Resilience of GANs against Adversarial Attacks

Kyrylo Rudavskyy and Ali Miri
Department of Computer Science, Ryerson University, Toronto, Canada

Keywords: Machine Learning, Generative Adversarial Network, Adversarial Attack, Security.

Abstract: The goal of this paper is to explore the resilience of Generative Adversarial Networks(GANs) against adversarial attacks. Specifically, we evaluated the threat potential of an adversarial attack against the discriminator part of the system. Such an attack aims to distort the output by injecting maliciously modified input during training. The attack was empirically evaluated against four types of GANs, injections of 10% and 20% malicious data, and two datasets. The targets were CGAN, ACGAN, WGAN, and WGAN-GP. The datasets were MNIST and F-MNIST. The attack was created by improving an existing attack on GANs. The lower bound for the injection size turned out to be 10% for the improvement and 10-20% for the baseline attack. It was shown that the attack on WGAN-GP can overcome a filtering defence for F-MNIST.

1 INTRODUCTION

A *Generative Adversarial Network (GAN)* was first proposed by Goodfellow *et al.* in 2014. It is a machine learning technique whose goal is to learn the distribution of a set of data (Lucic et al., 2018). This is accomplished similar to a two-player game, where the players are neural networks. One player - called the *generator* - transforms a sample from the normal distribution into a sample that resembles real data. The other player - called the *discriminator* - tries to assess if a sample is real or fake based on its knowledge of the real data. After a sufficient number of iterations, the generator will produce samples that are hard to distinguish from the real ones. Thus, it will learn to transform a normal distribution into the data distribution.

GANs have a variety of applications. A recent survey paper covers the more popular applications of GANs (Alqahtani et al., 2021). The authors describe a variety of use-cases in the audiovisual and medical domains. Moreover, GANs can generate synthetic data to compensate insufficient data (Wang et al., 2019), imbalanced data (Engelmann and Lessmann, 2021), or provide privacy (Choi et al., 2017). Finally, GANs are also used for malicious purposes such as deepfakes. These are videos or images where a person can appear to act or say almost anything (Yadav and Salmani, 2019).

1.1 Problem Statement

It is clear from the above that GANs can be used in mission-critical systems. Alternatively, one may wish to disrupt a GAN in case of potential malicious usage of the latter. In both cases, the security of these algorithms is of interest. Thus, the overarching goal of this paper is to explore the resilience of Generative Adversarial Networks against adversarial attacks.

It is a well-known fact that neural networks are susceptible to attacks (Miller et al., 2020; Kaviani and Sohn, 2021). Some researchers even think that circumstances enabling attacks on neural networks are an innate characteristic of the deep learning process (Ilyas et al., 2019). Since neural networks play a key role in most GANs, it is reasonable to assume that security problems from neural networks will affect them.

Particularly worrisome are *adversarial attacks*. These attacks were created to manipulate the output of a neural network by modifying the input (Miller et al., 2020). Since GANs are a cleverly designed system of two or more neural networks, an adversarial attack works by targeting one of these networks. In fact, designing GANs that are resilient to such attacks is an active area of research (Liu and Hsieh, 2019; Bashkirova et al., 2019; Xu et al., 2019; Zhou and Krähenbühl, 2019). A possible impact of such an attack is low-quality output, or synthetic data, that does not resemble the original.

What is less understood is the feasibility of such attacks in the real world. The reason is that scholars often focus on defending either a single component of

Rudavskyy, K. and Miri, A.
Resilience of GANs against Adversarial Attacks.
DOI: 10.5220/0011307200003283
In *Proceedings of the 19th International Conference on Security and Cryptography (SECRYPT 2022)*, pages 390-397
ISBN: 978-989-758-590-6; ISSN: 2184-7711

a GAN or a narrow case, such as defending a cycle-consistent GAN against itself. This is the gap that our research is trying to close. In short, we want to investigate the impact of an adversarial attack on synthetic data, which was not comprehensively examined before. The attack will be referred to as *Monkey-Wrench Attack (MWA)*.

