Google

# A Marauder's Map of Security and Privacy in Machine Learning:

An overview of current and future research directions for making machine learning secure and private.

Nicolas Papernot
*Google Brain*

@NicolasPapernot

# Is ML security any different from ~~real-world~~ computer security?

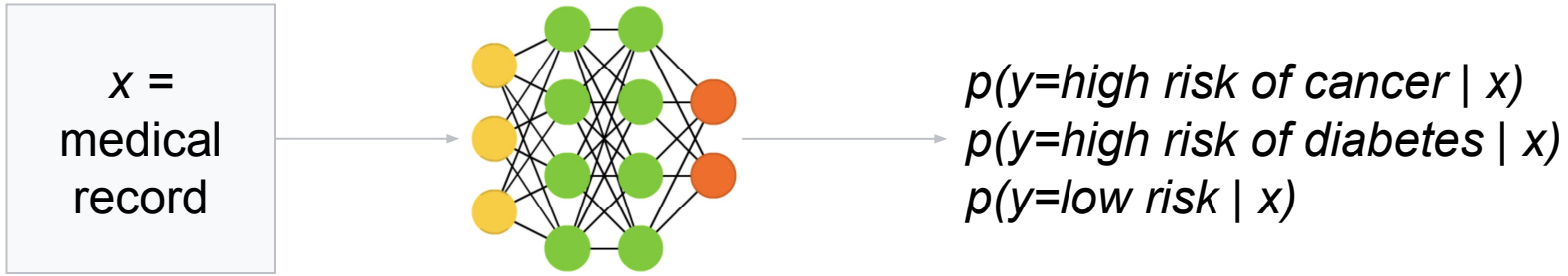

*"Practical security balances the cost of protection and the risk of loss, which is the cost of recovering from a loss times its probability" (*Butler Lampson, 2004)

**Is the ML paradigm fundamentally different in a way that enables systematic approaches to security and privacy?**

**Example:** ensembling models vs. OS

# Problems with the ML paradigm even in benign settings

**Example:** risk model for medical insurance provider

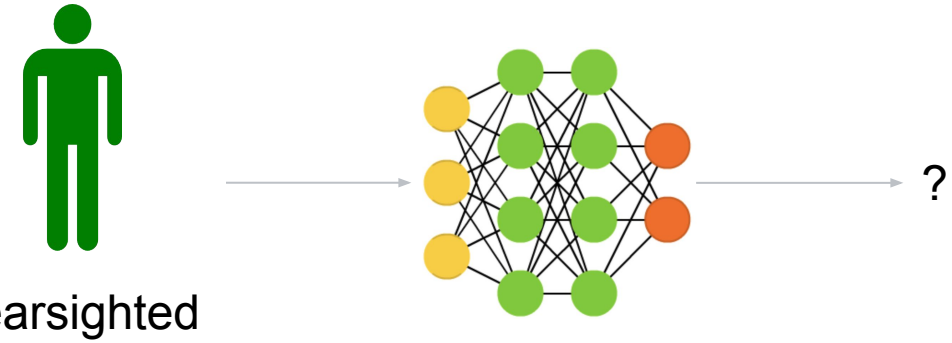



**Assumption:** training distribution = test distribution

**Training goal:** minimize cross-entropy between labels and model predictions

# Problems with the ML paradigm even in benign settings

**Example:** risk model for medical insurance provider




Nearsighted

~~**Assumption:** training distribution = test distribution~~

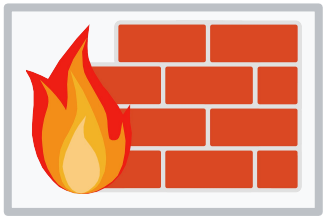
~~**Training goal:** minimize cross-entropy between labels and model predictions~~

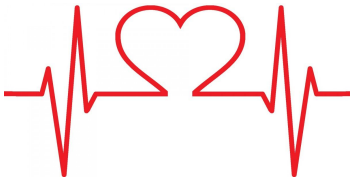
# The ML paradigm in adversarial settings

**What if an...**

**… adversary perturbs medical records to pay less insurance?**
**… adversary attempts to recover medical records used to train the model?**

"When a measure becomes a target, it ceases to be a good measure."

