


Article

Universal Adversarial Training Using Auxiliary Conditional Generative Model-Based Adversarial Attack Generation

Hiskias Dingeto and Juntae Kim * 

Department of Computer Science and Engineering, Dongguk University, Seoul 04620, Republic of Korea; hisku@dgu.ac.kr

* Correspondence: jkim@dongguk.edu; Tel.: +82-2-2260-3712

Abstract: While Machine Learning has become the holy grail of modern-day computing, it has many security flaws that have yet to be addressed and resolved. Adversarial attacks are one of these security flaws, in which an attacker appends noise to data samples that machine learning models take as input with the aim of fooling the model. Various adversarial training methods have been proposed that augment adversarial examples in the training dataset for defense against such attacks. However, a general limitation exists where a robust model can only protect itself against adversarial attacks that are known or similar to those it was trained on. To address this limitation, this paper proposes a Universal Adversarial Training algorithm using adversarial examples generated by an Auxiliary Classifier Generative Adversarial Network (AC-GAN) in parallel with other data augmentation techniques, such as the mixup method. This method builds on a previously proposed technique, Adversarial Training, in which adversarial examples produced by gradient-based methods are augmented and added to the training data. Our method improves the AC-GAN architecture for adversarial example generation to make it more suitable for adversarial training by updating different loss terms and testing its performance against various attacks compared to other robust adversarial models. In this way, it becomes apparent that generative models are better suited for boosting adversarial robustness through adversarial training. When tested using various attack types, our proposed model had an average accuracy of 97.48% on the MNIST dataset and 94.02% on the CelebA dataset, proving that generative models have a higher chance of boosting adversarial security through adversarial training.

Keywords: adversarial training; adversarial attacks; generative models; conditional generative adversarial network; auxiliary conditional generative adversarial networks



Citation: Dingeto, H.; Kim, J. Universal Adversarial Training Using Auxiliary Conditional Generative Model-Based Adversarial Attack Generation. *Appl. Sci.* **2023**, *13*, 8830. <https://doi.org/10.3390/app13158830>

Academic Editor: Luis Javier García Villalba

Received: 16 June 2023
Revised: 26 July 2023
Accepted: 27 July 2023
Published: 31 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Despite the popularity of machine learning models over the past decades, there is a risk of basic functions being disrupted by different attacks. Adversarial attacks have been gaining in popularity due to their simplistic attack generation. An attacker can produce these attacks by adding subtle noise to input data to make the input malicious. This malicious input, whether image, audio, video, or text, results in a normal sample that most machine learning models misclassify without any adversarial defense [1–9]. The effects of adversarial attacks can even be extended to areas such as numerical simulation and stability [10,11]. Adversarial training aims to resolve the security flaws many machine learning models face due to the malicious samples mentioned above [2,9,12–15]. Even though various methods have been proposed to defend models from these attacks, adversarial training is by far the most robust solution to the problem. Training a model adversarially requires augmentation of adversarial examples made through different methods and adding them to the training dataset. This training method works well if the malicious sample created by the attacker uses the same or similar methods to those used when generating the augmented adversarial dataset. There can be issues if an attacker uses a different attack method to

create malicious samples. When a model is not trained on a specific type of adversarial example, it fails to defend itself from the attack [16,17]. Anish Athalye et al., researchers in the field of adversarial attacks and defenses, have shown in their research [16] that it is possible to easily circumvent state-of-the-art adversarial defenses, providing evidence of the need for better defenses against adversarial attacks. Even though the authors did not explain why this phenomenon occurs, it can be hypothesized that it is due to holes in adversarial defenses created through adversarial training that allow attackers the chance of exploiting them. The process of adversarially training a model aims to teach it to recognize data samples even though there is a perturbation in the input sample. This depends entirely on the adversarial examples used to train it. Supposing that a model is provided with an adversarial sample that it was not trained to defend against, it cannot defend itself, which is a weakness in the current adversarial training process. Using one type of adversarial attack for training does not defend the model from the plethora of other attacks that can be exploited. Considering that over a thousand papers have already been published on the topic, it is clear that a universal defense is more necessary than ever [18].

This article's primary focus is showing that generative models can aid in the process of adversarial training by generating unrestricted adversarial examples, as mentioned in [19,20]. In this way, models can be trained on adversarial samples that are not confined to "narrow" distributions that traditional methods such as the Fast Gradient Sign Method [1], Projected Gradient Descent [9], Basic Iterative Method [21], and Jacobian-based Saliency Maps Attacks [22] usually generate. Unrestricted adversarial examples from generative networks provide a potential solution, as these samples are generated from scratch. If visualized on a decision boundary, traditional adversarial examples are close to the model's decision boundary. In contrast, unrestricted examples contain samples close to the model boundary and further away. In this paper, we propose that the approach of creating unrestricted samples from generative models has the potential to create a universal defense. Throughout this paper, the proposed technique is explored in depth with a conceptual explanation, experimental setup, results, and analysis of the outcome.

In summary, the contributions of this paper are:

- To propose the use of an auxiliary generative model for adversarial training purposes;
- To enhance model robustness by adopting the AC-GAN architecture and using it to generate adversarial samples for adversarial training;
- To show experimental test results of AC-GAN-based adversarially trained models and compare their attack robustness with adversarially trained models using different methods.

Using generative models allows for better adversarial robustness, as it creates samples from various distributions and prevents the models from being fooled by different types of adversarial examples. In this paper, background research is conducted to introduce the necessary concepts and compare them with previously proposed techniques in adversarial training, the methodology is presented, and the experimental results are analysed.

2. Background Research

2.1. Conditional Generative Adversarial Networks and AC-GANs

Generative Adversarial Networks (GANs) have been circulating in the machine learning community for the past several years. Goodfellow published the original idea in 2014 [23], and various additions to GANs have subsequently been used in many computing and machine learning-related areas. Compared to their counterparts that follow a similar concept of deep generative modeling, such as Variational Auto-encoders [24], GANs have been deemed superior in various aspects [25].

As explained in the original paper, the general structure of a GAN sets two models against each other to determine whether this process improves either of the models. One of the models, the Generator, tries to generate data that are as close to the original dataset as possible. The other model, the Discriminator, attempts to identify whether the data it receives are real.

The process carried out in GANs is best explained by comparison with the “Minimax Game”, i.e., using the Minimax Algorithm. The concept of a Minmax game is that there are two players; the maximizer tries to reach the highest score possible, while the minimizer attempts to reach the lowest score possible. In the case of GANs, the Discriminator continuously tries to differentiate the original data from the artificial data produced by the Generator, whether pictures, audio, or, ideally, any other type of data format. The Generator takes feedback from the Discriminator to improve its output and eventually generates data that the Discriminator cannot differentiate from the original data.

A variation of GANs that is worth mentioning here is Conditional GAN (CGAN), first proposed by M. Mirza et al. (2014) [26]. What makes CGANs different is that the model is conditioned on a class label, where the provided label allows for control over the labels of the samples to be generated. In his paper, we use an extension of CGANs named Auxiliary Classifier GAN, introduced by A. Odena et al. (2017) [27], to generate adversarial samples. In the case of CGANs, the Generator and Discriminator are conditioned on extra information y , where y is any auxiliary information, including class labels. ACGANs extend this idea further by adding an auxiliary classifier that allows the Discriminator to predict whether the image is real or fake and label the generated image as the target class provided to the Generator. Hence, the Discriminator maximizes the probability of correctly classifying the generated image and accurately predicting the class label of the generated image. The authors of [27] represented these two processes through the objective functions in Equations (1) and (2):

$$L_s = E [\log P(S = real|X_{real})] + E [\log P(S = fake|X_{fake})] \quad (1)$$

$$L_c = E [\log P(C = c|X_{real})] + E [\log P(C = c|X_{fake})] \quad (2)$$

where L_s is the log-likelihood of the correct source and L_c is the log-likelihood of the correct class. An architecture similar to the one proposed in [20] is used to generate adversarial examples, with a modification made to the architecture to generate adversarial examples from all class labels on the MNIST and CelebA datasets for adversarial training, along with minor adjustments to better fit our objective.

