Differentially Private Generative Adversarial Networks for Time Series, Continuous, and Discrete Open Data

Lorenzo Frigerio, Anderson Oliveira, Laurent Gomez, Patrick Duverger

Differentially Private Generative Adversarial Networks for Time Series, Continuous, and Discrete Open Data

Lorenzo Frigerio[1], Anderson Santana de Oliveira[2], Laurent Gomez[2], and Patrick Duverger[3]

[1] Polytech Nice
[2] SAP Labs France, Mougins, France
[3] Ville d'Antibes, Antibes, France

Abstract. Open data plays a fundamental role in the 21st century by stimulating economic growth and by enabling more transparent and inclusive societies. However, it is always difficult to create new high-quality datasets with the required privacy guarantees for many use cases. In this paper, we developed a differential privacy framework for privacy preserving data publishing using Generative Adversarial Networks. It can be easily adapted to different use cases, from the generation of time-series, to continuous, and discrete data. We demonstrate the efficiency of our approach on real datasets from the French public administration and classic benchmark datasets. Our results maintain both the original distribution of the features and the correlations among them, at the same time providing a good level of privacy.

Keywords: Differential Privacy · Generative Adversarial Networks.

1 Introduction

The digital revolution has changed societies and democracies across the globe, making personal data an extremely valuable asset. In such a context, protecting individual privacy is a key need, especially when dealing with sensitive information, such as political preferences. At the same time, the demand for public administration transparency has introduced guidelines and laws in some countries to release open datasets.

To limit personal data breaches, privacy-preserving data publishing techniques can be employed. This approach aims at adding the noise directly to the data, not only to the result of a query (like in interactive settings). The result is a completely new dataset, where analysts can perform an infinite number of requests without increasing the privacy costs, nor disclosing private information. Meanwhile, it is difficult to preserve the utility of the data.

A strong standard privacy guarantee widely accepted by the research community is differential privacy. It ensures that each individual participating in a database does not disclose any additional information by participating in it. Traditionally, many approaches tried to reach differential privacy by adding noise to the data in order to protect personal information [8, 5, 12], however they have never been able to provide satisfying results on real semantically-rich data; most of the implementations were limited to very specific purposes such as histogram queries or counting queries [18]. Generative models represent the most promising approach in this field. Interesting results have been obtained through Generative Adversarial Networks (GANs) [9]. These models are able to generate new samples coming from a given distribution. The advantage of generative models is that the noise to guarantee privacy is not added directly to the data, causing a significant loss of information, but it is added inside the latent space, reducing the overall information loss, but guaranteeing meanwhile privacy.

This paper extends the notion of dp-GAN, an anonymized GAN with a differential privacy mechanism, to handle continuous, categorical and time-series data. It introduces an optimization called clipping decay that improves the overall performances. This new expansion shapes the noise addition during the training. This allows to obtain a better data utility at the same privacy cost. A set of analysis on real scenarios evidence the flexibility and applicability of our approach, which is supported by an evaluation of the membership inference attack accuracy, proving the positive effects of differential privacy. We provide experimental results on real industrial datasets from the French public administration and over well-known publicly available datasets to allow for benchmarking.

The remainder of the paper is organized as follows: Section 2 provides the theoretical background for the paper; In Section 3, presents our framework for anonymization, together with the mathematical proofs of differential privacy. Section 4 provides a set of experiments on diverse use cases to highlight the flexibility and effectiveness of the approach. Section 5 discusses related work; and finally Section 6 concludes the paper.

2 Preliminaries

This section brings some important background for the paper.

2.1 Generative Adversarial Networks.

GANs (Generative Adversarial Networks) are one of the most popular type of generative models, being already defined as the most interesting idea in the last 10 years in machine learning[4], moreover, a lot of attention has been given to the development of new variations [3, 14, 11]. Given an initial dataset, a GAN is able to mimic its data distribution, for that, a GAN employs two different networks: a generator and a discriminator. The architecture of the two networks is separate from the definition of GAN; depending on the application, different network configurations can be used. The role of the generator is to map random noise into new data instances capturing the original data distribution. On the

[4] “GAN and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion”, Yann LeCun.

opposite side, the discriminator tries to distinguish the generated samples from the real ones estimating the probability that a sample comes from the training data rather than the generator. In this way, after each iteration, the generator becomes better at generating realistic samples, while the discriminator becomes increasingly able to tell apart the original data from the generated ones. Since the two networks play against each other, the two losses will not converge to a minimum like in a normal training process but this minmax game has its solution in the Nash equilibrium. Nevertheless, the vanilla GAN [9] suffers from several issues that make it hardly usable especially for discrete data. A new definition of the loss function, Wasserstein generative adversarial network (WGAN) [2] and Improved Training of Wasserstein GANs [10] partially solved this problem. We are going to use this latest loss function to train our dp-GAN models.

