Analysis of Adversarial Attacks on AI-based With Fast Gradient Sign Method

Sigit Wibawa
Department of Electrical Engineering, Universitas Bina Sarana Informatika, Jakarta, Indonesia

Abstract: Artificial intelligence (AI) has become a key driving force in sectors from transportation to healthcare, and is opening up tremendous opportunities for technological advancement. However, behind this promising potential, AI also presents serious security challenges. This article aims to investigate attacks on AI and security challenges that must be faced in the era of artificial intelligence, this research aims to simulate and test the security of AI systems due to adversarial attacks. We can use the Python programming language for this, using several libraries and tools. One that is very popular for testing the security of AI models is CleverHans, and by understanding those threats we can protect the positive developments of AI in the future. this research provides a thorough understanding of attacks in AI technology especially in neural networks and machine learning, and the security challenge we face is that adding a little interference to the input data causes the AI model to produce wrong predictions in adversarial attacks there is the FGSM model which with an epsilon value of 0.1 causes the model suffered a drastic reduction in accuracy of around 66%, which means that the attack managed to mislead the model and lead to incorrect predictions. in the future understanding this threat is the key to protecting the positive development of AI. With a thorough understanding of AI attacks and the security challenges we address, we can build a solid foundation to effectively address these threats.

Keywords*:* Adversarial, FGSM, Artificial intelligence, CleverHans, Mnist Dataset

Introduction

Artificial intelligence (AI) has been one of the most amazing innovations of this century, driving a technological revolution that has changed the way we live, work, and interact. With its ability to recognize patterns, make decisions, and learn from experience, AI has infiltrated almost every aspect of human life (Carlini & Wagner, 2016). However, the popularity and acceptance of AI also present serious challenges, especially in terms of security (Goodfellow et

Correspondents Author:
Sigit Wibawa, Department of Electrical Engineering, Universitas Bina Sarana Informatika, Jakarta, Indonesia
Email : sigit.stb@bsi.ac.id

Received: June 21, 2023; Accepted: July 29, 2023; Publication: August 1, 2023.

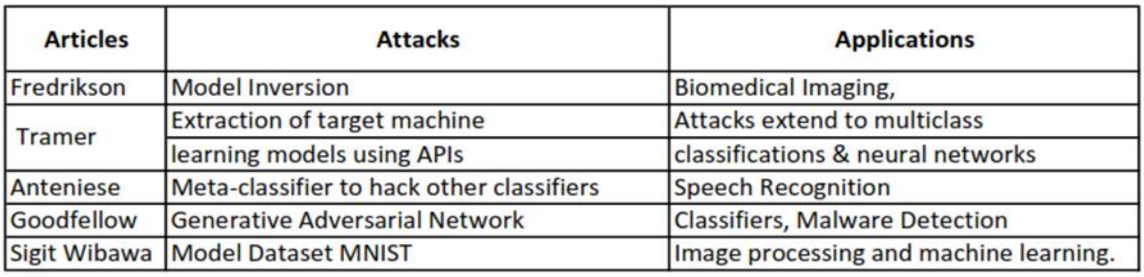
al., 2014) This article is intended to understand and discuss attacks on AI and the security challenges that need to be addressed in the era of artificial intelligence.

From previous research, various information has been obtained regarding attacks on AI, and following the latest research on security in artificial intelligence (AI) is a very important and growing issue (Athalye et al., 2018) Security attacks on artificial intelligence cover a wide range of threats and risks to AI systems and their applications. The following are some forms of security attacks that can occur on AI systems figure 1.

1	Adversarial Attacks	Adding small perturbations to input data causes the AI model to produce incorrect predictions.
2	Model Inversion Attack	Using model predictions to reconstruct training data or sensitive information used to train the model.
3	Model Exploitation	Exploiting the AI model to gain unauthorized access to information that should be restricted. For instance, obtaining predictions without knowledge of the actual input data.
4	Event-based Attack	Manipulating or altering input data at or before the time the model is used creates adverse impacts.
5	Model Unfairness Attack	Manipulating the AI model to produce unfair or discriminatory predictions against specific groups
6	Vulnerability to Unseen Input Data	When the AI model is vulnerable to unusual or unseen data that differs from the training data
7	Model Theft	Stealing an AI model that has been trained with significant investment by others and claiming it as one's own.
8	Model Evasion Attack	An attack in which the adversary attempts to send data that has been learned previously and successfully evades detection as an attack by the AI system.

