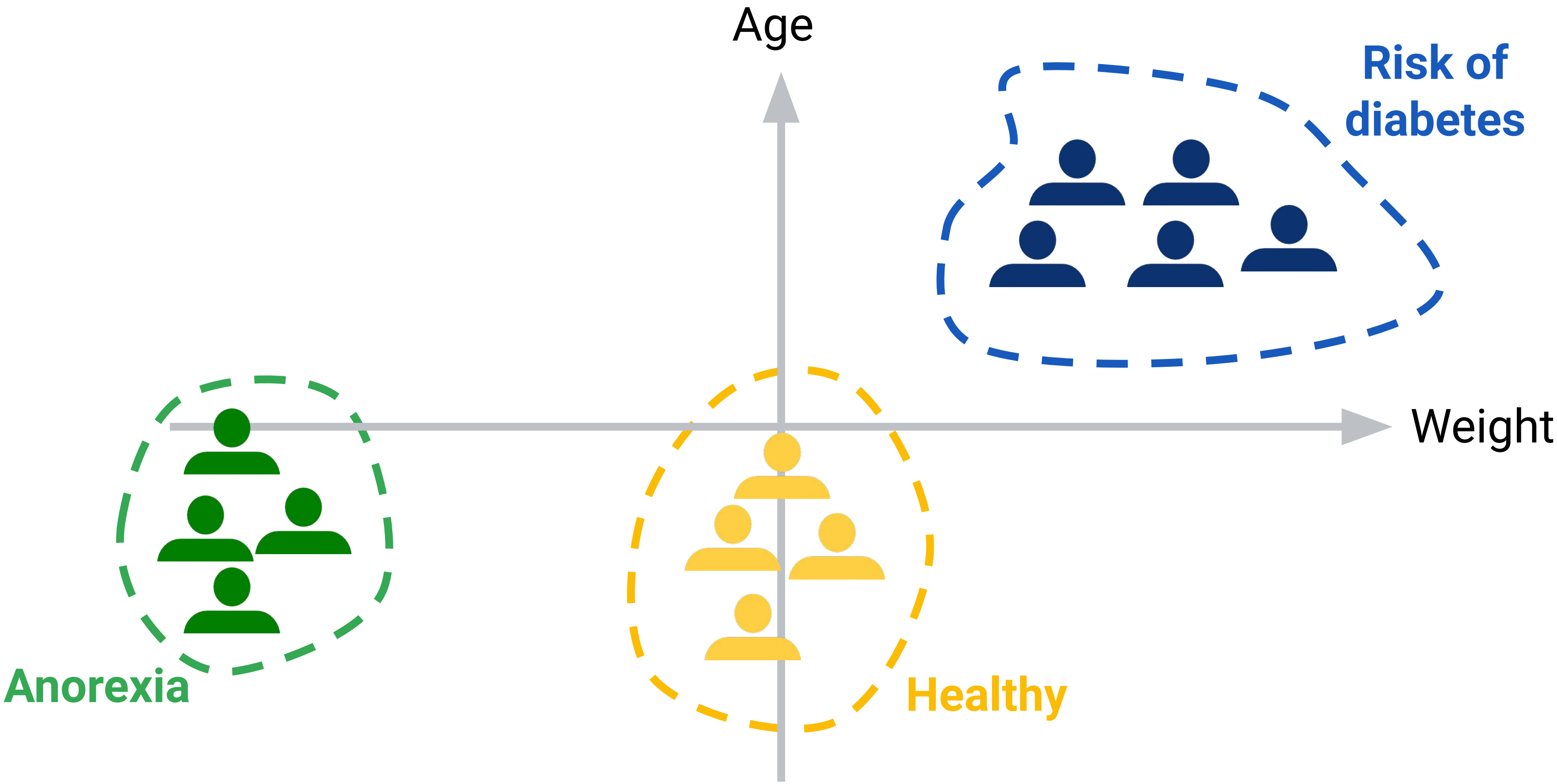
@NicolasPapernot

Security and Privacy in Machine Learning

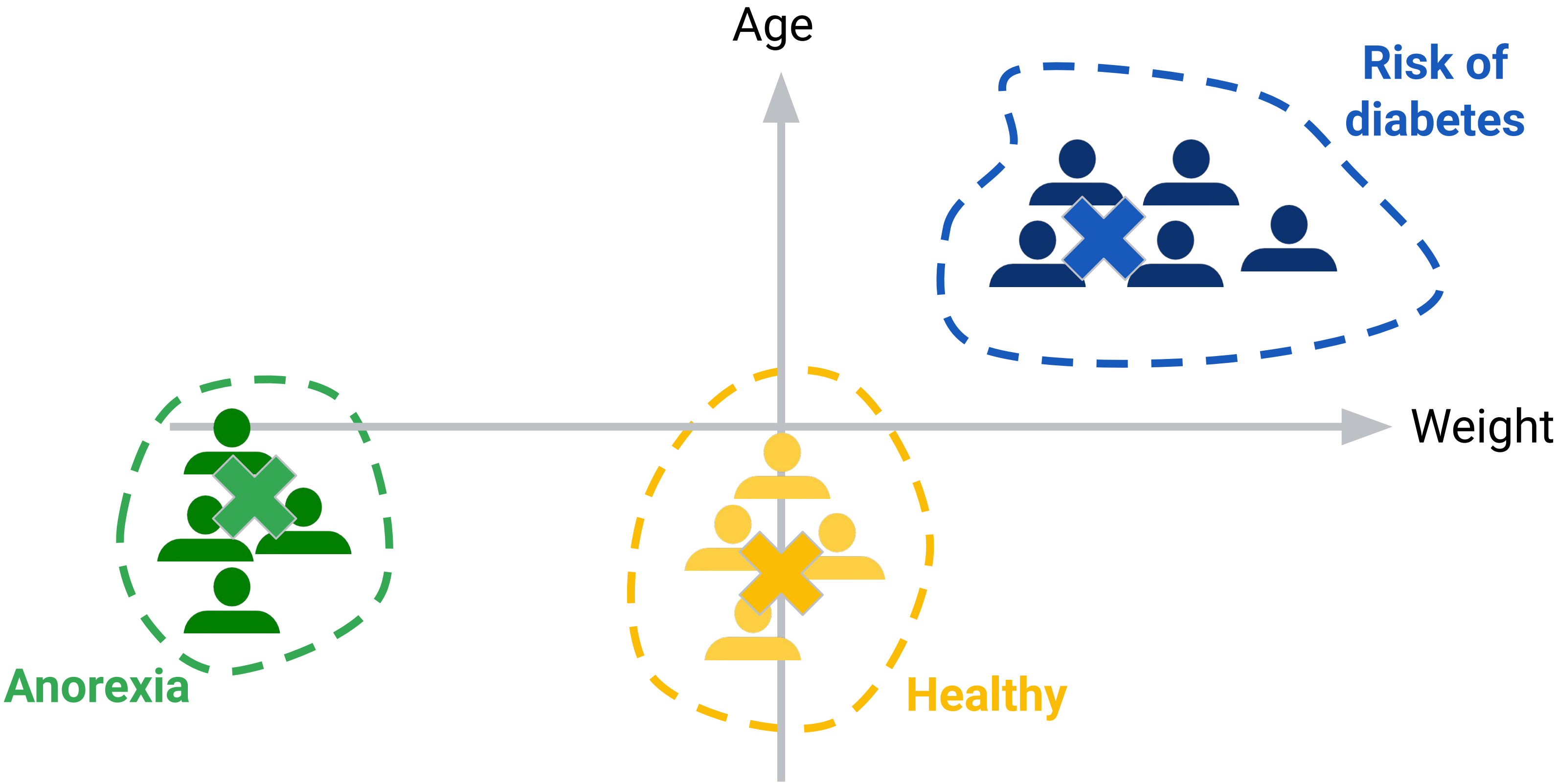
Nicolas Papernot
Google Brain

June 2019 - Microsoft

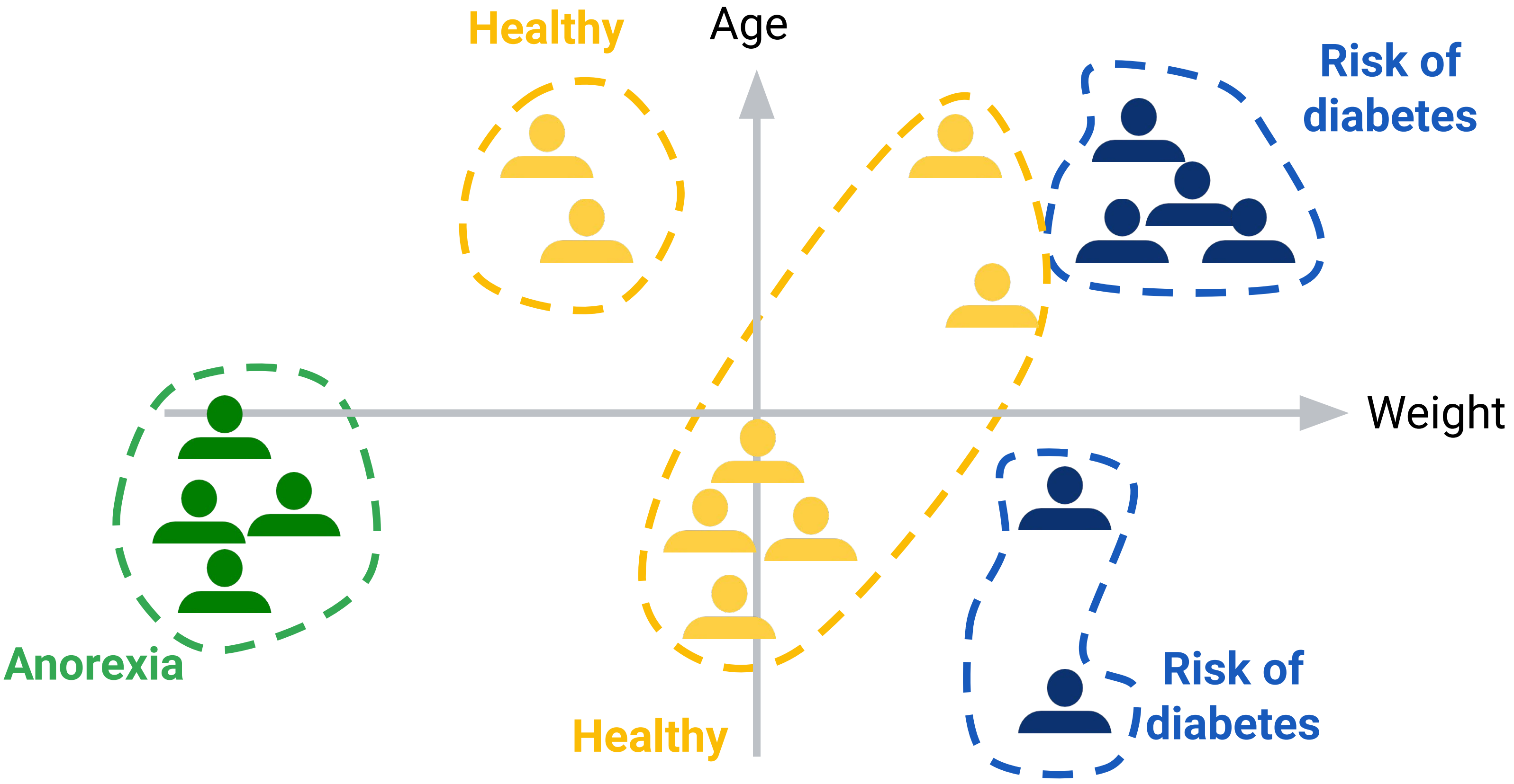
