Data Reconstruction Attack Against Principal Component Analysis

Saloni Kwatra(✉) and Vicenç Torra

Department of Computing Science, Umeå University, Umeå, Sweden
{salonik,vtorra}@cs.umu.se

Abstract. Attacking machine learning models is one of the many ways to measure the privacy of machine learning models. Therefore, studying the performance of attacks against machine learning techniques is essential to know whether somebody can share information about machine learning models, and if shared, how much can be shared? In this work, we investigate one of the widely used dimensionality reduction techniques Principal Component Analysis (PCA). We refer to a recent paper that shows how to attack PCA using a Membership Inference Attack (MIA). When using membership inference attacks against PCA, the adversary gets access to some of the principal components and wants to determine if a particular record was used to compute those principal components. We assume that the adversary knows the distribution of training data, which is a reasonable and useful assumption for a membership inference attack. With this assumption, we show that the adversary can make a data reconstruction attack, which is a more severe attack than the membership attack. For a protection mechanism, we propose that the data guardian first generate synthetic data and then compute the principal components. We also compare our proposed approach with Differentially Private Principal Component Analysis (DPPCA). The experimental findings show the degree to which the adversary successfully attempted to recover the users' original data. We obtained comparable results with DPPCA. The number of principal components the attacker intercepted affects the attack's outcome. Therefore, our work aims to answer how much information about machine learning models is safe to disclose while protecting users' privacy.

Keywords: Principal Component Analysis · Privacy · Data reconstruction attack · Membership Inference Attack · Generative Adversarial Networks

1 Introduction

It is well known that Machine Learning (ML) models can memorize the training data [5,6]. The more the accuracy of the ML models, the more is their ability

This study was partially funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

B. Arief et al. (Eds.): SocialSec 2023, LNCS 14097, pp. 79–92, 2023.
https://doi.org/10.1007/978-981-99-5177-2_5

to memorize [3]. Therefore, sharing such ML models leads to privacy violations. In order to share or deploy privacy-preserving machine learning models, it is important to understand how information leakage occurs and how much information ML models leak about individuals. For frameworks like Federated Learning (FL) [10], where the distributed devices share ML models trained on their local data with the aggregation server or with other distributed devices, knowing how much information machine learning models leak is an important question to address, especially when the ML models are trained on sensitive data. For example, medical data.

Different kinds of attacks are studied to evaluate the robustness of machine learning applications, including data poisoning attacks, model inversion attacks, and backdoor attacks. Membership Inference Attacks (MIA) [11] are the most relaxed attack, in the sense that it reveals minimal information about the individuals: whether or not a target sample is included in the training dataset on which the ML model was trained. MIA on medical data is harmful. For e.g., if the membership information is leaked from the ML model trained on Alzheimer's data. Data reconstruction attack lies at the other extreme of the information disclosure span. It is the most strict attack, as an adversary's successful data reconstruction attack can disclose all the information about an individual, which a machine learning model may have seen during its training.

In our work, we focus on Principal Component Analysis (PCA), which is a popular dimensionality reduction technique. In [15], the authors studied MIA against PCA, where the adversary intercepts some of the principal components and infer whether a particular sample participated in the computation of principal components. We show that the adversary can conduct a data reconstruction attack against PCA if the adversary intercepts some of the principal components obtained from the synthetic data generated using Conditional Tabular Generative Adversarial Network (CTGAN) [13]. Therefore, we show that even if the adversary has access to the principal components obtained from the synthetic data, which is considered safe for sharing, the adversary can attempt an extreme attack, like a data reconstruction attack, with considerable success. Differentially Private Principal Component Analysis (DPPCA) was already studied in the works [15], and [8]. In [15], the data curator adds Laplacian or Gaussian noise to the coefficients of the covariance matrix as a protection mechanism against privacy leakage. In our work, we generate a synthetic dataset before computing the principal components. The message of our work is as follows:

- In our work, we study the efficacy of synthetic datasets in combating attacks against machine learning models.
- If some of the principal components are leaked, we show that the membership attack (a less powerful attack) against PCA shown in work [15] can be converted into a more powerful attack, like a data reconstruction attack if the attacker has knowledge about the distribution of training data.
- From our work, we motivate to use protection mechanisms like generating synthetic data before training and sharing of machine learning/deep learning models.

- We also analyze the reconstruction attack's success when Differentially Private Principal Component Analysis (DPPCA) [8] is used.

The paper is organized as follows. Section 2 reviews some concepts needed in the rest of the paper. In particular, we discuss PCA, MIA against PCA, and CTGAN. Our suggested attack strategy is described in Sect. 3. Section 4 describes the approaches that were compared, including DPPCA. In Sect. 5, we provide and discuss the results. Section 6 gives the conclusion and future directions.