Figure 1 Security attacks on AI systems

Of the many attack concepts that occur in artificial intelligence (Hartmann & Steup, 2020) First of all, this article will explore the concept of adversarial attacks, in which an attacker can manipulate input data to trick the AI system and produce erroneous outputs (Li et al., 2021) We will analyze examples of adversarial attack cases on different types of AI models, such as images, text, and sound. This article will discuss the concept of an adversarial attack, in which an attacker uses multiple techniques to manipulate input data and make the AI model produce incorrect predictions. In the era of AI, understanding adversarial attacks is becoming increasingly important, especially given the potential damage they can cause to systems of security and public trust. There are several focuses on preference studies which are stated in the form of gap analysis, and the objectives of several previous studies refer to the same type of attack but different objects and applications described in the following figure 2.



Articles	Attacks	Applications
Fredrikson	Model Inversion	Biomedical Imaging,
Tramer	Extraction of target machine learning models using APIs	Attacks extend to multiclass classifications & neural networks
Anteniese	Meta-classifier to hack other classifiers	Speech Recognition
Goodfellow	Generative Adversarial Network	Classifiers, Malware Detection
Sigit Wibawa	Model Dataset MNIST	Image processing and machine learning.

Figure 2 Gap Analysis

Adversarial attacks are techniques in which an attacker intentionally manipulates input data fed into an artificial intelligence (AI) model to cause prediction errors or cause the model to produce unexpected outputs. This attack seeks to find loopholes in the AI model that can be exploited by adding small perturbations to the input data, which are often invisible to humans but can cause drastic changes to the model results. Adversarial attack techniques are based on exploiting the vulnerabilities or weaknesses of the AI model. Even though an AI model may be well trained and have high performance on training data, an adversarial attack can cause the model to fail correctly or trick the model into giving wrong predictions.

There are several common types of adversarial attacks, including:

1. Fast Gradient Sign Method (FGSM) Attack (Xu et al., 2019) This attack uses the gradient of the model's cost function to determine the direction in which the input data needs to be modified resulting in erroneous predictions. FGSM tends to be a relatively simple attack but is quite effective.
2. Resistance Gradient Projection Attack (PGD): This attack is a variation of FGSM that repeatedly applies FGSM attacks to generate more powerful distractions and harden the model for resistance to attacks.
3. Targeted Search Attack (Targeted Attack): This attack aims to make the AI model produce certain predictions desired by the attacker. The attacker looks for disturbances in the input data to steer the model in the desired direction.
4. Generative Attacks: These attacks involve using generative models, such as Generative Adversarial Networks (GANs), to generate input data that is controlled by the attacker and causes undesired output.

The information above is just a few examples of attacks where we chose the Fast Gradient Sign Method (FGSM) Attack as an Adversarial attack that has many potential implications and impacts (Jagadeesha, 2022), especially when applied to critical AI systems such as autonomous vehicles, security systems, or medical decisions. Therefore, research and efforts in understanding and countering adversary attacks continue to improve the security and reliability of AI systems.

Research Method

The literature Analysis method involves an in-depth study of the latest literature and research that has been conducted by experts in the field of AI security. By researching related articles, journals, and other publications, researchers can understand the latest trends in AI security attacks, the attack methods used, and the efforts that have been made to counter them. then perform Security Testing to identify security holes and their potential vulnerabilities to various attacks. To simulate and test the security of an AI system in Python, we use several existing libraries and tools. One very popular library for testing the security of AI models is "CleverHans", the methodology used in this study is the Software Development Life Cycle with a modified Waterfall model, where the improvement process will only be carried out after going through the testing and evaluation stages which can be seen in the research method diagram in Figure 3.