Machine learning is not magic: *ideal setting*



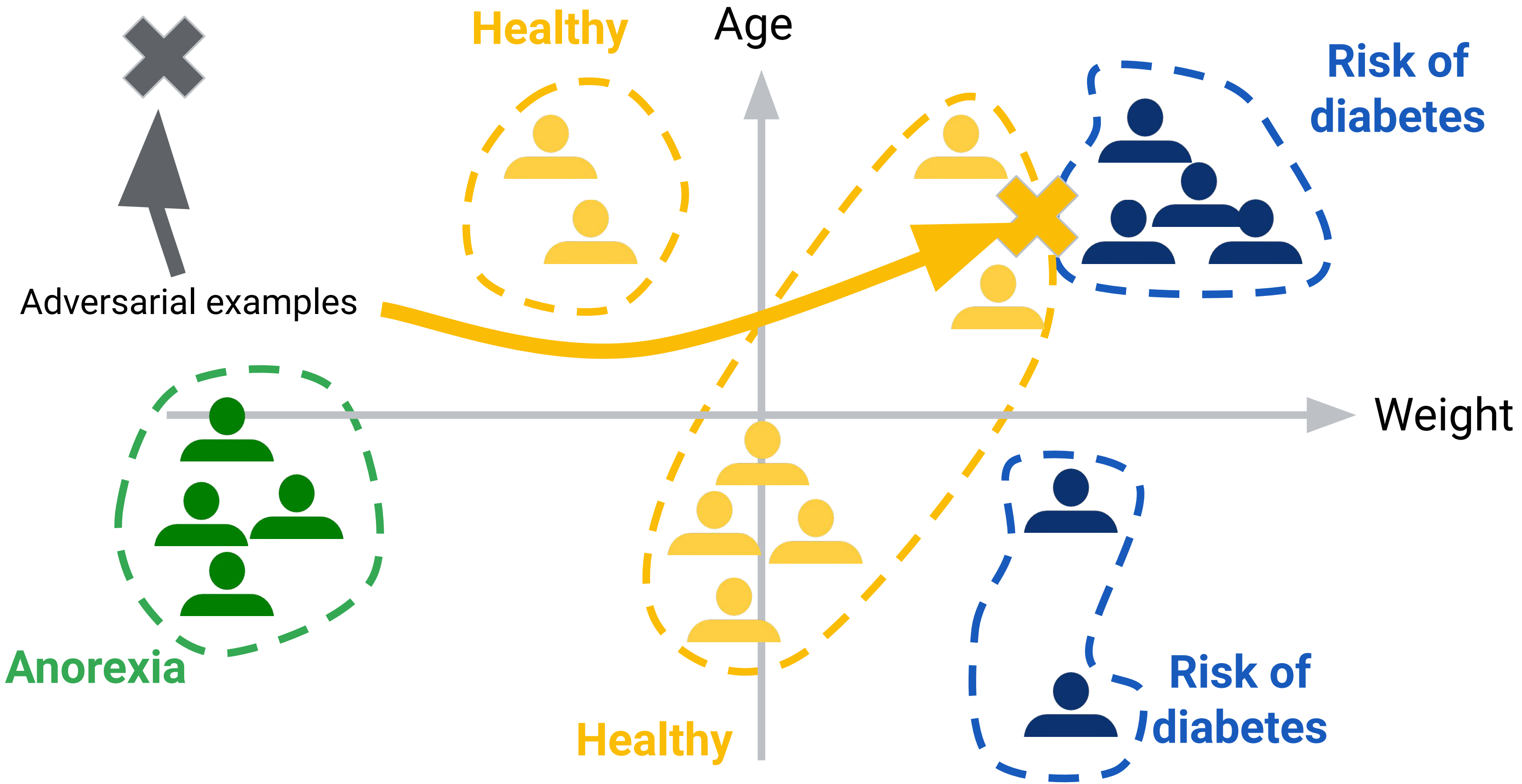
Machine learning is not magic: *ideal setting*



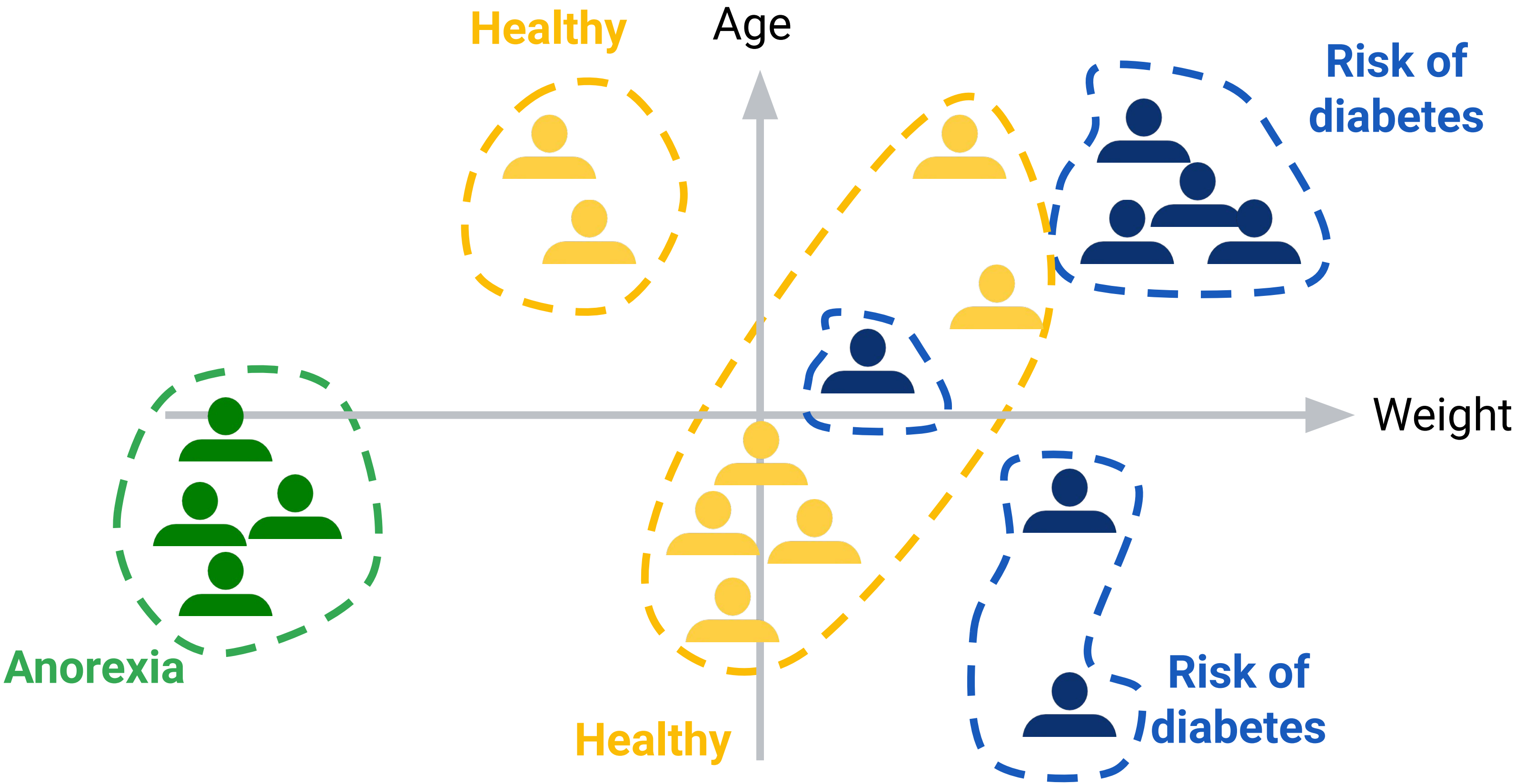
Machine learning is not magic: *(adversarial) real-world*



