

Security of Machine Learning

Arash Vahidi & Nicolae Paladi

RISE Cybersecurity

Arash Vahidi

- Sr. researcher at RISE Cybersecurity
- PhD in Formal Methods in automotive control, Chalmers University of Technology, 2005
- Working with separation technologies for embedded systems, cryptographic and network systems for high-security environment



Nicolae Paladi

- Sr. researcher at RISE Cybersecurity
- PhD in Trusted Computing for Cloud Infrastructure, Lund University, 2017.
- His research interests include cloud infrastructure security, SDN security, trusted computing, platform attestation, and secure virtualization.
- Nicolae organizes Sweden Confidential Computing meetup and the OpenStack meetup.



Outline

- Introduction and common terminology
- Importance of data
- Security for ML
- ML for security

Introduction

- As a security professional you probably have seen AI gradually penetrating your field as a new tool. We label this “AI for security”.
- On the other hand, you may also have been involved in security evaluation of solutions that utilize AI. We call this “Security for AI”.
- It may surprise some that AI and security overlap in many areas, but this is because **AI is an enabler technology**. Would anyone find it odd that “security” and “programming” overlap??

Important terminologies

- **Artificial intelligence** (AI) considers intelligence of a man-made machine and includes among others the field of **machine learning** (ML).
- Machine learning attempts to **train** (configure) a mathematical **model** to represent an underlying phenomena given (large amounts) of representative **data**.
- Data is usually an array of **feature** vectors that capture some aspects of this phenomena.
- **Inference** refers to using the learned model at runtime with live data.
- The exact structure of the model and the data as well as the training and the inference process may differ across applications. In this presentation we will mostly try to ignore such details.

Important terminologies (cont.)

- **True/False Positive** and **Negative** rates refer to ratio of predicted vs actual items in each class.
- For example, in an Intrusion Detection System False Positive means false alarms and False negatives means detections evaded.
- **Accuracy** = $(TP + TN) / (P + N)$

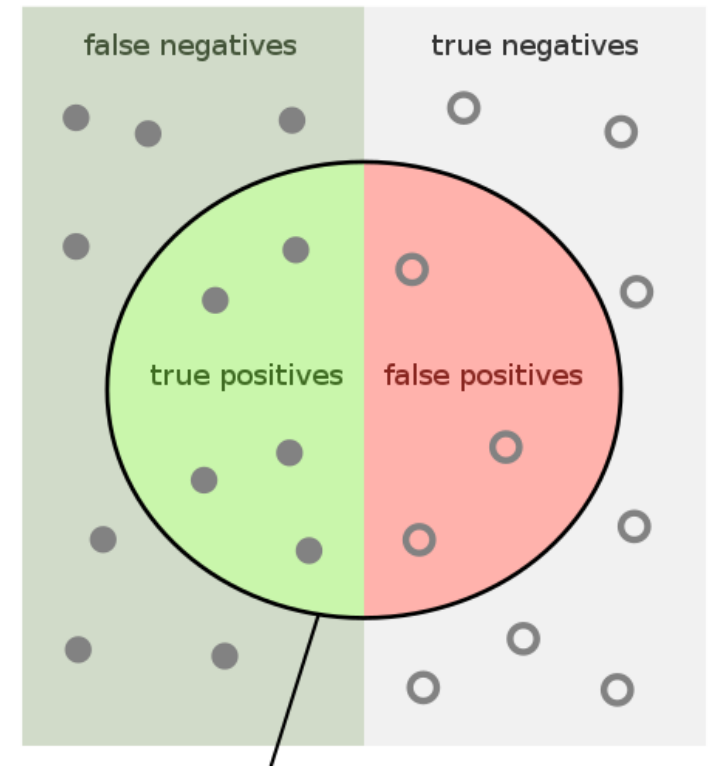


Image credit: wikipedia

Importance of data

If all your friends jumped
off a bridge then would
you too?

We will get back to this important question in a minute!

