Systems and Internet Infrastructure Security Laboratory

PennState

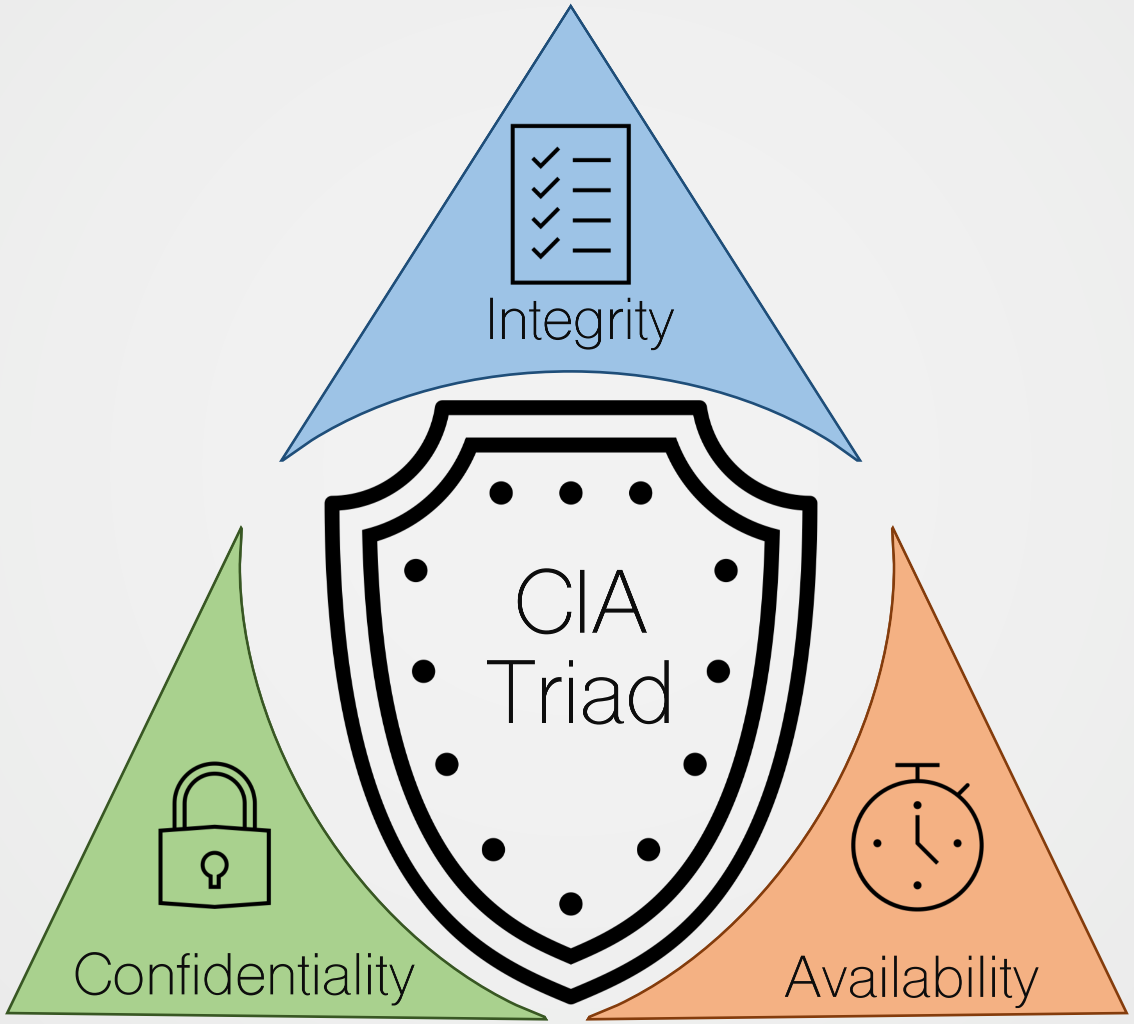
Adversarial Machine Learning

CMPSC 443: Introduction to Computer Security

Ryan Sheatsley

Tuesday, October 26th 2022

Information Security



Overview

1. Machine Learning
2. Integrity
3. Confidentiality
4. Availability

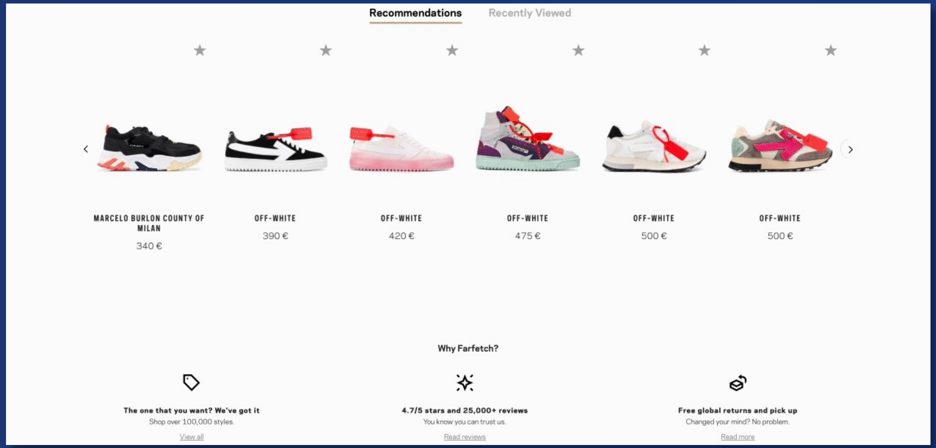
Overview

1. Machine Learning
2.
3.
4. Availability

Goal: View machine learning through the lens of a security specialist.

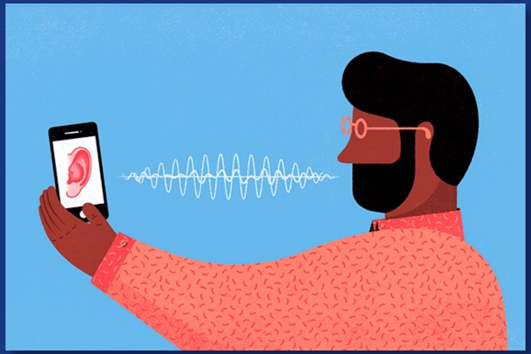
What is “Machine Learning?”

Product Recommendation



https://www.farfetchtechblog.com/en/blog/post/how-to-build-a-recommender-system-it-s-all-about-rocket-science-part-1/

Voice Assistants



https://www.geico.com/living/home/technology/voice-assistant/

Autonomous Driving

https://medium.datadriveninvestor.com/goal-setting-lessons-from-reinforcement-learning-d0c58b321391

Can machine learning make mistakes?

Can machine learning make mistakes?

Product Recommendation

https://twitter.com/whoschaos/status/939999586998943744?lang=en

Voice Assistants

Alexa tells 10-year-old girl to touch live plug with penny

28 December 2021

GETTY IMAGES

Amazon has updated its Alexa voice assistant after it "challenged" a 10-year-old girl to touch a coin to the prongs of a half-inserted plug.

The suggestion came after the girl asked Alexa for a "challenge to do".

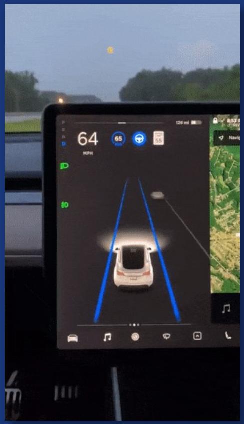
"Plug in a phone charger about halfway into a wall outlet, then touch a penny to the exposed prongs," the smart speaker said.

Amazon said it fixed the error as soon as the company became aware of it.

The girl's mother, Kristin Livdahl, described the incident on Twitter.

https://www.bbc.com/news/technology-59810383

Autonomous Driving



https://twitter.com/jordanteslatech/status/1418413307862585344?lang=en

Can we **force** *machine learning to make mistakes?*

Can we *force* machine learning to make mistakes?

Can we **force** machine learning to make mistakes?

"How did this happen?"

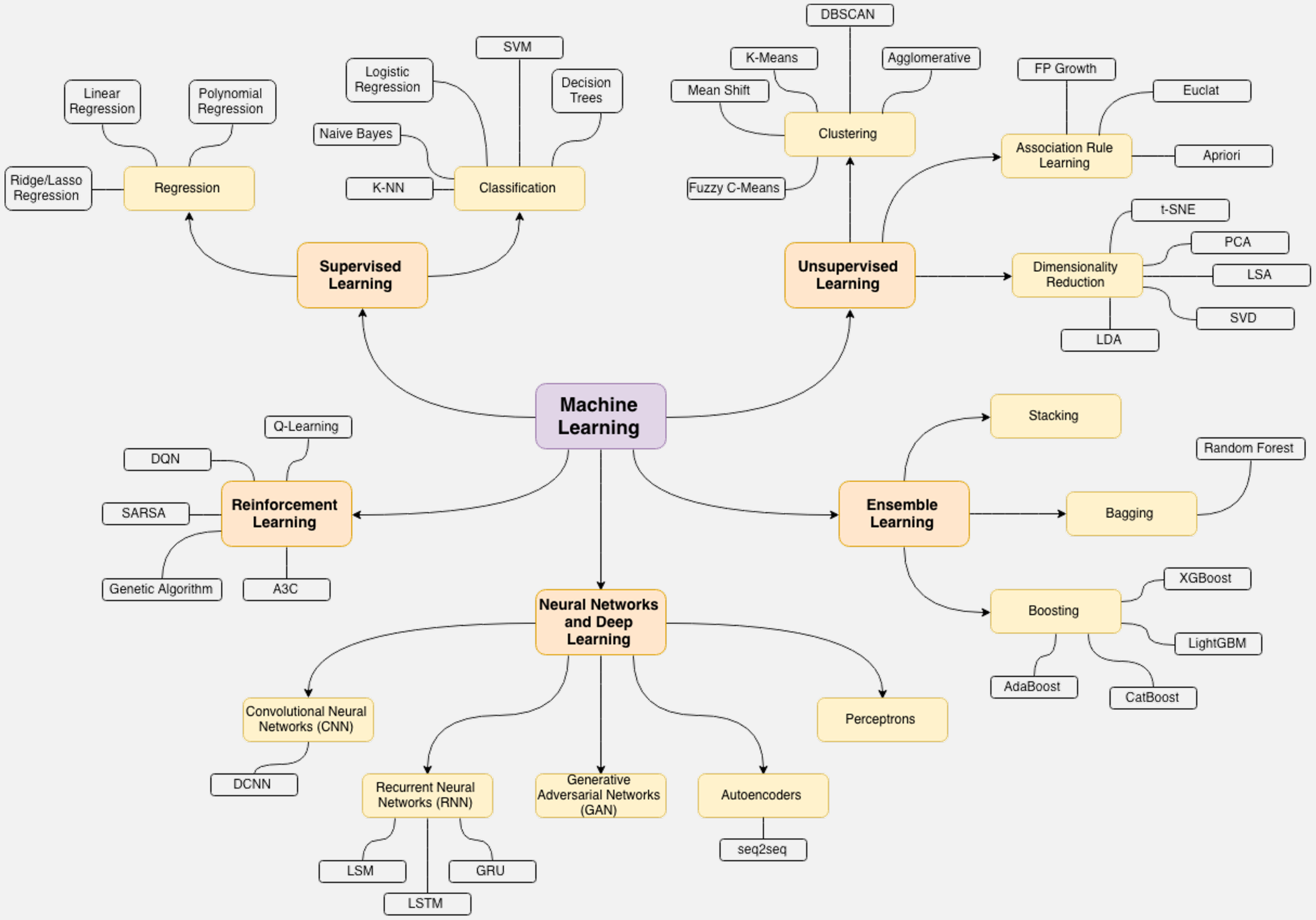
What (really) is "Machine Learning?"

What (really) is "Machine Learning?"

"The field of study that gives computers the ability to learn without explicitly being programmed."