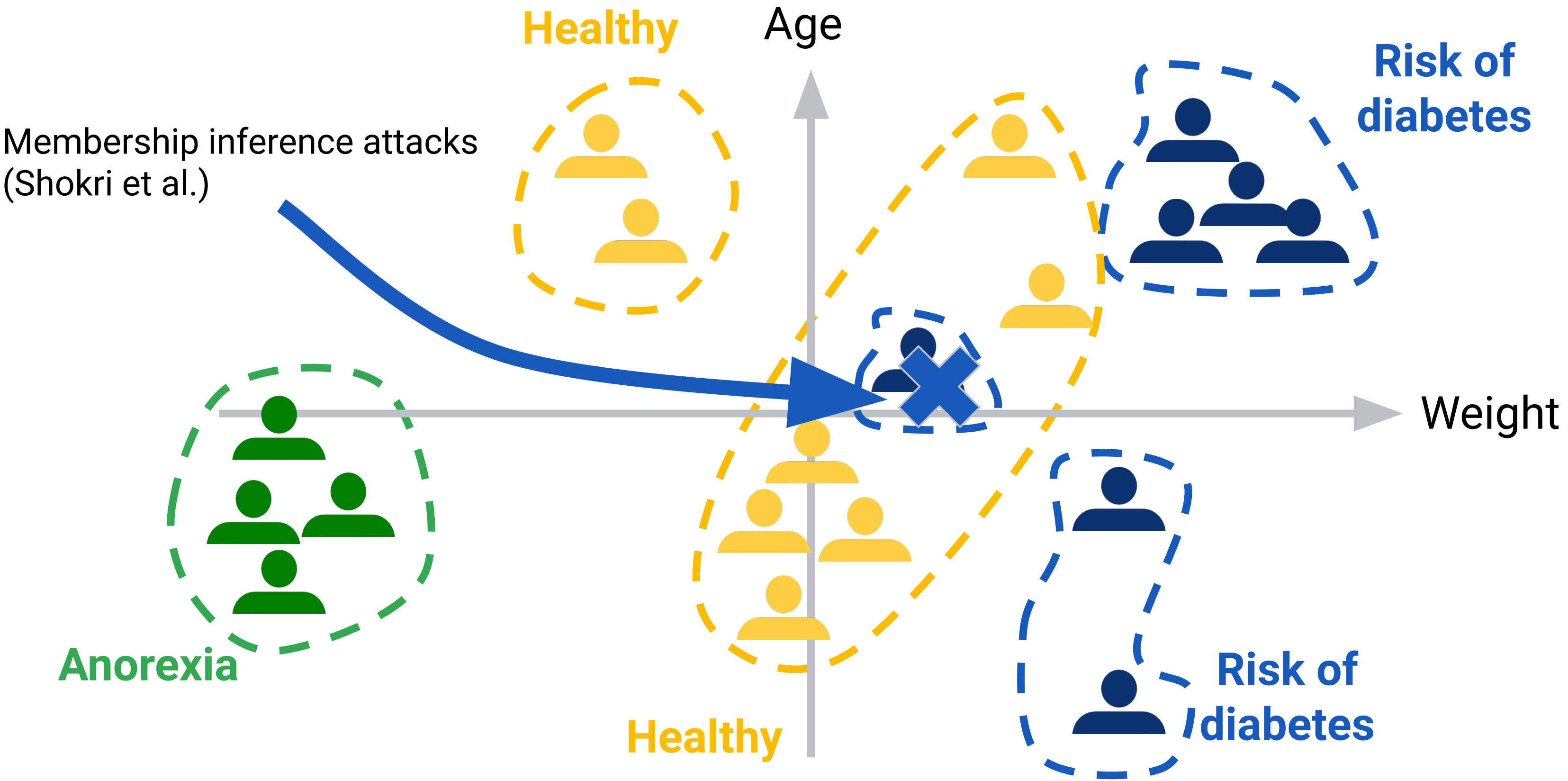
Machine learning is not magic: *(adversarial) real-world*



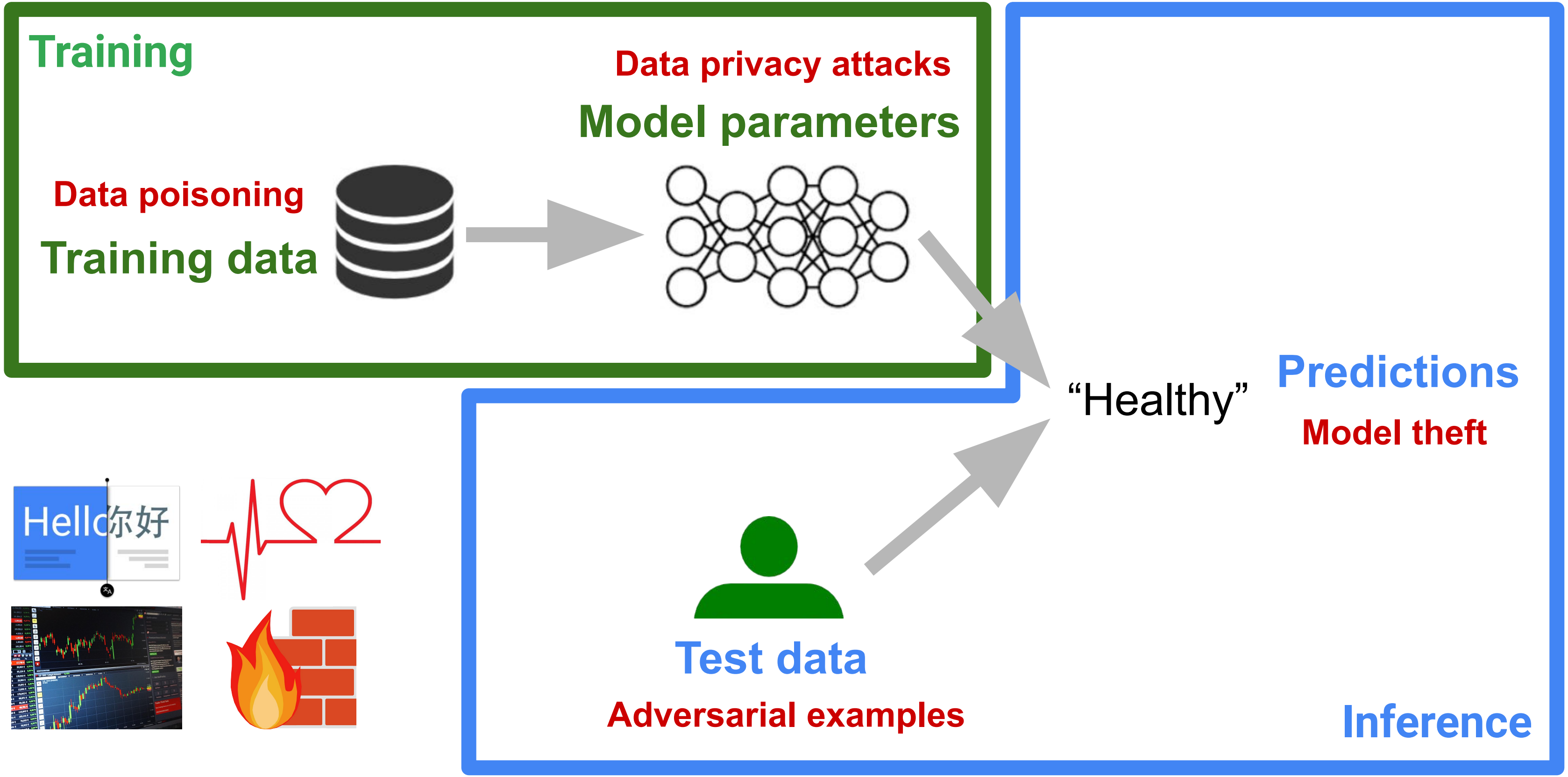
Machine learning is not magic: *(adversarial) real-world*