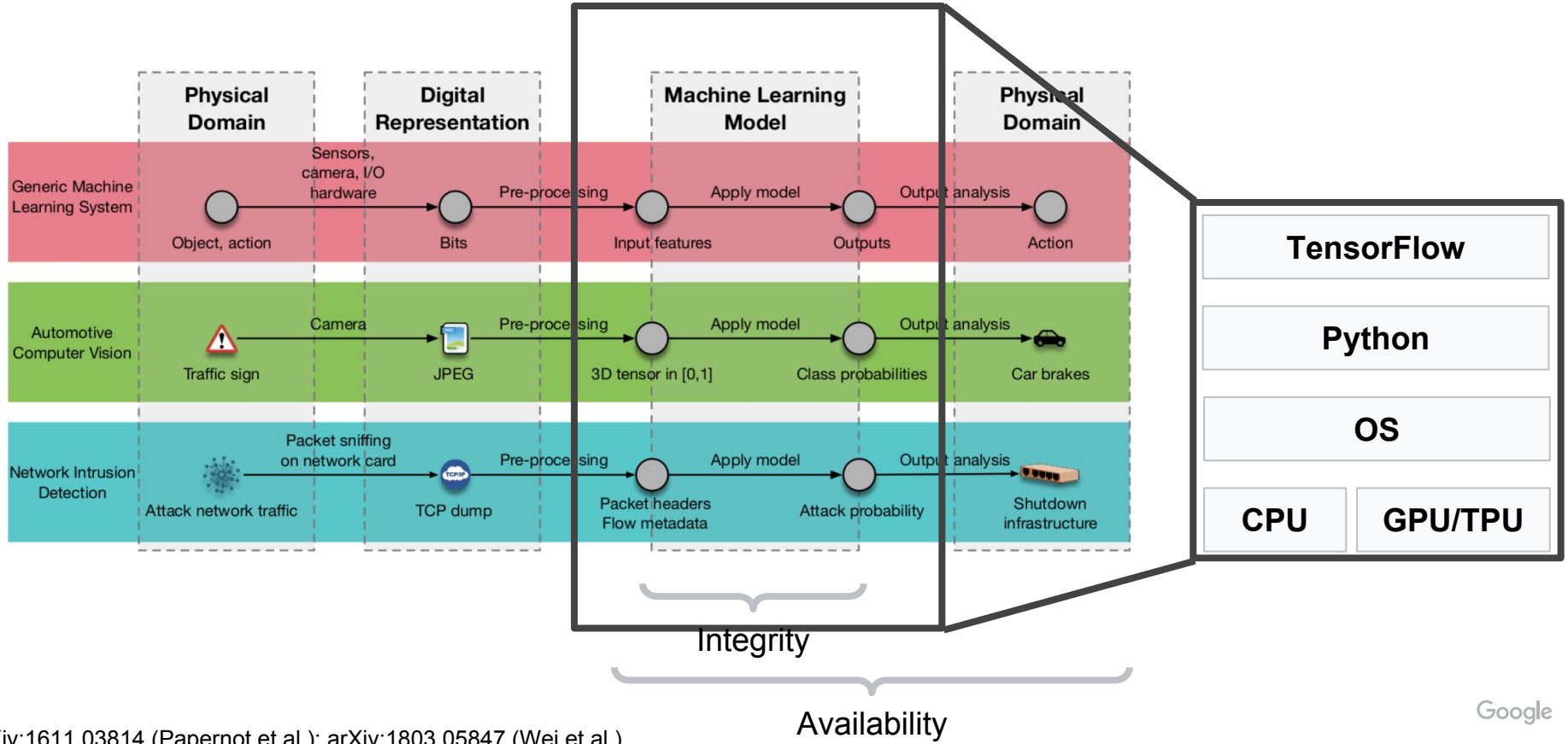





Google

arXiv:1708.06131 (Biggio et al.); arXiv:1312.6199 (Szegedy et al.); ACM:2813677 (Fredrikson et al.); arXiv:1606.04435 (Grosse et al.)

# What is the trusted computing base?




arXiv:1611.03814 (Papernot et al.); arXiv:1803.05847 (Wei et al.)

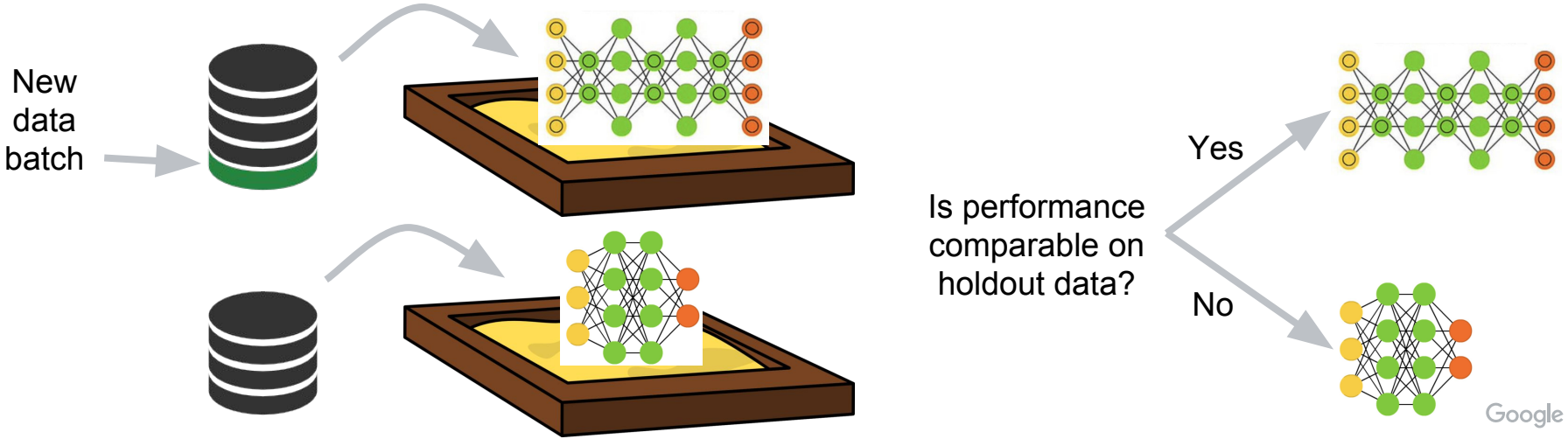
# Revisiting Saltzer and Schroeder's principles