Recently, AC-GANs have seen through different modifications to improve their performance. Despite improvements over the vanilla GAN and other types of conditional GANs, research has shown that they are susceptible to various instability issues, as shown in [28–30]. The results of Kang et al. (2021) [28] showed the most significant bump in performance by introducing an extension of the previously used cross-entropy loss cost function. This customized loss function, called the Data-to-Data Cross-Entropy Loss (D2D-CE), resolves the instability in the original AC-GAN when using the common cross-entropy loss function. The resulting ReAC-GAN model remains one of the highest-performing AC-GAN modifications by far and is grouped as one of the best performing conditional generative models, as shown in [31].

2.2. Adversarial Attacks

Adversarial attacks are attacks performed on machine learning algorithms to cause models to make mistakes during tasks such as classification. The attacks are usually performed by adding a small perturbation or noise to input examples, making them adversarial examples. C. Szegedy et al. (2014) [2] coined the term “Adversarial Examples” and proposed one of the first adversarial attacks; subsequently, various studies have tried to unveil the root cause of this phenomenon. The most conclusive research in explaining adversarial examples was carried out by I. Goodfellow et al. (2015) [1]. The authors attributed the ability of adversarial examples to fool models to neural networks not learning the “true underlying concepts that determine the correct output label”. In other words, the networks do not understand the actual training data sample, instead drawing a false mathematical perception to identify the class of the sample as long as there is no artificial noise that can affect that system. The authors compare this phenomenon to a “Potemkin

village”, i.e., a false construct that does not accurately represent reality. According to their research, the linear nature of neural networks is what allows for a small amount of noise (adversarial perturbation) to affect the output of the model. In other words, while machine learning models draw a mathematical relation between their received inputs and the outputs from the training data, the representation they create does not represent what human beings perceive. Given a clean Labrador picture (Figure 1a) and one that has been adversarially perturbed (Figure 1b,c), we as human beings see and perceive all three pictures as a Labrador-breed dog. However, a state-of-the-art machine learning model specializing in classifying dog breeds mistakenly classifies the adversarial images (Figure 1b,c) as Saluki and Weimaraner dog breeds, dogs that have significantly different appearances compared to the original image. These results show that our current machine learning models are not able to understand the world as humans do.

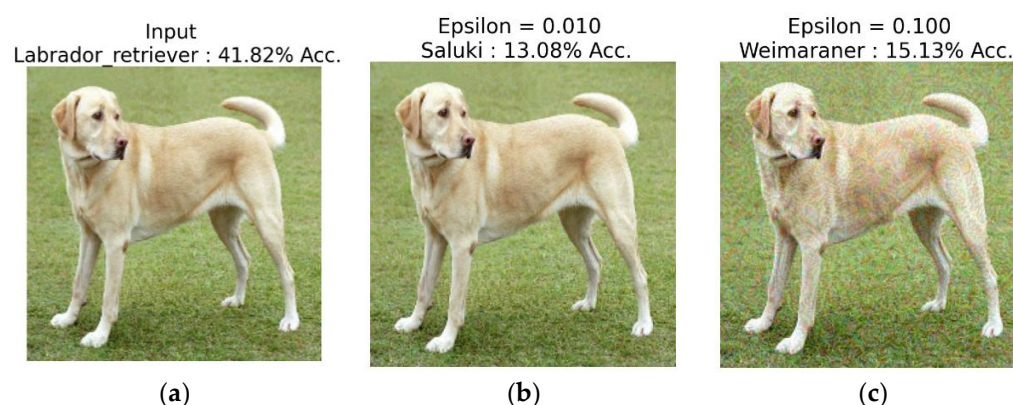


Figure 1. Clean (a) and adversarial (b,c) images of a Labrador-breed dog, showing the classification results and epsilon values for adversarial perturbation (Image Credit: TensorFlow Authors 2019 [32]).

Both C. Szegedy et al. (2014) [2] and I. Goodfellow et al. (2015) [1] proposed adversarial attacks based on the gradient of a given model, respectively, Limited-Memory BFGS (L-BFGS) and the Fast Gradient Sign Method (FGSM). After L-BFGS and FGSM, various types of attacks have used different algorithms to generate adversarial perturbation. Examples include, among many others, Jacobian-based Saliency Map Attack (JSMA) by N. Papernot et al. (2015) [22], Carlini and Wagner Attack (C&W) by N. Carlini et al. (2017) [33], Basic Iterative Method by A. Kurakin et al. (2017) [21], and Projected Gradient Descent (PGD) by A. Madry et al. (2018) [9]. The methods listed above are classified as gradient-based methods, in which perturbations are generated by utilizing the gradient such that the generated noise is large enough to change the resulting label of the model while at the same time being small enough to make no noticeable changes to the data sample. For familiarity, the section below briefly summarizes the adversarial attacks used in this paper:

- Fast Gradient Sign Method (FGSM) [1]: FGSM is an adversarial attack generation algorithm that is performed by calculating the gradient with respect to each image pixel and adjusting it to maximize the loss value.
- Projected Gradient Descent (PGD) [9]: This attack is an update of FGS; while FGSM takes only one step to calculate the generated noise, PGD improves the attack efficacy through a multi-step process.
- Simultaneous Perturbation Stochastic Approximation (SPSA) [34,35]: SPSA uses random approximations with finite difference estimates in cases where an analytic gradient can be used.
- Momentum Iterative Method (MIM) [36]: this attack uses momentum-based algorithms to “boost” adversarial attacks, i.e., improve the generated adversarial examples by adding momentum to the process and avoiding local maxima that are not sufficient during each iteration.

- Basic Iterative Method (BIM) [21]: BIM extends the idea of FGSM and implements the attack multiple times with smaller step sizes.
- Unrestricted Adversarial Examples [19,20]: the authors who proposed this idea took a different approach to generating adversarial examples; in this approach, a GAN learns to create adversarial examples from scratch by searching the latent space of the model that is being attacked. Because our architecture is based on their proposed solution, this alternative is discussed in detail in the following sub-section.

Machine learning plays many essential and security-sensitive roles, from image classification to spam filtering and malware detection. For this reason, adversarial attacks pose a huge risk to many application areas. The most common security issue here is adversarial attacks in the form of images. Many of the introductory works in this area were carried out using image data from C. Szegedy et al. (2014) [2] and I. Goodfellow et al. (2015) [1]. These authors focused on image-based attacks that engineer adversarial perturbation to add to the sample image. Despite the initial research focus on images, there has been research on adversarial examples that has successfully reduced the performance of machine learning models that work with other forms of data. Studies such as [4,5,37] have proven that adding perturbation to audio input can affect speech-to-text transcription neural networks. Similar attacks have been able to affect other Machine Learning applications as well, such as Natural Language Processing (NLP), areas including sentiment classification, fake news detection, neural machine translation, spam filtering, malware detection, bot detection, network intrusion detection, and even self-driving cars, as shown in [3,13,16,38–50].

