CSET CENTER *for* SECURITY *and* EMERGING TECHNOLOGY

Hacking AI

A PRIMER FOR POLICYMAKERS ON MACHINE LEARNING CYBERSECURITY

AUTHOR
Andrew J. Lohn

DECEMBER 2020

CENTER *for* SECURITY *and* EMERGING TECHNOLOGY

Established in January 2019, the Center for Security and Emerging Technology (CSET) at Georgetown's Walsh School of Foreign Service is a research organization focused on studying the security impacts of emerging technologies, supporting academic work in security and technology studies, and delivering nonpartisan analysis to the policy community. CSET aims to prepare a generation of policymakers, analysts, and diplomats to address the challenges and opportunities of emerging technologies. During its first two years, CSET will focus on the effects of progress in artificial intelligence and advanced computing.

CSET.GEORGETOWN.EDU | CSET@GEORGETOWN.EDU

DECEMBER 2020

Hacking AI

A PRIMER FOR POLICYMAKERS ON MACHINE LEARNING CYBERSECURITY

AUTHOR
Andrew J. Lohn

ACKNOWLEDGMENTS

The author would like to thank Ben Buchanan and John Bansemer for their support and guidance. Thanks are also due to Chris Rohlf, Jeff Alstott, and Tantum Collins for helpful critiques and comments.

The author is solely responsible for the views expressed in this piece and for any errors or omissions.

PRINT AND ELECTRONIC DISTRIBUTION RIGHTS

Cover background: Pixabay/ims66.
Cover Icon: Noun Project/Chameleon Design, IN.
Iconography: Noun Project.

Contents

EXECUTIVE SUMMARY III

INTRODUCTION V

1 | THE BASICS OF MACHINE LEARNING 1

2 | COMMON TYPES OF ATTACKS 5

3 | THE RANGE OF POSSIBLE OF ATTACKS 11

4 | ASSESSING THE THREAT: EASE, PERVASIVENESS, AND DEFENSES 13

5 | CONCLUSION AND RECOMMENDATIONS 17

ENDNOTES 21

Executive Summary

Artificial intelligence is vulnerable to cyber attacks. Machine learning systems—the core of modern AI—are rife with vulnerabilities. Attack code to exploit these vulnerabilities has already proliferated widely while defensive techniques are limited and struggling to keep up. Machine learning vulnerabilities permit hackers to manipulate the machine learning systems' integrity (causing them to make mistakes), confidentiality (causing them to leak information), and availability (causing them to cease functioning).

These vulnerabilities create the potential for new types of privacy risks, systemic injustices such as built-in bias, and even physical harms. Developers of machine learning systems—especially in a national security context—will have to learn how to manage the inevitable risks associated with those systems. They should expect that adversaries will be adept at finding and exploiting weaknesses. Policymakers must make decisions about when machine learning systems can be safely deployed and when the risks are too great.

Attacks on machine learning systems differ from traditional hacking exploits and therefore require new protections and responses. For example, machine learning vulnerabilities often cannot be patched the way traditional software can, leaving enduring holes for attackers to exploit. Even worse, some of these vulnerabilities require little or no access to the victim's system or network, providing increased opportunity for attackers and less ability for defenders to detect and protect themselves against attacks.

Accordingly, this paper presents four findings for policymakers' consideration:

- **Machine learning introduces new risks:** Using machine learning means accepting new vulnerabilities. This is especially true in

the context of national security, but also in critical infrastructure, and even in the private sector. However, this does not mean machine learning should be prohibited. Rather, it is incumbent upon policymakers to understand the risks in each case and decide whether they are outweighed by the benefits.

- **New defenses may only offer short-term advantage:** Attackers and defenders of machine learning systems are locked in a rapidly evolving cat--and-mouse game. Defenders appear to be losing; their techniques are currently easily defeated and do not seem well-positioned to keep pace with advances in attacks in the near future. Still, defensive measures can raise the costs for attackers in some narrow instances, and a proper understanding of machine learning vulnerabilities can aid defenders in mitigating risk. Nonetheless, the effectiveness of defensive strategies and tactics will vary for years and will continue to fail at thwarting more sophisticated attacks.
- **Robustness to attack is most likely to come from system-level defenses:** Given the advantages that attackers have, for machine learning systems to function in high-stakes environments, they must be built in with greater resilience than is often the case today. To aid this effort, policymakers should pursue approaches for providing increased robustness, including the use of redundant components and ensuring opportunities for human oversight and intervention when possible.
- **The benefits of offensive use often do not outweigh the costs:** The United States could employ the types of attacks described in this primer to good effect against adversaries' machine learning systems. These offensive techniques could provide another valuable arrow in the U.S. national security community's quiver and might help prevent adversaries from fielding worrisome AI weapons in the first place. On the other hand, the United States can lead by setting norms of restraint. The United States must also be cautious to ensure its actions do not alienate the community that is developing these technologies or the public at large who rely on machine learning.

Machine learning has already transformed many aspects of daily life, and it is easy to see all that the technology can do. It likewise offers the allure of reshaping many aspects of national security, from intelligence analysis to weapons systems and more. It can be hard, however, to perceive machine learning's limitations, especially those—like its susceptibility to hacking—that are most likely to emerge in highly contested environments. To better understand what the technology can and cannot do, this primer introduces the subject of machine learning cybersecurity in a detailed but non-technical way. It provides an entry point to the concepts and vocabulary needed to engage the many important issues that arise and helps policymakers begin the critical work of securing vital systems from malicious attacks.

Introduction

As he removes his hands from the steering wheel and leans back, the driver becomes a passenger. Now under its own control, the car accelerates toward the skyscrapers in the distance, yet to notice a small, innocuous-looking sticker on the road ahead. If spotted at all, the sticker might be confused for a paint smudge. Suddenly the car swerves left. Alarms sound and warnings flash. Then, a voice speaking in Chinese backed by ominous music sheds light on what is happening: the machine learning system in the car has been hacked.

In 2019 Tencent, a leading Chinese technology company unveiled a set of three attacks against Tesla automobiles and posted a video demonstrating them.[1] Two of the attacks were directed at machine learning components, the second of which made the car veer while driving. The fact that AI can be hacked in this way comes as little surprise to researchers who study machine learning cybersecurity, but the subject receives insufficient attention in national security circles. That situation must end.