2 Related Work

2.1 Principal Component Analysis

Given a set $\mathcal{D} = \{x_n \in R^d : n = 1 : N\}$ comprising N raw data samples corresponding to N individuals of dimension d. After subtracting the mean from the data, we obtain the centered data matrix and denote it as X. The PCA technique aims to determine a p dimensional subspace that approximates each sample x_n [1]. The formulation of PCA is as follows:

$$\min_{\pi_p} E = \frac{1}{N}\sum_{n=1}^{N} E_n = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{N}||x_n - \pi_p x_n||_2^2 \tag{1}$$

where E is the average reconstruction error and π_p is an orthogonal projector, which approximates each sample x_n by $\hat{x_n}$. The solution to the PCA problem can be obtained via the Singular Value Decomposition (SVD) of a sample covariance matrix Σ_{cov}, a positive semi-definite matrix. Therefore, its singular value decomposition is equivalent to spectral decomposition. SVD of Σ_{cov} is given by $\sum_{i=1}^{d} \lambda_i v_i v_i^T$, where $\lambda_1 \geq \lambda_2 \ldots \lambda_d$ are the eigenvalues, and $v_1, v_2 \ldots v_d$ are the corresponding eigenvectors of Σ_{cov}, respectively. Let V_p denote the matrix whose columns are the top p eigenvectors. $\pi_p = V_p V_p^T$ is the solution to the problem in (1).

2.2 Membership Inference Attack Against Principal Component Analysis

Membership Inference attack (MIA) infers whether a particular record is part of the training dataset on which the machine learning algorithm was trained. The authors of [15] addressed MIA against PCA for the first time. In [15], the authors assume that the adversary intercepts some of the principal components. Using the intercepted principal components, the adversary computes the reconstruction error of the target sample (a sample whose membership is to be determined by the attacker). The concept is that the samples belonging to the training set will incur lower reconstruction error in comparison with the samples not belonging to the training set. Hence, on the basis of a tunable threshold value t, the adversary can distinguish between the members and the non-members of the training set.

It is quite interesting to know why membership attacks work. Previous works, including [2,4,9], and [14] identified that the overfitting of machine learning models is a reason behind the success of membership attacks. The overfitting of ML models is usually because of high model complexity and the limited size of the training dataset. Deep learning models are overparameterized and complex, which, on the one hand, enables them to learn successfully from big data, but, on the other hand, may cause them to have an unreasonably high capacity of retaining the noise or the details of a specific training dataset. Moreover, ML models are trained in a lot of (often tens to hundreds) epochs on the same instances repeatedly, making the training instances very susceptible to model memorization. Also, in [2], Bentley et al. presented a theorem that says that the overfitting of the target models can lead to the performance of an MIA better than randomly guessing (i.e., 50% attack success rate (ASR)).

2.3 Conditional Tabular Generative Adversarial Network (CTGAN)

GANs learn to generate fake samples that mimic the real ones. GANs have two neural networks. One is the generator, which generates new data, while the other is the discriminator, which aims to correctly classify the real and fake data. GANs face certain challenges when applied to tabular data, including the need to simultaneously model discrete and continuous columns, the multi-modal non-Gaussian values within each continuous column, and the imbalance in categorical columns. CTGAN [13] proposed two modifications to tackle the issues faced by GANs when applied to generate tabular data. The first problem that CTGANs solved is finding the representative normalization of continuous data. A discrete variable can be represented using one-hot encoding. For example, to represent the working days of a week, we can use one-hot encoding with five columns. Mondays can be represented as {1,0,0,0,0}. Tuesdays can be represented as {0,1,0,0,0}, and so on. However, when dealing with continuous data, it is challenging to represent all the information carried by the continuous variable. A continuous variable has multiple modes. Therefore by merely feeding the model the value of the continuous variable at our sample, we risk losing information, such as what mode the sample belongs to? and what is its importance within that mode? CTGAN proposed using mode-specific normalization to avoid losing information, which first fits a VGM (Variational Gaussian Mixture model) to each continuous variable. A Gaussian mixture model finds the optimal k Gaussians to represent the data through expectation maximization. To handle an imbalance in discrete columns, CTGANs designed a conditional vector *cond*, and a training-by-sampling technique. The conditional generator can generate synthetic rows conditioned on one of the discrete columns. Using training-by-sampling, the *cond* and training data are sampled according to the log frequency of each category. Therefore, CTGAN can explore all possible discrete values.

