The ISC Int'l Journal of Information Security
ISeCure

July 2023, Volume 15, Number 2 (pp. 240–253)
http://www.isecure-journal.org

An Auto-Encoder based Membership Inference Attack against Generative Adversarial Network

Maryam Azadmanesh [1], Behrouz Shahgholi Ghahfarokhi [1,*], and Maede Ashouri Talouki [1]
[1] *Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran.*

ARTICLE INFO.

Article history:
Received: December 19, 2021
Revised: June 6, 2022
Accepted: September 4, 2022
Published Online: July 1, 2023

Keywords:
Membership Inference Attack, Generative Adversarial Network, Privacy, White-Box

Type: Research Article

doi: 10.22042/isecure.2022.320282.735

dor: 20.1001.1.20082045.2023.15.2.1.0

ABSTRACT

Using generative models to produce unlimited synthetic samples is a popular replacement for database sharing. Generative Adversarial Network (GAN) is a popular class of generative models which generates synthetic data samples very similar to real training datasets. However, GAN models do not necessarily guarantee training privacy as these models may memorize details of training data samples. When these models are built using sensitive data, the developers should ensure that the training dataset is appropriately protected against privacy leakage. Hence, quantifying the privacy risk of these models is essential. To this end, this paper focuses on evaluating the privacy risk of publishing the generator network of GAN models. Specially, we conduct a novel generator white-box membership inference attack against GAN models that exploits accessible information about the victim model, i.e., the generator's weights and synthetic samples, to conduct the attack. In the proposed attack, an auto-encoder is trained to determine member and non-member training records. This attack is applied to various kinds of GANs. We evaluate our attack accuracy with respect to various model types and training configurations. The results demonstrate the superior performance of the proposed attack on non-private GANs compared to previous attacks in white-box generator access. The accuracy of the proposed attack is 19% higher on average than similar work. The proposed attack, like previous attacks, has better performance for victim models that are trained with small training sets.

1 Introduction

Nowadays, machine learning models are used in various applications. Availability of large datasets is one of the critical factors in the success of these models, while datasets are often crowded and may contain sensitive data. Therefore, their confidentiality and privacy are essential. However, machine learning models are known to implicitly memorize inappropriate details of sensitive data during training. Therefore, assessing the privacy risks of machine learning models is necessary. For this purpose, many attacks are conducted against these models, which can infer information about training datasets. One such attack is the membership inference attack [1]. In a membership inference attack, given a data record and access to the learned model, the attacker determines if the record was in the model's training dataset or not.

* Corresponding author.
Email addresses: `m.azadmanesh@eng.ui.ac.ir`, `shahgholi@eng.ui.ac.ir`, `m.ashouri@eng.ui.ac.ir`
ISSN: 2008-2045 © 2023 ISC. All rights reserved.

GAN [2] is a class of generative models that learn the distribution of training data and generate synthetic data with a distribution very similar to that. The GAN architecture typically comprises two neural networks, a generator, and a discriminator. The task of the discriminator is separating generated samples from the training ones, and the generator tries to deceive the discriminator by generating samples that the discriminator misclassifies. Publishing the generator to generate an unlimited number of synthetic samples is a popular replacement for database sharing. Although in GAN architecture, the generator does not have access to the training data, the sensitive information of the training data is propagated through gradients from the discriminator to the generator. Therefore, if the generator overfits the training data, its weights and the generated synthetic samples leak information about the sensitive training data.

Membership inference attacks are categorized into black-box and white-box attacks. In the black-box setting, only the outputs of the model are accessible to the attacker, while in the white-box setting, the model internals and parameters are also available. As in common practices, only the generator is published to generate synthetic samples, [3] presented a more detailed classification of membership inference attacks against GANs. In their taxonomy, according to the order of accessible information about victim models, membership inference attacks against GANs can be categorized into (1) full black box generator, (2) partial black-box generator, (3) white-box generator, and (4) accessible discriminator. In the least knowledgeable setting, i.e., full black box generator, only the synthetic samples are accessible to the attacker. In the partial black-box generator, the attacker has no access to the generator internals but can provide the generator input (latent code) and view the corresponding synthetic sample. In the white-box generator, the generator internal is also accessible, and in the most knowledgeable setting (i.e., accessible discriminator), in addition to the generator, the discriminator is available. Membership inference attacks have better performance for victim models that are trained with small training datasets [3].

To the best of our knowledge, there exist a few membership inference attacks against GANs, and GAN-Leak [3] is the only attack conducted in a white-box generator setting that does not have high accuracy. Therefore, designing a high-accuracy attack against the generator network with white-box access is a requirement. To this end, in this paper, a white-box generator attack is presented, which optimally uses the available information about the victim model (synthetic samples and generator parameters) and provides higher accuracy than GAN-Leak. In the proposed attack, an auto-encoder is trained by the generated synthetic samples of the victim model, and the victim model's parameters are used to set the parameters of the decoder in the auto-encoder. Therefore, all available information about the victim model's generator in white-box access is used to conduct the attack. Our attack model can help developers and data holders to better quantify the privacy risk of publishing their generator models. Our attack model applies to various types of GANs. We investigate our attack against several victim models and three different datasets.

The remainder of the paper is organized as follows. Section 2 reviews recent research activities on membership inference attacks. Section 3 introduces GAN concepts used in the paper. Section 4 presents our attack. In Section 5, the attack is evaluated, and finally, in Section 6, the findings are summarized, and the conclusion is presented.

2 Related Work

This paper is mainly related to research in two directions, one direction is about membership inference attacks against machine learning (ML) models, and the other direction is about designing privacy-preserving mechanisms in GANs. Following, we review the top related research in both directions.

2.1 Membership Inference Attacks Against ML Models

A membership inference attack is an attack that can infer information about training data from a trained model. The first membership inference attack against discriminative neural network models was introduced by Shokri *et al.* [1]. To conduct the attack, the output of multiple shadow models is used to train the attack model. Shokri *et al.* demonstrate that overfitting is an essential factor in the success of membership inference attacks. Later, Salem *et al.* [4] conduct Shokri's attack with fewer assumptions. The authors show that with these fewer assumptions, the attacker can achieve a very similar performance as reported by[1]. Long *et al.* [5] discuss that overfitting is sufficient but not necessary for a membership inference attack and conduct an attack against well-generalized models to show that even these models contain vulnerable instances. Yeom *et al.* [6] formulate quantitative advantage of adversaries for membership inference attack in terms of generalization error and influence. They show that although overfitting is a sufficient condition for membership inference attack, it is not a necessary condition. Kaya *et al.* [7] investigate the impact of various regularization techniques on the success of the membership inference attack. They show that some regularizations may help membership inference attacks. Sablayrolles *et al.* [8] exploit a probabilistic framework to derive an optimal strategy for membership inference attacks. They show that the optimal attack only depends on the loss function, and thus, black-box attacks are as good as white-box attacks.

Li *et al.* [9] and Choquette-Choo *et al.* [10] conduct membership inference attacks by exploiting only predicted class labels, compared to other methods that use confidence scores of all classes. Long *et al.* [11] conduct a black-box attack in which the attacker attempts to minimize false positives by carefully selecting vulnerable records. Rezaei *et al.* [12] investigate the false alarm rates in the membership inference attacks. They show that the current membership inference attacks cannot achieve a low false alarm rate and high accuracy at the same time, since the features used in these attacks are not statistically different for training and test data records. Hu *et al.* [13] conduct a black-box attack, named BLINDMI, without using shadow models. In BLINDMI, first, a non-member training set is generated, and a differential comparison is performed between the target set and the generated set when one sample moves from the target set to the generated set. Nasr *et al.* [14] present a comprehensive framework for the privacy analysis of deep neural networks using white-box membership inference attacks. They measure the privacy leakage by leveraging the final model parameters and parameter updates during the training and fine-tuning processes. They design the attack in stand-alone and federated settings, concerning passive and active attackers assuming different adversary prior knowledge. They show that the gradients and outputs of the last layers leak more membership information. ML-Privacy [15] is a tool that quantifies the privacy risk of training data in the discriminative models by implementing Nasr's attack [14].

Leino *et al.* [16] conduct a white-box membership inference attack with more realistic assumptions than Nasr *et al.* [14]. They assume that the attacker does not have access to the training model of the victim model and conducts the attack by exploiting the idiosyncratic features of training data encoded in the victim model during training. The first membership inference attack against generative models was introduced by [17]. There, the authors present a full black-box generator attack and a white-box accessible discriminator attack. In the white-box attack with an accessible discriminator, they use the discriminator's output to learn statistical differences between members of training datasets and non-members. In the black-box generator attack, synthetic samples generated by the victim model are used to train a GAN model, and the outputs of its discriminator are used to conduct the attack similar to the white-box attack. Hilprecht *et al.* [18] have conducted a black-box inference membership attack by using generated samples of the model. The intuition behind that attack is the fact that the generator overfits the training data if it tends to generate outputs very close to the provided training data. Therefore the authors have inferred a record with the largest number of the nearest generated samples as a membership record. Liu *et al.* [19] propose a black-box inference attack, in which, given a record, the attacker trains a neural network that can reconstruct the record, and if the reconstruction error is small, the record is considered as a member of the training set. This attack requires retraining a new neural network for each record.

