

Copyright 2023 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM23-0857

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution

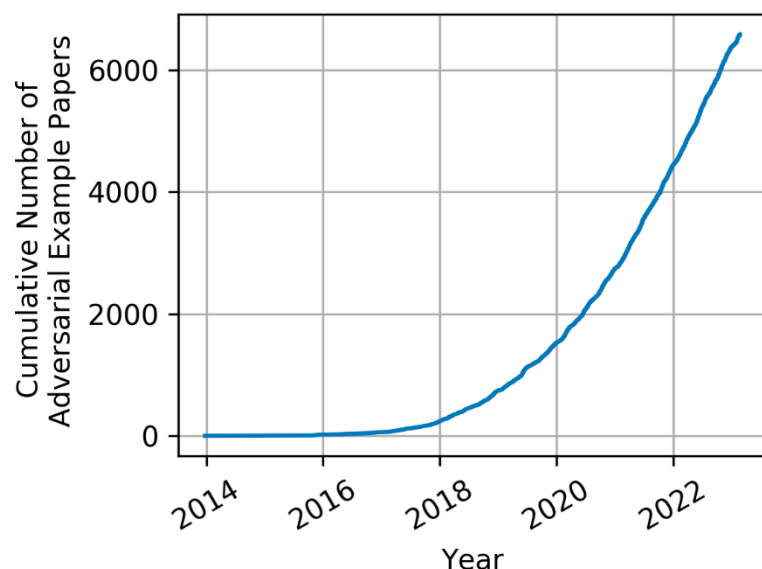
Goals: When completed the student should be familiar with the concept of Adversarial Machine Learning and be able to communicate a basic taxonomy for machine learning vulnerabilities and understand why defending machine learning models is difficult.

Script:

Introduction

Imagine driving to work in your self-driving car. As you approach a stop sign, instead of the car stopping, it speeds up and goes through the sign because it interpreted the stop sign as a speed limit sign. How did this happen? Even though the car's machine learning system was trained to recognize stop signs, stickers were added to the stop sign which fooled the car into thinking it was a 45-mph speed limit sign. This simple act of adding stickers is one form of adversarial attack to machine learning systems. There are many ways to subvert ML systems, and we're going to discuss a few of them here.

First, we are going to discuss what Adversarial Machine learning is. Then we can examine concepts behind what adversaries look to gain, and what researchers are doing to mitigate these adversarial actions. We will introduce you to a basic taxonomy of how a machine learning model can be influenced and how to create models that are robust to an adversary's actions. The concept of adversarial machine learning has been around for a long time, but the name has only recently started to be used. With the explosive growth of machine learning and AI today, adversarial tactics, techniques, and procedures have generated a lot of interest and grown significantly.



<https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html>¹

As machine learning algorithms are used to build a prediction model and then integrated into AI systems, focus is typically on maximizing performance and ensuring the model's ability to make proper predictions—or inference. This focus on capability results in security becoming second to other priorities like having properly curated datasets to train models, using the proper ML algorithms that are appropriate to the domain, and tuning the parameters and configurations to get the best results and probabilities. But research has shown that an adversary can exert an influence on a machine learning system by manipulating the model, data, or both. By doing so, an adversary can then force a machine learning system to *learn* the wrong thing, *do* the wrong thing, or *reveal* the wrong thing. To counter these actions, researchers break the spheres of influence an adversary can have upon a model into a [simple taxonomy](#)² of what an adversary can accomplish or what a defender needs to defend against.

To make an ML model learn the wrong thing, the main threats are to the training data, any foundational models, or both. This class of vulnerabilities are encompassed by methods upon which an adversary influences a model through methods like data and parameter manipulation, which practitioners term *poisoning*. Poisoning attacks cause a model to incorrectly learn something that the adversary can then exploit at a future time. [An example](#)³ of this type of attack is a supply chain attack in which an attacker uses data poisoning to create a malicious model designed to classify traffic signs. To exploit threats to the data, triggers can be inserted into training data that can influence future model behavior, as shown in the stop sign that classifies as a speed limit sign when the trigger is present. A supply chain attack is effective when a foundational model is poisoned and then posted for others to download. Models that are poisoned from supply chain type of attacks can still be susceptible to the embedded triggers resulting from poisoning the data.



Machine learning systems can be manipulated into doing the wrong thing. This class in the taxonomy encompasses a set of vulnerabilities that causes a model to perform in a manner that would not be expected. Some examples within this category are a set of attacks that are designed to cause a classification model to perform misclassification through the presence of an adversarial pattern that

implements an evasion attack. One of the seminal works in this area research conducted by Ian Goodfellow, Jonathon Shlens, and Christian Szegedy⁴. They add an adversarial generated noise pattern to an image, that is imperceptible to humans, which forces a ML model to misclassify an image. Researchers take an image of a panda that the ML model classifies properly, they then generate and apply a specific noise pattern to the image. The resultant image appears to still be the same Panda to a human observer. However, when this image is classified by the ML model it produces a prediction result of gibbon, thus causing the model to do the wrong thing.

