

# Computer

08.19

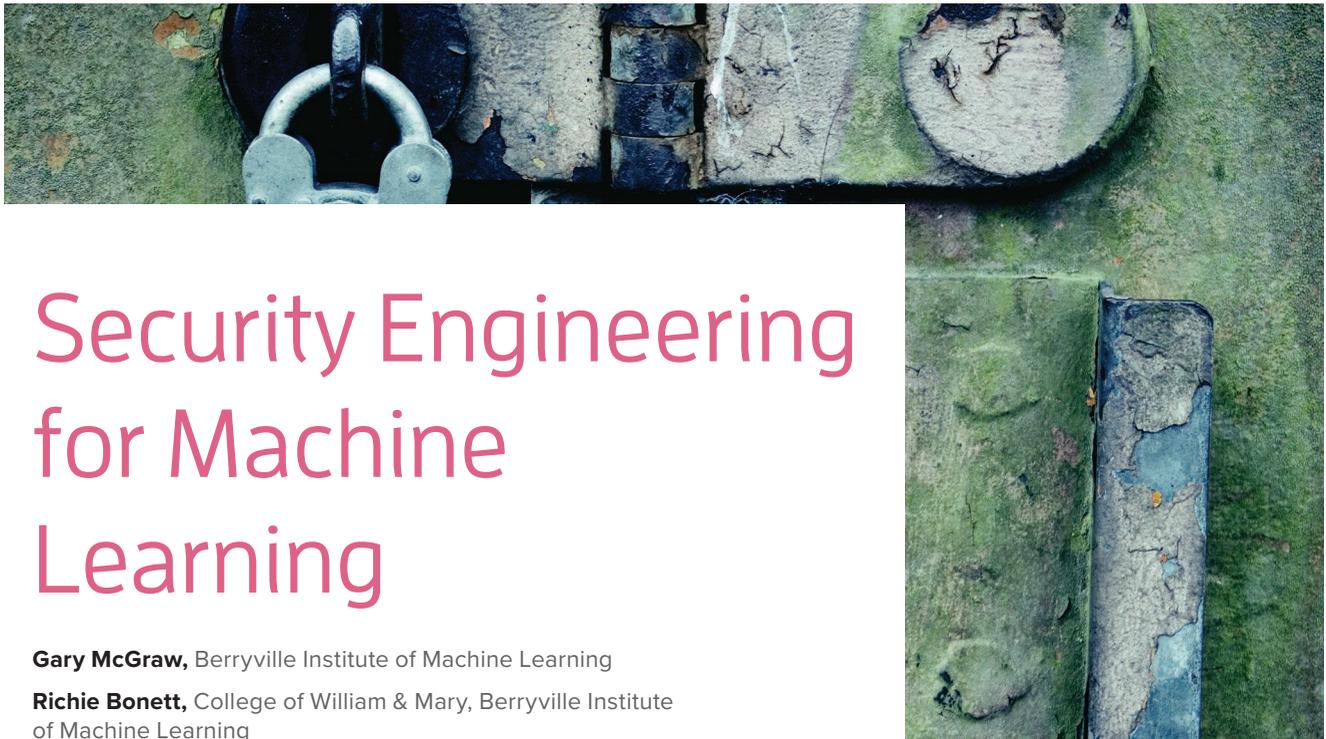


**GO/NO GO**

CS Election  
Candidates **77**

IEEE President-Elect  
Q&A **93**





# Security Engineering for Machine Learning

**Gary McGraw**, Berryville Institute of Machine Learning

**Richie Bonett**, College of William & Mary, Berryville Institute of Machine Learning

**Harold Figueroa and Victor Shepardson**, Ntrepid, Berryville Institute of Machine Learning

*Artificial intelligence is in the midst of a popular resurgence in the guise of machine learning (ML). Neural networks and deep learning architectures have been shown empirically to solve many real-world problems. We ask what kinds of risks ML systems pose in terms of security engineering and software security.*

**M**achine learning (ML) appears to have made impressive progress on many tasks, including image classification, machine translation, autonomous vehicle control, and playing complex games, such as chess, Go, and Atari video games.

