

You Don't Need Robust Machine Learning to Manage Adversarial Attack Risks

Edward Raff^{1*}, Michel Benaroch², Andrew L. Farris³

¹Crowdstrike

²Syracuse University

³Booz Allen Hamilton

edward.raff@crowdstrike.com, mbenaroc@syr.edu, Farris.Drew@bah.com

Abstract

The robustness of modern machine learning (ML) models has become an increasing concern within the community. The ability to subvert a model into making errant predictions using seemingly inconsequential changes to input is startling, as is our lack of success in building models robust to this concern. Existing research shows progress, but current mitigations come with a high cost and simultaneously reduce the model's accuracy. However, such trade-offs may not be necessary when other design choices could subvert the risk. In this article, we argue that a majority of real-world applications may not have any immediate need for models robust to adversarial machine learning (AML) by critically analyzing the literature, and our experience in designing around such constraints. This is done with an eye toward how one would then mitigate these attacks in practice, the risks for production deployment, and how those risks could be managed. In doing so we elucidate that many AML threats do not warrant the cost and trade-offs of robustness due to a low likelihood of attack or availability of superior non-ML mitigation. Our analysis also recommends cases where an actor should be concerned about AML to the degree where robust ML models are necessary for a complete deployment. Ultimately, we as a community need to better design benchmark threat models that allow research to progress with more applicability to real-world use, as a direction separate from the value in understanding the abilities/limitations of models in a "lab" threat model.

1 Introduction

Companies are increasingly concerned with adversarial attacks to their machine learning (ML) models. In adversarial attacks, a third party wishes to subvert a company's interests by using ML to trick their victims' ML models to behave in a way that injures or reflects poorly on the company. Simple examples illustrate the risk. Microsoft's Tay chatbot learned on-the-fly and was poisoned by Twitter users to produce racist tweets and producing significant backlash (Davis 2016; Wolf, Miller, and Grodzinsky 2017). Google's image classifier labeled two African Americans as "gorillas" which similarly caused public outcry (Barr 2015). Notably, both attacks were perpetrated by regular humans attempting to

confuse or find flaws in the algorithms. While these attacks are non-adversarial from a ML perspective, the significant risk of adversarial attacks stems from the thought: how dangerous could this be if vulnerability discovery is automated using ML? The risk of adversarial ML attacks is relevant to companies deploying ML models of all kinds and particularly ML security applications such as fraud, malware, and intrusion detection. Among the outcomes are fraud detection models that could be subverted, self-driving car companies that may be at risk of liability, loan applications that may create excess loss or be forced to behave in apparently discriminatory ways. Techniques for addressing or managing the risk of adversarial ML attacks are an active problem of research.

The canonical wisdom to manage the risk of adversarial attacks is to develop so-called robust ML models (Ilyas et al. 2019). A ML model is robust if a third party cannot reliably force the model to behave in a desired way. Robust ML models offer the benefit of being resistant (but not immune) to adversarial attacks. However, making ML models robust is non-trivial and involves a significant up-front training cost (Madry et al. 2018) and ongoing cost in the form of a higher error rate (Ilyas et al. 2019). Moreover, most companies for which the risk of adversarial attacks is real are not equipped to address that risk or to develop a strategy for managing it. In a sample from 28 companies in 11 industries, only 6 companies were ready to dedicate staff to building robust ML models (Siva Kumar et al. 2020). Simultaneously, most practitioners are completely unfamiliar with issues related to adversarial ML (Bieringer et al. 2022; Boenisch et al. 2021).

We argue that immediately tackling the issue of robustness is likely counterproductive to most companies. Instead, we recommend recognizing a distinction between security and robustness in practice. Security goes beyond a ML model's accuracy and involves the infrastructure around its maintenance, validation, and deployment that build confidence in a reliable process. Robustness of a ML model can leverage the same processes but is not just the process – it is the mechanisms by which adversarial ML attacks specifically are mitigated, and have a distinct cost. Our recommendation stems from the recognition that the need to deal with robustness is not uniform across companies or scenarios. More specifically, it stems from multiple factors, such as how adversarial ML works, the likelihood that an attack will

*Work originally done while at Booz Allen Hamilton
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

be deployed, the rate of attacks, and the cost of lower accuracy robust models. To wit, there are no readily available benchmarks or frameworks for researchers to study problems or solutions in the scope of an entire system.

To justify our argument we use the following strategy: First we review the relevant related literature in §2, and show that the perspective of non-ML solutions to AML problems is apparently absent from the literature. Then we review the primary types of adversarial ML attacks in use today and the threat models that describe the knowledge an attacker needs to successfully perform an adversarial attack in §3. In §4 we develop a stylized model of the cost trade-offs for the development of a robust ML model. This model then allows us to characterize, for each conjoint attack-threat model, the implied parameter values in our stylized model. By evaluating the model parameters for each attack-threat model pair, we demonstrate that adversarial attacks are less probable in most business contexts. To further justify our assumptions of risk and thus need of robust ML, in §5 we propose a number of general design choices that can be used to mitigate the risk of an AML attack. These recommendations are applied to the most prevalent and purported “real-world” AML attacks we found in §6 as case studies. Finally, §7 offers guidelines for contexts where pursuing robustness in ML models is warranted along with a standard operating procedure for managers to follow if adversarial ML is a risk to them. We present our conclusions in §8.

2 Related Work

The seminal work of Apruzzese et al. (2023) elucidated many successes and gaps between adversarial machine learning research and real-world incentives, issues, and practicalities of adversarial attacks. Their article rightfully focuses on what an AML researcher must consider in best developing a realistic attack/defense, such as the cost-driven assessment of how attacks are performed. Our work instead focuses on the corporate side, when AML attacks should and should not be considered, and what design choices a system engineer may consider to alleviate the risk of AML attacks. Other important macro-scale aspects of adversarial attacks have been discussed but not yet connected to larger system and process designs as an effective defense. (Mirsky et al. 2023) surveys the motivations and methods for how an attacker may decide and act against a victim. (Brown et al. 2018) made important observations on how the threat model of an attack could be greater than what many academics consider by going beyond the standard $\|\cdot\|_p \leq \epsilon$ restrictions. (Zhou et al. 2022) mapped adversarial attacks and defenses against lessons learned and frameworks from cybersecurity, but their focus remained on attacks and defenses based on machine learning, and not alteration of a larger system or the managerial decision process in determining if the risk is acceptable. There exist many general surveys and discussions of adversarial machine learning at many different levels, which broadly do not discuss the larger system design (Yuan et al. 2017; Hu et al. 2022; Wang et al. 2022; Li et al. 2021; Biggio and Roli 2018).

Mohseni et al. (2022) proposed a *Taxonomy of ML Safety* by looking at a safety critical design of machine learning

systems. However, we find that the survey provides no discussion on how design changes around the machine learning can mitigate the concern for attacks against the system. Deldjoo, Noia, and Merra (2021) look at adversarial attacks against recommender systems. While issues about real-world problems are discussed (e.g., attacking a real-world system must factor in the temporal nature of recommendations changing over time), no practical real-world attacks are documented “in the wild”. Wang et al. (2022) surveyed poisoning attacks against an ML model, and did not mark any examples of this occurring in real life. Though they do mention “Well-intentioned” poisoning to enforce things like copyright detection (i.e., defense for something that will be made public), they do not make the full jump to recognizing poisoning as a method of countering future data theft (i.e., defense for something intended to remain private). Their survey also does not identify any recognition of how classic cryptographic key signing can be used to mitigate the risk of poisoning attacks.

Hu et al. (2022) perform an extensive survey of model inversion attacks, and do note that Differential Privacy provides a provably secure method of mitigating these attacks. While they mention that DP often has a trade-off that may be too expensive, we note that it has had many successful uses in practice and thus provides a means for mitigating this class of attacks. The larger insight that a system can alter its design to better leverage differential privacy is not discussed. From a different perspective Paleyes, Urma, and Lawrence (2022) focuses on how to deploy modern machine learning systems and the challenges such deployments face, with adversarial machine learning being but one concern. While they enumerate the basic attack types and some notes on the risk, they provide no guidance on how to defend against such issues from either an ML or whole system design perspective.

