Is Tricking a Robot Hacking?

Ryan Calo | rcalo@uw.edu
University of Washington School of Law

Ivan Evtimov | ie5@cs.washington.edu
University of Washington Paul G. Allen School of Computer Science & Engineering

Earlence Fernandes | earlence@cs.washington.edu
University of Washington Paul G. Allen School of Computer Science & Engineering

Tadayoshi Kohno | yoshi@cs.washington.edu
University of Washington Paul G. Allen School of Computer Science & Engineering

David O'Hair | dohair@uw.edu
University of Washington School of Law

Legal Studies Research Paper No. 2018-05

Is Tricking A Robot Hacking?

Ryan Calo, Ivan Evtimov, Earlence Fernandes, Tadayoshi Kohno, David O'Hair
Tech Policy Lab
University of Washington

The term "hacking" has come to signify breaking into a computer system.[1] Lawmakers crafted penalties for hacking as early as 1986 in supposed response to the movie *War Games* three years earlier in which a teenage hacker gained access to a military computer and nearly precipitated a nuclear war. Today a number of local, national, and international laws seek to hold hackers accountable for breaking into computer systems to steal information or disrupt their operation; other laws and standards incentivize private firms to use best practices in securing computers against attack.

The landscape has shifted considerably from the 1980s and the days of dial-ups and mainframes. Today most people carry around the kind of computing power available to the United States military at the time of *War Games* in their pockets. People, institutions, and even everyday objects are connected via the Internet. Driverless cars roam highways and city streets. Yet in an age of smartphones and robots, the classic paradigm of hacking, in the sense of unauthorized access to a protected system, has sufficed and persisted.

All of this may be changing. A new set of techniques, aimed not at breaking into computers but at manipulating the increasingly intelligent machine learning models that control them, may force law and legal institutions to reevaluate the very nature of hacking. Three of the authors have shown, for example, that it is possible to use one's knowledge of a system to fool a machine learning classifier (such as the classifiers one might find in a driverless car) into perceiving a stop sign as a speed limit. Other techniques build secret blind spots into learning systems or reconstruct the private data that went into their training.

The unfolding renaissance in artificial intelligence (AI), coupled with an almost parallel discovery of its vulnerabilities, requires a reexamination of what it means to "hack," i.e., to compromise a computer system. The stakes are significant. Unless legal and societal frameworks adjust, the consequences of misalignment between law and practice include (i) inadequate coverage of crime, (ii) missing or skewed security incentives, and the (iii) prospect of chilling critical security research. This last one is particularly dangerous in light of the important role researchers can play in revealing the biases, safety limitations, and opportunities for mischief that the mainstreaming of artificial intelligence appears to present.

The authors of this essay represent an interdisciplinary team of experts in machine learning, computer security, and law. Our aim is to introduce the law and policy community within and

[1] We acknowledge that there is a second, classic, definition of hacking, which refers to deep technical explorations of computer systems without malice (https://tools.ietf.org/html/rfc1392). This definition contrasts hacking to "cracking." However, we use the more contemporary definition of hacking here.

beyond academia to the ways adversarial machine learning (ML) alter the nature of hacking and with it the cybersecurity landscape. Using the Computer Fraud and Abuse Act of 1986—the paradigmatic federal anti-hacking law—as a case study, we mean to evidence the burgeoning disconnect between law and technical practice. And we hope to explain what is at stake should we fail to address the uncertainty that flows from the prospect that hacking now includes tricking.

The essay proceeds as follows. Part I provides an accessible overview of machine learning. Part II explains the basics of adversarial ML for a law and policy audience, laying out the set of techniques used to trick or exploit AI as of this writing. This appears to be the first taxonomy of adversarial ML in the legal literature (though it draws from prior work in computer science).

Part III describes the current anti-hacking paradigm and explores whether it envisions adversarial ML. The question is a close one and the inquiry complex, in part because our statutory case study, the CFAA, is broadly written and has been interpreted expansively by the courts. We apply the CFAA framework to a series of hypotheticals grounded in real events and research and find that the answer is unclear.

Part IV shows why this lack of clarity represents a concern. First, courts and other authorities will be hard-pressed to draw defensible lines between intuitively wrong and intuitively legitimate conduct. How do we reach acts that endanger safety—such as tricking a driverless car into mischaracterizing its environment—while tolerating reasonable anti-surveillance measures—such as makeup that foils facial recognition—which leverage similar technical principles, but dissimilar secondary consequences?

Second, and relatedly, researchers interested in testing whether systems being developed are safe and secure do not always know whether their hacking efforts may implicate federal law.[2] Here we join a chorus of calls for the government to clarify the conduct it seeks to reach and restrict while continuing to advocate for an exemption for research aimed at improvement and accountability. Third, designers and distributors of AI-enabled products will not understand the full scope of their obligations with respect to security. We advance a normative claim that the failure to anticipate and address tricking is as irresponsible or "unfair" as inadequate security measures in general.

We are living in world that is not only mediated and connected, but increasingly intelligent. And that intelligence has limits. Today's malicious actors penetrate computers to steal, spy, or disrupt. Tomorrow's malicious actors may also trick computers into making critical mistakes or divulging the private information upon which they were trained. We hope this interdisciplinary project begins the process of reimagining cybersecurity for the era of artificial intelligence and robotics.

[2] Our focus is on the CFAA but, as we acknowledge below, other laws such as the Digital Millennium Copyright Act also establish penalties for unauthorized intrusion into a system. The DMCA, however, has an exception for security research.