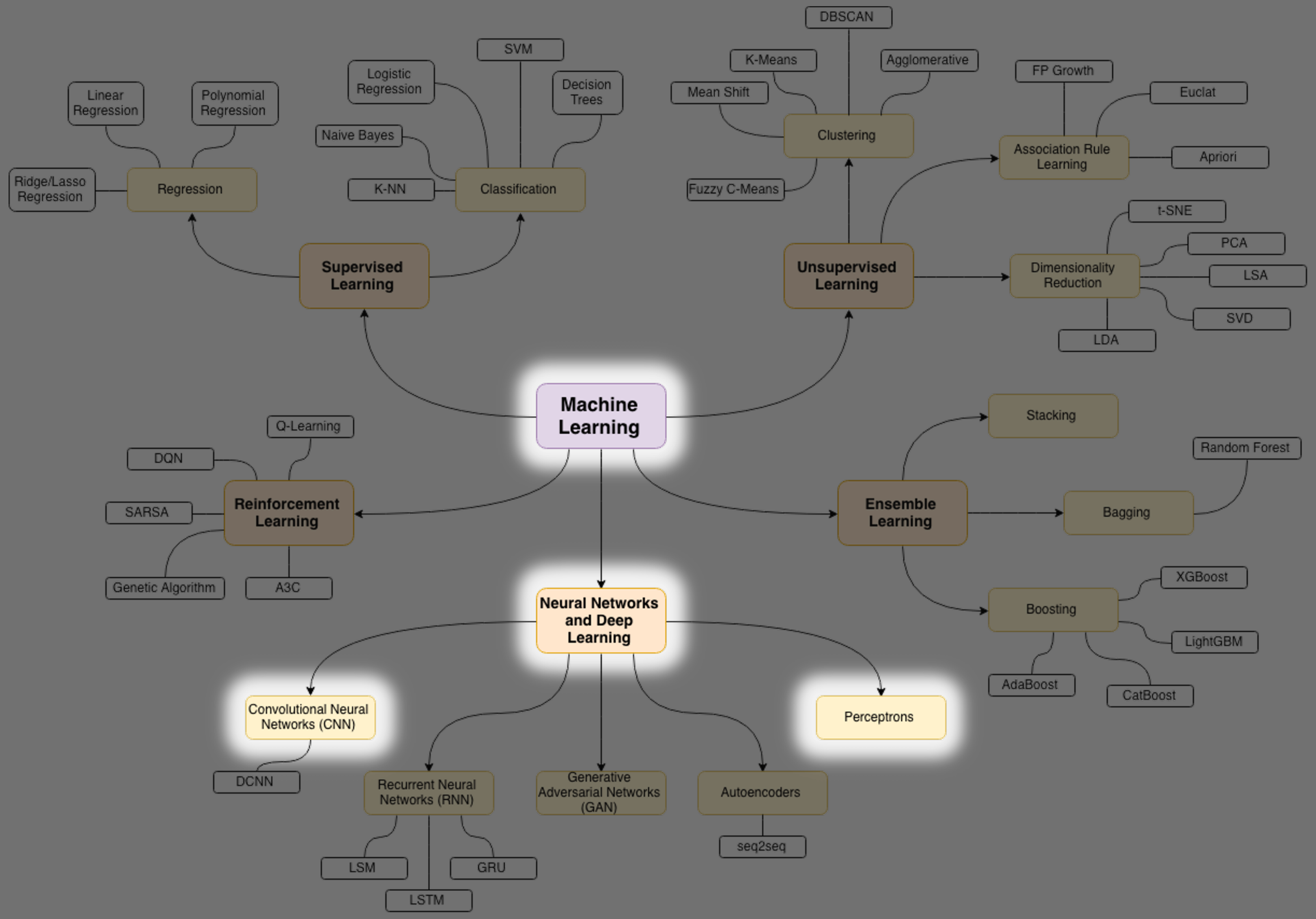
In 1962, Samuel's Checkers program defeats self-proclaimed checkers master, Robert Nealey, played on an IBM 7094 computer.

Source: https://www.ibm.com/ibm/history/ibm100/us/en/icons/ibm700series/impacts/



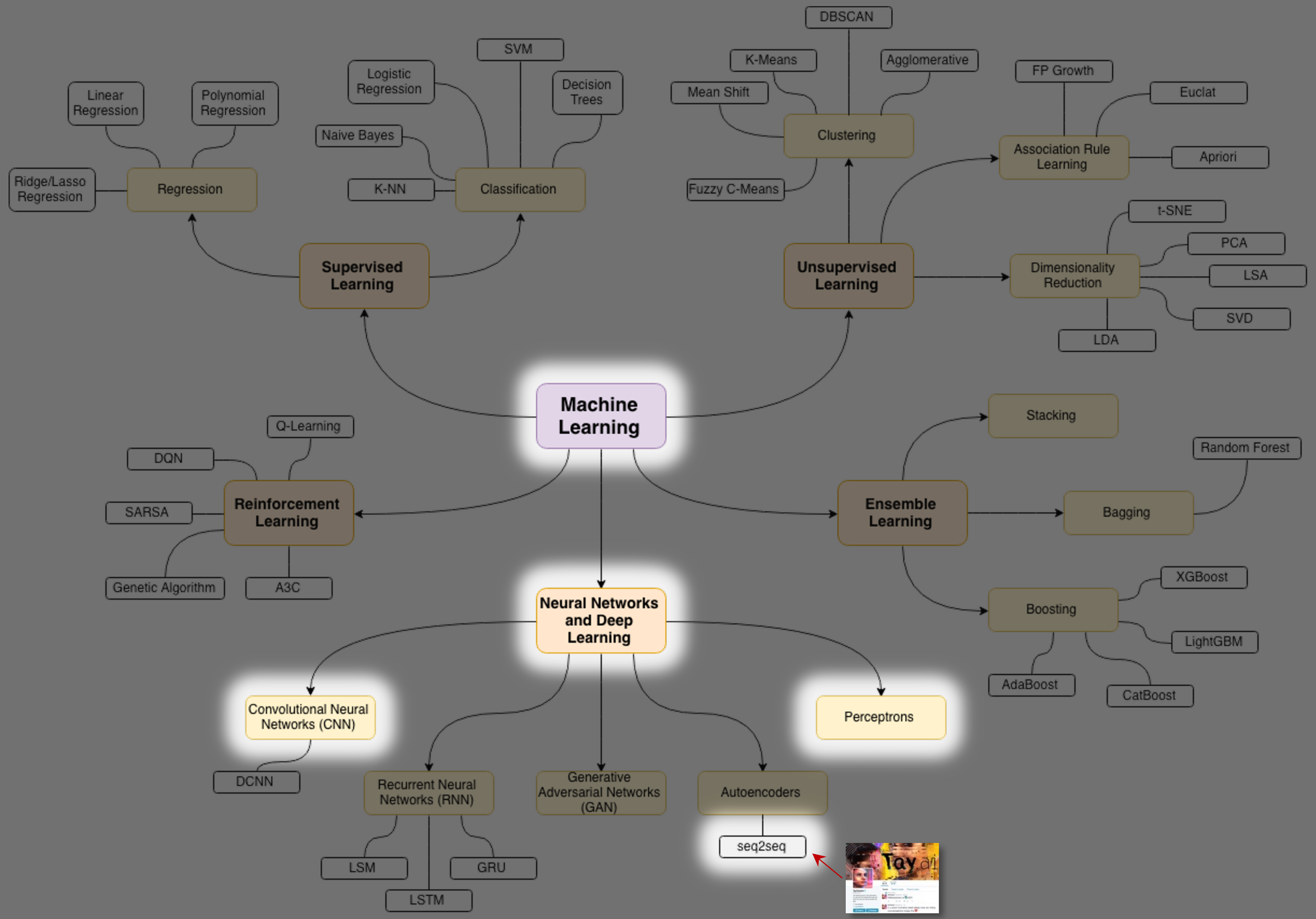
Machine Learning

Source: *https://github.com/trekhleb/homemade-machine-learning*



Source: *https://github.com/trekhleb/homemade-machine-learning*

Machine Learning

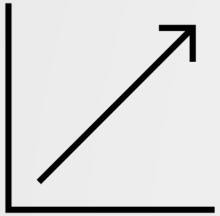


Machine Learning

Source: *https://github.com/trekhleb/homemade-machine-learning*

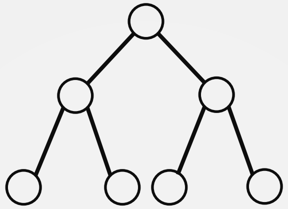
Machine Learning

- It begins with an *assumption…*



Maybe it's a line…

(Linear Regression)



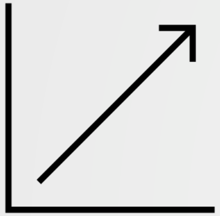
… or a collection of if-then-else rules…

(Decision Trees)

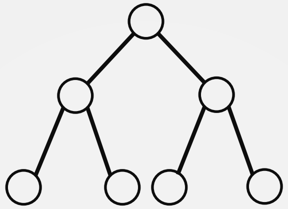
… or maybe you don't know…

Machine Learning

- It begins with an *assumption*…



Maybe it's a line…
(Linear Regression)



… or a collection of if-then-else rules…
(Decision Trees)

… or maybe you don't know…

- … and some *data*…

"badger"

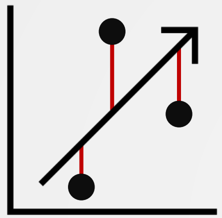
"mushroom"

"snake"

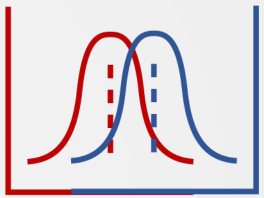
Having these allows us to use *supervised* learning algorithms (which Tay was likely using), otherwise, we use *unsupervised* approaches

Machine Learning

- … a measurement of *error*…



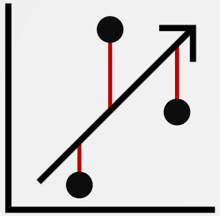
Maybe it's the distance from a line…
(Mean Squared Error)



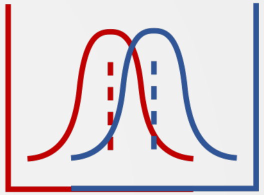
… or the difference between two distributions.
(Cross-Entropy)

Machine Learning

- … a measurement of *error*…



Maybe it's the
distance from a line…
(Mean Squared Error)



… or the difference
between two distributions.
(Cross-Entropy)

- … and way to *minimize* it.

i.e., through hill climbing…
(Gradient Descent)

… or minimizing disorder.
(Information Gain)

Machine Learning

- ... a measurement of *error*...

ns.

There are *many, many* ways to deploy machine learning models.

- ... and way to *minimize* it.

i.e., through hill climbing...

(Gradient Descent)

... or minimizing disorder.

(Information Gain)

Putting it all together

$$f_\theta$$

Model

Putting it all together

Data

$$f_\theta(x)$$

Model

Putting it all together

Error $\qquad$ Data

$$L(f_\theta(x), y)$$

Model $\qquad$ Labels

Putting it all together

$$\underset{\theta}{\min}\, \underbrace{L}_{}\!\left(f_\theta(x), y\right)$$

Error

Data

$$\min_{\theta} L(f_{\theta}(x), y)$$

Optimizer

Model

Labels

Overview

1. ~~Machine Learning~~
2. Integrity
3. Confidentiality
4. Availability

Putting it all together

$$\underbrace{\min_{\theta}}_{\text{Optimizer}} \overbrace{L}^{\text{Error}}(\underbrace{f_\theta}_{\text{Model}}(\overbrace{x}^{\text{Data}}), \underbrace{y}_{\text{Labels}})$$

Putting it all together

Error Data

What can go wrong?

Optimizer Model Labels

Putting it all together

Data (often *the* most valuable resource) is assumed collected faithfully...

Error

Data

$$\min_{\theta} L(f_\theta(x), y)$$

Optimizer

Model

Labels

Putting it all together

Error

Data

$$\min_{\theta} L(f_{\theta}(x), y)$$

Optimizer

Model

Labels

Data (often *the* most valuable resource) is assumed collected faithfully…

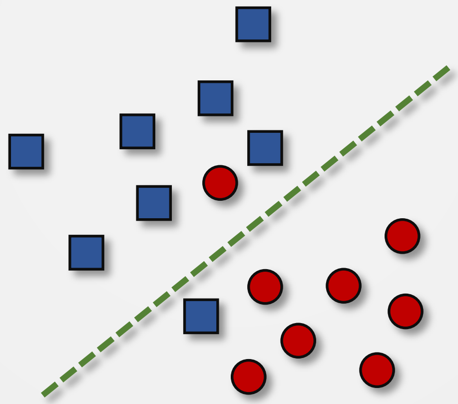
… and the corresponding labels are assumed to be accurately described.

Suppose not: Integrity Attacks, Pt. I

Q: What if an adversary *controls* (some portion) of your data?

