Generative Adversarial Networks: A Survey on Attack and Defense Perspective

CHENHAN ZHANG, University of Technology Sydney, Australia
SHUI YU, University of Technology Sydney, Australia
ZHIYI TIAN, University of Technology Sydney, Australia
JAMES J.Q. YU, University of York, United Kingdom

Generative Adversarial Networks (GANs) are a remarkable creation with regard to deep generative models. Thanks to their ability to learn from complex data distributions, GANs have been credited with the capacity to generate plausible data examples, which have been widely applied to various data generation tasks over image, text, and audio. However, as with any powerful technology, GANs have a flip side: their capability to generate realistic data can be exploited for malicious purposes. Many recent studies have demonstrated the security and privacy (S&P) threats brought by GANs, especially the attacks on machine learning (ML) systems. Nevertheless, so far as we know, there is no existing survey that has systematically categorized and discussed the threats and strategies of these GAN-based attack methods. In this paper, we provide a comprehensive survey of GAN-based attacks and countermeasures. We summarize and articulate: (1) what S&P threats of GANs expose to ML systems; (2) why GANs are useful for certain attacks; (3) what strategies can be used for GAN-based attacks; (4) what countermeasures can be effective to GAN-based attacks. Finally, we provide several promising research directions combining the existing limitations of GAN-based studies and the prevailing trend in the associated research fields.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Security and privacy**; • **Computing methodologies** → *Machine learning*; *Distributed computing methodologies*;

Additional Key Words and Phrases: Generative adversarial networks, GANs survey, deep learning, security and privacy, attack and defense

1 INTRODUCTION

1.1 Background

Generative modeling using neural networks has received wide attention in machine learning (ML) for its capacity to learn data in an unsupervised manner. Among these techniques, **Generative Adversarial Networks (GANs)** have emerged as the most prominent and widely utilized approach in recent years due to their implicit learning capability of complex and high-dimensional data distributions. GANs have demonstrated successful applications in the realm of computer vision, including tasks such as image generation, image resolution enhancement, and image texture synthesis [18]. GANs are capable of generating fake portraits that can be remarkably lifelike,

This research was supported by the Australian Research Council (ARC) DP200101374. The corresponding author of this paper is Shui Yu.

Authors' addresses: Chenhan Zhang, chenhan.zhang@student.uts.edu.au, University of Technology Sydney, Sydney, Australia; Shui Yu, shui.yu@uts.edu.au, University of Technology Sydney, Sydney, Australia; Zhiyi Tian, zhiyi.tian@student.uts.edu.au, University of Technology Sydney, Sydney, Australia; James J.Q. Yu, yujq3@sustech.edu.cn, University of York, Heslington, York YO10 5DD, York, United Kingdom.

0360-0300/2023/4-ART111 $15.00
https://doi.org/10.1145/3615336

making it difficult to distinguish them from real ones[1]. In addition to computer vision, many successes of GANs have also been witnessed in natural language processing [69, 71, 138], speech recognition [29, 110], and so on.

Birth of GANs. In the community of unsupervised learning, the generative model is one of the most promising techniques. Conventional generative models typically rely on approximate inference methods such as maximum likelihood and Markov chains [53, 99]. For example, restricted Boltzmann machines-based models (e.g., deep belief networks) are based on maximum likelihood estimation, and these models can represent latent distributions whose parameters closely match the empirical distribution of the training data [91]. However, there have been investigations conducted to explore the limitations of deep generative models, especially the challenges in maximum likelihood estimation. Thus, Goodfellow *et al.* identified the research gap and proposed a novel generative model — Generative Adversarial Networks (GANs) [45] — which attempts to address the limitation that exists in the previous generative models.

Table 1. Summary of related surveys on Generative Adversarial Networks (only papers are peer-reviewed and published are selected here).

Literature	**Year**	**Application-Specific**	**Coverage**						**Remark**
			Variants	**Metrics**	**Benchmark**	**Datasets**	**S&P**	**DS**	
Wang *et al.* [121]	2017	–	✓						GANs in parallel systems.
Hong *et al.* [57]	2019	–	✓	✓					Theoretical analysis on GANs' working principle.
Pan *et al.* [91]	2019	–	✓	✓					Latest advances of GANs.
Cao *et al.* [18]	2019	CV	✓	✓	✓	✓			Performance of GANs' variants on computer vision.
Yi *et al.* [136]	2019	medical imaging (CV)	✓	✓		✓			GANs' design principle for medical imaging.
Dutta *et al.* [30]	2020	cybersecurity					✓		Overview of GANs on cybersecurity.
Hajarolasvadi *et al.* [49]	2020	human emotion synthesis (CV)	✓	✓	✓	✓			GANs' design principle for human emotion synthesis.
Tschuchnig *et al.* [115]	2020	digital pathology (CV)	✓	✓					Domain transfer by GANs in digital pathology.
Gao *et al.* [42]	2020	spatial-temporal tasks	✓	✓		✓			GANs' usage for modeling spatial-temporal tasks.
Yinka-Banjo *et al.* [137]	2020	cybersecurity	✓				✓		Overview of GANs on cybersecurity.
Saxena *et al.* [103]	2021	–	✓	✓					General overview of GANs' design and optimization solutions.
Gui *et al.* [47]	2021	–	✓	✓		✓			Theoretical analysis on mode collapse.
Bond-Taylor *et al.* [13]	2021	–	✓	✓	✓				Comparison among GANs and other deep generative models.
De Rosa *et al.* [27]	2021	text generation (NLP)		✓	✓	✓			Benchmarks of GANs on text generation.
Navidan *et al.* [85]	2021	networking	✓	✓	✓	✓			Benchmarks of GANs on networking .
Zhou *et al.* [156]	2021	text-to-image synthesis	✓	✓	✓	✓			Benchmarks of GANs on text-to-image synthesis.
Toshpulatov *et al.* [114]	2021	3D face generation (CV)	✓	✓	✓	✓			Benchmarks of GANs on 3D face generation.
Li *et al.* [70]	2021	–	✓	✓					Theoretical analysis on GANs' design principle.
Wali *et al.* [119]	2022	speech processing	✓	✓		✓			Comprehensive study of GANs on speech processing.
Cai *et al.* [17]	2022	–					✓	✓	Overview of S&P applications with GANs.

Abbreviation: S&P–Security and Privacy, DS–Decentralized System

1.2 Motivation

GAN-based applications are expanding rapidly across various domains, including finance, healthcare, and infrastructure, to name a few. Unfortunately, not all of these applications are used for positive purposes. Some malicious actors attempt to acquire private data without authorization or corrupt the learning or inference process of models, which compromises the **security and privacy (S&P)** of the data or models' stakeholders [17]. Some are illegitimate, such as recovering private medical information without permission to access patients' raw data using GAN to learn the real image's distribution [19]. Furthermore, these malicious activities can pose severe threats to the life safety of individuals. For example, suppose an automatic pilot system's computer vision model is corrupted with poisoning examples generated by GAN. Consequently, the system may misjudge a traffic sign resulting in the car accelerating and crashing [34, 51]. Accordingly, there has been increasing attention on studying GANs in the context of S&P [17]. The S&P of any system can be investigated as an attack/defense problem [25]. This problem is measured concerning the attacker's goal and capabilities designed to achieve the goal as possible and the victims' capabilities designed to defend against the attack. GAN, as a generative model,

[1] Examples of plausible portraits generated by GANs can be found at the following website: https://thispersondoesnotexist.com/.

cannot perform the attacks solely without any strategic instructions. An insightful way to study it in the S&P context could be understanding how GANs are "weaponized" in an attack process along with the threats it brings.

Although there has been a number of surveys on GANs, most of them investigated the existing GAN-based efforts from a certain range of perspectives except for S&P. [57, 91, 103] presented a brief overview of the GANs taxonomy and incorporated some of the architectural variants and loss-enhanced variants of GANs. [27, 49, 114, 115, 119, 136, 156] provided comprehensive of GANs but in particular applications (e.g., text generation). [13, 90] emphasized comparing GANs and parallel techniques. While [85] involved security in the survey, it only paid close attention to related issues in the networking field. [30] and [137] identified "security" as "cybersecurity" and surveyed GAN-based applications in this domain. Recently, [17] summarized the GANs studies with regards to S&P, however, this survey is performed at an application level where the taxonomy is based on different application domains. Compared to [17], we put our survey in a unique scope — **investigate existing GAN-based attacks and corresponding countermeasures from the angle of threat and strategy**.

Furthermore, we survey existing GAN-based attack methods against decentralized/distributed ML systems[2] and further discuss the attack and defense strategies behind these methods. Decentralized ML systems are gaining prevalence due to further commercialization, the increasing requirements for collaboration and the rising demand for stringent data privacy criteria. The attacks against decentralized ML systems have garnered significant attention from researchers, especially on the recently popular federated learning systems. However, many attack methods available to centralized systems fail to handle the decentralized ML systems due to cross-silo restrictions. In this context, some unique properties enable GAN to bypass these barriers, making it a sharp weapon to attack FL systems. We believe that the studies on decentralized ML systems can expand readers' horizons and gain them a better understanding of GANs' properties, GAN-based attacks' tactics, and countermeasures.

Additionally, to showcase the distinctions and serve as a gateway for readers interested in exploring other GAN-related studies, we include a summary of GAN-oriented surveys in Table 1.

1.3 Highlights and Contributions

- To the best of our knowledge, this study represents the pioneering effort to provide a comprehensive review of the existing literature on GAN-based attacks and countermeasures. The review investigates various studies, considering the perspectives of threat and strategy, with a specific focus on both centralized and decentralized machine learning systems.
- We provide insights into different GAN-based attack methods. Integrating the views of conventional S&P threats, we provide a novel taxonomy that formalizes different scenarios and further assists researchers in identifying the threats, strategies, and GAN-based techniques employed in these cases.
- There have been various GAN-based attack and defense approaches — it is challenging and time-consuming to understand the broad picture of all works in this field. To offer readers a more accessible means of understanding existing works, we present a unified formulation, definition, and visualization among them in this survey.
- In addition to GAN-based attacks and countermeasures, we expand our discussion to a broader horizon that covers the definition of S&P problems in machine learning and some general attack and defense methods. Thus, this survey serves as a resource for the research community (experts) and brings a clear image to researchers outside this research domain (beginners).
- Lastly, as the GAN-oriented S&P problem is still at its seminal stage and there is plenty of room for either consolidating existing research or exploring new directions, we also shed light on several promising future research directions.

[2] To streamline the presentation, we use the term "decentralized ML systems" to refer to both distributed and decentralized ML systems throughout the rest of the paper.

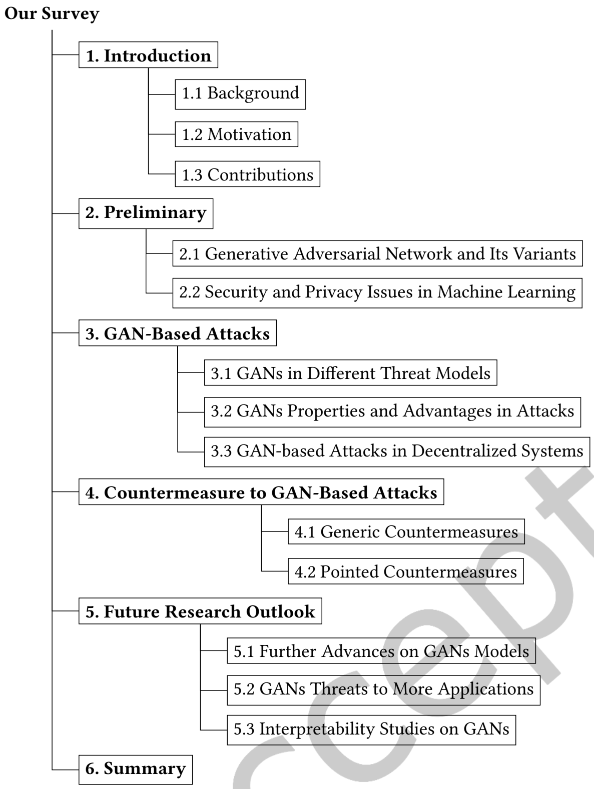

Our Survey
1. Introduction
1.1 Background
1.2 Motivation
1.3 Contributions
2. Preliminary
2.1 Generative Adversarial Network and Its Variants
2.2 Security and Privacy Issues in Machine Learning
3. GAN-Based Attacks
3.1 GANs in Different Threat Models
3.2 GANs Properties and Advantages in Attacks
3.3 GAN-based Attacks in Decentralized Systems
4. Countermeasure to GAN-Based Attacks
4.1 Generic Countermeasures
4.2 Pointed Countermeasures
5. Future Research Outlook
5.1 Further Advances on GANs Models
5.2 GANs Threats to More Applications
5.3 Interpretability Studies on GANs
6. Summary

Fig. 1. The structure of the survey.

1.4 Structure of the Survey

The structure of this survey can be seen in Figure 1. Section 2 introduces primitive GAN and its variants and provides a novel taxonomy of the S&P issues in ML. Based on the proposed taxonomy, we review and discuss GAN-based attacks in Section 3. Section 4 presents countermeasures to GAN-based attacks. We discuss promising future research directions in Section 5, and the paper is concluded in Section 6.

2 PRELIMINARY

2.1 Generative Adversarial Network and Its Variants

Before reviewing and discussing GANs as attack/defense weapons in machine learning tasks, we first introduce the primitive GANs and the variants as preceding knowledge to non-expert readers to make them understand the principle and development progress of GANs.