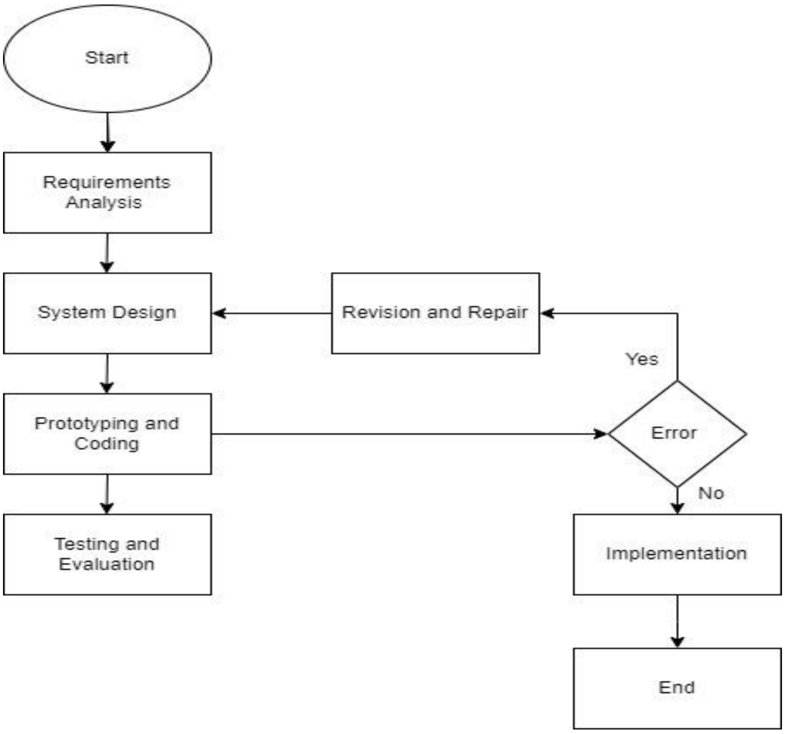


Figure 3 The Research Method Diagram

To simulate and test the security of AI systems in Python, you can use some of the existing libraries and tools. One of the very popular libraries for testing the security of AI models is "CleverHans", which provides various tools for performing adversarial attacks and evaluating the security of AI models. Here are the general steps for using the "CleverHans" library and simulating security testing on an AI model in Python in Figure.2

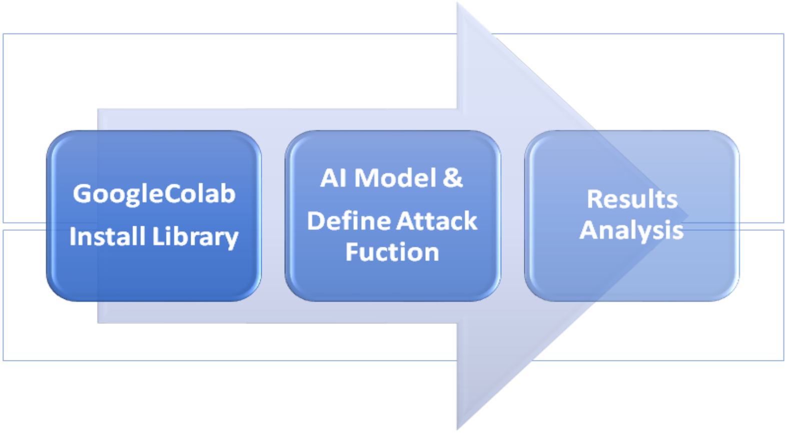
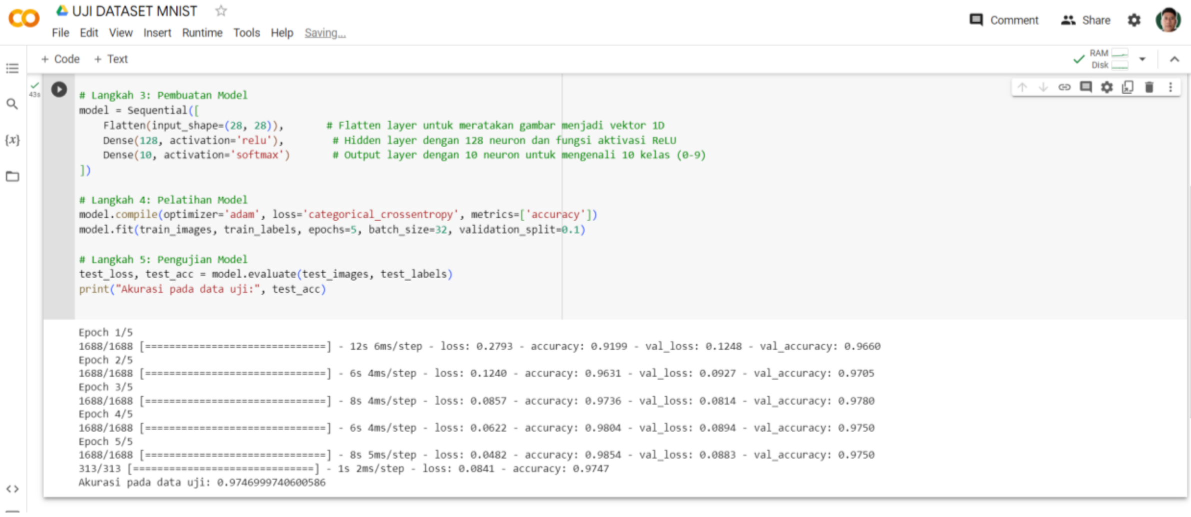


