The ISC Int'l Journal of Information Security
ISeCure

January 2025, Volume 17, Number 1 (pp. 43–57)
http://www.isecure-journal.org

Detection of Membership Inference Attacks on GAN Models

Ala Ekramifard [1], Haleh Amintoosi [1,*], and Seyed Amin Hosseini Seno [1]
[1] *Computer Engineering Department, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran*

ARTICLE INFO.

Article history:
Received: June 7, 2024
Revised: October 12, 2024
Accepted: December 25, 2024
Published Online: December 31, 2024

Keywords:
Machine Learning, Privacy, Generative Adversarial Network, Membership Inference Attacks

Type: Research Article

doi: 10.22042/isecure.2024.461639.1131

ABSTRACT

In the realm of machine learning, Generative Adversarial Networks (GANs) have revolutionized the generation of synthetic data, closely mirroring the distribution of real datasets. This paper delves into the privacy concerns associated with GANs, mainly focusing on Membership Inference Attacks (MIAs), which aim to determine if a specific record was used in training a model. Such attacks pose significant privacy risks, especially when sensitive data is involved. To combat this, we propose a novel detector model to identify and thwart MIAs within GANs. Our model, which operates as an additional layer of protection for Machine Learning as a Service (MLaaS) providers, leverages outputs from both the discriminator and generator to ascertain the membership status of data samples. We introduce two variants of the detector model—supervised and unsupervised—based on the availability of information from the discriminator. The supervised detector employs labeled data for training, while the unsupervised detector uses anomaly detection techniques. Also, an image detector uses the generator's output to identify potential adversary samples. Our experimental evaluation spans various GAN architectures and datasets, ensuring the robustness and generalizability of our approach. The paper also analyzes the impact of dataset size on the detector's effectiveness. Integrating our detector allows MLaaS providers to enhance privacy safeguards, effectively balancing model utility with data protection.

© 2025 ISC. All rights reserved.

1 Introduction

Machine learning (ML) models have emerged as powerful tools for discerning patterns within vast datasets. Among these, deep learning, characterized by neural networks with three or more layers, seeks to emulate human brain behavior. The remarkable progress in deep learning has facilitated diverse applications, including discriminative models employed for classification tasks and generative models responsible for creating realistic outputs, e.g., imagery.

In recent years, generative models have garnered significant interest. These models generate synthetic samples by capturing the underlying distribution of a given training dataset. Among them, the Generative Adversarial Network (GAN) stands out as the most widely adopted. GANs excel at producing synthetic data that closely aligns with the distribution of the original dataset. The GAN architecture comprises two neural networks, the generator and the discriminator, engaged in a competitive training process. While the generator endeavors to deceive the discriminator by generating counterfeit data, the discriminator's role

* Corresponding author.
Email addresses: ekramifard@um.ac.ir, amintoosi@um.ac.ir, hosseini@um.ac.ir
ISSN: 2008-2045 © 2025 ISC. All rights reserved.

is to differentiate genuine data from the fabricated counterpart.

Prominent technology providers, including Google, Microsoft, and Amazon, offer Application Programming Interfaces (APIs) that empower their customers to seamlessly integrate machine learning (ML) tasks into their services. One such paradigm, Machine Learning as a Service (MLaaS), provides ML engines as integral components within cloud computing environments. MLaaS not only streamlines business operations but also facilitates model training for researchers, even without domain-specific expertise.

Training ML models, especially in commercial contexts, raises significant concerns regarding sensitive data leakage and privacy breaches. ML models tend to memorize information from their training data, inadvertently enabling malicious actors to recover sensitive data employed during model training. This vulnerability exposes ML models to various privacy attacks, including Model Inversion Attack, Model Extraction Attack, and Membership Inference Attack (MIA). MIA focuses on determining whether a specific record was part of the training data for a victim ML model. Such attacks pose substantial privacy risks to individuals. For instance, suppose a patient's health data contributes to training a model specifically designed for disease prediction. If we discover this association, we can infer that the patient likely suffers from the disease in question.

While MLaaS offers remarkable convenience and accessibility, safeguarding privacy remains paramount. Researchers and practitioners must address these challenges to achieve a delicate balance between model utility and individual data protection.

In recent years, research efforts have extensively explored methods to attack and defend against membership inference attacks (MIA) in machine learning models. However, these investigations have primarily focused on broader strategies without precisely pinpointing the data instances that elevate the risk of privacy leakage. Consequently, there is an urgent need for a solution capable of detecting inference attacks and proactively preventing privacy breaches.

The landscape of visual generation has witnessed remarkable progress, primarily driven by the widespread adoption of Generative Adversarial Networks (GANs) and their expanding applications in artificial intelligence tools, including DALL·E. As this growth trajectory continues, it becomes imperative to underscore the significance of privacy preservation. Notably, inference attacks targeting generative models pose unique challenges, and the existing body of work in this domain remains relatively sparse.

In this article, we specifically address the issue of membership inference attacks within the context of generative models. While prior research has explored regularization techniques and differential privacy mechanisms to fortify GANs against MIA, these methods often come at the cost of generating lower-quality synthetic samples. Furthermore, none of the existing studies have proposed a dedicated detector model explicitly designed to thwart membership inference attacks.

Our proposed model aims to bridge this gap. By leveraging insights from the increasing adoption of GANs and recognizing the criticality of identifying data samples that could compromise privacy, we propose a novel detector. This model is an additional protective layer for Machine Learning as a Service (MLaaS) providers offering generative models. By integrating our detector, service providers can selectively restrict access to the target model's outputs when a generated sample is flagged as a potential privacy threat.

The primary contributions of our research are as follows:

(1) Novel Detector Model for Membership Inference Attacks on GANs:
   - We propose an innovative detector model designed to identify membership inference attacks within Generative Adversarial Networks (GANs).
   - Our model leverages the discriminator and generator outputs to determine the membership status of a given sample in the target model's training dataset.

(2) Model Variants Based on Knowledge Availability:
   - We explore different model variants based on the extent of information accessible from the discriminator.
   - Specifically, we introduce two detector models:
     - Supervised Detector: This model utilizes labeled data to train a classifier, enabling it to discern membership status effectively.
     - Unsupervised Detector: Leveraging unsupervised learning techniques, this model detects anomalies in the discriminator output without relying on labeled data.
   - Additionally, our image detector exploits the generator's output to identify potential adversary samples.

(3) Experimental Evaluation:
   - We conduct extensive experiments to as-

sess the performance of our proposed detector model across various GAN architectures, including Deep Convolutional GANs (DCGANs) and Variational Autoencoders (VAEs).
- Diverse datasets, such as CIFAR-10, LFW, and CelebA, are employed to ensure the generalizability of our approach.

(4) Impact Analysis of Key Factors:
- We analyze the influence of dataset size as a critical factor on the effectiveness of our membership inference detector.