3 Proposed Work: Threat Model and Attack Methodology

In our attack setting, the data curator/guardian generates synthetic/fake data $\mathcal{D}_{syn}$ using different percentages (10%, 30%, 50%, 70%,100%) of samples from the original data $\mathcal{D}$. The synthetic data is generated using CTGAN, as described in Sect. 2.3. The curator then computes the principal components P_k of the synthetic data $\mathcal{D}_{syn}$, and sends these to a reliable party. We suppose that the attacker $\mathcal{A}$ intercepts some or all of the Principal Components (PCs) computed on the synthetic data by eavesdropping on the communication channel. The previous works regarding MIA are reviewed in [7], there are two kinds of knowledge useful for attackers to implement MIAs against ML models: knowledge of **data distribution**, and knowledge of **machine learning model/algorithm**, which learns about the patterns in the training data. Knowledge of training data refers to the knowledge of the data distribution, which means that the attacker has access to the shadow dataset, which has the same distribution as the original data. This is a reasonable assumption, as the attacker can obtain the shadow dataset using statistics-based synthesis when the data distribution is known and model-based synthesis when the data distribution is unknown [11]. Hence, in our attack setting, we assume that the attacker can synthesize the shadow dataset using CTGAN. By knowing some of the principal components, and the constructed shadow dataset using CTGAN, the attacker can make a data reconstruction attack as follows.

Suppose we have an original data matrix X_{orig} of size $n \times p$. We obtain a data matrix X, after subtracting the mean vector μ from each row of X_{orig}. Let V be the $p \times k$ matrix of some k eigenvectors to reduce the dimension; these would most often be the k eigenvectors with the largest eigenvalues. Then the $n \times k$ matrix of PCA projection scores (Z) will be given by $Z = XV$. In order to be able to reconstruct all the original variables from a subset of principal components/eigenvectors, we can map it back to p dimensions with V^T. The result is then given by $\hat{X} = ZV^T$. Since we have a projection scores matrix, $Z = XV$, we obtain $\hat{X} = XVV^T$. We do not have access to the original data X; we assume that the attacker has knowledge about the distribution of X. Therefore, the attacker can synthesize the data X_{syn} with a similar distribution as X, and reconstruct the original data using $\hat{X} = ZV = X_{syn}V^TV$. We assume the attacker can access the synthetic data generated using the Conditional Tabular Generative Adversarial Network (CTGAN) to show experimental results. We generate the synthetic data using different percentages of records from the original data, including {10%, 30%, 50%, 70%, 100%}. To show the degree of success of the data reconstruction attack, we show the Reconstruction Accuracy (R.A.) in estimating the original data. We define R.A. as follows.

Definition 1. *Suppose R is the reconstructed data, which is the estimator for the original data O, where $R = \{R_1 \dots R_d\}$, and $O = \{O_1, \dots O_d\}$. Let δ be a reconstruction error, which can be tolerated to measure the level of reconstruction for a record. The reconstruction accuracy, R.A. is defined as follows:*

$$R.A. = \frac{\#\left\{\hat{R_j} : |\frac{Oj-R_j}{O_j}| \leq \delta, j = 1 \ldots d\right\}}{n} \qquad (2)$$

where # means count, and n is the number of records. Hence, R.A. is the percentage of reconstructed entries for which the relative errors are within δ. The diagram of our proposed attack is shown in Fig. 1, which explains our methodology.

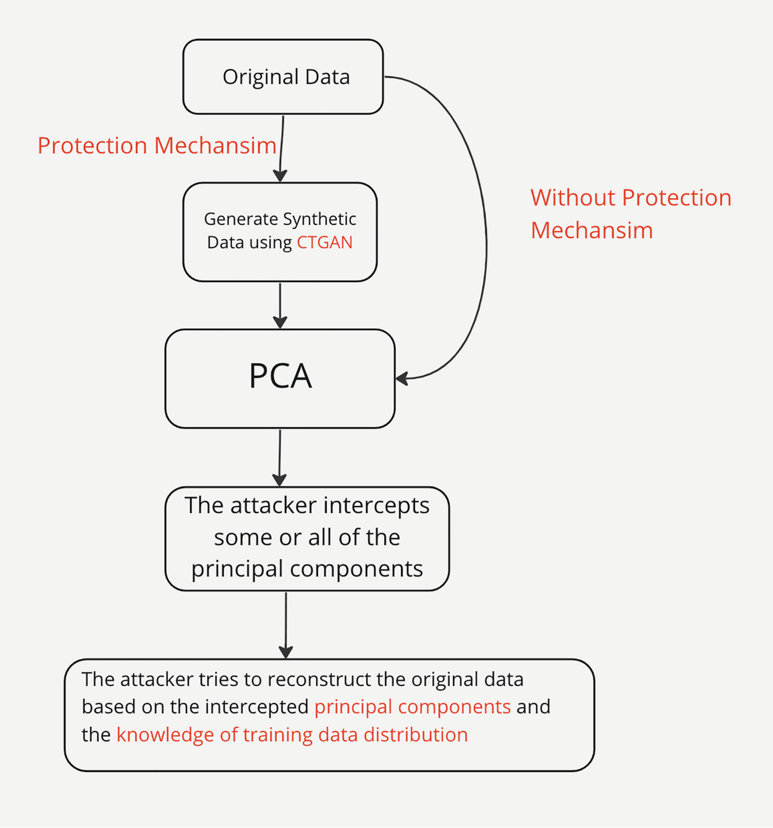

Original Data
Protection Mechansim
Generate Synthetic Data using CTGAN
Without Protection Mechansim
PCA
The attacker intercepts some or all of the principal components
The attacker tries to reconstruct the original data based on the intercepted principal components and the knowledge of training data distribution