2.1.1 Primitive GAN. The framework of primitive GANs is illustrated in Figure 2. The generator G generates samples that aim to match the latent distribution $p_{\text{data}}\,(X)$ of the real data samples X. A random noise vector Z, which follows a uniform distribution $p_Z(Z)$, is input to G. G then maps this noise vector to a new space that has the same dimensions as the real data to develop the fake samples $G(Z)$. The discriminator D ingests both samples from both $G(Z)$ and X to distinguish them — it is a binary classifier that evaluates the input samples and outputs a probability value of whether the samples are generated by G (i.e., fake data) or from real data. Both G and D

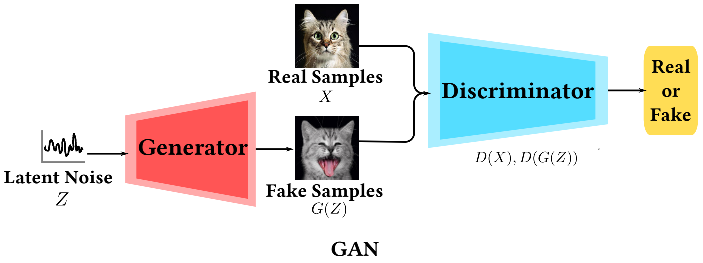


Fig. 2. The architecture of primitive GAN [45]. This architecture is also followed by most of the non-architectural variants of GAN, e.g., WGAN [3] and LSGAN [77].

are differentiable functions represented by neural networks with parameters θ_G and θ_D, respectively. G aims to maximize the probability of D. To achieve their individual goals, the two networks engage in a zero-sum minimax game, where G tries to maximize the probability, but D tries to minimize it. GANs address the drawbacks of traditional generative models, where deep learning training methods, e.g., backpropagation and dropout, are employed to train both networks without maximizing the likelihood that involves Markov chains or approximate inference. The objective functions of GAN are to minimize difference between $p_G(X)$ and $p_{\text{data}}(X)$, which is formulated as

$$\begin{cases} \mathcal{L}_{(D)} = -\frac{1}{2}\underset{X\sim p_{\text{data}}(X)}{\mathbb{E}}[\log D(X)] - \frac{1}{2}\underset{Z\sim p_Z(Z)}{\mathbb{E}}[\log(1 - D(G(Z)))] \\ \mathcal{L}_{(G)} = -\frac{1}{2}\underset{Z\sim p_Z(Z)}{\mathbb{E}}[\log D(G(Z))] \end{cases}, \tag{1}$$

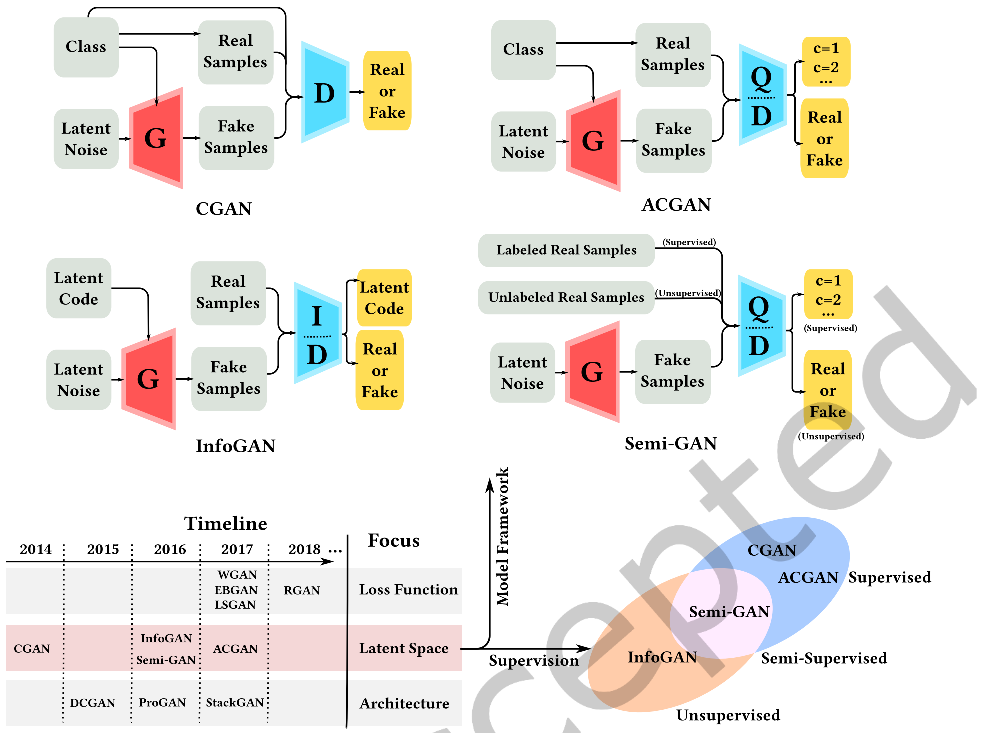
From the aims of D and G, we know that $D(G(Z))$ is expected to be close to 0 by D and 1 by G, respectively. D maximizes the output if an input sample is from real data and minimizes the output if the sample is generated by G — the term $\log(1 - D(G(Z)))$ is designed to this end. As D and G play a zero-sum game that G tries to maximize D's output, the final objective function is written as

$$\min_{\theta_G}\max_{\theta_D} V(D,G) = \underset{X\sim p_{\text{data}}(X)}{\mathbb{E}}[\log D(X)] + \underset{Z\sim p_Z(Z)}{\mathbb{E}}[\log(1 - D(G(Z)))], \tag{2}$$

where $V(\cdot)$ denotes a binary cross-entropy function. Note that if D yields a probability of 0.5, the equilibrium between G and D occurs that D cannot determine if the sample is from real data or generated by G. To update the model parameters, the training of G and D is performed alternatively, where one network is fixed while the other is backpropagated.

2.1.2 Variants of GAN. With the development of GANs, variants have been developed to overcome the deficiencies or extend the functions of the primitive GAN. As shown in Figure 3, these variants can be broadly categorized into three classes based on the aspect they target for improvement: **latent space**, **loss function**, and **architecture**.

Latent Space Enhancement. Mapping latent space to feature space is crucial in GANs. Primitive GANs use random input noise Z, resulting in inferior semantic features due to amorphousness and entanglement. Improved studies address this issue by introducing auxiliary and useful information to the latent space input. Latent space-enhanced GANs are in good graces of attack methods since they can effectively utilize limited knowledge gained by attackers to generate examples. Furthermore, some of them can assist attackers in inferring additional properties of the generated samples (e.g., class). In the sequel, we introduce the four most representative variants regarding the enhancement of latent space (see Figure 3 for their frameworks).



2014	2015	2016	2017	2018 ...	Focus
			WGAN EBGAN LSGAN	RGAN	Loss Function
CGAN		InfoGAN Semi-GAN	ACGAN		Latent Space
	DCGAN	ProGAN	StackGAN		Architecture

Fig. 3. Overview of GANs' evolution and most representative variants of GANs from a latent space perspective. For ACGAN and InfoGAN, Q represents the classifier that shares the model architecture (except the output layer) and parameters with the discriminator.

Conditional GAN (CGAN) [81] is the most widely-adopted latent space-enhanced GAN. Compared to the primitive GAN, the generator and the discriminator of CGAN add additional information C as a condition, which can be any information such as category information or other modal data. For the generator, a priori input noise $p(Z)$ and the conditional information C combine to form the joint hidden layer representation. The objective function of CGAN remains a two-player minimax game but with conditional probabilities:

$$\min_{\theta_G} \max_{\theta_D} V(D,G) = \underset{X\sim p_{\text{data}}(X)}{\mathbb{E}}[\log D(X|C)] + \underset{Z\sim p_Z(Z)}{\mathbb{E}}[\log(1 - D(G(Z|C)))], \tag{3}$$

where the loss function of the discriminator can be formulated as

$$\mathcal{L}_{(D)} = \underset{X\sim p_{\text{data}}(X)}{\mathbb{E}}[\log P(S = \text{real} \mid X)] + \underset{Z\sim p_Z(Z)}{\mathbb{E}}[\log P(S = \text{fake} \mid G(Z|C))], \tag{4}$$

where $S = \{\text{real}, \text{fake}\}$ denotes the source of the input data sample. It can be observed that both the generator and discriminator are expressed as conditional probabilities, specifically computed based on the given variable C. By modifying the conditions fed into the generator network, it is possible to generate examples belonging to different categories or classes.

Auxiliary Classifier GAN (ACGAN) [87] is an extension to CGAN with improved discriminator. Specifically, the discriminator of ACGAN does not confine to distinguishing true from false but can also differentiate between categories by introducing an auxiliary classifier developing the probability over different classes. The loss function of ACGAN is designed as $\mathcal{L}_{(D)} = \mathcal{L}_{(\text{source})} + \mathcal{L}_{(\text{class})}$ where $\mathcal{L}_{(\text{source})}$ is the same as the loss function of CGAN's discriminator (Eq. (4)) while $\mathcal{L}_{(\text{class})}$ denotes the classification loss:

$$\mathcal{L}_{(\text{class})} = \underset{X \sim p_{\text{data}}(X)}{\mathbb{E}} \left[\log P\left(C = c \mid X\right)\right] + \underset{Z \sim p_Z(Z)}{\mathbb{E}} \left[\log P\left(C = c \mid G(Z|C)\right)\right], \tag{5}$$

where c denotes a specific class label. The functional extension makes ACGAN a solution to many class-specific scenarios [118, 145].

InfoGAN [21] works in an unsupervised fashion, different from CGAN and ACGAN. The objective function of InfoGAN can be written as

$$\min_{\theta_G} \max_{\theta_D} V(D, G) = V_{\text{GAN}}(D, G) - \lambda I(C; G(Z, C)). \tag{6}$$

InfoGAN's input incorporates the latent semantic variable C and noise variable Z. Compared to the primitive GAN's objective function, an additional term is introduced, denoted as $\lambda I(C; G(Z, C))$[3], which computes the mutual information between C and the output of the generator. InfoGAN encourages high mutual information between the generated samples and selected latent variables. While InfoGAN also develops the probability complied with conditional information C like CGAN, they differ: CGAN supposes C is known, which contains specific information (e.g., class labels); however, InfoGAN assumes C is unknown, sampled from a prior $p(C)$. This distinction highlights the supervision difference between CGAN and InfoGAN.

Semi-supervised GAN (Semi-GAN) [86] exploits both supervised and unsupervised learning by training the discriminator in both two manners. In an unsupervised manner, the discriminator is trained similarly to a regular GAN to predict whether the example is from the real source. In a supervised manner, the discriminator is trained to classify the label of real samples. The merit of this design is that unsupervised training enables the model to learn effective feature extraction capabilities from a huge unlabeled dataset. Meanwhile, supervised training allows the model to exploit the learned features and allocate class labels.

Architecture and Loss Enhancements. The modification of architectural designs and loss functions can also contribute to enhancing GANs, solving problems such as training stability and collapse mode. In terms of architecture, *Deep convolutional GAN (DCGAN)* [95] adopts convolutional neural networks (CNNs) for both the discriminator and the generator instead of the multilayer perceptron (MLP) neural networks, achieving more stable training and more high-quality image generation. Such architectural evolution can also be seen in StackGAN [141], BigGAN [14], StyleGAN [63], etc. In terms of the loss function, *Wasserstein GAN (WGAN)* [3] introduces Wasserstein distance in the loss function design to solve the problems above. Such loss function-enhanced GANs include EBGAN [152], LSGAN [77], etc.

In addition, various practical applications have spawned many variants as well. For example, CycleGAN [157] and Pix2Pix [59] for image-to-image translation, 3DFaceGAN [105] and GANFit [43] for 3D face generation, to name a few. From the attack and defense perspective, the proliferation of GAN variants also provides attackers with a wider array of potential tools.

2.2 Security and Privacy in Machine Learning

To gain the audience a better insight into GAN-based attacks/defenses and their scope, we define security and privacy (S&P) in machine learning (ML) here — this also provides a systematical knowledge frame to non-expert readers. Typically, we can progressively explore the S&P issues in machine learning by identifying **CIA model** and **threat model**.

[3] Note that in this survey, we have simplified the expression of this term. For readers interested in more details, we recommend referring to the original literature [21]

2.2.1 CIA Model. Inherited from the cybersecurity, **confidentiality**, **integrity**, and **availability (CIA)** model is a traditional way to form the basis for the analysis of security systems. In terms of ML, confidentiality is usually associated with the model assets (e.g., training data, model algorithm, and model parameter). Some of the confidentiality attacks seek to divulge the model's architecture or parameters, which may be recognized as significant intellectual property of the model holder [92]. Others attempt to expose the data used to train the model and may compromise the data source's privacy, e.g., the patients' clinical data used to train medical decision models is often of foremost privacy. Therefore, confidentiality attacks are more closely connected to privacy issues. Comparatively, integrity attacks jeopardize the model by false negatives, i.e., inducing deviated model output. Availability attacks try to deny access to valuable model outputs or system features via false positives, cf., denial-of-service (DoS) [124].

2.2.2 Threat Model. A more comprehensive approach involves the identification of a threat model, which encompasses assessing the threat surface (also known as attack surface) of ML-based systems and understanding the attacker's capabilities, goals, and specifications. By gaining insights into when, where, and how an attacker may attempt to compromise the system, we can better understand the threats and strategies associated with GAN-based attack methods. Building upon existing studies of GAN-based attacks, we present a novel taxonomy with regard to the threat model, including some redefined concepts.

Attack Objective. According to the attackers' objectives, threat models can also be classified into **poisoning attack** (also known as causative attacks), **evasion attack**, and **inference attack**. Both poisoning and evasion attacks attempt to undermine the integrity that makes ML models yield incorrect results (e.g., predicted label and information retrieval) given input, which can be categorized as security attacks through CIA model. In the realm of machine learning model surface[4], there are two main phases involved: **inference phase** and **training phase**. Attacks at the inference phase (also known as exploratory attacks) do not manipulate the model itself but attempt to either deviate it from yielding correct outputs (similar to the concept of integrity attack as aforementioned) or gather information about the model architecture or parameters. Attacks at the training phase seek to exploit or compromise the model itself. Therefore, the difference between poisoning attacks and evasion attacks lies in that a poisoning attack is launched at the training phase that injects adversarial training data samples with the aim of corrupting the learned model to yield wrong outputs [12, 79]; conversely, evasion attack is launched at the inference phase that manipulates the testing data samples to deceive previously trained model. Inference attack[5], also known as exploratory attack, refers to that the attacker can exploit leak information about the features of training data, which is considered as a threat to the confidentiality of model assets through CIA model. According to different features that the attackers intend to infer, inference attacks can be further categorized into: **preimage inference**, **class representation inference**, and **membership inference**. Given a ML model, preimage inference (also known as model inversion) targets reconstructing training data from model parameters in which the requirement of inference accuracy is usually high (e.g., pixel-level [158]) [40]. A special variant of preimage inference targets at reconstructing model rather than data (also known as model extraction attack), which attempts to obtain an adversarial model that is functionally and statistically equivalent and statistically close to the target model. Comparatively, class representation inference does not aim to reconstruct actual training data but only class representatives. Our taxonomy distinguishes preimage inference and class representation inference as separate concepts due to their different threat levels. Membership inference attempts to predict whether or not an exact data point (e.g., an image) is contained in the model's training dataset [107]. From a view of the population, property inference aims to learn from the model properties about the training dataset seemingly independent of the model's actual goal [80]; we classify it as a variant of membership inference in this

[4] Commonly, threat surface of machine learning includes: *physical surface*, *data representation surface*, and *machine learning model surface* [7]. Particularly, according to the usage of GAN, this survey falls within the realm of data representation and machine learning model surfaces.
[5] Note that "inference attack" is different from "attack at the inference phrase".

survey. Furthermore, some attacks incorporate the above threat models, such as morphing attacks [104], and recently-emerged Deepfake techniques [129]: they extract sensitive information (e.g., biological characteristics of a person) and generate fake examples which may further be used to fool the detection for illegal purposes.

Attacker's capabilities. From the perspective of attackers' capabilities (or **attacker's observations**), there are three different scenarios of attacks: **white**, **grey**, and **black boxes**. A while-box attack assumes the attacker has full information about the model assets. On the contrary, a black-box attack supposes the attacker has no information about the model assets, but (s)he can *query* from the victim model by API services which are usually provided by **Machine Learning as a Service (MLaaS) platforms**. Comparatively, a grey-box attack comes somewhere in between; the attacker knows partial information[6]. It is a common pattern that the attacker uses a surrogate model (shadow model) to mimic the target model locally and develop specific examples that can affect the target model due to the difficulty of directly manipulating the target model. White-box settings allow the attacker to construct an identical model to the target model. In contrast, in grey- and black-box settings, the attacker can only employ an architecture- or function-analogous mode, or *distill* a model [130]. Accordingly, the dangerous level of the three attacks is: Black > Grey > White.