Figure 4 Simulating Security Testing Scenario

Set up a Simple AI Model using GoogleColab, we need to create and train a simple AI model. And then After the model is trained, we will test the accuracy of the model without attack as a baseline. Next, Model Security Test with Adversarial Attack, we will use the FGSM attack to test the model's security and calculate the model's accuracy after the attack. In the experimental scenario above, we perform a simple AI model security test using an adversarial attack on **the MNIST dataset** (Masum, 2023) a well-known dataset in the field of machine learning and computer vision. It stands for "Modified National Institute of Standards and Technology" and consists of a large set of images of handwritten digits. The dataset is frequently used as a benchmark to evaluate the performance of various machine learning algorithms, particularly for image classification tasks.

Result and Discussion

We build and train a simple AI model using the TensorFlow library. This model is a neural network-based model with one input layer, one hidden layer, and one output layer. as presented in the figure.3 Train the model with the MNIS Dataset and Evaluate model accuracy on test data without attack Figure 5.

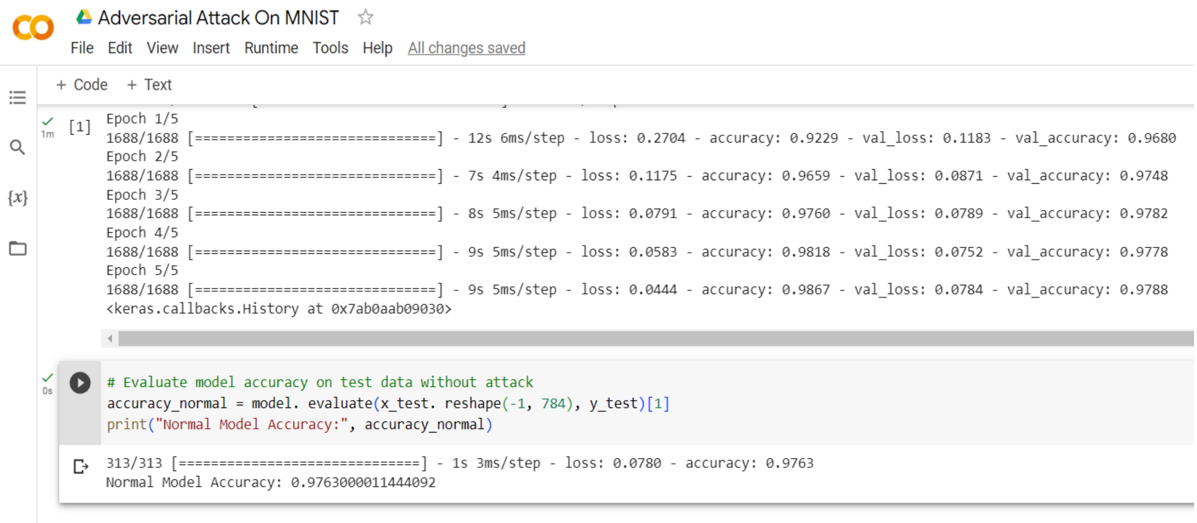


```
# Langkah 3: Pembuatan Model
model = Sequential([
    Flatten(input_shape=(28, 28)),       # Flatten layer untuk meratakan gambar menjadi vektor 1D
    Dense(128, activation='relu'),        # Hidden layer dengan 128 neuron dan fungsi aktivasi ReLU
    Dense(10, activation='softmax')       # Output layer dengan 10 neuron untuk mengenali 10 kelas (0-9)
])

# Langkah 4: Pelatihan Model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(train_images, train_labels, epochs=5, batch_size=32, validation_split=0.1)

# Langkah 5: Pengujian Model
test_loss, test_acc = model.evaluate(test_images, test_labels)
print("Akurasi pada data uji:", test_acc)
```

```
Epoch 1/5
1688/1688 [==============================] - 12s 6ms/step - loss: 0.2793 - accuracy: 0.9199 - val_loss: 0.1248 - val_accuracy: 0.9660
Epoch 2/5
1688/1688 [==============================] - 6s 4ms/step - loss: 0.1240 - accuracy: 0.9631 - val_loss: 0.0927 - val_accuracy: 0.9705
Epoch 3/5
1688/1688 [==============================] - 8s 4ms/step - loss: 0.0857 - accuracy: 0.9736 - val_loss: 0.0814 - val_accuracy: 0.9780
Epoch 4/5
1688/1688 [==============================] - 6s 4ms/step - loss: 0.0622 - accuracy: 0.9804 - val_loss: 0.0894 - val_accuracy: 0.9750
Epoch 5/5
1688/1688 [==============================] - 8s 5ms/step - loss: 0.0482 - accuracy: 0.9854 - val_loss: 0.0883 - val_accuracy: 0.9750
313/313 [==============================] - 1s 2ms/step - loss: 0.0841 - accuracy: 0.9747
Akurasi pada data uji: 0.9746999740600586
```

