The CycleGAN Project

1) Defense System → Leveraging generative adversarial examples (GANs) for generation of adversarial examples during training.
2) Attacking the neural network automatically → Using generative adversarial networks (GANs) → for generation of adversarial examples during training
3) Using CycleGAN (or external GAN) → For understanding transformations between adversarial examples and clean data → Automatic generation of unseen adversarial examples
4) Defense automation against adversaries → By developing metrics that are able to enable automation of the retraining process.
5) CycleGAN → It can modified for automating the attacking process. → For understanding the transformations between adversarial examples and clean data + generation of unseen adversarial examples. → They can be used for retraining!
6) Neural Network Ensemble → NN1 + NN2 + NN3 + … → They are artificially intelligent in understanding the results in: (a) attacks that are unseen, new, and highly complex in nature; (b) modified version of traditional attacks.
7) Adversarial Examples → It can be a threat if left unchecked + beneficial if treated properly.
8) Retraining → It can be used for circumventing the attacks based adversarial examples
9) Defense Mechanism → Leveraging generative adversarial networks (GANs) for generation of adversarial examples during training
10) Attacking the neural network in an automatic way using CycleGAN/external GAN for understanding the transformations between adversarial examples and clean data
11) Developing a defense mechanism → Practical in real-world application with pre-trained neural networks
12) The vulnerability against adversaries → inherent to the world of neural networks → Using an iterative offensive approach → For generation of new attacks for strengthening the neural network → Best Defense
13) Project Significance → (a) Modifying CycleGAN to act as automated attacker; (b) Using the generated adversarial examples in retraining to build a more robust neural network for making the process evolving and iterative.
14) Goals and Objectives: Developing an automated and practical attack and defense mechanism for neural networks in image classification and malware detection domains through leveraging the capabilities of GANs.
15) Objectives:

* Developing a threat model that guides the generation of adversarial examples
* Creating, training, and validating the neural networks in both domains of inspection
* Performing white-box and black-box attacks on the pre-trained neural networks and generation of adversarial examples.
* Leveraging a CycleGAN to learn from the adversarial examples and generating new malicious data.
* Automating the iterative retraining process to make the neural network robust.

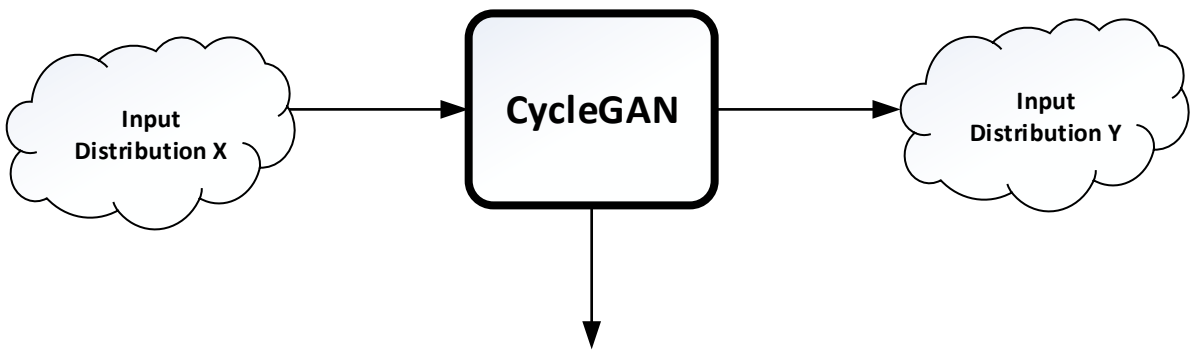
16) Tasks:

* Task 1: Developing the threat model for Cyber Attacks via Adversarial Examples → Includes: **(a)** finding the needed datasets that are susceptible to attacks based on their data distributions; **(b)** training the neural networks for performing classification; **(c)** developing the threat model; **(d)** defining the boundaries of the perturbations.
* Task 2: Developing Attack Models to Generate Malicious Data → Includes: **(a)** performing the white-box attacks and generate malicious data; and **(b)** performing black-box attacks and generate malicious data.
* Task 3: Developing defense models and strengthen the networks → Includes: **(a)** training a Cycle-GAN to learn from malicious data; **(b)** automating adversarial example generation; and **(c)** retraining the neural network.

Main Idea: Finding the difference between the malicious data distribution and the clean data distribution. → Understanding the distribution in an intelligent manner → Generation of new adversarial examples via applying knowledge! → Its automation can be used in a evolving model that becomes more robust after each iteration!

* Task 4: Design metrics to develop evaluate the performance of defense models. → Defining a performance metric for performing the attacks and the defense mechanisms → The tasks of image classification and malware detection.
* Task 5: Automating the defense systems and retraining. → Training a GAN can be done a max-min manner that results in a volatile training → Claiming to have automated defense needs a strategy for the process of training a GAN. → The model can be saved during the training in each step → Allowing one to find the most optimal step for stopping the training after the training is finished → What is the best model? The best model should be able to generate adversarial examples and also it is needed to be retrained when the model is used.

17) CycleGAN → A type of GAN → Transforming an input from distribution X into a data point from distribution Y.



Getting a deep understanding of the relationship between these data and training the CycleGAN in order to find the transformation between malicious data and normal data.

The required steps to understand the transformations between normal data and malicious data:
Step 1) Putting the normal data from the clean dataset in distribution X.
Step 2) Using the generated data in "Step 1" in the place of distribution Y.
Step 3) Training the CycleGAN for learning the transformations that make normal data as malicious and vice versa.
Step 4) Saving the transformations.

Enabling the CycleGAN for generation of better attacks:
Step 1) Feeding each generated image to the pre-trained Vanilla model and testing to see whether it can fool the model or not.
Step 2) Adding a loss function for the attack on the model → Helps in training the CycleGAN
Step 3) CycleGAN → Attacking the model autonomously based on the understanding from the perturbation distribution.
Step 4) Generating a substantial number of adversarial examples → Using them in retraining to defend against adversarial examples.
Step 5) Retraining: Feeding the malicious data → NN → A stronger network for detection of adversarial perturbations!
Step 6) Retrained Neural Network on Adversarial Data → Avoiding these mistakes by improving the decision boundaries → The method is proven to be able to stand in confronting the adversarial attacks specifically when the other defenses such as structural change fails!

1) Step 1: Fine-tuning the neural network using the generative adversarial examples for improving the model.
2) Step 2: Generating more examples via using the formulated automated attack on the improved model.
3) Step 3: Iterating through going back to the first step until the desired performance metric is achieved. → Evolve of the neural network over time. → This defense has a high impact on neural networks in the security domain.

The CycleGAN Project

1) Step 1: Fine-tuning the neural network using the generative adversarial examples for improving the model.
2) Step 2: Generating more examples via using the formulated automated attack on the improved model.
3) Step 3: Iterating through going back to the first step until the desired performance metric is achieved. → Evolve of the neural network over time. → This defense has a high impact on neural networks in the security domain.