In this paper, we use PGD [9] and Unrestricted Adversarial Examples [19,20] for adversarial training and use PGD [9], FGSM [1], SPSA [34,35], Unrestricted Adversarial Examples [19,20], MIM [36], and BIM [21] for testing purposes. FGSM and PGD are both classified as gradient-based adversarial example generation algorithms. These attacks use the gradients produced during backpropagation to generate an optimal perturbation added to the original input. Despite most attacks being gradient-based, attacks such as SPSA use “gradient-free optimization” to generate adversarial examples. In the specific case of SPSA, Uesato et al. (2018) [35] states that a technique proposed in [34] can be used to approximate the gradients through finite difference estimates in random directions, providing an efficient way of producing adversarial examples. Gradient-based and gradient-free attacks were used to challenge the universality of both previously proposed adversarial training methods and our proposed method.

2.3. Adversarial Training

Adversarial Training, as the name implies, is the process of training a model with adversarially perturbed images. The training process might use a dataset containing both clean and adversarial examples, as in [2], or a dataset with only adversarial examples, as in [13]. There have been various improvements and additions to the method; essentially, however, the algorithm should be able to augment or transform the training dataset with adversarial examples. Research has shown that training the model with perturbed examples significantly improves performance when an attack occurs [12–15].

Adversarial attacks can be divided into several classes based on their training techniques. According to T. Bai et al. (2021) [12], adversarial training is generally divided into six main types, with other variants not classified under the main branches. Most variations here are based on the initially proposed adversarial training model in [2]. Examples such as [15] have suggested using augmented adversarial examples generated from different target models to generate different types of adversarial samples, while other research, including [51–53], has changed the perturbation with respect to each pixel, i.e., epsilon (ϵ), in an adaptive fashion that allows for better generation. These attacks can be implemented on regression models, as done by Weixia et al. [54]. Although it is outside of our research scope in this paper, applying the proposed method to regression models is an interesting future direction.

The concept of using generative models for adversarial training is a relatively recent concept that has been gaining traction. In this paper, we compare Madry's adversarial training method proposed in [9] with our adversarially robust models based on Unrestricted Example generation using the AC-GANs proposed in [20] improved for Adversarial training. Research by X. Yin et al. (2022) [55] proposed using a generative classifier to boost adversarial performance, which is very different from our approach; similar research was conducted by F. Catak et al. (2020) [56]. In this latter case, the authors used autoencoders to generate adversarial examples instead of GANs. Our reasoning for using GANs is that they can be conditioned to produce the required results while generating clear samples. Research has shown that GANs have superior performance in generating well-defined quality images [57]. This paper proves that GANs have better potential to generate universal adversarial examples and, as a result, boost defenses. This improvement in robustness results from better coverage of adversarial samples generated from scratch. This generative nature adds a variety of samples that attackers might exploit; such samples are not usually discovered through traditional algorithms such as PGD and FGSM.

2.4. Unrestricted Adversarial Examples

As mentioned in [16] by T. Brown et al. (2018), there have been various studies on generating unrestricted adversarial examples where some specific norm constraint does not constrain the generated adversarial examples. More specifically, this paper focuses on Y. Song et al. (2018) [20], where the authors used a generative model to create adversarial samples from scratch. Their method is based on searching for these adversarial samples in the latent space of the model at hand. Unlike other methods, which generally add small perturbations, generating unrestricted adversarial examples through an AC-GAN performs a search that opens new possibilities for discovering adversarial samples that standard algorithmic attack methods cannot create. The authors demonstrated the legitimacy of their results through human evaluation on datasets such as MNIST, SVHN, and CelebA.

To further explain the generative model process, Y. Song et al. (2018) [20] formulated the objective functions of the Generator and Discriminator in an adversarial example generation context as follows in Equations (3) and (4):

$$\min_{\theta} -E_{z \sim P_z, y \sim P_y} [d_{\phi}(g_{\theta}(z, y)) - \log c_{\psi}(y|g_{\theta}(z, y))] \quad (3)$$

$$\begin{aligned} \min_{\phi, \psi} E_{z \sim P_z, y \sim P_y} [d_{\phi}(g_{\theta}(z, y))] - E_{x \sim P_x} [d_{\phi}(x)] - E_{x \sim P_x, y \sim P_{y|x}} [\log c_{\psi}(y|x)] \\ + \lambda E_{\tilde{x} \sim P_{\tilde{x}}} \left[\left(\left\| \nabla_{\tilde{x}} d_{\phi}(\tilde{x}) \right\|_2 - 1 \right)^2 \right] \end{aligned} \quad (4)$$

The equations were derived from the original Wasserstein GAN [58] and AC-GAN [27] formulations and customized to an adversarial example generation scenario. Equation (3) provides a minimization function for generating unrestricted adversarial examples. According to the equation, with c_{ψ} being the auxiliary classifier, it can be seen that the Generator's objective is to minimize the loss over the classification of the generated images. On the other hand, in Equation (4), the Discriminator is trying to minimize the loss between the output generated samples $g_{\theta}(z, y)$ and the output of the original samples, x . The last term of the equation is a gradient penalty term that encourages the discriminator's gradient to have a norm of 1.

Based on the above equations, our approach modifies this, aiming to primarily benefit adversarial training by allowing for the generation of adversarial samples for better robustness. The architecture was changed due to the constraint implemented when generating adversarial examples. Removing this constraint and implementing different loss functions can improve adversarial training coverage; this can be seen through the proposed methodology and experimental results in the following sections.

2.5. Mixup Data Augmentation

In addition to adding augmented unrestricted adversarial examples to the original dataset, we implemented a domain-agnostic data augmentation technique proposed by H. Zhang et al. (2018) in [59]. This is a method in which a combination of randomly selected image pairs is generated and added to the original dataset for training. The later sections show that the augmentation algorithm interpolates two MNIST numbers from the training set along with their labels. Training is accomplished by creating virtual training examples. This method can be summarized using the following two equations, which show how the virtual training examples are created.

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j \quad (5)$$

$$\tilde{y} = \lambda y_i + (1 - \lambda)y_j \quad (6)$$

Equations (5) and (6) show two examples drawn at random from the training data interpolated using the parameter $\lambda \in [0, 1]$. Both \tilde{x} and \tilde{y} are the merged input vector and label.

As shown in [60], mixup augmentation can improve adversarial robustness by linearly interpolating multiple samples and their labels. Due to these improvements, we used the mixup technique to create a Universal Adversarial Defense.

3. Methodology

3.1. Auxiliary Classifier GAN to Generate “Unrestricted Adversarial Examples”

As mentioned above, Auxiliary Classifier GANs can generate high-quality samples due to the auxiliary classifier. In this paper, we aim to improve adversarial sample generation through the AC-GAN proposed by Y. Song et al. (2018) in [20] by modifying it for adversarial training. This improvement achieves a wider range of adversarial example searches in the latent space of the model, allowing for broader coverage of adversarially trained defenses. T. Brown et al. (2018) [19] were the first to define the term “Unrestricted Adversarial Examples”, suggesting the generation of unconstrained samples as opposed to previous methods that were norm-constrained. According to [20], through the use of generative models it is potentially possible to find adversarial examples that are not limited by the size of the perturbation, which is a superset for generative-based adversarial samples.

Vanilla adversarial training is generally based on augmenting a clean dataset with adversarial images. These images are generated by adding small perturbations to clean samples using methods such as FGSM [13] and PGD [9]. Despite their widespread use, however, it has been shown in studies such as [1,12,15,25] that adversarially trained models lack generalization with respect to unseen attacks. This lack of generalization is due to the algorithms only generating constrained adversarial examples, meaning that defenses trained only on these samples can be circumvented by different kinds of attacks that use other l_p norms or larger epsilon values (ϵ) compared to those used during adversarial training [6,7].