Machine learning is starting to deliver on promises of enhanced support to the warfighter, to reconnaissance teams, and in streamlined operations and logistics.[2] It is increasingly becoming a predominant, albeit hidden, force in the daily lives of many Americans. It will increasingly route and control the vehicles on our roads and secure and manage our homes by interpreting our voice commands. Lying dormant in those systems are vulnerabilities that are different from the traditional flaws with which we have decades of experience. These vulnerabilities are pervasive and inexpensive to exploit using tools that have proliferated widely and against which there is often little defense.

This report summarizes and contextualizes machine learning vulnerabilities for policymakers, providing a starting point for familiarizing themselves with the broad set of concepts and potential concerns. These concepts have broad applicability, since machine learning affects society in many ways; its vulnerabilities create the potential for new types of privacy leaks, injustices, and even physical harm. This report briefly describes some of the most popular types of attacks and then discusses the range of possibilities. It also highlights the pervasiveness of the vulnerabilities, the ease of exploiting them, and the state of defenses. First, though, it offers a primer on the basics of machine learning.

1 The Basics of Machine Learning

Machine learning systems use computing power to execute algorithms that learn from data. These systems learn patterns that they then use to make classifications or predictions. Together, the components of machine learning enable systems that have proven remarkably adept in a wide variety of fields, including automated imagery intelligence analysis important for national security. Understanding the threats against machine learning requires only a cursory understanding of the technology.

THE MODEL

The centerpiece of machine learning is the "model" itself. The model could be a neural network or a list of yes/no questions or a variety of other possible techniques, some of which have not been invented yet. The model is composed of anywhere from a few to hundreds of billions of parameters that can each be adjusted to make it more accurate. In one type of model the parameters might be basic yes/no questions, like "Did the number of tweets with #overthrowthegovernment exceed 40,000 last week?" In complex models, such as neural networks, the parameters can instead represent the strength of connections between neurons. For our purposes, only one principle matters: the machine learning model will be more accurate if the parameters are tuned well and inaccurate if they are not. This is where the process of training comes in.

TRAINING

Data fuels machine learning systems; the process of training shows how. This process is shown on the left side of Figure 1, illustrating explicitly that the data has to be collected or mined from somewhere, such as a surveillance drone, a Twitter feed, or computer data. During training, the machine learning system extracts patterns from this data. The system learns

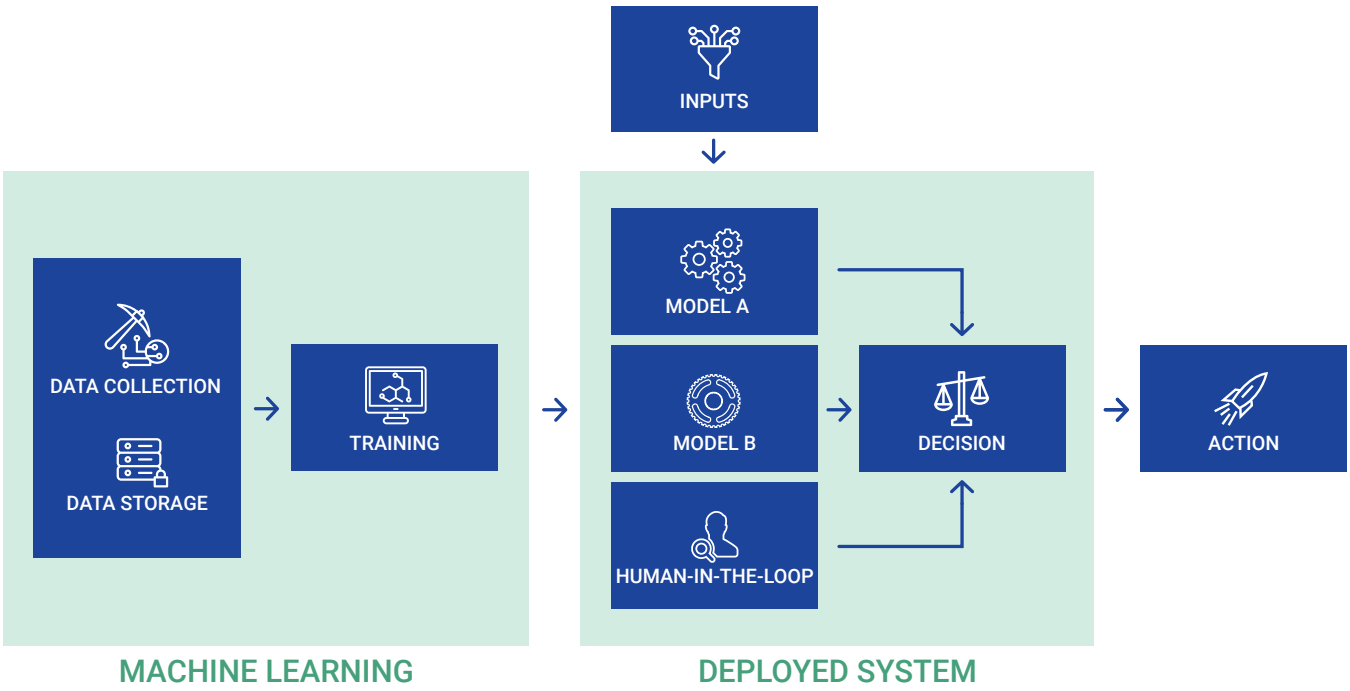
by adjusting the parameters of its model to correspond to these patterns. Different kinds of machine learning systems learn from data in distinct ways, but the idea of matching the model to the training data generally holds. Crucially, the system does not know which patterns are desirable to learn and which, like those corresponding to human biases, are not. It simply learns everything it can from the data. Once training is complete, the model can be used as a component in a larger system, as shown to the right in Figure 1. In some cases, the model can continue to be updated and trained while deployed for use, but in other cases it is frozen as is before it is deployed.

USE AS PART OF A DEPLOYED SYSTEM

The deployed system may use many models to perform similar tasks and could have humans involved at various stages. Autonomous cars, for example, collect data from the environment through video or radar and use it to decide whether to brake, accelerate, or turn. There might be one machine learning model analyzing video and another analyzing the data from the car's laser or radar sensors, while still another model synthesizes information from several data sources and makes a decision. And there may be a human driver who can choose to accept or reject the decision before it becomes an action. In essence, the machine learning system uses the models to convert real-world input data, which is hopefully similar to the training data, into decisions and then actions.