Attack Specification. Furthermore, according to the specification of the attack, we can also categorize the threat model into **targeted** and **untargeted attacks** (also known as dodge attack). Targeted attacks mean the attack is forged towards an assigned and clear instance, while the untargeted attack is not. For an evasion attack that targets a multiple-classifier that classifies an animal image, a targeted evasion attack could be making the output label which is originally "dog" to "cat". In contrast, untargeted attacks only concern if the output is correct and arbitrary incorrect labels are acceptable. For an inference attack, a targeted attack can be the attacker who wants to infer a specific class of data examples. Note that the attack specification is a major consideration for classification tasks, and particularly, attacks to binary classification task is naturally deemed as targeted. In addition, membership inference attacks are regarded as targeted as the "membership" is undoubtedly a specific objective.

Target System. The threat model can target centralized or decentralized ML systems. From the level of the system, the threat model could target at **centralized** or **decentralized ML systems**.

In traditional ML, the efficiency and accuracy of models are determined by computational power and training data available on a centralized computing device (e.g., a server). With the increase in data volume and model complexity, a centralized system limited by computational power is arduous to undertake such complicated computing tasks. In addition, centralized systems are also struggling with S&P issues. For one thing, data holders may be reluctant to contribute their data to a centralized system since their data may contain a mass of private and sensitive information. For another, a centralized system stores all sensitive information in a central custodian (usually a central server), presumably encountering single-point failure. To overcome these problems, decentralized systems have grown in popularity over the years due to their distributed storage and parallel computing natures — especially the recently emerged federated learning has been a research hotspot due to its precise stroke on S&P issues of ML [135]. **Federated learning (FL)** is among the most widely-adopted decentralized ML system [78]. FL assumes a scenario that there are N data holders (in different tasks, they may be named as "clients", "workers", or "participants"), all of whom hope to train a ML model by merging their data $\{X_1, X_2, \ldots, X_N\}$. However, the privacy policies such as GDPR [117] do not allow any direct exposure of raw data, i.e., they cannot directly consolidate their data by $X_{\text{sum}} = X_1 \bigcup X_2, \ldots, \bigcup X_N$ and use it to train a model f_{sum}. In such a condition, a FL system describes a learning process where data holders train a model f_{Fed} collaboratively without sharing their respective raw data, and f_{fed} should achieve an accuracy very close to that of f_{sum}. To this end, [78] proposed an algorithm named FedAvg that allows the training data to be kept locally and learns a

[6] Many studies merge the grey-box setting into the white box or black box settings [11, 17, 84]. For a rigorous definition, we differ the three terms by defining the white box and black setting as extreme situations.

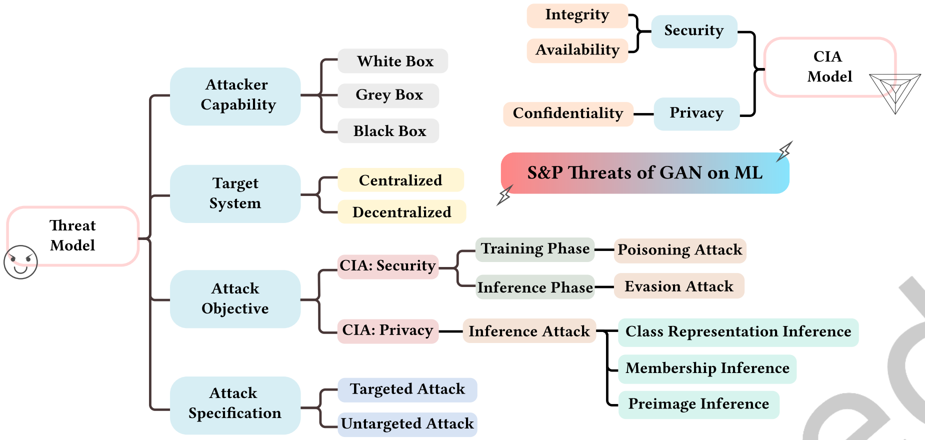


Fig. 4. S&P threats of GAN to ML from different perspectives.

shared model[7] by aggregating locally-computed updates by a central server. It is also worth mentioning that all participants of a FL system are actually operating in a grex-box setting, i.e., they can observe the changes of the shared model based on the received model parameters but nothing about the raw data. While FL can provide a certain level of data privacy by preventing a single entity from having a complete view of all training data, it is important to note that the ideal privacy-preserving effect can only be achieved under the assumption of all participants being trustworthy and the absence of malicious external parties. However, in practical scenarios, this ideal situation is often unattainable. Therefore, S&P attacks are also nonnegligible for FL systems.

3 GAN-BASED ATTACKS

In this section, we first review the existing studies in terms of GAN-based attacks according to their corresponding threat models, including the description of the attack patterns in different threat models. Subsequently, from another aspect, we discuss and compare the strategy of GAN-based attacks combing different properties/advantages of GANs. Lastly, we elaborate on and discuss GAN-based attacks against decentralized ML systems.

What are threats brought by GANs? Before our elaboration, given the properties of GANs and the S&P concerns in machine learning, we first identify several key reasons why GANs could pose significant threats, including the general principles of GAN-based attacks:

- Data manipulation and generation: GANs can be used to generate synthetic data that resembles real data. This poses risks when the generated data is used as adversarial examples to manipulate or deceive machine learning models. On the other hand, these data can also be used as illegitimate sample forgeries.
- Detection evasion: GANs can be used to generate examples specifically designed to evade detection by machine learning models. Leveraging the generator's ability to generate subtle modifications, attackers can create examples misclassified or ignored by detection systems, bypassing the defenses.
- Privacy breach: GANs can be used to learn and replicate sensitive information from private data, leading to privacy breaches. This becomes particularly worrisome in the context of MLaaS or decentralized ML systems due to the shared data and model information.

[7] In this paper, shared model, global model, and federated model are used interchangeably in the context of decentralized ML systems.

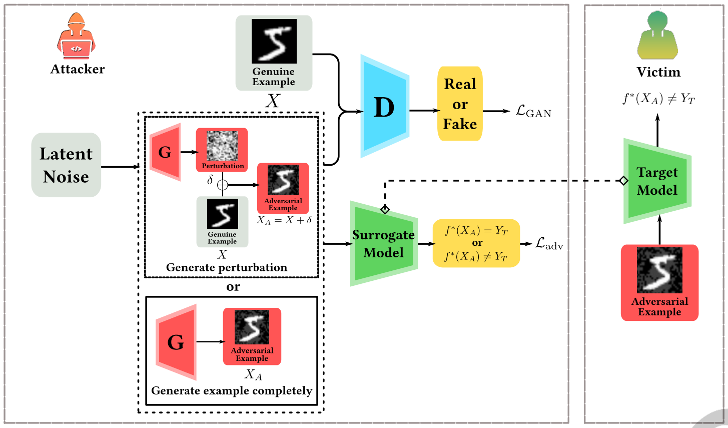


Fig. 5. GAN-based evasion attacks. Two typical patterns of GAN-based adversarial examples generation are illustrated in the dashed box: (1) Generate perturbations which are subsequently added to the genuine examples. (2) Generate complete adversarial examples. To improve the attack effect, the attacker generally integrates $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{adv}}$ and optimizes them in a bilevel manner where $\mathcal{L}_{\text{GAN}}$ here is used to encourage the adversarial examples to appear similar to the original data X by GAN while $\mathcal{L}_{\text{adv}}$ is used to prompt the adversarial examples to deviate the classification result of the target model.

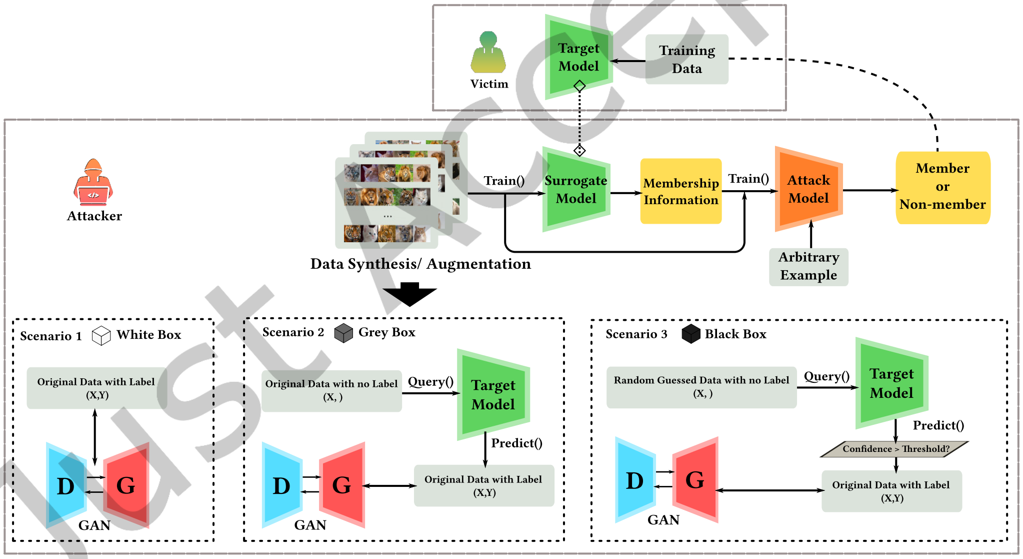


Fig. 6. GAN-based data synthesis for membership inference attacks [107]. Scenario 1 assumes that the attacker has a limited number of original data [144]. Scenario 2 assumes that the attacker has some unlabeled data records and can query the target model. Scenario 3 assumes that the attacker only can query the target model and use some randomized/guessed data to approximate the original data [5].

We list the potential S&P threats of GAN to ML systems from different aforementioned perspectives, as shown in Figure 4 — we highlight the aspects that GANs are particularly useful that break the limitations of traditional attack methods. In the following section, we will provide more in-depth discussions.

3.1 GANs in Different Threat Models

In this survey, we organize existing studies of GAN-based attacks based on the taxonomy illustrated in Figure 4. In this subsection, we first introduce related studies accordingly, which is also summarized in Table 2.

3.1.1 GANs in Evasion Attack and Poisoning Attack. GANs are widely adopted in **generating adversarial examples** for evasion attacks [6, 20, 58, 67, 72, 76, 109, 122, 123, 128, 130, 134, 139, 150, 151, 154, 155, 159]. Towards a trained model $f^*()$, GAN-generated adversarial examples X_A are expected to be classified as $f^*(X_A) \neq Y_T$ (untargeted attack), where Y_T denotes the true label; or $f(X_A) = Y_{\text{Tar}}$ (targeted attack) where Y_{Tar} denotes the target class. Furthermore, GANs are usually expected to develop adversarial examples X_A, which are more plausible (looks more natural) to avoid detection compared with traditional methods [154]. Conventional practices in evasion attacks attempt to generate perturbations (e.g., noises or backdoors) denoted by δ and add them to genuine test data samples as adversarial examples, i.e., $X_A = X + \delta$ [15, 36, 46]. Similar works based on GAN can be referred to [20, 72, 122, 128, 130, 150, 151, 155] (see upper dashed box in Figure 5). Differently, [6, 109, 123, 154] took full advantage of GANs that synthesize adversarial examples entirely from scratch (cf. adversarial autoencoding in [6], and see lower dashed box in Figure 5). Beyond simple image classification tasks, Wang *et al.* [122] devised a multi-task GAN whose generated adversarial examples can fool holistic scene understanding tasks (including semantic segmentation, object detection, and classification). As practical applications, Mangaokar *et al.* [76] proposed a method that can produce fake medical images to deceive both the machines and human beings (doctors) to the point of making incorrect diagnoses; [58, 67, 139] developed GAN-based models can generate malware that can bypass detection algorithms; Targeting at the image retrieval systems, [150, 151] attempted to generate adversarial examples that make retrieval results completely dissimilar from the query image.

On the other hand, [20, 64, 83, 134, 142, 143] employed GANs in the poisoning attacks. A general pattern can be described as follows. The generator G targets at generating poisoning data samples X_A which maximizes the error of the target model, meanwhile minimizes the accuracy of discriminator D that distinguishes the poisoning data samples from genuine data samples. The target model $f(\cdot)$ seeks to minimize the training loss evaluated on a training dataset that incorporates several poisoning data samples. The attacker attempts to generate adversarial training samples plausible to genuine data, and these poisoning data samples will be injected into the target model's training dataset. The model trained with poisoning examples can be formulated as $f_p^*(\cdot) \xleftarrow{X_A} f(\cdot)$, which is expected to yield wrong outputs given genuine test samples. Muñoz-González *et al.* [83] proposed an approach that adopts CGAN with assistant target label information to generate poisoning samples, which can be referred to as a typical pattern of GAN-based poisoning attack (see Figure 8). Zhang *et al.* [142, 143] introduced a GAN-based poisoning attack in a federated learning scenario by deploying GAN locally to generate poisoning data samples to corrupt the shared model. With the assistance of reinforcement learning, the GAN-based method proposed by Yang *et al.* [134] significantly improves the generation rate of poisoning data compared with gradient-based attack methods. Kasichainula *et al.* [64] applied GAN-based poisoning attack to corrupt text-to-image tasks. Comprehensively speaking, GAN-based poisoning attack methods share a similar philosophy with GAN-based evasion attack methods, where the difference only lies in the usage of the generated adversarial examples.

3.1.2 GANs in Inference Attack. The model information, such as parameters and gradients, are informative intermediaries that establish connections between the input and output of a model. GANs can effectively exploit these traces left by the training data within the model to extract private information. This characteristic makes GANs practical for conducting inference attacks [55, 56, 96, 108, 111, 125, 154] proposed to infer the class representation of data using GANs. To develop more natural adversarial examples, Zhao *et al.* [154] introduced an inverter (denoted as $I(\cdot)$) replacing the discriminator in the GAN to learn dense representations of given samples. The learning process involves minimizing the reconstruction error γ of real samples X, which can be formulated as

$$\min_{\gamma} \mathop{\mathbb{E}}_{X\sim p_{\text{data}}(X)} \|G(I_{\gamma}(X)) - X\| + \lambda \mathop{\mathbb{E}}_{Z\sim p_{Z}(Z)} [\mathcal{L}(Z, I_{\gamma}(G(Z)))]. \tag{7}$$

Hitaj *et al.* [55] and Wang *et al.* [108, 125] attempted to infer class representation of other participants' training data in federated learning (FL) scenarios. In a similar FL scenario, Ren *et al.* [96] treated the attack as a regression problem by integrating GANs and a gradient-based method [158]. Aiming at cybersecurity, Hitaj *et al.* [56] also proposed to use GANs to extract the representation of genuine passwords from password information leaks and to further produce high-quality password guesses.

For a more fine-grained goal, [2, 8, 96, 106, 111, 148] used GANs to conduct the preimage (model inversion) inference. The basic idea among these studies is using the generator of GANs to reconstruct the target examples X_{re} by $X_{\text{re}} = G(Z)$. Basu *et al.* [8] tried to recover input data analogous to those used to train the target model by training a surrogate model, and Aïvodji *et al.* [2] investigated the same case in a black box scenario. Shi *et al.* [106] proposed to first mount an exploratory attack on the target model and collect returned labels to construct training data examples, and then use these data to train an adversarial model that is functionally and statistically equivalent close to the target model. Zhang *et al.* [148] introduced partial public information into their GANs to improve the inversion effect. Furthermore, they provided theoretical evidence to support the notion that highly predictive models are more susceptible to preimage inference attacks. Ren *et al.* [96] leveraged the concept of GAN to develop a generative model at the central server of a FL system and demonstrated it could recover image-based privacy data from the shared gradient only. Sun *et al.* [111] investigated a malicious client which can use GANs to recover the training data of other clients from the shared model.