The AC-GAN model solves the problem of constrained adversaries by looking through the latent space and finding samples that qualify as adversarial examples. As shown in Equations (1) and (2), the Generator aims to produce samples that resemble images from the target class C but that are classified by the Generator as the source class S. The generated samples can be used for adversarial training. The general formulation, first stated in [9] (simplified by F. Tramer et al. (2020) [15]), is shown in the equation below:

$$\min_{\theta} E_{(x,y) \sim D} \left[\max_{\|x^{adv} - x\|_{\infty} \leq \epsilon} L(h(x^{adv}), y_{true}) \right] \quad (7)$$

Madry et al. (2018) [9] explained adversarial examples through a min-max formulation, as shown above. The inner maximization refers to an adversary that is trying to maximize the loss L of an input x^{adv} perturbed by a noise $n = x^{adv} - x$, which is less than ϵ . The outer minimization minimizes the loss from any perturbed sample. This objective function is the goal of every adversarially robust model. In our case, training is carried out using examples from an AC-GAN. Hence, the min-max formulation is modified as follows:

$$\min_{\theta} E_{(x,y) \sim D} \left[\max L(h(x^g), y_{true}) \right] \quad (8)$$

where x^g is an adversarial sample generated by an AC-GAN. Note that there are no constraints on x^g because it is an unrestricted adversarial example.

3.2. Modifying AC-GAN Architecture for Adversarial Training

Taking a closer look at the AC-GAN architecture proposed in [17], which is used in constructing Unrestricted Adversarial Examples from scratch, it can be noted that three different loss functions are proposed. According to the authors, the first loss L_0 improves the performance of the target classifier f in predicting y_{target} . It is expected that the classifier classifies, or rather misclassifies, the input with a source label y_{source} to the target label y_{target} , as shown in Equation (9).

$$L_0 \triangleq -\log f(y_{target} | g_{\theta}(z, y_{source})) \quad (9)$$

On the other hand, the loss term L_1 , shown in Equation (10) below, places a soft constraint on the search region of the noise z such that more diverse samples can be produced. In this paper, we argue that the loss function restricts the search region instead, and that replacing it with another term would be more efficient. This improvement is discussed further in the later parts of this section.

$$L_1 \triangleq \frac{1}{m} \sum_{i=1}^m \max \left\{ |z_i - z_i^0| - \epsilon, 0 \right\} \quad (10)$$

The last loss component, as shown in the equation below, encourages the auxiliary classifier c_{φ} to correctly classify the generated images, i.e., classify them as their source label.

$$L_2 \triangleq -\log c_{\varphi}(y_{source} | g_{\theta}(z, y_{source})) \quad (11)$$

To summarize all the loss functions mentioned above, the total loss helps the model find the latent space z that produces quality unrestricted adversarial examples; hence, minimizing the loss L helps to optimize the latent space provided to the Generator. The first loss L_0 helps the target classifier f to predict y_{target} , which is the wrong target that an attacker might exploit. On the other hand, L_1 constrains the search range of the latent space to a certain degree. Our research shows that this restriction reduces the efficacy and universality of the generated adversarial samples and weakens adversarial training on the said samples. Lastly, L_2 allows the auxiliary classifier to make the correct prediction y_{source} .

As shown in Figure 2, the Generator is trying to minimize the summation of all three loss functions. The hypothesis is that the soft constraint L_1 limits the search range of the Generator, thereby not allowing for an efficient defense. This is shown in the experiments section by comparing adversarial training on the original AC-GAN architecture proposed in [20] with our modified method. The experiments involve removing different loss terms, including the soft-constraint loss L_1 , adding a custom loss function from [28] named Data-to-Data Cross-Entropy Loss (D2D-CE), and training the proposed model on unrestricted examples to test against different types of attacks. Kang et al. 2021 [28] improved the performance of traditional AC-GAN by creating data-to-class and data-to-data relationships, making the produced samples more visually similar to the original target data. Therefore, testing different combinations of loss functions from the above four has proven that using D2D-CE loss and removing the soft constraint generates unrestricted samples, in turn

creating a more robust adversarial training output. The formulation of the D2D-CE loss as shown by Kang et al. (2021) [28] is provided in the equation below.

$$L_{D2D-CE} \triangleq \frac{1}{N} \sum_{i=1}^N \log\left(\frac{\exp([f_i^\top v_{y_i} - m_p]_- / \tau)}{\exp([f_i^\top v_{y_i} - m_p]_- / \tau) + \sum_{j \in N(i)} \exp([f_i^\top v_{y_i} - m_p]_+ / \tau)}\right) \quad (12)$$

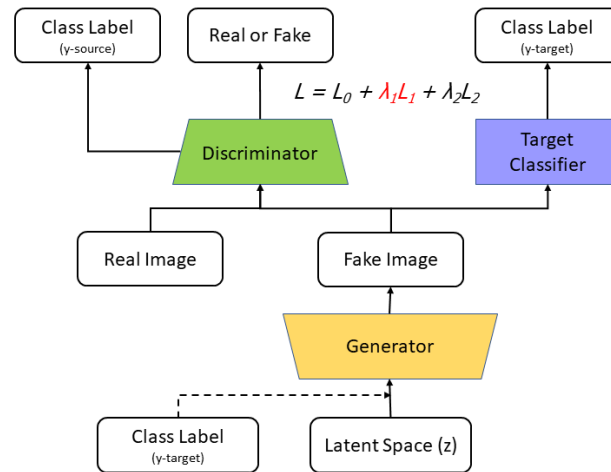


Figure 2. Modifications to the AC-GAN architecture; red (λ_1, L_1) shows the replaced loss function for generation of broader unrestricted adversarial samples.

Figure 3 below compares the proposed model to the original adversarial model trained on unrestricted examples from [20]. The models were tested on a range of attacks in order to test their universality. When looking at the drop in accuracy of the unrestricted adversarial model (blue line) compared to our proposed model, it is clear that the improvements made to the models improved the overall robustness against all the attacks. The results in the experiments section demonstrate how the different types of loss functions used in unrestricted example generation throughout the different versions of the models vary in accuracy when tested against various types of attacks.

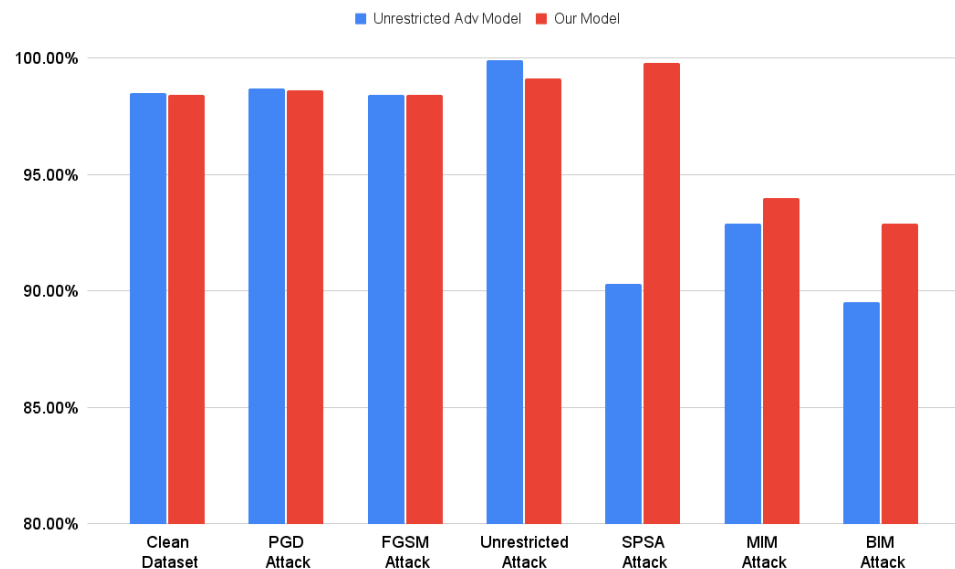


Figure 3. Visualization of adversarial robustness between different versions of proposed defenses.

