Science and Technology 2023, 13(1): 6-11
DOI: 10.5923/j.scit.20231301.02

Securing AI Systems from Adversarial Threats

Enoch Anbu Arasu Ponnuswamy

USA

Abstract Adversarial attacks on artificial intelligence (AI) systems have become an increasingly concerning issue in recent years. These attacks involve intentionally crafted input data to deceive or manipulate the output of the targeted AI model. Adversarial attacks have been demonstrated across a wide range of AI applications, including image and speech recognition, natural language processing, and autonomous vehicles. Such attacks can have significant impacts, from compromising the security of sensitive systems to causing physical harm in critical applications. In this article, we explore the concept of adversarial attacks on AI systems, provide examples of real-world attacks, and discuss the impacts of such attacks. We also examine some of the common defense strategies against adversarial attacks, as well as emerging methods for improving the robustness of AI models.

Keywords Securing AI systems from adversarial threats

1. Adversarial Attacks on AI systems

Adversarial AI attacks present a growing security challenge to the use of Artificial Intelligence (AI) and machine learning (ML) in businesses. These attacks, also known as Adversarial ML, deceive AI models by providing them with false data. This form of cyberattack has mostly been observed in image classification and spam detection.

The risks posed by adversarial AI attacks include:

- Social media engineering: This type of attack automatically "scrapes" user profiles on social media platforms and creates automated content to attract the target profile.
- Deepfakes: Used primarily in banking fraud, deep fakes can cause harm through blackmail and extortion, damaging an individual's credibility, document fraud, and social media harassment.
- Malware hiding: Threat actors use ML to hide malware within seemingly normal traffic, making it difficult for regular users to detect.
- Passwords: Adversarial AI attacks can analyze a large number of passwords and generate variations, making password cracking more efficient. Additionally, ML can be used to solve CAPTCHAs.

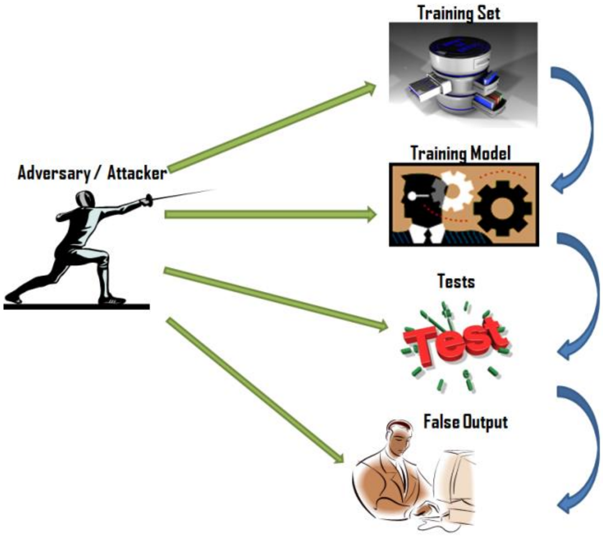
According to Alexey Rubtsov of the Global Risk Institute, adversarial machine learning takes advantage of the vulnerabilities and unique characteristics of ML models. This can result in incorrect decisions by ML-based trading algorithms, among other risks.

2. Examples of Adversarial AI Attacks

Adversarial AI attacks allow hackers to manipulate AI models into making incorrect decisions. This is accomplished by introducing a corrupted version of a legitimate input to the model.

There are various techniques used in adversarial AI attacks, some of which include:

- FastGradient Sign method (FGSM)
- Projected Gradient Descent (PGD)
- Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS)
- JSMA
- Deepcool
- Basic Iterative Method (BIM)
- Evolutionary algorithms
- Generative Adversarial Networks (GANs)



* Corresponding author:
enochcog@gmail.com (Enoch Anbu Arasu Ponnuswamy)

Received: Feb. 15, 2023; Accepted: Feb. 26, 2023; Published: Feb. 28, 2023
Published online at http://journal.sapub.org/scit

Each method has its strengths and weaknesses, and the choice of technique will depend on the specific use case and the desired outcome of the attack. However, regardless of the method used, the goal of adversarial AI attacks is to trick the AI model into making mistakes, which can have serious consequences in fields such as finance, healthcare, and national security.

1. FastGradient Sign method (FGSM)

The FastGradient Sign method (FGSM) is a popular and effective technique for adversarial attacks on deep learning models. It is a one-step gradient-based attack method that was introduced in the paper "Explaining and Harnessing Adversarial Examples" by Ian Goodfellow et al. FGSM is a white-box attack, meaning the attacker has complete knowledge of the model architecture and parameters.