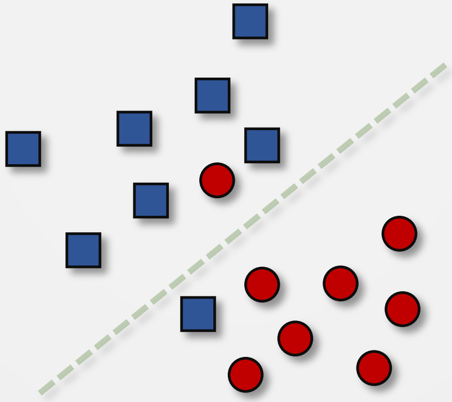
Suppose not: Integrity Attacks, Pt. I

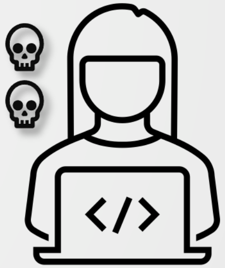
Q: What if an adversary *controls* (some portion) of your data?



Suppose not: Integrity Attacks, Pt. I

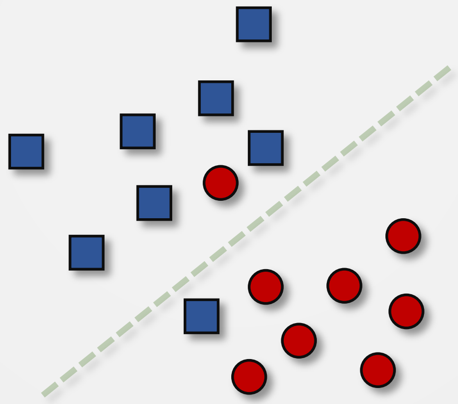
Q: What if an adversary *controls*
(some portion) of your data?





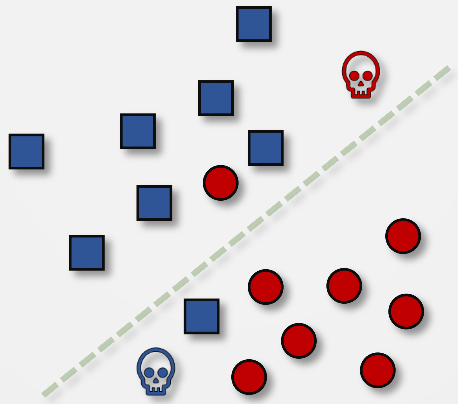
Suppose not: Integrity Attacks, Pt. I

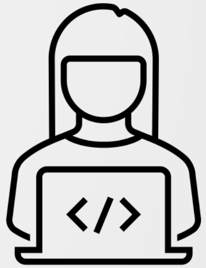
Q: What if an adversary *controls* (some portion) of your data?



Suppose not: Integrity Attacks, Pt. I

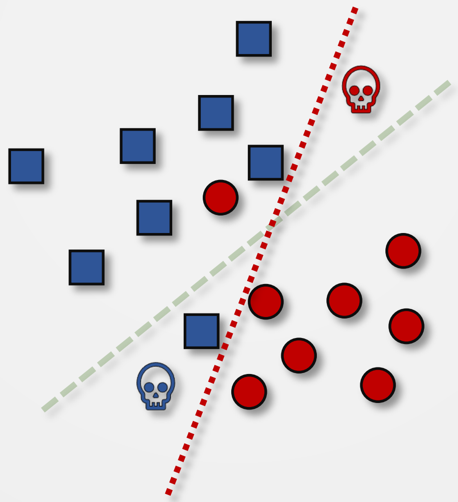
Q: What if an adversary *controls*
(some portion) of your data?

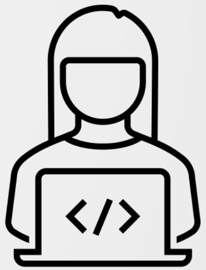




Suppose not: Integrity Attacks, Pt. I

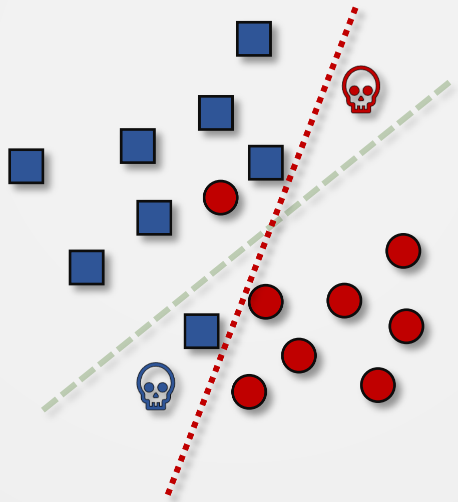
Q: What if an adversary *controls* (some portion) of your data?

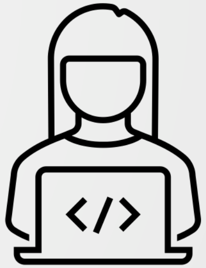




Suppose not: Integrity Attacks, Pt. I

Q: What if an adversary *controls* (some portion) of your data?



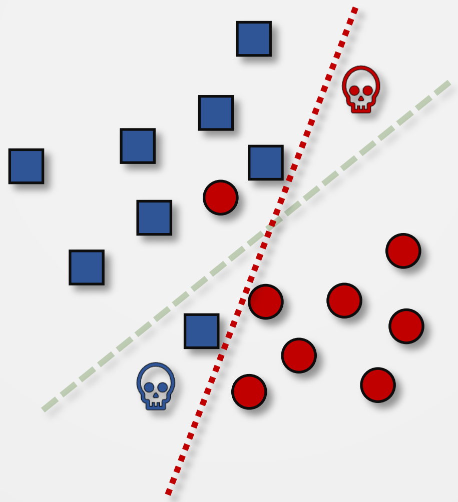


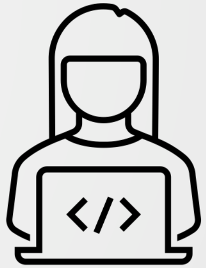
A: They can influence the decision boundary.

Suppose not: Integrity Attacks, Pt. I

Q: What if an adversary *controls* (some portion) of your data?

Under this threat model:



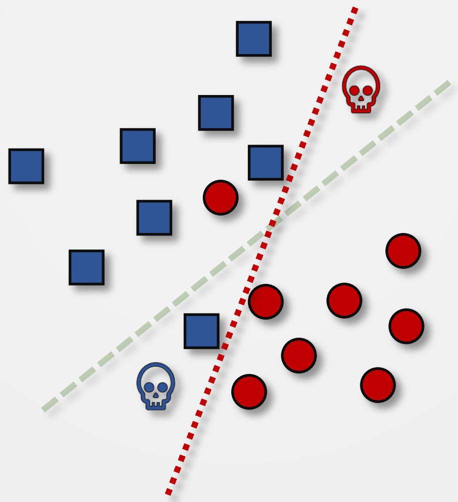


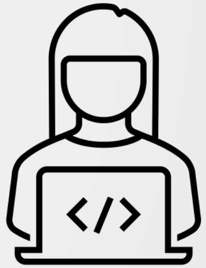
A: They can influence the decision boundary.

Suppose not: Integrity Attacks, Pt. I

Q: What if an adversary *controls* (some portion) of your data?

Under this threat model:
- *Threat:* An adversary who can add (data, label) pairs





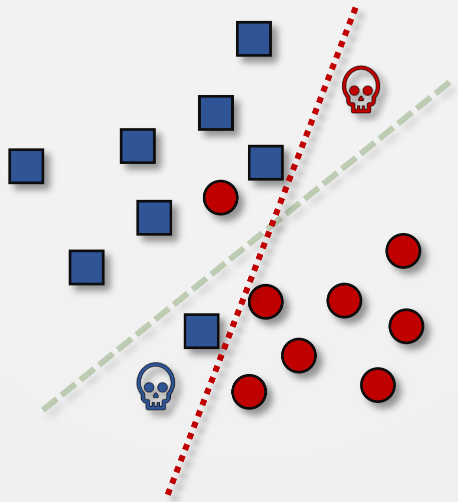
A: They can influence the decision boundary.

Suppose not: Integrity Attacks, Pt. I

Q: What if an adversary *controls* (some portion) of your data?

Under this threat model:

- *Threat*: An adversary who can add (data, label) pairs
- *Vulnerability*: Decision boundary can be manipulated



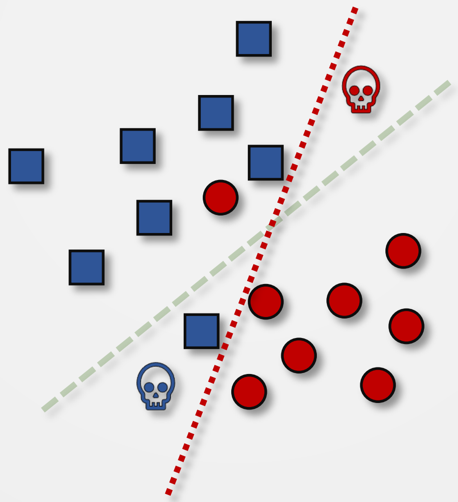
A: They can influence the decision boundary.

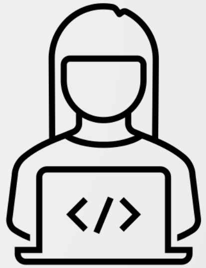
Suppose not: Integrity Attacks, Pt. I

Q: What if an adversary *controls* (some portion) of your data?

Under this threat model:

- *Threat*: An adversary who can add (data, label) pairs
- *Vulnerability*: Decision boundary can be manipulated
- *Exploit*: ?



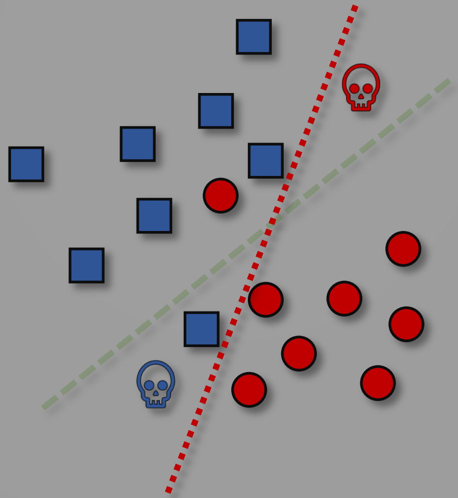


A: They can influence the decision boundary.

Suppose not: Integrity Attacks, Pt. I

Q: What if an adversary *controls*
(e portion) of your data?

Ur



A bot that only tweets profanities

A system that has backdoors

A: They can influence the decision boundary.

Suppose not: Integrity Attacks, Pt. I

Error

Data

$$\underbrace{\min_{\theta}}_{\text{Optimizer}} \overbrace{L}^{\text{Error}}\big(\underbrace{f_\theta}_{\text{Model}}(\overbrace{x}^{\text{Data}}), \underbrace{y}_{\text{Labels}}\big)$$

Optimizer

Model

Labels

Suppose not: Integrity Attacks, Pt. I

Error Data

$$\min_{\theta} L(f_{\theta}(x), y)$$

Optimizer Model Labels

Suppose not: Integrity Attacks, Pt. I

Error

Data

$$\min_{\theta} L(f_{\theta}(x), y)$$

Optimizer

Model

Labels

Suppose not: Integrity Attacks, Pt. I

Error Data

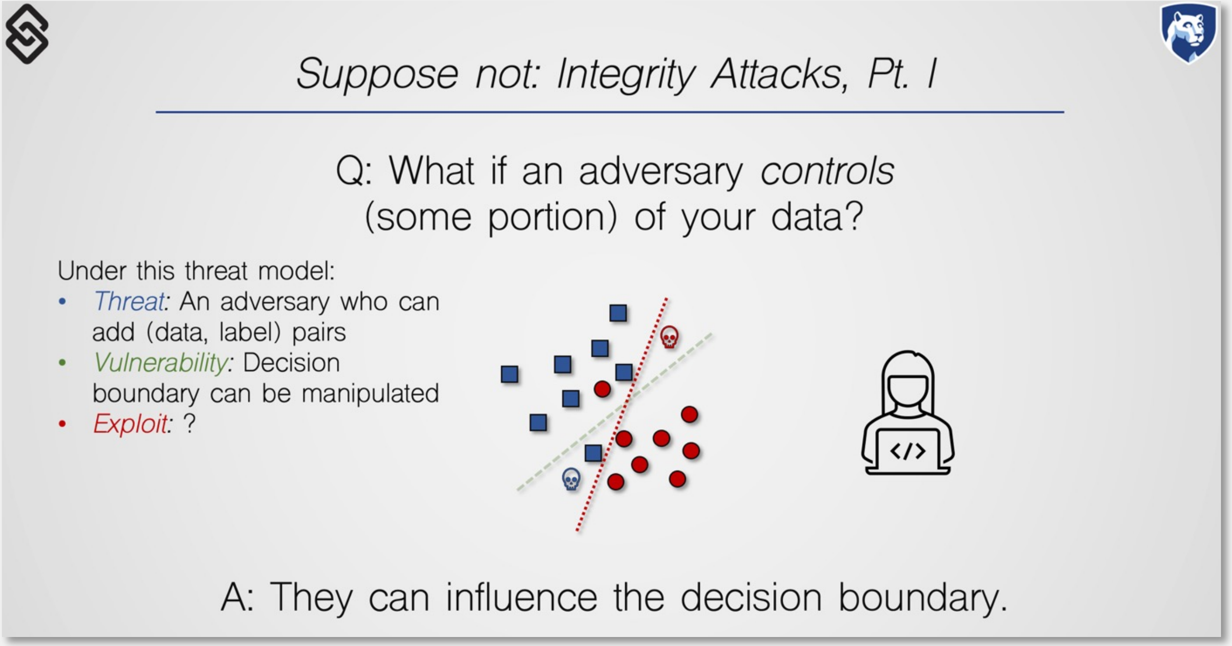
$$\min_{\theta} L(f_{\theta}(x), y)$$

Optimizer Model Labels

This is known as a *poisoning* attack

Suppose not: Integrity Attacks, Pt. II

Q: Okay sure, but what if they don't have control over the training data?