Later, [3] extended Hilprecht's attack model [18] to the white-box setting, the partial black-box generator setting, and the full-black-box setting. In the black-box attack, instead of counting the number of the nearest generated samples to the query, authors exploit the reconstruction distance. In the partial black-box attack, they use the latent code (z) to find a better reconstruction of the query sample, and in the white-box attack, by accessing the gradient information and using first-order optimization algorithms, they solve the reconstruction problem more accurately. To further improve the effectiveness of their attack, they also propose a calibration technique, which can distinguish between a difficult sample representation and a non-member of the victim model's training dataset. Hu *et al.* [20] propose a white-box accessible discriminator attack. In their attack, first, the data distribution is split into different regions, and a membership confidence score is assigned to each region. Then if a record of the target set falls in a region with a high member score, it is considered as a member.

Li *et al.* [21] formulate the membership privacy loss as a statistical divergence between the distribution of training set and non-training set samples, and propose a sample-based method to estimate the divergence. They test their framework using various queries against different classes of generative models. Webster *et al.* [22] conduct an identity membership inference attack against GAN models, in which instead of determining a record as a member of training records of the victim model, the attacker attempts to discern whether a sample with the same identity is used as a training record. Zhou *et al.* [23] propose a property inference attack against GAN models, in which the attacker attempts to infer a certain general property about the training records of the victim model.

2.2 Privacy-Preserving Mechanisms in GANs

Various methods have been proposed which provide privacy guarantees for GANs. Some of these methods provide a strong theoretical guarantee for sensitive data privacy, i.e., differential privacy. In contrast, some methods only empirically protect training data against a particular attack to generate better quality synthetic data. Since in some data-publishing scenarios, only the generator network is released publicly, some work only trains generators with a privacy guarantee. In other methods, both discriminator and generator are trained with a privacy guarantee.

To train both the discriminator and the generator in a differentially private manner, many approaches [24–27] utilize DP-SGD [28]. In these methods, in each training step, the discriminator's gradients are clipped based on clipping bounds, and then, to guarantee differential privacy, random noise is added to the clipped gradients. GAN-PATE [29] is another method, which trains both the discriminator and the generator in a differentially private manner. This method is based on the Private Aggregator of Teacher Ensemble (PATE) method [30]. In this method, the GAN discriminator is replaced with the PATE mechanism, which means that the K-teacher discriminator, and one student discriminator are trained, and the student-discriminator is trained on the generated synthetic samples labeled by the teachers.

Many approaches [31–34] exploit the PATE method to train only the generator with a differential privacy guarantee. In G-PATE [31], multiple teacher-discriminators and one student-generator are trained. To control the information flow from the discriminators to the generator, G-PATE [31] uses gradient discretization and noisy aggregation of teacher-discriminators' votes on the discrete gradients. DATALENS [32] improves the utility of G-PATE [31] with top-k gradient compression. GS-WGAN [33] also uses multiple teacher-discriminators and one student-generator. At each training step, a randomly selected discriminator and the generator, update their parameters. To prevent information leakage from the selected teacher to the student-generator, Abadi's method [28] with improved WGAN is used, and the gradient clipping bound is set to one. Han and Xue [34] propose another method that provides a privacy guarantee for the generator network. In this method, in each update of the generator's parameters, the discriminator loss is clipped, and the appropriate noise is added to it.

The first empirical defense against membership inference attack in ML models is introduced by Nasr *et al.* [35]. They design a multi-objective learning algorithm with the goal of minimizing the classification loss and maximizing the gain of the membership inference attack. PrivGAN [36] is an empirical defense for GAN models which defends against membership inference attacks. In PrivGAN architecture, the training dataset is divided into equal-sized non-overlapping subsets, and each partition would be used to train separate discriminator-generator pairs. The generators are trained not only to cheat the discriminators but also to cheat a built-in adversary whose goal is to identify which generator generated the synthetic sample. PAR-GAN [37] is another empirical defense against membership inference attacks for GAN models, where one generator and multiple discriminators are trained. Each discriminator is trained separately on a disjoint partition of the training data, and the generator is adversarially trained with multiple discriminators.

3 Background

GAN [2] is a class of unsupervised learning algorithms. GAN [2] architecture typically comprises two neural networks, a generator G and a discriminator D, in which G learns to map from a latent distribution p_z to the true data distribution p_{data}, while D discriminates between instances sampled from p_{data} and those generated by G. G's objective is to fool D by synthesizing instances that appear to be from p_{data}. The training objective is formulated as

$$\begin{aligned} &\min_{\theta_G}\max_{\theta_D}\mathbb{E}_{x\sim p_{data}}[\,log(\,D_{\theta_D}(\,x\,)\,)\,] \\ &+\mathbb{E}_{z\sim p_z}[\,log(\,1-D_{\theta_D}(\,G_{\theta_G}(\,z\,)\,)\,)\,] \end{aligned} \tag{1}$$

where θ_G and θ_D represent the parameters of the generator network and the discriminator network, respectively [2]. Figure 1 shows the GAN architecture.

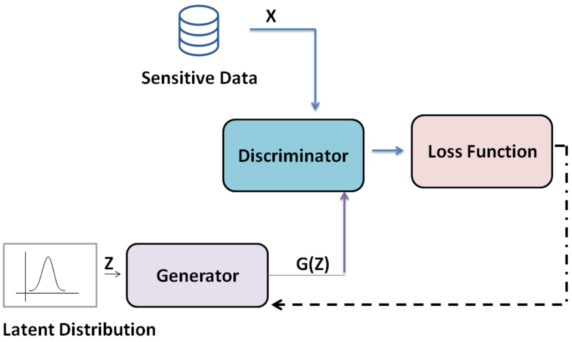


Figure 1. GAN architecture

Despite its simplicity, the original GAN formulation is unstable and inefficient to train. A number of recent works propose new training procedures and network architectures to improve training stability and convergence rate. In particular, the Wasserstein Generative Adversarial Network (WGAN) [38] and Improved Training of Wasserstein GANs (WGANGP) [39] attempt to minimize the earth mover distance between the synthesized distribution and the true distribution rather than their Jensen-Shannon divergence as in the original GAN formulation. Least Square Generative Adversarial Network (LSGAN) [40] adapts the least square loss function for the discriminator, and

Deep Regret Analytic Generative Adversarial Network (DRAGAN) [41] uses gradient penalty with GAN. In this paper, to evaluate the proposed attack against GAN models, WGANGP [39], LSGAN [40], and DRAGAN [41] are used as victim models. The training objective of WGANGP [39] is formulated as:

$$\min_{\theta_G}\max_{\theta_D}\mathbb{E}_{x\sim p_{data}}[D_{\theta_D}(x)] - \mathbb{E}_{z\sim p_z}[D_{\theta_D}(G_{\theta_G}(z))] + \lambda(\|\nabla_{\tilde{x}}D_{\theta_D}(\tilde{x})\|_2 - 1)^2 \tag{2}$$

where θ_G and θ_D represent the parameters of the generator network and the discriminator network, respectively. Also, $\tilde{x} = \epsilon x + (1-\epsilon)G_{\theta_G}(z)$, where ϵ is a random number sampled from $[0,1]$ according to a uniform distribution.

Similarly, the training objective of DRAGAN [41] is formulated as:

$$\begin{aligned}\min_{\theta_G}\max_{\theta_D}\mathbb{E}_{x\sim p_{data}}[log(D_{\theta_D}(x))]\\ +\mathbb{E}_{z\sim p_z}[log(1-D_{\theta_D}(G_{\theta_G}(z)))]\\ +\lambda(\|\nabla_{\tilde{x}}D_{\theta_D}(\tilde{x})\|_2 - 1)^2\end{aligned} \tag{3}$$

where θ_G and θ_D represent the parameters of the generator network and the discriminator network, respectively. Also, $\tilde{x} = \epsilon x + (1-\epsilon)G_{\theta_G}(z)$, where ϵ is a random number sampled from $[0,1]$ according to a uniform distribution. Finally, the objective function of LSGAN [40] is defined as:

$$\begin{aligned}\min_{\theta_D} V_{LSGAN}(\theta_D) &= \frac{1}{2}E_{x\sim p_{data}}[(D_{\theta_D}(x) - b)^2]\\ &+\frac{1}{2}\mathbb{E}_{z\sim p_z}[(D_{\theta_D}(G_{\theta_G}(z)) - a)^2]\\ \min_{\theta_G} V_{LSGAN}(\theta_G) &= \frac{1}{2}\mathbb{E}_{z\sim p_z}[(D_{\theta_D}(G_{\theta_G}(z)) - c)^2]\end{aligned} \tag{4}$$

where θ_G and θ_D denote the parameters of the generator and the discriminator, respectively. p_{data} is the real data distribution, and p_z is the prior distribution of the latent code. a and b are the labels for fake data and real data, respectively, and c denotes the value that the generator wants the discriminator to believe for fake data.