2.2 Differential privacy.

The state of the art anonymization technique is differential privacy. This concept ensures that approximately nothing can be learned about an individual whether she participates or not in a database. Differential Privacy defines a constraint on the processing of data so that the output of two adjacent databases is approximately the same. More formally: A randomized algorithm M gives (ϵ,δ)-differential privacy if, for all databases d and d', differing on at most one element and all $S \in Range(M)$,

$$Pr[M(d) \in S] \leq exp(\epsilon) \times Pr[M(d') \in S] + \delta. \tag{1}$$

This condition encapsulates the crucial notion of indistinguishability of the results of a database manipulation by introducing the so-called privacy budget ϵ. It represents the confidence that a record was involved in a manipulation of the database. Note that the smaller ϵ is, the more private the output of the mechanism. According to [6] the optimal value of δ is less than the inverse of any polynomial in the size of the database. Any function M that satisfies the Differential Privacy condition can be used to generate data that guarantees the privacy of the individuals in the database. In the non-interactive setting, a mechanism M is a function that maps a dataset in another one. The definition states that the probability of obtaining the same output dataset from M is similar, using either d or d' as input for the mechanism. Composability is an interesting property of Differential Privacy. If M and M' are ϵ and ϵ'-differential private respectively, their composition $M \circ M'$ is (ϵ+ϵ')-differentially private [5]. This property allows to craft a variety of mechanisms and combinations of such mechanisms to achieve differential privacy in innovative ways.

Concerning deep learning, Abadi et al. [1] developed a method to train a deep learning network involving differential privacy. This method requires the addition of a random noise, drawn from a normal distribution, to the computed gradients, to obfuscate the influence that an input data can have on the final model.

As for any anonymization methods, one must assess the likelihood of membership inference attacks. This kind of attack evaluates how much a model behaves differently when an input sample is part of the training set rather than the validation set. Given a machine learning model, a membership inference attack uses the trained model to determine if a record was part of the training set or not. In the case of generative models such as the one of GAN, a high attack accuracy means that the network has been able to model only the probability distribution of the training set and not the one of the entire population. This kind of attack has been proven to be effective especially when overfitting is relevant [17].

3 The framework

In our framework proposed we assume a trusted curator interested in releasing a new open dataset with privacy guarantees to the users present in it. Outside the trusted boundary, an analyst can use the generator model, result of our algorithm, to perform an indefinite number of queries over the data the generator produces. Such outputs can be eventually released as open data. Even by combining the generated data with other external information, without ever having access to the original training data, the analyst would not be able to violate the privacy of the information, thanks to the mathematical properties of differential privacy.

The dp-GAN model is constituted by two networks, a generator and a discriminator, that can be modelled based on the application domain. We adopted Long Short Term Memories (LSTM) inside the generator to model streaming data and multilayer perceptron (MLP) to model discrete data. In addition, to manage discrete data, we also used a trick that does not influence the training algorithm but it changes the architecture of the generator network. Specifically, an output is created for each possible value that a variable can assume and a softmax layer is added for each variable. The result of the softmax layer becomes the input of the discriminator network. Indeed, each output represents the probability of each variable instance; the discriminator compares these probabilities with the one-hot encoding of the real dataset. On the contrary, the output nodes associated with continuous variables are kept unchanged.

At the end of the training, the generator network can be publicly released; in this way, the analyst can generate new datasets as needed. Moreover, since the generator only maps noise into new data the process is really fast and data can be generated on the fly when a new analysis is required.

We used the differentially private Stochastic Gradient Descent (dp-SGD) proposed by [1] to train the discriminator network and the Adam optimizer to train the generator. The dp-GAN implementation relies on a traditional training in which the gradients computed for the discriminator are altered. This due to the fact that we want to limit the influence that each sample has on the model. On the contrary, the training of the generator remains unaltered; indeed, this network bases its training only on the loss of the discriminator without accessing directly the data.