The subsequent sections of this paper are organized as follows. In Section 2, we delve into the existing literature, specifically reviewing research related to membership inference attacks and generative models. Section 3 provides an in-depth exposition of our proposed detector model. We elucidate its design, functionality, and underlying principles. In Section 4, we present and analyze the outcomes of our extensive experimental evaluation. These results offer insights into the performance and efficacy of our detector across various GAN models and datasets. Finally, Section 5 encapsulates our findings, discusses implications, and outlines potential avenues for future research.

2 Background

This section begins with a discussion of generative models, followed by an overview of membership inference attacks.

2.1 Generative Model

Machine Learning (ML) algorithms primarily fall into two categories: supervised learning and unsupervised learning. A supervised ML model is trained on a labeled dataset, comprising a feature vector x and the label y. The function $f(x, \theta)$ representing the ML model, accepts a data sample x and outputs $y' = f(x, \theta)$, where θ denotes the parameter of the ML model that is trained with the dataset D_{train}. A trained model can predict the label of unseen data as a prediction vector of probabilities. The class with the highest probability is selected as the label for the input sample.

An unsupervised ML model, on the other hand, extracts patterns from data, whether labeled or unlabeled, without relying on the labels. Generative models, a subset of unsupervised ML, aim to generate new samples from the underlying data distribution. The Generative model, a type of deep neural network, approximates the probability distribution of the dataset. There exist two types of generative models, namely, Generative Adversarial Networks (GANs) [1–4] and Variational AutoEncoders (VAEs) [5, 6]. GANs strive to minimize the difference between fake and real samples, while VAEs generate novel data.

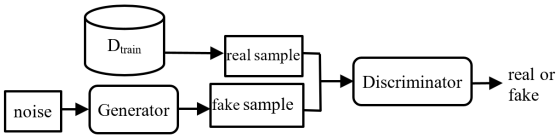

Figure 1. Architecture of GANs

Generative Adversarial Networks (GANs) are designed to generate new samples that closely approximate the distribution of the training dataset [1]. A GAN comprises two neural networks: a generator G and a discriminator D. The generator receives a latent vector z and generates a synthetic sample $G(z)$. The discriminator D receives both real samples from the training dataset and synthetic samples $G(z)$, and subsequently attempting to discern the difference between them. The models G and D are trained in a competitive network, with G learning to generate realistic samples to deceive D, as illustrated in Figure 1.

The Variational AutoEncoder (VAE) comprises of two neural networks: an encoder and a decoder [5, 6]. The encoder receives the input data and maps it into a latent space. In contrast, the decoder maps the encoded latent variable sampled back to the data space with minimal reconstruction error. Upon completion of the training, the decoder generates new data samples that closely approximate the training data distribution. VAEGAN [7] represents a fusion of VAE and a GAN, where the VAE decoder and the GAN generator are trained in unison.

Similar to other neural networks, GANs and VAEs employ Stochastic Gradient Descent (SGD) for training. SGD endeavors to find optimal parameters via

$$min_{\theta_G}\ max_{\theta_D}\ E_{x\in P_{data}}\ [\log(D_{\theta_D}(x))] \\ E_{z\in P_z}[\log(1 - D_{\theta_D}(G_{\theta_G}(z)))]. \quad (1)$$

In this context, θ_G and θ_D represent the parameters of the generator and the discriminator, respectively. P_{data} denotes the distribution of the real data samples, while P_z signifies the distribution of the latent code. The discriminator D is trained to distinguish synthetic samples from real ones, and the generator G is trained to generate new samples that closely resemble real data.

2.2 Membership Inference

Inference attacks pose a significant threat to the privacy of individual data within an aggregate genomic dataset [8]. In the context of a Membership Inference Attack (MIA), an adversary can conjecture whether a sample data has been utilized in the training pro-

cess of a classifier, based on the output probability vector of the Machine Learning (ML) model. The pioneering membership inference attack against machine learning models, proposed by Shokri *et al.* [9], incorporates multiple shadow models to emulate the behavior of the target model. The authors trained the inference attack as a binary classification model. For each input data, there exist two classes: class IN, which refers to the data in the training set, and class OUT, which refers to the data not present in the training set. The adversary trains an attack model on the labeled output of shadow models.

The MIA attack is categorized into black-box and white-box inference attacks based on varying degrees of adversarial knowledge. The adversary can access the training parameters of the target model, knowledge about data distribution, and the output of the target model. In black-box settings, the attacker only has access to the prediction vector computed by the target model. Conversely, in the white-box setting, the attacker has unrestricted access to the parameters of the target model.

When a GAN model is trained on a dataset containing sensitive attributes—such as personal identifiable information (PII), personal health records, behavior data, location information, financial data, or demographic details—it raises significant concerns about privacy violations. Sensitive attributes can include gender, race, age, medical conditions, income, and more. If the model inadvertently exposes or leaks information about these attributes, it may result in severe privacy breaches. These concerns are particularly relevant in scenarios involving membership inference attacks (MIAs), attribute inference attacks, and other model-based privacy attacks. To protect against sensitive data leakage and ensure privacy, our method can serve as a protective layer by implementing several key strategies to minimize the risk of exposing private data. These strategies are designed to guard against membership inference attacks that target models trained on sensitive data.

2.3 Related Work

The exploration of Membership Inference Attacks (MIAs) was initially conducted on classification models, where the adversary had the privilege of accessing the prediction vector of the Machine Learning (ML) model. The primary focus of these investigations was on deep learning models, which are more susceptible to overfitting, a characteristic that can be exploited by inference attacks. Further research extended the scope of these attacks to generative models [10], which are unsupervised learning models devoid of any prediction vector.

Shokri *et al.* [9] proposed the concept of shadow models, designed to emulate the behavior of the target model and generate training data for the attack model. The attack model was subsequently trained on the predictions of the shadow model, which comprised two distinct clusters. The label "IN" was used to denote data in the training dataset, while the label "OUT" was used to represent the contrary. Contrary to previous assumptions, Salem *et al.* [11] demonstrated the feasibility of MIAs with a single shadow model. They also formulated a threshold function to determine whether a sample is a member of the training set.

Li and Zhang [12] presented an alternative approach to the attack, which could be executed with just the prediction label of the target model, eliminating the need for the entire vector. They introduced the label-only MIA, which exhibited robust performance and highlighted the risks associated with membership privacy in machine learning models. In their methodology, the adversary assembles a shadow dataset, mirroring the distribution of the target model's training set. The adversary then queries the target model to obtain predicted labels. These labels, in conjunction with the shadow dataset, are used to train a shadow model that replicates the target model's behavior.

Nasr *et al.* [13] introduced the concept of a white-box Membership Inference Attack (MIA), where the adversary is privy to the parameters of the victim model. They demonstrated that the accuracy of a white-box attack surpasses that of a black-box attack. Leino and Fredrikson [14] constructed a Bayes-optimal white-box attack, which does not necessitate access to the training dataset of the victim model.