# Fail-safe defaults

**Example 1:** do not output low-confidence predictions at test time

**Example 2:** mitigate data poisoning resulting in a distribution drift

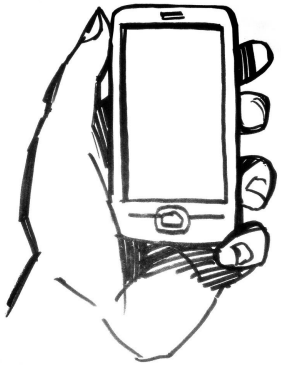
**Attacker:** submits poisoned points to gradually change a model's decision boundary
**Defender:** compares accuracy on holdout validation set ***before*** applying gradients



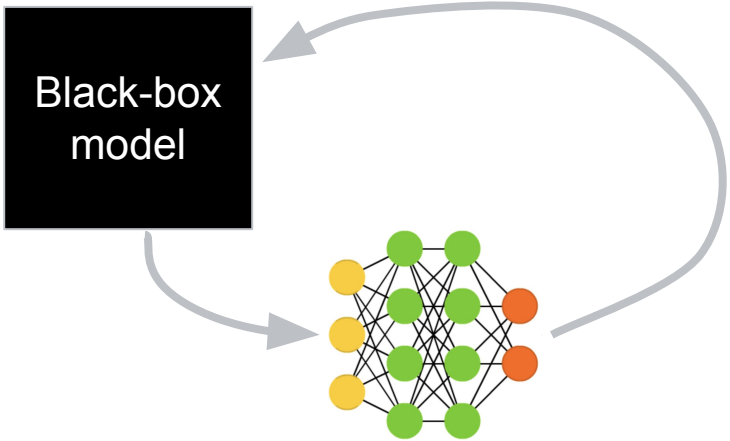
# Open design

**Example 1:** black-box attacks are not particularly more difficult than white-box attacks

Insider leaks model



Reverse engineering




Model extraction

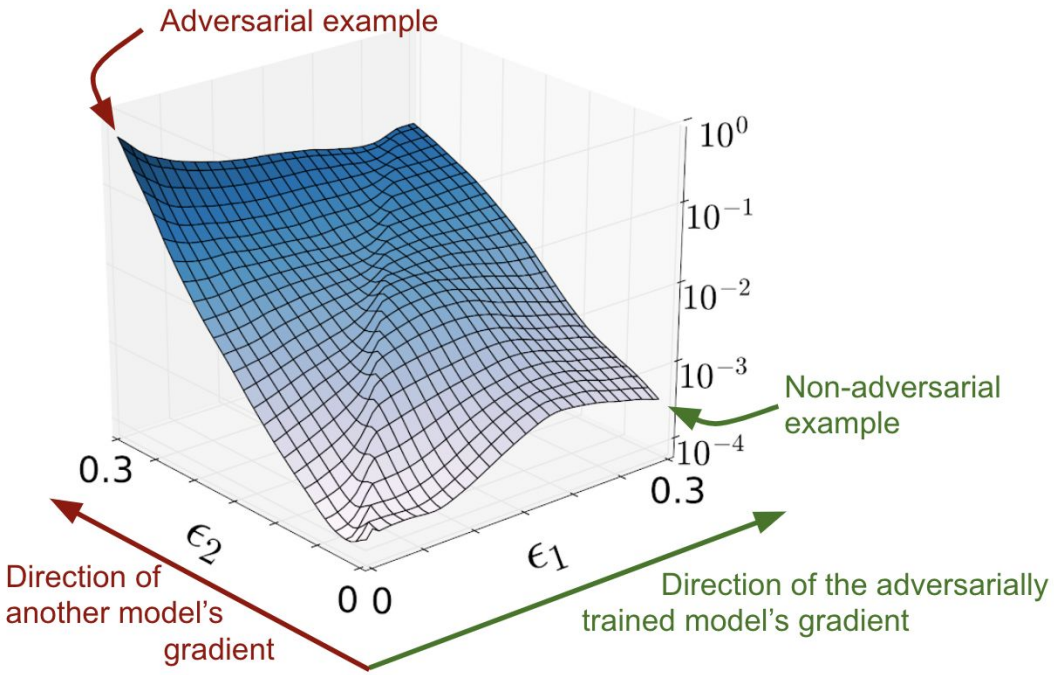