Suppose not: Integrity Attacks, Pt. II

Q: Okay sure, but what if they don't have control over the training data?

$$\min_{\theta} \underbrace{L}_{\text{Error}}(\underbrace{f_{\theta}}_{\text{Model}}(\underbrace{x}_{\text{Data}}), \underbrace{y}_{\text{Labels}})$$

Error — Data — Optimizer — Model — Labels

Suppose not: Integrity Attacks, Pt. II

Q: Okay sure, but what if they don't have control over the training data?

$$\min_{\theta} \overbrace{L}^{\text{Error}}(\underbrace{f_{\theta}}_{\text{Model}}(\overbrace{x}^{\text{Data}}), \underbrace{y}_{\text{Labels}})$$

Optimizer

Suppose not: Integrity Attacks, Pt. II

Q: Okay sure, but what if they don't have control over the training data?

Machine learning systems often follow a two-stage lifecycle: *training* and *inference*

Suppose not: Integrity Attacks, Pt. II

“Deployment”

Training

$$\min_{\theta} L(f_{\theta}(x), y)$$

Inference

$$f_{\theta}(x)$$

Suppose not: Integrity Attacks, Pt. II

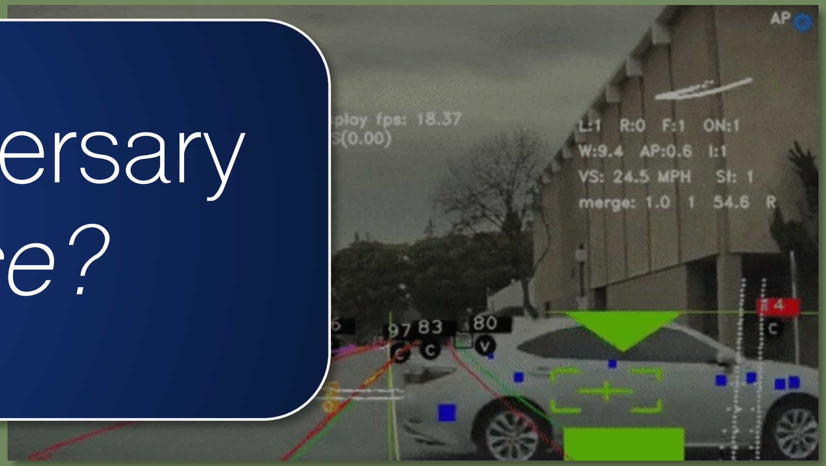
"Deployment"

Training

$$\min_{\theta} L(f_\theta(x), y)$$

Inference

$$f_\theta(x)$$



What can an adversary do *at inference*?

Suppose not: Integrity Attacks, Pt. II

An observation....

Suppose not: Integrity Attacks, Pt. II

$$\underset{\theta}{\min} \, \overset{\text{Error}}{\overset{\downarrow}{L}}(\underset{\substack{\uparrow \\ \text{Model}}}{f_\theta}(\overset{\text{Data}}{\overset{\downarrow}{x}}), \underset{\substack{\uparrow \\ \text{Labels}}}{y})$$

Optimizer: θ

Suppose not: Integrity Attacks, Pt. II

$$\min_{\substack{\cancel{\theta}\to x \\ \text{Optimizer}}} \overset{\text{Error}}{L}\Big(\underset{\text{Model}}{f_{\theta}}(\overset{\text{Data}}{x}),\ \underset{\text{Labels}}{y}_{\text{airplane}}\Big)$$

Suppose not: Integrity Attacks, Pt. II

Adversarial Examples

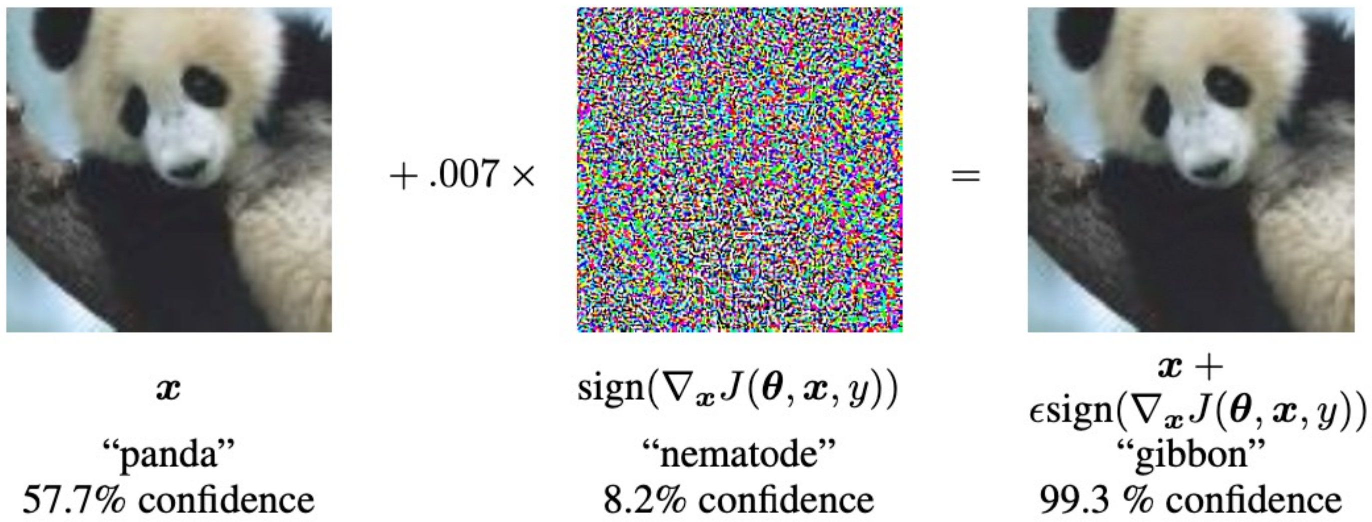


Figure 1: A demonstration of fast adversarial example generation applied to GoogLeNet (Szegedy et al., 2014a) on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet's classification of the image. Here our ϵ of .007 corresponds to the magnitude of the smallest bit of an 8 bit image encoding after GoogLeNet's conversion to real numbers.

Suppose not: Integrity Attacks, Pt. II

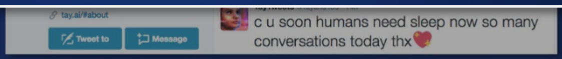
Can we **force** machine learning to make mistakes?

"How did this happen?"

Suppose not: Integrity Attacks, Pt. II

*Can we **force** machine learning to make mistakes?*

Adversarial examples are inputs *designed* to induce worst-case behavior



"How did this happen?"

Suspose not: Integrity Attacks, Pt. II

Adversarial Examples, formalized

$$\arg\min_{\epsilon} \|\epsilon\|_p$$

$$\text{such that: } f_\theta(x+\epsilon) \neq y$$
$$x+\epsilon \in B_r^p$$

Suppose not: Integrity Attacks, Pt. II

Adversarial Examples, formalized

Find me the smallest change...

$$\arg\min_{\epsilon} \|\epsilon\|_p$$

$$\text{such that: } f_\theta(x+\epsilon) \neq y$$
$$x + \epsilon \in B_r^p$$

Suppose not: Integrity Attacks, Pt. II

Adversarial Examples, formalized

Find me the smallest change...

$$\arg\min_{\epsilon} \|\epsilon\|_p$$

... that is misclassified by my model....

$$\text{such that: } f_\theta(x+\epsilon) \neq y$$

$$x + \epsilon \in B_r^p$$

Suppose not: Integrity Attacks, Pt. II

Adversarial Examples, formalized

Find me the smallest change…

$$\arg\min_{\epsilon} \|\epsilon\|_p$$

… that is misclassified by my model….

$$\text{such that: } f_\theta(x+\epsilon) \neq y$$

$$x + \epsilon \in B_r^p$$

… yet still close to the original sample.

Suppose not: Integrity Attacks, Pt. II

Adversarial Examples, formalized

Find me the smallest change...

$$\arg\min \|\epsilon\|_p$$

... that is misclassified by my model....