The approach is independent from the chosen loss and therefore can be applied to the vanilla GAN implementation [9] but also to the improved WGAN one. The dp-SGD works as follows: once the gradients are calculated, it clips them by a threshold C and alter them by the addition of a random noise with variance proportional to C. Each time an iteration is performed, the privacy cost increases and the objective is to find a good balance between data utility and privacy costs.

Our implementation is an extension to the improved WGAN framework combining it with the dp-SGD. Therefore, the loss functions are calculated as in a normal WGAN implementation, except that the computed gradients are altered to guarantee privacy. Moreover, for the first time up to our knowledge, the dp-GAN concept is adapted to handle discrete data. Algorithm 1 describes our training procedure.

Algorithm 1 Algorithm for training a GAN in a differentially private manner

```
Input: Samples from $x_1$ to $x_N$, group size $L$, number of samples $N$, clipping parameter $C$, noise scale $\sigma$, privacy target $\varepsilon$, number of iterations of the discriminator per each iteration of the generator $Ndisc$, batch size $b$, Wasserstein distance $\mathcal{L}$, learning rate $\eta$, number of discriminator' s parameters $m$, clipping decay $C_{decay}$.
Output: differentially private Generator G
Initialize weights randomly both for the Generator $\theta_{G(0)}$ and the discriminator $\theta_{D(0)}$
Convert discrete variables into their One-Hot encodings
while (While privacy cost $\leq \varepsilon$) do
    for $t = 0$ to $Ndisc$ do
        for $j = 0$ to $b$ do
            sample $L_{\mathrm{t}}$ with sample probability $L/N = q$
            For each $x_i$ in $L_{\mathrm{t}}$, compute $g_{\mathrm{t}}(x_{\mathrm{i}}) \leftarrow \nabla_\theta \mathcal{L}(\theta_{\mathrm{t}}, x_i)$    ▷ Compute gradient
            $g_{\mathrm{t}}(x_{\mathrm{i}}) \leftarrow g_{\mathrm{t}}(x_{\mathrm{i}})/max(1, \|g_{\mathrm{t}}(x_{\mathrm{i}})\| /C)$    ▷ Clip gradient
            $g_{\mathrm{t}} \leftarrow \frac{1}{L}(\sum_{i=0}^{L} g_{\mathrm{t}}(x_{\mathrm{i}}) + N(0, (\sigma * C)^2 I))$    ▷ Add noise
            $\theta_{\mathrm{D(t+1)}} \leftarrow \theta_{\mathrm{D(t)}} - \eta * g_{\mathrm{t}}$    ▷ Gradient descent
        end for
    end for
    $C$ *= $C_{decay}$    ▷ Clipping decay
    Update the overall privacy cost $\varepsilon$    ▷ Moment accountant
    Sample m values $\mathrm{z_i} \sim$ Random noise    ▷ Sample random noise
    $\theta_{G(t+1)} \leftarrow Adam(\nabla_\theta \frac{1}{m} \sum_{i=0}^{m} -D(G(z_{\mathrm{i}})))$    ▷ Update Generator
end while
return G
```

3.1 Clipping decay

The role of the clipping parameter is to limit the influence that a single sample can have on the computed gradients and, consequently, on the model. Indeed, this parameter does not influence the amount of privacy used. A big clipping parameter allows big gradients to be preserved at the cost of a noise addition with a proportionally high variance. On the contrary, a small clipping parameter limits the range of values of the gradients, but it keeps the variance of the noise small. The bigger the clipping parameter the bigger the gradients' variance. Similarly to what it is done with the learning rate, it is possible to introduce a clipping parameter decay. In this way, the gradients not only tend to descend over time to better reach a minimum but, in addition, they mimic the descending trend of the gradients allowing to clip the correct amount at each step. In fact, when the model tends to converge to the solution, the gradients decrease. Therefore, the noise may hide the gradients if its variance is kept constant. By reducing the clipping parameter over time, it is possible to reduce the variance in the noise in parallel with the decrease of the gradients, thus improving the convergence of the model. This without influencing the overall privacy costs that are not altered by the clipping parameter but only by the amount of noise added.