[5, 108, 125, 132, 144] leveraged GANs to conduct membership inference attacks. Following the attack pattern in a restricted black-box scenario [107], Bai *et al.* [5] proposed to use GAN to **augment the training data** used to train a surrogate model and further predict the membership of the synthesized data sample. The concept of utilizing GANs for data augmentation has also been embraced in subsequent studies concerning GAN-based membership inference attacks. Targeting at FL systems, Wang *et al.* [108, 125] proposed a GAN with a multi-task discriminator owned by a malicious server which is enabled to infer the client identity of a training sample. On the other hand, Zhang *et al.* [144] proposed a client-launched membership inference attack whose targets are other clients in a FL system. The local training data of a client may come from multiple data sources (e.g., a person or certain environment) — Xu *et al.* [132] proposed a client-launched membership inference attack to infer the source-level membership of other clients' local data records.

Observations. We have several direct or indirect observations from our investigation on the referenced GAN-based attack studies, such as:

- For black-box and targeted attacks, latent space-enhanced GANs like CGAN seem to be requisite in related studies.
- When considering more restricted black-box scenarios, unsupervised GANs like InfoGAN are usually adopted.
- In black-box evasion attacks, a common procedure for the attacker involves querying the target model to obtain labels for generated adversarial examples and subsequently training the discriminator.
- While GANs have shown the ability to generate entire adversarial examples, most studies prefer using GANs to generate adversarial perturbations for evasion attacks.
- In poisoning and inference attacks, GANs are adopted for large-scale data synthesis/augmentation.

We further discuss these observations and identify the factors behind them in the next Section.

Table 2. Summary of studies in terms of GAN-based attack.

Authorship & Year	**Threat Model**			**System**	**GAN Model**[8]	**GANs' Usage**	**Countermeasure-inclusive**
	A-C	**A-S**	**A-Obj**				
Hitaj *et al.* [55] 2017	▨	T	CRI	DeC	CGAN	Generate adversarial examples	✓
Baluja *et al.* [6] 2017	□	T	EA	C	GAN	Generate adversarial perturbations/examples	–
Hu *et al.* [58] 2017	■	T	EA	C	InfoGAN	Query the target model for labeling generated examples	–
Yang *et al.* [134] 2017	□	UnT	PA	C	GAN	Generate adversarial examples	✓
Zhao *et al.* [154] 2018	■	UnT	EA,CRI	C	WGAN	Generate adversarial examples	–
Xiao *et al.* [130] 2018	▨ ■	T, UnT	EA	C	CGAN	Generate adversarial perturbations	✓
Song *et al.* [109] 2018	□ ■	T, UnT	EA	C	ACGAN	Generate adversarial examples	✓
Shi *et al.* [106] 2018	■	T, UnT	PreI	C	CGAN	Augment data examples for inversion	✓
Zhang *et al.* [142, 143] 2019	▨ ■	T, UnT	PA	DeC	Semi-GAN	Generate adversarial examples	✓
Wang *et al.* [108, 125] 2019	▨	T	CRI, MI	DeC	CGAN	Reconstruct data examples with multiple properties	–
Wang *et al.* [123] 2019	▨	T,UnT	EA	C	ACGAN	Generate adversarial examples	✓
Zhu *et al.* [159] 2019	□	T,UnT	EA	C	CycleGAN	Generate style-differed adversarial examples	–
Aïvodji *et al.* [2] 2019	■	T	PreI	C	BEGAN	Reconstruct label-specific data examples	–
Basu *et al.* [8] 2019	□	T	CRI	C	GAN	Reconstruct label-specific low-dimensional data representations	–
Muñoz-González *et al.* [83] 2019	▨ ■	T	PA	C	CGAN	Generate adversarial examples	–
Hitaj *et al.* [56] 2019	▨	T	PreI	C	WGAN	Reconstruct data examples	–
Zhao *et al.* [151] 2019	■	UnT	EA	C	CGAN	Generate adversarial perturbations	–
Li *et al.* [67] 2019	■	T	EA	C	InfoGAN	Query the target model for labeling generated examples	✓
Liu *et al.* [72] 2019	▨ ■	UnT	EA	C	InfoGAN	Generate life-like adversarial perturbations	–
Wei *et al.* [128] 2019	■	UnT	EA	C	CGAN	Generate adversarial perturbations	–
Xu *et al.* [132] 2020	▨	T	MI	DeC	CycleGAN	Reconstruct mapping between gradients and data examples	–
Zhang *et al.* [144] 2020	▨	T	MI	DeC	GAN	Augment data for training attack model	–
Zhang *et al.* [148] 2020	□	T	PreI	C	InfoGAN	Reconstruct label-specific data examples	–
Chen *et al.* [20] 2020	□ ■	T, UnT	EA	C	CGAN	Generate adversarial perturbations	✓
Mangaokar *et al.* [76] 2020	□ ■	T	EA	C	CycleGAN	Style-transfer the prediction-related information but preserves the identity information	✓
Yuan *et al.* [139] 2020	■	T	EA	C	InfoGAN	Query the target model for labeling generated examples	✓
Zhou *et al.* [155] 2020	▨ ■	T	EA	C	CGAN	Generate adversarial perturbations	✓
Zhao *et al.* [150] 2020	▨	T	EA	C	CGAN	Generate adversarial perturbations	–
Bai *et al.* [5] 2021	■	T	MI	C	GAN	Augment data for training attack model	–
Kasichainula *et al.* [64] 2021	▨	UnT	PA	C	GAN	Generate feature space-differed adversarial examples	–
Ren *et al.* [96] 2021	▨	T	PreI, CRI	DeC	WGAN	Reconstruct label-specific data examples	✓
Sun *et al.* [111] 2021	▨	T	PreI, CRI	DeC	GAN	Reconstruct label-specific data examples	–
Wang *et al.* [122] 2021	□ ■	T	EA	C	Multi-task GAN	Generate data examples with multiple properties	–

Abbreviation: **A-C**–Attacker's Capabilities, □–White-box Attack, ▨–Grey-box Attack, ■–Black-box Attack,
A-S–Attack Specification, **T**–Targeted Attack, **UnT**–Untargeted Attack
EA–Evasion Attack, **PA**–Poisoning Attack, **CRI**–Class Representation Inference, **MI**–Membership Inference, **PreI**–Preimage Inference,
C–Centralized System, **DeC**–Decentralized System

3.2 GANs Properties and Advantages in Attacks

3.2.1 Stealthiness and Utility. The adversarial learning and mimicry-synthesis nature of GANs, as elaborated on in Section 2.1, makes it a promising technique for conducting stealthy and data-utilizable attacks. Generally, an adversarial attack does not tamper with the target model but adds subtle disturbances imperceptible to humans to the input samples, causing the model to give an incorrect output with high confidence. Thus, GAN-based attacks inherit this property that naturally makes themselves circumvent model-oriented defense mechanisms such as intrusion detection or anomaly diagnosis [17, 89]. That is also the reason why perturbation generation (see Figure 5) is more popular among GAN-based attacks since such perturbations are usually pixel-level, which is more difficult to be identified.

On this basis, GANs outperform conventional generative models in adversarial attacks in many ways. For evasion or poisoning attacks, while some generative models such as variational autoencoder (VAE) can develop adversarial examples exhibiting "blinding stains" in ML models, they may be unnatural (e.g., the examples generated by VAE are much more blurred than GAN's[9]) — as the worst case, its utility is slashed, and even out of the instances that the classifier can handle. Especially in complex domains such as linguistics, enforcing grammar and semantic similarity is difficult when perturbations are unnatural [68]. Additionally, these unnatural examples will likely be recognized by machines or human beings, making the attack rumbled [154]. For GANs, even if the generator develops perturbations rampantly, the discriminator can regulate the generation to ensure

[8] Note that the GAN model adopted in the literature may not be fully the same as the associated GAN's prototype; however, the notion of the associated GAN's prototype is at least borrowed to a great extent. Table 3 is explained in the same way.

[9] https://github.com/hwalsuklee/tensorflow-generative-model-collections

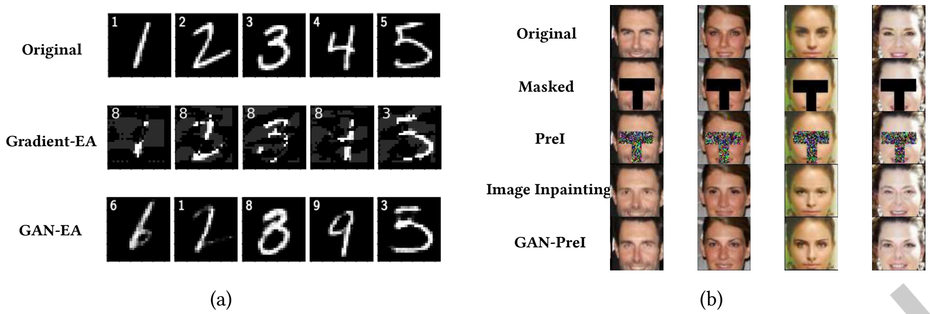


Fig. 7. Comparison of examples generated by GANs and other methods. (a) In evasion attack [154]. The referenced gradient-based approach is Fast Gradient Sign Method (FGSM) [46]. We can observe that the examples generated by GAN resemble more closely the real ones. (b) In preimage inference from masked images [148]. The referenced preimage inference method (the third row) is from [39]. When most of the identifiable features (eye, nose, and mouth) are hidden, the primitive preimage inference method [39] fails to reconstruct the original image. Comparatively, GAN-based approaches can recover the original image to a great extent, even better than image inpainting.

that the generated examples are small and unnoticeable. Massive prior studies have demonstrated that samples generated by GANs show a remarkable resemblance to the real samples in the training data, primarily due to the effectiveness of adversarial training (refer to Figure 7(a) for an example), which enables more stealthy attacks. Meanwhile, considering that to minimize the difference between generated examples and real ones, the optimization process of a stealthy attack usually introduces constraints respecting the level of perturbation such as $||X - X_A|| \leq \epsilon$, GANs can more effectively search the corrupting/deviating spots in such a narrow candidature space [38].

For preimage inference attacks, attackers try to reconstruct the private data samples from the leak gradients or outputs; however, the reconstructed examples may slightly resemble the actual data that determined the class or identity [148]. The reason is that conventional approaches accurately classify the broad areas of the input space. However, they may overlook or miss some other components or fine-grained details shown in the data [65]. Then the attacker may mistakenly believe that (s)he has reconstructed important information for that class when, in fact, (s)he has only obtained useless data. In comparison, GANs can **recover a larger portion of the space that contains the targeted sensitive information with greater certainty** (see an extreme instance in Figure 7(b) where sensitive information of the original image is masked), making the generated examples more utilizable.

From the perspective of attack efficiency, GANs are capable of parallelizing the generation while some other approaches (e.g., autoregressive-based generative models [101, 116]) are not, which can accelerate the attack process and makes the attack completed before the defense reaction in practical scenarios.

3.2.2 Transferability. An important criterion of traditional perturbation-based attacks is their transferability across different classifiers, and GANs have demonstrated their efficacy in terms of the transferability [76, 109, 123, 130, 134, 159].

On the one hand, the **transferability is with regard to the accessibility to the target model or data**, which depends on the attacker's observations. Simpler conditions happen in white- or grey-box attacks where the attacker can fully access or at least have some partial knowledge of the target model or data. In this case, GANs can create high-quality adversarial examples that apply to the target model without confronting any insurmountable barriers to the input-output relations. For black-box attacks, the concept and practice of surrogate models (a.k.a. shadow models) are introduced as a transferability strategy to bypass the information barrier [94].

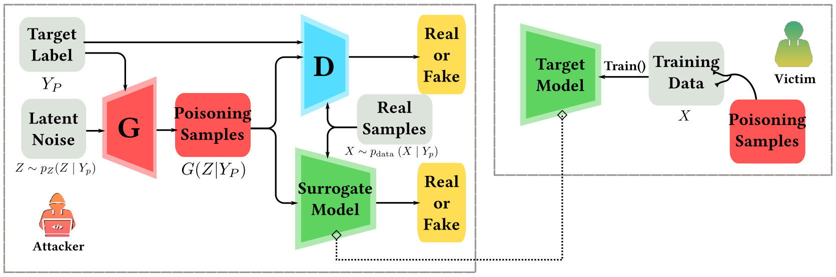


Fig. 8. An instance of GAN-based poisoning attack using auxiliary information [83].

A surrogate model can mimic the target model's behavior at the attacker's side, and a naive selection method could be empirical (e.g., a CNN is suitable for image classification while a recurrent neural network (RNN) is suitable for time series prediction). For evasion or poisoning attacks, it enables an attacker can first train a surrogate model and then generate adversarial examples against it, hoping the same adversarial examples will also be able to attack the other models [73, 82]. For inference attacks, the attacker constructs such a surrogate model to infer useful information about the target data or model. As the scenarios of membership inference attack illustrated and summarized in Figures 6, GAN can prompt massive data synthesis/augmentation guaranteed no matter under which conditions. For instance, even in a very restricted scenario (cf. Scenario 3 of Figure 6), GANs can produce utilizable data using the initially randomized data to train the surrogate model so that the prediction of a surrogate model can serve an accurate membership classification. Meanwhile, for a stronger black box attack effect, researchers are endeavoring to develop a better surrogate model matching algorithm [146] or avoid the adoption of surrogate model [100] in the restricted cases, which is to release the potential further and improve the GAN-based black box attack's effect.

On the other hand, transferability refers to the ability of knowledge learned from an accessible data source by the attacker to be applied in attacks. GANs have been proven to have a strong transfer capacity due to their deep neural network nature [24, 74, 127]. For the GAN-based attacks in which accessible information about the target model/data is limited, the integration with **transfer learning** techniques plays a significant role. The attacker can use a pre-trained GAN with massive prior knowledge to generate plentiful, high-quality, and diverse adversarial examples with great rapidity [123]. For example, Zhu *et al.* [159] transferred non-makeup images to makeup images where the perturbation information of the attack is hidden in the makeup areas. Mangaokar *et al.* [76] devised a transfer framework that takes a biomedical image of a patient as input and translates it to a new image that indicates an attacker-chosen disease condition. Some attack methods adopted the style-transfer-enabled variants of GAN for their attack purpose. For example, Pix2Pix is based on CGAN, designed for image-to-image translation and can construct a mapping between the source and target domains with paired training examples [59]. Wei *et al.* [128] leveraged a loss-enhanced Pix2Pix to generate adversarial examples, which shows good transferability when evaluated on different target models. CycleGAN is another transfer-enabled variant of GANs for image-to-image translation, which can handle the unsupervised challenge without paired training examples. Considering the condition of unpaired training examples in federated scenarios, Xu *et al.* [132] treated property vector as the source to obtain gradient as the target by CycleGAN. Another applicable technique regarding transferability is **knowledge distillation**. As many MLaaS platforms allow query accesses to the model, Xiao *et al.* [130] proposed to seize query-based accesses to distill the target model (model distillation) into a GAN to construct adversarial samples, whose efficiency surpasses the conventional transferability-based attack.