1.2 Findings

Our paper will show that adversarial attacks on GANs are challenging to execute in practice. This is likely because GANs have a degree of innate resilience to noisy data (Bora et al., 2018; Thekumparampil et al., 2018). Of the four GAN variants, only one proved to be vulnerable across datasets. In addition, the attack required modifying 10-20% of input data.

However, in conditions that more closely resemble a real-world application of GANs, the attack proved to be more successful. The GAN version that was more vulnerable is one of the most advanced architectures proposed to date (Lucic et al., 2018). Moreover, the modifications of input data evaded detection on the more sophisticated dataset. Advanced GAN architectures and sophisticated datasets are more likely to be employed in applications.

In the setting mentioned above, the attack was also improved. An adversarial crafting process was devised that improved attack performance over the baseline approach. Moreover, a lower bound on the injection size was established for both cases. Finally, our adversarial samples were harder for a human operator to identify as malicious.

The stealth of adversarial samples is significant because they produce unexpected results and might be spotted during production. However, if the operator fails to identify the malicious intent behind these irregularities, the operator may fail to attribute a system malfunction to them. This will lead to the operator forgoing additional security measures and putting the data into production.

The above high-level conclusions rest on a series of contingent results. Most of the latter will be derived empirically and substantiated in Section 4. Below is a list of findings that, to the best of our knowledge, were not produced before:

F.1. The improved attack is likely to outperform the existing attacks when the loss function is complex and the dataset is sophisticated. Performance is measured by observing higher output distortion for equal size injections.

F.2. If the above conditions are met, the improved attack will likely overcome the countermeasure applied herein with greater success than the existing attack. Success is defined the same way as above.

F.3. Our version of the attack required an injection of at least 10%, while the baseline had a lower bound in the 10-20% range.

F.4. It is likely possible to visually conceal the improved attack from a human operator.

2 RELATED WORKS

This section will commence with descriptions of GANs used in this paper starting with the original one (Goodfellow et al., 2014). Although the latter was not used, it will be provided as a base model of the technology. Let P_X represent an unknown data distribution, then the discriminator, D, is trained to assign a probability of sample belonging to P_X, while the generator, G, is trained to create samples that resemble the original data. This process estimates data distribution, P_X, by implicitly creating a generator distribution, Q_X. Finally, a GAN builds G by learning to transform samples from the normal distribution, P_Z. The process is expressed by a two-player minimax game with the following objective function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_X}[\log D(x)] + \mathbb{E}_{z\sim P_Z}[\log(1-D(G(z)))] \quad (1)$$

Solving it minimizes Jensen–Shannon Divergence (JSD) between the distributions P_X and Q_X (Gong et al., 2019).

A straightforward extension of a GAN - called a *Conditional GAN (CGAN)* - was achieved by conditioning it on a class label (Mirza and Osindero, 2014). It was further extended with an *Auxiliary Conditional GAN (AC-GAN)* by adding a third component that determines a sample's probability of belonging to a certain class (Odena et al., 2017).

A breakthrough in GANs came with the introduction of a *Wasserstein GAN (WGAN)* (Arjovsky et al., 2017). The authors use a different measure of similarities between distributions - Wasserstein or Earth Mover Distance (EM) which is smoother. Gulrajani *et al.* (Gulrajani et al., 2017) further improve it by replacing weight clipping with a gradient penalty. This architecture is called a *Wasserstein GAN with Gradient Penalty (WGAN-GP)*. The performance of GANs will be measured with Fréchet Inception Distance (FID) (Heusel et al., 2017).