3.3. Augmenting Model Training with Generated Images

After the adversarial samples are generated, they are used to augment the original dataset during the training of a new model. A convolutional model with four layers is used to generate adversarial examples from the AC-GAN and retrain the new dataset that includes the samples. Figure 4 below gives an overview of the process.

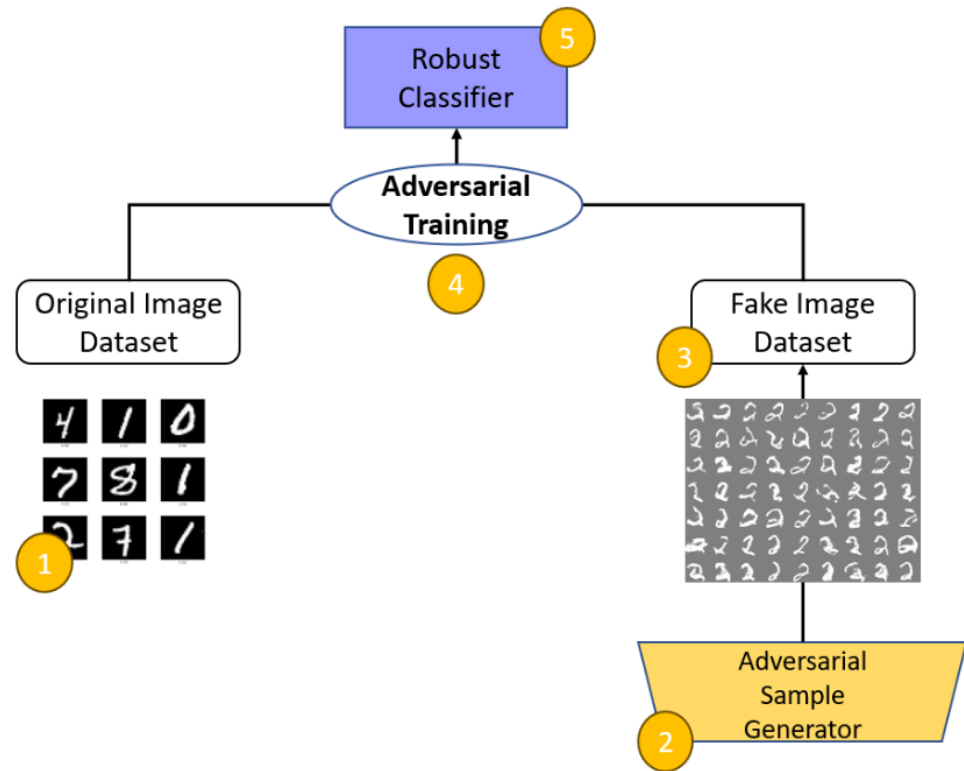


Figure 4. AC-GAN architecture used for adversarial training. The original MNIST dataset (1) is used in the adversarial sample generation (2) to create an augmented dataset. The adversarial dataset (3) is then merged with the original image dataset to undergo adversarial training (4). This training process ultimately results in a robust classifier.

This training method allows the model to identify adversarial samples that are not close to the learned distribution and to classify them correctly. Compared to adversarial training on samples generated from algorithms such as FGSM and PGD, the adversarial training method proposed in this paper allows the resulting models to identify adversarial examples that are not restricted to adding small noise to samples from the target dataset. For this reason, an attacker may find it challenging to exploit adversarial examples from different generation algorithms, which is shown in later sections where the proposed method outperforms commonly used adversarial training methods. Additionally, the generated adversarial samples underwent mixup augmentation to enhance their adversarial robustness. As explained in the background section, this technique blends different images from different classes to create augmented data containing both classes to a certain degree. In [59,60], the authors explain how using mixup data augmentation can greatly benefit adversarial robustness. Examples of MNIST dataset mixup augmentation are shown in Figure 5 below.



Figure 5. Examples of mixup-augmented MNIST dataset (the right and middle images show the original input and the left image shows the output after applying the mixup method).

4. Experimental Results

Experiments were conducted by modifying the AC-GAN setup to generate Unrestricted Adversarial Examples [20]. Because the original authors did not provide the base model, it was necessary to recreate the base model used to generate adversarial examples. The architecture that we used was a simple model with two convolutional layers. When trained on the MNIST dataset, it had a testing accuracy of 97.1%, which was sufficient for our intended purposes. A similar model was used on CelebA, achieving a clean accuracy of 96.4% on gender classification. As explained in the previous section, the previous AC-GAN configuration was modified to boost robustness for adversarially trained models. Table 1 lists the arrangement of each loss function for our model.

Table 1. Loss arrangements between different model versions.

Loss Functions	Unrestricted Adv. Model	Our Model
L_0	Yes	Yes
L_1	Yes	No
L_2	Yes	Yes
$D2D - CE$	No	Yes

Our experiments were run on an NVIDIA Ampere A100 GPU with 80 gigabytes of video RAM using TensorFlow version 2 for both experiments. The Adversarial Robustness Toolbox (ART) by M. Nicolae et al. (2019) [61], a library that provides different attacks and defenses, was used for adversarial defense implementation. Conversely, the CleverHans library proposed by N. Papernot et al. (2018) [62] was used for adversarial attack implementation.

A range of adversarial examples was generated using the AC-GAN model. Because the MNIST and CelebA datasets were used, a combination of each class label as a source, i.e., the number or gender that the image should look like, and a target s , i.e., the misclassification target class, was generated. The same model settings mentioned in [20] were used for the other components of the AC-GAN. Figures 6 and 7 show two generated samples with different sources and targets.

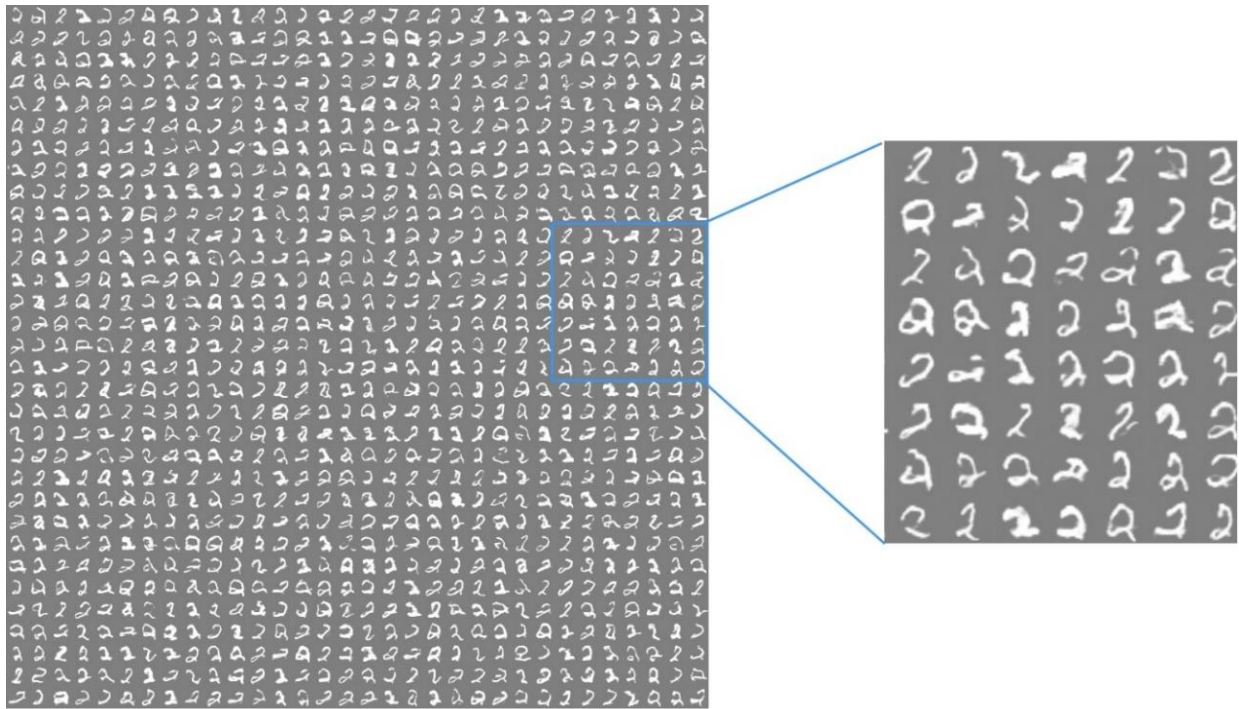


Figure 6. Adversarial samples (source 2 to target 9).