$$\text{such that: } f_\theta(x+\epsilon) \neq y$$

$$x + \epsilon \in B_r^p$$

... yet still close to the original sample

Auto Projected Gradient Descent

Adversarial Patch

Elastic Net

DeepFool

Shadow

Carlini-Wagner

Square

Feature Adversaries

Projected Gradient Descent

Wasserstein

ShapeShifter

Fast-Gradient Sign Method

Brendel & Bethge

NetwonFool

Virtual Adversarial Method

Basic Iterative Method

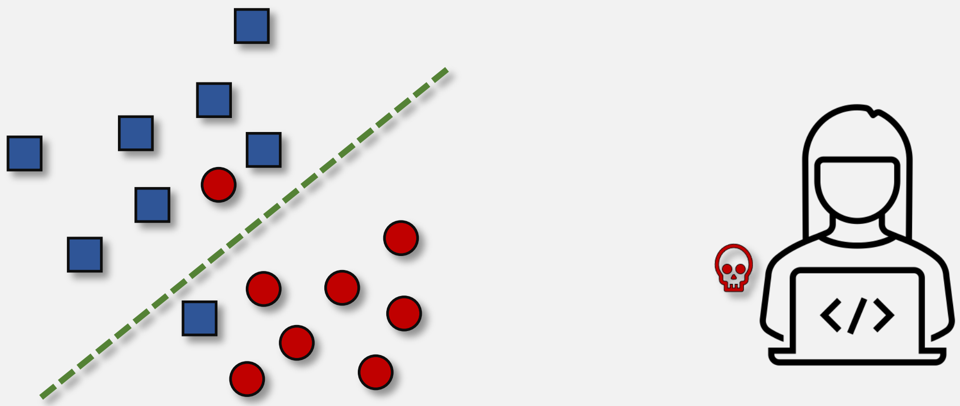
Fast Adaptive Boundary

Jacobian-based Saliency Map Approach

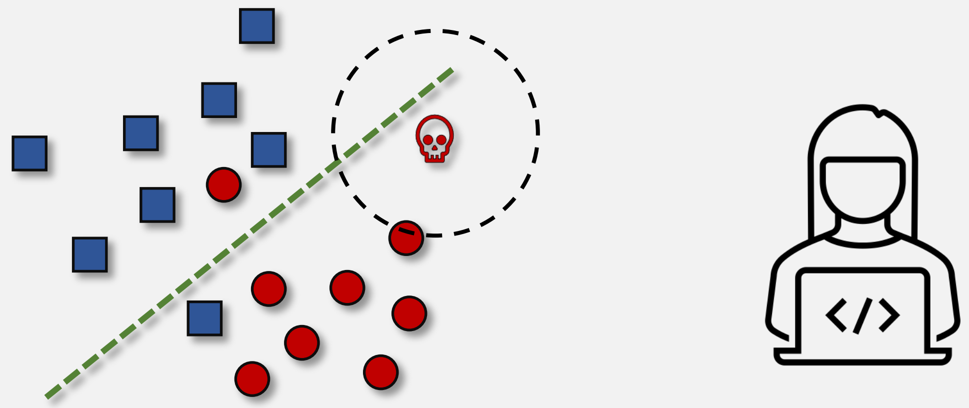
Iterative Frame Saliency

Universal Perturbation

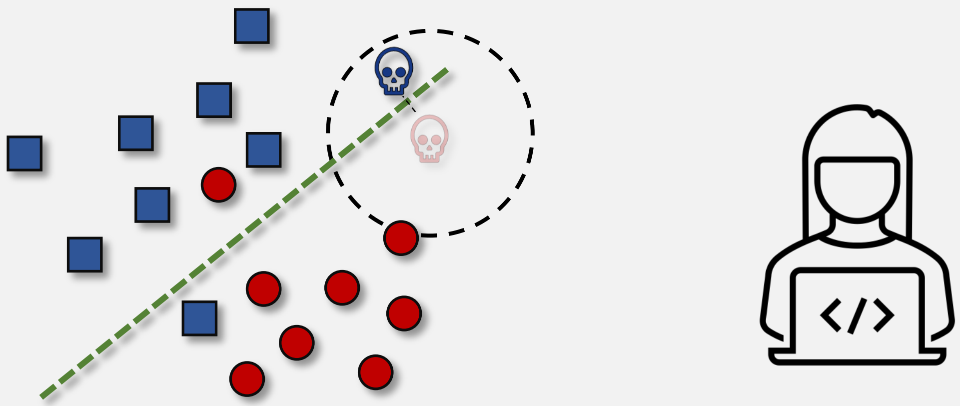
Suppose not: Integrity Attacks, Pt. II



Suppose not: Integrity Attacks, Pt. II



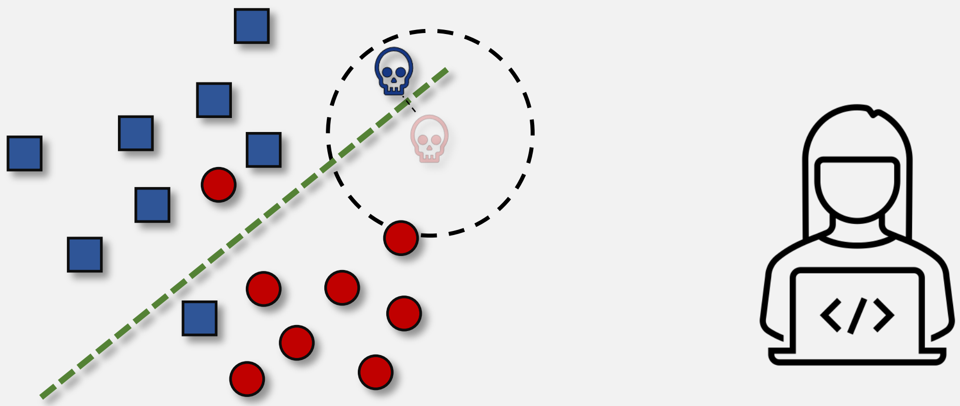
Suppose not: Integrity Attacks, Pt. II



This is known as an *evasion* attack

Suppose not: Integrity Attacks, Pt. II

Under this threat model:

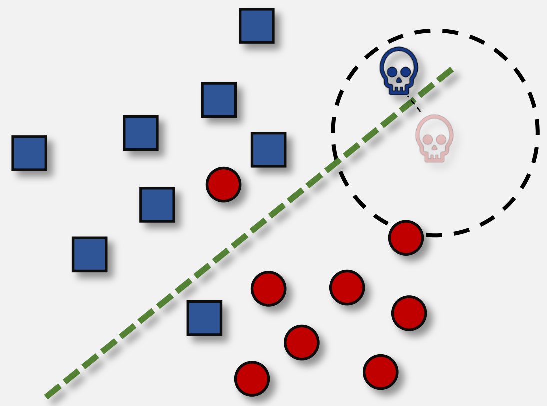


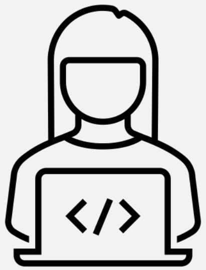
This is known as an *evasion* attack

Suppose not: Integrity Attacks, Pt. II

Under this threat model:

- *Threat*: An adversary can use model information to estimate input sensitivity



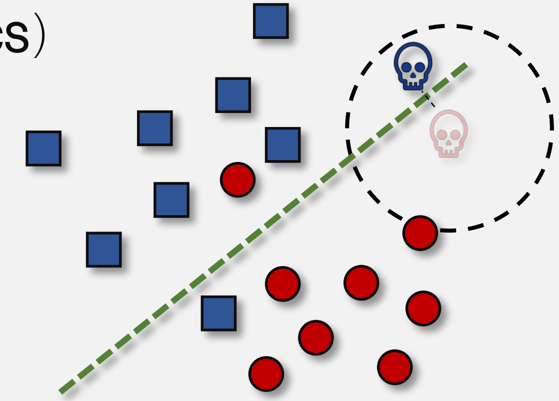


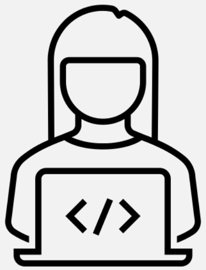
This is known as an *evasion* attack

Suppose not: Integrity Attacks, Pt. II

Under this threat model:

- *Threat*: An adversary can use model information to slightly manipulate inputs
- *Vulnerability*: Inputs can be misclassified (**while preserving underlying semantics**)



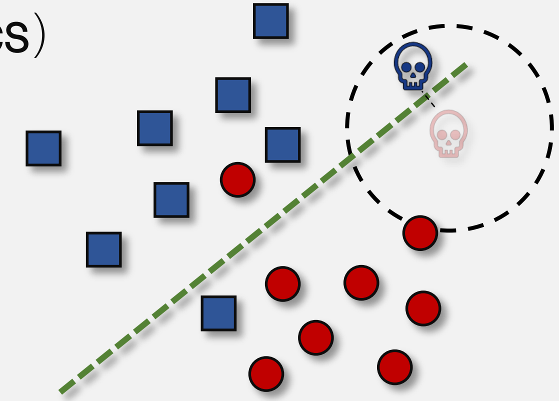


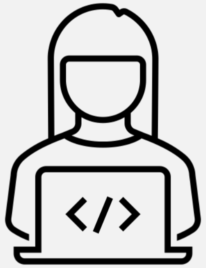
This is known as an *evasion* attack

Suppose not: Integrity Attacks, Pt. II

Under this threat model:

- *Threat*: An adversary can use model information to slightly manipulate inputs
- *Vulnerability*: Inputs can be misclassified (**while preserving underlying semantics**)
- *Exploit*: ?



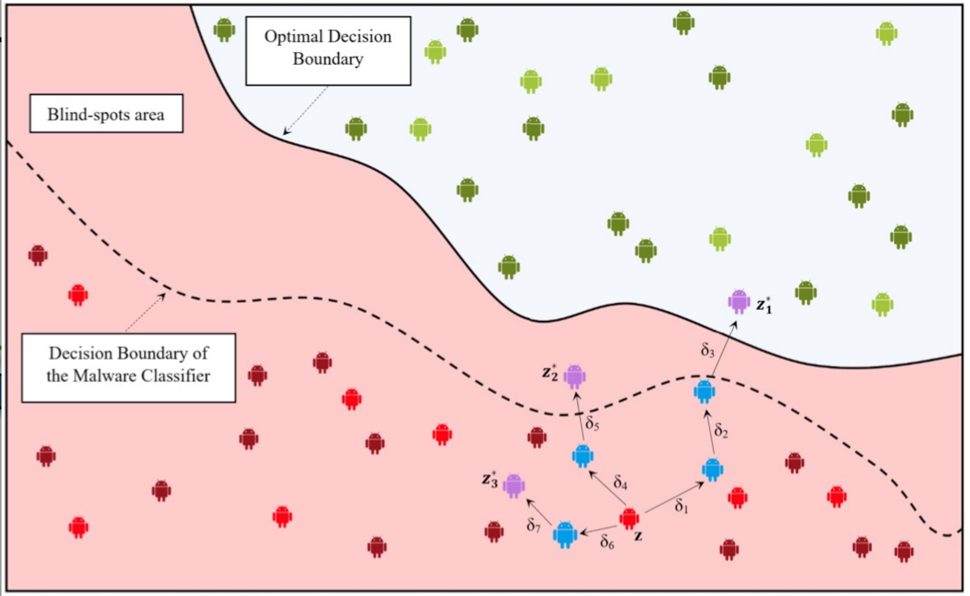


This is known as an *evasion* attack

Under this threat model:

- *Threat:* An adversary can use model information to slightly manipulate inputs
- *Vulnerability:* Inputs can be

servin



A self-driving vehicle controlled by adversaries

Malware that evades detection

Overview

1. ~~Machine Learning~~
2. ~~Integrity~~
3. Confidentiality
4. Availability

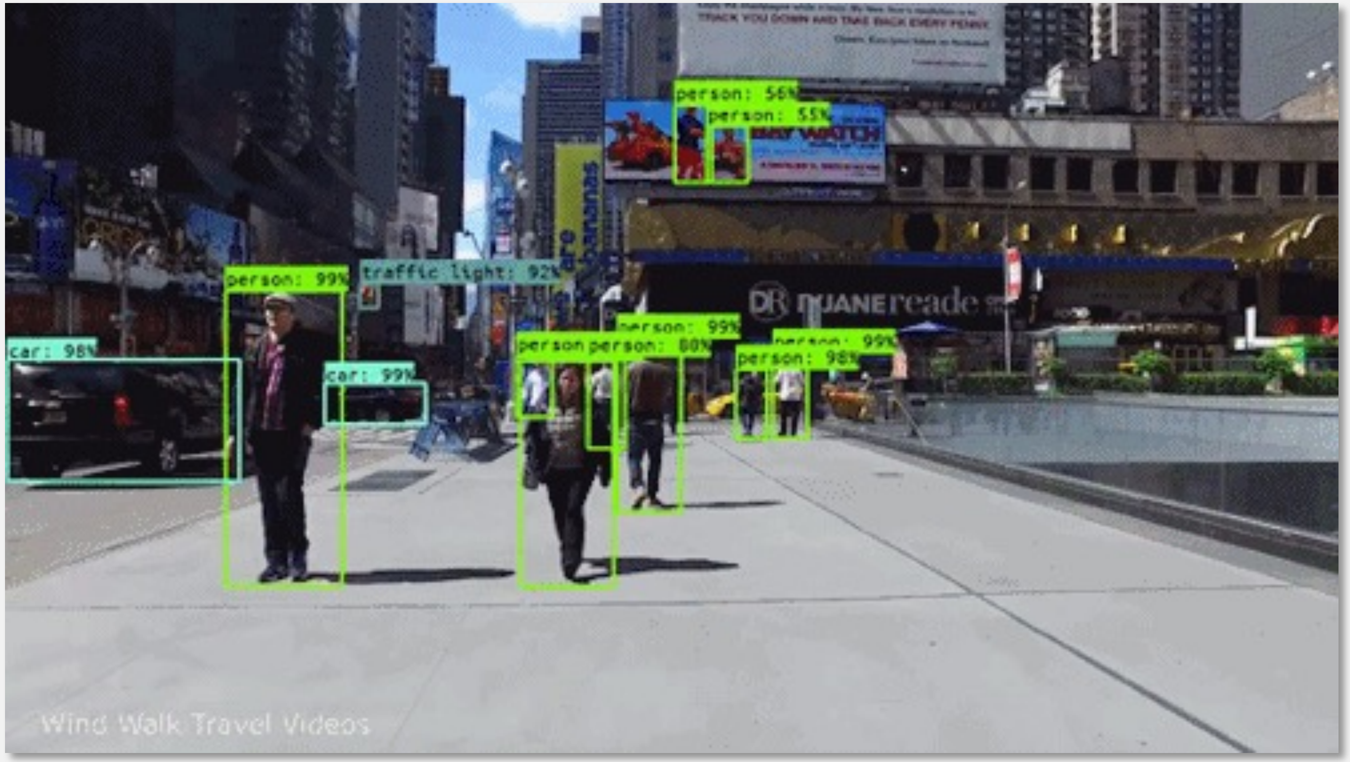
Putting it all together

$$\underbrace{\min_{\theta}}_{\text{Optimizer}} \overbrace{L}^{\text{Error}}(\underbrace{f_\theta}_{\text{Model}}(\overbrace{x}^{\text{Data}}), \underbrace{y}_{\text{Labels}})$$

Data (often *the* most valuable resource) is assumed collected faithfully...

... and the corresponding labels are assumed to be accurately described.