Machine learning is not magic: *(adversarial) real-world*

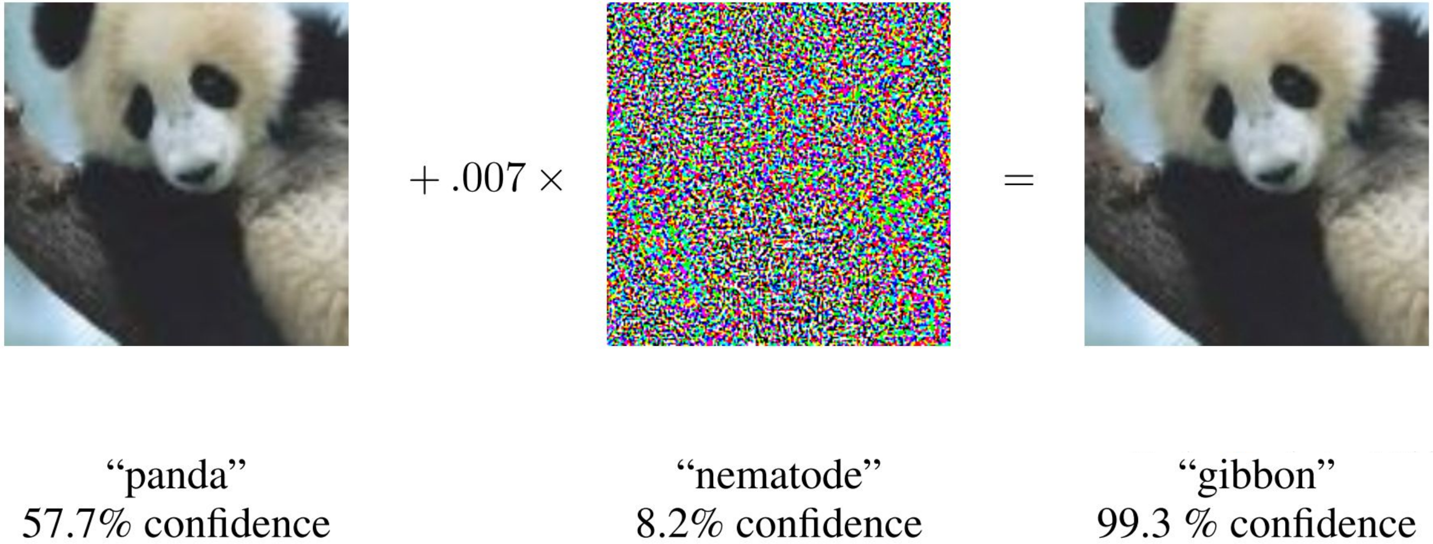


The ML paradigm in adversarial settings



Adapted from a slide by Ian Goodfellow

Adversarial examples



[GSS15] Goodfellow et al. Explaining and Harnessing Adversarial Examples

Crafting adversarial examples: fast gradient sign method

During training, the classifier uses a loss function to **minimize** model prediction errors

After training, **attacker** uses loss function to **maximize** model prediction error

1. Compute its gradient with respect to the input of the model

$$\nabla_x J(\theta, x, y)$$

2. Take the sign of the gradient and multiply it by a threshold

$$x + \varepsilon \cdot sgn(\nabla_x J(\theta, x, y))$$

[GSS15] Goodfellow et al. Explaining and Harnessing Adversarial Examples

Threat model of a black-box attack

Adversarial capabilities

~~Training data~~
~~Model architecture~~
~~Model parameters~~
~~Model scores~~

(limited) oracle
access: *labels*

Adversarial goal

Force a ML model remotely accessible through an API to misclassify

Example

amazon web services™ Microsoft Azure Google Cloud Platform

Our approach to black-box attacks

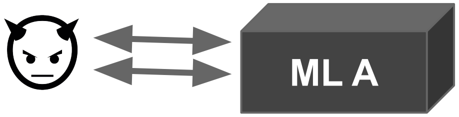
Alleviate lack of knowledge about model

Alleviate lack of training data

Adversarial example transferability

Adversarial examples have a **transferability** property:

samples crafted to mislead a model A are likely to mislead a model B

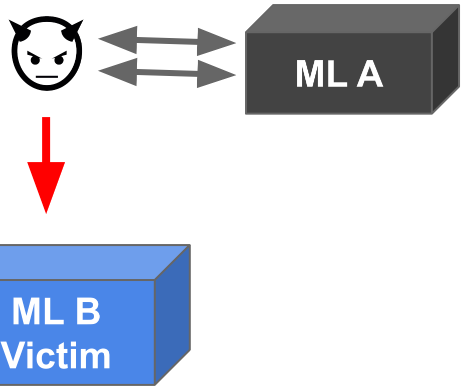


[SZS14] Szegedy et al. Intriguing properties of neural networks

Adversarial example transferability

Adversarial examples have a **transferability** property:

samples crafted to mislead a model A are likely to mislead a model B

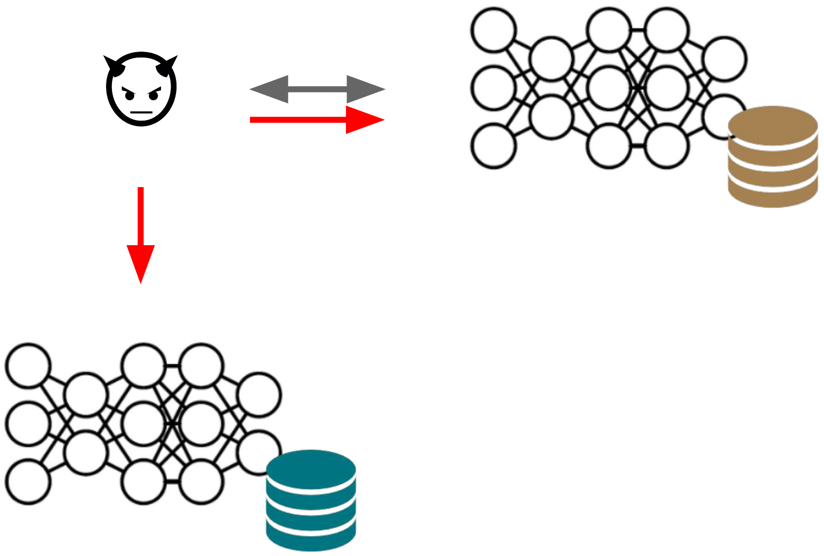


[SZS14] Szegedy et al. Intriguing properties of neural networks

Adversarial example transferability

Adversarial examples have a **transferability** property:

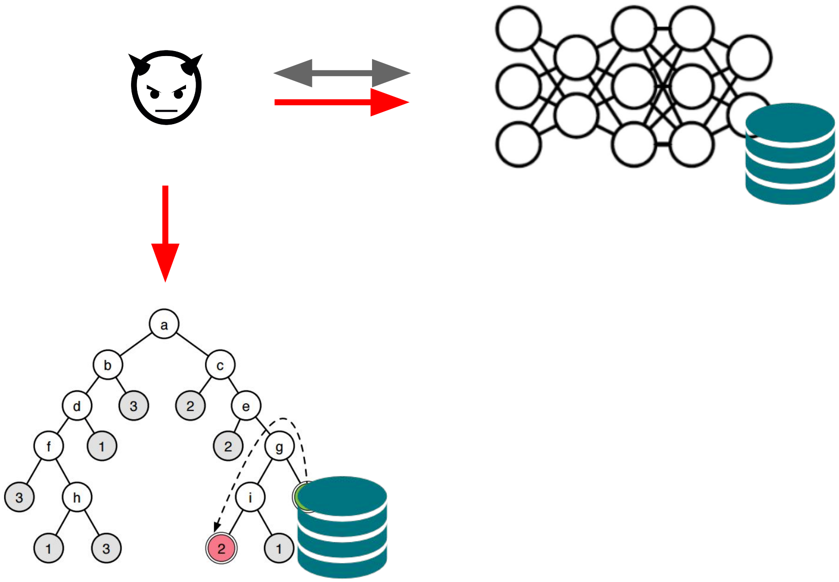
samples crafted to mislead a model A are likely to mislead a model B



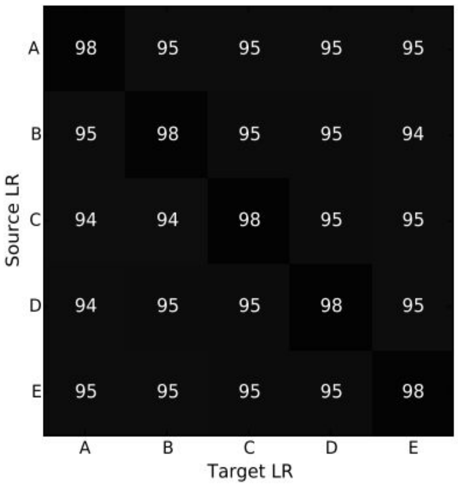
Adversarial example transferability

Adversarial examples have a **transferability** property:

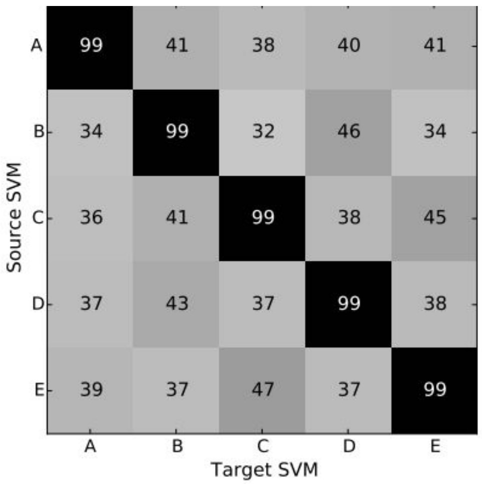
samples crafted to mislead a model A are likely to mislead a model B



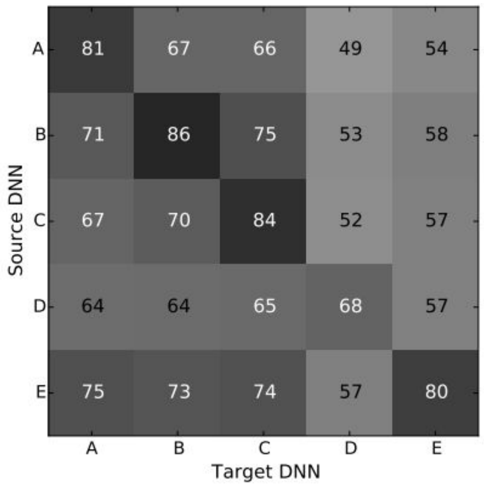
Intra-technique transferability: cross training data



Strong



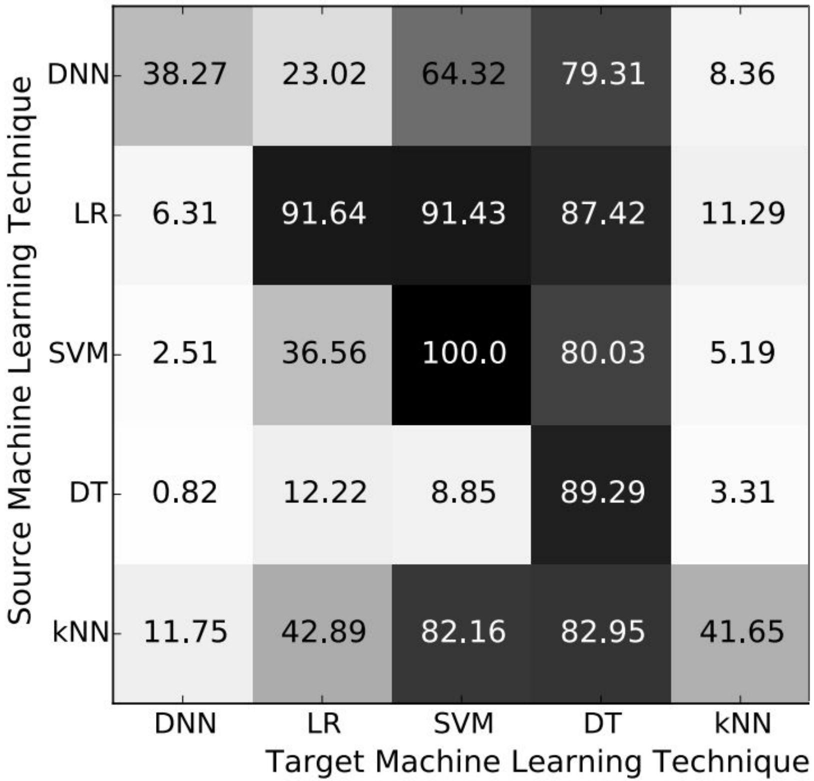
Weak



Intermediate

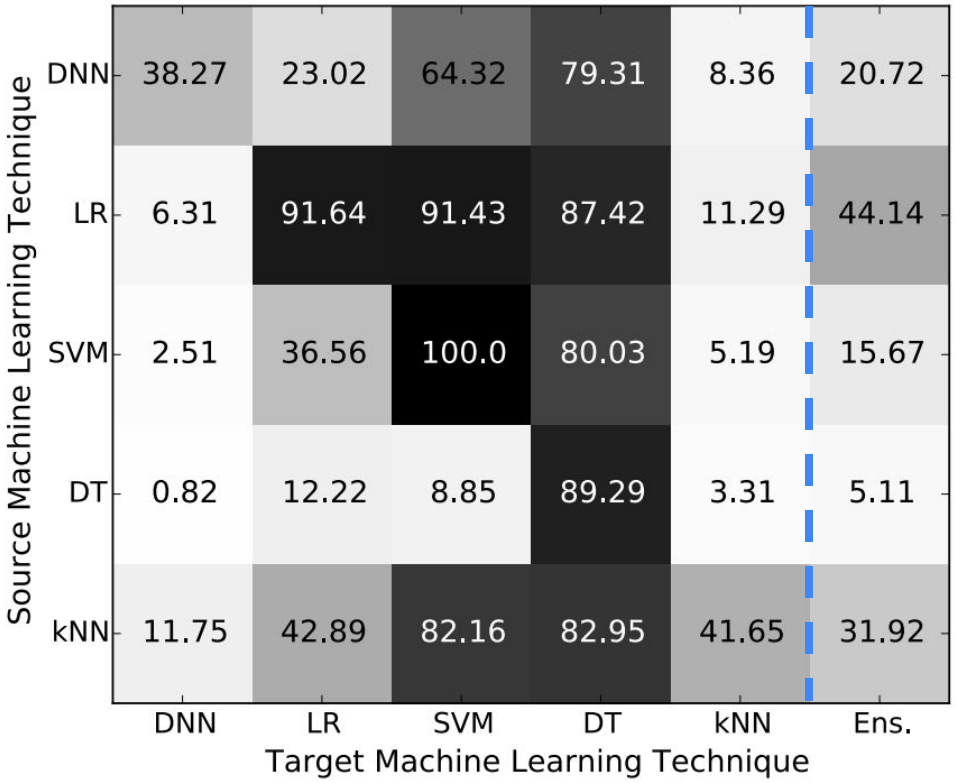
[PMG16b] Papernot et al. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples

Cross-technique transferability



[PMG16b] Papernot et al. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples

Cross-technique transferability



[PMG16b] Papernot et al. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples

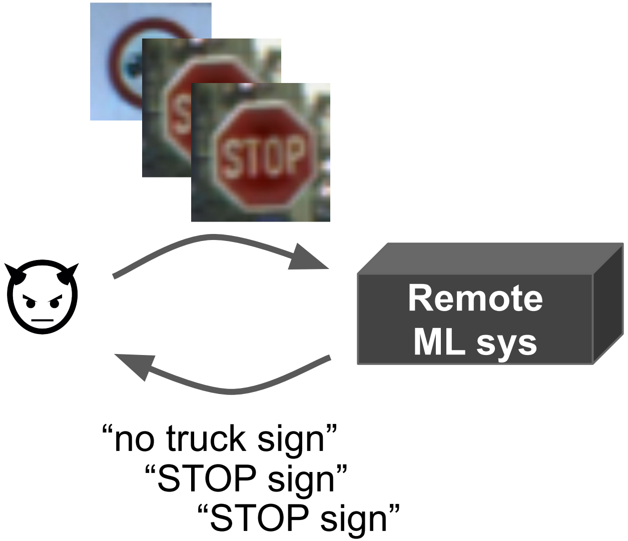
Our approach to black-box attacks

Alleviate lack of knowledge about model

Adversarial example transferability from a substitute model to target model

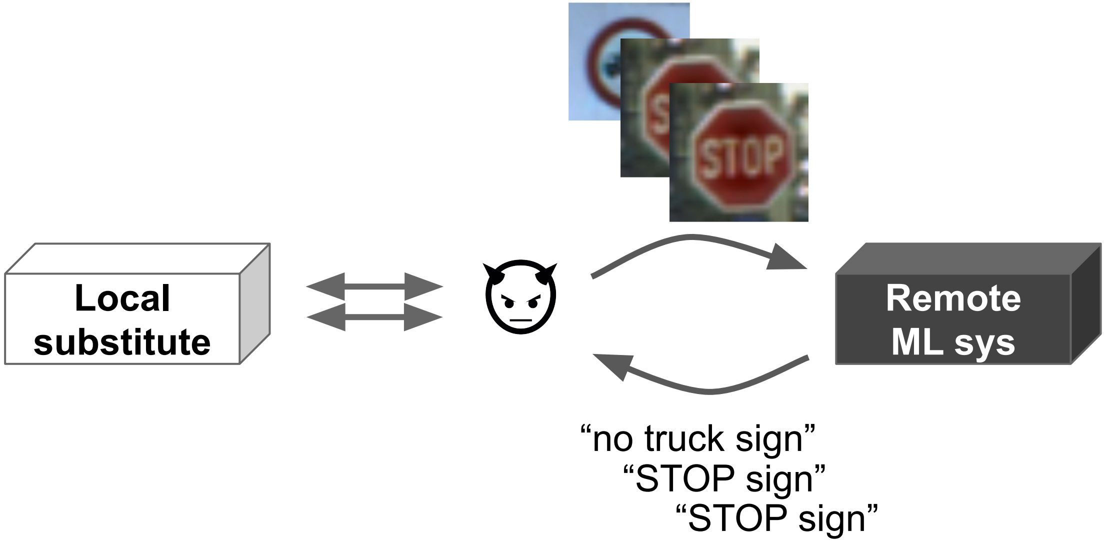
Alleviate lack of training data

Attacking remotely hosted black-box models



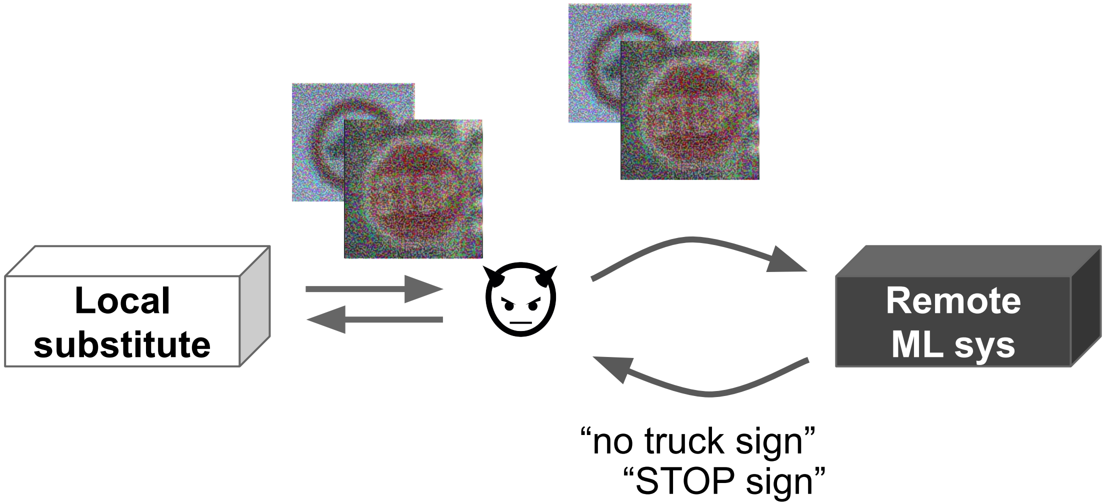
(1) The adversary queries remote ML system for labels on inputs of its choice.

Attacking remotely hosted black-box models



(2) The adversary uses this labeled data to train a local substitute for the remote system.

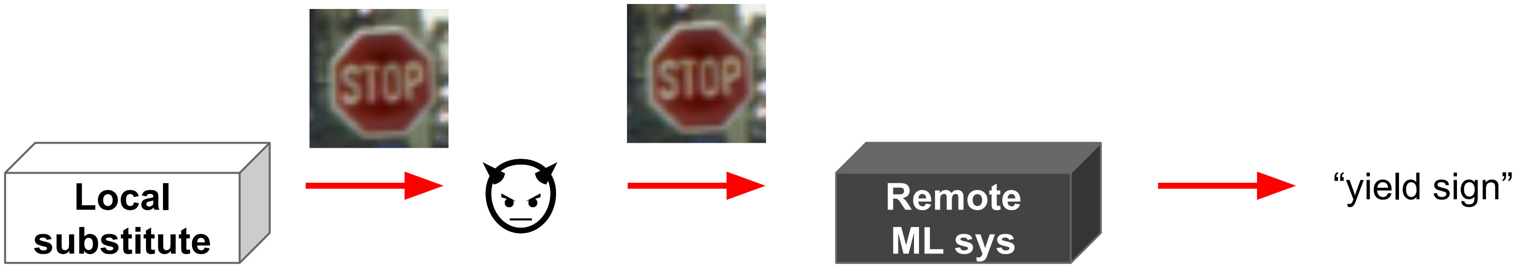
Attacking remotely hosted black-box models



$$S_{\rho+1} = \{\vec{x} + \lambda_{\rho+1} \cdot \mathrm{sgn}(J_F[\tilde{O}(\vec{x})]) : \vec{x} \in S_\rho\} \cup S_\rho$$

(3) The adversary selects new synthetic inputs for queries to the remote ML system based on the local substitute’s output surface sensitivity to input variations.

Attacking remotely hosted black-box models



(4) The adversary then uses the local substitute to craft adversarial examples, which are misclassified by the remote ML system because of transferability.

Our approach to black-box attacks

Alleviate lack of knowledge about model

↓

Adversarial example transferability from a substitute model to target model

+

Alleviate lack of training data

↓

Synthetic data generation

Results on real-world remote systems

Remote Platform	ML technique	Number of queries	Adversarial examples misclassified (after querying)
MetaMind	Deep Learning	6,400	84.24%
amazon web services™	Logistic Regression	800	96.19%
Google Cloud Platform	Unknown	2,000	97.72%

All remote classifiers are trained on the MNIST dataset (10 classes, 60,000 training samples)

[PMG16a] Papernot et al. Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples

Is S&P ***of*** ML any different from ~~real-world~~ computer security?

Security & privacy are hard. Faster CPUs or the Internet did not make them easier.

“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)

ML offers new forms of data analysis just like SQL tables a few years ago.

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

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.

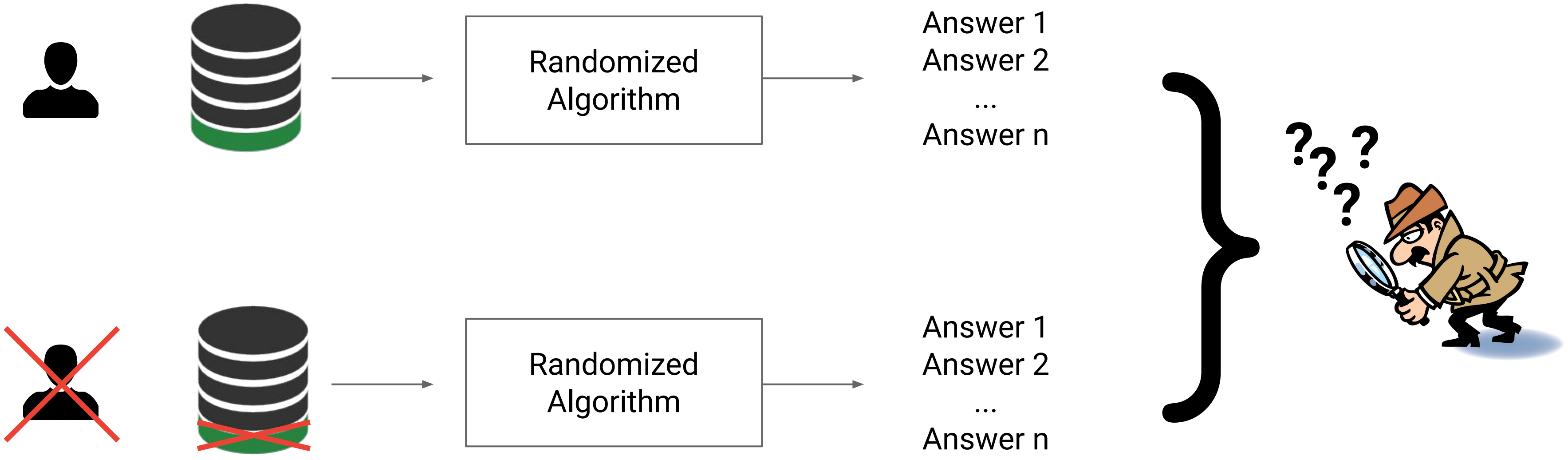
Compromise recording.
Mechanisms that reliably record compromises can be used in place of mechanisms that prevent loss.

* The Protection of Information in Computer Systems (1975)

Privacy-preserving machine learning

Designing algorithms with privacy guarantees understood by humans is difficult.

First question: how should we define privacy? Gold standard is now differential privacy.



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

IACR:3650 (Dwork et al.)