Projected Gradient Descent (PGD) is an adversarial attack and was chosen here because it is considered to be a universal first-order optimization attack (Madry et al., 2018). It works by adding perturbations to a sample, which forces a neural network to

misclassify the sample. Madry formulated it as follows. Let δ lie in l_∞-ball, $\mathcal{S}$, around a sample x from distribution $\mathcal{D}$ with corresponding label y and let θ be model parameters, then

$$\min_{\theta} \rho(\theta),\ \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{S}} L(\theta, x+\delta, y) \right] \quad (2)$$

The inner optimization produces adversarial perturbations, δ, which is the PGD attack proposed by the author. The outer optimization is a countermeasure to PGD. Its strength is varied by the ε parameter or the norm which can make it more noticeable to the naked eye.

Defense GAN (Def-GAN) is a pre-processing countermeasure against adversarial input attacks, such as PGD (Samangouei et al., 2018). This method will be used extensively in this paper. The idea is that assuming the defender possesses enough clean data to train a GAN, they can use that GAN to filter out adversarial samples.

Security of GANs was considered before (Liu and Hsieh, 2019; Zhou and Krähenbühl, 2019; Xu et al., 2019; Bashkirova et al., 2019). One method is to perform *adversarial training* of the discriminator but it fails at stronger levels of PGD (Liu and Hsieh, 2019). Another approach is to apply *adversarial regularization* to the discriminator which may negatively affect convergence (Zhou and Krähenbühl, 2019). A method to protect the generator was proposed (Xu et al., 2019). It is not applicable because the target of our work is the discriminator. Finally, the work by Xu *etal*. was extended to cycle-consistent GANs (Bashkirova et al., 2019). However, these differ significantly from the GANs considered here

3 METHODOLOGY

3.1 Attack Description

A GAN can be viewed as a "student-teacher" system and MWA works by nudging the learning vector in the wrong direction. Forcing the teacher - the discriminator - to assign a surplus of value to bad examples will provide wrong training directions to the student - the generator. Alternatively, the goal can be achieved by doing the opposite - making the teacher assign too little value to legitimate examples. In this paper these two types of bad examples are referred to as *Early Epoch Decoy (EED)*[1] and *Downgrade Decoy (DGD)*, respectively. They can also be seen as two variations of MWA.

Starting with the first type, decoys are created by replacing a subset of data with bogus information while keeping the labels intact. For example, one could simply replace these samples with normal noise. However, normal noise is easy to detect, so a different source of bogus data was used.

Instead of normal noise, we used samples generated by a GAN that was trained for only an epoch or two. In this paper, these samples are referred to as *EEDs*. For this purpose, a GAN is trained using similar settings as the victim GAN. As a result, adversarial samples will share similarities with the original data, making them harder to remove. *Moreover, by overemphasizing their value to the teacher, the hope is that the system will treat premature convergence as a desirable goal.*

The discriminator will be nudged to overestimate the value of decoys using the following technique. Once decoys are created, a PGD attack will be applied to this subset of data. The discriminator is essentially a classifier, so theoretically, it is possible to modify samples with PGD such that the discriminator will classify them according to the attacker's needs. In this case, the discriminator will classify decoys as more likely to be real.

In the *downgrade* variation the opposite occurs. PGD is applied such that the discriminator assigns little value to a subset of legitimate data. The concept of attacking a GAN by applying PGD to a subset of the training data is not new and was mentioned in Section 2. However, to the best of our knowledge, the *early epoch* approach had not been tried before.

Once malicious samples are produced, they are combined with the rest of the dataset and given to a victim GAN for training. If the above hypothesis holds, the victim will produce either more malformed samples or a larger number of them. In both cases, the generating process will be compromised.

Finally, MWA can be considered a **white-box attack** because the attacker assumes knowledge of GAN hyperparameters and network architectures but not direct access to the weights. Typically, a black-box attack assumes little knowledge of the above (Miller et al., 2020). However, this limitation can be overcome using a method called *transferability* that uses a surrogate model (Madry et al., 2018; Miller et al., 2020). The attack schematic can be seen in Fig. 1.