We note as well that there has been limited discussion on ML from an institutional perspective in terms of maintenance and holistic design, but none that we are aware of that have tackled the heart of design changes to deal with AML risks. Sculley et al. (2015) talked about the technical debt of building real-world solutions, and others have talked about issues in misspecification causing inflated expectations, disappointment, and lack of trust (D’Amour et al. 2020). From a broader system design perspective of computer systems at large, seminal work by Friedman and Nissenbaum (1996) discussed a number of ways bias can be introduced or emerge as a product of the larger picture (e.g., historical context, design choices, and system usage). These works share a broad theme of non-technical mitigations to technical problems, which is applicable to AML. Though we will include such design changes, we also discuss technical solutions from outside ML to the AML problem. Others have investigated industry-specific design concerns (Kaymakci, Wenninger, and Sauer 2021), refactoring/maintenance (Tang et al. 2021; Gesi et al. 2022; Arpteg et al. 2018), and reproducibility (Forde et al. 2018; Raff and Farris 2022; Raff 2019), but do generally focus on narrow problems and do not address larger systematic changes required to achieve technical-ML goals.

3 Machine Learning Models and Robustness

3.1 Adversarial ML Attacks

Adversarial examples are samples of input for a ML classification system that are very similar to a normal input example but cause the ML system to make a different classification. Adversarial examples exploit certain properties of ML classifiers and are explicitly and purposefully identified using specific algorithms called adversarial attacks. Though the mathematics of performing AML are not key to our survey as we focus on non-ML and design solutions, we briefly review them. In most AML literature this would be a d -dimensional feature vector \mathbf{x} passed into a model $f(\cdot)$ for which there is a desired output y . The goal mean for the adversary $A(\cdot)$ who has the power to perturb the input by some p -norm threshold ϵ , such that $\|A(\mathbf{x}) - \mathbf{x}\|_p \leq \epsilon$ and achieves the goal that $f(A(\mathbf{x})) \neq y$.

Many possible targets of attack exist. In one simple scenario, attacks could allow spam or phishing attacks to go undetected by existing ML models, by forcing detection (or classification) models to make incorrect conclusions. This kind of attack can exacerbate existing cyber-security issues. In another plausible scenario, ML systems for screening credit-card charges could be fooled into classifying fraudulent transactions as non-fraudulent, allowing the adversary to cause direct financial harm and self-enrichment. Other attack scenarios could result in personally identifiable information (PII) data leaks or result in data-theft, such as replicating a company's large investments in data labeling, warehousing, data cleaning, and model building.

It is useful to group adversarial attacks into three general types, ordered based on the nature of the risk to the enterprise (Siva Kumar et al. 2020):

- Poisoning attacks seek to modify the data used to train a victim's ML algorithm so that the attacker's goals are achieved whenever a model is trained on the poisoned data. Influencing models to have very low accuracy could amount to a "Denial of Service" attack. Poisoning could also insert "backdoors" that allow the adversary to control a model by including a special key in model input, or otherwise altering the ML model's behavior.
- Inversion attacks seek to obtain information about the model itself or the data used to train it, be it by observing its behavior or by physical inspection of its parameters. They allow an adversary to create their own copy of a ML model (i.e., theft of capability) or to infer the data used in the model (e.g., PII violations and extracting individuals' data from the model).
- Evasion attacks trick a victim ML model into making an errant prediction due to what should have otherwise been a benign manipulation of the input data. The classic example of an evasion attack is how non-robust Computer Vision models can be fooled into making an incorrect, nonsensical prediction by altering a single pixel in the input image.

The real-world risk of these attacks cannot be properly accounted for without considering the threat-model, which is a description of the assumptions of information required

for the attack to operate successfully. The three general cases are:

- White-box attacks: the adversary knows everything about the victim's ML model, including the algorithm used and any defensive techniques, and has its own copy of the training data.
- Grey-box attacks: the adversary knows some information, but not all details. Their ability to interact with the system is limited in some ways.
- Black-box attacks: the adversary has only minimal access to the model, such as via an API that receives input cases and returns answers. Their ability to perturb any data is limited to data before it reaches the system. They do not know what kind of model or data is used.

3.2 Robust ML Models

A touted solution to adversarial ML attacks is to build robust ML models that are less susceptible to attacks. "Robustness" is a term widely used in the context of ML to indicate that a ML system cannot be fooled by adversarial examples¹. The exact nature of how to make ML models robust is an issue of active research. For the purposes of this survey we will use a broad definition: *robust models are ones that are not easily subverted by an adversary*. One might infer that a robust model should be more accurate on all kinds of naturally occurring data and situations, though this is not commonly the case in practice.

Obtaining robustness is non-trivial and imposes two significant costs. Making a model robust can require a significant capital expenditure. While a conventional (non-robust) ML model can cost between \$40-\$100,000 to train (Strubell, Ganesh, and McCallum 2019), creating a robust ML model requires expertise and significantly more training that can easily be 100 times to over 1000 times as expensive computationally (Madry et al. 2018). Moreover, when re-training must be done on regularly (e.g., on a quarterly basis) this cost is further amplified.

Another cost trade-off is reduced accuracy. Today, robust models are usually less accurate than non-robust models. Non-robust models tend to learn correlative, not causal, relationships that are brittle and thus susceptible to exploitation (Ilyas et al. 2019). These correlations are often useful in terms of predictive accuracy, but by their nature are also non-truths than an adversary can exploit. By contrast, robust models currently have lower accuracy as they forgo weak signals that are correlative but useful.

Considering the cost trade-offs presented by robust models, we posit that model robustness may be warranted only in a small set of circumstances. To make our case, we present a stylized model that shows the current best course of action for most firms is to maximize the accuracy of their ML application models and focus less on producing robust variants of their models.

¹e.g., see <https://www.robust-ml.org/>

4 Stylized Model of Evasive Robustness-Security Trade Offs

The following stylized model formalizes the trade-offs in cost presented by robust ML models. It enables us to compare the risk associated with adversarial attacks against a non-robust model with those against a model robust to adversarial attacks. Following this convention, we model risk exposure to an attack as $RE = (\text{probability of an attack}) \times (\text{cost consequence of the attack})$.

For simplicity, assume the following parameters:

A – accuracy rate of the normal model, i.e., $(1 - A)$ is the error rate of the normal model

p – fraction of all predictions that are adversarial attacks; we pessimistically assume that all adversarial attacks on a non-robust model are successful, and optimistically assume that all adversarial attacks on a robust model are unsuccessful

z – reduction in accuracy rate in the robust model; the accuracy rate of the robust model is $(A - z)$

$(1 - (A - z)) = (1 - A + z)$ – error rate of a robust model

c_n & c_a – cost of a normal predictive error, and cost of an adversarial predictive error

Ignoring the cost of training a robust model, the break-even point between a normal vs. robust model is $RE_{\text{normal}} = RE_{\text{robust}}$, which can be expanded in terms of our assumptions as equation 1 and then simplifies to equation 2.

$$\underbrace{c_n(1-p)(1-A)}_{\text{Normal Errors}} + \underbrace{c_a p}_{\text{Normal Victim}} = \underbrace{c_n(1-p)(1-A+z)}_{\text{Robust Errors}} + \underbrace{c_a p(1-A+z)}_{\text{Robust Victim}} \quad (1)$$

$$pc_a z + c_n = p(c_a A + c_n z) \quad (2)$$

If we assume for simplicity that $c_n = c_a$, the equality simplifies to $A - p = z$, and the break-even condition would indicate that the cost of errors on adversarial attacks, c_a , must exceed the frequency of attack multiplied by the base accuracy of the model. For example, if a normal model was 95% accurate in production use, and we believe 1% of predictions are adversarial, a robust model must be at least 95%-(95%×1%)=94.05% accurate to be attractive to build. The more frequent attacks are, the more leniency there is to the penalty z . This only considers the cost of reduced accuracy in robust models, one of the two cost trade-offs of robustness.