4 Proposed Attack

4.1 Threat Model

In a membership inference attack, an adversary aims to infer whether a sample is used in the training data or not. Formally, given a target data point X_{target} and a trained machine learning model, M, a membership inference attack can be defined as the following function:

$$A : X_{target}, M \rightarrow \{0,1\} \tag{5}$$

where $0/1$ output means X_{target} is a non-member/member of M's training dataset. We assume that M is a generative adversarial network, and the generator is accessible to the attacker in a white-box manner. Therefore, in addition to the input and output of the generator, the attacker has access to the internal of the generator. Similar to other related work, we consider an honest-but-curious adversary that aims to identify individual records that were used to train the model. To this end, N records from the training dataset and N records from the test dataset, i.e., $\{x_1,\ldots,x_{2N}\}$ are given, and the attacker labels N records as members of the training dataset. This set is named attacker-set. The accuracy of the attack is defined as the proportion of actual training data in these N records.

4.2 Attack Model

Our membership inference attack exploits two observations about GANs. First, as Figure 1 shows, the sensitive training data is only fed into the discriminator and impacts the discriminator's weights and gradients. In the training procedure, this gradient information propagates from the discriminator back to the generator. Therefore, if the generator overfits the training data, its weights leak information about the sensitive training data. Second, a generator model tends to approximate the training data distribution, and if the generator overfits the training data, it tends to output a dataset close to the training data. Therefore, the attacker exploits two features of the victim model: (1) generator's parameters and (2) synthetic generated samples.

To construct an attack model which exploits these two features and can behave differently on the training data and non-training data, we train an auto-encoder. This auto-encoder is trained by the synthetic samples generated by the generator. Therefore, in the first step, by accessing the generator of the victim model, the attacker generates a number of synthetic samples to train the auto-encoder.

In the second step, the auto-encoder is trained using the synthetic samples. Auto-encoder consists of two parts, an encoder, and a decoder. The decoder architecture is the same as the generator, and its parameters are set to the generator's parameters. In the training procedure, these parameters are fixed (non-trainable). Therefore, the encoder aims to map the inputs to the features in the latent space (z) that if it is fed to the generator (the decoder), the input is reconstructed. The encoder trains on the samples generated by the generator. Formally, the training objective is formulated as

$$min_{\theta_{enc}} l(x, G_{\theta_G}(enc_{\theta_{enc}}(x))) \tag{6}$$

where $enc_{\theta_{enc}}$ and θ_{enc} encoder and encoder's parameters. G_{θ_G} denotes the generator, and x is the generated sample produced by the generator. l denotes a loss function penalizing for $G_{\theta_G}(enc_{\theta_{enc}}(x))$ being dissimilar from x. In the attack model's loss function, we use mean square error.

In the third step, the reconstruction error of the attack set is obtained using the trained auto-encoder. In other words, the attacker injects his/her attacker-set, $\{x_1, \ldots, x_{2N}\}$ into the auto-encoder, and obtains $\{l_1, \ldots, l_{2N}\}$, where l_i is $l(x_i, G_{\theta_G}(enc_{\theta_{enc}}(x_i)))$.

Finally, in the fourth step, the attacker separates the member/non-member records of the attack set. To do this, the attacker labels the N records with the lowest cost values as members of the training dataset. In other words, as the decoder is the generator of the victim model, the output of the auto-encoder can be considered as the nearest synthetic sample to the input. Therefore, the records with a minimum distance to their nearest synthetic samples are selected as training datasets. Figure 2 shows the high-level overview of the proposed white-box attack model.

5 Experimental Results

In this section, the proposed attack is compared to LOGAN accessible discriminator, LOGAN full black-box attack [1] [17], and GAN-leak white-box generator [2] [3]. In LOGAN accessible discriminator, the discriminator is accessible, and as described, this setting is the most knowledgeable attack. In the other attacks, only the generator is accessible. In the GAN leak white-box generator and the proposed attack, the generator is accessed in a white box manner, while in the LOGAN full black-box attack, only the synthetic samples are available.

To conduct the experiments, three benchmark datasets are used:

- MNIST, which consists of 70000 labeled handwritten digit images split into 60000 training and 10000 test samples. Each image is a 28×28 grayscale image.
- Fashion-MNIST, which comprises 70000 labeled images of 10 fashion categories separated into 60000 training and 10000 test samples. Each image is a 28×28 grayscale image.
- CelebA, which consists of 200000 celebrity face images. We have selected 60000 random images which are center-cropped and resized to 48×48.

As stated before, we select WGANGP [39], DRAGAN [41], and LSGAN [40] as victim models. In the victim models, the network architecture is similar to [42] and training data size is variable (from 64 to 4096 records). The learning rates of the discriminator and the generator are set to $5 * 10^{-5}$. In WGANGP and LSGAN, the number of iterations on the discriminator and the generators are 4 and $1 * 10^5$, respectively. In DRAGAN, the number of iterations on the discriminator and the generators are 1 and $1 * 10^5$, respectively. In WGANGP and DRAGAN, the coefficient of gradient penalty has a value of 10. The batch size is set to 64 in training all victim models. All models are implemented in Tensorflow. Figure 3 shows the sample synthesized results generated by the victim models when the training data size is 512.

Table 1. FID measure for different GAN models

	WGANGP	LSGAN	DRAGAN
MNIST	95.59	65.62	78.85
FASHION MNIST	123.61	115.06	126.83

To quantitatively assess the synthetic generated images of victim models, we use Frechet Inception Distance (FID) [43] and Jensen-Shannon scores on labeled (i.e., MNIST and Fashion MNIST) and unlabeled (i.e., CelebA) datasets, respectively.

FID score captures the similarity of the generated images to real ones. Formally FID is defined as follows:

$$FID = \| \mu_t - \mu_s \|^2 + tr(\sum\nolimits_t + \sum\nolimits_s - 2(\sum\nolimits_t \sum\nolimits_s)^{\frac{1}{2}}) \tag{7}$$

where $x_t \sim N(\mu_t, \sum_t)$ and $x_s \sim N(\mu_s, \sum_s)$ are element vectors of a specific layer of an Inception Network (e.g., 2048 element-wise activations of Inception v3 pool3 layer) for real and generated images, respectively and tr denotes trace of a matrix. Lower FID values indicate more similarity between the real and generated images, corresponding to better-generated image quality. Table 1 shows the FID for different victim GAN models on labeled datasets when the training data size is 512. As this table shows, the highest and lowest values of FID are related to the WGANGP and LSGAN, respectively. So, the images generated by LSGAN have the highest quality, and the images generated by WGANGP have the lowest quality compared to the others.

To evaluate Jensen-Shannon scores on the unlabeled dataset, another discriminator, D' is trained to classify real and synthetic samples. Using the output of the discriminator, Jenson-Shannon divergence between the conditional probability of the discriminator's output and a Bernoulli distribution with parameter $p = 0.5$ is measured. Formally, the Jenson-Shannon divergence of these distributions is defined as [2]:

$$S(G) = \frac{1}{2} KL(p(w \mid u) \parallel B_p) + \frac{1}{2} KL(B_p \parallel p(w \mid u)) \tag{8}$$

where B_p is Bernoulli distribution with parameter $p = 0.5$ and $p(w \mid u)$ is the conditional distribution of discriminator's output to predict u's label as w

[1] https://github.com/jhayes14/gen˙mem˙inf
[2] https://github.com/DingfanChen/GAN-Leaks