In Federated Learning (FL), a model is trained in a distributed process [4, 15], where parties train the model without divulging their data. Inference attacks in FL can transpire during the training process. Nasr *et al.* [13] elucidated how an attacker can infer the membership data of the dataset in FL. The adversary launches an attack on the Stochastic Gradient Descent (SGD) algorithm by modifying the local parameters and augmenting the loss. Truex *et al.* [16] demonstrated that the adversary can partake in federated learning and execute the inference attack. In this model, parties train their local model and share prediction probabilities with other parties when inferring a new sample. The discrepancies in prediction can leak membership information due to the inclusion of underlying training data.

MIAs on generative models present more significant challenge than on supervised learning because the adversary lacks access to the prediction vector of the victim generative models. MIAs on generative

models infer whether a data sample has been used in training the Generative Adversarial Network (GAN) model. The first membership inference attack on generative models was proposed by Hayes *et al.* [10]. The adversary inputs N member data samples and N non-member data samples, which they aim to distinguish, then sorts the probability values of the discriminator output vector and selects N data with the highest probability value as members. Conversely, in a black-box attack, the adversary generates a mimic GAN model and executes an attack analogous to the white-box setting. They evaluated various target models, such as DCGAN, DCGAN+VAE, and BEGAN, with DCGAN serving as the attacker model, and demonstrated that their white-box attack can function as a detector of overfitting in GAN and appropriate secure models that do not leak information about their training dataset.

Hilprecht *et al.* [17] proposed two distinct attacks against Generative Adversarial Network (GAN) and Variational Autoencoder (VAE) models. In black-box settings, they employed a Monte Carlo attack to generate synthetic data that closely resembles the training dataset. They utilized DCGAN and VAE as target models to evaluate their attack, reporting that VAE is more susceptible to Membership Inference Attacks (MIAs) than GAN.

Liu *et al.* [18] introduced a novel attack model and trained an attacker network to identify the seed required to regenerate the data sample x. The discrepancy between the generated data and x is used to infer whether x is part of the training dataset. Furthermore, Liu *et al.* [19] proposed a membership attack that infers if a bundle of n data samples was part of the training dataset. The attacker trains a neural network to generate the latent variable m as input for the victim generator model, which generates output $G(m)$ that aligns with sample x. The fundamental premise is that the attacker can generate data samples like x if x is part of the training set.

Chen *et al.* [20] presented various attack models in different settings based on the adversary's knowledge of the victim model. The adversary's objective is to generate a data sample closest to x. The distance between the generated sample and x is used to calculate the probability of data sample x being part of the training set. The generator can produce samples similar to members' data in this attack. They selected PGGAN, WGANGP, DCGAN, and MEDGAN as the victim models and demonstrated that the attacker's knowledge about the target model influences the vulnerability of victim models against attacks.

Hailong *et al.* [21] introduced a new attack that infers a sample as a member of training samples with high precision. In this method, all samples are clustered in the k region. If a new data sample x falls within a region where most of its data belongs to a training set, the adversary infers x as a member sample.

Azadmanesh *et al.* [22] proposed a white-box membership inference attack targeting GANs. They trained an auto-encoder to distinguish between member and non-member data by using the synthetic samples generated by the victim model and setting the decoder parameters in the auto-encoder based on the victim model's parameters. The results showed that the proposed attack outperforms GAN-leak.

Cavasin *et al.* [23] proposed an Identity Inference Against GAN (IIA) attack on fingerprint datasets generated by a generative adversarial network. It trains a shadow GAN to mimic the target model, generating samples that closely resemble those produced by the target. The discriminator then classifies whether a sample was generated by the target or the mimic model. The confidence values outputted by the discriminator allow sorting the samples in the query dataset. Ideally, the top k samples will have a higher likelihood of being members of the training set. This attack highlights the primary vulnerabilities of GANs regarding privacy issues in biometric data.

Alshantti *et al.* [24] implemented reconstruction attacks on tabular synthetic data across various domains such as healthcare, finance, and cybersecurity. They adapted the attack using existing membership inference techniques, demonstrating the potential privacy risks associated with using generative adversarial networks to create tabular synthetic datasets. The method employs the k-nearest neighbors' technique on generated data to identify the closest neighbors from each data sample in the dataset.

Overfitting is a critical factor in the success of MIAs. To mitigate the overfitting of target models, regularization techniques can be employed to defend against MIAs. Various methods have been proposed, including dropout [25], data augmentation, L2-norm regularization, label smoothing [26], model stacking, and adversarial regularization [27]. Notably, dropout has proven to be an effective method to defend against inference attacks on generative models [1, 17].

Differential Privacy (DP) provides a robust mechanism for safeguarding dataset privacy, offering a theoretical assurance against Membership Inference Attacks (MIAs). When a target model is trained utilizing a DP methodology, it refrains from memorizing any specific details about the data. Differential privacy has been employed as a defensive technique against inference attacks on classification and genera-

tive models [10, 20, 28, 29].

3 Membership Inference Detector in Generative Models

In order to prevent the membership inference attack, we propose a membership inference detector in generative models. This detector acts as a security layer between the MLaaS (Machine Learning as a Service) provider and its customers. Before offering synthetic samples generated by the generator to customers, they are passed through the detector. If a synthetic sample is identified as a potential threat to the privacy of the target model, it is discarded. Since our detection model serves as a protection layer to ensure the privacy of the target model, we assume that it has access to the training data of the target model as a trusted agent.

In this section, we present our membership inference detector for generative models. We present the notations used throughout the paper in Table 1. Our approach assumes that the detector, denoted as $Dtc1$, has access to the discriminator of the target model (Dis) and a dataset $X = \{D_{train}, D_{test}\}$, which consists of both training data used for training the target model and data that does not belong to the training set. We distinguish between two settings. In the first setting, called Detector with Access to Discriminator Output ($DADO$), the detector has access to the discriminator output. In the second setting, called Detector with Access to Generator Output ($DAGO$), the detector has access to generator output. Overall, in the former, the detector recognizes the adversary sample based on the discriminator output probability. In the latter, the detector recognizes adversary samples based on the synthetic samples themselves.

3.1 Detector with Access to Discriminator Output (DADO)

As shown in Figure 2, Dis receives samples from X as input and produces an output probability vector labeled "IN" for training data and "OUT" for non-training data. The detection model is built based on this collected dataset.

We formulate our detector as a binary classifier that aims to determine whether a sample x has been used to train the target model or not. Formally, we define $Dtc1(x, M(\theta))$, where M represents the GAN model and θ denotes the parameters of the victim model. If the detector infers that the input data is part of the training dataset, its output is 1; otherwise, it is 0.