If we factor in the second cost trade-off and consider the training cost of a robust model, the risk exposure becomes more lopsided. A non-robust model is at risk of victimization, fraud, and other issues. A robust model presents partial mitigation to that risk and brings with it the risk that a large premium will be paid for negative net impact. Specifically, if we add the cost premium for making the model robust, denoted D_R , the break-even point between a normal vs. robust model becomes:

$$c_n(1-p)(1-A) + c_a p = D_r + (1-A+z)(c_n(1-p) - c_a p) \quad (3)$$

If we continue to assume that $c_n = c_a$, the break-even condition indicates that $c_n = \frac{D_r}{Ap-z}$ the cost of errors on adversarial attacks, c_n , must exceed the cost premium, D_R , amplified by the frequency of adversarial attacks (against the baseline accuracy) modulated by the loss in accuracy z , for a robust model to be worthwhile building. The numerator will always be < 1 , so this can only increase the costs – and notably the penalty z can push the cost negative, indicating that the un-satisfiability of the inequality due to the added training costs.

The above analyses lead us to recommend against building a robust model for most companies. The benefit of a robust model is greatest when the standard model is the least accurate. This implies that the robust model will not be effective because it will be further degraded. Of course, the recommendation may be different under certain model parameters. This highlights the importance of determining the normative risks of errors when considering a robust model. More importantly, it is essential to determine if the risk is asymmetric and realistic in order to fully define the risk exposure. Next, we review these concerns in greater detail by tailoring the stylized model to various conditions.

4.1 A Stylized Model of Cost-Benefit in Building Robust Models

There are two key factors that one may argue against our initial recommendation. We list these two below, so that we may further analyze the spectrum of scenarios and risk factors.

- *Cost of Adversary Attack.* It is not realistic to assume equal costs for normal and adversary predictive errors ($c_n = c_a$). Consider an errant approval for a loan by an ML system. In the normal context, people may voluntarily return the money, or the legal system provides a means to compel the return of capital (at some expense). In the adversarial case, an attacker may have arranged a transfer to an uncooperative jurisdiction or arranged for money laundering via the dark web (van Wegberg, Oerlemans, and van Deventer 2018). In such cases, the cost for errors may differ by orders of magnitude ($c_a \gg c_n$), and thus make the development of robust predictive models viable.
- *Rate of Adversarial Attacks.* The rate of likely attacks and the rate of their success may vary by the type of attack and threat model. A model trained on only publicly available data that lacks any PII information is unlikely to be the target of theft, as there is no competitive advantage or unique value in the data used to construct the model. Similarly, a model that is used for internal purposes that do not interact with any customer is less likely to be targeted for evasion compared to a fraud model that interacts with real (and potentially adversarial) customers.

In light of this analysis, Table 1 presents a 3×3 grid that intersects types of adversarial attacks (poisoning, inversion,

and evasion) with threat models (white-, grey- and black-box) and derives, for each attack-threat model pair, implied parameter values for our stylized model. This captures the conditions for return on investment in developing robust of ML models. As seen in Table 1, the nine possible attack-/threat model combinations are categorized into four distinct groups that inform the risk analysis process. These groups are based on the viability of the attack-threat combination and tools that exist today to mitigate that threat. We emphasize that these characterizations are with respect to the generic/average company that may wish to use machine learning, and what constitutes a practical concern/threat; it is not a statement about the attack scenario plausibility to any specific actor or use case (which may make something normally Impractical more of a realistic concern). These categories are:

- **Realistic:** the attack could be carried out with a reasonable expectation of success. The threat-model supports the attack (i.e., the attack could happen in real life), and it can be achieved with measurable impact. A robust model would be an important defensive posture in these cases.
- **Unrealistic:** The attack is not practical in most cases and unlikely to occur, absent negligence. The cost of developing a robust model is not justified.
- **Solvable:** The attack could be carried out in practice, but there are readily available techniques that can very effectively mitigate the risk without the need to deploy a robust model.
- **Impractical:** the attack could be carried out, but the information required to perform the attack is so significant that it would present an unreasonable cost for the attacker. A robust model is unwarranted in this case because the probability of an attack is low.

Attack Type	Threat Model		
	Black-Box	Gray-Box	White-Box
Poisoning	$p \approx 0$ $z = \text{Large}$ $D_R = \text{Large}$ $c_a = \text{Large}$ $RE = p \times c_a \approx 0$ Unrealistic	$p \approx 0$ $z = \text{Large}$ $D_R = \text{Large}$ $c_a = \text{Large}$ $RE = p \times c_a \approx 0$ Unrealistic	$p \approx 0$ $z = \text{Large}$ $D_R = \text{Large}$ $c_a = \text{Large}$ $RE = p \times c_a \approx 0$ Impractical
Inversion and Modeling Stealing	$p = \text{low}$ $z = \text{Low}$ $D_R = \text{Low}$ $c_a = \text{High}$ $RE = p \times c_a > 0$ Solvable	$p = \text{low}$ $z = \text{Low}$ $D_R = \text{Low}$ $c_a = \text{High}$ $RE = p \times c_a > 0$ Solvable	$p \approx 0$ $z = \text{High}$ $D_R = \infty$ $c_a = \text{High}$ $RE = p \times c_a \approx 0$ Impractical
Evasion	$p = \text{Low}$ $z = \text{Low}$ $D_R = \text{Large}$ $c_a = \text{Low-High}$ $RE = p \times c_a > 0$ Realistic	$p = \text{Medium}$ $z = \text{Medium}$ $D_R = \text{Large}$ $c_a = \text{Low-High}$ $RE = p \times c_a > 0$ Realistic	$p \approx 0$ $z = \text{Large}$ $D_R = \text{Large}$ $c_a = \text{Low-High}$ $RE = p \times c_a \approx 0$ Impractical

Table 1: Table evaluating the relative risk of a stylized model of adversarial attacks. The values in each table entry correspond to the stylized model in §4, and are inferred by the scenario and our judgment.

As shown in Table 1, white-box attacks are generally impractical because the adversary is a powerful attacker who knows everything about the victim’s ML systems and data. Such an adversary probably has easier ways to effect negative outcomes. For example, white-box evasion attacks to alter medical imaging to change a patient’s diagnosis to/from cancer instead of benign (Finlayson et al. 2019). While the thought is horrifying, the amount of effort required to access and alter information, undetected, in order to pull off such an attack is considerably more than simply altering a medical record to achieve the same result (Raff, Lantzy, and Maier 2019). Similarly, in the context of malware detection, it is easier to evade all modern anti-virus systems by employing easy, commoditized “packing” functions that obfuscate the contents of the malware, without the adversary having to rely on ML to craft undetectable malware (Aghakhani et al. 2020). In cases such as these, alternate methods for attack eliminate the need for the attacker to perform an adversarial ML attack.

In practice, the difficulty of performing a real-world white-box attack is the likely reason for the lack of observed adversarial attacks in the wild. Nonetheless, white-box attacks can happen, especially when a cybersecurity incident results in a data exfiltration event (Nadler, Aminov, and Shabtai 2019). However, this would necessarily occur after a cybersecurity incident, making a robust model a secondary line of defense – rather than primary. In summary, the most likely conditions where a robust ML model is useful are those where a company’s IT infrastructure has already been compromised.

Compared to white-box attacks, gray and black-box attacks are more reasonable, especially when dealing with computer vision. Neural networks that are pre-trained on the publically available ImageNet dataset (He et al. 2015; Russakovsky et al. 2015) are ubiquitous starting points for building computer vision systems. This makes some of the details of such a network easy to guess. Nonetheless, when researchers have evaluated the feasibility of gray-box attacks in real-world settings with imperfect knowledge, attacks are far less successful than would normally be expected. These attacks have a 33% or lower success rate, compared to 100% success rate in the white-box case (Richards et al. 2021). This significantly reduces the scope of viable attacks happening in the real world, especially when we consider that black-box attacks have even less information available. As a result, the risk of black/gray attacks depends on the type of attack. This discussion leads us to focus on gray and black-box threat models, under which evasion attacks are realistic, inversion attacks are solvable, and poison attacks are unrealistic.