3.2.3 Latent Representation and Auxiliary Information. A significant advantage of GANs over some other generative models is the ability to efficiently utilize **latent representations** and **auxiliary information** [13]. As introduced in Section 2.1, the generator of GANs is designed by taking latent code (e.g., random noise) as the input, which processes a transition from low to high dimension. Such dimension-up problems are involved in different attack tasks, and thus GANs can play a part. For example, an attacker attempts to produce completely new images as adversarial examples for evasion attacks [109, 154] or synthesize a relatively high-dimensional gradient vector from a relatively low-dimensional feature vector as did in [132].

GANs can be used for more fine-grained and targeted attacks [20, 109]. We know that the primitive GAN is typically fed with random noise Z without intended semantic information. As a result, there is no guarantee that the generated examples will exhibit high interpretability or convey meaningful indications. Therefore, primitive GAN is not quite capable when adversarial examples with an assigned class representation are expected to be developed, or high-fidelity deceptive examples are required. Latent space-enhanced GANs, as introduced in 2.1.2 significantly solve the problem by introducing additional information (e.g., class labels) as constraints to intervene in the GANs' training. Such a pattern can be seen in the poisoning attack method proposed in [83] (see Figure 8). The attacker selects a set of target class labels Y_P. Y_P is used to constrain the input noise Z and real data X, and thus the input data will be sampled from distribution $X \sim p_{\text{data}}(X \mid Y_p)$ and $Z \sim p_Z(Z \mid Y_p)$. The objective function of GAN is

$$V(D,G) = \mathop{\mathbb{E}}_{X \sim p_{\text{data}}(X \mid Y_p)}[\log D(X \mid Y_P)] + \mathop{\mathbb{E}}_{Z \sim p_Z(Z \mid Y_p)}[\log(1 - D(G(Z \mid Y_P)))]. \tag{8}$$

In this way, the generator can produce poisoning examples with specific classes.

Furthermore, the demanding granularity requirement for auxiliary information in GANs poses a challenge, limiting the applicability of GANs to certain restricted scenarios. For instance, Zhang *et al.* [148] demonstrated the validity that only leveraged partial public information as auxiliary information, which can be very generic, to learn a distributional prior via generative adversarial networks (GANs) and use it to guide the inversion process. In an unsupervised manner, Liu *et al.* [72] considered the perceptual sensitivity of the target model to the adversarial patch and leveraged the attention mechanism to *learn* an effect attack area of an image as auxiliary information; with the *learned* auxiliary information, the attacker can develop powerful adversarial examples by placing the generated adversarial patch to this attack area. Both the above methods employed InfoGAN as it can exploit unrestricted latent code as auxiliary information (e.g., the learned target areas of an image by attention mechanism in [72]) when no specific auxiliary information (e.g., labels) can be used. In a nutshell, GAN-based attacks can be effective in both a warm start (the attacker has some knowledge) and a cold start (the attacker has little or no knowledge).

3.3 GAN-based Attacks in Decentralized Systems

Table 3. Comparison of GAN-based attacks in FL systems.

Authorship & Year	Attacker	Victim	Threat Model	Abidance of FL Protocol	GAN Model
Hitaj *et al.* [55] 2017	Client	Targeted client	CRI	Passive, Active	CGAN
Zhang *et al.* [142, 143] 2019	Client	Central server, Client	PA	Active	GAN
Wang *et al.* [108, 125] 2019	Central server	Targeted client	CRI, MI	Passive, Active	CGAN
Xu *et al.* [132] 2020	Client	Targeted client	MI	Passive, Active	CycleGAN
Zhang *et al.* [144] 2020	Client	Arbitrary client	MI	Passive	GAN
Ren *et al.* [96] 2021	Central server	Targeted client	PreI, CRI	Passive	WGAN
Sun *et al.* [111] 2021	Client	Targeted client	PreI, CRI	Active	GAN

Abbreviation: **PA**–Poisoning Attack, **CRI**–Class Representation Inference, **MI**–Membership Inference, **PreI**–Preimage Inference

Decentralized ML systems have gained widespread adoption in practical scenarios due to their scalability and privacy-preservation ability. However, attacks on decentralized ML systems are a cause for concern [75]. Particularly, the data distribution mimicry-enabled property of GANs can effectively help attackers cross the barriers among different entities of decentralized ML systems, even if the entities exist as "information-isolated islands". Therefore, we investigate GAN-based attacks against decentralized ML systems in this subsection. As one of the most typical, prevalent, and extensively researched decentralized ML systems [135], we primarily focus on **federated learning (FL) systems**. In addition to different attack objectives (e.g., poisoning attack and inference attack), the attacks against FL systems can be categorized into passive and active attacks. Meanwhile, the attacks can be launched by inner participants (i.e., client and central server) or outsiders (e.g., eavesdroppers). In this section, we summarize the existing studies according to the above taxonomy in Table 3 and discuss them in detail as follows.

3.3.1 Attack from Insider v.s. Attack from Outsider. GAN-based attacks can be conducted by **outsiders** or **insiders** of a FL system. Given the typically large number of participants in a federated learning system, it is highly likely that one or more participants may act maliciously. Insider attacks comprise those conducted by the central server and the clients of FL systems. Generally, insider attacks intend to be more impactful than outsider attacks. This is because insiders can easily and imperceptibly acquire valuable information from the FL system. They can utilize this privileged access to develop more effective attack models [75]. FL does not allow the participants to share their raw data; however, some may cast greedy eyes on others' data. In horizontal FL, the attacker, who is one of the clients, may target the identity information of other clients. In vertical FL, the targets of the attacker may be other clients' data features that (s)he does not have [135]. GANs empower attackers to locally produce mimic data samples. More devastating attacks could be orchestrated through the collusion of multiple malicious clients, such as in the case of a Byzantine attack [37]. In Byzantine attacks, one potential threat is that the clients can manipulate their outputs that mimic the distribution of the expected model updates by GANs; this allows clients to conceal their malicious activities effectively. Furthermore, an attacker can launch the Sybil attack [41] that simulates multiple forging clients to conduct more disruptive attacks than the single one. In another case, the attacks can be from the central server. The malicious central server can have similar motives to the clients, as the central server generally knows nothing about the data, making it covet the raw data from clients. As the federated scenario assumed in [125], the clients are required to upload their local model updates to the central server, which enables the central server to utilize GANs to recover the local data based on these local model updates.

Outsider attacks, similar to the term "man-in-the-middle attacks", are initiated by eavesdroppers who lurk within the communication channel between the central server and clients, or among the actual service users of the federated model in a MLaaS platform. Outsiders typically lack prior knowledge about the shared model, resulting in the attacks are usually black-box [58]. Furthermore, considering the lack of legitimacy for outsiders, their intrusive behaviors are typically subject to stringent system monitoring. Accordingly, the challenges of GAN-based attacks from outsiders reside in two main aspects: black-box inference and the ability to evade the detection mechanisms [97].

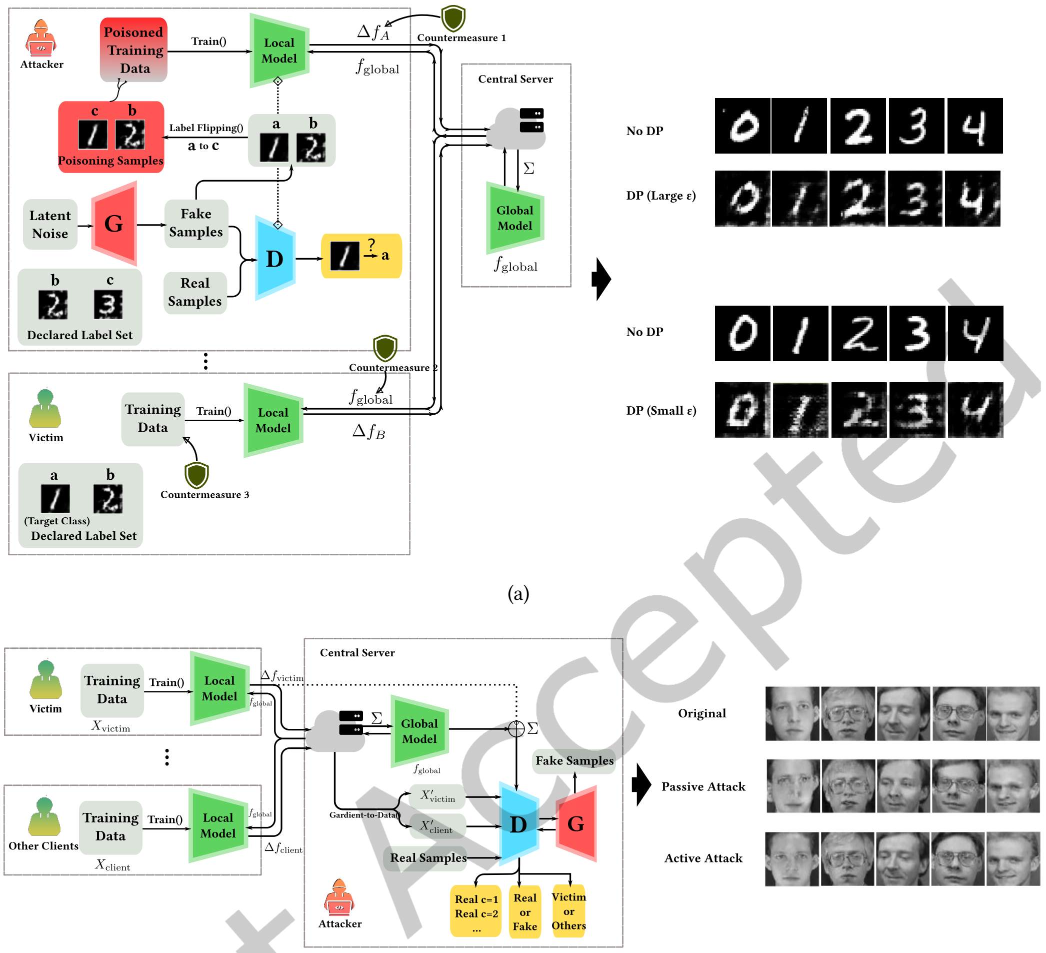
3.3.2 Privacy Attack v.s. Security Attack. Privacy attacks are a major threat to FL systems. Not only do malicious participants exist, but there are also honest-but-curious participants who attempt to infer the private information of other participants. We know that attackers need to query target model assets as much as possible to infer privacy information; however, for centralized systems, frequent queries from attackers who are outsiders to the target model inevitably raise alerts. By contrast, the protocol of FL system allows participants to access the shared model in each training iteration. If this is the case, an attacker who is in the disguise of a benign participant in a FL system can legitimately and repeatedly access the shared model, enabling them to execute the inference attack unobstructedly. A pioneering work of inference attacks against FL systems based on GANs can be referred to

[55], which is developed according to the above intuition. An illustration of the scheme proposed in [55] is shown in Figure 9(a). The attacker pretends to be a benign participant (client), who tries to steal private information from other participants. Given that GANs can learn the distribution from the classifier's output without the knowledge of the data, the attacker secretly deploys a GAN to induce the victim into disclosing more knowledge about a class that the (s)he is unfamiliar with. As the generated samples increasingly resemble the target class, it becomes increasingly challenging for the global model to accurately classify the target class. Consequently, this facilitates the disclosure of more information about the target class. Thus, the attacker can leverage the disclosed information to improve the generator, generating samples that closely resemble the target class for the purpose of privacy theft.

However, the inference attack approach proposed in [55] is limited. First, the global model's architecture is modified, and thus the learning process is impacted where a powerful attacker has to be assumed in this scenario. Second, the attack cannot obtain the exact samples from the victims but rather general samples that describe the properties of the targeted class, which cannot achieve a user-level privacy attack. Wang *et al.* [125] proposed to stride the two limitations. The schematic of the proposed approach is shown in Fig. 9(b). The attacker in this work is assumed to be the central server which covertly employs a GAN to create fake samples that confuse the clients. To achieve a user-level privacy attack, a multi-task-enabled GAN is introduced where the victim's reality, category, and identity are all considered. The principle of the multi-task-enabled GAN draws inspiration from CGAN, where the noise input to the generator and the real samples input to the discriminator are both conditioned on category and identity information. For the identity information, since the central server cannot directly access the samples from different clients, a gradient-based data reconstruction approach is adopted to recover the clients' local data at the central server, which is subsequently utilized for training the GAN. To provide GAN with auxiliary information, the attacker employs a dataset with real samples. In this scheme, the attacker can train the GAN imperceptibly and filch the clients' privacy information without modifying the global model and compromising the FL system.

FL systems have also been investigated as vulnerable to security attacks such as poisoning attacks [9, 142]. This is because FL systems ask the participants not to share their data. Consequently, the data and training process is invisible to the central server, and the security authenticity of a certain client's update cannot be verified. Nevertheless, conventional poisoning attack approaches assume that attackers possess the validation dataset that shares the same distribution as the training dataset [83], which is considered impractical in federated scenarios. A GAN-based poisoning attack method against FL systems is proposed by Zhang *et al.* [142] (see Figure 10). The attacker uses GAN to generate sufficient labeled examples and transform them into poisoning samples by label flipping. The attacker's objective is to compromise the global model by uploading poisoned local model parameters; this act subsequently distorts the original distribution of raw data and influences the inferences made by the learning model.

Summarily, it is also worth noting that the architectures of the above-introduced poisoning attack and the aforementioned GAN-based inference attack [55] exhibit similarities. First, both approaches employ GANs to generate samples that mimic the ones of the victim. Second, they employ label flipping techniques to change the labels of these generated samples to a specific class, where the attackers' models are trained using these manipulated data to generate specially-crafted gradients. However, their targets and the underlying motivations for employing the technique are significantly different. GAN-based poisoning attacks aim to maximize the attack effects (e.g., model performance degradation), such as causing model performance degradation, by uploading specifically crafted gradients. In contrast, GAN-based inference attacks, such as [55], strive to utilize meticulously designed gradients to maximize the amount of leaked information from the victim's local data in a deceptive manner. More generally, for whichever approach, we can conclude that while the attacker has no prior knowledge about the victim in a FL system, (s)he can **leverage the shared model and GAN's synthesizing capacity to compromise the victim indirectly**.



(a)

(b)

Fig. 9. Two instances of GAN-based inference attack in FL systems. (a) The attack launched by a client [55]. The victim declares his label set $[a, b]$. The attacker declares his label set $[b, c]$. If this, then that class b is the shared one. The objective of the attacker is to deduce as much meaningful knowledge about components in class a without any prior knowledge. The attacker uses a GAN to create samples that resemble the victim's samples from class a and introduces these synthetic samples labeled a into the FL procedure. Consequently, the victim needs to differentiate between classes a and c as far as possible, making him expose more knowledge about class a than supposed. The "shields" highlight three countermeasures to this attack (refer to Section 4.2). (b) The attack launched by the central server [125]. The discriminator of the devised GAN can handle three tasks: real-fake classification, categorization classification, and, most importantly, identification classification that can differentiate the victim from other participants, which cannot be handled by the method in [55]. X'_{victim} and X'_{client} denote the recovered training data of the victim and other clients, respectively. The discriminator at the central server is developed by sharing the same model structure of the global model or further aggregating with the victim's updates (except the output layer).