[1] Sometimes the term "Early Stop Decoys" is used instead of "Early Epoch Decoys"

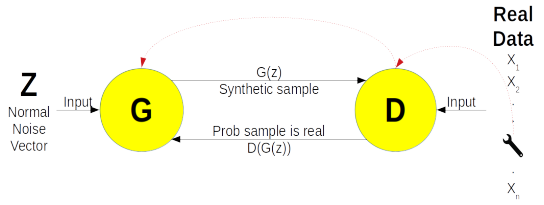

Figure 1: MWA Topology[2].

3.2 Data

This paper used two datasets: Modified National Institute of Standards and Technology Database (MNIST) (Lecun et al., 1998) and Fashion-MNIST (F-MNIST) (Xiao et al., 2017). F-MNIST was designed as a "direct drop-in replacement" (Xiao et al., 2017, p.1) for MNIST and is more sophisticated. The authors divided the dataset into 60k training and 10k testing samples. Finally, as recommended for vision datasets, images are normalized to [-1,1] with a mean of 0.5 and standard deviation of 0.5 (Goodfellow et al., 2016, p.419).

3.3 Decoys

As described in Section 3.1, the first step of MWA is to create decoys. Downgrade decoys are taken directly from the training data. Early epoch decoys, on the other hand, need crafting. They were produced by creating four types of GANs: CGAN, ACGAN, WGAN, and WGAN-GP. The neural network architectures of the discriminator and the generator for these GANs were borrowed from PyTorch-GAN package that was used as part of our implementation.[3]

To save computational power, only one class out of the ten available was used with CGANs and ACGANs. Since these two architectures are conditional, it is possible to specify which class one prefers to generate. However, GANs themselves were created from the entire training dataset.

3.4 Adversarial Perturbations

At this stage of the attack, decoys are modified such that the discriminator will either overestimate or underestimate them. This is accomplished by adding adversarial perturbations using PGD described in Section 2. The goal is to create a specially crafted noise - referred to as *adversarial perturbations* - and add it to the decoys. These perturbations change the discriminator's valuation. Since PGD requires a model to craft samples, a fully trained discriminator from the decoy production step (ref. Section 3.3) was used. Visual examples of the result can be seen in Figure 2.

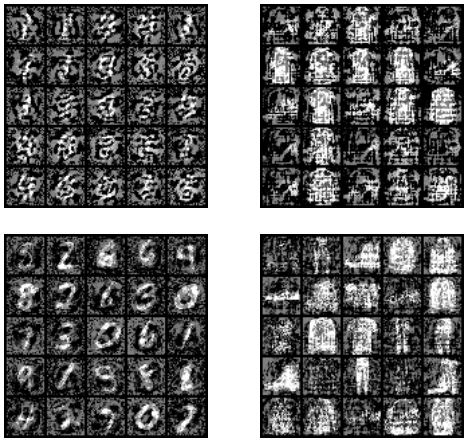

Figure 2: Examples of EEDs (top row) and DGDs (bottom row) trained on MNIST (left col) and F-MNIST (right col).

The biggest challenge at this stage was providing an appropriate loss function to PGD as stipulated in Equation 2. Binary Cross Entropy (BCE) was chosen for WGAN, WGAN-GP, and CGAN. The target was either 1.0 (early stop) or 0.0 (downgrades), and the input was whatever validity the discriminator assigned, passed through a Sigmoid function. Again, since the discriminator can be viewed as a binary classifier (i.e. real or fake), BCE is an appropriate choice.

The loss function for performing PGD on the ACGAN discriminator was constructed differently. The reason is that the discriminator is composed of two neural networks that share the same base network. One network measures data validity, while the other predicts the label. Both must be used during the PGD step because the networks are intertwined. For this reason, the loss function was similar to the one used in the actual AC-GAN. However, only the part that evaluates loss on the real data was used. The code was based on the Advertorch library created by BorealisAI (Ding et al., 2019). PGD parameter $\varepsilon = 1.0$ was used.