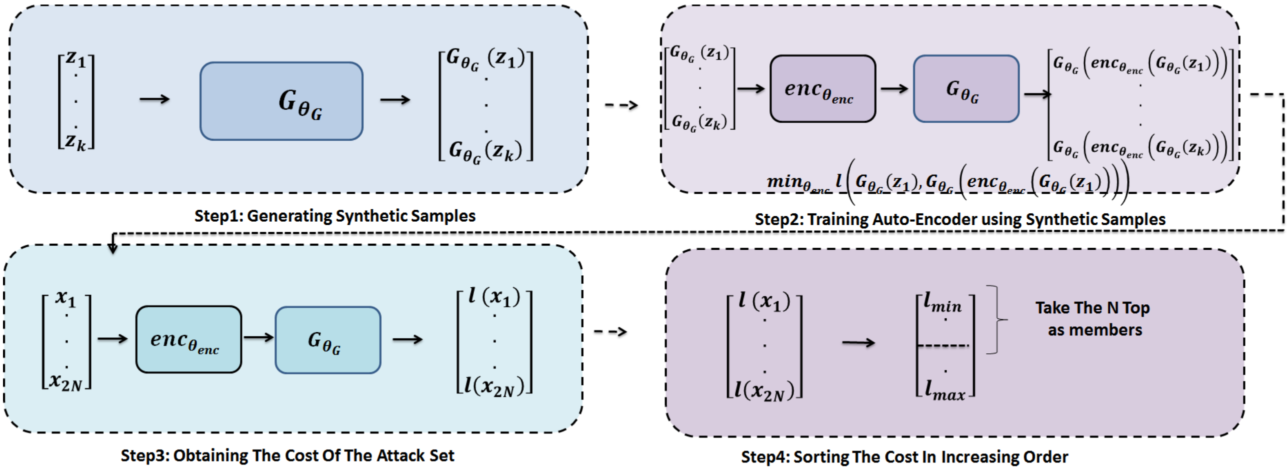

z_1
z_k
G_{θ_G}
$G_{\theta_G}(z_1)$
$G_{\theta_G}(z_k)$
Step1: Generating Synthetic Samples
$enc_{\theta_{enc}}$
$G_{\theta_G}(enc_{\theta_{enc}}(G_{\theta_G}(z_1)))$
$G_{\theta_G}(enc_{\theta_{enc}}(G_{\theta_G}(z_k)))$
$min_{\theta_{enc}}\ l(G_{\theta_G}(z_1), G_{\theta_G}(enc_{\theta_{enc}}(G_{\theta_G}(z_1))))$
Step2: Training Auto-Encoder using Synthetic Samples
x_1
x_{2N}
$l(x_1)$
$l(x_{2N})$
Step3: Obtaining The Cost Of The Attack Set
l_{min}
l_{max}
Take The N Top as members
Step4: Sorting The Cost In Increasing Order

Figure 2. High-level overview of the proposed white-box box model

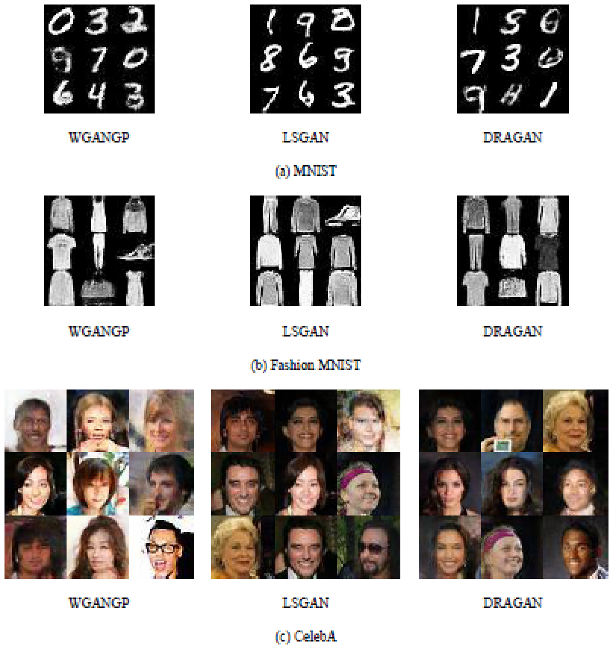


Figure 3. Sample images generated by different victim models after training

(real/synthetic sample label). The more the synthetic samples are similar to the real samples, the shorter the distance between the conditional distribution and the Bernoulli distribution. Therefore, the lower value of $S(G)$ indicates a better generator. Table 2 shows the Jenson-Shannon divergence for different victim GAN models on the CelebA dataset when the training data size is 512. As this table shows, LSGAN has the lowest value of the Jensen-Shannon score, so synthetic images of this model are more similar to real samples than the other ones.

Table 2. Jenson-Shannon divergence for different GAN models

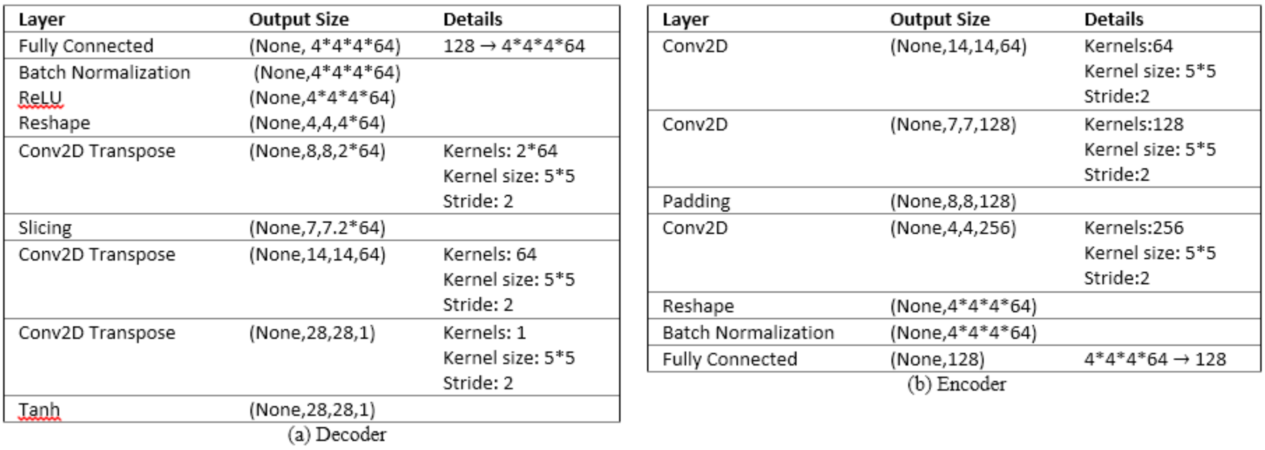
	WGANGP	LSGAN	DRAGAN
CelebA	0.1925±0.005	0.1793±0.0004	0.2285±0.001

5.1 Attack Model

The network structure of the attack model for MNITS and Fashion MNIST datasets are depicted in Figure 4. Also, Figure 5 shows the network structure of the attack model for CelebA datasets. As described, the decoder in the attack model is the same as the generator structure in the victim models. The learning rate and the number of training iterations are 0.001 and $4 * 10^5$, respectively. The batch size is set to 64, and 12800 synthetic samples are used to train the attack models.

5.2 Evaluation of the Proposed Attack Against Non-Private GANs

Figure 6, Figure 7, and Figure 8 show the attack accuracy of compared methods for different sizes of training data on MNIST, Fashion-MNIST, and CelebA datasets. In this experiment, the WGANGP victim model is used. As these figures show, the size of the training data is an important factor in overfitting the GAN models and so the accuracy of the membership inference attacks. The overfitting and the accuracy of the membership inference attacks decrease as the size of the training data increases. These figures also show that the LOGAN-accessible discriminator attack outperforms other attacks. It highlights the fact that in the GAN architecture, the training dataset directly impacts the discriminator. Therefore, when the attacker accesses the discriminator, she can conduct a more accurate attack. However, when only the generator is available, our attack outperforms the other attacks. According to Figure 6, Figure 7, and Figure 8, the proposed method, on average performs 20.05%, 11.09%,