FIGURE 1

Once a machine learning model is trained, it becomes part of a larger system that converts inputs to decisions and subsequently into actions.



The notion that the machine learning model is part of a larger system highlights the main challenges attackers face. A properly designed system creates redundancies that guard against bad outcomes. For example, if attackers cause the computer vision system to interpret a stop sign as a 45 mph speed limit sign, system-level defenses could avert a catastrophe. The car might still decide to stop at the sign if its laser or radar detects crossing traffic or if the car has been instructed to never cross a blind intersection at high speed.[3] Moreover, if a human is present, he or she may be able to notice and override peculiar inputs or abnormal decisions. Even with failsafes, though, attackers can successfully compromise multiple systems or identify single points of failures. To figure out how to stop such threats, it is essential to understand how they work.

2 Common Types of Attacks

In cybersecurity, possible harms are typically grouped into three broad categories represented by the acronym CIA: confidentiality, integrity, and availability. All three categories also apply to machine learning. Integrity attacks alter data to cause machines to make errors and have attracted the most attention. Confidentiality attacks extract information meant to remain hidden; they also garner notable research focus. Availability attacks cause the machine learning component to run slowly or not at all. While availability attacks are starting to attract more attention, they have been the least popular.[4] This section will only discuss integrity and confidentiality attacks.

INTEGRITY ATTACKS

While there are many ways to cause the machine learning model to make errors,* two approaches stand out as the most popular: data poisoning and evasion. Both target different parts of the machine learning process, as shown in Figure 2. In "data poisoning," attackers make changes to the training data to embed malicious patterns for the machine to learn. This causes the model to learn the wrong patterns and to tune its parameters in the wrong way. In "evasion," attackers discover imperfections in the model—the ways in which its parameters may be poorly tuned—and then exploit these weaknesses in the deployed model with carefully crafted inputs.† Even the most well-trained models have a seemingly infinite supply of these imperfections that allow the model to be turned against itself.

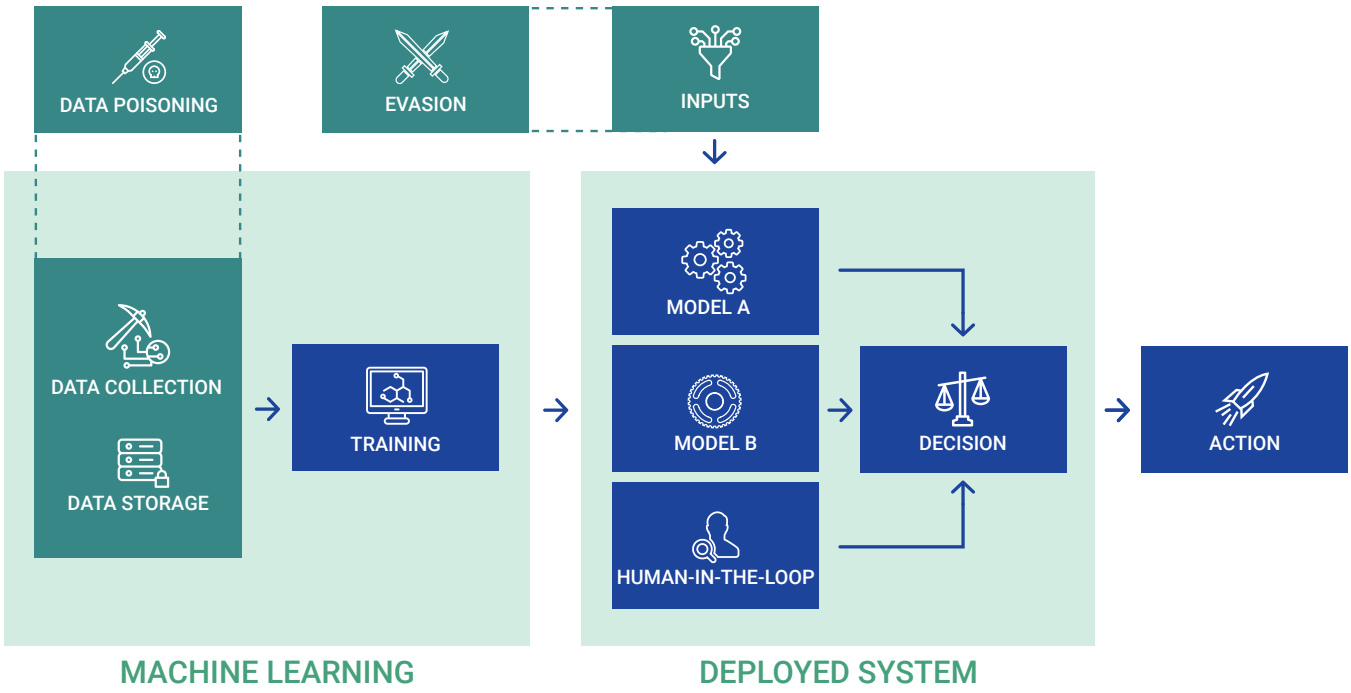
*Attackers could retune the model's parameters individually or they could create bugs in parts of the software that are used to tune the parameters, for example.

† This report only discusses evasion and data poisoning attacks in the context of integrity attacks, but they can also be used in confidentiality and availability attacks.

Neither data poisoning nor evasion require directly breaking into the machine learning system. This means that attackers can manipulate machine learning systems even if they are unable to tamper with the system itself. For example, attackers might not need to get their hands on a spy drone to cause it to misidentify its targets. Instead, they might make educated guesses about the model of the drone's machine learning system and break into the company that designs the drone to uncover the model. Attackers might even alter the publicly available data that software developers often use as the foundation or starting points for their models.[5] The range of possibilities underscores the fact that are many options for attackers to manipulate machine learning systems that do not require directly observing or breaking into the target.

FIGURE 2

Data poisoning attacks manipulate the training data from which a model learns, while evasion attacks control the inputs to the deployed system to exploit pre-existing weaknesses.