3.5 Detection

This section proposes a method for defending against MWA. The countermeasure used is based on Def-GAN described in Section 2. Def-GAN relies on a clean generator to detect adversarial samples. To source the generator, the same GAN was used as the one for the decoys and PGD steps above, sections 3.3 and 3.4 respectively. This decision was made to re-

[2]Wrench Icon By Estelle DB - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=57721428

[3]https://github.com/eriklindernoren/PyTorch-GAN

duce computational complexity.

However, using the same GAN possibly created a new problem. Since GANs used for crafting decoys and building the detector were the same, they were trained on the same data. Moreover, the detector usually would be trained on a subset of the data and not the whole of it. As a result, detection figures might be more optimistic than they should be.

Def-GAN was used in its capacity as a detector as described in the original paper. Detection performance was measured by creating ROCs and AUC for different thresholds.[4]

4 ANALYSIS

Four GAN variants were trained on both datasets containing 10% and 20% of malicious decoys constructed according to MWA from above. Each combination of architecture, dataset, percentage, and decoy type constitutes a *parameter set* for a single *experiment*. To produce viable statistics, each experiment was repeated *50 times*.

4.1 Null Hypothesis

To evaluate whether MWA had any effect on a GAN, two null hypotheses must be rejected:

H.1. The attack had no effect on a GAN.

H.2. Comparable results can be achieved with a simple injection of random, bogus data.

To address the first hypothesis, FID scores of an attack - parameterized as described above - will be compared to the FID scores of a clean GAN. A similar approach will be taken for the second null hypothesis. The difference is that instead of using a clean GAN, a substitute will be trained on data containing bogus samples. In this case, bogus samples will be EEDs before PGD. For simplicity, it will be referred to as a *bogus GAN*.

For an easy comparison, notch boxes are created from FID scores generated by a single combination of experiment parameters over 50 trials. If notches do not overlap, this means the difference between the two groups is statistically significant (0.05 p-value) (Krzywinski and Altman, 2014). However, to dismiss a null hypothesis, a stricter measure was taken than the standard p-value.

This difference was not sufficient to manifest visually on the generated samples during experiments. For this reason, the measure was tightened. To dismiss a null hypothesis, the entire boxes must not overlap. Moreover, such an approach guarantees statistical significance.

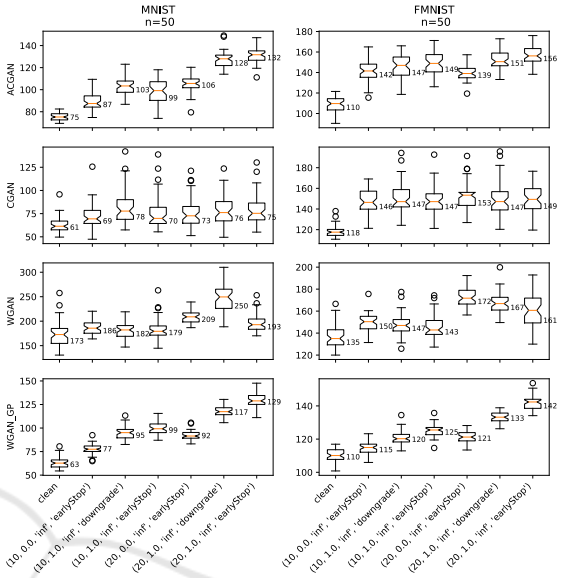

Figure 3: FID Scores of attacks on GANs. The tuples on the x-axis represent experiment parameters: injection size (%), PGD strength (ε), PGD norm, decoy type. The numbers on the boxplots indicate the medians. The columns with PGD strength of 0.0 represent the bogus GANs needed to reject **H.2.**. The clean column refers to **H.1.**. A higher score implies **less** similiarty.