We classify objects, described with vectors x in $D = \{D_{train}, D_{test}\}$, among 2 classes $H := \{1, 0\}$

Table 1. Notations used in the paper

Notations	Description
Dtc1	Detector in DADO
BIC	Detector in DAGO
M	Target model
x	Data sample
Dis	Discriminator of target model
θ	Parameters of target model
D_{train}	Training dataset
D_{test}	Test dataset

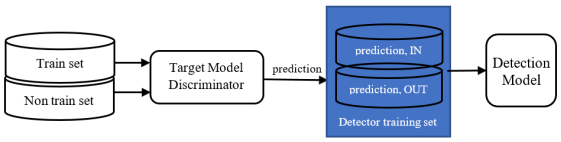

Figure 2. The structure of the proposed detector in Supervised DADO

where:

$$H(Dis(x)) = 1 \; if \; x \in D_{train}, H(Dis(x)) = 0 \; if \; x \notin D_{train} \quad (2)$$

Our goal is to build a detector that $Dtc1 : D \to H$, which minimizes the probability of mistakes:

$$Dtc1(x, M(\theta)) \neq H \quad (3)$$

Given a new input $x \in D$, it checks k nearest points x'' in the dataset D and predicts a majority vote among them. We define data samples as follows:

$$x', x'', x \in D = \{D_{train}, D_{test}\}$$
$$S_x \subseteq D, |S_x| = k, \forall x'' \in S_x, \forall x' \in D \backslash S_x$$
$$Edis(Dis(x), Dis(x')) \geq \max(Edis(Dis(x), Dis(x''))) \quad (4)$$

So, the optimal detector is formulated as follows:

$$Dtc1(x, M(\theta)) = 1 \, [mode \, H(Dis(x''))] \quad (5)$$

Where $Edis$ means Euclidean Distance Function and mode() means selecting the label of the highest occurrence.

3.1.1 Knowledge of the Detector

We assume that the detector aims to detect if a generated record was included in the training set of a generative model or not. The MIA can be categorized into the white-box or black-box settings. Generally, the white-box MIA obtains higher attack accuracy than the black-box since the adversary has more information [13]. Our detector is designed to prevent the white-box attacks that are harder to prevent compared to black-box ones. In GAN models, the discriminator $Dis(x, G(z))$ that outputs $Dis(x) \in [0, 1]$,

represents the probability that x was taken from the training set or the generator $G(z)$. The key idea behind $Dtc1$ is that the discriminator model often has different predictions on real samples and fake samples. We distinguish the settings based on the accessibility of the discriminator's output components. We consider two settings: labeled and unlabeled vector probably.

We assume that the detector has access to the trained target model, which includes the generator G_{target}, and a discriminator D_{target}. It also has a dataset X that contains both types of samples, i.e., those that have been used to train the target model, named X_{train} and those that are not used to do so, named X_{test}. We also consider two different settings. In the first settings, we assume that $Dtc1$ knows the label of X_{train} and X_{test}, and in the second settings, we assume that it does not.

3.1.2 Supervised Detector

In the labeled setting, the detector utilizes a dataset $X = \{X_{train}, X_{test}\}$, where X_{train} is used for training and X_{test} is used for testing. The goal is to train a detector model, denoted as $Dtc1$, by leveraging the differences in discriminator predictions between the training and testing sets.

To achieve this, we feed the dataset X as input to the discriminator. The discriminator then produces probability vectors denoted as $P = \{Dis_{target}(x_1), \ldots, Dis_{target}(x_n)\}$, where each element represents the probability that a data sample belongs to the target model. Next, we create a modified version of P, denoted as P', which includes the discriminator predictions for samples in X_{train} labeled as 1 and samples in X_{test} labeled as 0. In other words, $P' = \{(Dis_{target}(X_{train}), 1), \ldots, (Dis_{target}(X_{test}), 0)\}$.

The key insight behind this approach is that if a data sample is part of the training dataset of a GAN model, the discriminator will predict it to be GAN training data with a high probability. Consequently, these predicted probabilities will be very close to 1. To train our detector model, we employ the k-Nearest Neighbor (KNN) algorithm using P' as our training dataset. KNN is a simple yet effective algorithm that can achieve high accuracy in various supervised prediction problems. By utilizing this approach, we can leverage the differences in discriminator predictions between training and testing datasets to train an accurate detector model using KNN.

3.1.3 Unsupervised Detector

In the unlabeled setting, the detector aims to identify data samples in the dataset $X = \{X_{train}, X_{test}\}$ that were potentially used to train the target model. To achieve this, the detector employs a clustering approach. $Dtc1$ operates by utilizing the discriminator D_{target}, which takes the dataset X as input and produces probability vectors as output. These probability vectors represent the predictions of D_{target} for each data sample in X, denoted as $P = \{D_{target}(x_1), \ldots, D_{target}(x_n)\}$. To cluster these probability vectors, we apply the standard k-Means algorithm and the Gaussian Mixture Model (GMM) as clustering methods. These algorithms group similar probability vectors into two clusters. The resulting clusters are then used to train $Dtc1$. Once $Dtc1$ is trained using the clustered probability vectors, it can be used to infer whether a given data sample belongs to the training set of the target GAN model or not. By feeding a new data sample into $Dtc1$, it predicts whether that sample was likely used during training or not based on its learned knowledge from the clustered probability vectors.

3.2 Detector with Access to Generative Output (DAGO)

In this part, we explain DAGO, which has access to generator output and recognizes adversary samples based on the synthetic sample. DAGO receives the generator's output and identifies if the synthetic sample belongs to the training dataset or not. We employ a Convolutional Neural Network (CNN) to construct a Binary Image Classifier (BIC) model, which can distinguish between training and non-training data. CNNs, a category of neural networks, have demonstrated exceptional efficacy in the domains of image recognition, processing, and classification. CNN has Convolutional and Pooling Layers. Convolutional layers apply convolution operations to the input images, allowing the network to detect local patterns such as edges, textures, and more complex features. Pooling layers reduce the spatial dimensions of the feature maps.

The detector model is designed as a binary classifier that differentiates between two classes: "in" and "out". The model aims to classify each input sample into one of two categories: "in" (part of the training data) or "out" (not part of the training data). We use a dataset $X = \{X_{train}, X_{test}\}$, where X_{train} is used for training and X_{test} is used for testing. Samples in X_{train} labeled as "in" and samples in X_{test} labeled as "out". In other words, $P' = \{(X_{train}, \text{“in”}), \ldots, (X_{test}, \text{“out”})\}$. The Convolutional Neural Network (CNN) is trained using the dataset P' and learns to distinguish between the "in" samples from X_{train} and the "out" samples from X_{test}.

CNN is trained with three convolution and max

pooling layers, in addition to a fully connected layer of size 64, utilizing the Rectified Linear Unit (ReLU) activation function. We employ the sigmoid function as the activation function in the final layer to classify an object with a probabilistic value, which results in either 0 or 1 for binary classification. Adam optimizer is used for the network's optimization, with cross-entropy applied as loss function and learning rate is set to 0.001. The training process is conducted over a range of epochs, varying from 50 to 500.

