IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Adversarial Attacks and Defenses In Generative AI: A Comparative Analysis

Dr. Sanjay Kumar

Ph.D. Computer Applications, B.R.A. Bihar University

***Abstract*:** This research paper explores the landscape of adversarial attacks and defenses in generative artificial intelligence (AI) systems. It provides an overview of different attack techniques and their impact on AI models. The paper also discusses various defense mechanisms, including adversarial training, randomization techniques, and model monitoring. Evaluating the effectiveness of attacks and defenses is discussed, along with the need for ongoing research to stay ahead of evolving threats. The research aims to enhance understanding of adversarial threats in generative AI and promote the development of secure deployment strategies for these technologies.

Keywords: Artificial Intelligence, Adversarial, Generative, Gradient, Perturbation, Deep Learning.

1. INTRODUCTION

Generative AI refers to a category of artificial intelligence systems that can create new content, such as images, text, audio, or video, based on patterns and relationships learned from training data. These models can generate novel, original content that resembles the training data, enabling a wide range of applications in various industries, such as art, entertainment, and design. Generative AI models can be trained on large datasets and can learn complex patterns and structures within the data [1]. Once trained, they can generate new content by sampling from the learned probability distributions or using techniques like conditional generation, where the model generates content based on specific input conditions. Examples of generative AI models include deep learning-based image generators, such as GANs (Generative Adversarial Networks) and Variational Autoencoders (VAEs), text-based models like GPT-4, and audio and video generation models based on deep learning algorithms [2].

While generative AI has the potential to revolutionize various aspects of our lives, it also raises important ethical and security concerns, such as the potential for misuse, the generation of misleading or biased content, and the risk of adversarial attacks. Adversarial attacks in artificial intelligence (AI) refer to techniques where an attacker deliberately manipulates input data to cause AI models to produce incorrect or unexpected outputs. These attacks exploit vulnerabilities in the learning algorithms used by AI models, specifically deep neural networks (DNNs). As such, ongoing research and development are essential to ensure the safe and responsible deployment of generative AI systems.

2. ADVERSARIAL ATTACKS

The term "adversary" is used in the field of computer security to make a fool or misguide AI based learning model with malicious input. Adversarial attacks are intentional manipulations of input data to deceive AI based models. By adding subtle modifications or noise to the input, attackers can trick the model into making incorrect predictions or classifications [3]. These attacks target vulnerabilities in the model's decision-making process, exploiting its blind spots and weaknesses. Adversarial attacks can have severe consequences in real-world applications such as autonomous driving or malware detection systems. In the figure shown below we can see that how the attacker poisoning the target by sending adversarial input.