Amazon's Face Recognition Falsely Matched 28 Members of Congress With Mugshots



By [Jacob Snow](#), Technology & Civil Liberties Attorney, ACLU of Northern California

JULY 26, 2018 | 8:00 AM






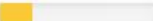







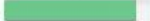

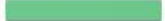
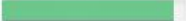

TAGS: [Face Recognition Technology](#), [Surveillance Technologies](#), [Privacy & Technology](#)



Amazon's face surveillance technology is the target of growing opposition nationwide, and today, there are 28 more causes for concern. In a test the ACLU recently conducted of the facial recognition tool, called "Rekognition," the software incorrectly matched 28 members of Congress, identifying them as other people who have been arrested for a crime.

The members of Congress who were falsely matched with the mugshot



Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
 Microsoft	94.0% 	79.2% 	100% 	98.3% 	20.8% 
 FACE++	99.3% 	65.5% 	99.2% 	94.0% 	33.8% 
 IBM	88.0% 	65.3% 	99.7% 	92.9% 	34.4% 



“Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification”,

*Joy Buolamwini, Timnit Gebru,
2018.*

Understanding bias

- It might be easy to visualize and understand the problem in case of gender and racial bias. But would you be able to spot other types of bias in your data or detect similar problems?
- Now, would you **trust** a critical ML system that scores 100.0% accuracy on training data - without having a full understanding of the training data **content** and **coverage**?

Data Readiness

- Based on work by Neil Lawrence, “data readiness” considers the following readiness levels (*somewhat simplified here*):
 - Level C: data exists (from C4, “I heard someone has some data” to C1, “here is some files I found”)
 - Level B: data exists and is a faithful representation (B1)
 - Level A: correct data exists and is appropriate for this task (A1)
- In my experience, we are usually around C1.

If all your friends jumped
off a bridge then would
you too?

Answer: an ML model would undoubtedly do!

Security for AI

Attacks against AI system

- Generally the following classes of attacks against AI systems are considered:
 - **Poisoning**: manipulate (training) data to affect future behaviour
 - **Evasion**: avoid positive classification (e.g. bypass spam detection)
 - **Impersonation**: dictate desired classification
 - **Inversion**: extract model or training data, or check for membership
- This is all in addition to the standard security issues (availability, integrity, ...).

Instability in AI models

- Many security vulnerabilities are caused by a systems unexpected reaction to a maliciously crafted input (e.g. buffer overflows).
- AI models can be **even more sensitive** to malicious input. And often we can't explain why...

("One pixel attack for fooling deep neural networks", J. Su et al., 2017)



Cup(16.48%)
Soup Bowl(16.74%)



Bassinet(16.59%)
Paper Towel(16.21%)



Teapot(24.99%)
Joystick(37.39%)



Hamster(35.79%)
Nipple(42.36%)

Instability in AI models (cont.)