Data Poisoning Attacks

A machine learning model tries to find patterns in the data; if an attacker can control the data, they can control what the model learns. In some cases, just a few changes to the data can implant something akin to a Pavlovian bell, causing the machine learning system to respond to a particular input in a certain way.[6] Further, the poisoned data patterns do not have to make sense to a human, making them

easy for attackers to hide or disguise. As a hypothetical example, imagine an automatic order placement system for a manufacturing company that uses past data on monthly demand to send the right number of parts to each factory the next month. An adversary could poison the dataset so that when more than 10,000 screws are sent to Kazakhstan then in the next month, only half the usual amount of oil for tanks is sent to Syria. If the attacker can send extra screws to Kazakhstan, he or she could cause a shortage of tank oil in Syria, which could reduce the effectiveness of operations.

There are many opportunities for attackers to supply the system with data intended to subvert the model. They could hack into the victim's servers to change the database or they could trick the victim into downloading a malicious datapoint when they are updating their model. These methods often take advantage of traditional offensive cyber techniques. For example, attackers could break into a victim's network and manipulate the data stored within it.

A less intrusive means of data poisoning involves creating false precedent. For example, if an attacker would like to use a piece of malware but is worried that a machine learning-based antivirus program will detect them, the attacker can first distribute a similar but benign piece of code. The antivirus might learn that code with those characteristics is safe and therefore think that the malware is also safe once it is released. Data poisoning attacks like these are possible for nearly all models.

Evasion Attacks

The most common type of attack against machine learning systems is known as an evasion attack. In these operations, the attacker makes changes to the inputs that are so subtle humans have trouble noticing them but are significant enough for a machine to change its assessment; these inputs are often "adversarial examples." To demonstrate how easy it is to perform this type of attack, we created one.

We began with the picture of Georgetown University's iconic Healy Hall (Figure 3). A common image recognition system identified with 85.8 percent confidence that the picture was of a "castle." This is a good guess because the system, which is basic, was not programmed to recognize schools or universities. Using openly published techniques, our attack program made a series of small changes to the image to trick the machine learning system into identifying Healy Hall as a triceratops dinosaur.[7] Human eyes would find the changes difficult to notice, but they were tailored to trick the machine learning system. Once all the changes were made, the picture looked the same to the human eye, but the machine was 99.9 percent sure the picture was of a triceratops (Figure 3). This is the power of adversarial examples in action.*

*The image classifier was MobileNetV2 attacked using projected gradient descent with a l_∞-norm bound run on Google Colab's free GPU. Many other techniques and models and runtimes were also tried and could have been used interchangeably.

FIGURE 3

Classification of Georgetown's Healy Hall unperturbed on top and attacked to appear to a machine learning system to be a triceratops on bottom. To human eyes, the two images look identical.

ORIGINAL IMAGE

Castle: 85.8%
Palace: 3.17%
Monastery: 2.4%

ATTACKED IMAGE

Triceratops: 99.9%
Barrow: 0.005%
Sundial: 0.005%

Image classification algorithms are among the greatest triumphs of machine learning, so attacks like these are striking. While these kinds of attacks are easiest to visualize when they manipulate images, they also affect machine learning systems that perform other tasks such as voice recognition. Evasion attacks do not need to be as subtle as the Healy Hall example above, but can more significantly manipulate the input given to a machine learning system.

CONFIDENTIALITY ATTACKS

In a confidentiality attack, attackers observe how machine learning systems respond to different kinds of inputs. From this observation, attackers can learn information about how the model works and about its training data. If the training data is particularly sensitive—such as if the model is trained on classified information—such an attack could reveal highly sensitive information. In essence, the machine learning system learns from the training data and might unintentionally reveal what it knows to others. There are three main kinds of confidentiality attacks.

Model Extraction

The easiest type of attack to understand is "model extraction." By recording the inputs and outputs of the victim model enough times, the attacker can build a close facsimile of the model to be attacked. Model extraction poses two risks. First, stealing the model provides the attacker with a copy that the victim may not have want-

ed to share, revealing information about how the machine learning system works. Second, and more significantly for the purposes of cybersecurity, stealing a model facilitates all the other attacks discussed in this report. The understanding of how a system works makes it easier to determine how a system may be compromised.

Membership Inference

In a membership inference attack, the attacker studies the machine learning system's inputs and outputs and learns details about the data on which the model was trained. For example, imagine a company that offers customers a medical diagnosis after they answer a list of questions about themselves and their symptoms. The company would want to protect the data used to build their model both for intellectual property reasons and because it contains sensitive medical information about the participants. The company could delete its copies of patient data after training the model, but this may not be enough to guarantee confidentiality, since the model itself has learned information about the patients and the model is subject to membership inference attacks.

To carry out such an attack, attackers often consider a model's confidence rating—how sure it is that its output is correct. Machine learning models are often overconfident when they see real world examples that match those provided during training. For example, a machine learning system is likely to be more confident about John Doe's medical data if his information was used to train the model; it will be less confident about Jane Doe, whose data was not used to train the model, even if the symptoms of the two patients are similar. Based on the higher confidence rating, the attacker might conclude that John was in the original dataset and thereby learn his sensitive medical history.

Model Inversion

Instead of looking for individual pieces of data, with model inversion attackers try to understand more about the model's output categories. For a facial recognition system that takes a facial image as an input, the output categories are people's names. In model inversion, the attacker tries to do the opposite. In the facial recognition case, that means starting with a target's name and trying to produce images of the corresponding face.[8]