4.2 Performance Summary

This subsection will analyze which attacks reject null hypotheses using Figure 3. Out of the 32 attack combinations, 13 passed both null hypotheses and can be considered successful. In the rest of the cases, the attack is considered failed. The results are summarized in the table below.

Table 1: Cases where both null hypothesis were rejected marked by a dataset were it happened. M - MNIST, F - FMNIST.

GAN	EED 10%	EED 20%	DGD 10%	DGD 20%
AC-GAN		M/F	M	M
WGAN				M
WGAN-GP	M/F	M/F	M/F	M/F

The first observation from the table is that the Wasserstein family of GANs is more vulnerable to both early epoch and downgrade variations of the attack. As was noted in Section 2, Wasserstein GANs use a more sophisticated measure to construct the loss function. This supports the part of finding **F.1.** that re-

[4]Used code from https://github.com/sky4689524/DefenseGAN-Pytorch

lates loss function complexity to attack success.

Our next observation is that success varies within each family. Again, as noted in Section 2, AC-GAN is a more advanced version of CGAN and WGAN-GP of WGAN. In both cases, advancement stems from improvements of the loss functions. Also, in both cases, these improvements made the loss functions more complex. This further supports the claim in finding **F.1.** that attack success improves with loss function complexity.

Finally, from Table 1 we can see that target data can be a significant factor in the success of the attack. The attack succeeded eight times on MNIST and five on F-MNIST. Moreover, EEDs succeeded consistently across datasets, while downgrades succeeded more often on MNIST. This implies that the claim in finding **F.1.** relating the improved attack's success to dataset complexity is correct.

4.3 Detection Analysis

Having established the conditions for a successful attack, it is important to establish whether it is possible to prevent it. The defensive strategy chosen here is based on filtering out malicious samples from the training data. The precise mechanics of this process were described in Section 3.5.

ROC curves plot the relationship between detected malicious samples and false alarms. The technical term for the first one is True Positive Rate (TPR) and False Positive Rate (FPR) for the second one. In our case, FPR implies the portion of the dataset that must be abandoned to eliminate a number of decoys that stems from a corresponding TPR.

It is impossible to establish a commonly acceptable FPR because it depends on the case. For example, in some situations, eliminating 10% of the data is unacceptable, while 50% might be dispensable in others. However, to proceed with the analysis, we will consider a 20-30% sacrifice of the dataset to be the threshold of an acceptable loss of data.

To visualize the effect of the countermeasure on the attack, a figure similar to Figure 3 will be created. However, this figure will contain FIDs that we would have received if we filtered the data before training the GAN. These numbers were simulated as follows.

Re-training the GANs on smaller numbers of decoys was computationally prohibitive. For this reason, an approximation was made. FID scores in Figure 3 exhibited approximately a linear relationship to the amount of decoys. This applies to both types of decoys and bogus data. Thus, the new FID scores in Figure 4 were simulated by exploiting this linear relationship. FID scores were linearly projected to lower

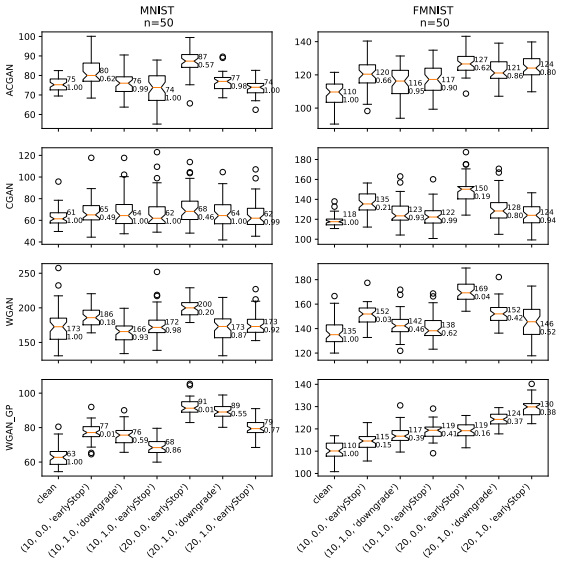

Figure 4: FID Scores After Filtering. The tuples on the x-axis represent the injection size , PGD strength, and decoy type. The numbers on the boxplots indicate the medians and the TPRs.

size injections. New sizes were determined using TPRs that correspond to the acceptable FPR range. TPRs were averaged over iterations.