Fig. 1. Data reconstruction attack against Principal Component Analysis

4 Compared Methodologies

We compared our approach with two alternative strategies. In one strategy, we use no protection mechanism before computing the principal components. In the other strategy, we use Differentially Private Principal Component Analysis (DPPCA) for computing the principal components. In this section, we describe these alternative strategies. The results are presented in Sect. 5.

4.1 No Protection Mechanism

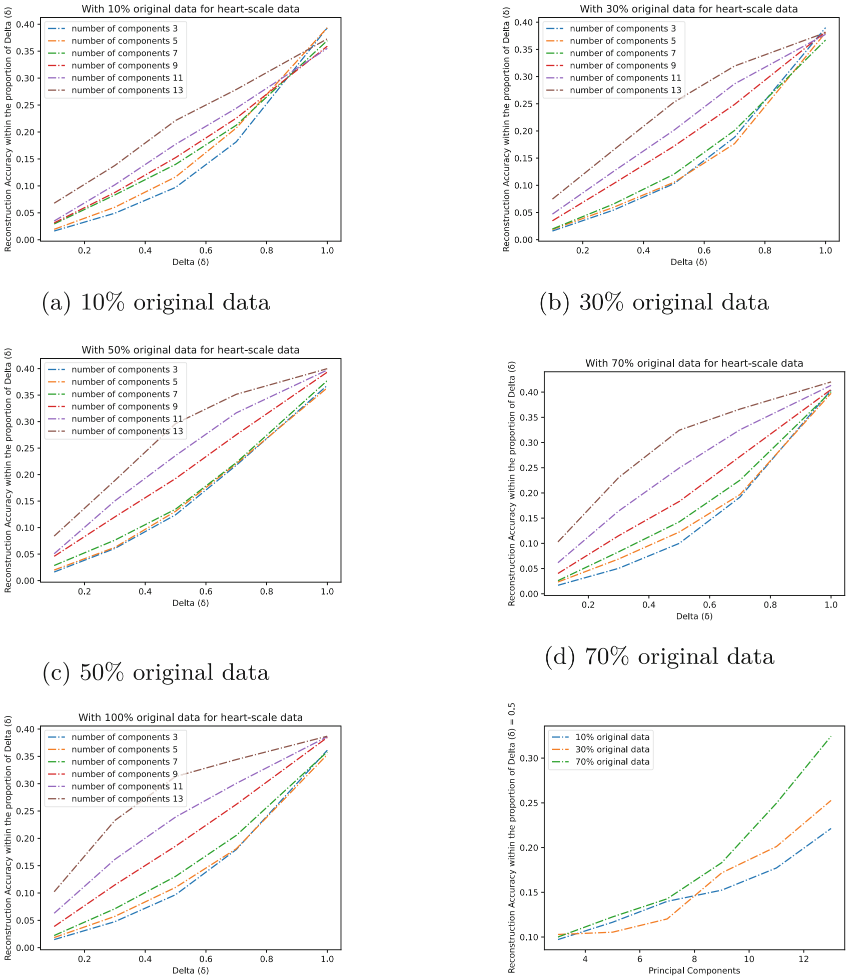
We first compare our proposed methodology with the case when the data curator uses no protection mechanism at all, computes the principal components of the original data, and shares these principal components with a reliable third party. Nevertheless, the attacker eavesdrops on the communication channel and obtains some or all of the principal components. Based on the knowledge of the training data distribution and the intercepted principal components, the attacker tries to reconstruct the original data of users. To be noted, the difference between our proposed methodology and the compared methodology is that in the proposed methodology, the principal components computed on the synthetic data are leaked, and in the compared methodology, the principal components computed on the original data are leaked.