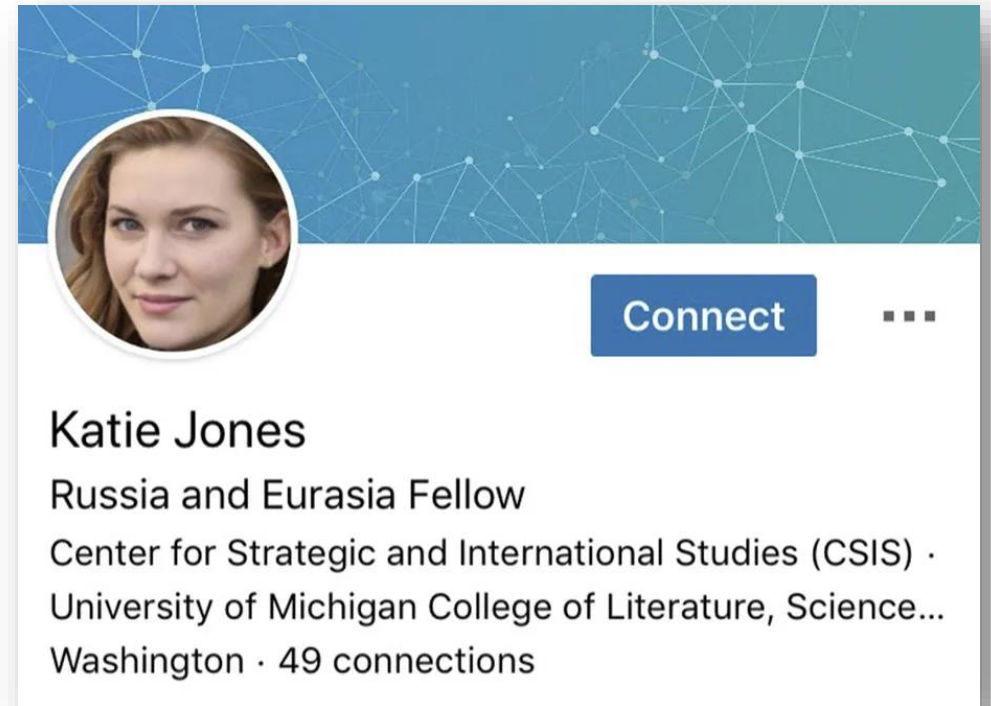
- According to Wolpert-Macready **No Free Lunch Theorem** (NFLT) if we consider all possible data generating distributions then all classification algorithms have the same error rate for previously unobserved data.
- At the same time according to the **manifold hypothesis**, the “interesting” information in a high-dimensional space is normally concentrated to a few low-dimensional manifolds.
- Could we then (1) train the model to better handle small variations in the input and (2) manually reject extreme outlier data points?

Adversarial Machine Learning

- Research by Szegedy et al., suggests that models that perform near human accuracy on training data can have ~100% error rate on select malicious data.
- Adversarial ML uses ML to craft input data to attack another ML system.
- It turns out that ML models can be hardened against adversarial attacks, which normally results in more robust models that better handle input noise and unexpected input.
- Is this the ML equivalent of hiring ethical hackers to harden your system?

Generative Adversarial Network (GAN)

- Essentially two ML models trained in parallel to craft respectively detect machine generated data.
- This can lead to more stable models but also to models that can generate realistic input (which is normally the main reason for use of GAN).
- GAN is a powerful tool, which can sometimes be abused.



Generative Adversarial Network (GAN)

thispersondoesnotexist.com

&

whichfaceisreal.com

Secure ML - design principles

- The standard security practices also applies to ML systems. For example, it is relatively easy to apply Saltzer and Schroeder's design principles for protection of information in computer system to this field.
- The principles are: economy of mechanism, fail-safe defaults, complete mediation, open design, separation of privilege, least privilege, least common mechanism, psychological acceptability, work factor, compromise recording.

Secure ML - design principles (cont.)

- **Open design:** This is the usual security through obscurity problem. And while you might consider your model a business secret, remember that black-box ML models can be extracted by attackers (model inversion).
- **Separation of privileges:** Access to data is also a privilege! See for example federated learning or early data anonymisation.
- **Psychological acceptability:** Developers must make it easier to understand why the models thinks a certain way.

Intrinsic and *post hoc* explainability



- A. Use **interpretable models**, e.g. sparse linear models, decision trees, decision rules.
- B. Generate **example data** to explain predictions.
- C. **Quantify contribution** of each data feature to the prediction.

Privacy issues in ML

- Important subjects in this area include data gathering and data anonymization as well as the standard security issues such as secure data processing.
- As these are standard security issues, we will instead focus on the specific ML issues inversion and membership attacks.
- In particular, we will have a (very) brief look at Federated Learning (FL) and Differential Privacy (DP).

Privacy issues in ML(cont.)

- **Federated learning** is a distributed system where users send not data but adjustments to their model as they learn locally.
- FL systems can still leak user data (some recent attacks use GAN).
- **Differential privacy** attempts to improve privacy by better hiding individual's contribution. For example, **ϵ -differential privacy** defines how much noise must be added to data to achieve this.

AI for security

Where can ML improve security?

- ML requires data to function. In most cases, this data must also be correctly labelled.
- Furthermore, the data should contain features that are important for the outcome (we may not know which beforehand) and a reasonable signal-to-noise ratio.
- And in the case of deep learning, we may need very large amounts of data.