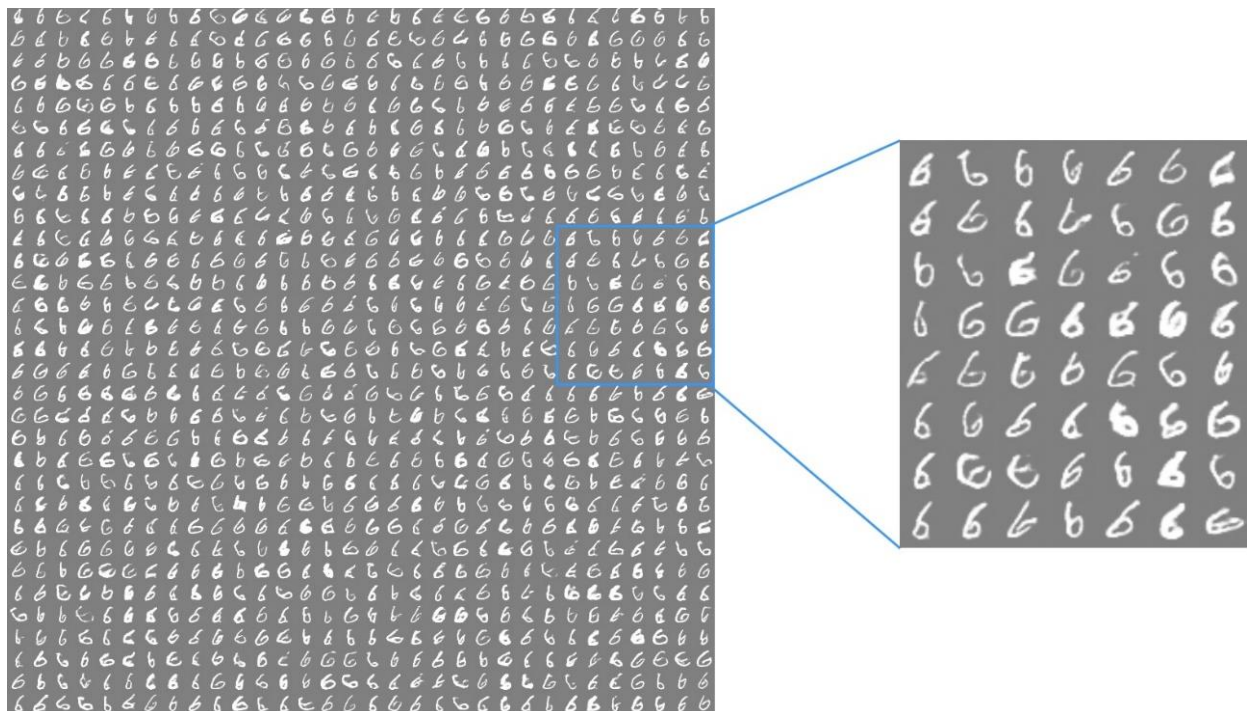


Figure 7. Adversarial samples (source 6 to target 8).

The same model architecture proposed by Madry et al. (2018) [9] was used to perform adversarial training and was compared with different versions of our adversarial training methods. Our models were trained separately on clean MNIST and CelebA datasets and augmented with unrestricted adversarial samples and mixup, such as those shown in the figures above. On the other hand, the model based on Madry et al. (2018) [9] was trained on adversarial images generated using a PGD attack, as recommended in their research paper.

The tests were carried out using FGSM [1], PGD [9], Unrestricted Adversarial Examples [19,20], MIM [36], SPSA [34,35], and BIM [21], as shown in Table 2. Through these experiments, the performance of our proposed model against unseen adversarial examples was measured. To ensure a common ground with which to check the accuracy, the model architecture was kept the same as the one mentioned above for all of the test models while using different adversarial training methods. These different adversarial models were compared in terms of accuracy with the clean models.

Table 2. MNIST accuracy comparison between a clean model, a model trained on PGD images, and models trained on our proposed method.

Attacks	Clean Model	Madry's Model	Unrest. Adv Model	Our Model (w/o Mixup)	Our Model
Clean	97.1%	98.7%	98.5%	98.4%	98.4%
PGD	13.5%	99.9%	98.7%	98.6%	98.3%
FGSM	13.4%	98.8%	98.4%	98.4%	98.8%
Unrestricted	13.9%	91.1%	99.9%	99.1%	99.1%
SPSA	37.7%	76.9%	90.3%	94.4%	99.8%
MIM	4.9%	50.0%	92.9%	90.2%	94.0%
BIM	9.3%	75.5%	89.5%	92.9%	98.5%

Categorical accuracy was used to measure how the predictions from the model matched the true one-hot labels. The average and variance in Table 3 summarize the results across various attacks.

Table 3. MNIST average and variance accuracy comparison between models (showing the variance and average performance of the model accuracy provided in Table 1).

Model	Average	Variance
Madry's Model	83.43%	3×10^{-2}
Unrestricted Adv. Model	94.70%	2×10^{-3}
Proposed Model	97.48%	7.2×10^{-4}

The original unrestricted adversarial model was based on the original AC-GAN architecture to generate Unrestricted Examples [20]. As explained in the methodology section, the original architecture was modified in different ways in order to produce a more robust model. The results in Tables 2–4 show the universality of our proposed model. Even though performance on PGD and FGSM is slightly lower than that of Madry's model or the Unrestricted Adversarial Model, our model largely outperforms them on other attacks, such as SPSA, MIM, and BIM. The accuracy of Madry's model specifically drops by up to half on these attacks. This drop in accuracy is because models that use previously proposed adversarial training methods defend a model from only a small number of inherently similar attacks.

Table 4. CelebA accuracy comparison between a clean model, a model trained on PGD images, and models trained on our proposed method.

Attacks	Clean Model	Madry's Model	Unrest. Adv Model	Our Model (w/o Mixup)	Our Model
Clean	96.4%	95.5%	91.2%	94.2%	96.8%
SPSA	26.4%	70.5%	88.5%	90.5%	91.7%
MIM	10.3%	50.2%	92.9%	94.1%	94.5%
BIM	7.4%	70.3%	88.6%	90.0%	93.1%

On the other hand, it can be shown that the performance improvement of the highest-performing model is due to the removal of the soft-constraint loss L_1 and addition of the $D2D - CE$ loss function proposed in [28]. This modification allows the AC-GAN to search for and find adversarial samples that are impossible to generate using standard methods such as PGD and FGSM. One thing to note here is that the unrestricted adversarial model, which was trained on unrestricted examples based on the original architecture in [20], outperforms our proposed model with regard to unrestricted attacks because the augmented adversarial training samples are in the same set as the attack examples. However, when comparing Madry's model with our model, the overall attack performance across all attacks is better, as shown in Table 3. In the case of the MNIST dataset, our proposed model has performance degradation on clean images; however, when considering the case of the CelebA dataset, its performance is better. There was a performance drop from 98.9% to 98.4% on clean samples when increasing the number of augmented samples, from which it can be speculated that there is a limit on how much augmentation improves the overall model performance without affecting performance on the clean dataset, and that this limit depends on the dataset.