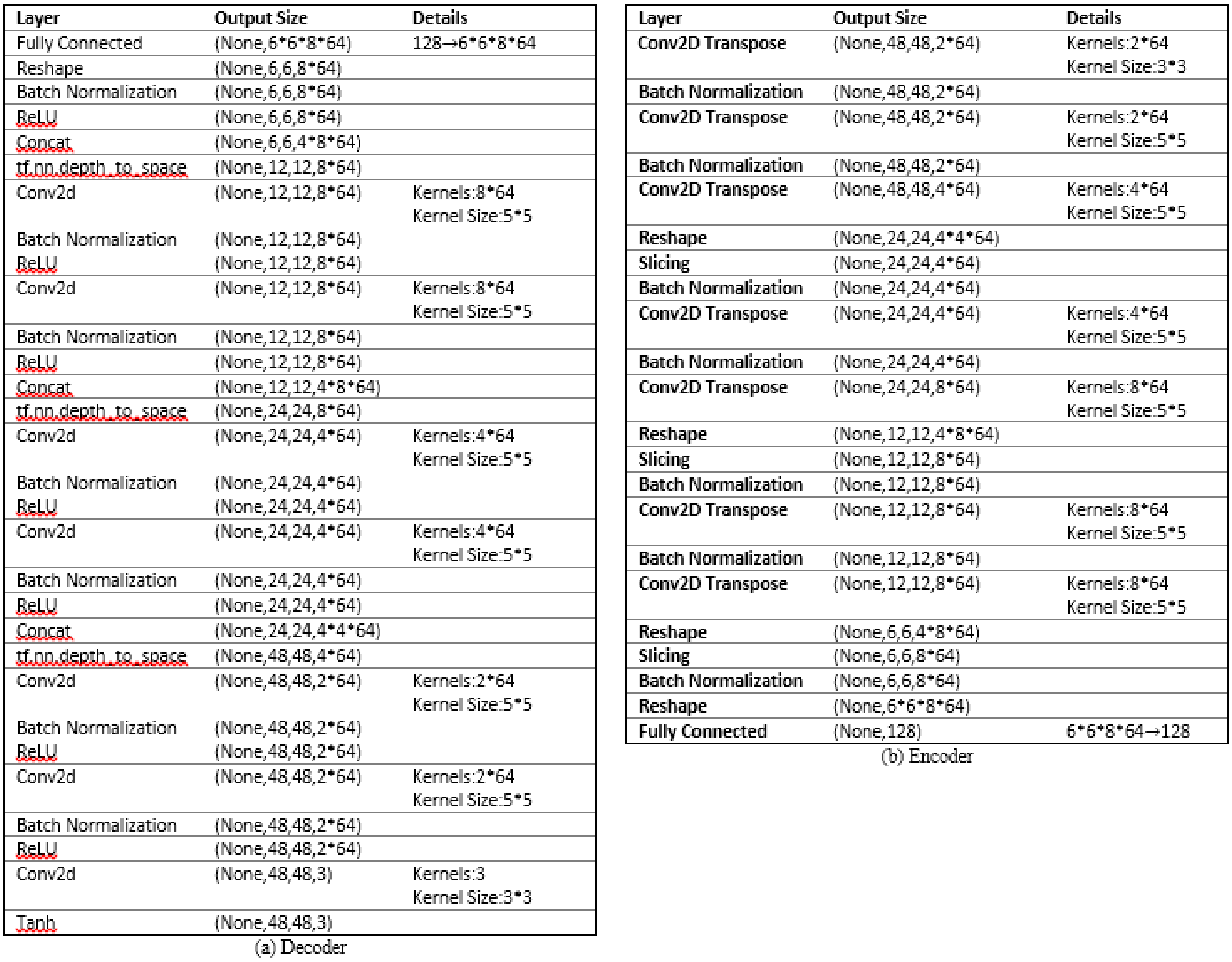
Layer	Output Size	Details
Fully Connected	(None, 4*4*4*64)	128 → 4*4*4*64
Batch Normalization	(None,4*4*4*64)	
ReLU	(None,4*4*4*64)	
Reshape	(None,4,4,4*64)	
Conv2D Transpose	(None,8,8,2*64)	Kernels: 2*64 Kernel size: 5*5 Stride: 2
Slicing	(None,7,7.2*64)	
Conv2D Transpose	(None,14,14,64)	Kernels: 64 Kernel size: 5*5 Stride: 2
Conv2D Transpose	(None,28,28,1)	Kernels: 1 Kernel size: 5*5 Stride: 2
Tanh	(None,28,28,1)	

(a) Decoder

Layer	Output Size	Details
Conv2D	(None,14,14,64)	Kernels:64 Kernel size: 5*5 Stride:2
Conv2D	(None,7,7,128)	Kernels:128 Kernel size: 5*5 Stride:2
Padding	(None,8,8,128)	
Conv2D	(None,4,4,256)	Kernels:256 Kernel size: 5*5 Stride:2
Reshape	(None,4*4*4*64)	
Batch Normalization	(None,4*4*4*64)	
Fully Connected	(None,128)	4*4*4*64 → 128

(b) Encoder

Figure 4. The structure of the attack model for MNIST and FASHION MNIST datasets



Layer	Output Size	Details
Fully Connected	(None,6*6*8*64)	128→6*6*8*64
Reshape	(None,6,6,8*64)	
Batch Normalization	(None,6,6,8*64)	
ReLU	(None,6,6,8*64)	
Concat	(None,6,6,4*8*64)	
tf.nn.depth_to_space	(None,12,12,8*64)	
Conv2d	(None,12,12,8*64)	Kernels:8*64 Kernel Size:5*5
Batch Normalization	(None,12,12,8*64)	
ReLU	(None,12,12,8*64)	
Conv2d	(None,12,12,8*64)	Kernels:8*64 Kernel Size:5*5
Batch Normalization	(None,12,12,8*64)	
ReLU	(None,12,12,8*64)	
Concat	(None,12,12,4*8*64)	
tf.nn.depth_to_space	(None,24,24,8*64)	
Conv2d	(None,24,24,4*64)	Kernels:4*64 Kernel Size:5*5
Batch Normalization	(None,24,24,4*64)	
ReLU	(None,24,24,4*64)	
Conv2d	(None,24,24,4*64)	Kernels:4*64 Kernel Size:5*5
Batch Normalization	(None,24,24,4*64)	
ReLU	(None,24,24,4*64)	
Concat	(None,24,24,4*4*64)	
tf.nn.depth_to_space	(None,48,48,4*64)	
Conv2d	(None,48,48,2*64)	Kernels:2*64 Kernel Size:5*5
Batch Normalization	(None,48,48,2*64)	
ReLU	(None,48,48,2*64)	
Conv2d	(None,48,48,2*64)	Kernels:2*64 Kernel Size:5*5
Batch Normalization	(None,48,48,2*64)	
ReLU	(None,48,48,2*64)	
Conv2d	(None,48,48,3)	Kernels:3 Kernel Size:3*3
Tanh	(None,48,48,3)	

(a) Decoder

Layer	Output Size	Details
Conv2D Transpose	(None,48,48,2*64)	Kernels:2*64 Kernel Size:3*3
Batch Normalization	(None,48,48,2*64)	
Conv2D Transpose	(None,48,48,2*64)	Kernels:2*64 Kernel Size:5*5
Batch Normalization	(None,48,48,2*64)	
Conv2D Transpose	(None,48,48,4*64)	Kernels:4*64 Kernel Size:5*5
Reshape	(None,24,24,4*4*64)	
Slicing	(None,24,24,4*64)	
Batch Normalization	(None,24,24,4*64)	
Conv2D Transpose	(None,24,24,4*64)	Kernels:4*64 Kernel Size:5*5
Batch Normalization	(None,24,24,4*64)	
Conv2D Transpose	(None,24,24,8*64)	Kernels:8*64 Kernel Size:5*5
Reshape	(None,12,12,4*8*64)	
Slicing	(None,12,12,8*64)	
Batch Normalization	(None,12,12,8*64)	
Conv2D Transpose	(None,12,12,8*64)	Kernels:8*64 Kernel Size:5*5
Batch Normalization	(None,12,12,8*64)	
Conv2D Transpose	(None,12,12,8*64)	Kernels:8*64 Kernel Size:5*5
Reshape	(None,6,6,4*8*64)	
Slicing	(None,6,6,8*64)	
Batch Normalization	(None,6,6,8*64)	
Reshape	(None,6*6*8*64)	
Fully Connected	(None,128)	6*6*8*64→128

(b) Encoder

Figure 5. The structure of the attack model for CelebA dataset

and 26.51% better than the GAN-Leak [3] white-box generator attack for MNIST, Fashion-MNIST, and CelebA datasets, respectively. So, it can be concluded that on average, the accuracy of the proposed attack is approximately 19% higher than similar work in our experiments. Figure 9 compares the execution time of the proposed attack with the GAN-Leak white box generator attack when the MNIST dataset is used. Since most of the execution time in the attack is related to encoder training, increasing the size of the attack set has little effect on the execution time. In contrast, the size of the attack set has a direct impact on the execution time of the related task, and the execution time increases as the attack set increases. So when the attack set size is larger than 512 (a common setting in attacks), the proposed attack not only provides higher accuracy but also runs in less time. It should be noted that the proposed attack and GAN-Leak white box generator attack are run in Tensorflow on a windows server with Intel Core i9, one GTX 2080 GPU, and memory with a size of 50 GB.