Digital Object Identifier 10.1109/MC.2019.2909955  
Date of publication: 30 July 2019

This has led to much breathless popular-press coverage of artificial intelligence and elevated deep learning to an almost magical status in the eyes of the public. ML, especially of the deep-learning sort, is not magic, however. It is simply sophisticated associative-learning technology based on algorithms developed over the past 30 years. In fact, much of the recent progress in the field can be attributed to faster CPUs and much larger data sets rather than to any particular scientific breakthrough.<sup>1</sup>

ML has become so popular that its application, although often poorly understood and partially motivated

by hype, is exploding. In our view, this is not necessarily a good thing. We are concerned with the systematic risk invoked by adopting ML in a haphazard fashion. Our research is focused on understanding and categorizing security-engineering risks introduced by ML at the design level.

While the idea of addressing the security risk in ML is not a new one, most previous work has focused on either particular attacks against running ML systems (a kind of



dynamic analysis) or on operational security issues surrounding ML. Just for the record, we encourage these lines of inquiry.

Our research focuses on three threads: building a taxonomy of known attacks on ML, exploring a hypothesis of representation and ML risk, and performing an architectural risk analysis (sometimes called a threat model) of ML systems in general. We report our progress here.

## A TAXONOMY OF ML ATTACKS

Attack taxonomies in security have a long history.<sup>2</sup> One of the motivations behind building such a taxonomy is to guide engineering tradeoffs made at the design level using real-world data about how fielded systems are attacked. For that reason, we are building a taxonomy of ML attacks.

In practice, fielded ML systems as targets run the gamut from white box, which are fully open source and trained on public data, to black box, which map inputs to outputs via an application programming interface to an unknown transformation function. Between the two extremes lie many other possibilities including ML systems based on an open-source model with proprietary hyperparameters and training data and a black-box model that leverages transfer learning from an existing white-box model.<sup>3</sup>

Attacks on ML systems can be divided into two types: manipulation attacks, which alter system behavior by tweaking input, training data, or the model itself, and extraction attacks, which surreptitiously discern secret information in the ML system. Additionally, attacks can be classified by which part of the system they target (input, training data, and model). This results in a taxonomy of six categories as shown in Table 1.

Input-manipulation attacks (also known as *adversarial examples* and

*evasion attacks*) are by far the most common kind of ML attack discussed in the literature. The attacker creates an input to an operating ML system that reliably produces a different output than its creators intend. Successful attacks include stop-sign misclassification, spam misidentification, and broken language processing.<sup>4</sup>

Training-data manipulation attacks (also known as *poisoning* and *causative attacks*) are attacks on an operating model via the training process. The attacker modifies the data corpus used to train ML systems, with the intent of impairing or influencing future system behavior. For example, an attacker may publish bogus data to interfere with medical diagnoses or influence financial time-series forecasting models.<sup>5</sup> In the infamous case of Microsoft Research's Tay, Internet trolls successfully implemented a data-manipulation attack to turn the chatbot into a bigot.

There are few examples of model-manipulation attacks in the literature. However, one can imagine an attacker publishing a white-box model with certain latent behavior that is meant to be unwittingly adopted by third parties and later exploited by the attacker. Given the increasing adoption of transfer learning and the fact that releasing code, and even model parameters, under a permissive open-source license is common in ML, we believe this attack category deserves attention.

Input-extraction attacks (also known as *model inversion*) apply in cases where model output is public but inputs are supposed to remain secret. In this case, an attacker, given outputs, attempts to recover inputs. Attacks include inferring features of medical records from the dosage recommended by an ML model and producing a recognizable image of a face given only the classification and confidence score in a face-recognition model.<sup>6</sup>

Training-data extraction attacks (also called *model inversion*) involve

extracting details of the data corpus that an ML model was trained on.<sup>7</sup> ML research focuses much of its attention on the learning model to the exclusion of attention on data, yet data are clearly known to be crucially important to a trained system's behavior. Real-world ML systems often incorporate proprietary data and data with serious privacy implications.

Model-extraction attacks target any less-than-fully white-box ML system and attempt to open the box and copy the target's behavior or parameters. Examples include theft of a proprietary model and enabling white-box attacks on what was designed to be a black-box model.<sup>8</sup>

Work on this taxonomy is ongoing. (In the interest of space, we have not included as many references as we would like in this section. See Berryville Institute of Machine Learning for more information: <https://berryvilleiml.com/references/>.)