Mixup augmentation, explained in detail in the previous section, is another factor that significantly improved the adversarial robustness of the model. Our findings indicate that mixup augmentation resulted in an average accuracy improvement of 2.13% on the MNIST dataset and 1.82% on the CelebA dataset, which is noticeably superior to not using this method.

The experimental results in Tables 2–4 show gaps in the defenses created by adversarial training that attackers might exploit. Research by H. Zhang et al. (2019) [63] showed that, unless adversarially trained models can cover these types of samples, attackers might be able to exploit such gaps to affect even state-of-the-art models. Our proposed model bridges this gap by boosting performance with less variance in model accuracy and better average performance across all the attacks, as shown in Table 3. Adversarial Training based on AC-GAN Adversarial Example Generation provides defense with more coverage for adversarial samples that might appear further away from the manifold.

In contrast to Madry's model, which was trained using PGD samples, our model employs Unrestricted Adversarial Samples during training. This unique approach grants our model superior defensive coverage owing to the remarkable capabilities of generative models to synthesize a diverse array of samples without any constraints. These generative models have the potential to generate samples that have never been encountered before, enabling our model to exhibit an unparalleled ability to withstand attacks. Our model demonstrates resilience against attacks it has never been exposed to, a notable distinction compared to Madry's model and even the Unrestricted Adversarial Model. This development paves the way for the concept of zero-shot adversarial defense, in which models possess the robustness to defend against adversarial attacks even in cases where the attack types have never been witnessed previously. This is an important step towards creating a robust model, as traditional adversarial defense can be broken by simple attacks that the model was never trained on. By overcoming the limitations of prior training data, our model showcases the potential for better defense in the field of adversarial training and machine learning security.

5. Conclusions

Previous adversarial training methods have produced various results and created adversarial defenses that have improved baseline accuracy. However, as these defense mechanisms improve, adversarial attacks become more resilient to defenses and circumvent them easily, as shown in various research, including [16,17]. This is caused by common adversarial defense methods, specifically, adversarial training restricted by small perturbations added to the original images, which results in "narrower" defensive coverage. Attackers can exploit adversarial samples generated using different algorithms, i.e., by producing adversarial examples not covered by the defenses to use in attacks. Standard

adversarial training methods become less accurate as datasets grow more complex, as discussed in [9,63], further worsening matters.

In this paper, we propose using AC-GAN, which is geared towards adversarial training, to generate adversarial samples that are “unrestricted,” a term taken from [19,20]. This ensures a larger area of defensive coverage, thereby providing universal defense. The AC-GAN architecture in [20] was modified for robust adversarial training. Our experiments demonstrate the robustness of our methods, as detailed in the Results section. From this, it can be concluded that generative models have better potential in finding adversarial samples that are not restricted and that allow for robust adversarial training.

As a future direction, this method can be implemented on datasets with a more complex data manifold, such as ImageNet and CIFAR. Such implementation would allow for a better understanding of conditional generative models such as AC-GANs in order to boost adversarial robustness. Another improvement could involve resolving the issue of a specific Generator being needed for each domain/dataset. The current process is both time- and resource-consuming. To simplify it, using meta-learning to improve the time required to generate adversarial examples could be an option. Future directions might include using Few-Shot Generative Models such as the one mentioned in [64] to generate adversarial samples without creating a generative model from scratch.

Author Contributions: Conceptualization, H.D. and J.K.; Methodology, H.D.; Software, H.D.; Formal analysis, H.D. and J.K.; Data curation, H.D.; Writing—original draft, H.D.; Writing—review & editing, J.K.; Visualization, H.D.; Supervision, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (2021R1A2C2008414) and by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2020-0-01789) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We utilized publicly available MNIST and CelebA datasets in order to train our models. The datasets can be accessed using the following links: <https://kaggle.com/datasets/hojjatk/mnist-dataset> and <https://www.kaggle.com/datasets/jessicali9530/celeba-dataset> (last accessed on 26 July 2020).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572.
2. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2014**, arXiv:1312.6199. [[CrossRef](#)]
3. Lin, Z.; Shi, Y.; Xue, Z. IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection. In *Advances in Knowledge Discovery and Data Mining*; Springer: Cham, Switzerland, 2022; Volume 13282, pp. 79–91.
4. Abdoli, S.; Hafemann, L.G.; Rony, J.; Ayed, I.B.; Cardinal, P.; Koerich, A.L. Universal Adversarial Audio Perturbations. *arXiv* **2020**, arXiv:1908.03173.
5. Carlini, N.; Wagner, D. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. *arXiv* **2018**, arXiv:1801.01944.
6. Kang, D.; Sun, Y.; Brown, T.; Hendrycks, D.; Steinhardt, J. Transfer of Adversarial Robustness Between Perturbation Types 2019. *arXiv* **2019**, arXiv:1905.01034. [[CrossRef](#)]
7. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical Black-Box Attacks against Machine Learning. *arXiv* **2017**, arXiv:1602.02697. [[CrossRef](#)]
8. Wang, W.; Wang, R.; Wang, L.; Wang, Z.; Ye, A. Towards a Robust Deep Neural Network in Texts: A Survey. *arXiv* **2021**, arXiv:1902.07285.
9. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv* **2019**, arXiv:1706.06083.
10. Mahdy, A.M.S.; Lotfy, K.; El-Bary, A.A. Use of optimal control in studying the dynamical behaviors of fractional financial awareness models. *Soft Comput.* **2022**, *26*, 3401–3409. [[CrossRef](#)]