The attack was successful only for WGAN-GP: on F-MNIST at 10% of EEDs and at 20% for both types of decoys. Two implications stem from the above. First, the attack survived the countermeasure only for the most advanced GAN (Lucic et al., 2018) and on the more sophisticated dataset. More, EEDs survived twice, while downgrades only once. This proves finding **F.2.** that our proposed attack is likely to evade detection for more complex loss functions and datasets.

Moreover, judging from FID dynamics in Figure 4, 10% is the lower bound for EEDs. For downgrades, the lower bound is somewhere in the 10-20% range. Together, these facts support finding **F.3.**.

Now let us consider visual inspection as a defensive tool. As was mentioned, MWA with downgrade decoys corresponds to efforts from earlier literature. These decoys are easier to identify as malicious (ref. Fig. 2). Individual items can be discerned and the noise always follows a similar pattern. This indicates a lack of randomness and a possible presence of intent.

The situation with EEDs is different. Often the original data cannot be discerned at all and the noise does not follow an obvious pattern (ref. Fig. 2). Thus, we conclude that it is likely possible to conceal EEDs from a human observer. This supports finding **F.4.**.

4.4 Decoy Comparison

For the purpose of this comparison, only cases that pass both null hypotheses will be examined. From Figure 3 we can see that EEDs outperformed downgrades in all cases, but two: (1) AC-GAN trained on MNIST with 10% decoys, and (2) WGAN trained on MNIST with a 20% injection.

It is possible to conclude that EEDs outperform downgrades quantitatively with standard statistical significance (Krzywinski and Altman, 2014). But is statistical significance enough to claim an improvement in this domain? When EEDs outperform downgrades, they do so with an average FID increase of approximately 5%. Which is the first argument to support finding **F.1.** that existing attack proposals were improved upon.

Now, let us consider decoy performance after applying the countermeasure. In this case, the attack is successful only for WGAN-GP trained on F-MNIST with a 10% and 20% injection. In the 20% case, the bogus GAN used as a basis for **H.2.** had a median FID of 119. EEDs and DGDs had FIDs of 130 and 124, respectively. The first one was 9% higher and the second one was 4% higher than the **H.2.** median. In the 10% case, only EEDs succeeded and outperformed **H.2.** with 3.5%.

This tells us that the early epoch approach outperforms the downgrades when the attack rejects the null hypotheses and evades detection. Moreover, this happens with standard statistical significance. Noting that the downgrade approach is an existing scientific proposal, the result forms another argument towards finding **F.2.** that an improvement was achieved. However, in both cases, improvement is true only for complex loss functions and datasets.

5 CONCLUSIONS AND FUTURE WORK

This paper aimed to explore the vulnerability of GANs to adversarial input attacks. Specifically, it was unclear what conditions are necessary for such an attack to succeed and whether it is possible to defend against. An attack was devised based on a known approach to answer these questions. The latter was improved by using different source data for crafting adversarial samples. Empirical results showed that this is indeed a superior approach.

The gains were made on the more sophisticated dataset or GANs with more advanced loss functions. A similar pattern held when trying to break through a countermeasure. The attacks overcame the defence only on the F-MNIST dataset and WGAN-GP architecture. Our version succeeded for injections of 10% and 20%, while the baseline succeeded only with 20%. In both cases, our version achieved a higher FID score, which indicates better performance.

However, when a countermeasure was not applied, both attacks were successful more often, with EEDs performing better. Finally, the adversarial samples proposed in this paper are more difficult to identify visually.