3.2 Moment accountant

A key component of the dp-GAN is the moment accountant. It is a method that allows to compute the overall privacy costs by calculating the cost of a single iteration of the algorithm and cumulating it with the other iterations. Indeed, thanks to the composability property of differential privacy it is possible to cumulate the privacy costs of each step to compute the overall privacy cost. Given a correct value of σ and thanks to weights clipping and the addition of noise, Algorithm 1 is $(O(\epsilon, \delta))$-DP with respect to the lot. Since each lot is sampled with probability $q = L/N$, each iteration is $(O(q\epsilon, q\delta))$-DP. In the formula, q represents the sampling probability (the number of samples inside a lot divided by the total number of samples present in the dataset). The clipping decay optimization has no influence on the moment accountant. Indeed, it alterates only the clipping parameter and not the variance of the noise that is the variable that influences the cost of a single iteration by changing the value of ϵ. Each time a new iteration is performed the privacy costs increase. However, thanks to the definition of moment accountant, these costs do not increase linearly. Indeed, by cumulating the privacy costs for each iteration, an overall level of $(O(q\epsilon\sqrt{T}), \delta)$-DP is achieved where T represents the number of steps (the number of epochs divided by q).

The moment accountant is based on the assumption that the composition of Gaussian mechanisms is being used. Assessing that a mechanism M is $(O(\epsilon, \delta))$-DP is equivalent to a certain tail bound on M's privacy loss random variable. The moment accountant keeps track of a bound on the moments of the privacy loss random variable defined as:

$$c(o; M, aux, d, d') = log \frac{Pr[M(aux, d) = o]}{Pr[M(aux, d') = o]} \tag{2}$$

In (2) d and d' represent two neighbouring databases, M the mechanism used, aux an auxiliary input and o an outcome. What we are computing are the log moments of the privacy loss random variable that can be cumulated linearly. In order to bound this variable, since the approach is the sequential application of the same privacy mechanism we can define the λ^{th} moment $\alpha M(\lambda, aux, d, d')$ as the log of the moment generating function evaluated at the value λ:

$$M(\lambda, aux, d, d') = logE_{\mathrm{o} \sim \mathrm{M(aux,d)}}[exp(\lambda c(o, M, aux, d, d'))]. \tag{3}$$

And consequently we can bind all possible $\alpha M(\lambda, aux, d, d')$. We define

$$\alpha M(\lambda) = max_{aux,d,d'} \alpha M(\lambda, aux, d, d') \tag{4}$$

Theorem 1. *Using the definition* (4) *then* $\alpha M(\lambda)$ *has the following characteristics: given a set of* k *consecutive mechanisms, for each* λ*:*

$$\alpha_M(\lambda) \leq \sum_{i=1}^{k} \alpha M_i(\lambda)$$

for any $\epsilon > 0$*, a mechanism* M *is* (ϵ, δ)*-differentially private for*

$$\delta = min_{\lambda} exp(\alpha_M(\lambda) - \lambda * \epsilon)$$

Proof (Proof of Theorem 1). A detailed proof of Theorem 1 can be found in [1].

Theorem 2. *Algorithm 1 is* $(O(q\epsilon\sqrt{T}), \delta)$*-differentially private for appropriately*[5] *chosen settings of the noise scale and the clipping threshold.*

Proof (Proof of Theorem 2). By Theorem 1, it suffices to compute, or bound, $\alpha M_i(\lambda)$ at each step and sum them to bound the moments of the mechanism overall. Then, starting from the tail bound we can come back to the (ϵ, δ)-differential privacy guarantee. The last challenge missing is to bind the values $\alpha M_t(\lambda)$ for every single step. Let μ_0 denote the Probability Density Function (PDF) of $N(0, \sigma^2)$, and μ_1 denote the PDF of $N(1, \sigma^2)$. Let μ be the mixture of two Gaussians $\mu = (1 - q)\mu_0 + q\mu_1$. Then we need to compute $\alpha(\lambda) = log(max(E_1, E_2))$ where

$$E_1 = E_{\mathrm{z}}[(\mu_0(z)/\mu(z))^{\lambda}] \tag{5}$$

$$E_2 = E_{\mathrm{z}}[(\mu(z)/\mu_0(z))^{\lambda}] \tag{6}$$

In the implementation of the moment accountant, we carry out numerical integration to compute α(λ). In addition, we can show the asymptotic bound

$$\alpha(\lambda) \leq q^2\lambda(\lambda + 1)/(1 - q)\sigma^2 + O(q^3/\sigma^3)$$

This inequation together with Theorem 1 implies Theorem 2. □

[5] The appropriate values for the noise scale and for the threshold will depend on the desired privacy cost and on the size of the dataset.