Transferability



ACM:2650798 (Šrndic and Laskov); arXiv:1602.02697 (Papernot et al.)

# Open design

**Example 2:** gradient masking can be circumvented by a black-box attack

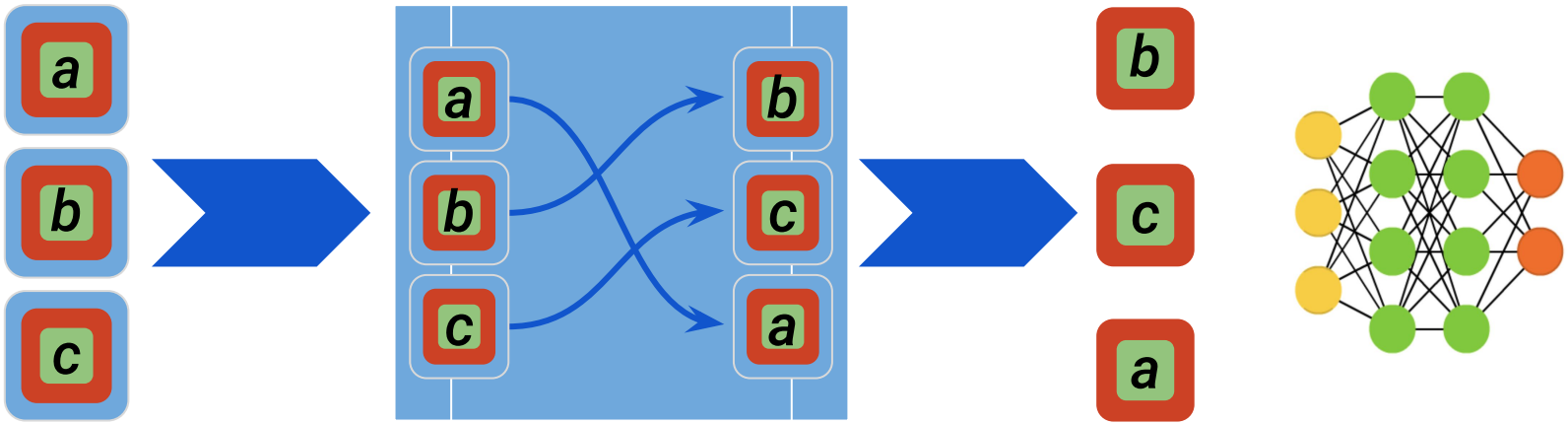





arXiv:1602.02697 (Papernot et al.); arXiv:1705.07204 (Tramer et al.); arXiv:1802.00420 (Athalye et al.)

# Separation of privilege

**Privacy** can be obtained in the **data pipeline** through federated learning or by having different parties encode, shuffle and analyze data in ESA.





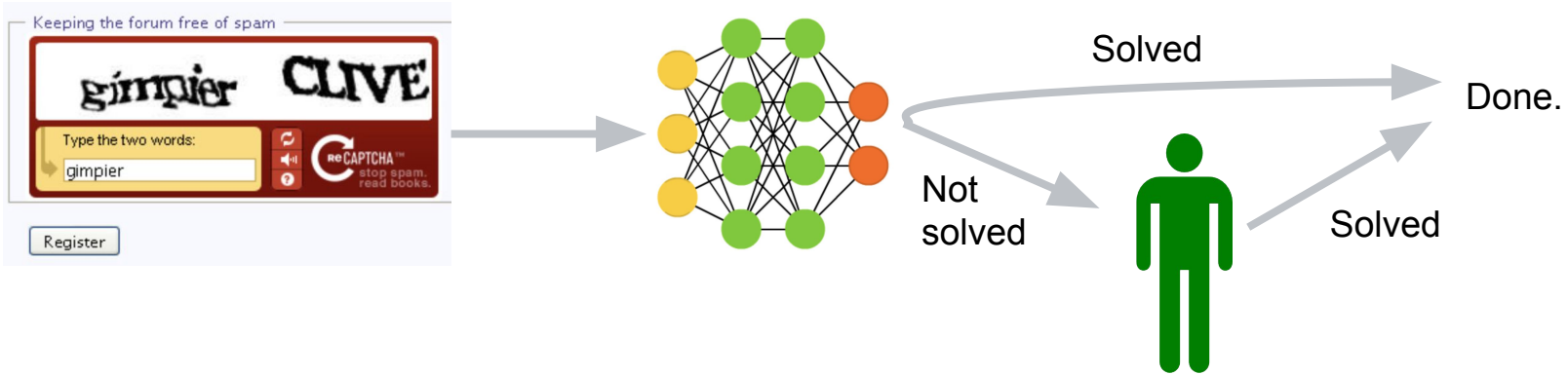

arXiv:1710.00901 (Bittau et al.); arXiv:1602.05629 (McMahan et al.)