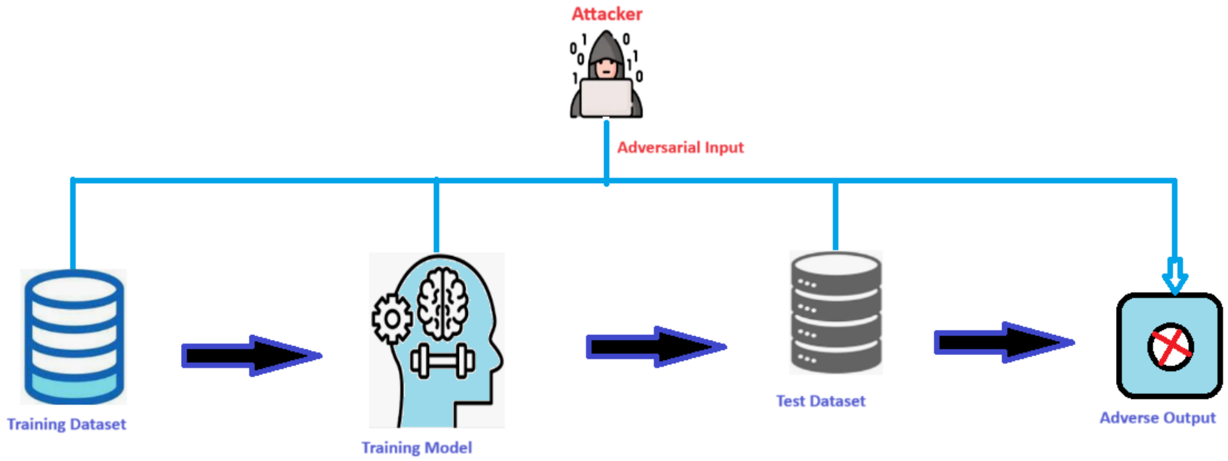

Figure – 1 Adversarial Poisoning

Adversarial attacks have a significant impact on generative artificial intelligence (AI) models, particularly those based on deep learning techniques such as generative adversarial networks (GANs) [4]. These attacks exploit vulnerabilities in the training and inference process of these models, leading to misleading or incorrect generated outputs. Understanding the impact of adversarial attacks is crucial for developing robust defenses and improving the reliability and trustworthiness of generative AI systems [5].

Adversarial attacks work by exploiting the vulnerabilities in AI models, specifically deep neural networks (DNNs), and manipulating input data in such a way that the model produces incorrect or unexpected outputs [6]. The goal of these attacks is to find small perturbations in the input data that are imperceptible to humans but can cause significant changes in the model's behavior.

There are many steps involved in adversarial attacks -

i. Selecting a target: The attacker chooses a specific target for the attack, such as causing misclassification of an image into a specific class or altering the output in a desired way.

ii. Crafting adversarial examples: Adversarial examples are created by introducing perturbations into the original input data. These perturbations can be additive (e.g., adding noise) or subtle modifications to existing features.

iii. Optimization process: The attacker formulates an optimization problem to iteratively find the optimal perturbation that maximizes the likelihood of achieving the desired outcome while minimizing perceptibility. This is often done using gradient-based methods.

iv. Perturbation calculation: By computing gradients with respect to the input data and applying them appropriately, adversarial examples are generated. Common algorithms like Fast Gradient Sign Method (FGSM) compute adversarial examples by taking small steps towards maximizing loss or minimizing accuracy on target labels.

v. Evaluation of success: The crafted adversarial example is then fed into the targeted AI model for evaluation, checking if it produces incorrect outputs or exhibits unexpected behavior compared to the original input.

vi. Iterative refinement: Attackers may refine their approach by iterating through multiple rounds of crafting and evaluating adversarial examples, gradually improving their effectiveness while maintaining imperceptibility constraints.

vii. Transferability across models: Another interesting aspect of adversarial attacks is their transferability across different models trained on similar datasets but with varying architectures or parameters. This means an attack crafted for one network can deceive another network without much effort required for re-crafting an attack specific for each target network configuration.

2.1 Types of Adversarial Attack

- **Gradient-based Attacks**

Gradient-based attacks manipulate the gradients derived from back propagation through the generative model’s layers to create adversarial examples. These attacks exploit the model's sensitivity to small changes in input data by perturbing it in a way that maximizes or minimizes certain characteristics [7][8].

The general steps involved in gradient-based attacks are as follows –

i. Select a target sample: The attacker chooses an input sample on which they want to generate an adversarial example. This sample can be an image, text, or any other data type that the generative model is trained on.

I. Compute gradients: The attacker computes the gradients of the generative model with respect to the loss function. This involves performing back propagation through the model's layers to obtain gradient values for each parameter.

II. Modify input using gradients: The attacker perturbs the original input sample by adding or subtracting a small step along the direction of gradient ascent or descent, depending on whether they want to maximize or minimize certain characteristics (e.g., increase confidence for a specific class, decrease overall probability).

III. Repeat steps 2 and 3 iteratively: Gradient-based attacks often involve multiple iterations of computing gradients and modifying input samples to refine and optimize the adversarial perturbations. This iterative process allows attackers to find more effective manipulations that lead to misclassification or desired outcomes.