4 Experiments

In this section, we evaluate empirically our framework. The experiments are designed to assess the quality of the generated data, measure the privacy of the generated models and understand how differential privacy influences the output dataset. Moreover, we evaluate the solidity of the different models against membership inference attacks. Since it is notoriously arduous to assess the results of a GAN, we decided to combine qualitative and quantitative analysis to obtain a reliable evaluation. Qualitative analysis allows us to graphically verify the quality of the results and to observe the effects of differential privacy; while quantitative analysis provides a most accurate evaluation of the results; in particular, we measured some distance metrics to compare the generated data with the real data. Finally, our process included evaluating our model on a classification problem. This highlights the high utility of the data even when anonymization is used. For all experiments, when differential privacy is used, δ is supposed to be less than 10^{-5}, a value that is generally considered safe [1] because it implies that the definition of differential privacy is true with a probability of 99.999%. Indeed δ is the probability that a single record is safe and not spoiled. We kept the value of δ fixed to be able to evaluate the privacy of a mechanism with a single value ϵ that summarizes in a clearer manner the privacy guarantees.

In the different settings we applied only minor changes to the dp-GAN architecture, since we proved that it adapts well to each of them. In particular, in every case the discriminator is composed of a deep fully connected network. On the other hand, the architecture of the generator is adapted to the different datasets used. To generate time-series we used an LSTM which output becomes the input of the discriminator. On the contrary, in the case of discrete datasets we used a fully connected network which outputs the probability distribution for each value that a variable can assume. The interested reader can find an exhaustive explanation of the experiments, including additional datasets in the following GIT repository: `https://github.com/Lory94/dp-GAN`.

4.1 Synthetic dataset

In order to provide a first evaluation of the performances of the dp-GAN and understand the effects of differential privacy, we conducted a first experiment on a synthetic dataset. The dataset is constituted by samples coming from six 2D-gaussian distributions with the same variance, but with different centers. The quality of the results using dp-GAN is similar in both marginals and joint distributions.

Fig. 1 plots the kernel density estimation of the data to visualize the bivariate distribution. As expected, differential privacy introduces a small noise in the results, thus increasing the variance of the six gaussian distributions, while at the same time replicating faithfully the original distribution.

We use the Wasserstein distance to measure the distance between two distributions, to ascertain the quality of the GAN models. Fig. 2 plots the distance values for the non-anonymized GAN, a dp-GAN, and a dp-GAN using clipping decay. Both the dp-GAN models have $\epsilon = 8$. The different measures tend to converge to similar results, especially when clipping decay is applied, demonstrating the high quality of the results. Indeed, clipping decay allows the Wasserstein distance to drop to values comparable to those of the non-anonymized version in the second half of the graph. The main difference resides in the higher number of epochs necessary to reach the convergence, due to the noise addition.

4.2 Time-series data

To test our implementation on a real dataset we decided to use a set of data coming from the IoT system of the City of Antibes, in France. This dataset is private because it contains sensitive information about the water consumption and water pipeline maintenance, obtained directly from sensors in each neighborhood. The purpose is to support public administration in releasing highly relevant open data, while hiding specific events in the time-series, and preventing individual re-identification. With minor adjustments, the solution can constitute a valid framework applicable to other purposes, such as electricity consumption and waste management.

The dataset is an extract of one month of measurements, where each sample is a time-series containing 96 values (one every 15 minutes). Each sample is labelled with the name of the neighborhood. The data has been normalized before the training. The goal is to generate a new time-series that contains the same number of records and the same distribution as the original dataset, while providing differential privacy guarantees, that is, each sample does not influence whatever analyses more than a certain threshold. In this way, anomalous situations such as maintenance works, a failure in a water pipe or an unexpected water usage by a person living in a certain area are protected and kept private.