When the amount of training data is limited, we can use the method transfer learning [30]. Transfer learning is a powerful technique in deep learning, where knowledge gained from training on one dataset (typically a large and general dataset) is transferred to a new, smaller dataset. This approach is beneficial when the target dataset is limited in size, as it allows leveraging pre-trained models to achieve better performance. Transfer learning can significantly improve model accuracy, especially when working with small datasets. Since the model has already learned valuable features, the training process on the new dataset is much faster.

4 Evaluation

This section is dedicated to the empirical evaluation of the proposed membership inference detector. The detector is designed as a binary classifier, with the primary objective of discerning whether a generated sample originates from the training dataset.

4.1 Experimentation Setup

We perform our experiments on three popular image datasets, including Labeled Faces in the Wild (LFW)[31], CIFAR-10 [32], and CelebA [33]. LFW contains 13,233 images of faces, CIFAR-10 includes 60,000 32x32 color images, and CelebA contains 20k face images. Also We choice UCI Adult [34] as a non-image dataset that includes 48,842 records with 14 attributes.

Handling class imbalance is crucial when the number of training and non-training samples differ significantly. Some techniques like SMOTE (Synthetic Minority Over-sampling Technique)[35] are useful. SMOTE generates synthetic samples for the minority class by interpolating between existing minority samples. It creates new samples that are combinations of the features of neighboring samples, thus helping to balance the dataset without duplicating data.

The performance of our detector is assessed on three GAN architectures: Deep Convolutional GAN (DCGAN) [36], a hybrid model combining DCGAN with a Variational Autoencoder (VAE-DCGAN) [7], and BigGANs [37]. DCGAN integrates the GAN framework with convolutional neural networks (CNNs), employing convolutional and convolutional-transpose layers in the generator and discriminator, respectively [36]. The VAE-DCGAN model, on the other hand, enhances the realism of generated samples by utilizing the learned feature representations in the GAN discriminator as the reconstruction error in the VAE [7]. BigGAN is a type of Generative Adversarial Network designed to generate high-quality, high-resolution images.

The efficacy of our model is quantified using standard performance metrics: accuracy, precision, recall, and F1. These metrics are defined in Equations (6), (7), (8), and (9) where True Positives (TPs), True Negatives (TNs), False Positives (FPs), and False Negatives (FNs) are computed as follows:

$$Accuracy = (TP + TN)/(TP + FN + FP + TN) \quad (6)$$

$$Precision = TP/(TP + FP) \quad (7)$$

$$Recall = TP/(TP + FN) \quad (8)$$

$$F1 = 2 * (Precision * Recall)/(Precision + Recall) \quad (9)$$

4.2 Experimentation Results

In this section, we present the results of our experiments, and the comparison with two well-known related works. We first present the experimentation results of the DADO model in both supervised and non-supervised setups. The the DAGO is evaluated. We strive to achieve a baseline accuracy of 0.5, equivalent to a random guess, by ensuring an equal number of member and non-member samples.

4.3 DADO

To provide a comprehensive evaluation of the DADO model ($Dtc1$), we subjected our detector model to rigorous testing on various target GAN models, encompassing both supervised and unsupervised settings. These tests were conducted on three distinct datasets: CIFAR10, LFW, and CelebA.

4.3.1 Supervised DADO

We present the evaluation outcomes of the Supervised detector delineated in Section 3.1.2, focusing on the CIFAR10 and LFW datasets. Figure 3(a) illustrates the performance of the inference detector on the target model, which has been trained on the CIFAR10 dataset over varying epochs. The results indicate that for DCGAN, BigGAN, and VAE-DCGAN target models trained for 100 epochs, the detector discerns training set membership with accuracies of 0.793, 0.731, and 0.684, respectively. For models trained up to 500

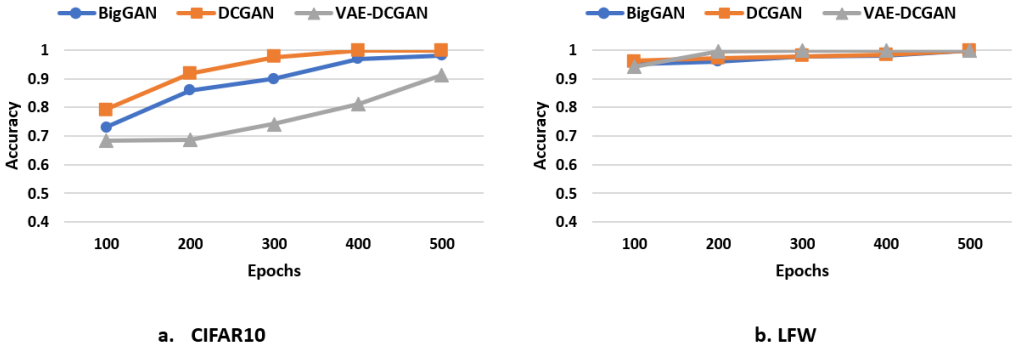

Figure 3. Accuracy of DADO Supervised detector for different datasets

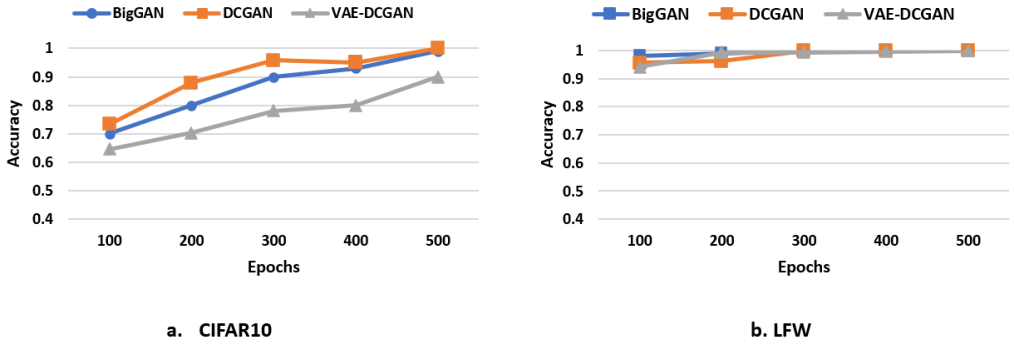

Figure 4. Accuracy of DADO k-Means detector for different datasets

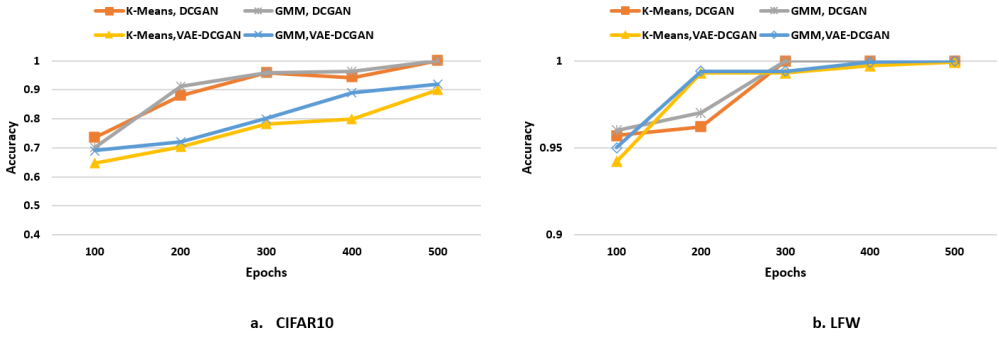

Figure 5. Accuracy of DADO Unsupervised detector for different datasets

epochs, these accuracies rise to 1, 0.982, and 0.912, respectively. This trend of increasing accuracy with the number of training epochs underscores the heightened vulnerability of both GANs trained on CIFAR10 to membership inference attacks.