Part I: Machine Learning

Artificial intelligence (AI) is probably best understood as a set of techniques aimed at approximating some aspect of human or animal cognition. It is a long-standing field of inquiry that, while originating in computer science, has since bridged many disciplines. Of the various techniques that comprise artificial intelligence, a 2016 report by the Obama White House singled out machine learning (ML) as particularly impactful.[3] ML indeed underpins many of today's most visible applications grouped under the umbrella term AI.[4] It refers to the ability of a system to improve performance by refining a model. The approach typically involves spotting patterns in large bodies of data that in turn permit the system to make decisions or claims about the world. The process is divided into two stages: training and inference. During training, available data is used to generate a model orientated toward a particular objective such as fraud detection. Next, during inference, the trained model is deployed to make claims or predictions about previously unseen data, such as new bank statements.

A. Typology of Machine Learning

There are three prevalent approaches to machine learning in the literature:

Supervised learning algorithms: In this scenario, there is a large set of labeled input/output pairs (e.g., images and labels for the objects in them). The goal of such algorithms is to predict the labels for data points that the model has not seen. The classic example of such systems are computer vision classifiers that assign a single category out of a fixed set to an image (e.g., "malignant" or "benign" for scans of tumors). However, more sophisticated models exist where the labels are descriptions of the locations of certain objects in an image (and not just their categories). For instance, a car detector can take in an entire scene and identify the position of cars in that image (in terms of pixel coordinates).

Reinforcement learning: In this scenario, ML models act as semi-autonomous agents that choose actions based on reward signals from their environment. During training, the models are updated in order to learn policies that describe what to do in every state. During inference, they apply those policies to choose how the agent they are controlling moves or otherwise changes its state. For instance, in financial trading, the state could be the portfolio of a trader and the "move" a sequence of buy and sell actions involving items in the trader's portfolio. Such a model could be trained on past transactions in the financial markets and the associated results.

Unsupervised learning algorithms: Here, there is no explicit labeling of the training data. The goal is to uncover interesting patterns that humans may not have spotted or even be able to interpret easily. A common application is "clustering" where groups of similar objects are

[3] https://obamawhitehouse.archives.gov/blog/2016/12/20/artificial-intelligence-automation-and-economy.
[4] We are not committed in any deep sense to the idea that ML falls within, rather than adjacent to, AI. However, we adopt for purposes of this essay the conventional frame that ML is a form of AI.

bunched together. “Similar” is defined based on the application but is generally a mathematical computation of some difference. An instance of an unsupervised learning system might be a bank observing credit card transactions that fall into distinct clusters—say, those that high-earning homeowners make as compared to those by high-risk individuals. A fraudulent or otherwise peculiar transaction would then not fall into any cluster nicely, thus being detected by the system.

There are other types of machine learning algorithms, including some that do not fit neatly into these categories. However, most practical machine learning deployed on real systems today fall into the first and second categories.

Machine learning is often associated with a particular instantiation known as “deep learning.” Deep learning involves the distillation of information presented in a complex format (for instance, pixels) down to easily interpretable labels (for instance, an object category) by layering the processing of information. For example, a first layer of an image processing deep learning algorithm might attempt to detect boundaries between objects in an image. Even though that information being output by the first layer might be somewhat distilled, it still will not be that useful to the computer. Hence, subsequent layers reduce the complexity more and more until finally the model outputs a simple concept, such as the category of the object in the image.

By itself, deep learning is not a new concept and indeed many machine learning models before deep learning attempted to do just that with layers handcrafted by researchers. For instance, to classify faces, scientists would try to define which regions of the face were important for predicting identity by specifying how to process the images. One of deep learning’s innovations was to let each computational layer adjust itself automatically based on the training data. This is achieved by mathematically defining how close the output of the final computational layer is to what is desired and how to update the intermediate layers so that the overall output gets closer to the target. Thus, with enough time, the model will strive to get better at outputting the label “dog” for all images of dogs.

Furthermore, the internals of deep learning models are represented using matrix multiplications. Computer scientists had been studying how to make those operations execute quickly and in parallel for many decades before deep learning took off. Thus, deep learning also has the benefit of naturally parallelizing computations at a time when the performance of non-parallel computing power is flattening out.

B. Training Data

Whether supervised or unsupervised, deep or shallow, a commonality across all machine learning approaches is the centrality of training data. Training data can be provided to machine learning algorithms in many different ways. Many datasets are created in laboratories where conditions can be specified precisely. For example, when building a face recognition training

set, taking images in a lab could provide exact details on the position of the subject's head, the camera exposure settings and lighting conditions, and other such parameters.

However, this is not always practical, especially for data-hungry models, such as today's deep learning frameworks. These algorithms need many more precisely-labeled examples than individual researchers could possibly generate. Thus, in many applications, the training set and its labels are automatically generated or crowdsourced. For instance, to generate an image recognition dataset, one might simply select all images that come up in a Google Image search for the categories of interest. If a researcher wanted to build a classifier of car models, they would search for each model on Google Images and download the resulting pictures. Alternatively, one could collect a lot of images with unknown labels and then ask volunteers online to label them. A third option is to attempt to infer the labels from user activity. For example, to generate a text prediction dataset from words typed in a keyboard, one might simply look at what a user chooses to type next.