Attacks on Confidentiality



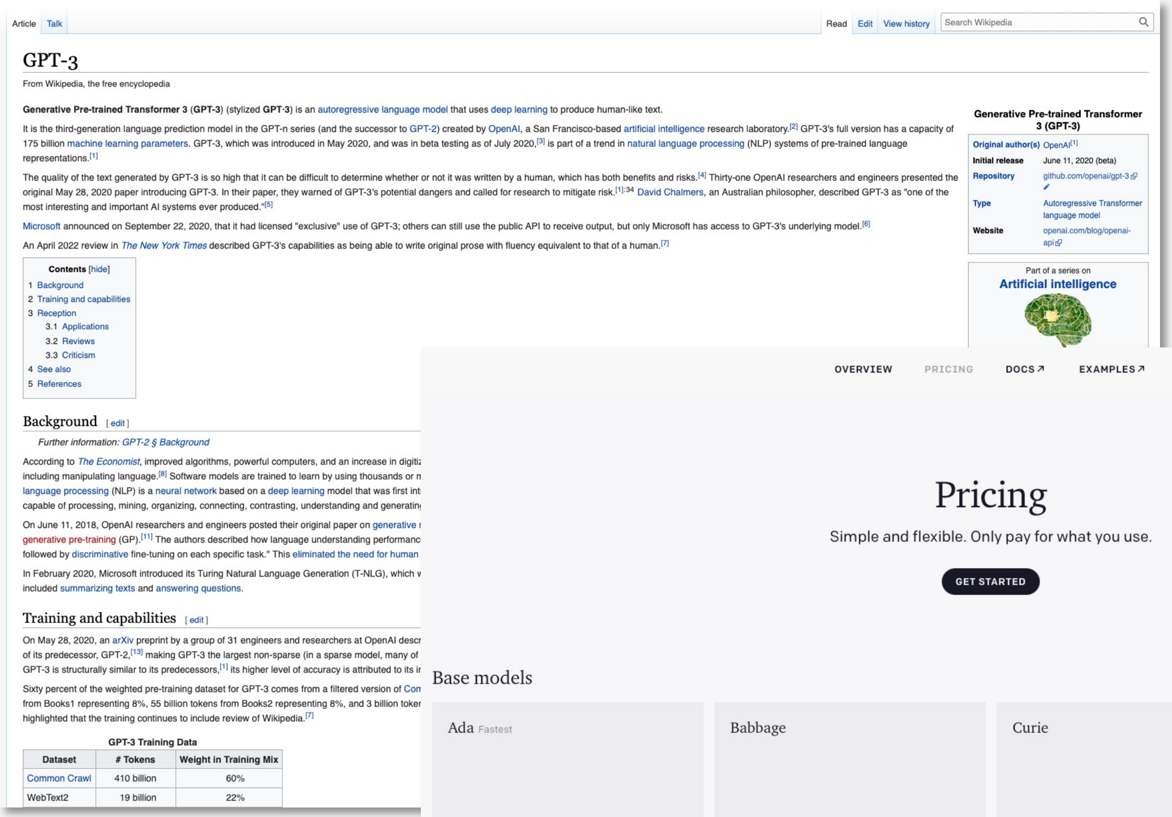
Google AI

Microsoft Azure

amazon Rekognition

Attacks on Confidentiality

Source: https://www.youtube.com/watch?v=eQLcDmfmGB0 & https://ai.google


Article Talk Read Edit View history Search Wikipedia

GPT-3

From Wikipedia, the free encyclopedia

Generative Pre-trained Transformer 3 (**GPT-3**) (stylized **GPT·3**) is an autoregressive language model that uses deep learning to produce human-like text.

It is the third-generation language prediction model in the GPT-n series (and the successor to GPT-2) created by OpenAI, a San Francisco-based artificial intelligence research laboratory.[2] GPT-3's full version has a capacity of 175 billion machine learning parameters. GPT-3, which was introduced in May 2020, and was in beta testing as of July 2020,[3] is part of a trend in natural language processing (NLP) systems of pre-trained language representations.[1]

The quality of the text generated by GPT-3 is so high that it can be difficult to determine whether or not it was written by a human, which has both benefits and risks.[4] Thirty-one OpenAI researchers and engineers presented the original May 28, 2020 paper introducing GPT-3. In their paper, they warned of GPT-3's potential dangers and called for research to mitigate risk.[1]:34 David Chalmers, an Australian philosopher, described GPT-3 as "one of the most interesting and important AI systems ever produced."[5]

Microsoft announced on September 22, 2020, that it had licensed "exclusive" use of GPT-3; others can still use the public API to receive output, but only Microsoft has access to GPT-3's underlying model.[6]

An April 2022 review in *The New York Times* described GPT-3's capabilities as being able to write original prose with fluency equivalent to that of a human.[7]

Generative Pre-trained Transformer 3 (GPT-3)	
Original author(s)	OpenAI[1]
Initial release	June 11, 2020 (beta)
Repository	github.com/openai/gpt-3
Type	Autoregressive Transformer language model
Website	openai.com/blog/openai-api

Part of a series on **Artificial intelligence**

Contents [hide]

Background [edit]

Further information: GPT-2 § Background

According to *The Economist*, improved algorithms, powerful computers, and an increase in digiti
including manipulating language.[8] Software models are trained to learn by using thousands or m
language processing (NLP) is a neural network based on a deep learning model that was first int
capable of processing, mining, organizing, connecting, contrasting, understanding and generatin

On June 11, 2018, OpenAI researchers and engineers posted their original paper on generative
generative pre-training (GP).[11] The authors described how language understanding performanc
followed by discriminative fine-tuning on each specific task." This eliminated the need for human

In February 2020, Microsoft introduced its Turing Natural Language Generation (T-NLG), which w
included summarizing texts and answering questions.

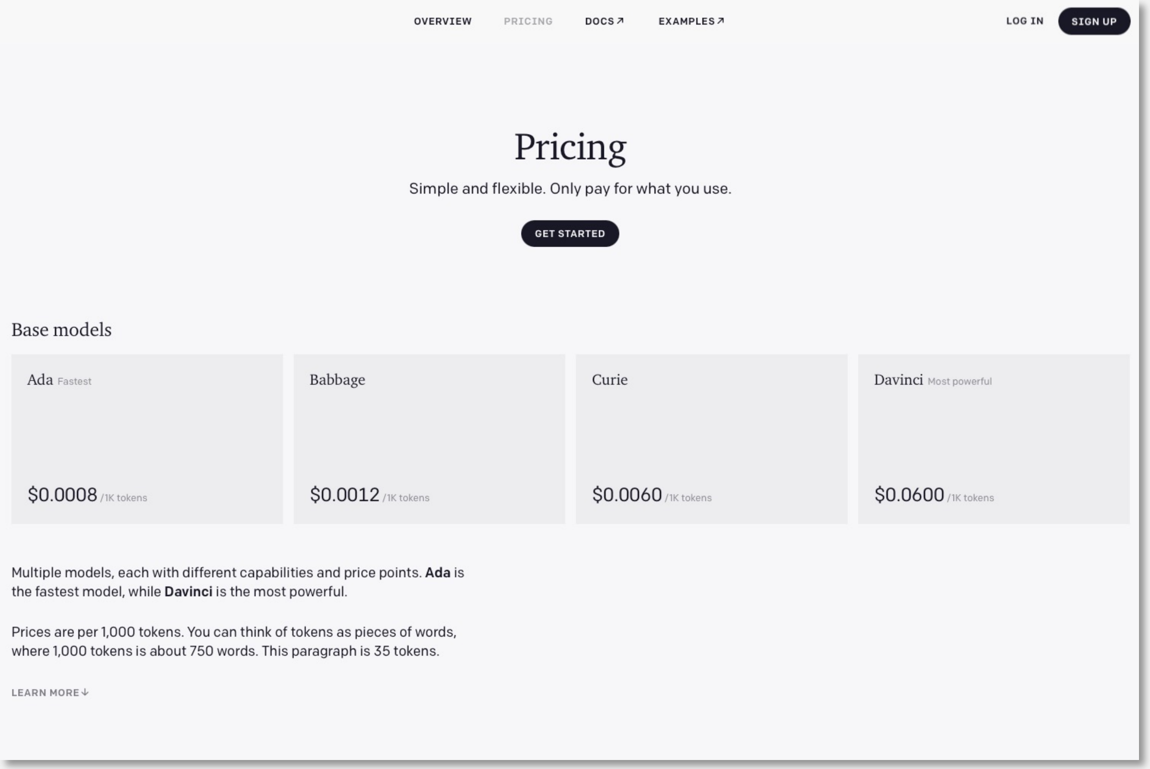
Training and capabilities [edit]

On May 28, 2020, an arXiv preprint by a group of 31 engineers and researchers at OpenAI descr
of its predecessor, GPT-2,[13] making GPT-3 the largest non-sparse (in a sparse model, many of
GPT-3 is structurally similar to its predecessors,[1] its higher level of accuracy is attributed to its i

Sixty percent of the weighted pre-training dataset for GPT-3 comes from a filtered version of Com
from Books1 representing 8%, 55 billion tokens from Books2 representing 8%, and 3 billion toker
highlighted that the training continues to include review of Wikipedia.[7]

GPT-3 Training Data

Dataset	# Tokens	Weight in Training Mix
Common Crawl	410 billion	60%
WebText2	19 billion	22%


OVERVIEW PRICING DOCS ↗ EXAMPLES ↗ LOG IN SIGN UP

Pricing

Simple and flexible. Only pay for what you use.

GET STARTED

Base models

Ada Fastest	Babbage	Curie	Davinci Most powerful
$0.0008 /1K tokens	$0.0012 /1K tokens	$0.0060 /1K tokens	$0.0600 /1K tokens

Multiple models, each with different capabilities and price points. **Ada** is the fastest model, while **Davinci** is the most powerful.

Prices are per 1,000 tokens. You can think of tokens as pieces of words, where 1,000 tokens is about 750 words. This paragraph is 35 tokens.