## A WORKING HYPOTHESIS ON REPRESENTATION

Our work is informed by a hypothesis about representation in ML systems that we are actively exploring. Control over input, output, and hidden representations is essential to understanding the attacks we described in the preceding section.

ML systems are conventionally evaluated on a held-out test set drawn from the same distribution as the training data. This prevents overfitting to specific examples in the training data but guarantees nothing about

TABLE 1. The six attack categories.

Input manipulation	Input extraction
Training data manipulation	Training data extraction
Model manipulation	Model extraction

generalization to a different data distribution in production. Input-manipulation attacks exploit precisely this weakness by targeting a region of input space in which system behavior is not understood. Similarly, data-manipulation attacks mold the training distribution to an attacker's intent. In an adversarial setting, we must understand ML representations over the entire potential input space, not just the training-data distribution. Representations that are unstable and corruptible can be easily (and often undetectably) tampered with. Improved representation strategies can lead directly to more secure ML systems.

Better representation approaches may also lead to more robust operation in challenging contexts well beyond avoiding adversarial dynamic activity. System robustness in the face of both limited and very noisy data can protect against catastrophic failure, especially when ML systems are applied to situations that stray beyond their training.

These ideas are not new. In our view, basic principles in representation have been discovered multiple times in multiple disciplines and published under multiple names. For example, in the numerical computation and statistical communities, phenomena such as ill-conditioning, collinearity, and outliers have long been described and are well understood. Their detrimental effects on computation and estimation are modeled through concepts such as condition numbers and statistical leverage and mitigated through techniques for regularization and outlier detection.

Our view is that an overfocus on pure learning strategies without regard to representational fluidity may be accidentally adding risk to current ML systems. We would like to take advantage of the progress that exists in various adjacent fields to explore representation issues that can improve ML systematically (mostly from a security perspective). Increased attention to representation can help in two ways: achieving more stable and efficient signal-content

representations and supporting the complementary concern of modeling signal typicality.

### EXAMPLE: ANOMALY DETECTION IN TRAINING DATA

Anomaly-detection ideas can be directly applied to input data based on some measure of the data's typicality during both ML training and operations. During training, such an approach can protect against anomalous input with high leverage that may poison the model. Anomaly detection in input can also be applied during operation to assess the typicality of the test input against the training data and offer a model-independent way of determining whether the ML system is likely to perform as expected. In both cases, anomaly scores that describe observed data drift can give us an indication of when we're interpolating and when we're extrapolating.

### EXAMPLE: DEFENSIVE INPUT TRANSFORMATION

Input transformation can be used to defend against some kinds of ML attacks, especially in the input-manipulation category. There is often a great deal of extraneous variation (for example, nonrelevant variation with respect to the ML system tasks) found in the raw input to an ML system. As a result, the ML system is likely to include some of this extraneous information in its learned hidden representations. In some sense, the bad extra information becomes entangled with the good.

Because of this, the ML system can become susceptible to bad-extra-information-based attacks. As an example, just because an image is slightly noisy, an ML recognition system should not make a silly categorization error (for example, turtle → rifle or stop sign → speed limit sign). Well-known input-manipulation attacks do exactly this with low-level noise, relying on entanglement of the noise signal with the task-relevant signal in the distributed/learned representation being built and used by the ML system.

This is not a new phenomenon. In linear-inversion problems, such as image deblurring, a numerically rank-deficient, ill-conditioned operator cannot be inverted in the presence of noise without careful consideration of the representation implicit in the process inversion. Information from subspaces associated with small singular values must be attenuated or discarded altogether. ML systems should take advantage of this knowledge.

### WILD SPECULATION

Evolved sensory systems found in nature do this kind of attenuation and discarding thing all the time [think of the bandwidth limitations in human hearing (hertz) and vision (nanometers), for example]. Raw input in biological systems is limited in a task-opportunity and risk-dependent way. The auditory and visual systems of different mammal, bird, and insect species have all evolved to reflect niche opportunities and risks (and are all divergent from each other in numerous ways; bandwidth is an easy one to observe).

In our view, the adaptations displayed by these systems are neither completely reliant on nor entirely gleaned through Hebbian learning but, rather, implemented in aspects of the anatomy and physiology of various organisms that were established through genomic evolution. As we experiment with learning systems, we should use a variety of learning algorithms, some of which may be able to achieve different kinds of search and increase robustness by introducing different types of error and nonlinearity.