The countermeasure remains an open question. Other approaches were proposed that may perform better but with additional costs. Given that some GANs are defenceless using the current countermeasure, future work will investigate the cost-benefit equilibrium of other defences.

REFERENCES

Alqahtani, H., Kavakli-Thorne, M., and Kumar, G. (2021). Applications of Generative Adversarial Networks (GANs): An Updated Review. *Archives of Computational Methods in Engineering*, 28(2):525–552.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 214–223. PMLR.

Bashkirova, D., Usman, B., and Saenko, K. (2019). Adversarial self-defense for cycle-consistent GANs. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 635–645.

Bora, A., Price, E., and Dimakis, A. G. (2018). AmbientGAN: Generative models from lossy measurements. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W. F., and Sun, J. (2017). Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. In *Proceedings of the 2nd Machine Learning for Healthcare Conference*, pages 286–305. PMLR.

Ding, G. W., Wang, L., and Jin, X. (2019). Advertorch v0.1: An Adversarial Robustness Toolbox based on PyTorch. *arXiv:1902.07623 [cs, stat]*.

Engelmann, J. and Lessmann, S. (2021). Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications*, 174:114582.

Gong, M., Xu, Y., Li, C., Zhang, K., and Batmanghelich, K. (2019). Twin Auxiliary Classifiers GAN. *Advances in neural information processing systems*, 32:1328–1337.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, Cumberland, UNITED STATES.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA. MIT Press.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein GANs. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*. Curran Associates, Inc.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Proceedings of the 30th Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial Examples Are Not Bugs, They Are Features. In *Proceedings of the 32nd Annual Conference on Neural Information Processing System, Vancouver, BC, Canada*, Vancouver, BC, Canada. Curran Associates, Inc.

Kaviani, S. and Sohn, I. (2021). Defense against neural trojan attacks: A survey. *Neurocomputing*, 423:651–667.

Krzywinski, M. and Altman, N. (2014). Visualizing samples with box plots. *Nature Methods*, 11(2):119–120.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (Nov./1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Liu, X. and Hsieh, C.-J. (2019). Rob-GAN: Generator, Discriminator, and Adversarial Attacker. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11226–11235.

Lucic, M., Kurach, K., Michalski, M., Bousquet, O., and Gelly, S. (2018). Are GANs created equal? a large-scale study. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 698–707, Red Hook, NY, USA. Curran Associates Inc.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Miller, D. J., Xiang, Z., and Kesidis, G. (2020). Adversarial Learning Targeting Deep Neural Network Classification: A Comprehensive Review of Defenses Against Attacks. *Proceedings of the IEEE*, 108(3):402–433.

Mirza, M. and Osindero, S. (2014). Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat]*.

Odena, A., Olah, C., and Shlens, J. (2017). Conditional Image Synthesis with Auxiliary Classifier GANs. In *Proceedings of the International Conference on Machine Learning*, pages 2642–2651. PMLR.

Samangouei, P., Kabkab, M., and Chellappa, R. (2018). Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Thekumparampil, K. K., Khetan, A., Lin, Z., and Oh, S. (2018). Robustness of conditional GANs to noisy labels. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Proceedings of the 31st Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10292–10303.

Wang, Z., Myles, P., and Tucker, A. (2019). Generating and Evaluating Synthetic UK Primary Care Data: Preserving Data Utility & Patient Privacy. In *Proceedings of the 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, pages 126–131, Cordoba, Spain. IEEE.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747 [cs, stat]*.

Xu, Z., Li, C., and Jegelka, S. (2019). Robust GANs against Dishonest Adversaries. In *Proceedings of the International Conference on Machine Learning Workshop on Security and Privacy of ML*.

Yadav, D. and Salmani, S. (2019). Deepfake: A Survey on Facial Forgery Technique Using Generative Adversarial Network. In *Proceedings of the 2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 852–857.

Zhou, B. and Krähenbühl, P. (2019). Don't let your Discriminator be fooled. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.