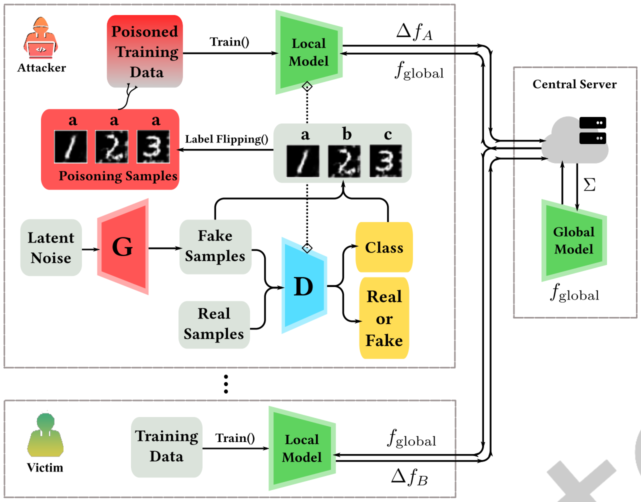


Fig. 10. An instance of GAN-based poisoning attack in FL systems [142]. The attacker first receives the global model f_{global} and develops a secret replica of f_{global} as the initialized discriminator D. Then, the generator G synthesizes fake data samples and thereafter inputs these samples to D. The fake data samples are classified as corresponding classes Y_O by D. These samples are assigned to the target class labels Y_P by label flipping and finally stored in the training dataset of the attacker. Since the FL is executed iteratively, the generator G of the attacker can produce plenty of synthetic samples analogous to the original samples. The attacker can train its current local model with the poisoned data and upload Δf_A to corrupt the global model.

3.3.3 Active Attack v.s. Passive Attack. For the cases in which insiders launch attacks, according to their abidance of FL protocol, the attacks could be *active* or *passive*. In passive cases, attackers would not break the FL protocol — simply observe the updated model parameters and performs inference without changing anything in the local or global collaborative training procedure. Contrarily, in active cases, attackers would break the FL protocol and perform a more powerful attack against other participants. In addition to the case introduced in Section 3.3.2, Wang *et al.* [125] introduced an active case that the attacker (central server) configures an affiliated server that is only connected to the victim. In this manner, the victim is effectively isolated from other clients, enabling the central server to launch more targeted attacks toward the victim. Thus, GANs can generate higher-quality samples (see the comparison in the right part of Figure 9(b)) attributed to the concentration on the target real samples. Comparatively, passive attacks are more challenging than active attacks due to attackers' restricted capabilities and authority.

4 COUNTERMEASURE TO GAN-BASED ATTACKS

To mitigate or eliminate the adverse impact on the utility, performance, and privacy of model assets and stakeholders caused by GAN-based attacks, different strategies have been adopted or devised as countermeasures (defenses) against the attacks. In this section, we review and discuss the countermeasures to GAN-based attacks, which are performed as follows: we first introduce the conventional and commonly-adopted countermeasures to the attacks on ML models/systems. Then, we present some specific countermeasures to general adversarial attacks and GAN-based attacks. Lastly, we particularly introduce some countermeasures to GAN-based attacks against decentralized systems. A collection of countermeasure studies to GAN-based attacks can be seen in Table 4.

Table 4. Summary of studies with regards to countermeasures to GAN-based attack .

Authorship & Year	**Threat Model**	**System**	**Initiative**	**Strategy**
Yang *et al.* [134] 2017	PA	C	Passive	Assess the training loss of an indeterminate training example.
Yan *et al.* [133] 2019	CRI	DeC	Active	Add backdoor layer to the shared model and observe its change.
Ching *et al.* [23] 2020	CRI, PreI	DeC	Passive	Partition the shared model and make the partitions trained by the both client and server.
Xiong *et al.* [131] 2020	CRI	DeC	Active	Detect abnormal gradient updates to identify the attacker.
Zhang *et al.* [147] 2020	CRI	DeC	Passive	Create fake samples by GAN to train the shared model and further obfuscate the attacker.
Chen *et al.* [22] 2020	CRI	DeC	Passive	Isolate the participants from the actual model parameters by introducing a trusted third party.
Chen *et al.* [20] 2020	EA	C	Passive	Combination of adversarial training [46], thermometer encoding [16], and large-margin GM loss [120].
Mangaokar *et al.* [20] 2020	EA	C	Active	Check disparities in color components [66] (blind); mesoscopic-level examination [1] (supervised).
Jiang *et al.* [62] 2020	EA	C	Passive	Use CycleGAN to improve attack and defense capacity mutually.

Abbreviation: **EA**–Evasion Attack, **PA**–Poisoning Attack, **CRI**–Class Representation Inference, **PreI**–Preimage Inference, **C**–Centralized System, **DeC**–Decentralized System

4.1 Generic Countermeasures

The threat of GAN-based attacks has garnered significant research attention and raised widespread concerns. The conventional defense approaches against S&P attacks can be generally categorized into **hardware-assisted approaches**, **cryptographic approaches**, and **differential privacy-based approaches**.

Hardware-assisted approaches such as **Trusted Execution Environment (TEE)** [98] and Dynamic Root of Trust Measurement (DRTN) [33] are developed from underlying architecture, which involves developing separate hardware modules or operating systems for executing ML tasks. In this way, malicious actions can be blocked to a great extent and thus systems can offer strong S&P guarantees to all model assets. Notwithstanding, the requirement of specific hardware configurations impedes their applicability on different computing devices.

Homomorphic Encryption (HE) [4] is a cryptographic technique widely adopted in data privacy-protection of ML that enables computation to be directly performed on encrypted data (i.e., ciphertext) with no need for decrypting the data, thereby making the computation's result maintain encrypted. Albeit the protection of data privacy, HE is computationally expensive, especially for decentralized systems where clients are usually edge devices (e.g., smartphones), and may significantly reduce the training efficiency of the shared model [140].

Differential Privacy (DP) [31], and similar approaches such as k-Anonymity [112] harness the addition of noise to the data or the use of generalization techniques to obfuscate certain sensitive attributes to the point where a third party cannot distinguish the individual, rendering the data unrecoverable. The most primitive and widely-known DP is ϵ-DP [32], which is formulated as

$$\Pr\left[\mathcal{A}\left(X_1\right) \in S\right] \leq \exp(\varepsilon) \cdot \Pr\left[\mathcal{A}\left(X_2\right) \in S\right] \tag{9}$$

where $\mathcal{A}$ is a randomized algorithm (e.g., adding random noise); X_1 and X_2 are two datasets that differ on a single element; ϵ is the privacy budget (level of DP). Given a specific output set S, $\mathcal{A}$ endeavors to make the distributions of output $\mathcal{A}\left(X_1\right)$ and $\mathcal{A}\left(X_2\right)$ undistinguishable. Compared with the above two branches of approaches, DP is considered a more feasible countermeasure to privacy attacks due to its ease of operation and negligible computational overhead. Nonetheless, recent research has also disclosed the limitations of DP [52]. First, adding noise naturally depreciates the utility of the data for model training [61, 153]. Furthermore, [2, 55, 96, 148] revealed the ineffectiveness of DP as a countermeasure against GAN-based attacks. Specifically, Zhang *et al.* [148] and Aïvodji [2] *et al.* ascribed the failure to that DP (canonical record-level DP) only hides the presence of a single data record in the training set — limiting the learning of individual training instances, in return, may facilitate the learning of generic features of a class and thus actually contribute to preimage inference. Hitaj *et al.* [55] argued that in FL systems, applying differential privacy (DP) to the model parameters is not effective as the noise introduced through DP is not retained once the model is well-trained. Thus, pruning or obfuscating shared parameters by DP will not contribute to defending the GAN-based attacks since the GAN-based models can retain their effectiveness so long as the target model on the client side has high accuracy. Particularly, Hitaj *et*

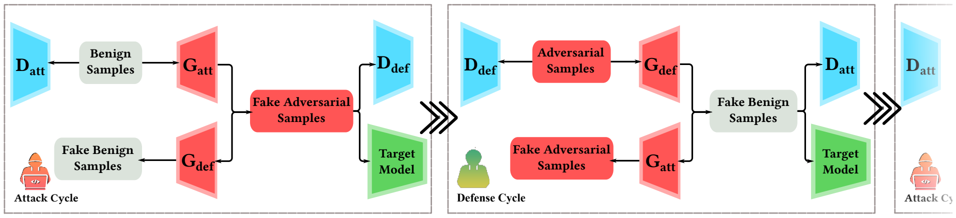


Fig. 11. CycleGAN-based attack and defense process [62]. G_{att} and G_{def} denote the generators of attacker and defender, respectively. D_{att} and D_{def} denote the generators of attacker and defender, respectively. The generator of CycleGAN is designed to translate an example from a source domain to a target domain, and the two domains here are the feature spaces of benign and adversarial examples. Denote the benign and adversarial examples as X and X_A. In the attack cycle, G_{att} takes X as the input and generates fake adversarial examples X'_A. Then, X'_A is sent to D_{def}, which is used to distinguish X'_A and X_A. The defense cycle is symmetric to the attack cycle. G_{def} takes X_A as the input and generate fake benign examples X'. Next, X'_A is sent to D_{att}, which is used to tell X' and X apart. Furthermore, a white-box attack is considered that the target model F is considered here, which takes X'_A and X' as the inputs where the measured losses are used to optimize the attack.

al. also indicated that while DP aims to prevent attackers from distinguishing whether a data record X is part of the training set, GANs have the capability to generate a synthetic example X_A that is indistinguishable from the original X, thereby circumventing this defense. As the results presented in the right part of Figure 9(a), whether with small or big privacy budgets (ϵ), GANs can generate semantic-distinguishable results, which has satisfied the demand of low-fidelity attack tasks such as class representation inference.

4.2 Pointed Countermeasures

Countermeasures can be explicitly designed for different threat models. For countering adversarial attacks, many methods were proposed to **modify training or test samples**. Revolving around the test data examples, Guo *et al.* [48] and Das *et al.* [26] applied image transformations such as bit-depth reduction and JPEG compression to test examples before fed to the classifier, which achieves a fast removal of most adversarial perturbations. However, useful information may be sacrificed in the image transformation. In a different approach that aims at enhancing classifiers' resilience against adversarial examples, Papernot *et al.* [93] employed distillation to extract additional knowledge about the training examples. This additional knowledge helped to identify perturbations that the classifier is sensitive to, which were then utilized during classifier training. By reducing sensitivity to input perturbations, the approach enabled the development of smoother classifier models. Bhagoji *et al.* [10] leveraged linear transformation such as Principal Component Analysis (PCA) to achieve the useful information identification of the training examples to improve the robustness of the trained classifier further. For countering gradient-based preimage inference attack, Fredrikson *et al.* [39] proposed to reduce the precision where confidence scores are developed (e.g., the scores generated by softmax functions). However, this countermeasure is only available in the black box setting since the model information is accessible to the attacker in the white box setting. When countering a GAN-based attack, this countermeasure would also reduce the approximation capability of the surrogate model to the target model's decision boundaries [2]. For black-box attacks where the attackers always have to perform a large number of queries to the MLaaS platforms, Shi *et al.* [106] indicated that limiting the number of queries or identifying a large number of queries as malicious could be simple yet effective countermeasures.

The adversarial learning nature of GAN enables to **counter GAN-based attacks using another GAN—"like cures like"**. Leveraging the cycle consistency notion of CycleGAN, Jiang *et al.* [62] integrated GAN-based attack and defense (both attacker and defender operate a respective GAN) in a CycleGAN by taking the output of a GAN as the input of another (see Figure 11). Following the terms in Figure 11, the objective functions of the attacker and defender's generators can be formulated as

$$\begin{aligned}\mathcal{L}_{(G_{\text{att}})}(G_{\text{att}}, D_{\text{def}}) &= \mathop{\mathbb{E}}_{X\sim p_{\text{data}}(X)}[\log D_{\text{def}}(X_A)] + \mathop{\mathbb{E}}_{Z\sim p_Z(Z)}[\log(1 - D_{\text{def}}(G_{\text{att}}(X)))],\\ \mathcal{L}_{(G_{\text{def}})}(G_{\text{def}}, D_{\text{att}}) &= \mathop{\mathbb{E}}_{X\sim p_{\text{data}}(X)}[\log D_{\text{att}}(X)] + \mathop{\mathbb{E}}_{Z\sim p_Z(Z)}[\log(1 - D_{\text{att}}(G_{\text{def}}(X_A)))].\end{aligned} \tag{10}$$

In such a way, the attack and defense capacity could promote mutually. A further insight is that GANs are born with a privacy-preserving peculiarity since only the discriminator can access real samples directly while the generator cannot during the training process of GAN. That is to say: if the victim adopts a GAN to handle his learning tasks, there is a natural cover for privacy defense [50, 102]. For readers interested in GAN-based techniques for safeguarding S&P in machine learning, a comprehensive resource to explore is the work by Cai *et al.* [17].

Furthermore, according to the countering initiative of defenders (victims), the countermeasures can be either passive or active. Passive defenses are usually performed at the victim side, aiming to minimize the attacks' influence locally. The majority of previously mentioned cryptography-based, DP-based, and data transformation approaches can be classified into this category. On the contrary, active defenses mean actively detecting the potential threats and blocking them offshore (usually implemented at the side of potentially malicious users or the communication channels connected to them)— or, more aggressively and thoroughly, striking back on the attacker to knock out their attack models.

4.2.1 Countermeasures to GAN-based Attacks in Decentralized Systems. The countermeasures to GAN-based attacks in decentralized systems can be greatly strategized by the aforementioned passive or active patterns. We take three countermeasures to the inference attack proposed in [55] to exemplify in the sequel. For passive defense, computation-efficient approaches are highly favored. Ching *et al.* [23] proposed a defense approach that utilizes model partition (see Countermeasure 2 in Figure 9(a)). Instead of training the entire model at the client end, the proposed approach conceals the model from users and edge servers. This enables the concealed parts of the model to be trained with the assistance of these entities. By employing this approach, the attacker is unable to steal integral information from the FL system to generate samples resembling the victim's. Particularly, to mitigate the risk of data leakage, it is essential to perform the computations of the first and last layers of the model on the client end. In [147], the authors introduced another strategy that inherits the aforementioned "like cures like" idea to defend against GAN-based inference attacks (see Countermeasure 3 in Figure 9(a)). As the threat model intends to learn the distribution of the victim's data using GAN, this study introduces a GAN at the victim end that manipulates the victim's training data by generating fake data before inputting them into the shared model for training. The purpose is to prevent the attacker from learning the real distribution. By employing this approach, the attacker is restricted to learning the distribution of the manipulated data, rather than the actual distribution of the victim's data. Particularly, to ensure that the manipulated data obfuscates the attacker, making the recovered examples by the attacker's GAN indistinguishable from human beings, an unsupervised learning pattern is employed. This pattern formulates a corresponding objective function as $\max_Z \mathcal{L}_{(\text{obf})} = \log Var(G(Z))$, where the variance of $G(Z)$ is expected to be maximized to distort the generated samples. Moreover, addressing the concern that the manipulated data could potentially decrease the accuracy of the shared model, the objective function of the generator of the victim's GAN is modified to minimize the feature distance between the generated samples and the original samples, which is formulated as $\min_Z \mathcal{L}_{(G)} = \|G(Z) - X\|_2^2$.