The goal of FGSM is to add a small perturbation to the input data so that the model's prediction changes to a desired incorrect label. The method achieves this by computing the gradient of the loss function concerning the input data and using this gradient to create an adversarial example.

The sign of the gradient is used to determine the direction in which the input should be perturbed to maximize the loss. The magnitude of the perturbation is determined by a hyperparameter called the step size.

The FGSM attack can be formulated as follows: given an input x, its corresponding label y, a deep learning model f(x), and a loss function L(y, f(x)), the adversarial example x' is calculated as:

x' = x + epsilon * sign(grad_x L(y, f(x)))

where epsilon is the step size and sign(grad_x L(y, f(x))) is the sign of the gradient of the loss concerning the input x.

The FGSM attack is highly effective in fooling deep learning models. It is fast and easy to implement, making it a popular choice for both researchers and attackers. However, the FGSM attack is limited in that it only considers one-step perturbations and does not account for the non-linearity of deep learning models. To overcome these limitations, researchers have proposed variations of the FGSM attack, such as the Basic Iterative Method (BIM) and the Projected Gradient Descent (PGD) method.

The FGSM attack is a powerful tool for both researchers and attackers in the field of adversarial machine learning. It provides a simple and effective method for creating adversarial examples and has motivated further research into more sophisticated and robust adversarial attacks.

2. Basic Iterative Method (BIM)

The basic Iterative Method (BIM) is a widely used technique in adversarial AI attacks. The goal of BIM is to modify the inputs to a machine learning model in such a way that the model's predictions are perturbed. This is achieved by making small modifications to the inputs in multiple iterations until the desired level of misclassification is achieved.

BIM is an effective technique for fooling deep neural networks, which are widely used in a variety of applications, including image classification, speech recognition, and natural language processing. By using BIM, an attacker can cause the model to misclassify inputs with high confidence, even when the modifications made to the inputs are small and imperceptible to humans.

The basic idea behind BIM is to perform gradient descent on the loss function of the model, to increase the loss for the target class. This is achieved by updating the inputs in the direction that maximizes the loss. The number of iterations and the step size can be adjusted to control the magnitude of the perturbation and the rate of convergence.

The Basic Iterative Method (BIM) is a powerful tool for adversarial AI attacks, which can cause deep neural networks to make incorrect predictions with high confidence. It is important to be aware of the potential security implications of BIM and to take measures to protect against this type of attack.

3. Projected Gradient Descent (PGD)

Projected Gradient Descent (PGD) is a widely used technique for adversarial machine learning. It is a type of gradient-based attack that perturbs the inputs of a machine-learning model to cause misclassification. PGD works by iteratively modifying the inputs in the direction of the gradients of the loss function, to maximize the loss for the target class.

PGD is a powerful technique for adversarial attacks, as it can cause deep neural networks to make incorrect predictions with high confidence. It is also more effective than other gradient-based attacks, such as Fast Gradient Sign Method (FGSM), as it allows for more control over the magnitude of the perturbations. This makes it easier to find the optimal perturbations for a specific target model and target class.

The PGD algorithm is relatively simple to implement and can be applied to a wide range of machine learning models, including image classifiers, speech recognition systems, and natural language processing models. The success of PGD attacks is largely dependent on the choice of the perturbation size and the number of iterations, which can be adjusted to balance the trade-off between the strength of the attack and the imperceptibility of the perturbations. The Projected Gradient Descent (PGD) is a powerful and effective technique for adversarial machine learning. It is important to be aware of the potential security implications of PGD attacks and to take measures to protect against this type of attack. This can include regular evaluation of the model's robustness and the use of adversarial training to increase the model's robustness against PGD attacks.

4. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a type of deep learning model that has become increasingly popular in recent years. They are designed to generate new, synthetic data samples that are similar to a given training dataset. A GAN consists of two components: a generator and a discriminator. The generator is responsible for producing synthetic samples, while the discriminator is trained to differentiate between the synthetic samples and the real samples from the training dataset. The two components are trained together in a competition, with the generator trying to produce samples that are indistinguishable from the real samples, and the discriminator tries to correctly identify the synthetic samples.

The training process continues until the generator produces samples that are of sufficient quality to fool the discriminator. At this point, the generator can be used to generate new samples that are similar to the training data. These synthetic samples can be used for a wide range of applications, including data augmentation, imputing missing data, and generating new images, videos, audio, or text. GANs have been used for a wide range of applications, including image synthesis, style transfer, and super-resolution. They have also been used in the generation of realistic-looking images and videos, as well as in the creation of synthetic training data for other machine-learning models.