Example 1: network monitoring (IDS)

1. Record a few weeks of network traffic
2. Label the malicious packets or streams as such
3. Train a deep-learning model to distinguish between normal and malicious traffic

Possible issues: (1) How do we label data correctly? (2) Can this detect future attacks? (3) Would the model work if deployed to other networks? (4) Will legitimate but unusual traffic cause false alarms?

Example 2: malware detection

1. Gather a large number of normal executables
2. Gather all known malware
3. Train a deep-learning model to distinguish between the two

Possible issues: (1) Can this detect future malware? (2) How common are false positives? (3) Given the model, could malware creators construct new malware that evade detection?

Example 3: CC fraud detection

1. Gather a large number of CC transactions
2. Label fraudulent transactions as such
3. Train a deep-learning model to distinguish between the two

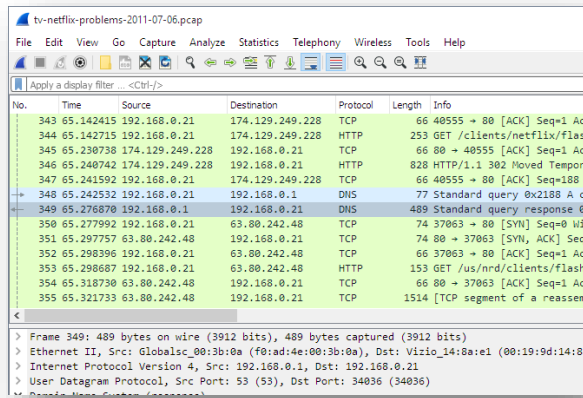
Possible issues: (1) How do we label data correctly? (2) What about security/privacy issues? (3) What are the real-world implications of a false negative?

Example 4: find bugs and vulnerabilities

1. Use commit data from a number of repositories
2. When a security issue is fixed, label affected code as a bug
3. Train a deep-learning model to distinguish between the two

Possible issues: (1) How do we represent code as a feature vector? (2) What if the security fix also changed unrelated parts? (3) Would this generate a lot of false positives/negatives? (4) How hard is it to interpret the results?

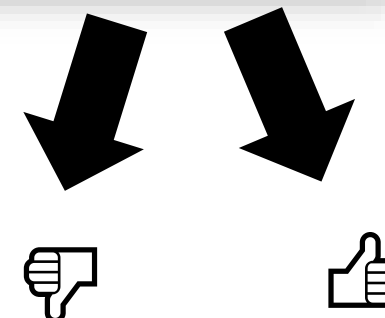
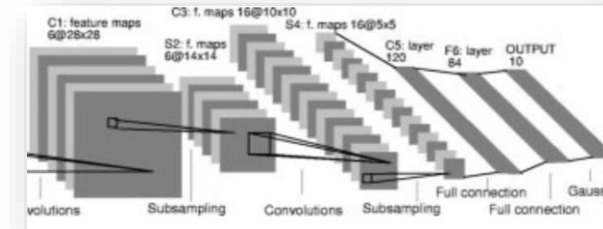
Example: IDS and image recognition



tv-netflix-problems-2011-07-06.pcap

No.	Time	Source	Destination	Protocol	Length	Info
343	65.142415	192.168.0.21	174.129.249.228	TCP	66	40555 → 80 [ACK] Seq=1 Ac
344	65.142715	192.168.0.21	174.129.249.228	HTTP	253	GET /clients/netflix/flash
345	65.230738	174.129.249.228	192.168.0.21	TCP	66	80 → 40555 [ACK] Seq=1 Ac
346	65.240742	174.129.249.228	192.168.0.21	HTTP	828	HTTP/1.1 302 Moved Tempor
347	65.241592	192.168.0.21	174.129.249.228	TCP	66	40555 → 80 [ACK] Seq=188
348	65.242532	192.168.0.21	192.168.0.1	DNS	77	Standard query 0x2188 A c
349	65.276870	192.168.0.1	192.168.0.21	DNS	489	Standard query response 0
350	65.277992	192.168.0.21	63.80.242.48	TCP	74	37063 → 80 [SYN] Seq=0 Wi
351	65.297757	63.80.242.48	192.168.0.21	TCP	74	80 → 37063 [SYN, ACK] Seq
352	65.298396	192.168.0.21	63.80.242.48	TCP	66	37063 → 80 [ACK] Seq=1 Ac
353	65.298687	192.168.0.21	63.80.242.48	HTTP	153	GET /us/mrd/clients/flash
354	65.318730	63.80.242.48	192.168.0.21	TCP	66	80 → 37063 [ACK] Seq=1 Ac
355	65.321733	63.80.242.48	192.168.0.21	TCP	1514	[TCP segment of a reassem