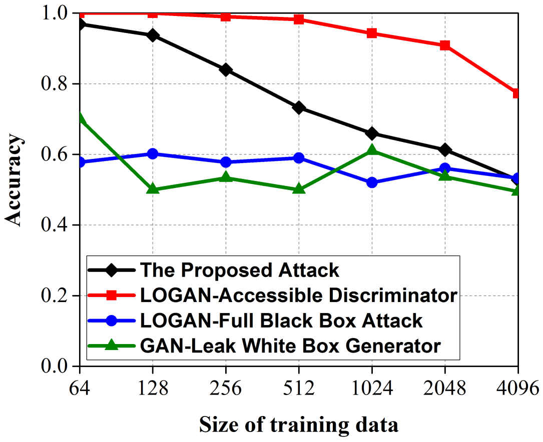


Figure 6. Comparison of different attacks' accuracy versus different sizes of training data using MNIST dataset

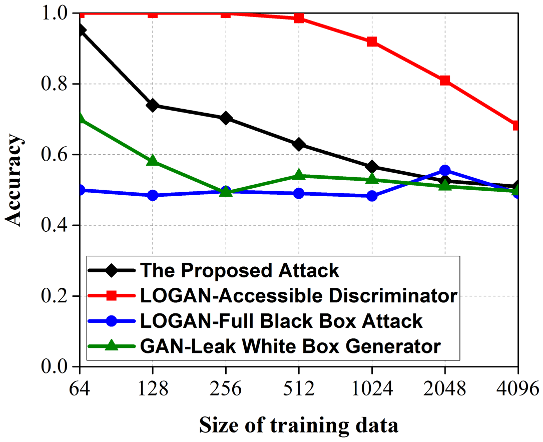


Figure 7. Comparison of different attacks' accuracy versus different sizes of training data using Fashion-MNIST dataset

As Figure 6, Figure 7, and Figure 8 show, the size of the training dataset in victim models is one of the important factors in the success of the attack. To examine the size of the dataset above which the proposed attack does not perform better than random guessing, the accuracy of the attack against the victim models with larger training data size is evaluated. Table 3 shows the accuracy of the proposed attack for different sizes of training data on the MNIST dataset for the WGAN-GP victim model. As this table shows, for the WGAN-GP victim model on MNIST dataset, when the data size is approximately larger than 15000, the attack cannot be better than random guessing (i.e., accuracy=0.5).

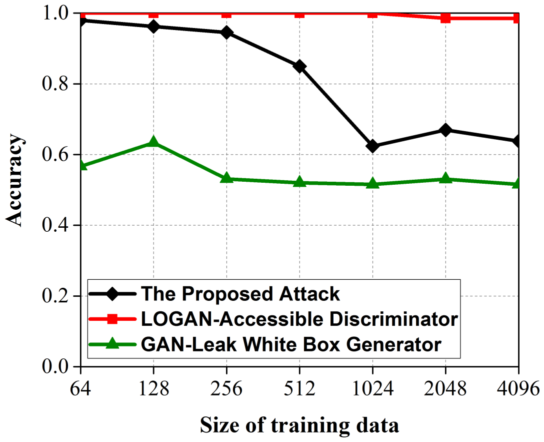


Figure 8. Comparison of different attacks' accuracy versus different sizes of training data using CelebA dataset

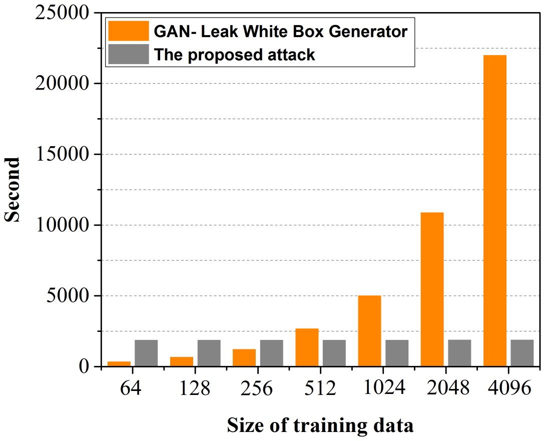


Figure 9. Comparison of different attacks' execution time versus different sizes of training data using MNIST dataset

Figure 10, Figure 11, and Figure 12 show the accuracy of membership inference attacks against different victim GAN models. These figures confirm the fact that, assuming the accessibility of the discriminator results in the most effective attack. Therefore, the LOGAN discriminator accessible attack outperforms the others. However, between the generator white-box attacks, our proposed attack outperforms GAN-leak[3] attack. These figures also show that LSGAN and WGANGP are the most and the least resistant

Table 3. Jenson-Shannon divergence for different GAN models

	5000	6000	7000	8000	9000	10000	15000	20000
Accuracy	0.5752	0.5628	0.5592	0.5471	0.5335	0.5273	0.5111	0.49731

GAN models against the membership inference attacks, respectively.

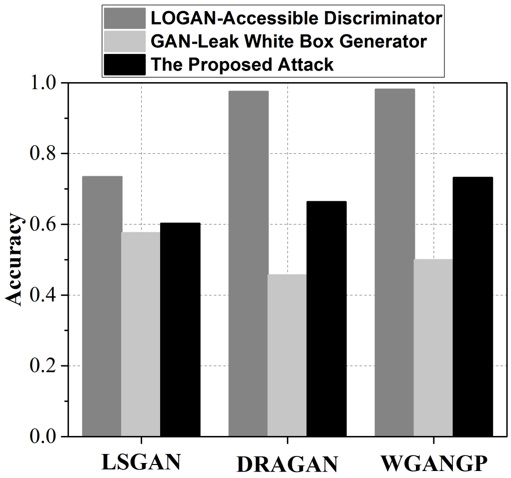


Figure 10. Comparison of the accuracy of different attacks on different victim models using MNIST dataset

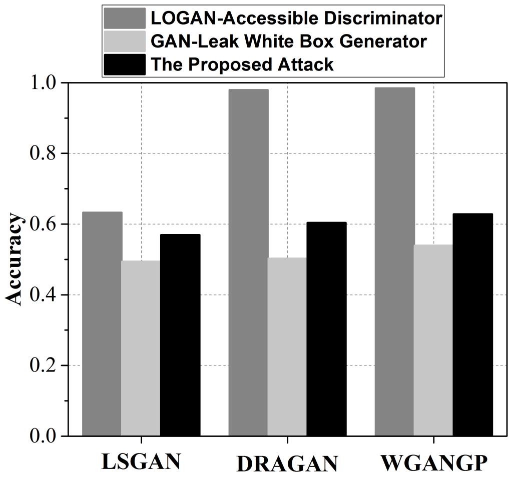


Figure 11. Comparison of the accuracy of different attacks on different victim models using Fashion-MNIST dataset

It is also appropriate to investigate the effect of hyper-parameters of the attack model on the accuracy of the attack. To do this, the experiments are conducted with different values of the total number of iterations (T), learning rate, and the total number of training samples (m) for the attack model, when the MNIST dataset is used. The attacks are conducted against the WGANGP victim model. Figure 13 shows the accuracy of the proposed attack for different values of the total number of training samples of the attacker model (m). In this experiment, the learning rate, the total number of iterations, and the batch size are set to 0.001, $4 * 10^5$, and 64, respectively. The attack is conducted against victim models with a different total number of training samples (N). As this figure shows, changing the number of synthetic samples used in the attack model's training (m) does not have a significant effect on the attack performance. Figure 14 shows the accuracy of the proposed attack versus different values of the total number of iterations (T) for training the attack model. In this experiment, the learning rate, the total number of attack model's training data,

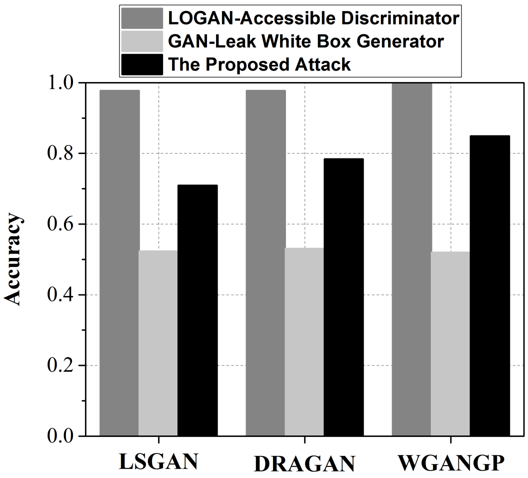


Figure 12. Comparison of the accuracy of different attacks on different victim models using CelebA dataset

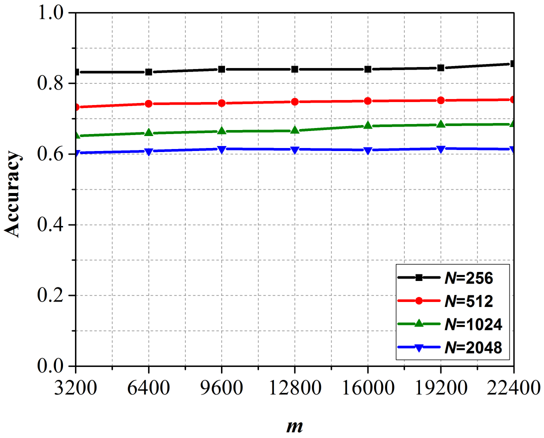


Figure 13. Proposed attacks' accuracy versus different sizes of attacker model's training data (m) for various training sizes of victim model (N) using MNIST dataset

and the batch size are set to 0.001, 12800, and 64, respectively. The attack is conducted against victim models with different sizes of training samples (N). As this figure shows, changing the number of iterations used in the attack model's training does not have a significant effect on the attack performance.