Differentially Private Stochastic Gradient Descent

Obtain privacy by:

1. Control sensitivity of parameter updates
2. Noise parameter updates before they are applied

```
Initialize parameters θ

For t = 1..T do
    Pick a random set B of examples
    Compute gradient of loss for each of the examples
    Ensure norm of gradients < C by clipping
    Add Gaussian noise to gradients (as a function of C)
    Average noisy gradients
    Update parameters by a multiple of this average
```

Deep Learning with Differential Privacy (CCS, 2016)
Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, Zhang

Google

TensorFlow Privacy

```
optimizer = tf.train.GradientDescentOptimizer(
        learning_rate=FLAGS.learning_rate)
```

↓

```
optimizer = DPGradientDescentGaussianOptimizer(
        l2_norm_clip=FLAGS.l2_norm_clip,
        noise_multiplier=FLAGS.noise_multiplier,
        learning_rate=FLAGS.learning_rate)
```

Sundar Pichai
@sundarpichai

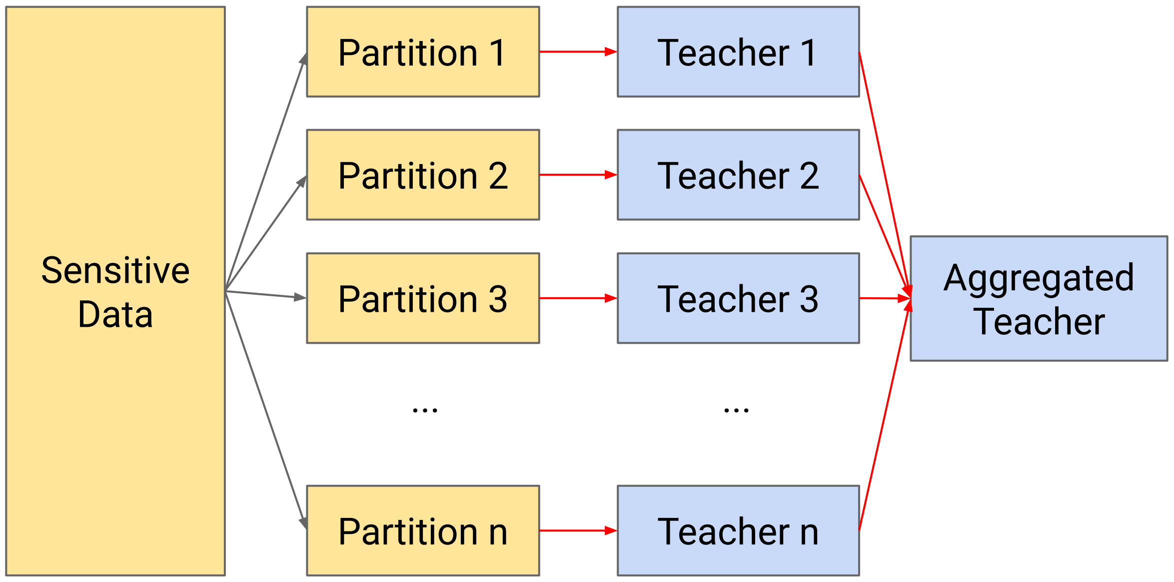
TensorFlow Privacy, using techniques based on differential privacy, will make it easier for developers to train ML models with privacy and better protect users' data in their AI development. #GoogleAI

Google is making it easier for AI developers to keep users' data private
More privacy with just a few lines of extra code.
theverge.com

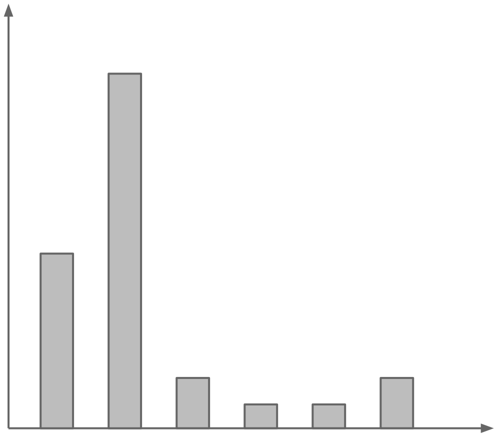
12:09 PM · Mar 6, 2019 · Twitter Web Client

299 Retweets **1.8K** Likes

PATE: Private Aggregation of Teacher Ensembles

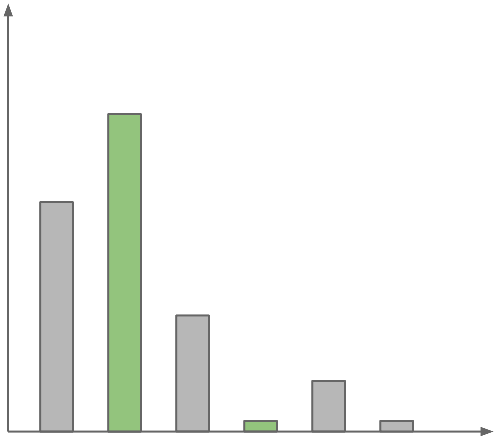


PATE: Private Aggregation of Teacher Ensembles



Count votes

$$n_j(\vec{x}) = |\{i : i \in 1..n, f_i(\vec{x}) = j\}|$$

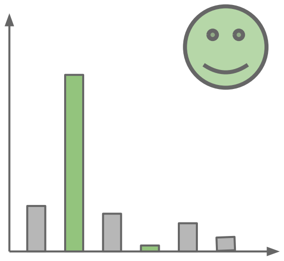


Take maximum

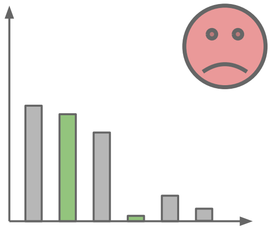
$$f(x) = \arg\max_j \left\{ n_j(\vec{x}) \right\}$$

PATE: Private Aggregation of Teacher Ensembles

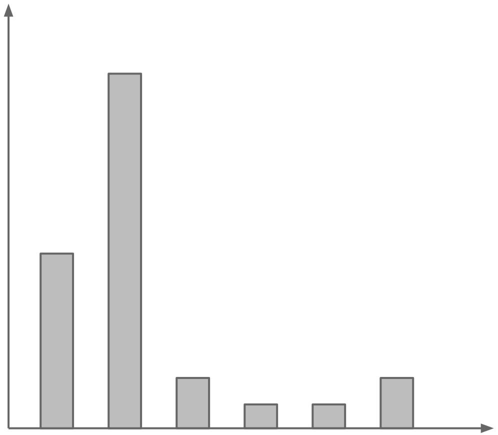
If most teachers agree on the label,
it does not depend on specific partitions,
so the privacy cost is small.



If two classes have close vote counts,
the disagreement may reveal private information.