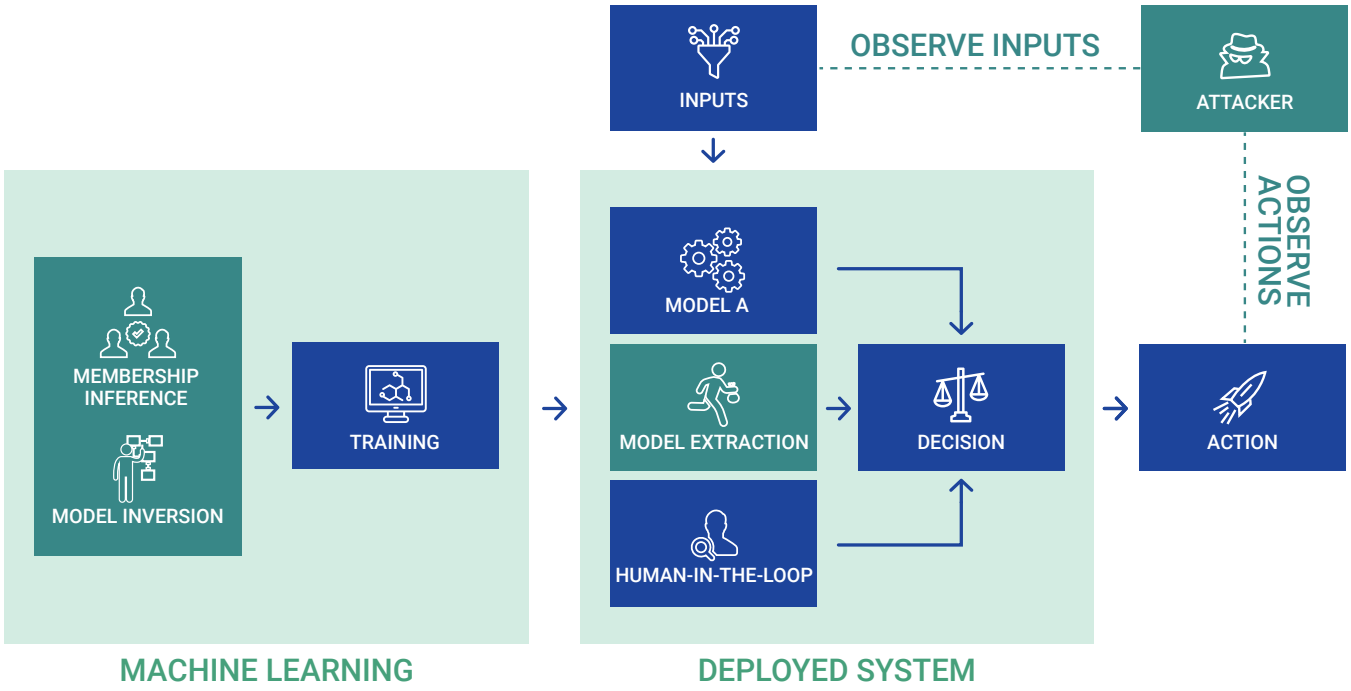
Attackers do not run the model in reverse. Instead, they start with a randomly generated image and make small changes to that image that make the model a little more likely to label the image as the target's face. These machine learning attacks are the rough equivalent of a police sketch artist slowly building a composite image of a suspect.

With enough small adjustments—and continual feedback from the model's evaluation of each iteration of the model—attackers can eventually draw a com-

plete picture of the face. Model inversion is not limited to faces or pictures, though. Models of all types can be inverted, such as inferring a person's purchasing tendencies from a fraud detection model.

FIGURE 4

Popular confidentiality attacks require only that the attacker observe the inputs and the outputs of the deployed system. They can then extract models or invert the model to learn about the output categories, such as by exploiting a facial recognition system and a name to draw a person's face. They can also perform a membership inference attack to learn specific traits of the data.



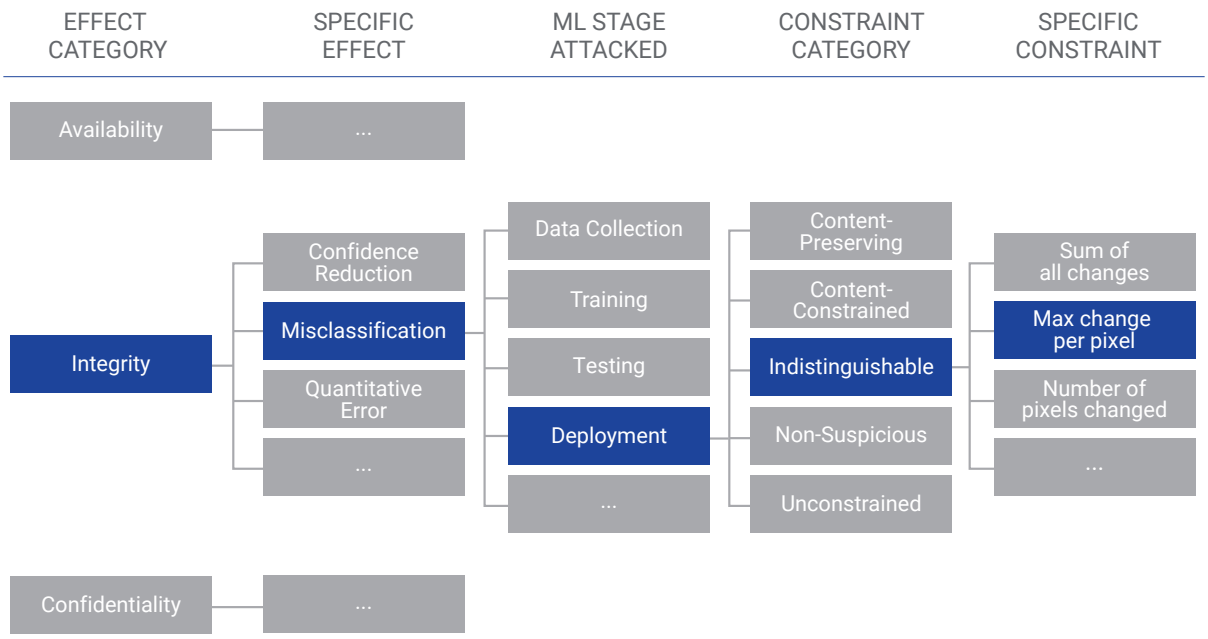
3 The Range of Possible Attacks

Attackers must make a range of choices about when and how to carry out their operations. They can direct different kinds of attacks at different stages of the machine learning process, from data collection to deployment. For example, attackers can use both data poisoning and evasion attacks to cause misclassification, but poisoning attacks target the training process whereas evasion targets the model after deployment.

Attackers can also choose how aggressive or stealthy to be. For example, evasion attacks have different degrees of subtlety.[9] At one extreme, the attackers can create whatever inputs they want, such as random squiggles to try to bypass a cellphone's facial recognition system. Those attacks are obvious to a human inspecting the images either in real-time or forensically after an attack. At the stealthier extreme, attackers may restrict themselves to changes that are indistinguishable to humans, as the Healy Hall image demonstrates. Even just within the category of indistinguishable attacks, there are many options. For example, to attack an image detection system, attackers may limit the number of pixels that can be changed, the amount of change per pixel, or the average change across all pixels, and so forth. The technical details of how these attacks can be most efficient are not necessary to gain a conceptual understanding; we mention them only to suggest that there are many options for the attackers to choose from.