Evasion attacks are the most realistic because they require less information about the victim’s ML model. In the Evasion attack scenario, only commonly available API access is required to submit data and observe an outcome. The attacker can submit multiple queries via the API, creating an attack one step at a time. This creates a trade-off for handling the evasion attack avenue: pay the training cost premium for robustness or take on greater risk. Analytically, we see why the surprising suggestion may be to delay robustness.

Inversion attacks, where information is leaked by the model, are also realistic but solvable. Like Evasion attacks, inversion attacks need only API access. However, tools to mitigate this risk exist for Inversion attacks. For attacks trying to obtain the original training data or extract PII information, a technique known as Differential Privacy (Dwork et al. 2006) provides a tool that is: (1) easy to add in an API, and (2) provides provable security to the results. While Differential Privacy comes at some cost of accuracy because it works by adding randomness to the process, it can be fine-tuned to balance between the extremes. Notably, Differential Privacy has been used successfully by the U.S. Census Bureau (Machanavajjhala et al. 2008), Google (Erlingsson, Pihur, and Korolova 2014), LinkedIn (Rogers et al. 2020), and Microsoft (Ding, Kulkarni, and Yekhanin 2017). Though not a complete solution to data theft, it can help slow down the theft process (Cheng et al. 2020). In sum, under this setting, we do not see a need to add robustness to the ML model because a different tool allows us to obtain stronger guarantees as a post-processing impact with proven industry success.

Model theft through an inversion attack is problematically unrealistic. If an adversary wishes to steal a model, wouldn't it be easier to just build their own version? Stealing a model requires time and resources, combined with the fact that the stolen model will only be as good as or worse than the original. This leaves the attacker always behind the victim in model capabilities. For example, OpenAI recently created the GPT language model (Radford et al. 2019) and licensed it to Microsoft for exclusive use \$1 billion (noa 2020a). Yet much of the capability was replicated by the open-source community and released for free a year later (Black et al. 2022).

Last are poisoning attacks. In more thorough evaluations of poisoning attacks that still favor the attacker, they are shown to be often ineffective (Radiya-Dixit et al. 2022). Similarly, there are practical options to avoiding poisoning through data oversight processes (i.e., supply-chain validation applied to your data labeling to ensure you know who is labeling and how) or rolling back to data versions before poisoning attacks became a public threat. If the attacker needs to modify significant amounts of data, their efforts are likely better spent in other ways.

5 Design Mitigations That Can Avoid the Need for Robust ML

Having elucidated a trade-off between robust and non-robust models, that is mitigated by the likelihood of an attack occurring against the average entity, we now seek to answer how such risks can be managed without relying on robust ML. This is, we argue, the most desirable outcome because it allows obtaining benefits of robustness with lesser costs. We say lesser because mitigations are not free, they could create additional friction for a user or more work for an implementer. Still, our contention is the below recommendations are better in the larger degree of certainty they provide operators to make an informed risk decision, and can confidently reduce the likelihood of attack p in the stylized model from §4. In §6 we will review several “real-world”

adversarial ML attacks that largely could have been mitigated by these recommendations. Because they are meant to be general-purpose recommendations, we avoid specific scenarios unless didactic in nature.

5.1 Poisoning Mitigation

Cryptographic Signatures of Data/Label Pairs A simple strategy we have not seen discussed in the case of poisoning is to create an auditable trail of validity. That is, the threat model of poisoning attacks is often that the attacker can alter the label of your already collected data, or the content of the image. If one augments the data labeling pipeline with a cryptographic digital signature (Goldwasser, Micali, and Rivest 1988) of the tuple (data, label) a poisoning attack's likelihood becomes significantly reduced. Any alteration of the data, or the label, will result in the signature failing to validate, and thus knowledge that an attack (or data corruption) has occurred. Systems for designing and implementing key management are widely used with NIST guidance (Barker et al. 2013).

While it may still be plausible for an attacker to poison the source of the data, it imposes considerably stronger requirements on the adversary to be effective. Either they must:

1. Create a perturbed image, which will get labeled correctly (because they alter before entry to the labeling process) and thus must require a more powerful attack to subvert a downstream model despite correct labeling.
2. Infect the labeling process, e.g., by being hired as a labeler, and generating sufficient bad labels to alter the results while also not getting detected as a nefarious labeler.
3. Somehow obtain both (1) and (2) simultaneously.

In all three cases, the attacker can only impact new data, which gives the defender the ability to roll back to a known good state as a further mitigation. In addition case (2) becomes increasingly more difficult if data is passed to multiple labelers to obtain better quality labels, which is a standard recommended practice (Ratner et al. 2020; Whitehill et al. 2009; Ratner et al. 2016).

5.2 Model Inversion Mitigation

Modify Predictive Task to Enable Better Differential Privacy Differential privacy works best when it is naturally challenging for one datum's contents to be distinguished from others. This tends to occur with increasing frequency as the amount of data used increases, but is still susceptible to outliers in the data. An option we do not see discussed to improve the conditions of differential privacy success is to re-cast the features or predictive task used in the process. Applying normalizing transformations such as the Box-Cox transform naturally make the data better behaved to a limited distribution, and outliers or rare classes can be lumped into a single “other” category to increase the probability mass of a singular event (easier to make private) than many unique or extreme values (hard to keep private). While this may reduce the utility of the model, it provides a means of engineering around limitations of differential privacy.

Air-Gaped Storage of Archival vs Active Data Separate from the use of signed data/label pairs discussed in Mitigation 5.1, a further mitigation against model theft is to use an air-gapped separated between an archival store of trusted labeled data, and a working production set of data. This working set can then be *defensively poisoned* or “watermarked” (Maini, Yaghini, and Papernot 2021; Song and Shokri 2020; Liu et al. 2018), such that if theft of the data occurred via a cyber-security incident, it may be possible to identify a third party using the stolen data.

5.3 Evasion Mitigation

Restrict Cases Where Predictions are Made While a seemingly tautological argument, one can mitigate the risk of evasion attacks by predicating the prediction on a first factor. For example, requiring sign-up with a credit card, use of an RSA key to receive API access, or other barriers to use can effectively mitigate the risk of attack. The key to such cases is to make the barrier to entry a greater risk or effort to attack than the ML model itself, and it sets a new floor to the minimum amount of effort required to attack, and thus lower risk.

Restricting predictions need not literally mean “restrict when predictions are made via a source of friction”. Another form of restriction is to use additional non-ML and human-crafted rules, so long as they are done so with the intent to limit the ability to subvert the other rule itself. This is an intrinsically easier task to do when one is already hand-crafting a business process or rule to reason through its validity, and if it could be easily subverted, is likely not a good rule to use.

Finally, restricting predictions can also mean restricting the useful lifetime of the predictions. Every attack intrinsically requires some amount of time to construct, and if the utility of the attack expires because the underlying model has changed in a non-trivial way, a significant barrier to attack effectiveness is created. For example, quarterly retraining of a production model is a slow means of expiring the useful life of an attack (essentially creating concept drift for the attacker), and given real-world misspecifications can dramatically reduce attack success (Richards et al. 2021).

Audit Predictions with Gold-Label Evaluation A practice we recommend should be done in any situation regardless of concern for adversarial attack, randomly auditing the predictions made by going through a vigorous labeling process imposes a probabilistic ceiling on the largest value of p , the probability of a prediction being attack, that may occur in practice. Critically this then allows one to more empirically apply the stylized risk framework of §4.

It is worth noting that auditing can go beyond simple input/output checks by incorporating lessons from cybersecurity & marketing: i.e., “know your customer”. For example, if a user of a machine-translation service appears to live in France, and is querying multiple models in multiple language pairs that do not include French, there is a greater risk they are performing some subversive behavior. By obtaining customer information and knowledge about intended and emergent use cases, errant and unusual behaviors can be

flagged for follow-up to validate their authenticity. Whether the flagging results in an automated response would be a factor of risk of attack success vs user friction in using the service.