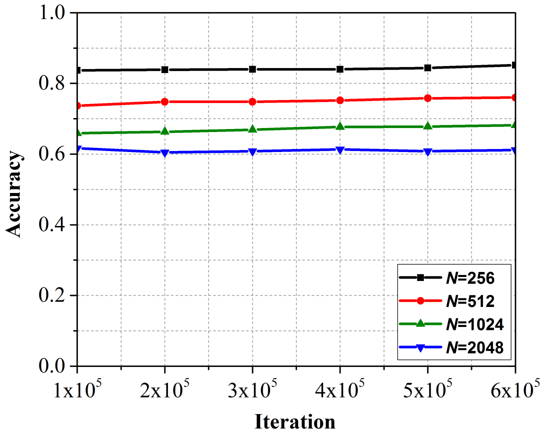


Figure 14. Proposed attack's accuracy versus the different number of training iterations of attacker model for various training sizes of victim model (N) using MNIST dataset

Figure 15 shows the accuracy of the proposed attack versus different values of the learning rate. In this experiment, the total number of iterations (T) in the training attack model, the total number of attack model's training data (m), and the batch size are set to $4 * 10^5$, 12800, and 64, respectively. The attack is conducted against victim models with different sizes of training samples (N). As this figure shows, an excessive increase in the learning rate drastically reduces the accuracy of the attack.

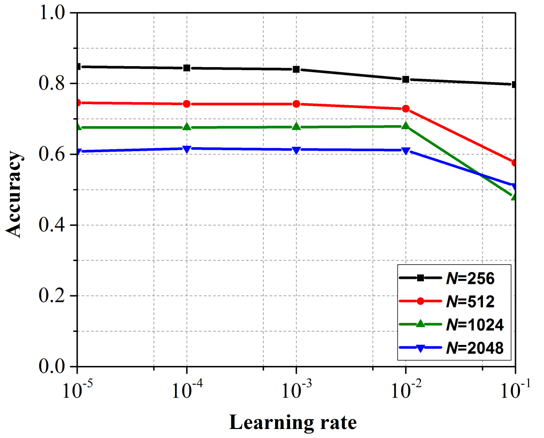
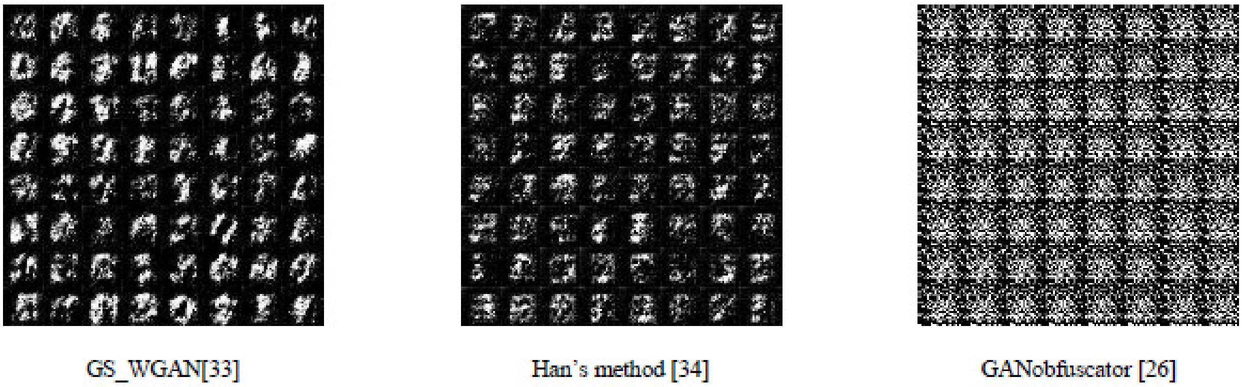


Figure 15. Proposed attack's accuracy versus different values of the learning rate for various training sizes of victim model (N) using MNIST dataset

5.3 Evaluation of the Proposed Attack Against Privacy-Preserving GANs

To evaluate the accuracy of the proposed attack against privacy-preserving mechanisms, three differential private mechanisms, i.e., GANobfuscator [26], GS-WGAN[33], and the method proposed by Han and Xue [34] are investigated. GANobfuscator [26] is a privacy-preserving mechanism that provides a differential privacy guarantee for both the discriminator and the generator networks. However, GS-WGAN [33] and the method proposed by Han and Xue[34] only provide differential privacy for the generator network. In this experiment, WGANGP [39] with network architecture similar to [42], is used as victim model. The victim model is trained on the MNIST dataset. The learning rates of the discriminator and the generator are set to $5 * 10^{-5}$. The number of iterations on the discriminator and the generator are 4, and $1 * 10^5$, respectively. The coefficient of gradient penalty (λ) has the value of 10 and the batch size (m) is set to 64. The hyperparameters of Adam optimizer, i.e. β_1 and β_2, are set to 0.5 and 0.9. For GANobfuscator[26], the training dataset is split into publicly available data and private data, by a ratio of 2 to 98, respectively. The public data is used for adaptive clipping. In GS-WGAN [33], centralized training with one discriminator is used. In Han's method [34], their proposed adaptive algorithm is used to calculate the clipping bound of discriminator loss. The size of the training dataset is set to 512. The parameters of differential privacy, i.e. confidence parameter δ and privacy budget (ϵ), are set to 10^{-5} and 50, respectively. Figure 16 shows the random synthetic images corresponding to different mechanisms, and Table 4 shows the FID for different victim GAN models. As Figure 16 shows, the synthetic images generated by these mechanisms are of low quality. Also, as Table 4 shows, among the privacy-preserving mechanisms, GS-WGAN[33] generates higher quality images, and the values of FID are very high compared to their non-private counterparts ($FID = 95.59$ in Table 1). This is because the variance of noise in differential private mechanisms is a decreasing function of the number of training samples. Therefore, privacy-preserving techniques for GANs are not yet practically efficient. Moreover, when the number of training samples is small (which is the case where most attacks are effective), the variance of noise will be significant to provide differential privacy and will cause poor quality samples which makes the model useless. It should be noted that a reasonable amount for a privacy budget is less than 11, but since a smaller amount of privacy budget (more privacy) leads to lower quality images, the results for $\epsilon = 50$ are presented. Table 5 shows the accuracy of the proposed membership inference attack against different



GS_WGAN[33]

Han's method [34]

GANobfuscator [26]

Figure 16. Synthetic generated images of different privacy-preserving mechanisms with $\epsilon = 50$ and $\delta = 10^{-5}$ on MNIST dataset assuming that the size of training dataset is 512

Table 4. FID measure for different privacy-preserving mechanisms

	GS-WGAN[33]	Han's method [34]	GANobfuscator [26]
FID	278.41	304.28	354.36

Table 5. Proposed attack's accuracy for different privacy-preserving mechanisms on MNIST dastset

	GS-WGAN[33]	Han's method [34]	GANobfuscator [26]
Accuracy	0.4921	0.4550	0.4238

privacy-preserving mechanisms. As this table shows, the proposed membership inference attack cannot do better than random guessing. The reason is the low-quality synthetic images as discussed above.

6 Conclusion

Publishing generators in GAN models to generate synthetic samples is common in practice. But, these models may leak information about their training data. Therefore, a privacy risk assessment of these models has been attended recently. In this paper, a white-box generator attack against the privacy of GAN models has been proposed. In this attack, an auto-encoder is trained, where the decoder architecture is the same as the generator, and its parameters are set to the generator's parameters. The cost values of the trained auto-encoder are used to separate training members from non-members.

The proposed attack has been evaluated concerning various GAN models and training configurations. The results demonstrate that the proposed attack outperforms GAN-leak [3], the only related white-box generator attack. But LOGAN discriminator accessible attack outperforms the proposed attack, which highlights the fact that in the GAN architecture, the training dataset directly impacts the discriminator. Therefore, the attacker can conduct attacks more accurate attacks by accessing the discriminator. The proposed attack has also been evaluated against privacy-preserving GANs that are using differential private mechanisms. The results show that due to the low quality of the generated images in small training datasets, the attack against them is ineffective. A comprehensive evaluation of the proposed attack on diverse generative models, and datasets, especially non-image datasets, can be regarded in future research. Also, an evaluation of the proposed attack against empirical privacy-preserving GANs is suggested as future work.

Acknowledgment

The authors would like to acknowledge the financial support of Information and Communication Technology Park for this project under grant number 01-99-02-000157. Also, the authors would like to acknowledge the financial support of the Ministry of Science, Research and Technology for this project under grant number 01-99-02-000157.

References

[1] R. Shokri, M. Stronato, C. Song and V.Shamatikov. Membership Inference Attacks Against Machine Learning Models. In IEEE Symposium on Security and Privacy (SP), pages 1–16. 2017.

[2] I. Goodfellow, J.Pougget-Abadie, M. Mirza, B. Xu, D. Warde-Farely, S. Ozair, A. Courvalle and Y. Bongio. Generative Adversarial Nets. In 27th International Conference on Neural Information Processing Systems, pages 2672-2680. 2014.