Fig. 3 compares real samples and generated ones, using non-anonymized-GAN and a dp-GAN with $\epsilon = 6$. We plot the sensor values for a generated sample and the closest sample coming from the original data, in terms of the

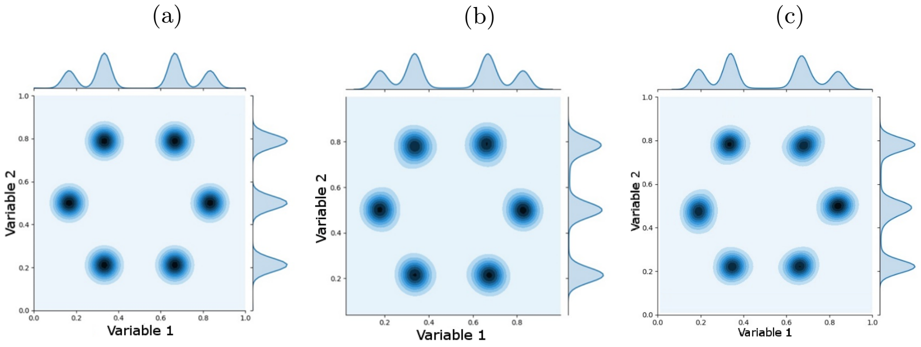

Fig. 1. Kernel estimation for: (a) the original points, (b) WGAN, and (c) dp-GAN

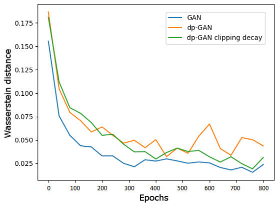

Fig. 2. Wasserstein distance using anonymized and non anonymized GANs

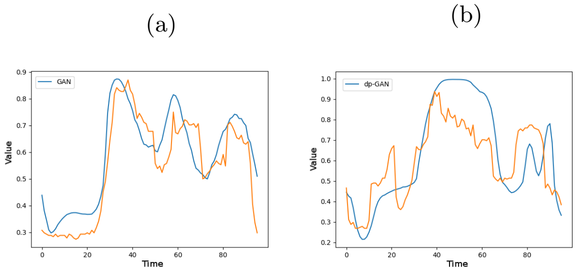

Fig. 3. Generated sample from a non-anonymized GAN (a) and dp-GAN (b), in blue. In orange, the closest sample present in the dataset, in terms of dynamic time warping.

dynamic time warping distance. The distribution of the original time series is kept, but in the dp-GAN samples, the curves tend to be smoother, hiding some of the variability of the original data.

For time-series data, the quality assessment for GANs represents a challenge. While for images the inception score [13] has become the standard measure to evaluate the performance of a GAN, there is no counterpart for the assessment of time series. We believe that this represents an interesting area of research for the future.

4.3 Discrete data

We analyzed the performances of our model on the UCI adult dataset: this dataset is an extract of the US census and contains information about working adults, distributed across 14 features. A classification task on it is a reliable benchmark, because of its widespread use in several studies. Records are classified depending on whether the individual earns more or less than $50k$ dollars each year.

To use accuracy as an evaluation metric, we decided to sample the training and test data in such a way that both classes would be balanced. We built a random forest classifier on the dataset generated by the dp-GAN. We evaluated the accuracy on the test set and compared it with the one of the model built on the real non-anonymized dataset. If the dp-GAN model behaves correctly, all the correlations between the different features should be preserved. Therefore, the final accuracy should be similar to what was achieved by using the real training set. We also tracked the privacy costs to verify that the generated data were correctly anonymized. Finally, we examined how much membership inference attacks can influence our model and compared it to a non-anonymized GAN model.

Table 1 evaluates the degradation of the performances when differential privacy is adopted. The target accuracy of 77.2% was reached by using the real training set to train the random forest classifier. As expected, a non-anonymized

Method	Epsilon	Accuracy
Real dataset	Infinite	77.2 %
GAN	Infinite	76.7 %
dp-GAN	3	73.7 %
dp-GAN clipping decay	3	75.3 %
dp-GAN	7	75.0 %
dp-GAN clipping decay	7	76.0 %

Table 1. Classification accuracy for training sets generated by different models

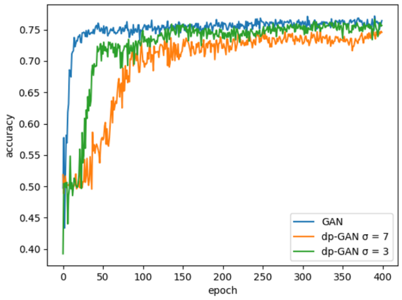

Fig. 4. classification accuracy Average for 5 runs for different noise values

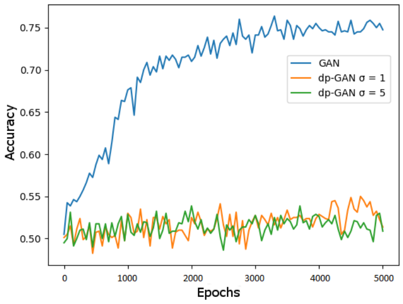

Fig. 5. Membership inference attack accuracy for non-anonymized GAN, dp-GAN with $\sigma = 1$ and $\sigma = 5$