### TOWARD A THOROUGH ARCHITECTURAL RISK ANALYSIS OF ML

We are interested in building security into ML systems from a security-engineering perspective. This means understanding how ML systems are designed for security (including what representations they use), teasing out possible engineering tradeoffs, and making such tradeoffs explicit. We are

also interested in the impact of including an ML system as a component in a larger design. Our basic motivating question is how do we secure ML systems proactively while we are designing and building them?

Early work in security and privacy in ML has taken an operations-security tack focused on securing an existing ML system and maintaining its data integrity. For example, Nicolas Papernot uses Salzter and Schroeder's famous security principles to provide an operational perspective on ML security.<sup>8</sup> In our view, this article does not go far enough into ML design to satisfy our goals. A key objective of our work is to develop a basic architectural risk analysis (sometimes called a *threat model*) of a typical ML system.<sup>9</sup> Our analysis will take into account common design flaws, such as those described by the IEEE Center for Secure Design.<sup>10</sup>

**S**ecuring a modern ML system must involve diving into the engineering and design of the ML system itself. Our work sets out a taxonomy of known attacks against existing ML systems, describes a hypothesis of representation that may help make ML systems more secure, and hints toward a more complete architectural risk analysis of ML. 

## REFERENCES

1. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015. doi: 10.1038/nature14539.
2. G. McGraw and G. Hoglund, *Exploiting Software*. Reading, MA: Addison-Wesley, 2004.
3. B. Wang, Y. Yao, B. Viswanath, H. Zheng, and B. Y. Zhao, "With great training comes great vulnerability: Practical attacks against transfer learning," in *Proc. 27th USENIX Security Symp.*, 2018, pp. 1281–1297.
4. X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, 2019, pp. 1–20.
5. S. Alfeld, X. Zhu, and P. Barford, "Data poisoning attacks against

- autoregressive models," in *Proc. 30th AAAI Conf. Artificial Intelligence*, 2016.
6. M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Computer Communications Security*, 2015, pp. 1322–1333.
7. R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. 2017 IEEE Symp. Security Privacy*, 2017, pp. 3–18.
8. N. Papernot, "A marauder's map of security and privacy in machine learning," presented at the 11th ACM Workshop Artificial Intelligence and Security With 25th ACM Conf. Computer and Communications Security, Toronto, Canada, Oct. 19, 2018.
9. G. McGraw, *Software Security*. Reading, MA: Addison-Wesley, 2006.
10. I. Arce et al., "Avoiding the top 10 software security design flaws," IEEE Center for Secure Design,

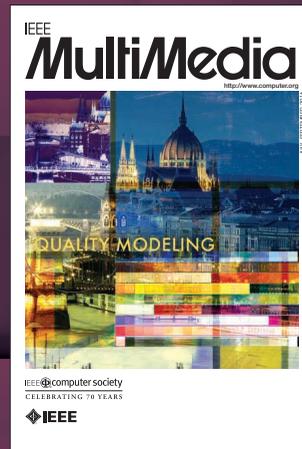
Nov. 13, 2015. [Online]. Available: <https://cybersecurity.ieee.org/blog/2015/11/13/avoiding-the-top-10-security-flaws/>

**GARY MCGRAW** is a cofounder of the Berryville Institute of Machine Learning. Contact him at [gem@garymcgraw.com](mailto:gem@garymcgraw.com).

**RICHIE BONETT** is with the College of William & Mary and the Berryville Institute of Machine Learning. Contact him at [richiebonett@gmail.com](mailto:richiebonett@gmail.com).

**HAROLD FIGUEROA** is with Ntrepid and the Berryville Institute of Machine Learning. Contact him at [harold.figueroa@gmail.com](mailto:harold.figueroa@gmail.com).

**VICTOR SHEPARDSON** is with the Berryville Institute of Machine Learning. Contact him at [victor.shepardson@gmail.com](mailto:victor.shepardson@gmail.com).



IEEE MultiMedia serves the community of scholars, developers, practitioners, and students who are interested in multiple media types and work in fields such as image and video processing, audio analysis, text retrieval, and data fusion.

**Read It Today!**

[www.computer.org/multimedia](http://www.computer.org/multimedia)