Figure 3(b) depicts the performance of the inference detector on the target model trained on the LFW dataset. For DCGAN target models trained across a range of 100 to 500 epochs, the detector discerns the training set membership with an accuracy that varies from 0.962 to 1. For BigGAN target models, the achieved accuracy ranges from 0.95 to 1.For VAE-DCGAN target models, the achieved accuracy ranges from 0.943 to 1.

In conclusion, we ascertain that the detector can infer the training set with perfect accuracy when DCGAN, BigGAN, and VAE-DCGAN serve as the target models. Furthermore, it is observed that the performance of the detector is enhanced with an increase in the number of training epochs.

4.3.2 Unsupervised DADO

In this section, we report the evaluation results of the unsupervised setting of the detector described in Section 3.1.3 on CIFAR10 and LFW.

Figure 4(a) shows the results of the inference detector on the target model trained on the CIFAR10 dataset in unsupervised settings by the k-Means algorithm. For DCGAN, BigGAN, and VAE-DCGAN target models trained for 100 epochs, the detector

infers the training set membership with 0.734, 0.71, and 0.646 accuracy, and for models trained up to 500 epochs with 1, 0.99, and 0.9 accuracy, respectively. Like supervised settings, when the number of epochs required to train the target model increases, the detector's accuracy improves.

Figure 4(b) plots the results of the inference detector on the target model trained on the LFW dataset. For DCGAN as target models trained from 100 to 500 epochs, the detector infers the training set membership with the accuracy of 0.97 to 1. For BigGAN as target models trained from 100 to 500 epochs, the detector infers the training set membership with the accuracy of 0.957 to 1. For VAE- DCGAN target models, the accuracy reaches 0.942 to 0.999.

Figure 5 plots the results of the inference detector in unsupervised settings and compare Gaussian Mixture Model (GMM) and k-Means. The results indicate that clustering using two different methods yields similar outcomes.

4.3.3 Summary

In summary, in supervised settings, the detector $Dtc1$ demonstrated a commendable mean accuracy of 0.917. In contrast, in unsupervised settings, it achieved a slightly lower, yet still substantial, mean accuracy of 0.910. Remarkably, even on the VAE-DCGAN model, which posed the most challenging scenario for $Dtc1$, our model still managed to secure a mean accuracy of 0.766, significantly surpassing the baseline.

It is noteworthy that our detector exhibited superior performance on the LFW dataset. Moreover, it demonstrated exceptional efficacy on the DCGAN model, achieving perfect accuracy, precision, and recall when trained on the LFW dataset using DAGAN. However, when deployed on the VAE-DCGAN model, our detector's efficiency experienced a marginal decrease by 0.003%.

In Table 2, we listed the best performance results of our models against the selected related works. As illustrated in the table, with DCGAN serving as the victim model, the detector successfully inferred the training set with an impressive accuracy of 100% on the LFW and 0.99 on the CIFAR datasets. However, on the CelebA dataset, characterized by higher complexity data, our detector's accuracy declined to 0.8 when DCGAN was employed as the victim model. This can be attributed to the reduced likelihood of our model memorizing such complex data. Similarly, when VAE-DCGAN was utilized as the victim model, a decrease in accuracy was observed compared to previous results. The accuracy was recorded to be 0.99 on LFW and 0.84 on CIFAR datasets, but it fell to 0.79 on CelebA. This suggests that the degree of overfitting is diminished in the VAE-DCGAN model.

We calculate the F1 score, the harmonic means of precision and recall, for different thresholds. The threshold that maximizes the F1 score is the detector's output threshold, which varies between 0.43 and 0.57 in different scenario.

4.4 DAGO

In this section, we present the evaluation outcomes of the image detector, as detailed in Section 3.2, on the CIFAR10 and LFW datasets. Figure 6 provides a graphical representation of the precision and recall metrics of the inference detector over a range of epochs. The results indicate that the efficiency of DAGO at 50 epochs is less than 0.95. However, as the number of training epochs escalates, precision and recall of the detector progressively approach 1.

In conclusion, our observations substantiate that using the generator's output to recognize adversary samples is an effective strategy for their identification. Furthermore, we discern that an increase in the number of training epochs significantly enhances the performance of the detector. Thus, the amalgamation of leveraging the generator's output and extending the training duration emerges as a successful strategy for optimizing the detector's performance.

4.5 Comparison to Attacks

We conduct a comparative analysis of our detector with two prominent membership inference attacks: logan [10] and Gan-leaks [20]. As explicated in Section 2, logan encompasses attack models against GANs wherein the discriminator is accessible. Concurrently, Gan-leaks proposes various attack models contingent on the quantum of information the attacker receives from the victim model. Given that the white-box settings exhibit the highest efficiency, we compare these settings.

The detector model is evaluated on both DCGANs and VAE-DCGAN. Figure 7 and Figure 8 juxtapose the performance of DADO with logan, predicated on different datasets and GANs, across varying epochs. It is discernible that Supervised settings consistently outperform logan on both datasets and across different GANs, thereby facilitating superior detection of privacy breaches. Moreover, Unsupervised settings predominantly perform better than logan and, in certain instances, exhibit comparable performance. Figure 9 compare the performance of the DADO Supervised and DAGO(BIC) with logan. As depicted in the figure, our detectors' accuracy is high in fewer epochs, while at epoch 500, they are almost identi-

Table 2. DADO Supervised performance for different models and datasets

Data	LFW			CIFAR			CelebA		
	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec
DCGAN	1	1	1	0.996	0.994	0.999	0.801	0.797	0.802
VAE-DCGAN	0.997	0.996	0.998	0.84	0.812	0.884	0.791	0.782	0.799
Acc: Accuracy, Pre: Precision, Rec: Recall									