# Work factor

**Example 1:** watermarking vs. backdoor insertion

*Does the attacker or defender commit first?*

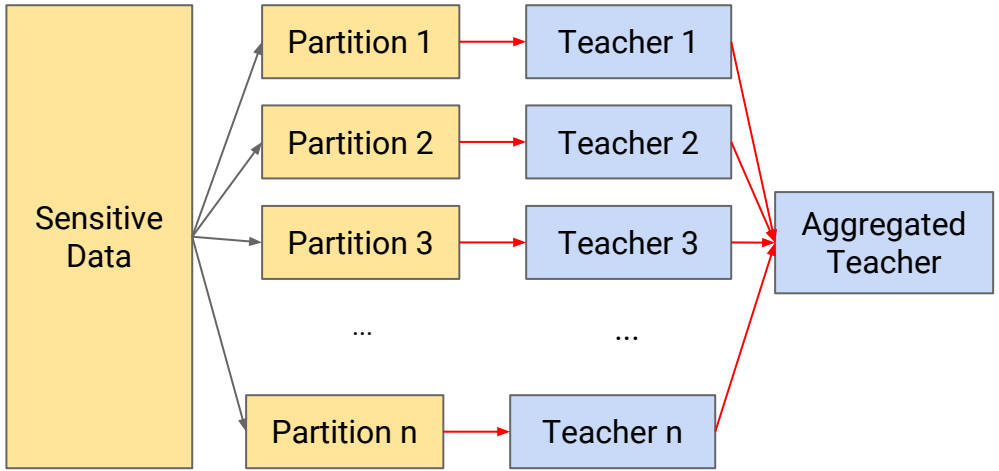
**Example 2:** using adversarial ML to make CAPTCHAs difficult to solve through ML does not increase the adversary's work factor




ACM:1081950 (Lowd and Meek); ACM:1929858 (Motoyama et al.)

# Psychological acceptability

Psychological acceptability suffers when (a) captcha is too hard or (b) outcome is inscrutable.




arXiv:1606.03490 (Lipton); arXiv:1803.04765 (Papernot et al.); arXiv:1610.05755 (Papernot et al.)

# Saltzer and Schroeder's principles

**Economy of mechanism.**
Keep the design of security mechanisms simple.

**Fail-safe defaults.**
Base access decisions on permission rather than exclusion.

**Complete mediation.**
Every access to an object is checked for authority.

**Open design.**
The design of security mechanisms should not be secret.

**Separation of privilege.**
A protection mechanism that requires two keys to unlock is more robust and flexible.

**Least privilege.**
Every user operates with least privileges necessary.

**Least common mechanism.**
Minimize mechanisms depended on by all users.

**Psychological acceptability.**
Human interface designed for ease of use.

**Work factor.**
Balance cost of circumventing the mechanism with known attacker ressources.

**Compromise recording.**
Mechanisms that reliably record compromises can be used in place of mechanisms that prevent loss.

# Model assurance and admission control

# Model assurance and admission control

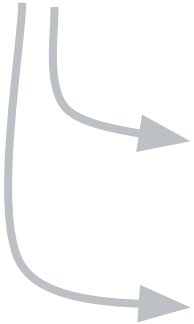
**Machine learning objective:** average-case performance
→ **Testing**

**Security objective:** worst-case performance
→ **Verification**

Exposure
(arXiv:1802.08232,
Carlini et al.)



Differential
privacy analysis



**Model assurance.** (training time)
Establish with confidence that system matches security requirements.

**Admission control.** (test time)
Do we admit an answer for a given input into our pool of answers?
*Combine input validation and sandboxing techniques.*

# How to specify policies for ML security & privacy?

## Security

**Informal security policy:** learning system accurately models *exactly* the end task which the system was designed to solve.