Sources of training datasets turn out to be extremely important in channeling the social impacts of ML. Whether created synthetically in a lab, purchased from a vendor, or scraped from the internet, the dataset a model encounters during its training phase will dictate its performance at the application or interference phase.[5] If the training data is historically sourced, it will contain the biases of history. It the training data is not representative of a particular population, the system will not perform well for that population. A generation of contemporary scholars are adding to the existing literature around machine bias.[6] As our focus is on computer security, we refer to the conversation here only to acknowledge its importance.

C. Measuring Performance

In academia, it is common practice to compare the performance of machine learning algorithms along standardized benchmarks. For example, for the task of identifying objects in images, researchers have collected millions of images and chosen 1000 object types to construct the ImageNet benchmark.

Measuring performance is not a straightforward process, as many different scores exist that vary significantly depending on application. In fact, what we think of as "accuracy" in common speech (i.e., the percentage of correct answers an algorithm produces to a set of questions) is often a very poor metric of how well a model performs. Having a system that is accurate, is not a victory unless the system is aiming for the correct target. For example, an accuracy metric is especially bad in medical diagnosis systems as it does not account for how often a disease appears in the population. Thus, a naive algorithm for a rare disease that simply classifies every case it sees as "not sick" would score a pretty high accuracy as most cases it sees will indeed

[5] See Amanda Levendowski, How Copyright Law Can Fix Artificial Intelligence's Implicit Bias Problem, __ Washington Law Review __ (forthcoming 2018).
[6] E.g., Conference on Fairness, Accountability, and Transparency (FAT*), https://fatconference.org/index.html.

be not sick (the disease is rare). However, this would obviously be a very bad algorithm as it will never predict when a person really has the disease. Thus, machine learning researchers have to be careful to choose metrics that capture this and similar tradeoffs in performance and make it hard for models to "cheat." It should be noted that even these metrics are often heuristics and there is no perfect transferability to real-world "usefulness" of the model.

Another important aspect of ML models is that it is very easy to make them "overfit." One can broadly think of this as "cheating" on the chosen performance metric---overfit models achieve good performance scores but fail to generalize in real scenarios. A trivial way to cause overfitting is to memorize all the labels for images in a training set and then reproduce those labels when asked for any particular image's prediction. In this case, the algorithm would achieve really good accuracy on its training set but will be completely useless for any practical application. ML researchers guard against this by holding out a test set that the model has never seen in order to evaluate its realistic performance. However, there are more subtle ways of causing overfitting and sometimes it is not even intentional. For instance, a model that is given only blue flowers to look at during training and only tested on blue flowers might focus only on the color and thus learn to classify only blue objects as flowers. Since the test set contains no flowers of other colors, the model would score well but fail in the real world where non-blue flowers exist. Thus, researchers need to ensure that their training and test sets are properly balanced and include a large enough sample of relevant features. One way to detect overfitting is for the benchmark holders to keep the evaluation dataset hidden from the model developers until a final version of the model is presented.

Until recently, there were few good machine learning algorithms that could rival human performance on common benchmarks. For instance, identifying what object is in a picture or finding out who a facial image belongs to used to be highly challenging for computers. However, a confluence of greater availability of large datasets, advances in parallel computing, and improvements in processing power helped deep learning models achieve human-level or better performance. Subsequently, researchers applied deep learning in a host of other areas, which led to the past decade's explosion in deep learning applicability and use.

Unfortunately, the power to perform well comes at a cost. Due to the large number of parameters, deep learning models are hard to interpret. While it is trivial to look at the matrices their training has generated, it is not clear what they are computing individually or as a whole. Therefore, it is not easy to explain what any intermediate layer is doing or how it is contributing to the overall prediction. This remains an active area of computer science research.

Part II: Adversarial Machine Learning

There are many ways machine learning algorithms can fail naturally. A relatively new area of study evidences the ways people can cause ML to make predictable errors by exploiting system blind spots. Researchers to date have identified three main approaches to "adversarial" machine learning. These include: (1) fooling a trained classifier or detector into

mischaracterizing an input in the inference phase, (2) skewing the training phase to produce specific failures during inference, and (3) extracting the (sometimes sensitive) underlying data from a trained model.[7] We discuss each of these approaches in turn.

A. Adversarial ML since 2013

Germinal work from 2013 by Szegedy et al. discovered that, in the domain of image recognition, changing only a few pixels of an image in a particular way causes the model interpreting that image to predict a wrong label.[8] Later work showed that even more sophisticated models were vulnerable to such human-imperceptible "adversarial examples" and provided powerful algorithms to find these malicious inputs. Researchers also established that attackers have some latitude in picking how the model they target will misbehave. An adversary could, for example, select a target class to send the model's prediction of the adversarial inputs to. For instance, an adversary could make a warning label appear as a particular message of their choosing such as an expiration date.