However, GANs are not without their challenges. One of the biggest challenges is stability during training, as the generator and discriminator can easily get stuck in a local optimum. Another challenge is mode collapse, where the generator only produces a limited set of outputs, rather than a diverse set of samples. Generative Adversarial Networks (GANs) are a powerful and flexible deep learning model for generating synthetic data. They have a wide range of potential applications, including image synthesis, data augmentation, and the creation of synthetic training data. Despite the challenges, GANs are an important area of research and have the potential to have a major impact on many areas of computer science and engineering.

5. Evolutionary algorithms

Evolutionary algorithms are a type of optimization algorithm inspired by the process of natural selection and evolution. They are designed to find solutions to complex optimization problems by mimicking the process of evolution. An evolutionary algorithm begins with a population of candidate solutions, also known as individuals or chromosomes. These solutions are evaluated based on their fitness, or how well they meet the objectives of the optimization problem. The fit individuals are then selected and recombined to form new individuals, to create even fitter solutions.

This process of selection, recombination, and mutation continues over multiple generations, with the fittest individuals being selected and recombined to form new individuals in each generation. The goal is to find the global optimum, or the best possible solution, by continuously refining and improving the population of individuals. There are several types of evolutionary algorithms, including genetic algorithms, differential evolution, and particle swarm optimization. These algorithms differ in the way they select and recombine individuals, as well as the way they apply mutations. Evolutionary algorithms have been applied to a wide range of optimization problems, including multi-objective optimization, constraint optimization, and combinatorial optimization. They are particularly useful for problems where the objective function is non-linear, complex, or unknown, as they do not require any assumptions about the structure of the objective function.

Despite their success, evolutionary algorithms also have some limitations. One of the main limitations is that they are often slow and computationally expensive, especially for large-scale problems. They can also be sensitive to the choice of parameters, such as the population size and mutation rate, which can impact the quality of the solutions found. The evolutionary algorithms are a powerful optimization method inspired by the process of natural selection and evolution. They have been successfully applied to a wide range of optimization problems and have the potential to find high-quality solutions to complex problems. However, they are computationally expensive and can be sensitive to the choice of parameters, making them best suited for problems where other optimization methods are not feasible.

6. Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS)

Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) is an optimization algorithm used for solving large-scale optimization problems. It is an iterative method that seeks to find the optimal solution to a problem by minimizing an objective function. L-BFGS is a quasi-Newton method, meaning that it uses an approximation of the Hessian matrix to determine the search direction in each iteration. Unlike other quasi-Newton methods, L-BFGS uses limited memory to store information about the approximation, hence its name "Limited-memory". This allows it to handle large-scale optimization problems that may not fit into memory, making it an efficient and effective optimization method for many real-world applications. The L-BFGS algorithm works by using gradient information to iteratively improve the approximation of the Hessian matrix. The approximation is updated based on the gradient information from the current iteration and previous iterations. The search direction is then determined using this approximation and used to update the solution. The process continues until convergence, or until a stopping criterion is met.

One of the main advantages of L-BFGS is its efficiency and low memory requirements. The limited memory approach allows it to handle large-scale optimization problems that may not fit into memory, making it a popular choice for many real-world applications. L-BFGS is also well-suited for optimization problems with noisy gradient information, as it can handle this noise effectively and still converge to a high-quality solution. Despite its strengths, L-BFGS also has some limitations. It may not always converge to the global optimum, and its performance can be sensitive to the choice of parameters, such as the initial guess, the stopping criterion, and the approximation of the Hessian matrix. Additionally, it is not always the best method for problems with sparse or non-smooth gradient information.

The L-BFGS is a powerful optimization algorithm that has proven to be effective and efficient for large-scale optimization problems. Its limited memory approach and ability to handle noisy gradient information make it a popular choice for many real-world applications. However, it may not always converge to the global optimum and its performance can be sensitive to the choice of parameters, making it best suited for problems where other optimization methods are not feasible.

7. Deepcool

Deepcool is a highly effective form of attack that produces adversarial AI examples with high misclassification rates and fewer perturbations. This untargeted adversarial sample generation technique reduces the Euclidean distance between the original and perturbed samples.

8. JSMA

On the other hand, JSMA is a feature selection-based method where hackers minimize the number of modified features to cause misclassification. In this attack, flat perturbations are iteratively added to the features based on their saliency values in decreasing order.

These are just a few examples of Adversarial AI, and it is important for businesses to understand the methods used and how to defend against them. Protecting AI systems from adversarial threats requires a combination of techniques such as adversarial training, model ensembles, and robust optimization.