FIGURE 5

Attackers have many options to choose from when they target machine learning systems. The blue path in Figure 5 shows the attack used to make Figure 3, in which Georgetown University's Healy Hall was turned into a triceratops (at least according to a machine learning system).



As Figure 5 shows, in the Healy Hall example we chose to carry out an attack on integrity that forced a system to misclassify an image, and we carried out the attack against a machine learning system that was deployed by indistinguishably manipulating an image and only slightly changing a large number of pixels. In response to different operational priorities, we might have used an alternative attack design, such as making larger changes to a smaller number of pixels or worrying less about making the original image and the manipulated image look more alike. We could have achieved a similar effect by data poisoning instead. Still other kinds of attacks would have targeted confidentiality and extracted information about the model rather than manipulating its outputs.

Adversarial attacks are powerful, counterintuitive, and subtle. There are many tools available online for carrying out these kinds of operations, which are expected to become increasingly common. That said, there is a risk of focusing too much on adversarial examples and neglecting other serious types of attacks against machine learning systems. Recognizing the range of possible threats should lead to a broader and more robust defense of machine learning systems; to best inform this defense, we need to examine the ease, persuasiveness, and severity of the threats.

4 Assessing the Threat: Ease, Pervasiveness, and Defenses

The attacks described in this report are likely to become common in the future. They are easy to conduct, the vulnerabilities they exploit are pervasive, and the attacks are difficult to defend against. The combination of these three characteristics means that managing machine learning vulnerabilities is a complex problem, even when compared to other problems in cybersecurity.

EASE OF ATTACK

Conducting attacks on machine learning systems often requires less expertise than it takes to design the models and fewer resources than it takes to train them; it is easier to destroy than to create. Just as the offense has long held an advantage in traditional cyber operations, it appears to also have the edge in machine learning for the time being.

The tools for conducting the common attacks discussed in this report have already proliferated widely. They can be found and downloaded freely from the internet and are not difficult to build. We were able to make many versions of the attack shown in Figure 3 over the course of a single afternoon. None of the versions took more than 20 lines of code and each could run in about a second.

The attacks in Figure 3 executed quickly because they avoided the hard part of machine learning: training the model. Not all attacks avoid the training stage, so some can require substantially more than a second of effort. Attackers who cannot steal the model either have to build one of their own or use trial-and-error on the victim's deployed system.[10] For example, attackers often need large amounts of time to train a model for data poisoning attacks to confirm that the poison will have the desired effect.

Even when the attacks are sometimes time-consuming, the number of actions an attacker must take can be surprisingly small.[11] Changing as little as a single data-point can sometimes be enough to have the desired negative effects on the model's performance.[12] The same is true for confidentiality attacks, where observing the inputs and outputs of a machine learning system just a few hundred or thousand times can be enough to determine how machine learning models work.[13] And, as we have seen, evasion attacks can be successful with only imperceptible changes.[14]

Of course, coding the attacks is only one part of an operation. The attacker needs information to conduct the attack and a way to launch it against the target. To do this, the three confidentiality attacks previously discussed only need to observe the inputs and outputs of the model or the deployed system. Similarly, neither data poisoning nor evasion require direct access to the target. In short, there are many opportunities for attackers to achieve their goals and the attacks themselves do not require much expertise to create. However, the difficulty of introducing them to the victim and ensuring they cause the intended malicious outcome will vary on a case-by-case basis.

Some attacks do require substantial information about the target, and obtaining that information can be somewhat complicated. For the Healy Hall attack, there was an exact copy of the model to be attacked. It was as if a crashed spy drone had been recovered and could be used to design camouflage to fool a similar drone's machine learning systems. Depending on our objective, we may not need such direct access to the target; oftentimes information about something similar to the targeted model will work. The techniques used in the Healy Hall attack can be used to simultaneously fool many different image classifiers.[15] Without knowing what model the other classifiers used, we could send them our doctored image of Healy Hall and there is a good chance they would misclassify it. Absent having the model, though, it would be harder to force these classifiers to misidentify the building as a triceratops.[16] In general, causing specific failures—such as getting a facial recognition system to misclassify someone as a specific other person—requires more information about the target model.

PERVASIVENESS

All machine learning models are susceptible to attack; different kinds of models are vulnerable in different ways. In models with few inputs, for example, the victim stands a better chance of noticing data poisoning attacks. Similarly, models with few inputs are often more robust against evasion attacks because there are fewer ways for an attacker to manipulate those inputs. In contrast, more complex models offer more opportunities for attackers. That partly explains why image recognition systems are so vulnerable: each pixel is an input, creating many manipulation opportunities for the attackers.

But simpler models that are trained on just a few data points have drawbacks, too. For example, they are more vulnerable to confidentiality attacks. If there are only five people in a training database, then each of those people contributes a lot to the tuning of parameters in the model. In models trained on millions of people, each person contributes only a little to any parameter and so information can be harder to extract via confidentiality attack.

As a result of these and other systemic weaknesses, all machine learning systems have vulnerabilities. Some of the most common examples of machine learning models are vision systems; evasion attacks against vision systems receive significant attention, as this report has shown, but there are prominent examples of attacks against audio and text systems, as well. Subtle changes can be made so that the computer "hears" whatever the attacker chooses. In voice-controlled homes or phones, attackers may gain unauthorized access. Systems that process text are also vulnerable to manipulation and evasion. For example, Twitter's AI for identifying misleading tweets about COVID-19 flagged one reading, "Do not give oxygen to the idea, which comes up with great frequency, that we are approaching some kind of strong AI"—a statement that has nothing to do with COVID-19. These failures offer a reminder of the shortcomings of machine learning systems.[17] It is easy to imagine that governments will benefit from the ability to manipulate an adversary's machine learning systems and will perceive an imperative to defend their own.

DEFENSES

Reliable defenses against these types of attacks are hard to come by, but some developments are more promising than others. Protecting information about the data sources used to train a model—to guard against membership inference, for example—is an area of comparative promise. A technique called differential privacy can mathematically limit how much information can be gleaned about any individual person or datapoint.* The designers of machine learning systems can use those techniques to manage their degree of risk and constrain the information available to those seeking to breach confidentiality.[18] Most of these techniques today force the defender to sacrifice performance for privacy and so they are not widely implemented, but future privacy-preserving techniques may be more efficient.[19] In mission critical systems or cases of extreme data sensitivity, however, these performance tradeoffs may be more acceptable. Differential privacy has other limits, too. For example, it only protects an individual's contributions to the training data and will not help obscure traits common among groups of contributors.