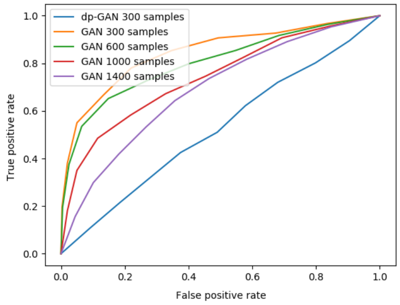

Fig. 6. ROC curves for membership inference attacks for GAN and dp-GAN using generated samples of different size

GAN is able to produce high quality data: its accuracy loss is very low, 0.5%, compared to the target. Interestingly, even when we adopted the dp-GAN framework the accuracy remained high. Using $\epsilon = 7$ and clipping decay, we obtained results similar to the ones without anonymization. In addition, Table. 1 points out the positive effect of clipping decay, that it is able to increase the accuracy of about 1%.

Fig. 4 highlights the effects on the classification accuracy when dp-GAN is adopted using different amounts of noise. The main effect is to slow down the training process, but not to significantly impact accuracy. Indeed, the added noise requires more epochs to reach convergence, which is amplified by σ. In most of the use cases, this is a minor drawback, considering the classification accuracy. Moreover, the dp-GAN is trained once; then the generator can be released to produce samples on demand.

Fig. 5 shows the analysis of the accuracy of membership inference attacks on the model using different training procedures with different levels of privacy guarantees. This analysis has been done at different epochs of the training process. The accuracy of the model increases over time, however this makes the model more subject to membership inference attacks. As can be seen from Fig. 6, training the model with no anonymization rapidly increases the accuracy of attacks: this highlights the problems that still afflict many generative models that cannot effectively generalize the training data. On the contrary, by increasing the privacy level, the accuracy of the attacks tends to remain close to 50%. This is obtained at the costs of losing about 1% of accuracy during the final classification.

The size of the dataset is another important factor that influences significantly the results, since GANs need a good amount of data to generalize effectively. Fig. 6 confirms the results obtained in [17], but at the same time it shows how differential privacy works well even when the dataset is small. Indeed, the dp-GAN provides random accuracy towards membership inference attacks independently from the size of the dataset. It is interesting to notice that since the dataset is small, the level of privacy ϵ is big compared to what it is commonly used; however, the effects of differential privacy can be still perceived clearly.

5 Related work

Differential privacy on machine learning models. In [16] it is proposed an innovative approach for training a machine learning model in a differentially private manner. On the contrary of the dp-SGD, they proved that it is possible to reach differential privacy by transferring knowledge from some models to others in a noisy way. A set of models, called teachers, are trained on the real dataset and a student model learns in a private way what the teachers have grasped during the training. However, it is still unclear how this implementation can be extended to a non-interactive setting. [19] developed a dp-GAN based on the dp-SGD providing some optimizations in order to improve performances focusing on the generation of images. In contrast, our work highlights that dp-GAN can be adapted to a variety of different use cases and in particular we developed a variation dedicated to discrete data. In addition, we provide, also, an overview of the effects that differential privacy has on membership inference attacks; [17] pointed out how severe the risk of this kind of attack in a general machine learning model can be. We have confirmed the issue while highlighting that the noise introduced by differential privacy reduces overfitting and consequently the accuracy of membership inference attacks.

Generative adversarial networks on discrete data. [20] developed SeqGAN, an approach dedicated to the generation of sequences of discrete data. SeqGAN is based on a reinforcement learning training where the reward signal is produced by the discriminator. However, it is not clear how this approach can be extended to include differential privacy. On the contrary, [15] uses Cramer GANs to combine discrete and continuous data. These recent works did not address data privacy concerns.