6 Analysis of “Real-World” Situations

Much literature and documentation exist today on “real-world” cases of adversarial machine learning. If we take real-world to mean that the model/events under consideration occurred in a non-academic setting (i.e., a business, government, or organization that did not desire the events to happen) in an intended malicious fashion (i.e. an actual or risk of harm occurring with intentionality from the perpetrator), these conditions are often not satisfied. To emphasize this, we survey a number of published surveys, papers, and publicly documented examples of allegedly real-world cases of adversarial machine learning. In doing so we document a number of issues that occur that prevent this, and pair with them how to design mitigations from §5 could have alleviated risk. We categorize the potential issues that would limit the attack’s risk as:

1. The work involves a threat model where the adversary must either choose to forgo an easier alternative to achieve the same goals.
2. The work involves a threat model where considering a real-world motivation and goals of an attacker, the attacker’s goals would not be satisfied. In such a case, there is no reason to perform the attack.
3. The defender can use existing techniques *outside of machine learning* to largely mitigate the likelihood of being attacked or attack success.
4. The attack has occurred in a purely academic context with curated datasets, and does not consider the scope of a full system.
5. The attack was performed by academics, but in a manner simulating real world situations - or even against real production systems, but did not occur with malice. While the attack can take place, its not clear who would want to perform the attack in real-life situations, and what the actual level of risk is.
6. The attack was performed by a third party and disclosed to the victim as a part of “vulnerability disclosure”, and the victim took remediating actions. There are clear abilities and reasons for an adversary to perform the attack, but it might not have been confirmed in the wild.

Our case studies are derived primarily from MITRE Atlas case-study list of real-world adversarial attacks². We filter from this list any case study that either: 1) does not have any other reference to the event with details. Or 2) Relied on human-only efforts to perform the attack (e.g., the Microsoft Tay example). We augment this with examples of notable or highly cited works that purport to be “practical” or “real-world” examples of adversarial attacks. This leaves us with eight examples as summarized in Table 2. We note that as shown, we do not have examples of Gray-Box inversion or

²<https://atlas.mitre.org/studies/>

Poisoning attacks, or black-box examples of poisoning attacks, despite our efforts to find examples of these situations explicitly. We speculate this is due to the difficulty of such situations as noted in Table 1.

We now go through each case study and briefly summarize it, the issues with its realism per our six issue types, and how the scenario could have been remediated. Such remediation may not be perfect but highlights what we believe would be the most time/cost-efficient method of addressing the risk of adversarial attack. In all cases we find that robust ML methods appear to be most effective when either 1: a prior cyber-security event resulted in data theft, making white-box attacks an enhanced risk, or 2: the model requires deployment to end-users where a motivated adversary can reverse-engineer the details of the model from the deployed executable, allowing effective white-box attack. In all other situations, we find that a more thorough red-team style analysis and the mitigation strategies from §5 could be sufficient. When appropriate, we make note of any particular “take-away” lesson from each case study.

Malware Evasion via “Good Strings” Situation: Anti-Virus product Cylance has a machine-learning-based detector and white-list used to avoid false positives. The white list could be reverse engineered from the product, and then tokens used by the white list inserted into malicious files. This resulted in benign predictions (?).

Issues: Issue 6.

Remediation: Robust machine learning methods are one of the only viable options in this scenario and are reported to be a part of the mitigation used. While other techniques could have helped, the fundamental issues that enable the tack are challenging to mitigate any other way.

Take away: The attack works in particular because AV companies make their products available to home users, giving anyone sufficient access to perform the attack. Notably, this is also a case where better literature may have helped, as it was a rediscovery of the “good word” attack on spam filters that has a number of potential mitigations (Lowd and Meek 2005; Jorgensen, Zhou, and Inge 2008; Fleshman et al. 2019; Incer et al. 2018).

Machine Translation Model Theft Situation: A machine-translation model can be replicated by querying the product with sentences to gain examples in another language, and then a replicate model can be trained that matches the performance of the original service (Wallace, Stern, and Song 2020).

Issues: Issue 4, Issue 5, Issue 2.

Remediation: Mitigation 5.3 None is needed as the attack considers only one pair of languages, and not the over 100 languages that such products support³. Considering all pairs of languages would require running the attack over 5000 times and collecting original real sentence data for each language to then translate. At this scale of effort, there is little

³<https://translate.google.com/intl/en-GB/about/languages/>

reason to steal the model rather than build their own translation pipeline from the ground up.

Take away: The effort needed to perform an inversion attack needs to be obviously less than the effort to build a product/system to perform the same end goal in a natural manner. If there is the possibility that the attack is of comparable cost, but also carries with it a risk of legal repercussions and inability to compete long-term, then the total cost accounting for risk and opportunity cost is likely higher.

Facial Recognition Dataset Leak Situation: The training data and code for a facial recognition service were obtainable by anyone because a web service was improperly configured. This created the potential for white-box attacks if previously exploited (Whittaker 2020; Cameron, Mehrotra, and Wodinsky 2020).

Issues: Issue 6

Remediation: Mitigation 5.2, Mitigation 5.2: Robust machine learning methods should become part of the solution, but risk could have been reduced by having a watermarked “live” dataset, with the original un-altered data air-gapped from the internet. Better design by only allowing a limited query response (i.e., “match/no-match”) could have further reduced risk.

Take away: The white-box attack threat and need for robust ML was a secondary defense caused by a lapse in basic cyber-security.

Face Detection Avoiding “Hat” Situation: A method is proposed to print out an “adversarial patch” (a piece of printed paper with distorted content) and place it on a hat, such that wearing the hat inhibits facial recognition systems (Komkov and Petiushko 2021).

Issues: Issue 4, Issue 2, Issue 1.

Remediation: Mitigation 5.3, Mitigation 5.3: The attack does not take into account how a system would be used, and their own results show efficacy drops significantly when applied to other recognition models. Simply using a dynamic threshold, adding cropping, or a human pre-processor to crop out the obvious sticker, would mitigate the attack. Notably, if the goal is to avoid recognition, *the overt sticker itself signals to the other party that the individual is trying to hide their identity* — and may thus draw more scrutiny than otherwise.

Take away: Playing out the “game” of how a hypothetical larger system may react to an attack may itself mitigate the attack. By trying to avoid detection via attacking the system, the attacker may yield the inverse effect of making their presence more obvious. This observation applies to other notable works that attempt to avoid detection via ostentatious garments (Wu et al. 2020).

Spam Filter Evasion Situation: A spam filter provided by the company ProofPoint was evaded by observing metadata the product places in emails, building a copy-cat model, and performing a transfer attack against the deployed model (Intelligentia 2024).

Issues: Issue 2, Issue 6.

Remediation: Mitigation 5.3 None, in particular, are needed, as multiple other parts of the full detection sys-

Attack Type	Mitigations	Threat Model		
		Black-Box	Gray-Box	White-Box
Poisoning	Cryptographic Signatures of Data/Label pairs Modify Predictive Tasks to			*Face Detection "hat" *Facial Recognition Leak
Inversion and Model Stealing	Enable Better Differential Privacy, Air-Gaped Storage of Archival vs Active Data	*Translation model Theft *Search Result Copying		*Audio Speaker Verification
Evasion	Restrict Predictions Use, Audit Predictions	*Government Tax Theft	*Spam Filter	**"Good Strings" Evasion

Table 2: Summary of the threat model used for each case study. The mitigations proposed in §5 reduce the attack likelihood p in all threat models.

tem were not attacked (noa 2020b), leaving the system as a whole still functional and low risk.

China Government Tax Office Theft Situation: A real-world government facility in China used facial recognition as a means of validating invoices for payment. Attackers stole photos of other people, and used deep-fakes to mimic responsiveness and validate as the stolen party’s identity to invoice the government using the stolen identity. \$77 million was fraudulently obtained (Olson 2021).

Issues: Issue 3.

Remediation: Mitigation 5.3: While not perfect, the attack could have been significantly mitigated by using more challenging biometric authentication like fingerprints. An even better option would be to require in-person registration and issuance of a cryptographic key to sign and verify identity. Multiple similar methods could be combined.