Other computer scientists discovered that adversarial examples also transfer across models performing the same task. Thus, attackers could generate malicious inputs for a proxy classifier and use them to cause failure in another similar system. For instance, Liu et al. demonstrated that one could take images of various objects, add adversarial noise for a publicly available model, send them to a commercial, image recognition service with unknown internals, and cause that commercial model to predict "veil" for an image of a dog.[9] Finally, a growing body of work is focusing on how to produce physical adversarial examples. For instance, two recent high-profile papers demonstrated that wearing specifically crafted glasses can trick face recognition systems[10] and that applying adversarial stickers to a road sign can cause the sign to be misinterpreted by an autonomous-vehicle's image classifier.[11]

Note that the attacks discussed so far happen *after* the model is trained and after it has been deployed; i.e., at inference time. The attacker can execute those attacks without interfering with the training procedure, by simply presenting the model with modified inputs. However, an attacker needs to know the precise internals of the model being targeted or at least those of a similar model. In the latter case, the attacker could make use of the fact that adversarial

[7] Papernot, Nicolas, et al. "Towards the science of security and privacy in machine learning." 3rd IEEE European Symposium on Security and Privacy, London, UK.
[8] Szegedy, Christian, et al. "Intriguing properties of neural networks." arXiv preprint arXiv:1312.6199 (2013).
[9] Liu, Yanpei, et al. "Delving into transferable adversarial examples and black-box attacks." arXiv preprint arXiv:1611.02770 (2016).
[10] Sharif, Mahmood, et al. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016.
[11] Evtimov, Ivan, et al. "Robust physical-world attacks on machine learning models." arXiv preprint arXiv:1707.08945 (2017). [update cite]

examples generated for similar models transfer to generate malicious inputs for an available model and attack a hidden one.

Another set of attacks has focused on interfering with model training. An adversary who could tamper with the training data can, in theory, compromise the model in any arbitrary way. For instance, the adversary could label all pictures of rabbits in the training set as pictures of dogs. A model will then naturally learn that dogs look like rabbits. Similarly, the adversary could be more subtle and train the model so that every picture of a rabbit with a particular patch of hair gets classified as a dog. However, the adversary need not control the training set or even its labels to "backdoor" an error into the model in this way. One recent work demonstrated that an adversary with full access to the trained model only, i.e., white-box access, can build in a "trojan trigger."[12] This trigger would only cause misclassification if it is presented to the model, but will not otherwise impact the performance of the model. This could become problematic for models that are distributed online or are fully trained by a third party as a service.

A third type of attack on deep learning models may seek to compromise the privacy of data contained within the training set.[13] In this type of attack, an adversary needs to obtain the full model (its internal structure and weights). The attacker can then seek to either infer membership of particular individuals or reconstruct the training data. For instance, a naive text prediction model that may be incorporated in a smartphone keyboard could be inverted to extract sensitive data the user has typed in the past, such as a Social Security Number, a Date of Birth, or private messages in an otherwise end-to-end encrypted messaging app such as Signal. It is generally possible to protect against such attacks by employing a mathematical technique known as differential privacy.[14] At a high level, this technique allows you to add noise to the data in a way that preserves its useful properties for the whole dataset but makes it hard for adversaries to reveal information about individual members. However, research is still ongoing on the performance tradeoffs when employing this protective technique.

B. Limitations of Adversarial ML

It is important to acknowledge that most applications of adversarial machine learning today are limited to academic proofs of concept and do not necessarily reflect current vulnerabilities in deployed systems. In the case of adversarial examples, it is more than likely that deployed systems employ some pre- or post-processing to their models such that adversarial examples can be detected or filtered out (although no defense has worked to date). In addition, no adversarial examples have been shown that defeat multiple different models at the same time. For instance, a self-driving car that perceives adversarial stop signs that an image classifier mistakes for speed limit signs might still detect the sign correctly via its LiDAR technology.

---

[12] Liu, Yingqi, et al. "Trojaning attack on neural networks." (2018).
[13] See Shokri, Reza, et al. "Membership inference attacks against machine learning models." 2017 IEEE Symposium on Security and Privacy (SP) for one example.
[14] Fredrikson, Matthew, et al. "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing." USENIX Security Symposium. 2014.

Furthermore, the most powerful attacks occur for now only with full "white box" knowledge of the models that are targeted. This might be too much to assume since many models are likely to remain proprietary. A lot of computer science research also points out that such full access is not necessary to mount an attack because adversarial examples designed for one model can often attack similar, unknown models as well. However, those attacks that do transfer across models generally include much higher distortions, distortions that might be noticeable to humans. Similar limitations exist for model inversion attacks.

While the space of attacking machine learning models is still technologically young, we later in the paper present several case studies that might be close to actualization in the near future. We do not believe that adversarial tampering with machine learning models is less of a threat today than malicious programs were to early operating systems. It is likely that the attackers' technology will advance and the time to think about defenses and the possible implications for our policy framework is now.

Part III: Anti-Hacking Laws

Legislation often reacts to specific threats or harms. The Computer Fraud and Abuse Act (CFAA) is a good example.[15] According to popular lore, President Reagan saw the movie *War Games* and met with his national-security advisers the next day to discuss America's cyber vulnerabilities. The CFAA is said to be the result of their deliberations. Enacted, at any rate, in 1986, the CFAA aimed to combat computer-related crimes. Since its implementation, the CFAA has been the nation's predominant anti-hacking law. While drafted to combat traditional computer hacking, "the CFAA has evolved into a behemoth of a federal felony statute."[16] This Part lays out the statutory definitions that the CFAA relies on for applicability, e.g., what is a "protected" computer, etc., but a theme throughout CFAA's actual usage shows that, "almost anything with at least a microchip and some relation to interstate commerce is a protected computer and open to CFAA prosecution."[17]

A. CFAA Statutory Language

The CFAA is designed to be disincentive to the compromising of "protected computers" on threat of prosecution or civil lawsuit.[18] A computer is any "electronic, magnetic, optical, electrochemical, or other high speed data processing device performing logical, arithmetic, or storage functions, and includes any data storage facility or communications facility directly related to or operating in conjunction with such device." The CFAA specifically excludes from its

[15] *See,* Obie Okuh, Comment, *When Circuit Breakers Trip: Resetting The CFAA To Combat Rogue Employee Access*, 21 Alb. L.J. Sci. & Tech. 637, 645 (2011).
[16] Matthew Ashton, Note, *Debugging The Real World: Robust Criminal Prosecution In The Internet of Things*, 59 Ariz. L. Rev. 805, 813 (2017).
[17] *Id.*
[18] Computer Fraud and Abuse Act, 18 U.S.C.A. § 1030 (2008).

ambit, “automated typewriters or typesetter, a portable hand-held calculator, or other similar device.”