Differential privacy without deep learning. Interesting results have been also obtained through other types of generative models. In the context of non-interactive anonymization, [21] developed a differentially private method for releasing high-dimensional data through the usage of Bayesian networks. This kind of network is able to learn the correlations present in the dataset and generate new samples. In particular, the distribution of the dataset is synthetized through a set of low dimensional marginals where noise is injected and from which it is possible to generate the new samples. However, the approach suffers from an extremely high complexity, thus being unpractical to anonymize large datasets. We have also analysed the literature to prove that the amount of privacy that dp-GAN guarantees is comparable to the one of the other most common implementations. Although there is no specific value for which ϵ is considered safe, we obtained most often lower privacy costs compared to [7, 4], which are the two most relevant works dealing with real-life datasets. Similar privacy costs have been used in the most recent literature in the differential privacy field [1, 16].

6 Conclusion

In this paper we extended the notion of dp-GANs to privacy-preserving data publishing of continuous, time-series and discrete data. We introduced clipping decay to preserve data utility while providing data privacy: it can be used for any differentially private gradient descent on any neural network to improve learning accuracy. We have shown how our implementation is resistant to membership inference attacks, being suitable for open data releases. In the future, we will work on the reduction of privacy costs and investigate the potential benefits of transfer learning to data anonymization.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: 23rd ACM Conference on Computer and Communications Security (ACM CCS). pp. 308–318 (2016), https://arxiv.org/abs/1607.00133
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 214–223. PMLR, International Convention Centre, Sydney, Australia (06–11 Aug 2017), http://proceedings.mlr.press/v70/arjovsky17a.html
3. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 2172–2180. Curran Associates, Inc. (2016)
4. Differential Privacy Team, A.: Learning with privacy at scale. Apple Machine Learning Journal (2017)
5. Dwork, C.: Differential privacy: A survey of results. In: In Theory and Applications of Models of Computation. pp. 1–19. Springer (2008)

6. Dwork, C., Roth, A.: The Algorithmic Foundations of Differential Privacy. Now Foundations and Trends (2014). https://doi.org/10.1561/0400000042, https://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=8187424
7. Erlingsson, Ú., Pihur, V., Korolova, A.: Rappor: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. pp. 1054–1067. ACM (2014)
8. Geng, Q., Viswanath, P.: The optimal noise-adding mechanism in differential privacy. IEEE Transactions on Information Theory **62**, 925–951 (2016)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
10. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5767–5777. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf
11. Juefei-Xu, F., Boddeti, V.N., Savvides, M.: Gang of gans: Generative adversarial networks with maximum margin ranking. CoRR **abs/1704.04865** (2017)
12. Kellaris, G., Papadopoulos, S., Xiao, X., Papadias, D.: Differentially private event sequences over infinite streams. Proc. VLDB Endow. **7**(12), 1155–1166 (Aug 2014). https://doi.org/10.14778/2732977.2732989, http://dx.doi.org/10.14778/2732977.2732989
13. Lucic, M., Kurach, K., Michalski, M., Gelly, S., Bousquet, O.: Are gans created equal? a large-scale study. In: Advances in neural information processing systems. pp. 697–706 (2018)
14. Mirza, M., Osindero, S.: Conditional generative adversarial nets. CoRR **abs/1411.1784** (2014)
15. Mottini, A., Lheritier, A., Acuna-Agost, R.: Airline passenger name record generation using generative adversarial networks. CoRR **abs/1807.06657** (2018)
16. Papernot, N., Abadi, M., Erlingsson, Ú., Goodfellow, I.J., Talwar, K.: Semi-supervised knowledge transfer for deep learning from private training data. CoRR **abs/1610.05755** (2016), http://arxiv.org/abs/1610.05755
17. 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 (May 2017). https://doi.org/10.1109/SP.2017.41
18. Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. IEEE Transactions on Knowledge and Data Engineering **23**(8), 1200–1214 (Aug 2011). https://doi.org/10.1109/TKDE.2010.247
19. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially private generative adversarial network. CoRR **abs/1802.06739** (2018), http://arxiv.org/abs/1802.06739
20. Yu, L., Zhang, W., Wang, J., Yu, Y.: Seqgan: Sequence generative adversarial nets with policy gradient. CoRR **abs/1609.05473** (2016), http://dblp.uni-trier.de/db/journals/corr/corr1609.html=YuZWY16
21. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: Privbayes: Private data release via bayesian networks. ACM Trans. Database Syst. **42**(4), 25:1–25:41 (Oct 2017). https://doi.org/10.1145/3134428, http://doi.acm.org/10.1145/3134428