Take away: A system was needlessly made more vulnerable to attack by employing facial identification via machine learning, when other alternatives are more reliable in isolation or even in combination with facial identification.

Search Result Copying Situation: Google provided evidence that Microsoft’s Bing was copying search results in 2011 by tracking search queries in browsers and extensions (Singel 2011).

Issues: None.

Remediation: No remediations were apparently needed. While Microsoft denies they were copied in the manner described, no legal action was taken (noa 2011). The long-term result has been little to no discernable impact on Google’s market dominance⁴.

Take away: Poisoning “attacks” are useful from the defender’s perspective to gain information about whether or not the information is being stolen/used by others. However, no ML was necessary in this case from the attacker or defender, despite the subject (recommender systems via search) being an intrinsic ML problem.

Audio Practical Speaker Verification Situation: A system for user verification by recognizing an individual’s speech is used attacked, by creating a universal perturbation

that can be played while speaking — and fool the system into accepting a false speaker as valid (Zhang et al. 2021).

Issues: Issue 4, Issue 3, Issue 2.

Remediation: Mitigation 5.3, Mitigation 5.2: The attack appears to require a room of known size and content, with a specific distance between speaker and microphone. More broadly, the attack can be defeated by white-listing to pre-registered devices (e.g., a specific phone number) that adds another layer of defense to the system. We note as well the setup already includes mitigation against leaking information about whom is authorized by requiring the use of a random phrase to be spoken, rather than a user-specified one.

Take away: While impressive, the constraints apparently necessary to make the attack work in an increasingly realistic physical conditions do not inform how the larger process can be changed to mitigate threats.

7 When and Why Should we Focus on Robustness

Given our current analysis, it may seem that there is little reason to deal with the robustness question. This is wrong and not our message. Robustness is a challenge to implement, but it’s also a challenge for attackers to develop attacks in the real world, and the failures that must occur to make an attack more likely (data leakages or cyber security incidents) also make many adversarial attacks redundant. But, given sufficient time and resources, an adversary will be able to successfully deploy an adversarial ML attack. If a company focuses only on standard security and improved ML modeling, the adversarial attack will eventually become the lowest-effort attack vector. This means that the risk trade-off will change over time, and robustness should be on a long-term roadmap for companies at a minimum.

That said, our results do present important themes of when an adversarial attack is an especially high risk, and so robustness of ML methods should be a primary concern. Adversarial ML attacks are a special risk for governments, for whom the adversaries are literal nation-states with enormous resources to pursue many attack vectors simultaneously, as well as banks and financial institutions for which attacks are a continuous threat due to the high potential reward. In addition, companies that produce ML models as

⁴<https://www.statista.com/statistics/216573/worldwide-market-share-of-search-engines/>

a part of a software supply chain, where their models will be used by customers down-stream (noa 2021) and have obtained significant commercial success. While such attacks are currently a fashion of classic cyber-security issues, a realistic threat is for a hacker to compromise the supply chain and poison/steal/evade the ML models in use, so that they can stealthily influence or attack the downstream users of this system.

This also explains why other targets, like health institutions, do not yet seem to suffer from adversarial ML attacks. Hospitals' payroll, accounts receivable, and payable, are attack vectors that may use ML-based fraud detection. But attacks today are primarily focused around ransomware (Mansfield-Devine 2016; Spence et al. 2018), as the benefit to an adversarial attack is low, and the effort high, for an attacker today.

For institutions that satisfy being either (1) high-value reward for attackers if successful, (2) exist as a provider in a ML supply chain, or (3) are a government entity or provider to a government entity, we recommend the following three-part strategy:

- **Machine Learning Risk Assessments:** A critical component to managing adversarial attacks is to develop a better understanding of the models currently used. Using the expertise of its employees or an external contractor/specialist, companies should review applications that integrate ML and catalog plausible attacks and the circumstances under which these attacks may occur. A risk assessment of these models will illuminate which attack types and threat models the client is susceptible to. Once cataloged, a course of action can be identified to mitigate or prevent these risks.
- **Robust Machine Learning Mitigations:** Once the highest risk models have been identified, more advanced machine learning development can occur to improve the model's robustness. General purpose techniques for robustness have been improving each year, and are currently reasonably effective to apply in cases like computer vision (Carlini et al. 2022; Nie et al. 2022). However, by incorporating knowledge about the specific problem being solved, it is possible to build significantly more robust defenses at a lower total cost. This has been shown successfully in a defense being effective for multiple years in computer vision (Raff et al. 2019) and for malware detection (Fleshman et al. 2019).
- **Extrinsic Risk Reduction:** Finally, we note that many avenues of reducing adversarial attacks involve no machine learning at all. Instead, changes in process or environmental factors can reduce the cost of being attacked and the risk of attacks occurring. For example, the aforementioned example of Microsoft's Tay chatbot was allowed to update and redeploy the model based on live Twitter data, without any human signoff. Instead, a process for curating the data coming in, and reviewing new model updates before deployment, would have significantly deterred the risk. The cost of such reviews are ultimately minor compared to the cost of both the public relations fallout, and the cost of developing a robust ver-

sion of Tay.

Another example is that current laws allow some potential recourse, but no clear answers, on the liability and legal procedures around adversarial ML (Shankar et al. 2018). Companies can identify the changes that would simplify and support a health ecosystem of ML providers and risks so that companies can operate with confidence.

As researchers, we must note that little academic work exists on all three of these key factors. They are critical open questions that we must tackle, with direct and immediate real-world impact. These issues touch many areas of research, including Computer-Human Interaction (how will people use/misuse a system), natural language processing (how can a standard operating procedure be ingested/automatically analyzed), multi-agent systems (how will other companies/actors respond in the environment to a proposed change), and even economics (what motivates the bad actors and what would disincentive the acts who would target me).

7.1 The Intrinsic Value of Attack/Defense Research

Beyond the risk analysis we have performed, we make special note that this should not be seen as a dismissive article against research in adversarial attacks and defenses. Indeed, we argue that, absent any real-world attacks occurring, the questions are of a fundamentally important scientific nature. They speak to questions about intrinsic user trust in a system, that such innocuous changes can cause dramatic deviations from expectations, and speak to a fundamental scientific question: what do our methods learn and why? We believe and argue that this is an ever-green argument for this research direction to continue. But it is not, by definition, fertile ground for improving real-world deployments and risks.

8 Conclusions

Our work elucidates the position that not all situations require robust machine learning to defend against adversarial attacks, and that a larger risk assessment should be performed. In real-life deployments, the cost of adding robustness may exceed its benefits. This insight is formalized with a stylized model, from which we infer the kinds of attack scenarios that most businesses should be concerned with. To this point, we have cataloged a number of design changes that can be employed to mitigate the risks of adversarial attacks without incurring the difficulty and cost of building robust models and reviewed multiple publicly documented "real-world" adversarial cases to identify where these design mitigations could have been applied. In performing this analysis, we have identified common themes in cases where such robustness is necessary. Our hope is that this article will serve as a realization that the question of secure ML is broader than just ML itself, but how ML systems are designed in the context of a real-world problem to be solved. Our community currently lacks the tools, frameworks, infrastructure, and even publication homes to foster and grow this kind of real-world-oriented research that should spring from the adversarial machine learning discovery.