[3] D. Chen, N. Yu, Y. Zhang and M. Fritz. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. In the 2020 ACM SIGSAC Conference on Computer and Communications Security, pages 343-362. 2020.

[4] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses

on Machine Learning Models. In 26th Annual Network and Distributed System Security Symposium, pages 1-16. 2019.

[5] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter and K. Chen. Understanding Membership Inference in Well-Generalized Learning Models. arXiv:1802.04889. 2018.

[6] S. Yeom, I. Giacomelli, M. Fredrikson and S. Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In 2018 IEEE 31st Computer Security Foundations Symposium,pages 268-282. 2018.

[7] Y. Kaya, S. Hong, and T. Dumitras. On the Effectiveness of Regularization against Membership Inference Attacks. arXiv preprint arXiv:2006.05336. 2020.

[8] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier and H. Jegou. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In Proceedings of the 36th International Conference on Machine Learning, pages 1–11. 2019.

[9] Z. Li and Y. Zhang. Membership Leakage in Label-Only Exposures. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pages 880–895. 2021.

[10] C. A Choquette-Choo, F. Tramer, N. Carlini and N. Papernot. Label-only Membership Inference Attacks. In International Conference on Machine Learning, pages 1964–1974. 2021.

[11] Y. Long, L. Wang, D. Bu, V. Bindschaedler, X. Wang, H. Tang, C. A. Gunter, and K. Chen. A Pragmatic Approach to Membership Inferences on Machine Learning Models. In 2020 IEEE European Symposium on Security and Privacy, pages 521–534. 2020.

[12] S. Rezaei and X. Liu. On the Difficulty of Membership Inference Attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7892–7900. 2021.

[13] B. Hui, Y. Yang, H. Yuan, P. Burlina, N. Zhenqiang Gong, and Y. Cao. Practical Blind Membership Inference Attack via Differential Comparisons. In Network and Distributed Systems Security Symposium, pages 1–17. 2021.

[14] M. Nasr, R. Shokri and A. Houmansadr. Comprehensive Privacy Analysis of Deep Learning Standalone and Federated Learning under Passive and Active White-box Inference Attacks. In IEEE Symposium on Security and Privacy, pages 739–853. 2019.

[15] S. Kumar Murakonda, Reza Shokri. ML Privacy Meter: Aiding Regulatory Compliance by Quantifying the Privacy Risks of Machine Learning. In https://arxiv.org/abs/2007.09339. 2020.

[16] K. Leino and M. Fredrikson. Stolen Memories: Leveraging Model Memorization for Calibrated White-box Membership Inference. In 29th USENIX Security Symposium (USENIX Security 20), pages 1605–1622. 2021.

[17] J. Hayes, L. Melis, G. Denerzis and E. De Cristofaro. Stolen Memories: LOGAN: Membership Inference Attacks against Generative Models. In Proceedings on Privacy Enhancing Technologies, vol. 2019, no. 1, pages 133–152. 2019.

[18] B. Hilprecht, M. Harterich, and D. Bernau. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. In Proceedings on Privacy Enhancing Technologies, vol. 4, pages 232–249. 2019.

[19] K. S. Liu, C. Xiao, B. Li, and J. Gao. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. In Proceedings on Privacy Enhancing Technologies, vol. 4, pages 232–249. 2019.

[20] K. S. Liu, C. Xiao, B. Li, and J. Gao. Membership Inference Attacks against GANs by Leveraging Over-representation Regions. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pages 2387–2389. 2021.

[21] X. Liu, Y. Xu, S. Mukherjee, J. L. Ferres. MACE: A Flexible Framework for Membership Privacy Estimation in Generative Models. arXiv:2009.05683. 2020.

[22] R. Webster, J. Rabin, L. Simon, and F. Jurie. This Person (Probably) Exists. Identity Membership Attacks Against GAN Generated Faces. arXiv preprint arXiv:2107.06018. 2021.

[23] J. Zhou, Y. Chen, C. Shen, and Y. Zhang. Property Inference Attacks againts GANs. arXiv preprint arXiv:2111.07608. 2021.

[24] R. Torkzadehmahani, P. Kairouz, and B. Paten. DP-CGAN: Differentially private synthetic data and label generation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),pages1–8. 2021.

[25] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou. Differentially private generative adversarial network. arXiv preprint arXiv:1802.06739. 2018.

[26] C. Xu, J. Ren, D. Zhang, Y. Zhang, Z. Qin, and K. Ren. GANobfuscator: Mitigating information leakage under GAN via differential privacy. IEEE Transactions on Information Forensics and Security, vol. 14, no. 9, 2019, pages 2358–2371. 2019.

[27] X. Zhang, S. Ji, T. Wang. Differentially private releasing via deep generative model. arXiv preprint arXiv:1801.01594. 2018.

[28] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and

Communications Security,pages 308-318. 2016.

[29] J. Jordon, J. Yoon, and M. Schaar. PATE-GAN: Generative synthetic data with differential privacy guarantees. In Seventh International Conference on Learning Representations,pages 1–21. 2019.

[30] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private aggregator. In Proceedings of the International Conference on Learning Representations (ICLR),pages 1–17. 2017.

[31] Y. Long, S. Lin, Z. Yang, C. A. Gunter, and B. Li. Scalable differentially private generative student model via PATE. In arXiv preprint arXiv:1906.09338. 2019.

[32] B. Wang, F. Wu, Y. Long, L. Rimanic, C. Zhang, and B. Li. DataLens: Scalable privacy preserving training via gradient compression and aggregation. arXiv preprint arXiv: arXiv:2103.11109. 2021.

[33] D. Chen, T. Orekondy, and M. Fritz. GS-WGAN: A gradient-sanitized approach for learning differentially private generators. In 34th Conference on Neural Information Processing Systems, NeurIPS,pages 1–12. 2020.

[34] C. Han, and R. Xue. Differentially private GANs by adding noise to discriminator's loss. Computer and Security, vol. 107, pages 1–14. 2021.

[35] M. Nasr, R. Shokri, and A. Houmansad. Machine learning with membership privacy using adversarial regularization. In the ACM SIGSAC Conference on Computer and Communications Security, pages 634–646. 2018.

[36] S. Mukherjee, Y. Xu, A. Trivedi, and J. Ferres. PrivGan: protecting GANs from membership inference attack at low cost. In Proceedings on Privacy Enhancing Technologies, pages 142–163. 2021.

[37] W. Hui Wang, H. Gao, and X. Shi. PAR-GAN: Improving the Generalization of Generative Adversarial Networks Against Membership Inference Attacks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 127–137. 2021.

[38] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein Generative Adversarial Networks. In International Conference on Machine Learning, pages 214–223. 2017.

[39] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and Aaron C. Courville. Improved Training of Wasserstein GANs. In Annual Conference on Neural Information Processing Systems (NIPS), pages 5767–5777. 2017.

[40] N. Kodali, J. Hays, J. Abernethy, Z. Kira. On Convergence and Stability of GANs. In ICLR 2018 Conference Blind Submission, pages 1–18. 2018.

[41] X. Mao, Q. Li, H. Xie, R. Y.K. Lau, Z. Wang, and S. P. Smolley. Least Squares Generative Adversarial Networks. In 2017 IEEE International Conference on Computer Vision, pages 1–17. 2017.

[42] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In preprint arXiv:1511.06434. 2015.

[43] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Annual Conference on Neural Information Processing Systems (NIPS),pages 6626–6637. 2017.

Maryam Azadmanesh received her B.Sc. degree in Information Technology (IT) Engineering from the University of Birjand in 2010 and M.Sc. degree in Information Technology Engineering from the Sharif University of Technology in 2012. She is currently a Ph.D. student in IT Engineering (Information Security) at the University of Isfahan, Isfahan, Iran. Her research interests include data privacy and security, differential privacy and machine learning.

Behrouz Shahgholi Ghahfarokhi received a B.Sc. degree in computer engineering, an M.Sc. degree in artificial intelligence, and a Ph.D. degree in computer architecture from the University of Isfahan, Iran in 2004, 2006, and 2011, respectively. He joined the University of Isfahan in 2011 and is currently an Associate Professor with the Faculty of Computer Engineering. His research interests are computer networks, network security, and intelligent systems.

Maede Ashouri-Talouki is an associate professor in the IT Engineering department of the University of Isfahan (Iran). She received her B.Sc. degree and M.Sc. degree in Computer Engineering from the University of Isfahan (Iran) in 2004 and 2007, respectively. In 2012, she received her Ph.D. degree at the University of Isfahan in computer engineering. In 2013, she joined the University of Isfahan (Iran). Her research interests include IoT security, cloud access control, user privacy and anonymity, cryptographic protocols and network security.