4.2 Differentially Private Principal Component Analysis

The goal of PCA is to find the principal components of a dataset, which are the directions in which the data varies the most. In [8], the authors proposed a new approach to perform differentially private PCA (DPPPCA) on high-dimensional datasets. The algorithm in this paper involves perturbing the covariance matrix of the dataset in a differentially-private manner to ensure that the PCA output is also differentially-private. Specifically, the algorithm takes as input a dataset X with n samples and d dimensions and a privacy parameter ϵ. It then computes the covariance matrix S of the dataset, which is a $d \times d$ symmetric matrix. To perturb the covariance matrix while maintaining privacy, the algorithm adds a noise matrix N to S, where N is also a symmetric matrix. The noise matrix is generated using the Laplace mechanism, which adds independent Laplace noise to each entry of N, scaled by the privacy parameter ϵ. The algorithm then performs eigendecomposition on the perturbed covariance matrix $S+N$ to obtain the principal components of the dataset. The eigendecomposition is performed using a numerical algorithm, such as the power iteration method. Finally, the algorithm outputs the top k principal components of the dataset, where k is a user-specified parameter. The output is also differentially-private, as the added noise ensures that the output does not reveal information about any individual sample in the dataset. They also provide theoretical bounds on the privacy loss and the accuracy of the method.

Table 1. Description of datasets

Dataset	Number of Samples	Number of Attributes
Heart-scale	270	13
a9a	32561	123
Mushrooms	8124	112



(a) 10% original data

(b) 30% original data

(c) 50% original data

(d) 70% original data

(e) 100% original data

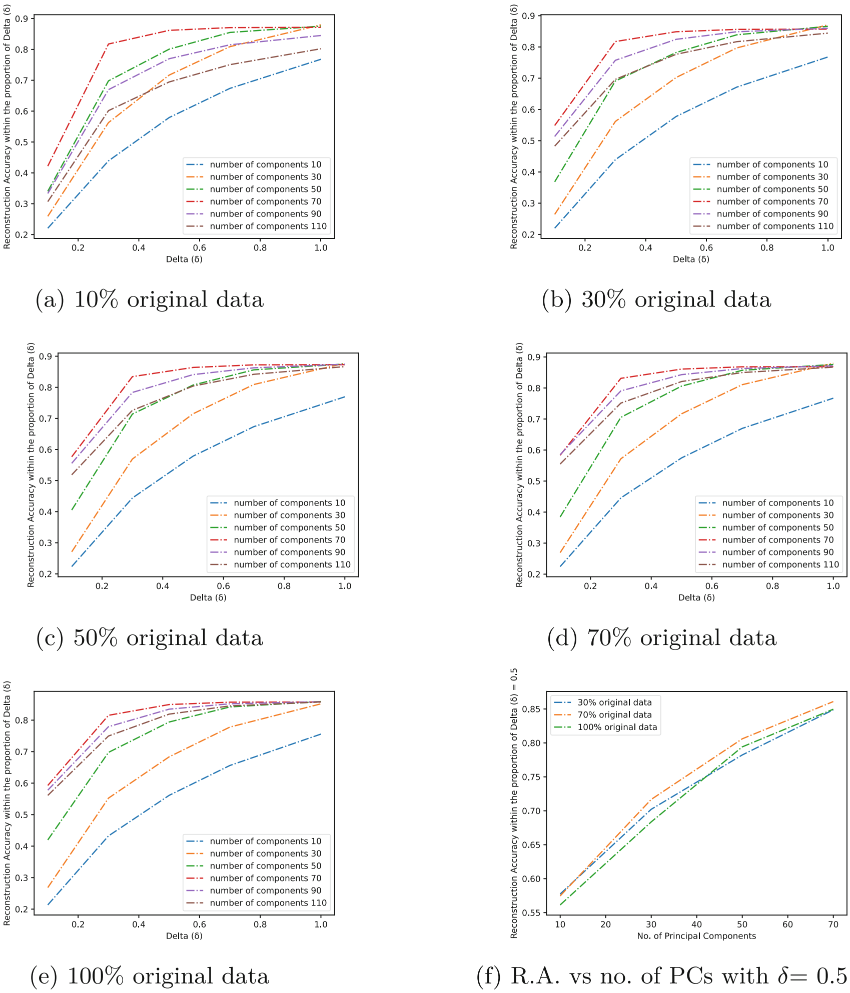
(f) R.A. vs no. of PCs with $\delta = 0.5$

Fig. 2. R.A. within the limit Delta (δ) for heart-scale data

5 Experimental Results and Analysis

We experimented on three publicly available binary classification datasets: Heart-scale, a9a, and mushrooms. The datasets can be found on[1]. The number of samples and the number of attributes of these datasets are described in Table 1. It can be seen that the range for the number of samples is from 270 to 32,561, and the number of attributes is from only 13 to 123. Each dataset has some preprocessing steps involved. The scale for the heart-scale dataset is

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html.