11. Khader, M.M.; Swetlam, N.H.; Mahdy, A.M.S. The Chebyshev Collection Method for Solving Fractional Order Klein-Gordon Equation. *WSEAS Trans. Math.* **2014**, *13*, 31–38.
12. Bai, T.; Luo, J.; Zhao, J.; Wen, B.; Wang, Q. Recent Advances in Adversarial Training for Adversarial Robustness. *arXiv* **2021**, arXiv:2102.01356. [[CrossRef](#)]
13. Huang, R.; Xu, B.; Schuurmans, D.; Szepesvari, C. Learning with a Strong Adversary. *arXiv* **2016**, arXiv:1511.03034. [[CrossRef](#)]
14. Pang, T.; Xu, K.; Du, C.; Chen, N.; Zhu, J. Improving Adversarial Robustness via Promoting Ensemble Diversity. *arXiv* **2019**, arXiv:1901.08846. [[CrossRef](#)]
15. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble Adversarial Training: Attacks and Defenses. *arXiv* **2020**, arXiv:1705.07204. [[CrossRef](#)]
16. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *arXiv* **2018**, arXiv:1802.00420. [[CrossRef](#)]
17. Carlini, N.; Wagner, D. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. *arXiv* **2017**, arXiv:1705.07263. [[CrossRef](#)]
18. A Complete List of All Adversarial Example Papers. Available online: <https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html> (accessed on 14 September 2022).
19. Brown, T.B.; Carlini, N.; Zhang, C.; Olsson, C.; Christiano, P.; Goodfellow, I. Unrestricted Adversarial Examples. *arXiv* **2018**, arXiv:1809.08352.
20. Song, Y.; Shu, R.; Kushman, N.; Ermon, S. Constructing Unrestricted Adversarial Examples with Generative Models. *arXiv* **2018**, arXiv:1805.07894.
21. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2017**, arXiv:1607.02533.
22. Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; Swami, A. The Limitations of Deep Learning in Adversarial Settings. *arXiv* **2015**, arXiv:1511.07528.
23. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661. [[CrossRef](#)]
24. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2014**, arXiv:1312.6114. [[CrossRef](#)]
25. Goodfellow, I. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv* **2017**, arXiv:1701.00160. [[CrossRef](#)]
26. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784. [[CrossRef](#)]
27. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis With Auxiliary Classifier GANs. *arXiv* **2017**, arXiv:1610.09585.
28. Kang, M.; Shim, W.; Cho, M.; Park, J. Rebooting ACGAN: Auxiliary Classifier GANs with Stable Training. *arXiv* **2021**, arXiv:2111.01118. [[CrossRef](#)]
29. Kang, M.; Park, J. ContraGAN: Contrastive Learning for Conditional Image Generation. In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 21357–21369.
30. Miyato, T.; Koyama, M. cGANs with Projection Discriminator. *arXiv* **2018**, arXiv:1802.05637. [[CrossRef](#)]
31. Papers with Code—ArtBench-10 (32 × 32) Benchmark (Conditional Image Generation). Available online: <https://paperswithcode.com/sota/conditional-image-generation-on-artbench-10> (accessed on 11 August 2022).
32. Adversarial Example Using FGSM | TensorFlow Core. Available online: https://www.tensorflow.org/tutorials/generative/adversarial_fgsm (accessed on 13 September 2022).
33. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. *arXiv* **2017**, arXiv:1608.04644.
34. Spall, J.C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control* **1992**, *37*, 332–341. [[CrossRef](#)]
35. Uesato, J.; O’donoghue, B.; Kohli, P.; Oord, A. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. *arXiv* **2018**, arXiv:1802.05666. [[CrossRef](#)]
36. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting Adversarial Attacks with Momentum. *arXiv* **2018**, arXiv:1710.06081. [[CrossRef](#)]
37. Deng, J.; Chen, S.; Dong, L.; Yan, D.; Wang, R. Transferability of Adversarial Attacks on Synthetic Speech Detection. *arXiv* **2022**, arXiv:2205.07711.
38. Xu, Y.; Zhong, X.; Yepes, A.J.; Lau, J.H. Grey-Box Adversarial Attack and Defence For Sentiment Classification. *arXiv* **2021**, arXiv:2103.11576.
39. Le, T.; Wang, S.; Lee, D. MALCOM: Generating Malicious Comments to Attack Neural Fake News Detection Models. *arXiv* **2020**, arXiv:2009.01048. [[CrossRef](#)]
40. Zhang, X.; Zhang, J.; Chen, Z.; He, K. Crafting Adversarial Examples for Neural Machine Translation. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 1967–1977.
41. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Srndic, N.; Laskov, P.; Giacinto, G.; Roli, F. Evasion Attacks against Machine Learning at Test Time. In *Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7908, pp. 387–402.

42. Dalvi, N.; Domingos, P.; Mausam, S.; Sanghai, S.; Verma, D. Adversarial classification. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; Association for Computing Machinery: New York, NY, USA, 2004; pp. 99–108.
43. Lowd, D.; Meek, C. Adversarial learning. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, IL, USA, 21–24 August 2005; Association for Computing Machinery: New York, NY, USA, 2005; pp. 641–647.
44. Martins, N.; Cruz, J.M.; Cruz, T.; Henriques Abreu, P. Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review. *IEEE Access* **2020**, *8*, 35403–35419. [[CrossRef](#)]
45. Experimental Security Research of Tesla Autopilot.pdf. Available online: https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf (accessed on 26 July 2023).
46. Rigaki, M.; Elragal, A. Adversarial Deep Learning Against Intrusion Detection Classifiers. In Proceedings of the ST-152 Workshop on Intelligent Autonomous Agents for Cyber Defence and Resilience, Prague, Czech Republic, 18–20 October 2017.
47. Wang, Z. Deep Learning-Based Intrusion Detection With Adversaries. *IEEE Access* **2018**, *6*, 38367–38384. [[CrossRef](#)]
48. Apruzzese, G.; Colajanni, M.; Ferretti, L.; Marchetti, M. Addressing Adversarial Attacks against Security Systems Based on Machine Learning. In Proceedings of the 2019 11th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 28–31 May 2019; Volume 900, pp. 1–18.
49. Yang, K.; Liu, J.; Zhang, C.; Fang, Y. Adversarial Examples Against the Deep Learning Based Network Intrusion Detection Systems. In Proceedings of the MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 29–31 October 2018; pp. 559–564.
50. Wu, D.; Fang, B.; Wang, J.; Liu, Q.; Cui, X. Evading Machine Learning Botnet Detection Models via Deep Reinforcement Learning. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
51. Balaji, Y.; Goldstein, T.; Hoffman, J. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv* **2019**, arXiv:1910.08051. [[CrossRef](#)]
52. Ding, G.W.; Sharma, Y.; Lui, K.Y.C.; Huang, R. MMA Training: Direct Input Space Margin Maximization through Adversarial Training. *arXiv* **2020**, arXiv:1812.02637. [[CrossRef](#)]
53. Cheng, M.; Lei, Q.; Chen, P.-Y.; Dhillon, I.; Hsieh, C.-J. CAT: Customized Adversarial Training for Improved Robustness. *arXiv* **2020**, arXiv:2002.06789. [[CrossRef](#)]
54. Zhang, W.; Li, D.; Min, X.; Zhai, G.; Guo, G.; Yang, X.; Ma, K. Perceptual Attacks of No-Reference Image Quality Models with Human-in-the-Loop. *arXiv* **2022**, arXiv:2210.00933. [[CrossRef](#)]
55. Yin, X.; Kolouri, S.; Rohde, G.K. GAT: Generative Adversarial Training for Adversarial Example Detection and Robust Classification. *arXiv* **2022**, arXiv:1905.11475.
56. Catak, E.O.; Sivaslioglu, S.; Sahinbas, K. A Generative Model based Adversarial Security of Deep Learning and Linear Classifier Models. *arXiv* **2020**, arXiv:2010.08546. [[CrossRef](#)]
57. Sami, M.; Mobin, I. A Comparative Study on Variational Autoencoders and Generative Adversarial Networks. In Proceedings of the 2019 International Conference of Artificial Intelligence and Information Technology (ICAIT), Yogyakarta, Indonesia, 13–15 March 2019; pp. 1–5.
58. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875. [[CrossRef](#)]
59. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. *arXiv* **2018**, arXiv:1710.09412. [[CrossRef](#)]
60. Zhang, L.; Deng, Z.; Kawaguchi, K.; Ghorbani, A.; Zou, J. How Does Mixup Help with Robustness and Generalization? *arXiv* **2021**, arXiv:2010.04819. [[CrossRef](#)]
61. Nicolae, M.-I.; Sinn, M.; Tran, M.N.; Buesser, B.; Rawat, A.; Wistuba, M.; Zantedeschi, V.; Baracaldo, N.; Chen, B.; Ludwig, H.; et al. Adversarial Robustness Toolbox v1.0.0. *arXiv* **2019**, arXiv:1807.01069. [[CrossRef](#)]
62. Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown, T.; Roy, A.; et al. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv* **2018**, arXiv:1610.00768.
63. Zhang, H.; Chen, H.; Song, Z.; Boning, D.; Dhillon, I.S.; Hsieh, C.-J. The Limitations of Adversarial Training and the Blind-Spot Attack. *arXiv* **2019**, arXiv:1901.04684.
64. Ojha, U.; Li, Y.; Lu, J.; Efros, A.A.; Jae Lee, Y.; Shechtman, E.; Zhang, R. Few-shot Image Generation via Cross-domain Correspondence. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; IEEE: New York City, NY, USA, 2021; pp. 10738–10747.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.