In contrast to passive defense, active defense in decentralized ML systems offers timely and accurate warnings prior to attacks. It usually aims to build a resilient defense system in real-time to mitigate, transfer, and reduce the risks faced by the clients. Active defense can substantially decrease the overall computational overheads of defense systems [28]. Xiong *et al.* [131] presented an approach that actively detects GAN-based attacks during the initial training phase of the victim model (see Countermeasure 1 in Figure 9(a)). The threat model assumed the same as the one in [55] as well. However, they assumed that the central server knows information about the distribution of data classes among clients, without having access to the specific data itself — the practicality of this assumption is disputable. This approach leverages the difference between victims' and survivors' gradient updates at each global epoch to detect the anomaly. Specifically, this approach commences by extracting the feature from gradient update vectors using multiple auto-encoders [54], and then uses an unsupervised clustering approach [35] to identify the class that a client should not have based on the central server's knowledge. This makes it more difficult for attackers to evade detection during the initial stages of the attack. Notably, this early detection prevents the further spreading of the attack effect, thus enabling the protection of more training data.

5 FUTURE RESEARCH OUTLOOK

Although it has been many years since GAN first appeared, GAN-related studies never fade out from research hotspots. In particular, with the evolution of GAN-based techniques, their threats regarding the security and privacy of machine learning systems are expected to be further investigated. This section provides an outlook for promising future research directions to fill the existing research gap further.

5.1 Further Advances on GANs Models

In addition to strategic considerations such as when and where to use GANs, the effectiveness of GAN-based attacks and defenses relies heavily on the capabilities of the GAN models themselves. In the context of attack scenarios, the operational space and effectiveness of GANs can be significantly restricted compared to normal usage. This limitation arises from the need for stealthiness and the potential countermeasures they may face.

Model collapse and training difficulty are two main issues of existing GANs. Although GANs have shown their potential for generating considerably natural examples, they commonly suffer the curse of mode collapse, meaning they can capture only a limited variety of modes in the data. GANs, in particular, fail to learn some of the modes when trained with multi-modal distribution data samples. Consequently, GANs could only reach sub-optimal solutions in most cases rather than the true equilibrium. The mode collapse renders the generated samples often lacking diversity. As illustrated in Section 2.1.2, many variants of GANs have been devised with more stable architectures, improved learning objectives, and so on, to solve these problems. It has been studied that loss function-enhanced GANs variants often show more improvement in training than architectural-enhanced GANs; however, they still cannot prompt mode diversity in the generated samples. Although the proper design of architecture and loss function can be orchestrated in a GAN to address the problems, their effectiveness is inherently limited. To tackle the training difficulty of GANs, researchers also experiment with different tricks such as training strategies bettering and hyperparameters tuning, etc. Unfortunately, some good results of these solutions conversely sacrifice the quality or diversity of the generated samples. On the other hand, some GANs variants, especially the loss function-enhanced GANs, ask for stringent training requirements, which cannot adopt the tricks.

Comprehensively speaking, the trade-off relationships between the existing solutions are pronounced, and no such a well-rounded solution exists to different problems. Technical breakthroughs of existing GANs are more likely to be achieved by fundamental and theoretical analysis, which is extremely desired in the community.

5.2 GANs Threats to More Applications

5.2.1 Regression Learning. GANs have been mainly studied and applied to the computer vision field, where classification tasks are the preoccupation. Meanwhile, researchers have started to generalize GANs to regression tasks. Regression tasks describe several predictor variables exploited to predict one or multiple numerical response variables with, usually, sequential data [60]. Compared to classification tasks with a large structured output space, the output space of regression tasks is relatively smaller. It has been demonstrated that semi-supervised GAN can be used to handle different types of regressions models (e.g., deep neural networks, XGboost, and Gaussian process) by the involvement of auxiliary or constraint information [88].

From the perspective of the application, regression learning is widely applied to tasks that may involve a great deal of private sequential information: loan or insurance risk estimation, personalized medicine dispensation, financial market analysis, etc. For privacy attacks on these tasks, GANs could be used to infer a sequence of data records from the regression models such that compromise the data holders' privacy — or at a coarse granularity, the sequential pattern, which can also be used for trend prediction. Little effort has been made to investigate the GAN-based attack/defense in terms of regression tasks, which could be a worthy future research direction.

5.2.2 Graph-structured Data and Graph Learning. The capability of handling non-Euclidean data such as social networks point cloud data make geometric deep learning, especially graph learning, attract broad research attention in recent years. Manifold GAN-based graph generation approaches have emerged in this trend, which has been applied to but is not limited to, node feature generation, link generation, and complete topology (graph structure) generation. Related attack methods have been proposed using GAN to generate additional nodes or edges that modify the original graph structure or perturb their features or weights, demonstrating considerable destructiveness. The challenges lie in how to make attacks more efficient as graph-structured data usually renders related learning involving considerable modeling depth and breadth, which is computationally costly.

On the other side, the connectivity existing in different graph entities renders its privacy protection a problematic task. For example, it has been shown that edges with more significant influence are more likely to be recovered. Generic approaches such as differential privacy can hardly defend without dropping the task accuracy [149]. One possible way is extracting high-level, structure-free graph representation (i.e., does not contain explicit graph structure information). However, the challenge lies in finding representations that can effectively deceive the powerful learning capabilities of GANs, while also preserving the utility for downstream tasks. These issues are deserving of further investigation.

5.2.3 More Complex ML Systems. As introduced in Section 3.3, there have been several studies investigating GAN-based attack/defense in federated learning systems, which shows the efficacy of GANs to attack decentralized ML systems — especially the capacity of reconstructing information from the untouchable data source (i.e., other clients in a FL system). However, existing studies have two main limitations. On the one hand, most of them focus on simple-framework systems. Taking the example of a federated learning (FL) system, although it is often perceived as a decentralized system, the utilization of centralized aggregation methods such as FedAvg implies that the system retains some centralized aspects. Typically, the system topology follows a star-like or tree-like structure when multiple hierarchies are involved. Recent research shifts their attention to fully-decentralized FL (conceptualized as swarm learning [126]) in which a central server does not exist and, instead, participants communicate in a mesh topology. One concern is as such systems involve more dense connections, and thus more frequent communication among participants possibly results in privacy leakage during communication can be caught by the attacker. If this is the case, for GAN-based attacks, it is of necessity to explore what would be good entry points to maximize the attack effect.

On the other hand, most existing GAN-based studies only evaluate their proposed methods when there are no or very simple defense measures. Nevertheless, real-world industrial ML systems are equipped with comprehensive protective mechanisms. For instance, cryptography-based defense methods, often overlooked in academic research due to their computational cost, are extensively utilized in industry. How to design GAN-based attack methods against ML systems with complex structure, functionality, and more powerful defense would deserve researchers' attention for improving the practicability of current research.

5.3 Interpretability Studies on GANs

While GANs have been widely applied to various attack approaches, there is still a lack of comprehensive theoretical studies in this domain. Particularly, the generator of GANs possesses the nature of deep learning, which is in essence a black-box function. To understand: why the method can work, what is the upper bound of the method, and what data features would be utilized for attacks (i.e., vulnerability), it would be better to conduct interpretability research.

One possible solution is using information theory. Recent years have witnessed many studies that used information bottleneck (IB) [113] to explore the interpretability of ML. IB mainly provides a principle for learning useful representation Z of original data X, which encourages Z to be maximally informative about the target Y to develop accurate prediction; meanwhile, it discourages Z from mingling with additional information from X that is irrelevant for predicting Y. The above-mentioned InfoGAN was a successful attempt to apply IB to latent representation *disentanglement*. For GAN-based attacks, it is also interesting to know if the IB theory could claim the computational benefit of the attack model with different GAN architectures, etc. Another possible solution is leveraging quantitative analysis, which was previously widely adopted in the financial field. Recent research attempted to use the Shapley value as a metric to quantify the value of each training example of a ML model to the model performance. Revealingly, Ghorbani *et al.* [44] found that, in their training data poisoning (noisy labeling) case study, low Shapley value data examples effectively capture the poisoned points. Related research is still at the early stage; investigating the vulnerability of data to GAN-based inference or adversarial attack by quantitative analysis can be a future research direction to explore. On the flip side, by making us understand the key factors in models and data in the S&P context, interpretability research can help us develop more effective defense mechanisms.

6 SUMMARY

Nowadays, GANs have been regarded as powerful tools with the potential to pose threats to security and privacy due to their remarkable generation capabilities. This survey aims to systematically analyze the security and privacy threats of GANs to machine learning. To ensure non-expert readers can grasp the concepts, we begin by providing a background on GANs, including an overview of the technical principles of primitive GANs and some notable variants. Subsequently, we provide a novel taxonomy that categorizes the threats posed by GANs to ML systems and discuss existing related works in each category. Considering both centralized and decentralized ML systems, we then explore the specific properties of GANs that make them advantageous for attacks and the underlying strategies employed in existing attack methods. In addition, from the opposite side of things, we investigate the countermeasures to GAN-based attacks, including indicating existing methods' limitations. Lastly, drawing from the insights gained and the current research trends in the community, we propose promising directions for future exploration by researchers.

ACKNOWLEDGMENTS

This research was supported by the Australian Research Council (ARC) DP200101374.

REFERENCES

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. 2018. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 1–7.

[2] Ulrich Aïvodji, Sébastien Gambs, and Timon Ther. 2019. Gamin: An adversarial approach to black-box model inversion. *arXiv preprint arXiv:1909.11835* (2019).

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[4] Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gjøsteen, Angela Jäschke, Christian A Reuter, and Martin Strand. 2015. A Guide to Fully Homomorphic Encryption. *IACR Cryptol. ePrint Arch.* 2015 (2015), 1192.

[5] Yang Bai, Degang Chen, Ting Chen, and Mingyu Fan. 2021. GANMIA: GAN-based Black-box Membership Inference Attack. In *ICC 2021-IEEE International Conference on Communications*. IEEE, 1–6.

[6] Shumeet Baluja and Ian Fischer. 2017. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387* (2017).

[7] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. 2006. Can machine learning be secure?. In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. 16–25.

[8] Samyadeep Basu, Rauf Izmailov, and Chris Mesterharm. 2019. Membership model inversion attacks for deep networks. *arXiv preprint arXiv:1910.04257* (2019).

[9] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*. PMLR, 634–643.

[10] Arjun Nitin Bhagoji, Daniel Cullina, Chawin Sitawarin, and Prateek Mittal. 2018. Enhancing robustness of machine learning systems via data transformations. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 1–5.

[11] Siddhant Bhambri, Sumanyu Muku, Avinash Tulasi, and Arun Balaji Buduru. 2019. A survey of black-box adversarial attacks on computer vision models. *arXiv preprint arXiv:1912.01667* (2019).

[12] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389* (2012).

[13] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. 2021. Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. https://doi.org/10.1109/TPAMI.2021.3116668

[14] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations*.

[15] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665* (2017).

[16] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. 2018. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*.

[17] Zhipeng Cai, Zuobin Xiong, Honghui Xu, Peng Wang, Wei Li, and Yi Pan. 2022. Generative Adversarial Networks: A Survey Toward Private and Secure Applications. *ACM Computing Surveys (CSUR)* 54, 6 (2022), 38.

[18] Yang-Jie Cao, Li-Li Jia, Yong-Xia Chen, Nan Lin, Cong Yang, Bo Zhang, Zhi Liu, Xue-Xiang Li, and Hong-Hua Dai. 2019. Recent Advances of Generative Adversarial Networks in Computer Vision. *IEEE Access* 7 (2019), 14985–15006. https://doi.org/10.1109/ACCESS.2018.2886814

[19] Qi Chang, Hui Qu, Yikai Zhang, Mert Sabuncu, Chao Chen, Tong Zhang, and Dimitris N Metaxas. 2020. Synthetic learning: Learn from distributed asynchronized discriminator gan without sharing medical image data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13856–13866.

[20] Jinyin Chen, Haibin Zheng, Hui Xiong, Shijing Shen, and Mengmeng Su. 2020. MAG-GAN: Massive attack generator via GAN. *Information Sciences* 536 (2020), 67–90.

[21] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2180–2188.

[22] Zhenzhu Chen, Anmin Fu, Yinghui Zhang, Zhe Liu, Fanjian Zeng, and Robert H Deng. 2020. Secure collaborative deep learning against GAN attacks in the Internet of Things. *IEEE Internet of Things Journal* 8, 7 (2020), 5839–5849.

[23] Cheng-Wei Ching, Tzu-Cheng Lin, Kung-Hao Chang, Chih-Chiung Yao, and Jian-Jhih Kuo. 2020. Model Partition Defense against GAN Attacks on Collaborative Learning via Mobile Edge Computing. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 1–6.

[24] Hyungjoo Cho, Sungbin Lim, Gunho Choi, and Hyunseok Min. 2017. Neural stain-style transfer learning using GAN for histopathological images. *arXiv preprint arXiv:1710.08543* (2017).

[25] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. 2004. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 99–108.

[26] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. 2018. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 196–204.

[27] Gustavo H De Rosa and João P Papa. 2021. A Survey on Text Generation using Generative Adversarial Networks. *Pattern Recognition* (2021), 108098.

[28] Dorothy E Denning. 2014. Framework and principles for active cyber defense. *Computers & Security* 40 (2014), 108–113.

[29] Chris Donahue, Bo Li, and Rohit Prabhavalkar. 2018. Exploring speech enhancement with generative adversarial networks for robust speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5024–5028.

[30] Indira Kalyan Dutta, Bhaskar Ghosh, Albert Carlson, Michael Totaro, and Magdy Bayoumi. 2020. Generative Adversarial Networks in Security: A Survey. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 0399–0405.

[31] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*. Springer, 1–19.

[32] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 265–284.

[33] Karim Eldefrawy, Gene Tsudik, Aurélien Francillon, and Daniele Perito. 2012. SMART: Secure and Minimal Architecture for (Establishing Dynamic) Root of Trust.. In *Ndss*, Vol. 12. 1–15.

[34] Gamaleldin Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. 2018. Adversarial examples that fool both computer vision and time-limited humans. *Advances in neural information processing systems* 31 (2018).

[35] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *kdd*, Vol. 96. 226–231.

[36] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1625–1634.

[37] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local Model Poisoning Attacks to {Byzantine-Robust} Federated Learning. In *29th USENIX Security Symposium (USENIX Security 20)*. 1605–1622.

[38] Cheng Feng, Tingting Li, Zhanxing Zhu, and Deeph Chana. 2017. A deep learning-based framework for conducting stealthy attacks in industrial control systems. *arXiv preprint arXiv:1709.06397* (2017).

[39] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1322–1333.

[40] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in Pharmacogenetics: An {End-to-End} Case Study of Personalized Warfarin Dosing. In *23rd USENIX Security Symposium (USENIX Security 14)*. 17–32.

[41] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. 2018. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866* (2018).

[42] Nan Gao, Hao Xue, Wei Shao, Sichen Zhao, Kyle Kai Qin, Arian Prabowo, Mohammad Saiedur Rahaman, and Flora D Salim. 2020. Generative adversarial networks for spatio-temporal data: A survey. *arXiv preprint arXiv:2008.08903* (2020).