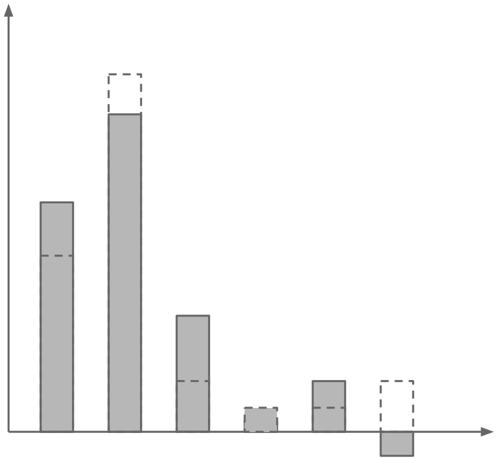


PATE: Private Aggregation of Teacher Ensembles



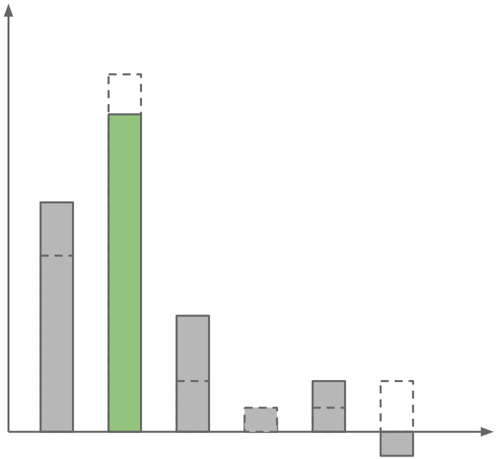
Count votes

$$n_j(\vec{x}) = |\{i : i \in 1..n, f_i(\vec{x}) = j\}|$$



Add Laplacian noise

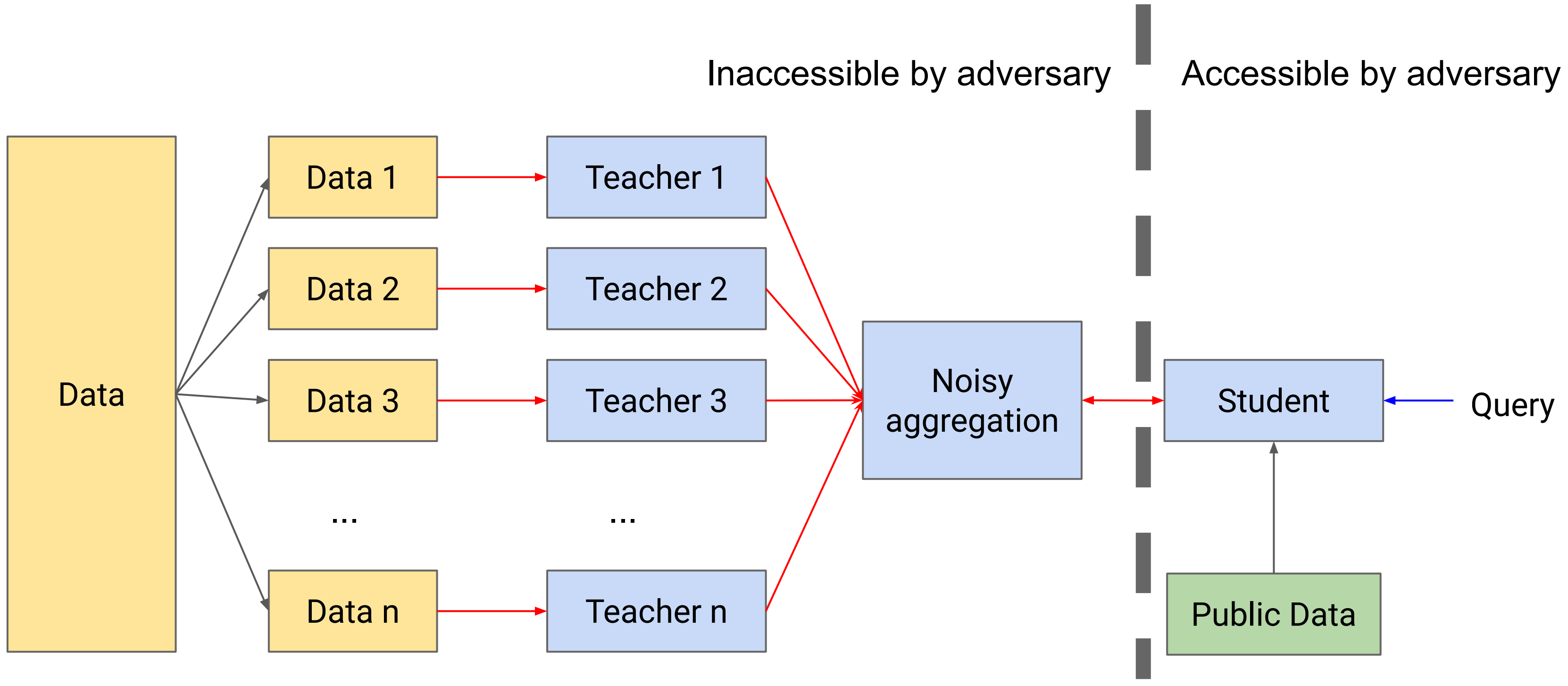
$$Lap\left(\frac{1}{\varepsilon}\right)$$

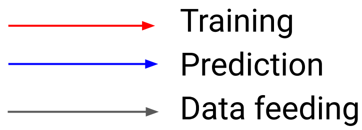


Take maximum

$$f(x) = \arg\max_j \left\{ n_j(\vec{x}) + Lap\left(\frac{1}{\varepsilon}\right) \right\}$$

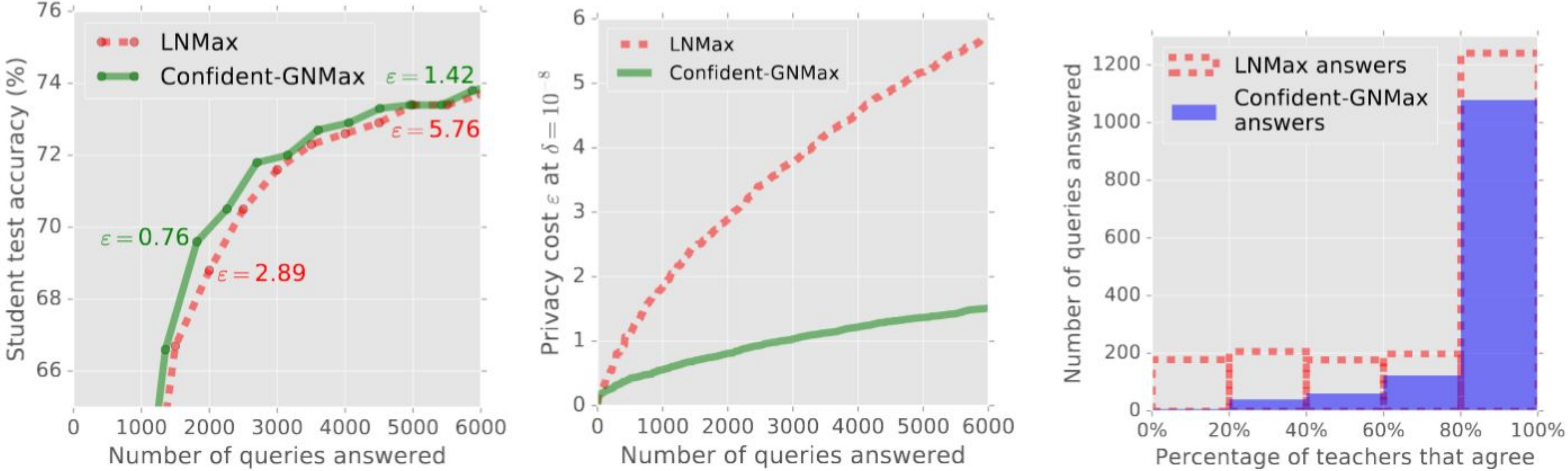
PATE: Private Aggregation of Teacher Ensembles





PATE: Private Aggregation of Teacher Ensembles (ICLR 2017)
Papernot, Abadi, Erlingsson, Goodfellow, Talwar

Aligning privacy with generalization

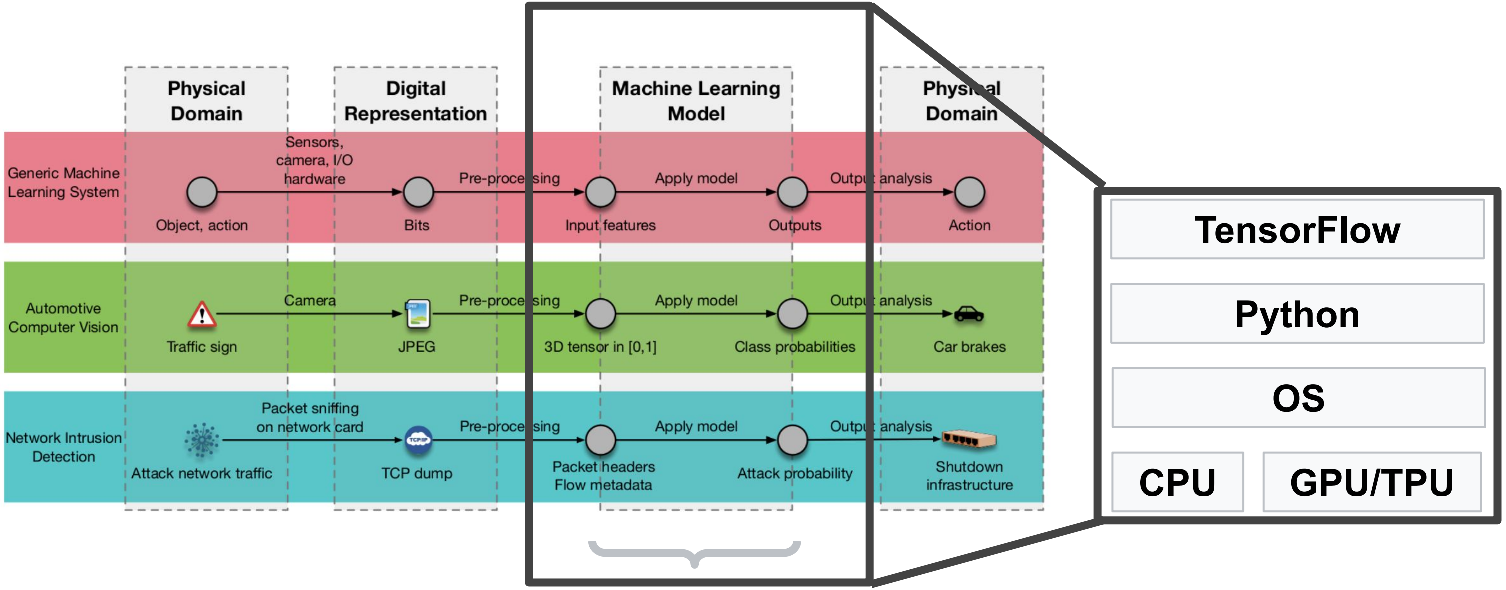


Scalable Private Learning with PATE (Papernot*, Song* et al., ICLR 2018)

Moving forward

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?



Google

Towards the Science of Security and Privacy in Machine Learning (Papernot et al.)

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?

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?

Blog:
cleverhans.io

Open-source libraries
github.com/tensorflow/cleverhans
github.com/tensorflow/privacy

A Marauder's Map of Security and Privacy in Machine Learning
https://arxiv.org/abs/1811.01134

Email: nicolas@papernot.fr
Twitter: @NicolasPapernot

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

Charles Goodhart