→ Correct implementation (e.g., no numerical instabilities)
→ Solves the end task (e.g., correct predictions on all valid inputs)
→ Only solves the end task (e.g., no backdoor or other poisoned data)

**Open problem:** how to formalize ML security policy with *precise semantics* while avoiding *ambiguity*?

## Privacy

**Privacy policy:** learning behavior does not reflect any private information

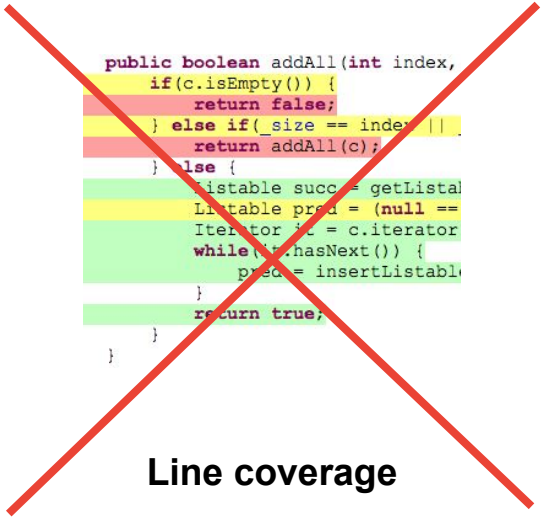
**Formal requirement specification:** differential privacy



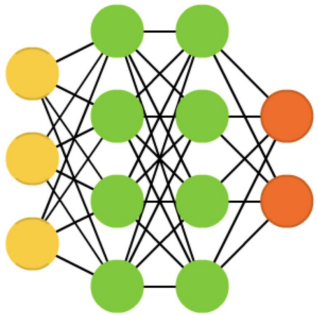
IACR:3650 (Dwork et al.)

# How to measure coverage in ML systems?

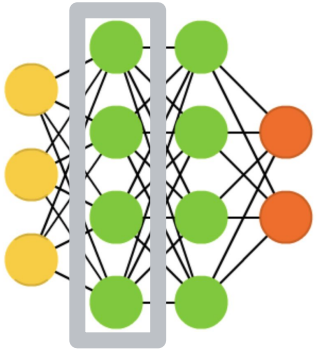
**Logic coverage:** how much of the logic was exercised during verification?



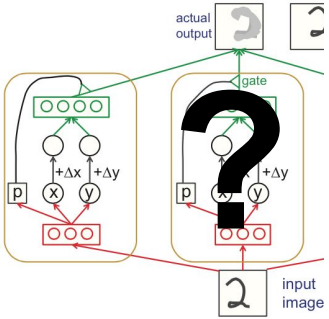

**Line coverage**



**Neural coverage**



**Distance between activation vectors**




**Capsules activity vectors?**

**Input coverage:** how to define the set of valid inputs to the system?
how to bootstrap heuristics?

Google

arXiv:1705.06640 (Pei et al.); arXiv:1807.10875 (Odena and Goodfellow); arXiv:1710.09829 (Sabour et al.)

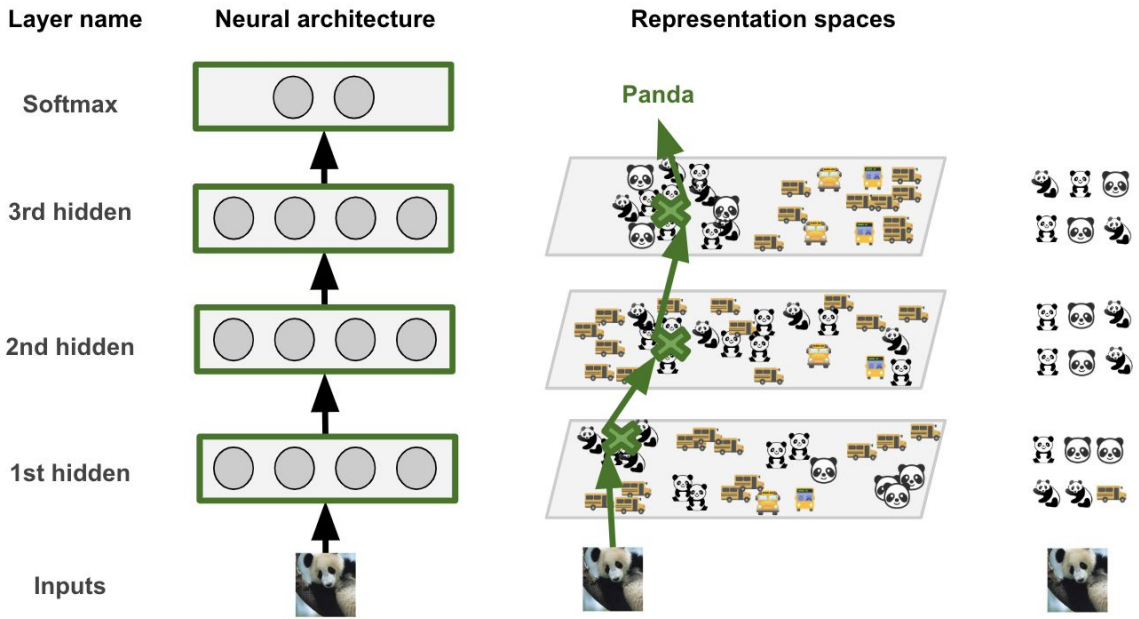
# Test-time admission control of (input,output) pairs

**Weak authentication** (similar to search engines) calls for **admission control:**

*Do we admit a sandboxed model's output into our pool of answers?*

Difficult because task distribution is unknown in ML.