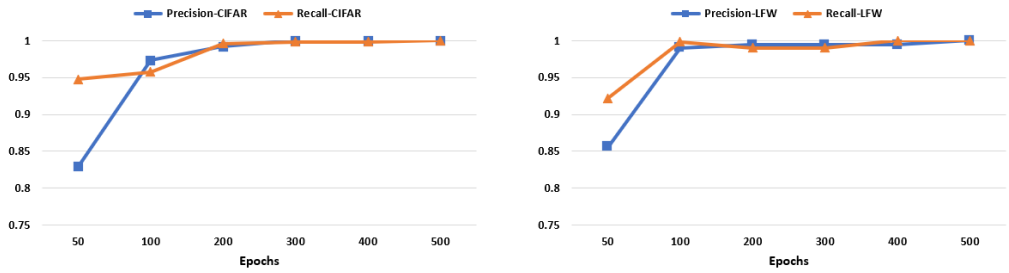

Figure 6. Performance of DAGO for different datasets

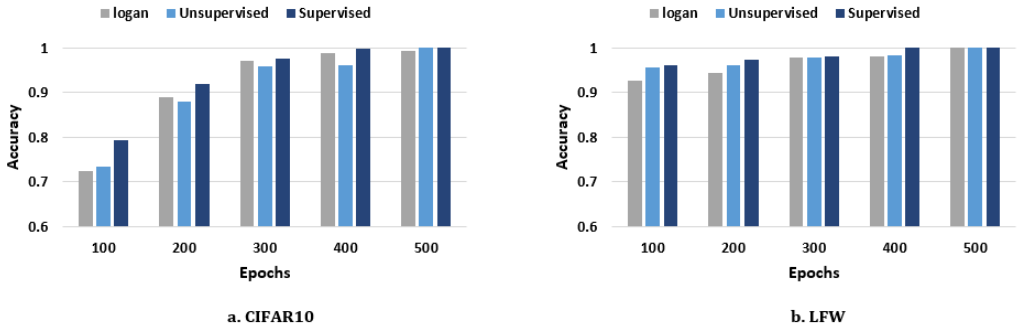

Figure 7. Comparison of different models on DCGAN

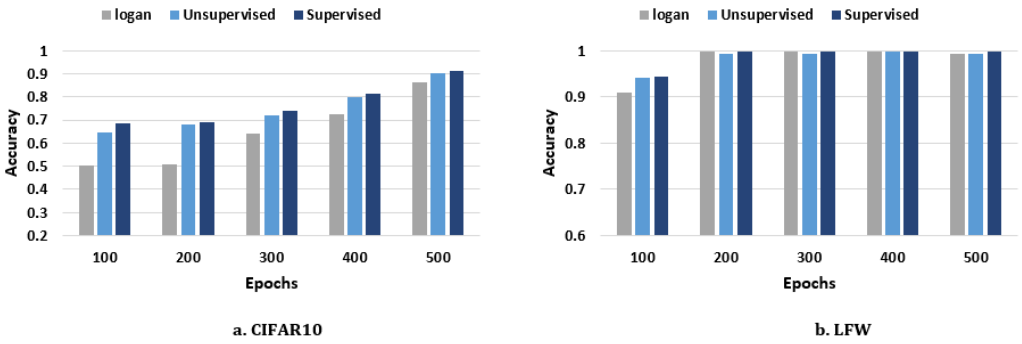

Figure 8. Comparison of different models on VAE-DCGAN

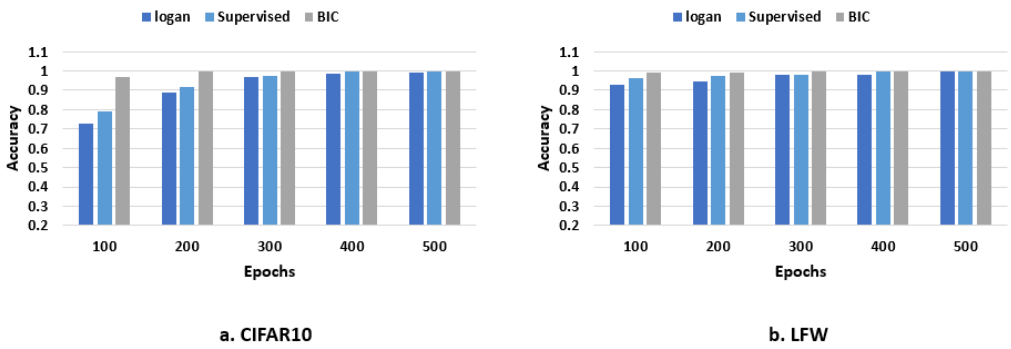

Figure 9. Comparison of different models with logan

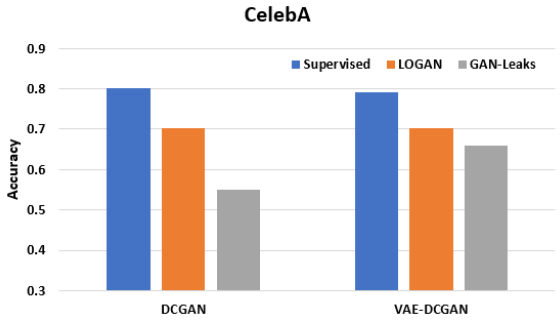

Figure 10. Comparison of different models on CelebA

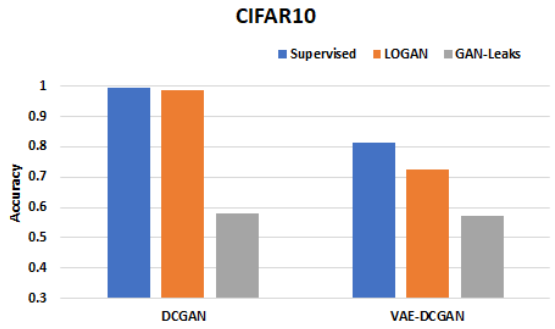

Figure 11. Comparison of different models on CIFAR10

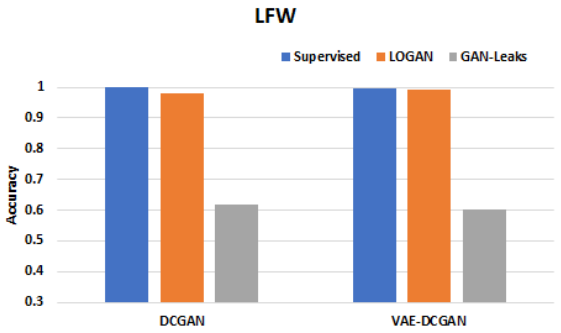

Figure 12. Comparison of different models on LFW

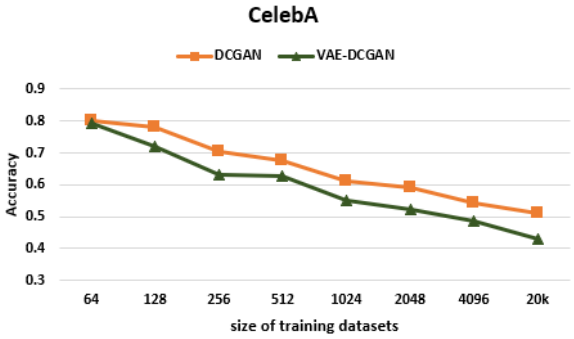

Figure 13. Supervised detector performance on different training set size

cal. In all instances, the binary image detector consistently manifests higher accuracy compared to the others.

Figure 10, Figure 11, and Figure 12 present a comparative analysis of the supervised setting with Logan and Gan-leaks across diverse datasets. Both GAN models have been trained with the minimum data size throughout 500 epochs. In both GAN models, the detector outperforms the attack models. This advantage bolsters the capability to identify privacy leaks in GAN models.