3. Adversarial Attacks and Their Impacts

The discovery of adversarial examples highlights the limitations and fragility of Deep Neural Networks (DNNs). It shows that DNNs don't necessarily understand the concept of an object like a human does, which raises questions about their reliability and trustworthiness in real-world applications.

Moreover, the vulnerability of DNNs to adversarial attacks could be exploited by malicious actors to intentionally mislead the system. Researchers have demonstrated several potential scenarios where adversarial attacks can have serious consequences, such as:

Misleading self-driving cars: Eykholt et al. developed a technique to create artificial "graffiti" that can confuse DNNs typically used in self-driving cars into misinterpreting traffic signs. This could potentially cause a self-driving car to crash.

Making a person invisible to security cameras: Thys et al. created "adversarial patches" that can fool the YOLOv2 convolutional neural network, which is used for visual monitoring. The adversarial patch makes the DNN unable to detect a person in the image.

False transcriptions of voice recordings: Cisse et al. carried out a black box attack on the Google Voice voice-to-text application by creating adversarial audio samples. The authors established that human subjects couldn't distinguish between the original audio and the adversarial audio, and when using Google Voice, the system created false transcriptions of the adversarial audio that lost their meaning.

The implications of adversarial attacks are far-reaching, and they show the importance of developing robust and secure DNNs to prevent potential harm in real-world applications.

4. How to Defend against Adversarial AI Attacks?

Defending against Adversarial AI attacks requires a comprehensive approach to machine learning management. Organizations need to adopt MLOps, a set of best practices that combine the benefits of machine learning, DevOps, and data engineering.

To start, organizations must conduct a risk assessment of all ML implementations and assign ownership and governance of all ML-related content and operations. New security policies specifically designed to defend against Adversarial AI attacks must also be established.

The implementation of MLOps follows an iterative and incremental process, starting with the design of ML use cases, prioritizing them based on requirements engineering, and checking the data availability. The next step is the development of the ML model, which involves data engineering and model engineering. Model testing and validation are crucial to ensure the accuracy of the model.

The final step is the deployment of the ML model and the development of a CI/CD pipeline for operational monitoring and triggering. This helps organizations to constantly monitor their ML models and detect any breaches in the ML defense, keeping their systems secure from Adversarial AI attacks.

5. Effective Methods for Defending against Adversarial Attacks

The discovery of adversarial examples has revealed a vulnerability in Deep Neural Networks (DNNs) that makes them susceptible to misclassification. Adversarial examples are artificially crafted inputs that are designed to mislead a DNN, causing it to make an incorrect prediction. This vulnerability is not just a theoretical concern but has real-world implications, including the potential for malicious actors to use adversarial attacks to intentionally mislead DNNs for nefarious purposes.

To defend against adversarial attacks, researchers have developed a variety of adversarial training methods aimed at endowing DNNs with more robust models that are less vulnerable to adversarial attacks. Here, we will refer four of these methods: Training Data Augmentation, Regularization, Optimization, and Distillation.

1. Training Data Augmentation

One approach to defending against adversarial attacks is to generate adversarial examples and add them to a DNN's training data. This process is known as Training Data Augmentation. The idea is that by exposing the DNN to adversarial examples during training, the model will learn to recognize and classify these examples correctly, making it less susceptible to adversarial attacks.

2. Regularization

Another approach to defending against adversarial attacks is Regularization. This method involves adding terms in the training loss function to steer gradient descent toward parameters that are more resistant to adversarial attacks. One common form of regularization is weight decay, which penalizes large weight values and helps prevent overfitting. Another is a dropout, which randomly sets a portion of the network's activations to zero during training, effectively reducing the number of features the network uses to make predictions.

3. Optimization

A third approach to defending against adversarial attacks is Optimization. In this method, the training objective is changed from minimizing the loss to minimizing the maximum loss achievable by an adversarial attack. This forces the DNN to learn a more robust representation of the data that is less susceptible to adversarial attacks.

4. Distillation

The final method we will discuss is Distillation. This method involves training a smaller, more compact model to mimic the predictions of a larger, more complex model. The smaller model is then used to make predictions, effectively reducing the size and complexity of the network and making it less susceptible to adversarial attacks.

In supposition, the development of adversarial attacks has raised concerns about the security and robustness of DNNs. To defend against these attacks, researchers have developed a variety of adversarial training methods, including Training Data Augmentation, Regularization, Optimization, and Distillation. By combining these methods, organizations can create more robust DNNs that are less susceptible to adversarial attacks and ensure the security of their ML systems.