[43] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. 2019. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1155–1164.

[44] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*. PMLR, 2242–2251.

[45] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[46] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[47] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. 2021. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *IEEE Transactions on Knowledge and Data Engineering* (2021), 1–1. https://doi.org/10.1109/TKDE.2021.3130191

[48] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. 2018. Countering Adversarial Images using Input Transformations. In *International Conference on Learning Representations*.

[49] Noushin Hajarolasvadi, Miguel Arjona Ramírez, Wesley Beccaro, and Hasan Demirel. 2020. Generative Adversarial Networks in Human Emotion Synthesis: A Review. *IEEE Access* 8 (2020), 218499–218529. https://doi.org/10.1109/ACCESS.2020.3042328

[50] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola. 2019. Md-gan: Multi-discriminator generative adversarial networks for distributed datasets. In *2019 IEEE international parallel and distributed processing symposium (IPDPS)*. IEEE, 866–877.

[51] Atiye Sadat Hashemi and Saeed Mozaffari. 2019. Secure deep neural networks using adversarial image generation and training with Noise-GAN. *Computers & Security* 86 (2019), 372–387.

[52] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. 2019. Differential privacy techniques for cyber physical systems: a survey. *IEEE Communications Surveys & Tutorials* 22, 1 (2019), 746–789.

[53] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18, 7 (2006), 1527–1554.

[54] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[55] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 603–618.

[56] Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, and Fernando Perez-Cruz. 2019. Passgan: A deep learning approach for password guessing. In *International Conference on Applied Cryptography and Network Security*. Springer, 217–237.

[57] Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. 2019. How generative adversarial networks and their variants work: An overview. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–43.

[58] Weiwei Hu and Ying Tan. 2017. Generating adversarial malware examples for black-box attacks based on GAN. *arXiv preprint arXiv:1702.05983* (2017).

[59] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

[60] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 19–35.

[61] Bargav Jayaraman and David Evans. 2019. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*. 1895–1912.

[62] Lingyun Jiang, Kai Qiao, Ruoxi Qin, Linyuan Wang, Wanting Yu, Jian Chen, Haibing Bu, and Bin Yan. 2020. Cycle-consistent adversarial GAN: the integration of adversarial attack and defense. *Security and Communication Networks* 2020 (2020).

[63] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4401–4410.

[64] Keshav Kasichainula, Hadi Mansourifar, and Weidong Shi. 2021. Poisoning Attacks via Generative Adversarial Text to Image Synthesis. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 158–165.

[65] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.

[66] Haodong Li, Bin Li, Shunquan Tan, and Jiwu Huang. 2020. Identification of deep network generated images using disparities in color components. *Signal Processing* 174 (2020), 107616.

[67] Heng Li, ShiYao Zhou, Wei Yuan, Jiahuan Li, and Henry Leung. 2019. Adversarial-example attacks toward android malware detection system. *IEEE Systems Journal* 14, 1 (2019), 653–656.

[68] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220* (2016).

[69] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial Learning for Neural Dialogue Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2157–2169.

[70] Yanchun Li, Qiuzhen Wang, Jie Zhang, Lingzhi Hu, and Wanli Ouyang. 2021. The theoretical research of generative adversarial networks: an overview. *Neurocomputing* 435 (2021), 26–41.

[71] Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 3158–3168.

[72] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. 2019. Perceptual-sensitive gan for generating adversarial patches. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 1028–1035.

[73] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770* (2016).

[74] Mohamed Loey, Florentin Smarandache, and Nour Eldeen M Khalifa. 2020. Within the lack of chest COVID-19 X-ray dataset: a novel detection model based on GAN and deep transfer learning. *Symmetry* 12, 4 (2020), 651.

[75] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133* (2020).

[76] Neal Mangaokar, Jiameng Pu, Parantapa Bhattacharya, Chandan K Reddy, and Bimal Viswanath. 2020. Jekyll: Attacking Medical Image Diagnostics using Deep Generative Models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 139–157.

[77] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2794–2802.

[78] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[79] Shike Mei and Xiaojin Zhu. 2015. Using machine teaching to identify optimal training-set attacks on machine learners. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[80] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 691–706.

[81] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[82] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2574–2582.

[83] Luis Muñoz-González, Bjarne Pfitzner, Matteo Russo, Javier Carnerero-Cano, and Emil C Lupu. 2019. Poisoning attacks with generative adversarial nets. *arXiv preprint arXiv:1906.07773* (2019).

[84] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 739–753.

[85] Hojjat Navidan, Parisa Fard Moshiri, Mohammad Nabati, Reza Shahbazian, Seyed Ali Ghorashi, Vahid Shah-Mansouri, and David Windridge. 2021. Generative Adversarial Networks (GANs) in networking: A comprehensive survey & evaluation. *Computer Networks* (2021), 108149.

[86] Augustus Odena. 2016. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583* (2016).

[87] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*. PMLR, 2642–2651.

[88] Greg Olmschenk, Zhigang Zhu, and Hao Tang. 2019. Generalizing semi-supervised generative adversarial networks to regression using feature contrasting. *Computer Vision and Image Understanding* 186 (2019), 1–12.

[89] Salima Omar, Asri Ngadi, and Hamid H Jebur. 2013. Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications* 79, 2 (2013).

[90] Arghya Pal and Yogesh Rathi. 2021. A review of deep learning methods for MRI reconstruction. *arXiv preprint arXiv:2109.08618* (2021).

[91] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. 2019. Recent progress on generative adversarial networks (GANs): A survey. *IEEE Access* 7 (2019), 36322–36333.

[92] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. 2016. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814* (2016).

[93] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*. IEEE, 582–597.

[94] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2016. Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples. *CoRR* abs/1602.02697 (2016).

[95] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[96] Hanchi Ren, Jingjing Deng, and Xianghua Xie. 2021. GRNN: Generative Regression Neural Network–A Data Leakage Attack for Federated Learning. *arXiv preprint arXiv:2105.00529* (2021).

[97] Maria Rigaki and Sebastian Garcia. 2018. Bringing a gan to a knife-fight: Adapting malware communication to avoid detection. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 70–75.

[98] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. 2015. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1. IEEE, 57–64.

[99] Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Deep boltzmann machines. In *Artificial intelligence and statistics*. PMLR, 448–455.

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

[101] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. 2017. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517* (2017).

[102] Gokula Krishnan Santhanam and Paulina Grnarova. 2018. Defending against adversarial attacks by leveraging an entire GAN. *arXiv preprint arXiv:1805.10652* (2018).

[103] Divya Saxena and Jiannong Cao. 2021. Generative Adversarial Networks (GANs) Challenges, Solutions, and Future Directions. *ACM Computing Surveys (CSUR)* 54, 3 (2021), 1–42.

[104] Ulrich Scherhag, Christian Rathgeb, Johannes Merkle, Ralph Breithaupt, and Christoph Busch. 2019. Face recognition systems under morphing attacks: A survey. *IEEE Access* 7 (2019), 23012–23026.

[105] Gil Shamai, Ron Slossberg, and Ron Kimmel. 2019. Synthesizing facial photometries and corresponding geometries using generative adversarial networks. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 3s (2019), 1–24.

[106] Yi Shi, Yalin E Sagduyu, Kemal Davaslioglu, and Jason H Li. 2018. Generative adversarial networks for black-box API attacks with limited training data. In *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 453–458.

[107] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[108] Mengkai Song, Zhibo Wang, Zhifei Zhang, Yang Song, Qian Wang, Ju Ren, and Hairong Qi. 2020. Analyzing user-level privacy attack against federated learning. *IEEE Journal on Selected Areas in Communications* 38, 10 (2020), 2430–2444.

[109] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. 2018. Constructing unrestricted adversarial examples with generative models. *Advances in Neural Information Processing Systems* 31 (2018).

[110] Anuroop Sriram, Heewoo Jun, Yashesh Gaur, and Sanjeev Satheesh. 2018. Robust speech recognition using generative adversarial networks. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5639–5643.

[111] Yuwei Sun, Ng Chong, and Hideya Ochiai. 2021. Information Stealing in Federated Learning Systems Based on Generative Adversarial Networks. *arXiv preprint arXiv:2108.00701* (2021).

[112] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.

[113] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[114] Mukhiddin Toshpulatov, Wookey Lee, and Suan Lee. 2021. Generative adversarial networks and their application to 3D face generation: A survey. *Image and Vision Computing* (2021), 104119.

[115] Maximilian E Tschuchnig, Gertie J Oostingh, and Michael Gadermayr. 2020. Generative adversarial networks in digital pathology: a survey on trends and future potential. *Patterns* 1, 6 (2020), 100089.

[116] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *SSW* 125 (2016), 2.

[117] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10, 3152676 (2017), 10–5555.

[118] Abdul Waheed, Muskan Goyal, Deepak Gupta, Ashish Khanna, Fadi Al-Turjman, and Plácido Rogerio Pinheiro. 2020. Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection. *Ieee Access* 8 (2020), 91916–91923.

[119] Aamir Wali, Zareen Alamgir, Saira Karim, Ather Fawaz, Mubariz Barkat Ali, Muhammad Adan, and Malik Mujtaba. 2022. Generative adversarial networks for speech processing: A review. *Computer Speech & Language* 72 (2022), 101308.

[120] Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. 2018. Rethinking feature distribution for loss functions in image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 9117–9126.

[121] Kunfeng Wang, Chao Gou, Yanjie Duan, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. 2017. Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica* 4, 4 (2017), 588–598. https://doi.org/10.1109/JAS.2017.7510583

[122] Lin Wang and Kuk-Jin Yoon. 2021. Psat-gan: Efficient adversarial attacks against holistic scene understanding. *IEEE Transactions on Image Processing* 30 (2021), 7541–7553.

[123] Xiaosen Wang, Kun He, and John E Hopcroft. 2019. AT-GAN: A generative attack model for adversarial transferring on generative adversarial nets. *arXiv preprint arXiv:1904.07793* 3, 4 (2019).

[124] Xianmin Wang, Jing Li, Xiaohui Kuang, Yu-an Tan, and Jin Li. 2019. The security of machine learning in an adversarial setting: A survey. *J. Parallel and Distrib. Comput.* 130 (2019), 12–23.

[125] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2512–2520.

[126] Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N Ahmad Aziz, et al. 2021. Swarm learning for decentralized and confidential clinical machine learning. *Nature* 594, 7862 (2021), 265–270.

[127] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. 2018. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 79–88.

[128] Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. 2019. Transferable Adversarial Attacks for Image and Video Object Detection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 954–960. https://doi.org/10.24963/ijcai.2019/134

[129] Mika Westerlund. 2019. The emergence of deepfake technology: A review. *Technology Innovation Management Review* 9, 11 (2019).

[130] Chaowei Xiao, Bo Li, Jun Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. In *27th International Joint Conference on Artificial Intelligence, IJCAI 2018*. International Joint Conferences on Artificial Intelligence, 3905–3911.

[131] Yayuan Xiong, Fengyuan Xu, and Sheng Zhong. 2020. Detecting GAN-based Privacy Attack in Distributed Learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[132] Mingxue Xu and Xiangyang Li. 2020. Subject property inference attack in collaborative learning. In *2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Vol. 1. IEEE, 227–231.

[133] Xiaodan Yan, Baojiang Cui, Yang Xu, Peilin Shi, and Ziqi Wang. 2019. A method of information protection for collaborative deep learning under GAN model attack. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18, 3 (2019), 871–881.

[134] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. 2017. Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340* (2017).

[135] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[136] Xin Yi, Ekta Walia, and Paul Babyn. 2019. Generative adversarial network in medical imaging: A review. *Medical image analysis* 58 (2019), 101552.

[137] Chika Yinka-Banjo and Ogban-Asuquo Ugot. 2020. A review of generative adversarial networks and its application in cybersecurity. *Artificial Intelligence Review* 53, 3 (2020), 1721–1736.

[138] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[139] Junkun Yuan, Shaofang Zhou, Lanfen Lin, Feng Wang, and Jia Cui. 2020. Black-box adversarial attacks against deep learning based malware binaries detection with GAN. In *ECAI 2020*. IOS Press, 2536–2542.

[140] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. {BatchCrypt}: Efficient Homomorphic Encryption for {Cross-Silo} Federated Learning. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. 493–506.

[141] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 5907–5915.

[142] Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. 2020. PoisonGAN: Generative Poisoning Attacks Against Federated Learning in Edge Computing Systems. *IEEE Internet of Things Journal* 8, 5 (2020), 3310–3322.

[143] Jiale Zhang, Junjun Chen, Di Wu, Bing Chen, and Shui Yu. 2019. Poisoning attack in federated learning using generative adversarial nets. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 374–380.

[144] Jingwen Zhang, Jiale Zhang, Junjun Chen, and Shui Yu. 2020. Gan enhanced membership inference: A passive local attack in federated learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[145] Lvmin Zhang, Yi Ji, Xin Lin, and Chunping Liu. 2017. Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In *2017 4th IAPR Asian conference on pattern recognition (ACPR)*. IEEE, 506–511.

[146] Rui Zhang, Hui Xia, Chunqiang Hu, Cheng Zhang, Chao Liu, and Fu Xiao. 2022. Generating Adversarial Examples with Shadow Model. *IEEE Transactions on Industrial Informatics* (2022).

[147] Xianglong Zhang and Xinjian Luo. 2020. Exploiting defenses against GAN-based feature inference attacks in federated learning. *arXiv preprint arXiv:2004.12571* (2020).

[148] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 253–261.

[149] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, and Enhong Chen. 2021. GraphMI: Extracting Private Graph Data from Graph Neural Networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3749–3755. https://doi.org/10.24963/ijcai.2021/516

[150] Guoping Zhao, Mingyu Zhang, Jiajun Liu, Yaxian Li, and Ji-Rong Wen. 2020. AP-GAN: Adversarial patch attack on content-based image retrieval systems. *GeoInformatica* (2020), 1–31.

[151] Guoping Zhao, Mingyu Zhang, Jiajun Liu, and Ji-Rong Wen. 2019. Unsupervised adversarial attacks on deep feature-based retrieval with GAN. *arXiv preprint arXiv:1907.05793* (2019).

[152] Junbo Zhao, Michael Mathieu, and Yann LeCun. 2017. Energy-based generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017*.

[153] Lingchen Zhao, Qian Wang, Qin Zou, Yan Zhang, and Yanjiao Chen. 2019. Privacy-preserving collaborative deep learning with unreliable participants. *IEEE Transactions on Information Forensics and Security* 15 (2019), 1486–1500.

[154] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342* (2017).

[155] Hang Zhou, Dongdong Chen, Jing Liao, Kejiang Chen, Xiaoyi Dong, Kunlin Liu, Weiming Zhang, Gang Hua, and Nenghai Yu. 2020. Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10356–10365.

[156] Rui Zhou, Cong Jiang, and Qingyang Xu. 2021. A survey on generative adversarial network-based text-to-image synthesis. *Neurocomputing* 451 (2021), 316–336.

[157] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.

[158] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in Neural Information Processing Systems* 32 (2019).

[159] Zheng-An Zhu, Yun-Zhong Lu, and Chen-Kuo Chiang. 2019. Generating adversarial examples by makeup attacks on face recognition. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2516–2520.