IV. Evaluate success of attack: After generating an adversarial example, the attacker evaluates its success by feeding it into the target generative model and checking if it produces a different output than intended. Success rates are measured based on how frequently adversaries can achieve their desired outcomes.

Let's consider an image of digit 7; the attacker adds small distortions to the original image, which results in the model labeling this image as a gibbon, with high confidence as shown in figure 2. The process of adding these distortions is explained below -

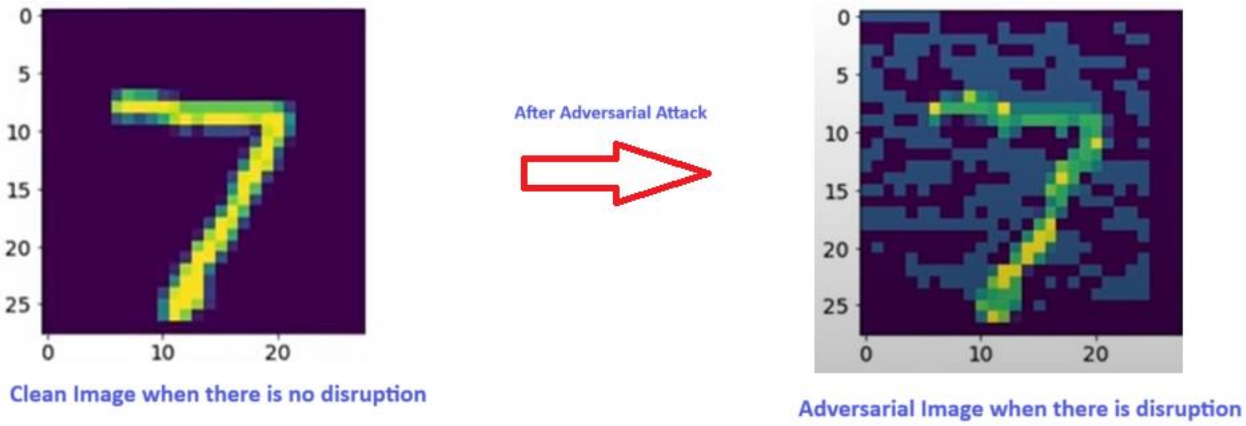

Figure 2 – An Adversarial Image

In this method (Fast Gradient Sign Method - FGSM), for an input image, the method uses the gradients of the loss with respect to the input image to create a new image that maximizes the loss. This new image is called an adversarial image. This can be expressed using below expression.

$$\mathbf{X_{adv} = X + \varepsilon * sign(\triangledown_{x} J\ (\theta, X, Y))}$$

= Original Image Pixels + epsilon * gradient of loss function with respect to the Original Image Pixels

Where X_{adv} - Adversarial Image
X - Original Image
Y - Original Input Label
ε - Multiplier to ensure the distortions is small
θ - Model parameters
J - Loss

Gradients are taken with respect to the input image. This is done because the objective is to create an image that maximizes the loss [9]. A method to accomplish this is to find how much each pixel in the image contributes to the loss value, and add a noise accordingly. This works very fast because it is easy to find how each input pixel contributes to the loss by using the chain rule and finding the required gradients. Hence, the gradients are taken with respect to the image. In addition, since the model is no longer being trained and so the model parameters remain constant. The only goal is to fool an already trained model.

- **Optimization-based Attacks**

Optimization-based attacks aim is to find adversarial perturbations that maximize a specific objective function while maintaining perceptual in distinguishability from the original input sample. These attacks typically involve solving an optimization problem to iteratively find the minimal perturbation that leads to misclassification or achieves the desired outcome.