References

2011. Google accuses Bing of 'copying' its search results. *BBC News*.
- 2020a. Microsoft teams up with OpenAI to exclusively license GPT-3 language model.
- 2020b. Response to CVE-2019-20364 | Proofpoint US.
2021. Defending Against Software Supply Chain Attacks. Technical report, National Institute of Standards and Technology.
- Aghakhani, H.; Gritti, F.; Mecca, F.; Lindorfer, M.; Ortolani, S.; Balzarotti, D.; Vigna, G.; and Kruegel, C. 2020. When Malware is Packin' Heat; Limits of Machine Learning Classifiers Based on Static Analysis Features. In *Proceedings 2020 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society. ISBN 978-1-891562-61-7.
- Apruzzese, G.; Anderson, H. S.; Dambra, S.; Freeman, D.; Pierazzi, F.; and Roundy, K. A. 2023. "Real Attackers Don't Compute Gradients": Bridging the Gap between Adversarial ML Research and Practice. In *Proceedings of the 1st IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*.
- Arpteg, A.; Brinne, B.; Crnkovic-Friis, L.; and Bosch, J. 2018. Software Engineering Challenges of Deep Learning. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 50–59.
- Barker, E.; Smid, M.; Branstad, D.; and Chokhani, S. 2013. A Framework for Designing Cryptographic Key Management Systems. Technical Report NIST Special Publication (SP) 800-130, National Institute of Standards and Technology.
- Barr, A. 2015. Google Mistakenly Tags Black People as 'Gorillas,' Showing Limits of Algorithms. *Wall Street Journal*.
- Bieringer, L.; Grosse, K.; Backes, M.; Biggio, B.; and Krombholz, K. 2022. Industrial practitioners' mental models of adversarial machine learning. 97–116. ISBN 978-1-939133-30-4.
- Biggio, B.; and Roli, F. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84: 317–331. ArXiv: 1712.03141.
- Black, S.; Biderman, S.; Hallahan, E.; Anthony, Q.; Gao, L.; Golding, L.; He, H.; Leahy, C.; McDonnell, K.; Phang, J.; Pieler, M.; Prashanth, U. S.; Purohit, S.; Reynolds, L.; Tow, J.; Wang, B.; and Weinbach, S. 2022. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. ArXiv:2204.06745 [cs].
- Boenisch, F.; Battis, V.; Buchmann, N.; and Poikela, M. 2021. "I Never Thought About Securing My Machine Learning Systems": A Study of Security and Privacy Awareness of Machine Learning Practitioners. In *Mensch und Computer 2021*, MuC '21, 520–546. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-8645-6.
- Brown, T. B.; Carlini, N.; Zhang, C.; Olsson, C.; Christiano, P.; and Goodfellow, I. J. 2018. Unrestricted Adversarial Examples. *arXiv preprint*. ArXiv: 1809.08352v1.
- Cameron, D.; Mehrotra, D.; and Wodinsky, S. 2020. We Found Clearview AI's Shady Face Recognition App.
- Carlini, N.; Tramer, F.; Krishnamurthy, Dvijotham; and Kolter, J. Z. 2022. (Certified!!) Adversarial Robustness for Free! ArXiv:2206.10550 [cs].
- Cheng, Z.; Li, Z.; Zhang, J.; and Zhang, S. 2020. Differentially Private Machine Learning Model against Model Extraction Attack. In *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 722–728.
- D'Amour, A.; Heller, K.; Moldovan, D.; Adlam, B.; Alipanahi, B.; Beutel, A.; Chen, C.; Deaton, J.; Eisenstein, J.; Hoffman, M. D.; Hormozdiari, F.; Houlisby, N.; Hou, S.; Jerfel, G.; Karthikesalingam, A.; Lucic, M.; Ma, Y.; McLean, C.; Mincu, D.; Mitani, A.; Montanari, A.; Nado, Z.; Nataraajan, V.; Nielson, C.; Osborne, T. F.; Raman, R.; Ramasamy, K.; Sayres, R.; Schrouff, J.; Seneviratne, M.; Sequeira, S.; Suresh, H.; Veitch, V.; Vladymyrov, M.; Wang, X.; Webster, K.; Yadlowsky, S.; Yun, T.; Zhai, X.; and Sculley, D. 2020. Underspecification Presents Challenges for Credibility in Modern Machine Learning. ArXiv: 2011.03395.
- Davis, E. 2016. AI amusements: the tragic tale of Tay the chatbot. *AI Matters*, 2(4): 20–24.
- Deldjoo, Y.; Noia, T. D.; and Merra, F. A. 2021. A Survey on Adversarial Recommender Systems: From Attack/Defense Strategies to Generative Adversarial Networks. *ACM Computing Surveys*, 54(2): 35:1–35:38.
- Ding, B.; Kulkarni, J.; and Yekhanin, S. 2017. Collecting Telemetry Data Privately. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In Halevi, S.; and Rabin, T., eds., *Theory of Cryptography*, Lecture Notes in Computer Science, 265–284. Berlin, Heidelberg: Springer. ISBN 978-3-540-32732-5.
- Erlingsson, ; Pihur, V.; and Korolova, A. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, 1054–1067. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-2957-6.
- Finlayson, S. G.; Bowers, J. D.; Ito, J.; Zittrain, J. L.; Beam, A. L.; and Kohane, I. S. 2019. Adversarial attacks on medical machine learning. *Science (New York, N.Y.)*, 363(6433): 1287–1289.
- Fleshman, W.; Raff, E.; Sylvester, J.; Forsyth, S.; and McLean, M. 2019. Non-Negative Networks Against Adversarial Attacks. *AAAI-2019 Workshop on Artificial Intelligence for Cyber Security*. ArXiv: 1806.06108.
- Forde, J.; Head, T.; Holdgraf, C.; Panda, Y.; Perez, F.; Nalvarte, G.; Ragan-kelley, B.; and Sundell, E. 2018. Reproducible Research Environments with repo2docker. In *Reproducibility in ML Workshop, ICML'18*.