*Other mathematical techniques such as secure multi-party computing, homomorphic encryption, and federated learning are also promising. Those three techniques, though, solve the different problem of keeping the developer from accessing private data rather than keeping the developer's data private.

Defenses against other attacks are less promising. To guard against the other confidentiality attacks of model inversion and model extraction, the defender can reduce their vulnerability by limiting the number of times customers can use their model or by intentionally decreasing its accuracy.[20] But those steps can interfere with the business case and limit the value of machine learning; if a company makes money each time the model is used, then limiting the number of uses is not very appealing. Other approaches, like keeping the model on a classified server and making sure cleared analysts are the only ones who see its outputs, limit the risk of model stealing and model inversion but at the cost of restricting the model's use and adding security constraints.

Defending against attacks on integrity is harder still; it is a game of Whack-a-Mole where new attacks are invented and defenses are developed, and then those defenses are defeated and so on.[21] This dynamic applies both to defenses that try to detect attacks and those that try to make the models immune to them. And defending against one attack can invite others. For example, freezing a model and cutting off its access to new information means no additional data poisoning is possible, but letting it continue to update its defenses can pressure evasion attacks to evolve to keep pace.

These defenses are typically only somewhat effective and only for very highly constrained attacks of specific types.[22] For example, to guard against attacks that make imperceptible changes to a picture, a defense might be effective against attackers who limit the *average* change per pixel while not protecting against those who limit the *maximum* change per pixel.[23] A subtle change in an attacker's operations can change how effective the defense is.[†]

This is a vital and perhaps alarming point: machine learning vulnerabilities are hard to fix. Fixing them is more akin to addressing hardware vulnerabilities—which are notoriously challenging—than it is to the relative ease of patching traditional software vulnerabilities.[24] For some of the attacks discussed in this paper there is no clear solution on the horizon. The persistence of these weaknesses should prompt caution when using machine learning in national security contexts against sophisticated adversaries.

[†] There are actually many ways to calculate the average and there are many more interesting ways to keep the changes small that are more esoteric, such as using the Wasserstein distance.[25]

5 Conclusion and Recommendations

Historically, where vulnerabilities have existed in traditional cyber systems, attackers have often exploited them for nefarious or destructive ends. The same will likely be true of vulnerabilities in machine learning. This is not a call to eliminate machine learning from ongoing modernization, as vulnerabilities do exist in non-AI systems as well. Rather, it is a wake-up call: machine learning brings with it new vulnerabilities that must be understood well enough to make informed decisions about risks and investments. A few findings follow from our analysis.

MACHINE LEARNING INTRODUCES RISK IN ACQUISITION AND MODERNIZATION

Machine learning is deeply integrated into various facets of society and will likely continue to gain traction. In some cases, such as integrity attacks against movie recommendation systems, there may be relatively little incentive to attack and so the risk of using machine learning is low. In contrast, the risks to national security systems are substantial, and there are many well-resourced and highly motivated adversaries seeking to attack. A first step in assessing the risks of deploying machine learning systems in such a competitive context is understanding the range of options available to the potential attackers, which include the model stealing, model inversion, membership inference, data poisoning and evasion attacks discussed in this report. A subsequent step is understanding the defensive options that exist and their effectiveness.

NEW DEFENSES MAY ONLY OFFER SHORT-TERM ADVANTAGE

One of the perpetual questions in cybersecurity is whether the attacker or the defender has the upper hand. It is hard to answer this question until the field of machine learning cybersecurity settles on specific offensive and defensive techniques. Even then the answer may not be clear, as attackers and defenders engage one another, both sides will discover new techniques. Currently, defenses do not look promising, and many traditional cybersecurity techniques are not easily applied to machine learning. In general, attackers can move more quickly than defenders and the costs are higher to retrain a model than they are to find a new attack.

The offense-defense balance changes as machine learning systems reach different levels of model complexity. Some techniques that appear to be effective or ineffective at first behave differently when applied to more or less powerful systems. For example, some defenses that are promising for securing imaging systems that read low-resolution handwritten digits are not promising for imaging systems that are powerful enough to recognize high-resolution pictures of cars and animals.

ROBUSTNESS TO ATTACK IS MOST LIKELY TO COME FROM SYSTEM-LEVEL DEFENSES

Given the difficulty in finding reliable defenses against the wide range of attack options, systemic defenses seem essential. Defenders should assume that attackers will successfully compromise some parts of machine learning systems. To limit the damage attackers can do, we should build redundancy and increase resilience. Especially given how hard it is to fix underlying weaknesses in machine learning systems, designing architectures that maximize robustness and prevent cascading failures is key.

For instance, a commonly cited example of an attack involves placing a sticker on a stop sign that makes it appear to autonomous vehicles to be a 45 mph sign. Although this attack is possible and easy to perform, it only achieves a destructive effect if the car drives into a busy intersection. If the car has many ways to decide to stop, such as by knowing that intersections usually have stop signs, relying on lasers for collision avoidance, observing other cars stopping, or noticing high speed cross-traffic, then the risk of attack can remain low despite the car being made of potentially vulnerable machine learning components. The systemic-level defense—to not rely on just one input in making the decision to accelerate through the intersection—thwarts the attack.

THE BENEFITS TO OFFENSIVE USE OFTEN DO NOT OUTWEIGH THE COSTS

The United States is not the only country fielding AI systems, and the opportunity to exploit these vulnerabilities in adversaries' systems may be tempting. There are obvious military benefits of causing an enemy weapon to misidentify its targets or send an adversary's autonomous vehicles off course. There are also the obvious intelligence benefits of stealing adversaries' models and learning about the data they have used.

On the other hand, the United States is among the countries best positioned to benefit from progress in AI technologies. It has the most to lose if these technologies are vulnerable. Demonstrating global norms of restraint against attacking AI may be a wise stance. Even if nation-states do not adhere to global norms against attacking military AI in conflict, there may be benefits to clarifying the lines against attacking civilian systems or critical infrastructure. Clarifying rules and norms would help manage the problem of machine learning security both domestically and internationally. A posture of restraint when it comes to attacking machine learning systems may also help the United States government win the support of AI talent that national security officials have been eager to court.