6. The Future Scope of Securing AI Systems from Adversarial Threats

One of the critical areas of focus for securing AI systems is to increase their robustness. Adversarial attacks typically exploit vulnerabilities in AI models, making them behave in unexpected and potentially dangerous ways. By developing more robust AI models, future developments in AI can make systems more resistant to attacks and improve their ability to detect and defend against adversarial threats.

Machine learning is a powerful tool that can be used to enhance security in AI systems. It can analyze vast amounts of data and identify patterns and anomalies that could be indicative of potential threats. By using machine learning algorithms, organizations can improve their ability to protect their AI systems by detecting potential vulnerabilities and developing more effective defense mechanisms.

Another critical aspect of securing AI systems is to develop ethical considerations for their development. The use of AI can have significant implications for society, and it is essential to ensure that the development of AI systems is done in an ethical and responsible manner. By considering ethical concerns such as data privacy, bias, and fairness, organizations can create AI systems that are more trustworthy and reliable. Collaboration between industry, academia, and government is also essential for securing AI systems from adversarial threats. Each sector brings unique perspectives and expertise to the table, and by sharing information and knowledge, organizations can create a more secure and robust AI ecosystem. Collaboration can also help to identify potential threats and develop more effective defense mechanisms.

In short, securing AI systems from adversarial threats is critical for the responsible integration of AI into modern technology. The future scope of securing AI systems involves increasing their robustness, using machine learning to enhance security, developing ethical considerations, and collaboration between different sectors. By adopting a proactive and multi-pronged approach, organizations can better protect their AI systems and ensure the safe and responsible use of AI in society.

7. Conclusions

As AI systems become increasingly integrated into our daily lives, the risk of adversarial attacks continues to grow. The examples of adversarial attacks on AI systems demonstrate that these attacks can have serious consequences, including security breaches, data theft, and physical harm. In order to effectively defend against such attacks, it is crucial to employ a combination of strategies that address both the technical vulnerabilities of AI systems and the social and ethical considerations surrounding their use. The most effective methods for defending against adversarial attacks involve a combination of proactive measures, such as building AI models that are inherently resistant to attacks, and reactive measures, such as developing techniques to detect and mitigate attacks in real-time. While there is no foolproof defense against adversarial attacks, continued research and development in this area will be essential to improving the security and reliability of AI systems in the future.

REFERENCES

[1] Shengjie Xu; Yi Qian; Rose Qingyang Hu, "Cybersecurity in the Era of Artificial Intelligence," in *Cybersecurity in Intelligent Networking Systems*, IEEE, 2023, pp.1-16, doi: 10.1002/9781119784135.ch1.

[2] https://link.springer.com/article/10.1007/s43681-021-00113-9

[3] https://cybersecurity.springeropen.com/articles/10.1186/s42400-018-0012-9

[4] https://ieeexplore.ieee.org/document/9827763

[5] https://ieeexplore.ieee.org/document/9927611/authors#authors

[6] https://ieeexplore.ieee.org/document/9433761

[7] https://ieeexplore.ieee.org/document/10030535

[8] https://www.sciencedirect.com/science/article/abs/pii/S095741742200272X#preview-section-snippets

[9] https://cybersecurity.springeropen.com/articles/10.1186/s42400-019-0027-x

[10] https://eudl.eu/pdf/10.4108/eai.7-7-2021.170285

[11] Detection based Defense against Adversarial Examples from the Steganalysis Point of View (Jiayang Liu, Weiming Zhang, Yiwei Zhang, Dongdong Hou, Yujia Liu, Hongyue Zha, Nenghai Yu).

[12] Adversarial Attacks and Defenses: An Interpretation Perspective Ninghao Liu, Mengnan Du, Ruocheng Guo‡, Huan Liu, Xia Hu.

[13] https://towardsdatascience.com/limited-memory-broyden-fletcher-goldfarb-shanno-algorithm-in-ml-net-118dec066ba

[14] https://www.ibm.com/blogs/research/2018/04/ai-adversarial-robustness-toolbox/

[15] https://skimai.com/blog-what-is-an-adversarial-ai-attack/

[16] https://blog.f-secure.com/5-adversarial-ai-attacks-that-show-machines/

[17] https://www.netapp.com/blog/using-AI-to-increase-security-public-sector/

[18] https://www.infoworld.com/article/3215130/how-to-prevent-hackers-ai-apocalypse.html

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

[20] https://thenextweb.com/news/how-to-protect-your-ai-systems-against-adversarial-machine-learning-syndication