and observe the variation of accuracy rates by using the CelebA [9] dataset. The bottom model, named as Gaussian phase noise generating (GPNG) scheme was proposed by [15]. The second model, namely, differential privacy-preserving distributed stochastic gradient descent (DPDSGD) was developed by Abadi [1]. The privacy-preserving noise generating model with federated SGD (PPNGFSGD) is proposed in this paper. The last one is the distributed SGD without any process in privacy protection. As $y$-axis in Fig. 3, the clipping gradient means when the training data increases, the more noises will be added into the set, the training accuracy in $x$-axis will be changed correspondingly.

The clipping gradient degrades the integrity of the estimate as shown in this test, if the clipping parameters are too low, the average clipped pattern may appear in different directions based on the true pattern. On the other hand, raising the normal forces the gradients to add less noise as we did. In practice, a good way to select a value is to replace a mediator of the non-clipped gradient terms for the training, the training accuracy in $x$-axis will be altered correspondingly.

In Fig. 3, by adding less noises, each privacy deficit is proportionally smaller, thus, we have more on the accumulated privacy budget, which means how many noises will be added into the data, what distribution of the data is subjected by adding noises to the original data. The selection thus has a significant impact on accuracy.

For instance, if the normal increase, the less noises will affect on the methods PPNGFSGD and DPDSGD. As controlled, the accuracy will be increased by
using non-privacy SGD, whilst the clipping norm is climbing. But for GPNG, due to this method without control via noise level, the accuracy is lower than others.

In nutshell, if gradient clipping norm has less training data and less noises, the gap between original non-privacy SGDs will be less. While gradient clipping norm is being increased, the accuracy rates from our proposed method and non-noise-added SGD method are better than those of PPNGFSGD and GPNG having different gradient clipping norms. The average distance with original non-privacy SGD is the smallest one of our solution among the three privacy-preserving methods.

In Fig. 3, x-axis means noise level and y-axis is accuracy rate. At the low noise level, it shows the accuracy grows smoothly, on the high noise level, it drops very fast. Without privacy, the SGD approach tends to have a better accuracy than the new approach, but the distance between the two is not obvious. The experimental results show that the levels of inaccuracy privacy-preserving methods are acceptable, the deviation is acceptable under control.

Figure 4 reflects the results of noises added into the dataset [9]. Figure 4(a) is the original dataset without adding noises. Figure 4(b) was generated from our proposed method PPNGFSGD and Fig. 4(b) was from GPNG. As shown in Fig. 4(c), the CPNG is hard to find any information, but we still are able to view the shapes of human faces from the images 4(b), it is the reason why the results in Fig. 2 and Fig. 3 show higher accuracy rates than that of GPNG.

5 Conclusions

Our framework is applied to adaptive control based on well-trained weights and parameters, such as the lot size, the gradient norm boundary, and the noise level. Our experiments with multiple noise levels in model training show a trend of improvement compared with GPNG, it is interesting to consider more sophisticated schemes for aptly choosing these parameters. In the near future, our project will continue focusing on sampling approach so as to uplift the training efficiency.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016)
2. Agarwal, N., Suresh, A.T., Yu, F.X.X., Kumar, S., McMahan, B.: CPSGD: communication-efficient and differentially-private distributed SGD. In: Advances in Neural Information Processing Systems, pp. 7564–7575 (2018)
3. Aggarwal, C.C., Yu, P.S.: A general survey of privacy-preserving data mining models and algorithms. In: Aggarwal, C.C., Yu, P.S. (eds.) Privacy-Preserving Data Mining. Advances in Database Systems, vol. 34, pp. 11–52. Springer, Boston (2008). https://doi.org/10.1007/978-0-387-70992-5_2
4. Chaudhuri, K., Monteleoni, C.: Privacy-preserving logistic regression. In: Advances in Neural Information Processing Systems, pp. 289–296 (2009)

5. De Brabanter, J., De Moor, B., Suykens, J.A., Van Gestel, T., Vandewalle, J.P.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)

6. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14

7. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1

8. Hsieh, F.Y., Bloch, D.A., Larsen, M.D.: A simple method of sample size calculation for linear and logistic regression. Stat. Med. 17(14), 1623–1634 (1998)

9. Krizhevsky, A., Nair, V., Hinton, G.: The CIFAR-10 Dataset, vol. 55 (2014). http://www.cs.toronto.edu/kriz/cifar.html

10. Lee, S., Rao, R., Narasimha, R.: Characterization of self-similarity properties of discrete-time linear scale-invariant systems. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221), vol. 6, pp. 3969–3972. IEEE (2001)

11. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (2015)

12. Lou, Y., Yu, L., Wang, S., Yi, P.: Privacy preservation in distributed subgradient optimization algorithms. IEEE Trans. Cybern. 48(7), 2154–2165 (2017)

13. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: l-diversity: privacy beyond k-anonymity. ACM Trans. Knowl. Discov. Data (TKDD) 1(1), 3-es (2007)

14. Margineantu, D.D., Dietterich, T.G.: Pruning adaptive boosting. In: International Conference on Machine Learning, vol. 97, pp. 211–218. ICML (1997)

15. Richards, M.A.: Coherent integration loss due to white Gaussian phase noise. IEEE Sig. Process. Lett. 10(7), 208–210 (2003)

16. Shabtai, A., Elovici, Y., Rokach, L.: A Survey of Data Leakage Detection and Prevention Solutions. Springer, Boston (2012). https://doi.org/10.1007/978-1-4614-2053-8

17. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1310–1321 (2015)

18. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzzi- ness Knowl.-Based Syst. 10(05), 557–570 (2002)

19. Vapnik, V.: Principles of risk minimization for learning theory. In: Advances in Neural Information Processing Systems, pp. 831–838 (1992)

20. Whittle, P.: Estimation and information in stationary time series. Arkiv för matematik 2(5), 423–434 (1953)

21. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemom. Intell. Lab. Syst. 2(1–3), 37–52 (1987)

22. Wombacher, A.: Data workflow-a workflow model for continuous data processing. Data Process. (2010)

23. Zhu, Y., Liu, L.: Optimal randomization for privacy preserving data mining. In: Proceedings of the Tenth ACM SIGKDD, pp. 761–766 (2004)