(a) 10% original data

(b) 30% original data

(c) 50% original data

(d) 70% original data

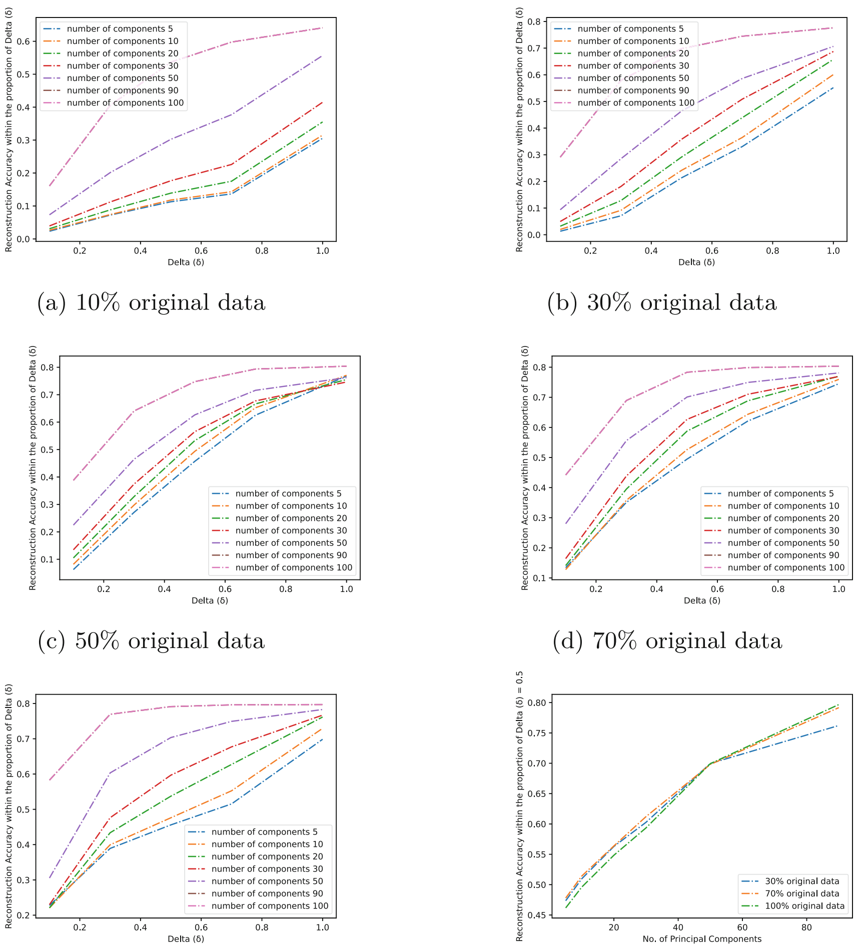
(e) 100% original data

(f) R.A. vs no. of PCs with δ= 0.5

Fig. 3. R.A. within the limit Delta (δ) for a9a data

[−1,1]. After preprocessing, the adult dataset is converted into the a9a dataset. There are 14 features in the original adult data set, eight of which are categorical and six of which are continuous. The continuous features in this data set are discretized into quantiles, and a binary feature represents each quantile. In addition, a categorical feature with m categories is converted to m binary features. In the mushrooms dataset, each nominal attribute is expanded into several binary attributes. Also, the original attribute 12 has missing values and is not used.

In our experiments, we generated synthetic datasets using different percentages of original data. We apply PCA to the generated synthetic datasets. Assuming that the adversary intercepted some of these principal components, we try to reconstruct the data from which the principal components were computed.



(a) 10% original data

(b) 30% original data

(c) 50% original data

(d) 70% original data

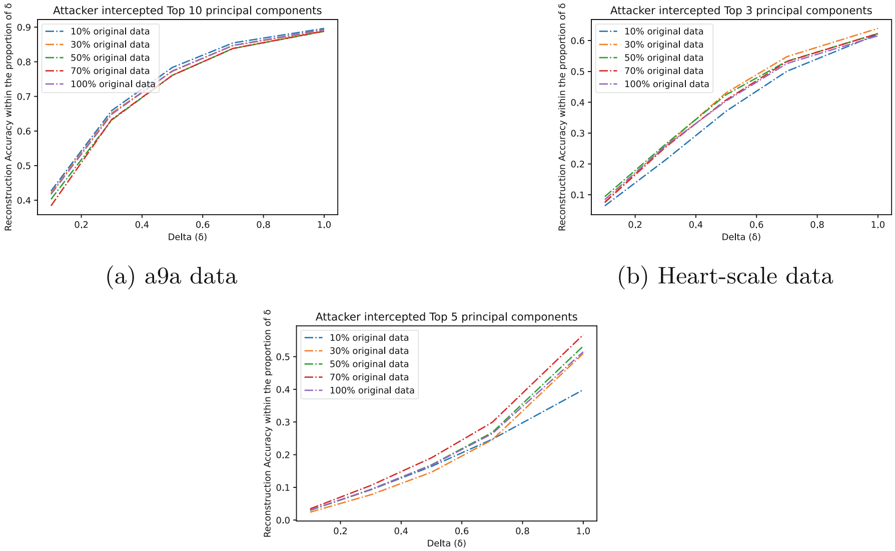
(e) 100% original data

(f) R.A. vs no. of PCs with $\delta = 0.5$

Fig. 4. R.A. within the limit Delta(δ) for mushrooms data

We obtain the reconstruction accuracy, as shown in Fig. 2, 3, and 4, respectively. We have an upper cap for the Reconstruction Accuracy (R.A.), as the maximum reconstruction error we can obtain is the difference between the original and synthetic data generated using CTGAN using all the original data records. We are measuring the capability of CTGANs to generate a different-looking but similar distribution of synthetic data and the privacy breach caused by the leakage done by the principal components. We summarize our main findings as follows.