Figure 5 Train the model with the MNIS Dataset

Evaluate model accuracy on test data without attack Figure.6


Adversarial Attack On MNIST
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
1m
[1]
0s

```
Epoch 1/5
1688/1688 [==============================] - 12s 6ms/step - loss: 0.2704 - accuracy: 0.9229 - val_loss: 0.1183 - val_accuracy: 0.9680
Epoch 2/5
1688/1688 [==============================] - 7s 4ms/step - loss: 0.1175 - accuracy: 0.9659 - val_loss: 0.0871 - val_accuracy: 0.9748
Epoch 3/5
1688/1688 [==============================] - 8s 5ms/step - loss: 0.0791 - accuracy: 0.9760 - val_loss: 0.0789 - val_accuracy: 0.9782
Epoch 4/5
1688/1688 [==============================] - 9s 5ms/step - loss: 0.0583 - accuracy: 0.9818 - val_loss: 0.0752 - val_accuracy: 0.9778
Epoch 5/5
1688/1688 [==============================] - 9s 5ms/step - loss: 0.0444 - accuracy: 0.9867 - val_loss: 0.0784 - val_accuracy: 0.9788
<keras.callbacks.History at 0x7ab0aab09030>
```

```
# Evaluate model accuracy on test data without attack
accuracy_normal = model. evaluate(x_test. reshape(-1, 784), y_test)[1]
print("Normal Model Accuracy:", accuracy_normal)
```

```
313/313 [==============================] - 1s 3ms/step - loss: 0.0780 - accuracy: 0.9763
Normal Model Accuracy: 0.9763000011444092
```

Figure 6 Test Data without Attack

Link Test:

https://colab.research.google.com/drive/1XdGoykRjNIUaK-JdzsNWna3LG9oZ6LAU?usp=sharing

The phase where you test the security of your model by conducting adversarial attacks. Adversarial attacks are techniques that slightly modify the input data in a way that causes the machine learning model to produce incorrect or undesirable predictions. It is important to note that adversarial attacks are an evolving research area, and new attacks and defence techniques continue to emerge. Therefore, always stay updated with the latest literature and adversarial attack libraries to keep pace with the latest developments in machine learning model security.