Protected computers are computers “exclusively for the use of a financial institution or the United States Government, or, in the case of a computer not for such use, used by or for a financial institution or the United States Government and the conduct constituting the offense affects that use by or for the financial institution of the Government.” The CFAA also protects any computer, whether or not connected to the government, “which is used in or affecting interstate or foreign commerce or communication, including a computer located outside the United States that is used in a manner that affects interstate or foreign commerce or communication of the United States.” The courts have deferred to the government on the former definition. The latter definition encompasses seemingly any computer with connections to the United States but carries with it certain limitations around damages discussed below.

The CFAA applies to both external and internal actors trying to compromise protected computers. External actors incur liability when they “intentionally access a [protected] computer *without authorization*.” Internal persons face liability if they already have access to a protected computer, but use the system in such a way that “*exceeds* [their] authorized access.” Generally an insider would be a current or former employee. However, as we will see, the language of exceeding authorized access has also been brought to bear by companies on users who persist in violating a terms of service despite being warned against it.

Importantly for our purposes, the CFAA prohibits not only “accessing” a computer to “obtain” information, but also “knowingly cause[ing] the transmission of a program, information, code, or command, and as a result of such conduct, intentionally causes damage . . . to a protected computer,” as long as this conduct “causes damage *without authorization*.” Thus, for example, code that encrypts a hard drive or corrupts data, or a botnet attack that shuts down a server, can violate the CFAA even though no information has been obtained as such. There are additional ways to violate the CFAA involving espionage, extortion, and trafficking in passwords. However, by the terms of the statute, there is no liability for the design or manufacture of hardware or software with vulnerabilities.

CFAA has both a criminal and civil component, meaning of course that individuals and companies can sue for a violation. The criminal component is tiered, with penalties as high as 20 years imprisonment for repeated offenses or offenses that threaten death or bodily injury. Attacking a government computer is a per se violation if the computer is being used “in furtherance of the administration of justice, national defense, or national security.” Otherwise, the defendant must do at least $5,000 in aggregate damages, harm medical equipment, threaten public safety or health, injure someone, or target many computers to be liable either criminally or in civil court.

B. CFAA Interpretation

The CFAA's statutory text leaves much room for hypothesizing how these broad-definitional parameters apply to facts on the ground. A series of well-publicized cases help define the range of situations to which CFAA applies.

Before CFAA liability can result, the actor must try to gain, or exceed, access to a "protected computer." The CFAA gives a non-exhaustive list of what can qualify as a protected computer. Subsequent interpretation has shown that protected computer is given a quite expansive definition. Courts have deemed that cell phones are considered computers under the CFAA; further, given cell phones typical uses, i.e., interstate commerce or communication, they would also be considered protected computers.[19] In determining that cell phones count as computers, the court looked at the facts that cell phones keep track of the number of incoming/outgoing calls, i.e., "performing logical, arithmetic, or storage functions" under the CFAA. Further, the court emphasized that the cell phone used "software" as in integral part of its function.

Courts tend to be particularly expansive in their interpretation of the statute when the facts of the case implicate a public safety concern. In United States v. Mitra, the court's interpretation stretched the CFAA's transmission requirement to include sending out a radio signal. The radio signal was used to interfere, via jamming the signal, with the dispatching station's function for the local police department and 911 call center. The CFAA's transmission element requires the transmission of "a program, information, code, or command," to trigger CFAA liability. The transmission definition expands, more liberally, when public safety is compromised; i.e., Mitra compromising the 911-call-centers function by way of a radio signal.

Another case analyzed whether information transmitted without authorization need specifically to be "malicious" to constitute a CFAA violation. Fink v. Time Warner Cable found that the CFAA does not require the information transmitted to be malicious for the actor to incur liability.[20] Here, Time Warner Cable remotely accessed their customers computers to transmit a "reset packet" to prevent undesired functions by way of throttling peer-to-peer file sharing. The reset packet had no malicious intent, but the unauthorized access and transmission alone were sufficient to violate the CFAA, and meet the CFAA's damage requirement by customers claiming the services they purchased were diminished by the reset packages.

Blocking access to public safety services swayed the court in Mitra to apply the CFAA; subsequent courts have furthered the analysis and said blocking access to websites by means of denial of service attacks, or DDoS attacks. In dealing with DDoS attacks against websites, the court focuses on the "intent to cause damage" provision of the CFAA.[21] Defendant Carlson directed thousands of emails at a single email address to try and compromise the function of the

[19] *See, United States v. Kramer*, 631 F.3d 900 (8th Cir. 2011).
[20] *See*, *Fink v. Time Warner Cable*, 810 F. Supp. 2d 633 (S.D.N.Y. 2011).
[21] *See*, *United States v. Carlson*, No. 05-3562, 209 Fed.Appx. 181 (3d Cir. 2006).

website. The court found Carlson was aware of, and motivated by, the potential damage that his actions could cause and found him in violation of the CFAA. In an analogous case, the defendants attempted to disrupt the operations of a business by directing "swarms" of phone and email messages at their respective addresses.[22] The concentrated attacks at a business's personal accounts were methods "that diminish[ed] the plaintiff's ability to use data or a system . . . causes damage," and violates the CFAA. The courts have broadened the definition of "hacking" by adding CFAA liability for blocking access to services or platforms.