1. We found that even after using just 10% samples from the original data, the R.A. is close to 90% when the attacker intercepted 110 principal components.



(a) a9a data

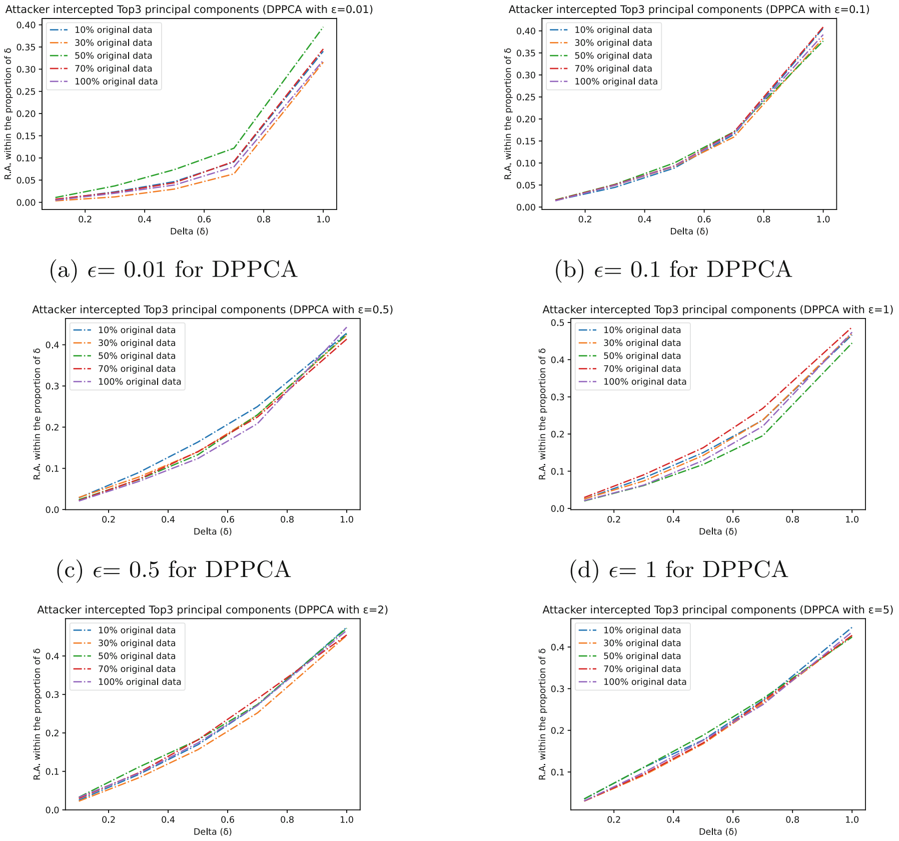
(b) Heart-scale data

(c) Mushrooms data

Fig. 5. R.A. without protection mechanism prior to the computation of PCs

R.A. is close to 70% when the attacker intercepted 10 principal components, as shown in Fig. 3a for the a9a dataset.

2. For the a9a dataset, the R.A. is more in comparison with the heart-scale data in Fig. 2 and mushrooms data in Fig. 4 dataset. The reason behind more R.A. in the case of the a9a dataset is that a9a has more categorical features. Hence, the generation of synthetic data using CTGAN could provide less protection in the case of the a9a dataset.
3. The maximum R.A. for heart-scale data, as shown in Fig. 2 is close to 40%. It is less because we have a protection mechanism using synthetic data generation before the computation of principal components.
4. The minimum reconstruction in the case of mushroom data in Fig. 4a is close to 20% when the attacker intercepted 5 or 10 principal components and only 10% of the original data was used in constructing the synthetic data.
5. In Figs. 2f, 3f, and 4f for heart-scale, a9a, and mushrooms dataset, respectively, we show a trend between R.A. and the number of principal components intercepted by the attacker. Our results show that R.A. increases as the number of principal components increases, which is also expected from theory.
6. We generated synthetic datasets from different percentages of original data. From Figs. 3a to 3e, we observe that as we increase the percentage of samples used in generating the synthetic data, the gap between the lines for R.A. in the graph widens, indicating the increase in R.A. with the increase in the



(a) ϵ= 0.01 for DPPCA

(b) ϵ= 0.1 for DPPCA

(c) ϵ= 0.5 for DPPCA

(d) ϵ= 1 for DPPCA

(e) ϵ= 2 for DPPCA

(f) ϵ= 5 for DPPCA

Fig. 6. R.A. using DPPCA on heart-scale data when the attacker intercepted Top 3 PCs

percentage of samples used from original data for generating synthetic data using CTGAN.