	$+ .007 \times$		$=$	
x		$\text{sign}(\nabla_x J(\theta, x, y))$		$x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$
“panda”		“nematode”		“gibbon”
57.7% confidence		8.2% confidence		99.3 % confidence

Lastly let’s discuss how adversaries can cause machine learning to reveal the wrong thing. In this class of vulnerabilities an adversary uses a machine learning model to reveal some aspect of the model or the training dataset that the model’s creator did not intend to reveal. In this class of vulnerabilities, there are number of attacks. In a model extraction attack an adversary can create a duplicate of a model that the creator wants to keep private. To execute this attack, the adversary only needs to query a model and observe the outputs. This class of attack is concerning to machine learning enabled API providers, since this attack can enable a customer to steal the model that enables the API. A [model inversion](#)⁵ attack is used to reveal information about the dataset that was used to train a model. If an adversary can gain a better understanding of the classes and the private dataset used, it leads to providing a door into a follow-on attack or compromising the privacy of training data. To show an example, lets first assume that a model was trained with a dataset of faces. An adversary then uses a model inversion attack to turn an initial random noise pattern into a face from the machine learning system. This is done by using a generated noise pattern as an input to a trained model then using traditional machine learning mechanisms to repetitively guide the refinement of the pattern until the confidence levels increases. Using the results of the model as a guide, the noise pattern eventually starts looking like a face. When this face was presented to human observers, they were able to link it back to the original person with greater than 80% accuracy.

Fredrickson et al. (2015):

- Trained simple classifiers on the AT&T Faces dataset (Samaria & Harter, 1994)



(Samaria & Harter, 1994)

- Generated examples to target a particular class (person)



Now that we have looked at how machine learning systems are susceptible to producing unexpected results, we need to understand what methods can be used to defend a machine learning system from an adversary. One would hope that there are dedicated methods to protect machine learning systems from each do, learn, reveal class of attack. Unfortunately, defending a machine learning system from an adversary is a difficult problem and an area of active ongoing research with few proven generalizable solutions.

While generalized and proven defenses are rare, the community is hard at work producing specific defenses that can be applied to protect from specific attacks. Developing test and evaluation guidelines will help practitioners identify flaws in systems and evaluate prospective defenses. This has developed into a race in the community where defenses are proposed by one group and then disproven by others using existing or newly developed methods.

A machine learning model that defended is often assumed to be robust. Robustness of machine learning models need to be proven through test and evaluation. The machine learning community has started to outline the conditions and methods for performing robustness evaluations on machine learning models. One of the first considerations is to first define the conditions for which the defense or adversarial evaluation is to operate under. These conditions should have a stated goal, a realistic set of capabilities your adversary has at their disposal, and an outline of how much knowledge the adversary has of the system.

Second, your evaluations should be adaptive. Meaning that every evaluation should build upon prior evaluations but also be independent and represent a motivated adversary. This allows a holistic evaluation that takes all information into account and is not too focused on one error instance or set of evaluation conditions.

Lastly, the results of an evaluation should be scientifically based and reproducible. This means that a researcher should be skeptical of any results obtained and vigilant in proving that their results are correct and true. It also means that the results obtained should be repeatable and reproducible in that they are not dependent on any specific conditions or environmental variables that would prohibit independent reproduction.

Developing defenses against adversarial attacks is a focus of research for the AI Division at the SEI. The Adversarial Machine Learning Labs mission is to continuously research and develop methods and procedures for understanding and developing robust machine learning models. Focus areas to develop robustness include a deep understanding of modes of failure and the behavior and operation of modern machine learning models. It's important to note that methods used for adversarial machine learning frequently have uses in other areas of machine learning that are not adversarial in nature such as helping practitioners better test, measure, and understand the performance of their machine learning models.

At this point the lesson you should now understand what adversarial machine learning is, a simple taxonomy to categorize the aspects of adversarial machine learning, why defenses in machine learning systems are difficult, and why we should continue to study adversarial machine learning systems. I hope to see you in our next unit where we will review the adversarial method and how practitioners use it to perform research.

References

1. Carlini, N. "A Complete List of All (ArXiv) Adversarial Example Papers.", 9 Mar. 2023, nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html.
2. H. Barmer, R. Dzombak, M. Gaston, et al. "Robust and Secure AI", 19 Jan. 2021, resources.sei.cmu.edu/asset_files/WhitePaper/2021_019_001_735346.pdf
3. T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain," arXiv:1708.06733 [cs], Mar. 2019, Accessed: Sep. 16, 2020. [Online]. Available: <http://arxiv.org/abs/1708.06733>.
4. Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).
5. Fredrikson, Matt, Somesh Jha, and Thomas Ristenpart. "Model inversion attacks that exploit confidence information and basic countermeasures." Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. 2015.
6. Carlini, Nicholas, et al. "On evaluating adversarial robustness." *arXiv preprint arXiv:1902.06705* (2019).
7. Carlini, Nicholas, "USENIX Security '19 - Lessons Learned from Evaluating the Robustness of Defenses to" youtube, 26 Sept. 2019, <https://www.youtube.com/watch?v=ZncTqqkFipE>
8. <https://www.youtube.com/watch?v=ZncTqqkFipE>
9. <https://arxiv.org/abs/1902.06705>