Hacking under the CFAA has even been defined to include using a website's services in a way that violates the owner's terms of service---as long as the violator has been adequately warned by the website's owner.[23] Defendant Vachami was violating Facebook's Terms of Use Agreement by sending automated messages to Facebook users and received a cease and desist letter regarding his actions. By continuing to violate the Terms of Use Agreement, the court concluded Vachami knowingly "exceeded authorized access" and violated the CFAA. By classifying this behavior as hacking under the CFAA, the court actually cabined a previous ruling. Later overturned, the lower court in United States v. Drew held that simply violating a website's Terms of Service, analogous to Facebook's Terms of Use Agreement, and causing damage constituted hacking under the CFAA without the need for the website's owner to warn the defendant.[24]

When it comes to computers that do not have terms of service that the user assents to, there is no CFAA liability if the user discovers, then exploits, a system vulnerability, as long as the user did not "circumvent any security protocols" programmed into the computer. In the interesting if unpublished case United States v. Kane, the defendant discovered that a electronic poker machine had a flaw in its software that allowed him to push a series of buttons in a particular order and cause the machine to declare him the winner, resulting in a windfall of earnings.[25] The court agreed with prosecution in deeming the electronic poker machine a "protected computer," but would not extend CFAA liability to defendant due to his lack of circumventing, or "traditional hacking."

Notably, CFAA has no research exception.[26] Thus, security researchers attempting to discover potentially dangerous security flaws in protected computers can, in theory, be prosecuted using the full weight of the CFAA. This stands in contrast to other federal law. The Digital Millennium Copyright Act (DMCA), the law protecting circumventions of copyright protections on digital mediums, has an expressly carved out research exception; specifically, for encryption research.

---

[22] *See*, *Pulte Homes, Inc. v. Laborers' Int'l Union of N. Am.*, 648 F.3d 295 (6th Cir. 2011).
[23] *See*, *Facebook, Inc. v. Power Ventures, Inc.*, 844 F.3d 1058 (9th Cir. 2016).
[24] *See*, *United States v. Drew*, 259 F.R.D. 449 (C.D. Cal. 2009).
[25] *United States v. Kane*, No. 2:11-cr-00022-MMD-GWF, 2015 U.S. Dist. LEXIS 177544 (D. Nev. Dec. 16, 2015). (Unpublished cases have limited precedential effect.).
[26] *See*, Derek E. Bambauer & Oliver Day, *The Hacker's Aegis*, 60 Emory L.J. 1051, 1105 (2011).

The DMCA exempts encryption researchers who, "circumvent a technological measure for the sole purpose of . . . performing the acts of good faith encryption research."[27]

While the CFAA is perhaps the best known anti-hacking statute, it is hardly the only law or standard to address computer security. Additional laws make roughly the same assumptions as the CFAA. For example, at an international level, the Budapest Convention on Cybercrime defines a cyber crime of "illegal access," i.e., "the access to the whole or part of any computer system without right."[28] While it does not have a stated definition of hacking, the Federal Trade Commission has developed a series of investigations and complaints involving inadequate security. Where a company's security practices fall sufficiently short of best practice, the FTC pursues the company under a theory of "unfair or deceptive practice." These proceedings invariably involve the exposure of personal information due to inadequate security protocols and envision hacking in the same way as the CFAA.[29]

C. Applying CFAA to Adversarial ML

In this final section, we attempt to apply the language and interpretation of the CFAA to a specific set of case studies. These case studies are hypothetical but grounded in actual research. Again, as we describe above, adversarial ML is subject to certain limitations related in part to the research context. Here we assume for the sake of argument that techniques of adversarial ML can be transferred into real world settings.

Planting adversarial sound commands in ads. A perpetrator of intimate partner violence buys a local television advertisement in the jurisdiction he suspects his ex now resides. Embedded in the ad is an adversarial sound input that no person would recognize as meaningful. The attack causes his ex's personal assistant in range of the TV to publish her location on social media.

Causing a car crash by defacing a stop sign to appear like a speed limit. An engineer extensively tests the detector used by the driverless cars company where she works. She reports to the founder that she's found a way to knowingly deface a stop sign to trick the car into into accelerating instead of stopping. The founder suspends operations of his own fleet but defaces stop signs near his competitor's driverless car plant. A person is injured when a competitor driverless car misses a stop sign and collides with another vehicle.

Shoplifting with anti-surveillance makeup. An individual steals from a grocery store equipped with facial recognition cameras. In order to reduce the likelihood of detection, the individual wears makeup she understands will make her look like another person entirely to the machine learning model. However, she looks like herself to other shoppers and to grocery store staff.

---

[27] Digital Millennium Copyright Act of 1988, 17 U.S.C.S. § 1201 (Lexis 2018).
[28] The authors would like to thank Jesse Woo for furnishing this example.
[29] *See*, Daniel J. Solove & Woodrow Hartzog, *The FTC and the New Common Law of Privacy*, 114 Columbia Law Review 583 (2014).