7. It is noted that there is not much difference in the R.A. when the CTGAN uses less percentage (e.g., 10%) of samples from the original data compared to using all the samples from the original data for generating the synthetic data. This shows the capability of CTGAN in successfully generating synthetic data similar to the original data using fewer samples from the original data.
8. When no protection mechanism is used, we show that the R.A. increases. For e.g., in Fig. 5b, the R.A. for the heart-scale data approaches 60%, which is higher in comparison with the case when DPPCA is used (Refer Fig. 6), and when the principal components were computed on the synthetic data (Refer Fig. 2).

9. In Fig. 6, we use DPPCA on the heart-scale data. We observe that the lesser the value of ϵ, the shallower the graph for R.A.
10. Both DPPCA and the generation of synthetic data technique outputs comparable R.A. The performance of DPPCA depends on the value of a privacy parameter ϵ. The lower the value of ϵ, the higher the privacy.
11. Therefore, from our experiments, we can conclude that generating synthetic data from the original data and then training machine learning models on the synthetic data is a good way to combat attacks against machine learning models to an extent.

6 Conclusion and Future Works

We proposed a data reconstruction attack against PCA by extending the work related to membership inference attacks in [15] and [7]. Specifically, we made two assumptions for attempting a data reconstruction attack against PCA; one is that the attacker knows some of the principal components computed on the synthetic dataset generated by the data curator, and the other is that the attacker has knowledge about the data distribution. Knowing that the data reconstruction attack is more harmful than the membership attack, we obtained reasonably good results in terms of reconstruction accuracy. We studied the efficacy of synthetic datasets generated using Conditional Tabular Generative Adversarial Networks as a protection mechanism in combating data reconstruction attacks. In the future, we would like to explore the behavior of other machine learning models against MIA and data reconstruction attacks. In the work [12], it is shown that synthetic data cannot protect the outlier records but performs well in terms of utility, whereas DP synthetic data provides high privacy gains but at the cost of degrading the utility of data. Hence, we would also like to conduct the privacy and utility analysis of synthetic and DP synthetic datasets.

References

1. Abdi, H., Williams, L.J.: Principal component analysis. Wiley Interdiscip. Rev. Comput. Stat. **2**(4), 433–459 (2010)
2. Bentley, J.W., Gibney, D., Hoppenworth, G., Jha, S.K.: Quantifying membership inference vulnerability via generalization gap and other model metrics. arXiv preprint arXiv:2009.05669 (2020)
3. Brown, G., Bun, M., Feldman, V., Smith, A., Talwar, K.: When is memorization of irrelevant training data necessary for high-accuracy learning? In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, pp. 123–132 (2021)
4. Farokhi, F., Kaafar, M.A.: Modelling and quantifying membership information leakage in machine learning. arXiv preprint arXiv:2001.10648 (2020)
5. Feldman, V.: Does learning require memorization? A short tale about a long tail. In: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, pp. 954–959 (2020)

6. Feldman, V., Zhang, C.: What neural networks memorize and why: discovering the long tail via influence estimation. Adv. Neural. Inf. Process. Syst. **33**, 2881–2891 (2020)
7. Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S., Zhang, X.: Membership inference attacks on machine learning: a survey. ACM Comput. Surv. (CSUR) **54**(11s), 1–37 (2022)
8. Imtiaz, H., Sarwate, A.D.: Symmetric matrix perturbation for differentially-private principal component analysis. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2339–2343. IEEE (2016)
9. Jha, S.K., et al.: An extension of Fano's inequality for characterizing model susceptibility to membership inference attacks. arXiv preprint arXiv:2009.08097 (2020)
10. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
11. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 3–18. IEEE (2017)
12. Stadler, T., Oprisanu, B., Troncoso, C.: Synthetic data-anonymisation groundhog day. In: 31st USENIX Security Symposium (USENIX Security 2022), pp. 1451–1468 (2022)
13. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional GAN. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
14. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: analyzing the connection to overfitting. In: 2018 IEEE 31st Computer Security Foundations Symposium (CSF), pp. 268–282. IEEE (2018)
15. Zari, O., Parra-Arnau, J., Ünsal, A., Strufe, T., Önen, M.: Membership inference attack against principal component analysis. In: Domingo-Ferrer, J., Laurent, M. (eds.) PSD 2022. LNCS, vol. 13463, pp. 269–282. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-13945-1_19