**Example:**
define a well-calibrated estimate of uncertainty to reject outliers




arXiv:1803.04765 (Papernot et al.)

# Towards auditing ML systems

# The case for auditing in ML

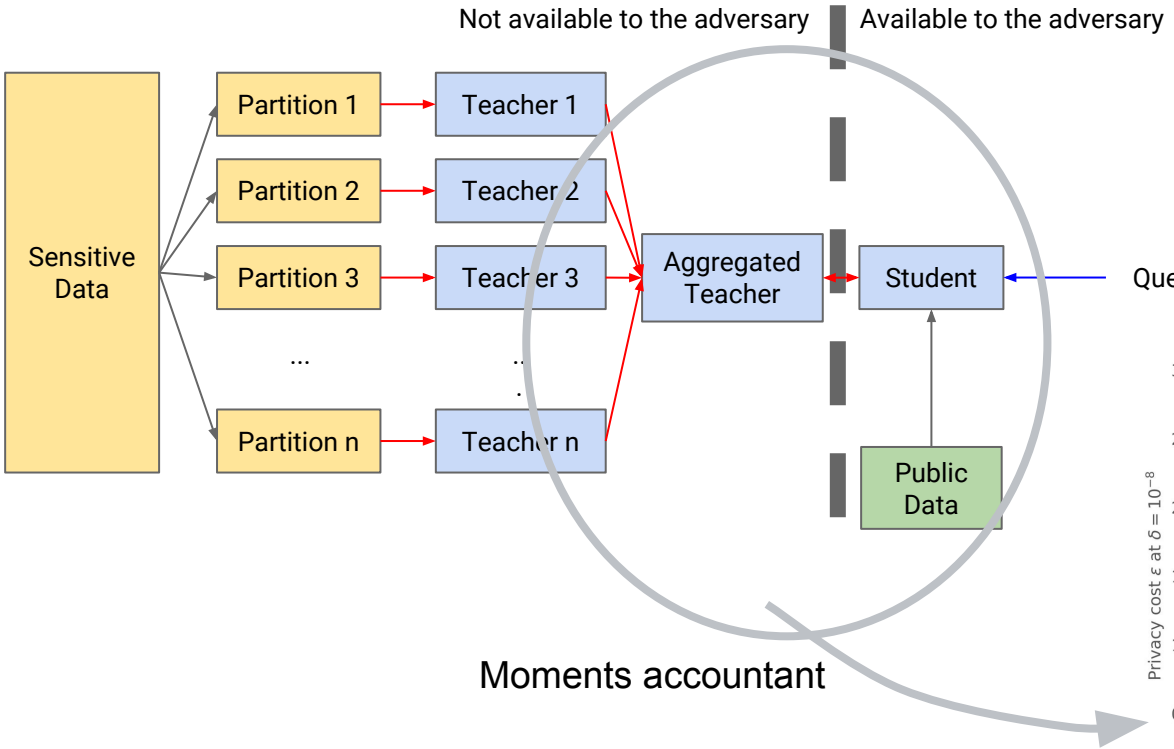
**Auditing:** (1) *identify* information to collect
(2) *analyze* it

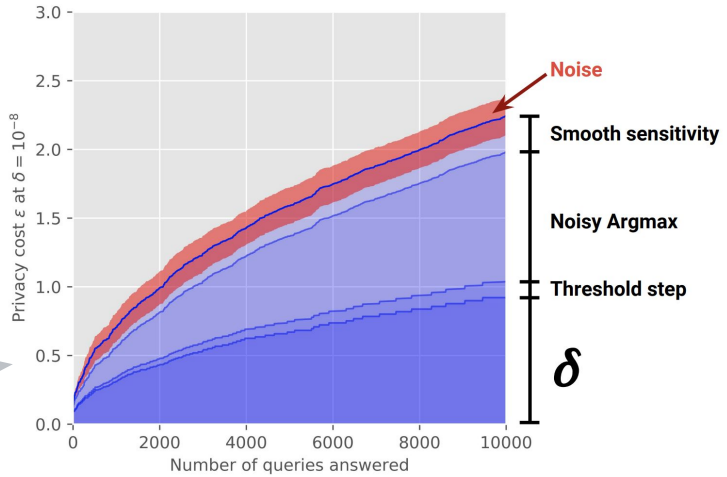

*When systems have weak authentication and authorization, auditing is an important component of security. (John et al., 2010)*