Poisoning a crowd-sourced credit rating system. A financial start up decides to train an ML model to detect "risky" and "risk averse" behavior so as to assign creditworthiness scores. A component of the model invites internet users to supply and rate sample behaviors on a scale from risky to risk averse. A group of teenagers poison the model by supplying thousands of images of skateboarders and rating them all as risk averse. One teenager from the group whose social network page is full of skateboarding pictures secures a loan from the start up and later defaults.

Data inversion across international borders. A European pharmaceutical company trains and releases a companion model with a drug it produces that helps doctors choose the appropriate dosage for patients. The model is trained on European data but subsequently released to doctors in the United States. A malicious employee in the U.S. with access to the model uses an algorithm to systematically reconstruct the training set, including personal information.

There is a case to be made that the CFAA could apply to each of these scenarios. The adversarial sound in the first scenario could constitute the "transmission" of a "command" to a "protected computer," i.e., the victim's phone. Assuming the revelation of the victim's location leads to physical harm, perhaps in the form of violence by the perpetrator, the damage requirement of CFAA has been satisfied. Similarly, by defacing the stop sign, the malicious competitor can be said to have caused the transmissions of "information"---from the stop sign to the car---that led to a public safety risk. In both instances, had the attacker broken into the phone or car by exploiting a security vulnerability and altered the firmware or hardware to cause the precise same harm, the CFAA would almost certainly apply.

On the other hand, a perhaps equally strong case could be made that CFAA does not apply. In neither scenario does the defendant circumvent any security protocols or violate a terms of service. The transmission of an adversarial sound seemingly does not cause damage without authorization to a protected computer. Rather, it causes damage to a person through an authorized mechanism---voice control---of a protected computer. With respect to the driverless car scenario, it feels like a stretch to say that minor changes to the visual world that a sensor may come across constitute the "transmission" of "a program, information, code, or command" on par with a denial-of-service attack. Regardless, there is again arguably no damage to the detector "without authorization" as required under Section 1030(a)(5)(A).

However a court comes to characterize the driverless car scenario, the same logic arguably applies to the shoplifter who evades facial recognition---at least for purposes of the CFAA. Like the founder who defaces the stop sign to mislead the car's detector, the shoplifter who alters her face to mislead the facial detector has arguably transmitted information purposely to trick the grocery store into misperceiving her so she can steal. Obviously there are differences. The founder causes physical harm, the shoplifter financial. The founder has no right to alter a stop sign whereas the shoplifter has a right to apply makeup to her own face. But from a CFAA perspective, the two situations feel closely analogous.

The example of mistraining the credit rating system is similarly ambiguous. From one perspective, the teenagers are exploiting a flaw in the design of the system in order to embed a trojan horse in the form of a correlation between skateboarding and creditworthiness. Certainly if the group circumvented a security protocol and changed the valence of skateboarding by hand their actions would fall within the scope of the CFAA. From another perspective, however, the teens were just playing by the rules---however misconceived. The state or start up could no more prosecute or sue them under CFAA than the designer of a flawed poker machine in Kane that paid out every time a specific sequence is entered.

Resolution of the final scenario depends, once again, on whether tricking a system into divulging private information is the same as hacking into the system to steal that information. Presumably the European pharmaceutical company---beholding to strict EU law---did not design the model anticipating exfiltration of data. But nor did the perpetrator access the model without authorization. He merely queried the model in surprising way.

Part IV: What's At Stake

To sum up the argument thus far: Contemporary law and policy continues to conceive of hacking as breaking into or disabling a computer. Devices increasingly leverage machine learning, and potentially other techniques of artificial intelligence, to accomplish a range of tasks. These "smart" systems are not so smart that they cannot be tricked. A burgeoning literature in computer science is uncovering various techniques of adversarial machine learning that, at least in experimental settings, are capable of misleading machines and even forcing dangerous errors. What a comparison between the leading anti-hacking law and adversarial machine learning reveals is ambiguity. It simply isn't clear how or when the CFAA or similar laws applies to "tricking" a robot as opposed to "hacking" it. This ambiguity has a number of potentially troubling consequences, which this Part now explores.

A. Line-drawing and overreach

Our first concern is that line-drawing problems will lead to uncertainty, which in turn could fuel prosecutorial overreach. The CFAA already faces criticizing for its hazy boundaries,[30] and both companies and prosecutors have pushed the envelope in arguably problematic ways.[31] A thoughtless application of CFAA to adversarial machine learning could exacerbate the problem by providing the CFAA with a dangerous new scope.

To illustrate, consider again the problem with drawing a line between subtly defacing a stop sign to make it appear like a yield sign and subtly altering one's makeup to fool facial recognition. It seems plausible enough that a prosecutor would bring a CFAA violation in the former case and, further, that a court would permit the state to go forward. It may make intuitive sense to a judge

30

31

that providing false inputs to car detectors in order to disrupt operations is analogous to transmitting malicious code or engaging in a denial of service attack. Coupled with the tendency of courts to be more solicitous of the state in CFAA cause involving a public safety hazard, we can readily imagine a judge blessing the state's CFAA theory.