Norms alone will not solve this problem. Whether or not the United States decides to pursue attacks on machine learning systems, adversaries will make their own decisions about restraint, or lack thereof. Given that machine learning's vulnerabilities are pervasive, easy to exploit, and hard to defend, managing the risks they pose is too large a task for the technology community to handle alone. It is incumbent upon policymakers to understand the threats well enough to assess the dangers that the nation, its military and intelligence arms, and its civilians face when they use machine learning. In some cases, that exposure may be acceptable, and in others, it may not. But in all cases, the management of risk must be informed by technical understanding. This primer is meant to help with that endeavor.

Endnotes

1. *Tencent Keen Security Lab Experimental Security Research of Tesla Autopilot.* (2019).
2. Morgan, F. E. et al. *Military Applications of Artificial Intelligence: Ethical Concerns in an Uncertain World.* https://apps.dtic.mil/sti/citations/AD1097313 (2020).
3. Eykholt, K. et al. Robust Physical-World Attacks on Deep Learning Models. *arXiv [cs.CR]* (2017).
4. Shumailov, I., Zhao, Y., Bates, D., Papernot, N., Mullins, R. & Anderson, R. Sponge Examples: Energy-Latency Attacks on Neural Networks. *arXiv [cs.LG]* (2020).; Tabassi, E., Burns, K. J., Hadjimichael, M., Molina-Markham, A. D. & Sexton, J. T. A taxonomy and terminology of adversarial machine learning. (2019) doi:10.6028/NIST.IR.8269-draft.
5. Gu, T., Dolan-Gavitt, B. & Garg, S. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv [cs.CR]* (2017).; Kurita, K., Michel, P. & Neubig, G. Weight Poisoning Attacks on Pre-trained Models. *arXiv [cs.LG]* (2020).
6. Saha, A., Subramanya, A. & Pirsiavash, H. Hidden Trigger Backdoor Attacks. *arXiv [cs.CV]* (2019).
7. Madry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv [stat.ML] (2017).
8. Zhang, Y. et al. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. *arXiv [cs.LG]* (2019).
9. Gilmer, J., Adams, R. P., Goodfellow, I., Andersen, D. & Dahl, G. E. Motivating the Rules of the Game for Adversarial Example Research. *arXiv [cs.LG]* (2018).
10. Shukla, S. N., Sahu, A. K., Willmott, D. & Zico Kolter, J. Hard Label Black-box Adversarial Attacks in Low Query Budget Regimes. *arXiv [cs.LG]* (2020).; Ilyas, A., Engstrom, L., Athalye, A. & Lin, J. Black-box Adversarial Attacks with Limited Queries and Information. *arXiv [cs.CV]* (2018).
11. Shafahi, A. et al. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks. in Advances in Neural Information Processing Systems 31 (eds. Bengio, S. et al.) 6103–6113 (Curran Associates, Inc., 2018).
12. Park, S., Weimer, J. & Lee, I. Resilient linear classification: an approach to deal with attacks on training data. in *Proceedings of the 8th International Conference on Cyber-Physical Systems* 155–164 (Association for Computing Machinery, 2017).
13. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K. & Ristenpart, T. Stealing machine learning models via prediction apis. in *25th ${USENIX} Security Symposium ({USENIX}$ Security 16)* 601–618 (2016).
14. Su, J., Vargas, D. V. & Sakurai, K. One Pixel Attack for Fooling Deep Neural Networks. I*EEE Trans. Evol. Comput.* 23, 828–841 (2019).
15. Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D. & McDaniel, P. The Space of Transferable Adversarial Examples. *arXiv [stat.ML]* (2017).; Papernot, N., McDaniel, P. & Goodfellow, I. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *arXiv [cs.CR]* (2016).
16. Liu, Y., Chen, X., Liu, C. & Song, D. Delving into Transferable Adversarial Examples and Black-box Attacks. *arXiv [cs.LG]* (2016).
17. Carlini, N. et al. Hidden voice commands. in *25th ${USENIX} Security Symposium ({USENIX}$ Security 16)* 513–530 (2016). Tweet from Ludwig Yeetgenstein, June 26, 2020, https://twitter.com/yeetgenstein/status/1276518982565146624.
18. Truex, S., Liu, L., Gursoy, M. E., Wei, W. & Yu, L. Effects of Differential Privacy and Data Skewness on Membership Inference Vulnerability. in *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)* 82–91 (2019).

19. Jayaraman, B. & Evans, D. Evaluating differentially private machine learning in practice. in *28th ${USENIX} Security Symposium ({USENIX}$ Security 19)* 1895–1912 (2019).; Rahman, M. A., Rahman, T., Laganière, R., Mohammed, N. & Wang, Y. Membership Inference Attack against Differentially Private Deep Learning Model. *Trans. Data Priv. 11*, 61–79 (2018).
20. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K. & Ristenpart, T. Stealing machine learning models via prediction apis. in *25th ${USENIX} Security Symposium ({USENIX}$ Security 16)* 601–618 (2016).
21. Athalye, A., Carlini, N. & Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *arXiv [cs.LG]* (2018).
22. Hartnett, G. S., Lohn, A. J. & Sedlack, A. P. Adversarial Examples for Cost-Sensitive Classifiers. *arXiv [stat. ML]* (2019).; Xie, C., Wu, Y., van der Maaten, L., Yuille, A. & He, K. Feature Denoising for Improving Adversarial Robustness. *arXiv [cs.CV]* (2018).
23. Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D. & Jana, S. Certified Robustness to Adversarial Examples with Differential Privacy. *arXiv [stat.ML]* (2018). Wong, E. & Zico Kolter, J. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv [cs.LG]* (2017).
24. Lohn, A. What do Meltdown, Spectre and RyzenFall Mean for the Future of Cybersecurity? TechCrunch (2018).
25. Wong, E., Schmidt, F. R. & Zico Kolter, J. Wasserstein Adversarial Examples via Projected Sinkhorn Iterations. *arXiv [cs.LG]* (2019).

CSET CENTER *for* SECURITY *and* EMERGING TECHNOLOGY

CSET.GEORGETOWN.EDU | CSET@GEORGETOWN.EDU