> Frame 349: 489 bytes on wire (3912 bits), 489 bytes captured (3912 bits)
> Ethernet II, Src: Globalsc_00:3b:0a (f0:ad:4e:00:3b:0a), Dst: Vizio_14:8a:e1 (00:19:9d:14:8a:e1)
> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.21
> User Datagram Protocol, Src Port: 53 (53), Dst Port: 34036 (34036)

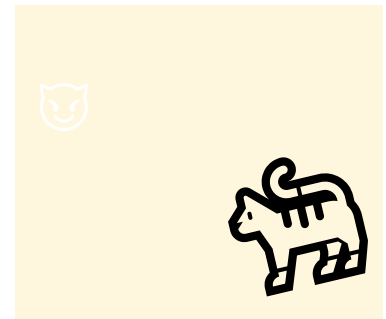
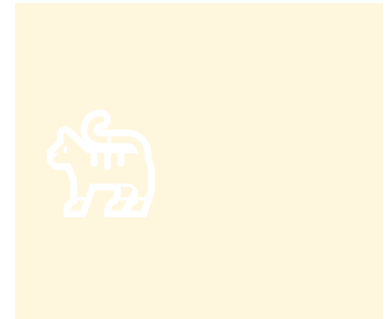


Would this be accurate?

- Assume we have an accuracy of 99.9%. Is this a good result?
 - Assume we recorded 100000 packets, 10 of which is malicious,
 - Recall that $\text{accuracy} = (\text{TP} + \text{TN}) / (\text{P} + \text{N})$,
 - Given this **imbalance of data**, an algorithm that always guesses “no attack” will have around 99.99% accuracy.
- As an exercise, you could calculate how many microseconds it takes before the first false alarm is raised on a 100Gbps link.

Can it be broken (without using math)?

- Convolutional neural networks have a number of properties that helps them ignore certain variations in images.
- These properties are not always desired in security.
- In practice, an attack could be as easy as reusing parts of a legitimate packet, shifting some fields left/right or toggling one bit in a malicious packet to evade detection.



More on breaking ML models for security

- Different ML constructs have a different inherent weaknesses.
 - Some are limited by how much they can remember from a sequence of events
 - Some are limited by what part of data they focus on
 - Some have trouble distinguishing between permutations of data
 - Sometimes different parts of data can cancel out each other, which can be used to hide the malicious parts.
 - ...
- Any reasonably skilled hacker with minimal knowledge in machine learning could probably find ways to exploit many current ML solutions.

Final words

- Machine Learning is becoming more common in security.
- As security professionals, we need to have some basic understanding of how ML works.
- In particular, we need to understand when ML does not work, when it can be broken and what differs a good ML solutions from a really bad one.
- And at the very least, we need to agree on a common terminologies to better understand each other.

Further reading

- “Datasheets for Datasets” Timnit Gebru et al., 2020.
- “Data Readiness Levels”, Neil D. Lawrence, 2017.
- “A Marauder’s Map of Security and Privacy in Machine Learning”, Nicolas Papernot, 2018.
- “The security of machine learning”, Marco Barreno et al., 2010.
- “Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges”, C. Rudin et al., 2013.
- “Intriguing properties of neural networks”, C. Szegedy et al., 2014.

Thank You