The general steps involved in optimization-based attacks are as follows-

i. Define the objective function: The attacker defines an objective function that captures their goal, such as maximizing the target class probability or minimizing the distance between a generated sample and a target class.

ii. Set up an optimization problem: The attacker formulates an optimization problem that aims to minimize both the perturbation magnitude and distance from the original data while maximizing or minimizing the objective function defined in step 1. This can involve using techniques like binary search, Lagrange duality, or other optimization algorithms.

iii. Solve the optimization problem: The attacker solves the formulated optimization problem iteratively to find adversarial perturbations that satisfy their goal. This often involves performing gradient-based descent/ascent steps while considering constraints on perturbation magnitude or perceptual changes.

iv. Evaluate success of attack: After generating an adversarial example, the attacker evaluates its success by feeding it into the target generative model and checking if it produces a different output than intended (e.g., misclassification into a specific class). Success rates are measured based on how frequently attackers achieve their desired outcomes.

It is worth noting that different optimization-based attack methods may have variations in their specific techniques, formulations, and algorithms for solving optimization problems [10]. However, they all aim to find minimal adversarial perturbations by optimizing an objective function while ensuring perceptual in distinguishability from original input samples.

- **Decision-Based Attacks**

Decision-based attacks, also known as black-box attacks, do not have access to the internal parameters or gradients of the target generative model. Instead, these attacks rely solely on the model's predictions to iteratively generate adversarial examples.

The general steps involved in decision-based attacks are as follows-

i. Query the target model: The attacker interacts with the target generative model by submitting input samples and receiving their corresponding predictions. This step requires access to the prediction interface of the model without any knowledge of its internal workings.

ii. Generate a surrogate model: The attacker creates a surrogate model that approximates the behavior of the target generative model using a separate set of data. This surrogate can be a simpler and more interpretable model like a decision tree or logistic regression.

iii. Create an optimization problem: The attacker formulates an optimization problem that aims to find adversarial perturbations using only query access to the target generative model and limited information from the surrogate model. The objective is typically defined based on misclassification probability or confidence score differences between classes.

iv. Solve the optimization problem: The attacker solves this optimization problem iteratively by querying both models (target and surrogate) and updating their understanding of how they make decisions based on feedback received from each query.

v. Evaluate success of attack: After generating an adversarial example, the attacker evaluates its success by feeding it into both models and checking if it produces different outputs than intended (e.g., misclassification into a specific class). Success rates are measured based on how frequently adversaries can achieve their desired outcomes despite having limited information about the target generative model.

Decision-based attacks require more interactions with the targeted system compared to gradient-based or optimization-based attacks since they lack direct access to gradients or parameters [11]. These attacks highlight potential vulnerabilities in AI systems where adversaries can manipulate input samples without detailed knowledge about how models internally operate.

It's important to note that attackers don't necessarily have full knowledge about model internals or access rights; they leverage gradient information obtained from running inference rather than relying on complete access to training data.

The success of these attacks highlights weaknesses and vulnerabilities in DNNs, particularly their sensitivity towards small changes in input data space. The underlying reasons behind this vulnerability include linear nature of DNNs' decision boundaries, over-reliance on certain features, distributional shifts, and non-robust representations among others.

3. WAYS TO DEFEND ADVERSARIAL ATTACK

Defending against adversarial attacks in generative AI involves implementing various techniques to enhance the robustness and security of the models. Here are some strategies that can be used for defense [12][13].

i. Adversarial training: One common defense technique is to incorporate adversarial examples during the training process. By augmenting the training data with carefully crafted adversarial samples, the model learns to be more resilient to such attacks. This helps improve generalization and reduces the model's vulnerability to adversarial perturbations.

ii. Defensive distillation: Defensive distillation involves training a distilled model, where the outputs of a large and accurate model (teacher) are used as soft targets for training a smaller model (student). This technique can help make models more resistant to adversarial attacks by reducing their sensitivity to small input perturbations.

iii. Randomization: Adding random noise or transformations during both training and inference can make it harder for attackers to generate effective adversarial examples. Randomizing inputs at different stages of processing, such as augmenting images with random rotations or translations, can disrupt attack methods relying on specific input patterns.