So what of the latter case? How does a court that decides for the state when the defendant tricked a robot car then turn around and decide against it when the defendant changes her appearance to trick an AI-enabled camera in various contexts? The line cannot be that one intervention can cause physical harm and the other does not. Tricking a car will not always cause harm, and fooling facial recognition in theory could—for example, in our shoplifter example. Moreover, the CFAA does not require harm to flow from unauthorized access if, as often, the protected computer at issue belongs to the government and is using in the furtherance of security. Thus, wearing makeup at the airport with the intent not to be recognized by TSA cameras could rise to a CFAA violation, at least in the wake of a precedent saying holding that defacing a stop sign with the intent that it not be recognized by a driverless car does violate CFAA.[32]

Note that the CFAA not only punishes the act of hacking, but it also punishes any "attempted offense" that "would, if completed" cause damage or loss. This attempt provision also aligns oddly with adversarial ML. Automated attempts to locate vulnerabilities in protected computers and, where present, exploit those vulnerabilities are clearly attempts for purposes of CFAA. But what of wearing anti-surveillance makeup all day, in a variety of settings? And does the person who defaces a stop sign "attempt" to attack each and every car that passes, even if a human is at the wheel? These too remain open questions.

B. Chilling research

Our second concern flows from the first. If courts interpret the CFAA too broadly in the context of adversarial ML, then researchers may fear running afoul of the CFAA—which has no research exception—when testing real systems for resilience. The case that CFAA overreach chills security and other research has already been made repeatedly.[33] Researchers may fear to compromise proprietary systems or scrape digital platforms for data even if it is clear that their purpose is not malicious. The CFAA has a private cause of action and firms may still have an incentive to chill such research to avoid embarrassment. There are safety valves—such as the requirement of harm for private litigants—but the threat of lawsuit alone could suffice to dissuade some research.

[32] Law enforcement may indeed want facial recognition avoidance to constitute a crime. But our intuition is that most would see facial recognition avoidance as a reasonable means by which to preserve privacy and liberty interests, and in any event of a different order from tricking a vehicle into misperceiving a road sign.

[33]

Thus, our argument is not one of kind but of degree. As noted in the Obama White House report on AI,[34] by the AI Now Initiative,[35] and by the U.S. Roadmap to Robotics,[36] outside researchers have a critical role in examining AI systems for safety, privacy, bias, and other concerns. The community is relying upon the ability of impartial individuals within academia, the press, and civil society to test and review new applications and independently report on their performance. Should courts come to expand CFAA's ambit to include manipulation of AI, including for testing purposes, the result would be to remove an important avenue of AI accountability.

C. Incentive misalignment

The first two concerns deal with too broad an interpretation of hacking. The last problem in ways deals with the reverse: If adversarial ML is *not* hacking, then do firms who release AI-enabled produces and services have any legal obligation to ensure that these systems are resilient to attack? As alluded to above, the CFAA is not the only anti-hacking law or policy to assume a particular mental model. The FTC also requires products and services to employ reasonable measures against hacking. If it is too easy to compromise a system, then the FTC can bring—and repeatedly has brought—complaints against the firms that make those systems.

Tricking a robot can sometimes accomplish functionally the same ends as hacking it. Thus, an adversary might "steal" private information by hacking into an AI-enabled system or by reverse engineering its training data. Similarly, an adversary could temporarily shut down a system through a denial of service attack or by poisoning its training data to make the system suddenly useless in one or more contexts. To the extent that the prospect of an FTC enforcement proceeding for inadequate security incentivizes firms to take basic precautions against attack, we might worry that the failure of the Commission to envision susceptibility to adversarial ML as akin to poor security would under-incentivize companies to build robust systems.

It is fair to point out a potential tension here. How could we be arguing, on the one hand, that it is dangerous to widen the scope of hacking to encompass adversarial ML when it comes to the threat of prosecution or litigation, but also dangerous not to when it comes to security standards? There may be a tension here. But note that the FTC and other bodies are not limited to enforcing security under broad standards such as "unfairness and deception." The FTC could create a separate category of unfairness for inadequate resilience to known adversarial machine learning techniques, without committing to the idea that tricking is hacking.

Conclusion

Computer security is undergoing if not a paradigm shift, then a significant evolution. Computer systems continue to be a target for malicious disruption and the exfiltration of data. As contemporary applications increasingly leverage machine learning and other techniques of

34

35

36

artificial intelligence to navigate the digital and physical world, these systems present new concerns as well. Recent research, including by some of the authors, demonstrates how the added affordances of AI also entail novel means of compromising computers. Researchers have shown in experimental settings that machine learning can be misdirected during both inference and training and that training data can sometimes be reconstructed. In short, robots can be tricked.

Collectively, the prospect of adversarial machine learning may require law and policy to undergo a significant evolution of its own. Contemporary anti-hacking and security law assumes hacking to involve breaking into or temporarily incapacitating a computer with code. The misalignment between the early understanding of hacking and today's techniques creates ambiguity as to where and how the law applies. This ambiguity is dangerous to the extent that it invites prosecutorial overreach, chills research, or leads to underinvestment in hardening measures by firms releasing ML-enabled products and services.

Ultimately it is up to courts, policymakers, and industry to come to grips with the prospect of tricking robots. Our role is not to dictate a precise regulatory framework. We do have a few recommendations, however, that follow from our concerns. We recommend clarifying the CFAA so as to cabin prosecutorial discretion. We recommend clarifying the CFAA and related laws to exempt research into AI resilience so we can continue to test systems for safety, privacy, bias, and other values. And we recommend incentives for firms to build AI systems that are more resilient against attack, perhaps in the form of Federal Trade Commission scrutiny should a firm release that cyber-physical system that is too easy (whatever that comes to mean) to trick. This is, of course, only the beginning of the conversation. We very much look forward to the thoughts of other experts.