4.6 Different Training Set Size

We executed a series of experiments employing training datasets of varying sizes to train both DCGAN and VAE-DCGAN models. Subsequently, we assessed the performance of our supervised detector on the CelebA dataset. Figure 13 provides a graphical representation of the detector's performance, delineating its efficacy across an array of GAN models on the CelebA dataset.

Our observations indicate that the detector exhibits superior performance when the training dataset' size is relatively small. For instance, when the victim models are trained with a limited set of only 64 images, the accuracy of the detector escalates significantly to 0.8. Conversely, an increase in the size of the victim training set corresponds to a decline in the performance of the detector. These observations underscore a distinct correlation between the size of the victim training set and their susceptibility to mem-

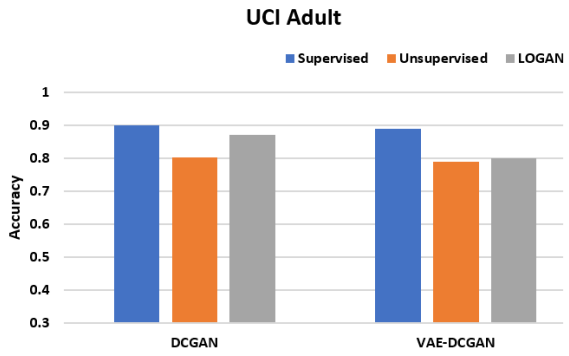

Figure 14. Different models on UCI Adult dataset

bership inference attacks. Specifically, victim models trained with smaller datasets tend to memorize data from their training set more readily, thereby rendering them more vulnerable to inference attacks [20]. Our experimental results substantiate that smaller training datasets can facilitate enhanced performance of our supervised detector in detecting membership inference attacks on GAN models.

4.7 Non Image Dataset

In addition to the image-based evaluation, we extend our analysis to include non-image data. This allows us to assess our models across different data modalities. We utilize the UCI Adult dataset to test our models. This dataset includes 14 attributes, such as age, sex, work class, education, marital status, race, and native country.

Figure 14 plot the results of the inference DADO models and LOGAN, in different target models. The evaluation results indicate that the models perform well on non-image data. In the supervised setting, our detector outperforms LOGAN; however, in the unsupervised setting, it performs weaker than LOGAN.

5 Conclusion

In this manuscript, we have unveiled a pioneering membership inference detector for generative models, which exhibits versatility in its applicability to many victim models. Our primary objective was to mitigate privacy concerns inherent in GANs by proposing a dual approach encompassing both supervised and unsupervised techniques, predicated on the output of the discriminator. These techniques empower us to pinpoint potential privacy leaks and assess the performance of our approach across various scenarios.

Through a series of rigorous experiments, we have substantiated the efficacy of our model in detecting membership across a diverse array of generative models. Additionally, we have highlighted its capability to identify synthetic samples that could potentially jeopardize the privacy of the underlying dataset. This underscores the practical utility of our model in alleviating privacy threats associated with generative models.

Looking ahead, our future research will focus on retraining the target model with adversarial samples identified by the detection model, along with a controlled amount of noise. We anticipate that this retraining will curtail the privacy leakage. Furthermore, we intend to broaden our investigations by deploying our detector on other GAN models and datasets. This will facilitate the validation of its effectiveness across disparate domains and further consolidate its robustness. In essence, our work signiïcantly contributes to addressing privacy concerns in generative models and paves the way for future advancements in this domain.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. volume 27. Curran Associates, Inc., 2014.

[2] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[3] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. volume 19, page 121, 2014.

[7] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. pages 1558–1566, 2016.

[8] Nils Homer, Szabolcs Szelinger, Margot Redman,

David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4:e1000167, 2008.

[9] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. pages 3–18, 2017.

[10] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. pages 133–152, 2019.

[11] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

[12] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. pages 880–895, 2021.

[13] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. pages 739–753, 2019.

[14] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated {White-Box} membership inference. pages 1605–1622, 2020.

[15] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. pages 1273–1282, 2017.

[16] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 2019.

[17] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proc. Priv. Enhancing Technol.*, 2019:232–249, 2019.

[18] Kin Sum Liu, Bo Li, and Jie Gao. Generative model: Membership attack, generalization and diversity. *CoRR, abs/1805.09898*, 2018.

[19] Kin Sum Liu, Chaowei Xiao, Bo Li, and Jie Gao. Performing co-membership attacks against deep generative models. pages 459–467, 2019.

[20] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. pages 343–362, 2020.

[21] Hailong Hu and Jun Pang. Membership inference attacks against gans by leveraging over-representation regions. pages 2387–2389, 2021.

[22] Maryam Azadmanesh, Behrouz Shahgholi Ghahfarokhi, and Maede Ashouri Talouki. An auto-encoder based membership inference attack against generative adversarial network. *ISeCure, 15.2*, pages 240–253, 2023.

[23] Saverio Cavasin, Mari Daniele, Simone Milani, and Mauro Conti. Fingerprint membership and identity inference against generative adversarial networks. *Pattern Recognition Letters*, pages 231–241, 2024.

[24] Abdallah Alshantti, Adil Rasheed, and Frank Westad. Privacy re-identification attacks on tabular gans. 2024.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15:1929–1958, 2014.

[26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. pages 2818–2826, 2016.

[27] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. pages 634–646, 2018.

[28] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. pages 1964–1974, 2021.

[29] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. Differentially private data generative models. *arXiv preprint arXiv:1812.02274*, 2018.

[30] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, pages 1–40, 2016.

[31] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008.

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. pages 3730–3738, 2015.

[34] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[35] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, , and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling tech-

nique. *Journal of artificial intelligence research*, pages 321–357, 2002.

[36] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[37] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. 2019. github.com/huggingface/pytorch-pretrained-BigGAN.

Ala Ekramifard received the B.Sc. and M.Sc. degrees in computer engineering from the Ferdowsi University of Mashhad, Mashhad, Iran. She is currently Ph.D. candidate in Computer Engineering at Ferdowsi University of Mashhad. Her research interests include Data Privacy, Privacy-Preserving Machine Learning, and Data Security.

Haleh Amintoosi is an associate professor at the Ferdowsi University of Mashhad, Mashhad, Iran. She is also a visiting senior lecturer at the School of Computer Science and Engineering, University of New South Wales (UNSW), Sydney, NSW, Australia. Her main research interests include Trust and Privacy in Crowdsourcing and Crowdsensing Systems, Authentication Protocols, and Blockchain. Haleh received her Ph.D. degree in computer science from UNSW.

Seyed Amin Hosseini Seno is professor at the Ferdowsi University of Mashhad, Mashhad, Iran. His research interests include Computer Networks, QoS, the IoT, and Network Security. He received the Ph.D. degree from Universiti Sains Malaysia, Malaysia.