iv. Gradient masking: By limiting access to gradients or obfuscating them, gradient-based attacks like FGSM (Fast Gradient Sign Method) become less effective. Techniques like defensive distillation and adding noise during gradient computation can hamper attackers' ability to compute effective gradients for crafting adversarial perturbations.

v. Ensemble methods: Training multiple models independently and combining their predictions through voting or averaging can improve robustness against adversarial attacks. Attackers would need to simultaneously fool multiple models, making it more challenging for them.

vi. Input verification: Employing input verification techniques that analyze the properties of incoming samples before they are processed by the generative AI system can help detect potential malicious inputs or identify samples that deviate significantly from expected distributions.

vii. Model monitoring and retraining: Regularly monitoring models in production environments allows detecting any changes in performance caused by potential adversaries' efforts promptly. If an attack is identified, retraining models using updated defenses or additional data may help mitigate vulnerabilities.

viii. Adversary-aware evaluation metrics: Traditional evaluation metrics might not fully capture how well a generative AI system performs under adversarial conditions. Developing adversary-aware measures that consider worst-case scenarios or specific attack strategies provides additional insights into system vulnerabilities.

It's important to note that no defense mechanism is entirely foolproof, and new attack techniques continue to emerge as adversaries adapt their methodologies. Therefore, employing a combination of these defense strategies alongside ongoing research into mitigating vulnerabilities is crucial for enhancing security in generative AI systems.

4. CONCLUSION

As generative AI continues to advance and find applications across various domains, addressing the issue of adversarial attacks becomes increasingly important. By understanding the landscape of these attacks and implementing effective defense mechanisms informed by ongoing research efforts - we can ensure a more secure deployment and utilization of generative AI technologies. Adversarial attacks have significant implications on generative artificial intelligence systems like GANs by compromising their performance, reliability and trustworthiness. Adversaries exploit vulnerabilities within these systems, resulting degradation in output quality, fidelity and diversity. Defending against these challenges involves developing specific defense mechanisms such as robust training methodologies, adversarial-trained discriminators, regularization techniques, ensemble methods and preprocessing techniques. These defenses help improve resilience, reliability and accuracy while minimizing susceptibility towards malicious manipulations. Striking a balance between generating high-quality outputs while ensuring security is an ongoing research challenge but critical for leveraging the full potential of generative AI in various real-world applications.

5. REFERENCES

[1] https://www.cmu.edu/intelligentbusiness/expertise/genai-principles.pdf

[2] https://www.semanticscholar.org/reader/ac675900f7c6c14c8488e09a2a6e8525bcf9d45a

[3]https://medium.com/@yashgaherwar2002/adversarial-machine-learning-attacks-preventions-640c5ffc2404

[4] https://ietresearch.onlinelibrary.wiley.com/doi/epdf/10.1049/cit2.12028

[5]https://www.sciencedirect.com/science/article/pii/S2214212620308607/pdfft?md5=8d371c6eac7edb1588042bdcaf0842d6&pid=1-s2.0-S2214212620308607-main.pdf

[6] https://engineering.purdue.edu/ChanGroup/ECE595/files/chapter3.pdf

[7]https://medium.com/@zachariaharungeorge/a-deep-dive-into-the-fast-gradient-sign-method-611826e34865

[8] https://defence.ai/ai-security/gradient-based-attacks/

[9] https://arxiv.org/pdf/2303.06302

[10] https://dl.acm.org/doi/pdf/10.1145/3128572.3140448

[11] https://openreview.net/pdf?id=SyZI0GWCZ

[12] https://imerit.net/blog/four-defenses-against-adversarial-attacks-all-una/

[13]https://openaccess.thecvf.com/content_cvpr_2018/papers/Liao_Defense_Against_Adversarial_CVPR_2018_paper.pdf