- Friedman, B.; and Nissenbaum, H. 1996. Bias in Computer Systems. *ACM Trans. Inf. Syst.*, 14(3): 330–347. Publisher: Association for Computing Machinery Place: New York, NY, USA.
- Gesi, J.; Liu, S.; Li, J.; Ahmed, I.; Nagappan, N.; Lo, D.; de Almeida, E. S.; Kochhar, P. S.; and Bao, L. 2022. Code Smells in Machine Learning Systems. ArXiv:2203.00803 [cs].
- Goldwasser, S.; Micali, S.; and Rivest, R. L. 1988. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2): 281–308.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hu, H.; Salcic, Z.; Sun, L.; Dobbie, G.; Yu, P. S.; and Zhang, X. 2022. Membership Inference Attacks on Machine Learning: A Survey. *ACM Computing Surveys*, 54(11s): 235:1–235:37.
- Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial Examples Are Not Bugs, They Are Features. In *NeurIPS*. ArXiv: 1905.02175.
- Incer, I.; Theodorides, M.; Afroz, S.; and Wagner, D. 2018. Adversarially Robust Malware Detection Using Monotonic Classification. In *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*, 54–63. New York, NY, USA: ACM. ISBN 978-1-4503-5634-3. Series Title: IWSPA '18.
- Intelligentia, N. 2024. Mitre Atlas: Proofpoint evasion – A case study for cisos - why do ai.
- Jorgensen, Z.; Zhou, Y.; and Inge, M. 2008. A Multiple Instance Learning Strategy for Combating Good Word Attacks on Spam Filters. *J. Mach. Learn. Res.*, 9: 1115–1146. Publisher: JMLR.org.
- Kaymakci, C.; Wenninger, S.; and Sauer, A. 2021. A Holistic Framework for AI Systems in Industrial Applications. In Ahlemann, F.; Schütte, R.; and Stieglitz, S., eds., *Innovation Through Information Systems*, Lecture Notes in Information Systems and Organisation, 78–93. Cham: Springer International Publishing. ISBN 978-3-030-86797-3.
- Komkov, S.; and Petiushko, A. 2021. AdvHat: Real-World Adversarial Attack on ArcFace Face ID System. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 819–826. ISSN: 1051-4651.
- Li, D.; Li, Q.; Ye, Y. F.; and Xu, S. 2021. Arms Race in Adversarial Malware Detection: A Survey. *ACM Computing Surveys*, 55(1): 15:1–15:35.
- Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018. Trojaning Attack on Neural Networks. In *Network and Distributed Systems Security (NDSS) Symposium*.
- Lowd, D.; and Meek, C. 2005. Good Word Attacks on Statistical Spam Filters. In *Conference on email and anti-spam (CEAS)*, 125–132.
- Machanavajjhala, A.; Kifer, D.; Abowd, J.; Gehrke, J.; and Vilhuber, L. 2008. Privacy: Theory meets Practice on the Map. In *2008 IEEE 24th International Conference on Data Engineering*, 277–286. ISSN: 2375-026X.
- Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*. ArXiv: 1802.10217.
- Maini, P.; Yaghini, M.; and Papernot, N. 2021. Dataset Inference: Ownership Resolution in Machine Learning. In *ICLR*. ArXiv: 2104.10706v1.
- Mansfield-Devine, S. 2016. Ransomware: taking businesses hostage. *Network Security*, 2016(10): 8–17.
- Mirsky, Y.; Demontis, A.; Kotak, J.; Shankar, R.; Gelei, D.; Yang, L.; Zhang, X.; Pintor, M.; Lee, W.; Elovici, Y.; and Biggio, B. 2023. The Threat of Offensive AI to Organizations. *Computers & Security*, 124: 103006.
- Mohseni, S.; Wang, H.; Xiao, C.; Yu, Z.; Wang, Z.; and Yadawa, J. 2022. Taxonomy of Machine Learning Safety: A Survey and Primer. *ACM Computing Surveys*. Just Accepted.
- Nadler, A.; Aminov, A.; and Shabtai, A. 2019. Detection of malicious and low throughput data exfiltration over the DNS protocol. *Computers & Security*, 80: 36–53.
- Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; and Anandkumar, A. 2022. Diffusion Models for Adversarial Purification. In *Proceedings of the 39th International Conference on Machine Learning*, 16805–16827. PMLR. ISSN: 2640-3498.
- Olson, P. 2021. Faces Are the Next Target for Fraudsters. *Wall Street Journal*.
- Paley, A.; Urma, R.-G.; and Lawrence, N. D. 2022. Challenges in Deploying Machine Learning: a Survey of Case Studies. *ACM Computing Surveys*. Just Accepted.
- Radiya-Dixit, E.; Hong, S.; Carlini, N.; and Tramer, F. 2022. Data Poisoning Won't Save You From Facial Recognition.
- Radrof, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. Technical report. Publication Title: OpenAI Technical Report.
- Raff, E. 2019. A Step Toward Quantifying Independently Reproducible Machine Learning Research. In *NeurIPS*. ArXiv: 1909.06674.
- Raff, E.; and Farris, A. L. 2022. A Siren Song of Open Source Reproducibility. In *ML Evaluation Standards Workshop at ICLR 2022*.
- Raff, E.; Lantzy, S.; and Maier, E. J. 2019. Dr. AI, Where Did You Get Your Degree? In Koch, F.; Koster, A.; Riaño, D.; Montagna, S.; Schumacher, M.; ten Teije, A.; Guttman, C.; Reichert, M.; Bichindaritz, I.; Herrero, P.; Lenz, R.; López, B.; Marling, C.; Martin, C.; Montani, S.; and Wiratunga, N., eds., *Artificial Intelligence in Health*, 76–83. Cham: Springer International Publishing. ISBN 978-3-030-12738-1.
- Raff, E.; Sylvester, J.; Forsyth, S.; and McLean, M. 2019. Barrage of Random Transforms for Adversarially Robust Defense. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6528–6537. Long Beach, CA.

- Ratner, A.; Bach, S. H.; Ehrenberg, H.; Fries, J.; Wu, S.; and Ré, C. 2020. Snorkel: rapid training data creation with weak supervision. *The VLDB Journal*, 29(2): 709–730.
- Ratner, A. J.; De Sa, C. M.; Wu, S.; Selsam, D.; and Ré, C. 2016. Data Programming: Creating Large Training Sets, Quickly. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*, 3567–3575. Curran Associates, Inc.
- Richards, L. E.; Nguyen, A.; Capps, R.; Forsythe, S.; Matuszek, C.; and Raff, E. 2021. Adversarial Transfer Attacks With Unknown Data and Class Overlap. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security (AISec '21)*. Association for Computing Machinery. ArXiv: 2109.11125.
- Rogers, R.; Subramaniam, S.; Peng, S.; Durfee, D.; Lee, S.; Kancha, S. K.; Sahay, S.; and Ahammad, P. 2020. LinkedIn’s Audience Engagements API: A Privacy Preserving Data Analytics System at Scale. ArXiv:2002.05839 [cs].
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.
- Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.-F.; and Dennison, D. 2015. Hidden Technical Debt in Machine Learning Systems. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, 2503–2511. Cambridge, MA, USA: MIT Press. Series Title: NIPS’15.
- Shankar, R.; Kumar, S.; Brien, D. R. O.; and Albert, K. 2018. Law and Adversarial Machine Learning. *arXiv*. ArXiv: 1810.10731v2.
- Singel, R. 2011. Google catches bing copying; Microsoft says “so what?”.
- Siva Kumar, R. S.; Nyström, M.; Lambert, J.; Marshall, A.; Goertzel, M.; Comissioneru, A.; Swann, M.; and Xia, S. 2020. Adversarial Machine Learning-Industry Perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, 69–75.
- Song, C.; and Shokri, R. 2020. Membership Encoding for Deep Learning. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 344–356. New York, NY, USA: ACM. ISBN 978-1-4503-6750-9.
- Spence, N.; Bhardwaj, N.; Paul, D.; and Coustasse, A. 2018. Ransomware in Healthcare Facilities: A Harbinger of the Future? *Management Faculty Research*.
- Strubell, E.; Ganesh, A.; and McCallum, A. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650. Florence, Italy: Association for Computational Linguistics.
- Tang, Y.; Khatchadourian, R.; Bagherzadeh, M.; Singh, R.; Stewart, A.; and Raja, A. 2021. An Empirical Study of Refactorings and Technical Debt in Machine Learning Systems. In *Proceedings of the 43rd International Conference on Software Engineering, ICSE '21*, 238–250. Madrid, Spain: IEEE Press. ISBN 978-1-4503-9085-9.
- van Wegberg, R.; Oerlemans, J.-J.; and van Deventer, O. 2018. Bitcoin money laundering: mixed results? An explorative study on money laundering of cybercrime proceeds using bitcoin. *Journal of Financial Crime*, 25(2): 419–435. Publisher: Emerald Publishing Limited.
- Wallace, E.; Stern, M.; and Song, D. 2020. Imitation Attacks and Defenses for Black-box Machine Translation Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5531–5546. Online: Association for Computational Linguistics.
- Wang, Z.; Ma, J.; Wang, X.; Hu, J.; Qin, Z.; and Ren, K. 2022. Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems. *ACM Computing Surveys*. Just Accepted.
- Whitehill, J.; Wu, T.-f.; Bergsma, J.; Movellan, J.; and Ruvolo, P. 2009. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.
- Whittaker, Z. 2020. Security lapse exposed Clearview AI source code.
- Wolf, M. J.; Miller, K. W.; and Grodzinsky, F. S. 2017. Why We Should Have Seen That Coming: Comments on Microsoft’s Tay “Experiment,” and Wider Implications. *The ORBIT Journal*, 1(2): 1–12.
- Wu, Z.; Lim, S.-N.; Davis, L. S.; and Goldstein, T. 2020. Making an Invisibility Cloak: Real World Adversarial Attacks on Object Detectors. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, volume 12349, 1–17. Cham: Springer International Publishing. ISBN 978-3-030-58547-1 978-3-030-58548-8. Series Title: Lecture Notes in Computer Science.
- Yuan, X.; He, P.; Zhu, Q.; Bhat, R. R.; and Li, X. 2017. Adversarial Examples: Attacks and Defenses for Deep Learning. *arXiv*. ArXiv: 1712.07107.
- Zhang, W.; Zhao, S.; Liu, L.; Li, J.; Cheng, X.; Zheng, T. F.; and Hu, X. 2021. Attack on Practical Speaker Verification System Using Universal Adversarial Perturbations. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2575–2579. ISSN: 2379-190X.
- Zhou, S.; Liu, C.; Ye, D.; Zhu, T.; Zhou, W.; and Yu, P. S. 2022. Adversarial Attacks and Defenses in Deep Learning: from a Perspective of Cybersecurity. *ACM Computing Surveys*. Just Accepted.