**Auditing design** is informed by specification of security policy.

**Benefits:** reactive and proactive identification of threats
increased work factor and psychological acceptability

# Auditing the learning algorithm: an example for privacy

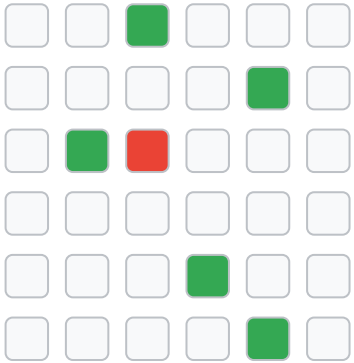


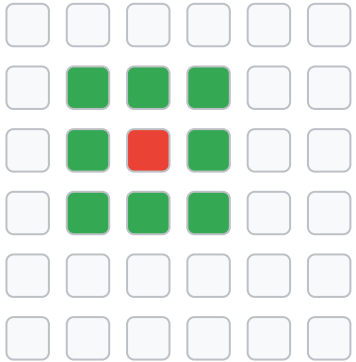

arXiv:1607.00133 (Abadi et al.); arXiv:1802.08908 (Papernot*, Song* et al.); arXiv (Carlini et al.)

# Auditing the inference algorithm

**Example 1:** Record number of queries made on (quasi) training points.



Benign queries



Membership inference region

**Example 2:** Analyze queries to identify possible model extraction

# Formal frameworks that align ML goals with security and privacy

# A comparison with cryptography

Cryptography made a lot of progress once security game (including adversarial capabilities and goals) was identified and defined formally:


Is ML more amenable to the formal specification of security and privacy goals because a large part of the system can be expressed mathematically?

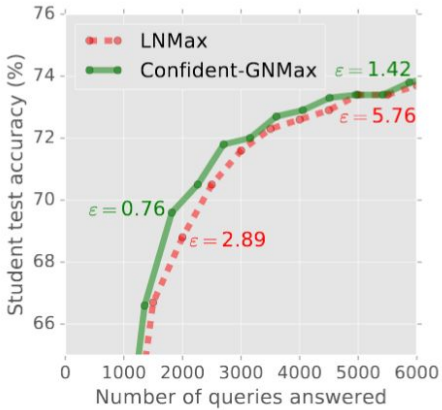
# A great example: differential privacy

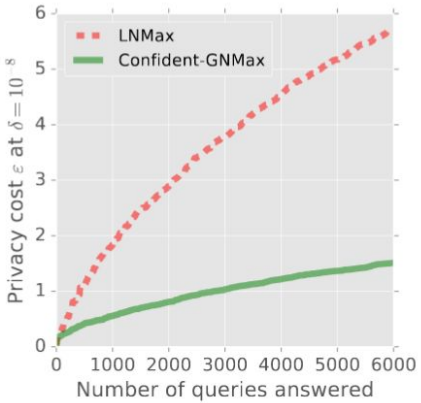
Framework is both **intuitive** and **rigorous**:

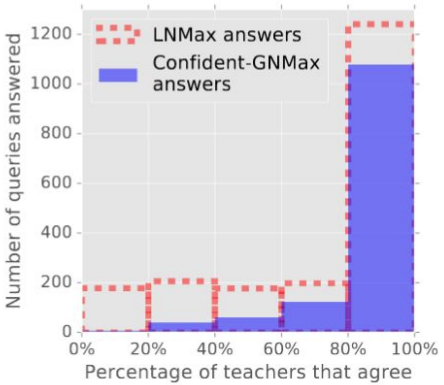
$$Pr[M(d) \in S] \leq e^{\varepsilon} Pr[M(d') \in S]$$

No assumptions made about adversarial capabilities, knowledge or goals.

**Aligns** (worst-case) **privacy** requirements with (average-case) **generalization**




arXiv:1802.08908 (Papernot*, Song* et al.)

# What would a similar framework for ML security look like?

Several questions need to be answered:

- ➔ Should guarantees be formulated wrt training data, algorithm or both?
- ➔ Should the framework encompass training and test time adversaries?
- ➔ How can we provide domain-agnostic formalism?

## Efforts need to specify ML security and privacy policies.

What is the right abstraction and/or language to formalize security and privacy requirements with precise semantics and no ambiguity?

## Admission control and auditing may address lack of assurance.

How can sandboxing, input-output validation and compromise recording help secure ML systems when data provenance and assurance is hard?

## Security and privacy should strive to align with ML goals.

How do private learning and robust learning relate to generalization? How does poisoning relate to learning from noisy data or distribution drifts?

**Ressources:**
cleverhans.io
github.com/tensorflow/cleverhans

**Contact information:**
nicolas@papernot.fr
@NicolasPapernot