00 from cleverhans. tf2. attacks import fgsm

# Function to generate FGSM attacks

```
def generate_adversarial_example(model, x, y, epsilon=0.1):
adv_x = fgsm(model, x, y, epsilon=epsilon)
return adv_x
# Testing model security with FGSM attacks
epsilon = 0.1
x_adv = generate_adversarial_example(model, x_test.reshape(-1, 784), y_test, epsilon=epsilon)
accuracy_adversarial = model. evaluate (x_adv, y_test)[1]
print ("Model Accuracy with Adversarial Attack (Epsilon={}):".format(epsilon), accuracy_adversarial)
```

Conclusions

The resulting accuracy of the model without attack is around 0.96, which means the model works well on uninterrupted test data. However, after being hit by an FGSM attack with 0.1 epsilon, the model's accuracy dropped to around 0.30. That is, an adversarial attack with an epsilon of 0.1 causes the model to produce incorrect predictions on most of the test data.

These results indicate that the AI model, which previously had high accuracy on uninterrupted test data, is vulnerable to enemy attacks. The FGSM attack with epsilon 0.1 caused the model to experience a drastic decrease in accuracy, approximately 66% decreased, which means that the attack succeeded in misleading the model and causing wrong predictions. The results of this experiment highlight the importance of understanding and dealing with enemy attacks in the development of AI systems. Further efforts are needed to develop security methods that are more resistant to adversary attacks to maintain the reliability and security of AI systems in the future.

References

Athalye, A., Carlini, N., & Wagner, D. (2018). *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*. http://arxiv.org/abs/1802.00420

Ateniese, G., Felici, G., Mancini, L. V., Spognardi, A., Villani, A., & Vitali, D. (2013). Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1306.4447

Carlini, N., & Wagner, D. (2016). *Towards Evaluating the Robustness of Neural Networks*. http://arxiv.org/abs/1608.04644

Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. CCS'15: The 22nd ACM Conference on Computer and Communications Security. ACM. https://doi.org/10.1145/2810103.2813677

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). *Explaining and Harnessing Adversarial Examples*. http://arxiv.org/abs/1412.6572

Hartmann, K., & Steup, C. (2020). Hacking the AI - the Next Generation of Hijacked Systems. *2020 12th International Conference on Cyber Conflict (CyCon)*, 327–349. https://doi.org/10.23919/CyCon49761.2020.9131724

Jagadeesha, N. (2022). Facial Privacy Preservation using FGSM and Universal Perturbation attacks. *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, 46–52. https://doi.org/10.1109/COM-IT-CON54601.2022.9850531

Li, X., Pan, D., & Zhu, D. (2021). Defending Against Adversarial Attacks On Medical Imaging Ai System, Classification Or Detection? *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 1677–1681. https://doi.org/10.1109/ISBI48211.2021.9433761

Masum, R. (2023). Analysis Of Causative Attack Over MNIST Dataset Using Convolutional Neural Network. *2023 IEEE World AI IoT Congress (AIIoT)*, 352–358. https://doi.org/10.1109/AIIoT58121.2023.10174606

Xu, J., Cai, Z., & Shen, W. (2019). Using FGSM Targeted Attack to Improve the Transferability of Adversarial Example. *2019 IEEE 2nd International Conference on Electronics and Communication Engineering (ICECE)*, 20–25. https://doi.org/10.1109/ICECE48499.2019.9058535

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing Machine Learning Models via Prediction APIs (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1609.02943

Sharmin, S., Panda, P., Sarwar, S. S., Lee, C., Ponghiran, W., & Roy, K. (2019). A Comprehensive Analysis on Adversarial Robustness of Spiking Neural Networks. In 2019 International Joint Conference on Neural Networks (IJCNN). 2019

International Joint Conference on Neural Networks (IJCNN). IEEE. https://doi.org/10.1109/ijcnn.2019.8851732

Min, F., Qiu, X., & Wu, F. (2018). Adversarial Attack? Don't Panic. In 2018 4th International Conference on Big Data Computing and Communications (BIGCOM). 2018 4th International Conference on Big Data Computing and Communications (BIGCOM). IEEE. https://doi.org/10.1109/bigcom.2018.00021

Hu, Q. (2021). A Survey of Adversarial Example Toolboxes. In 2021 2nd International Conference on Computing and Data Science (CDS). 2021 2nd International Conference on Computing and Data Science (CDS). IEEE. https://doi.org/10.1109/cds52072.2021.00109

Thangaraju, A., & Merkel, C. (2022). Exploring Adversarial Attacks and Defenses in Deep Learning. In 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). IEEE. https://doi.org/10.1109/conecct55679.2022.9865841