LEARN MORE ↓

Attacks on Confidentiality

Source: https://en.wikipedia.org/wiki/GPT-3 & https://openai.com/api/pricing/

Error Data

$$\min_{\theta} L(f_\theta(x), y)$$

Optimizer Model Labels

Attacks on Confidentiality

$$\min_{\theta} L(f_\theta(x), y)$$

Error

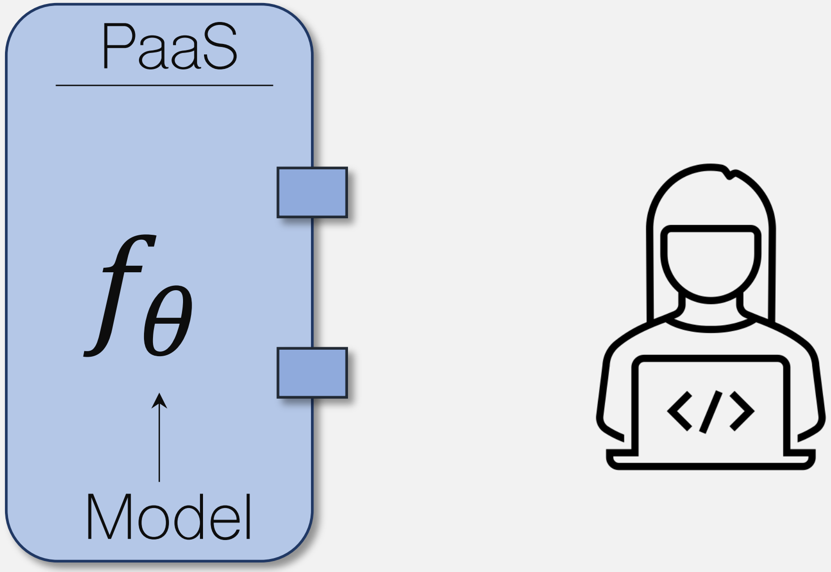
Data

Optimizer

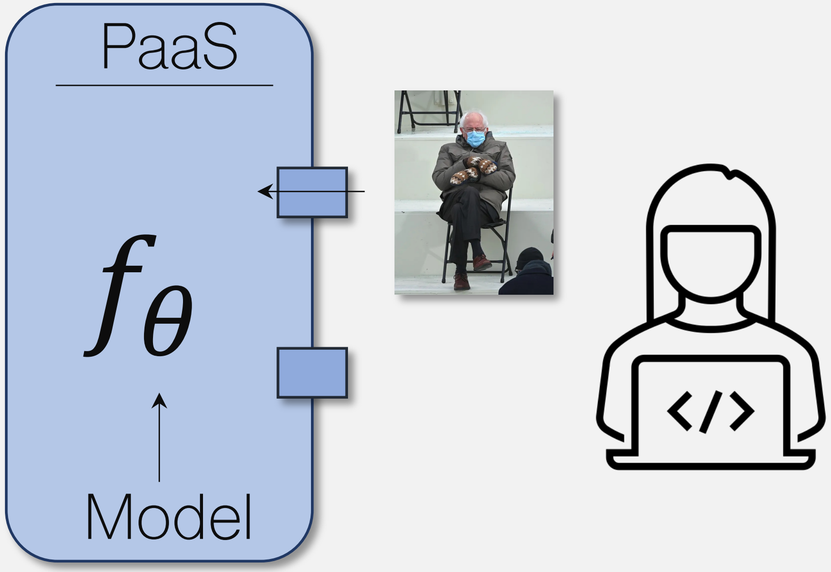
Model

Labels

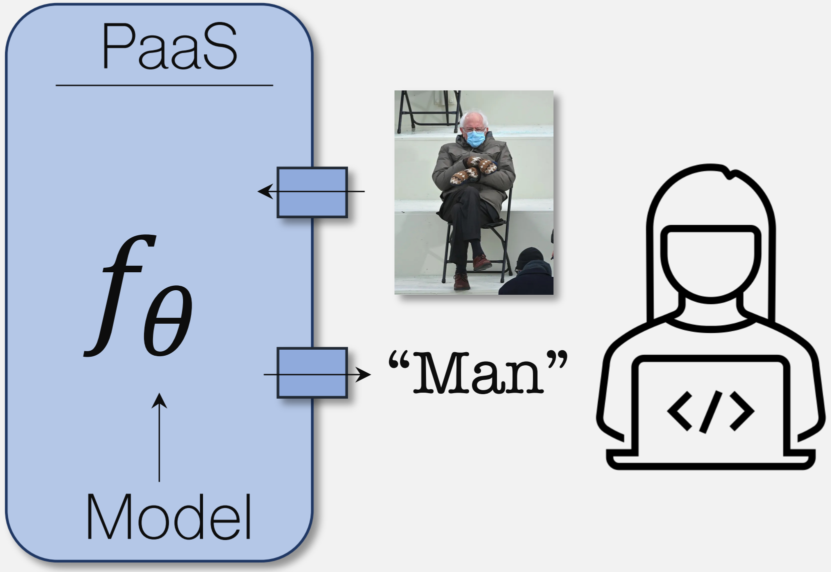
Attacks on Confidentiality



Attacks on Confidentiality

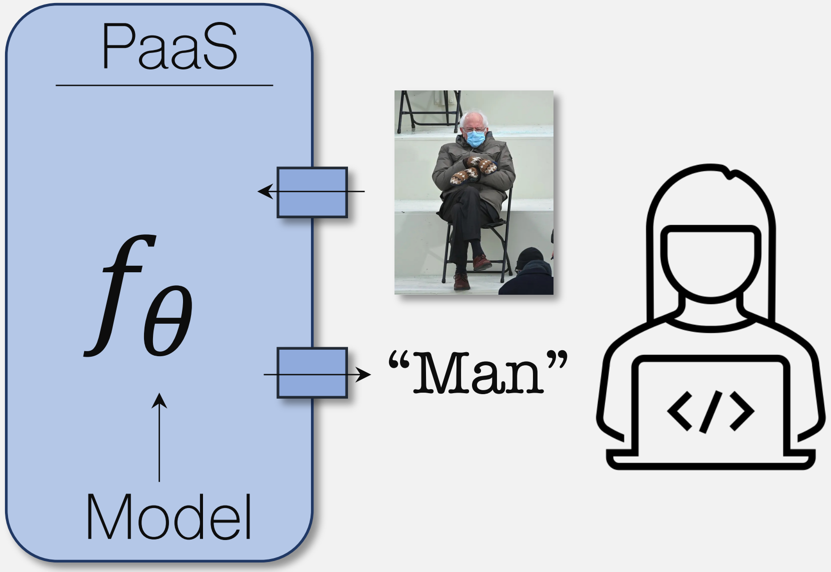


Attacks on Confidentiality



Attacks on Confidentiality

How can this be *exploited??*



Attacks on Confidentiality

Model Theft

Stealing Machine Learning Models via Prediction APIs

Florian Tramèr
EPFL

Fan Zhang
Cornell University

Ari Juels
Cornell Tech, Jacobs Institute

Michael K. Reiter
UNC Chapel Hill

Thomas Ristenpart
Cornell Tech

Abstract

Machine learning (ML) models may be deemed confidential due to their sensitive training data, commercial value, or use in security applications. Increasingly often, confidential ML models are being deployed with publicly accessible query interfaces. ML-as-a-service ("pre-

of possibly confidential feature-vector inputs (e.g., digitized health records) with corresponding output class labels (e.g., a diagnosis) serves to train a predictive model that can generate labels on future inputs. Popular models include support vector machines (SVMs), logistic regressions, neural networks, and decision trees.

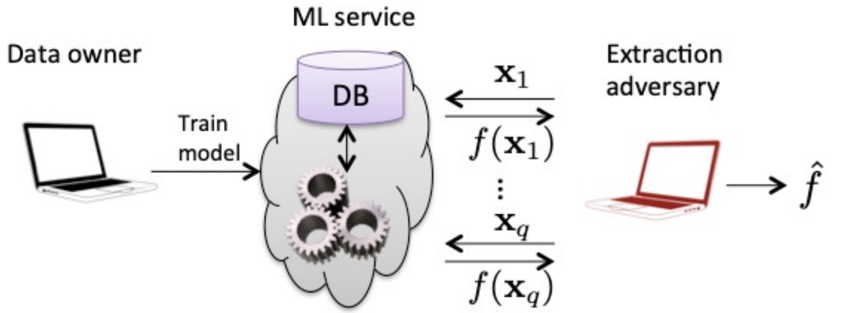


Figure 1: **Diagram of ML model extraction attacks.** A data owner has a model f trained on its data and allows others to make prediction queries. An adversary uses q prediction queries to extract an $\hat{f} \approx f$.

Source: *https://arxiv.org/pdf/1609.02943.pdf*

Input

$$f_\theta(x)$$

Model

Model Theft

Input

1st Feature

2nd Feature

$$f_\theta(x) = \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \beta$$

Model

1st Parameter

2nd Parameter

Bias

Model Theft

Input

1st Feature

2nd Feature

$$f_\theta(x) = \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \beta$$

Model

1st Parameter

2nd Parameter

Bias

Model Theft

$$f_\theta(\langle 1, 0 \rangle) = \theta_1 + \beta$$

$$f_\theta(\langle 0, 1 \rangle) = \theta_2 + \beta$$

$$f_\theta(\langle 0, 0 \rangle) = \beta$$

Model Theft

$$f_\theta(\langle 1, 0 \rangle) = \theta_1 + \beta$$

$$f_\theta(\langle 0, 1$$

$$f_\theta(\langle 0, 0 \rangle) = \beta$$

With $d + 1$ queries, perfect extraction is possible

Model Theft

Parameters

Inputs

Fidelity

Model	Unknowns	Queries	$1 - R_{test}$	$1 - R_{unif}$	Time (s)
Softmax	530	265	99.96%	99.75%	2.6
		530	100.00%	100.00%	3.1
OvR	530	265	99.98%	99.98%	2.8
		530	100.00%	100.00%	3.5
MLP	2,225	1,112	98.17%	94.32%	155
		2,225	98.68%	97.23%	168
		4,450	99.89%	99.82%	195
		11,125	99.96%	99.99%	89

Table 4: **Success of equation-solving attacks.** Models to extract were trained on the Adult data set with multiclass target 'Race'. For each model, we report the number of unknown model parameters, the number of queries used, and the running time of the equation solver. The attack on the MLP with 11,125 queries converged after 490 epochs.

Model Theft

Source: https://arxiv.org/pdf/1609.02943.pdf

Parameters

Inputs

Fidelity

Model	Unknowns	Queries	$1 - R_{\text{test}}$	$1 - R_{\text{unif}}$	Time (s)
Softmax	530	265	99.96%	99.75%	2.6
		530	100.00%	100.00%	3.1
OvR	530	265	99.98%	99.98%	2.8
		530	100.00%	100.00%	3.5
MLP	2,225	1,112	98.17%	94.32%	155
		2,225	98.68%	97.23%	168
		4,450	99.89%	99.82%	195
		11,125	99.96%	99.99%	89

Table 4: **Success of equation-solving attacks.** Models to extract were trained on the Adult data set with multiclass target 'Race'. For each model, we report the number of unknown model parameters, the number of queries used, and the running time of the equation solver. The attack on the MLP with 11,125 queries converged after 490 epochs.

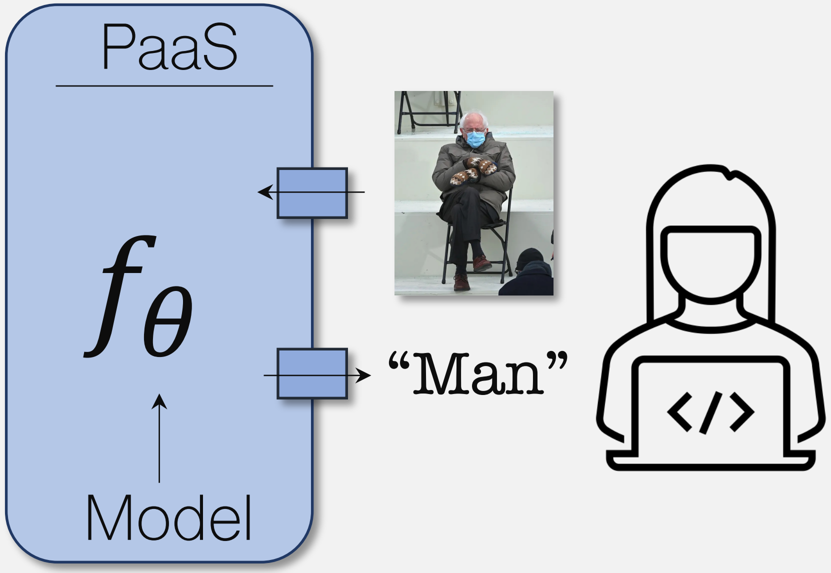
Model	OHE	Binning	Queries	Time (s)	Price ($)
Circles	-	Yes	278	28	0.03
Digits	-	No	650	70	0.07
Iris	-	Yes	644	68	0.07
Adult	Yes	Yes	1,485	149	0.15

Table 7: **Results of model extraction attacks on Amazon.** OHE stands for one-hot-encoding. The reported query count is the number used to find quantile bins (at a granularity of 10^{-3}), plus those queries used for equation-solving. Amazon charges $0.0001 per prediction [1].

Model Theft

Source: https://arxiv.org/pdf/1609.02943.pdf

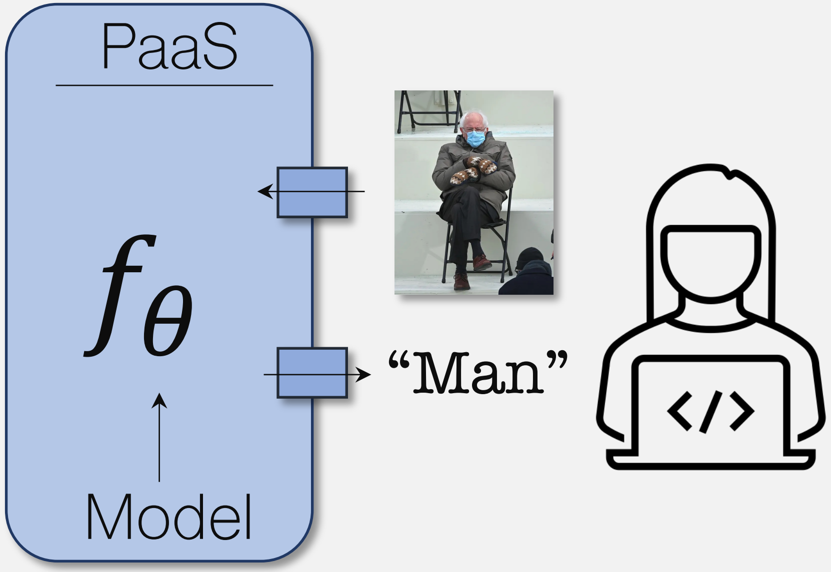
Under this threat model:



Attacks on Confidentiality

Under this threat model:

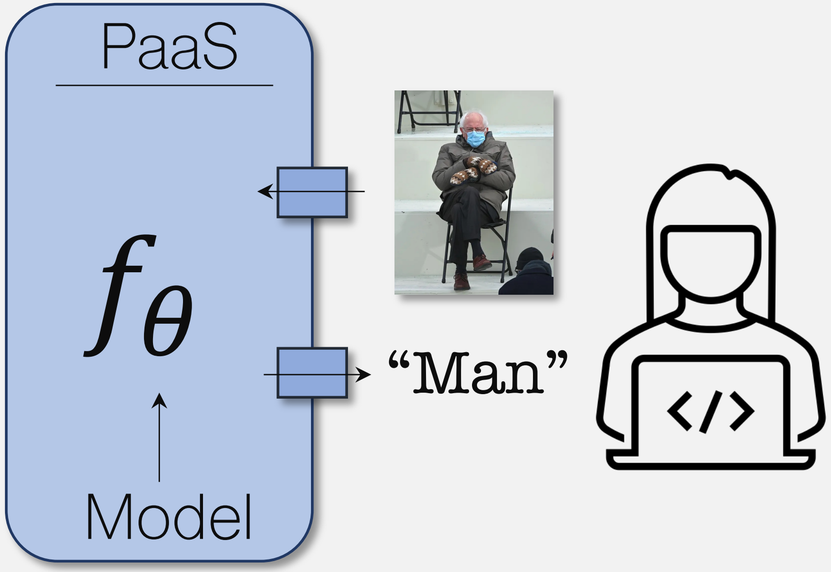
- *Threat:* An adversary can query arbitrary inputs



Attacks on Confidentiality

Under this threat model:

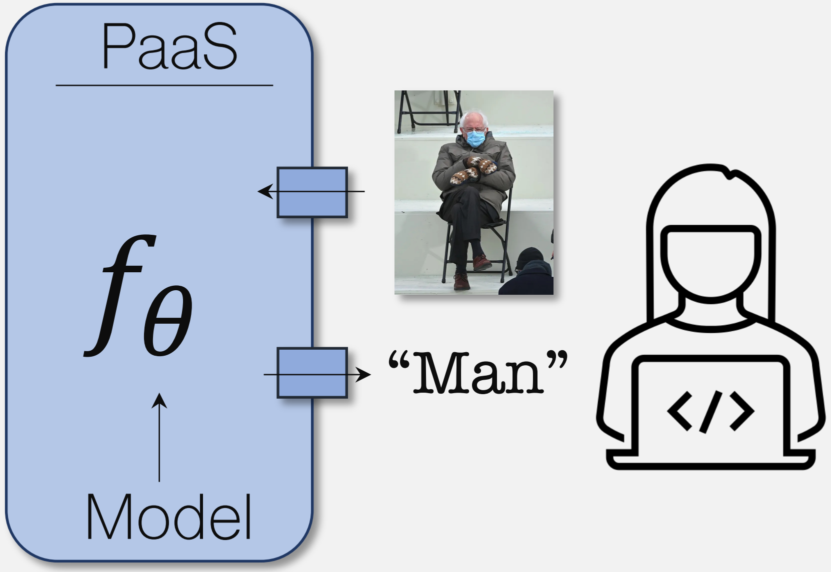
- *Threat:* An adversary can query arbitrary inputs
- *Vulnerability:* Inputs can leak varying degrees of model information



Attacks on Confidentiality

Under this threat model:

- *Threat:* An adversary can query arbitrary inputs
- *Vulnerability:* Inputs can leak varying degrees of model information
- *Exploit:* ?



Attacks on Confidentiality

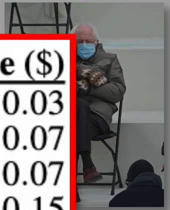
Under this threat model:

- *Threat:* An adversary can query arbitrary inputs
- *Vulnerability:* Inputs can leak varying degrees of model information
- *Exploit:* ?

PaaS

Model	OHE	Binning	Queries	Time (s)	Price ($)
Circles	-	Yes	278	28	0.03
Digits	-	No	650	70	0.07
Iris	-	Yes	644	68	0.07
Adult	Yes	Yes	1,485	149	0.15

Table 7: **Results of model extraction attacks on Amazon.** OHE stands for one-hot-encoding. The reported query count is the number used to find quantile bins (at a granularity of 10^{-3}), plus those queries used for equation-solving. Amazon charges $0.0001 per prediction [1].



an”

Figure 2: **Training data leakage in KLR models.** (a) Displays 5 of 20 training samples used as representers in a KLR model (top) and 5 of 20 extracted representers (bottom). (b) For a second model, shows the average of all 1,257 representers that the model classifies as a 3, 4, 5, 6 or 7 (top) and 5 of 10 extracted representers (bottom).

ks on

entiality

Intellectual property can be stolen (cheaply)

Training data can be recovered (privacy)

https://arxiv.org/pdf/1609.02943.pdf

Overview

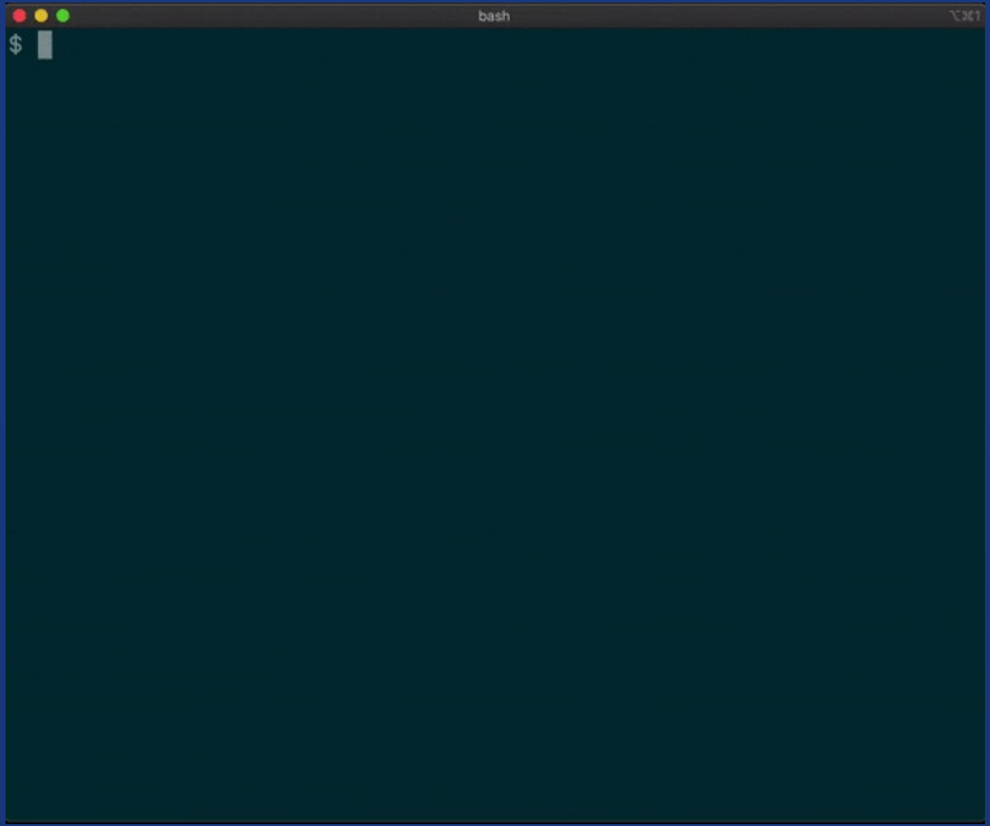
1. ~~Machine Learning~~
2. ~~Integrity~~
3. ~~Confidentiality~~
4. Availability

Attacking Availability

What does it mean for machine learning to be "available?"

Attacking Availability

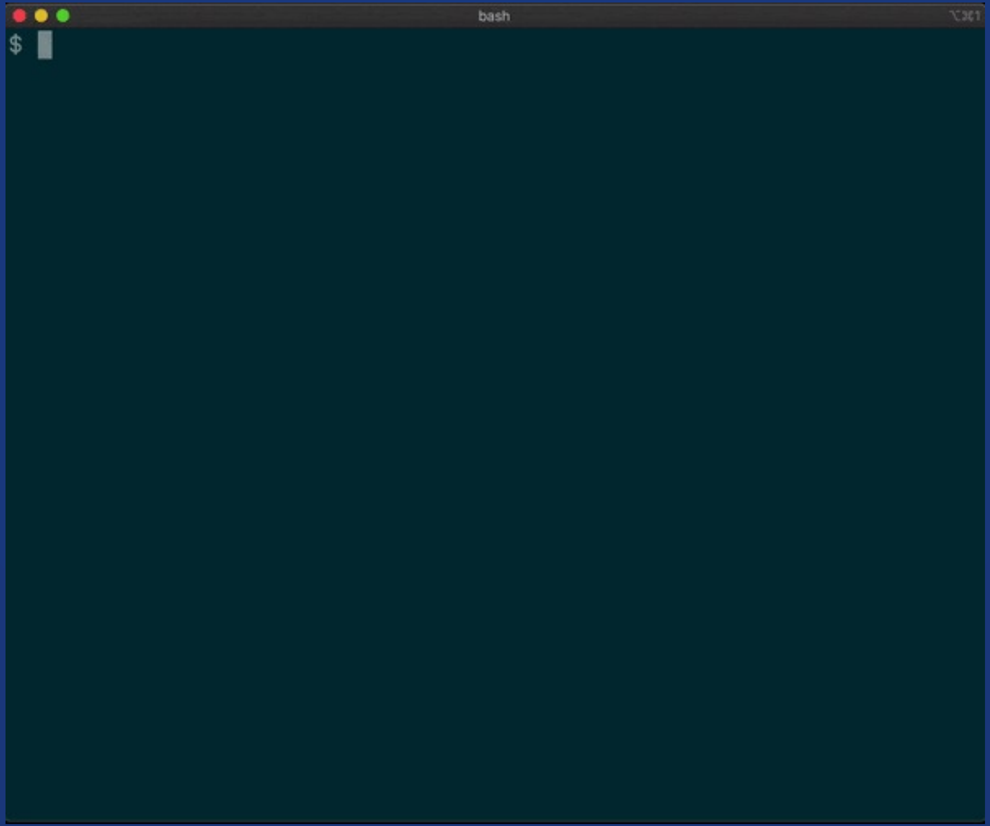
What does it mean for machine learning to be "available?"



GTP-3: The 4.3-million-dollar comedian

Attacking Availability

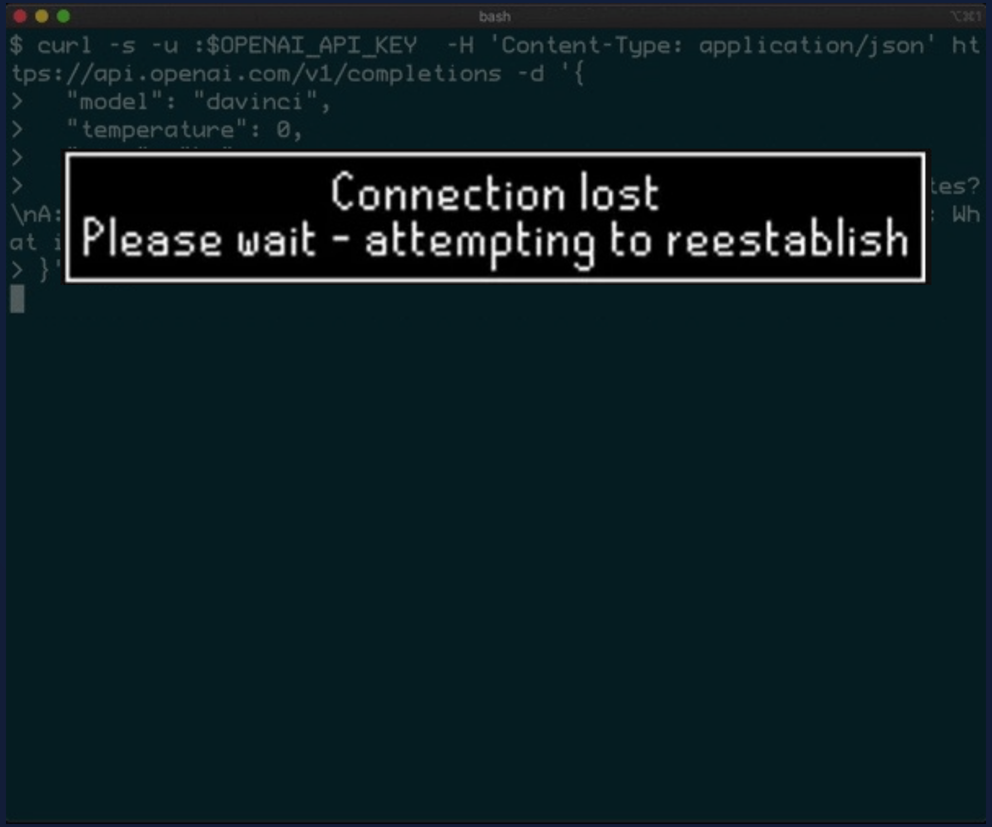
What does it mean for machine learning to be "available?"



GTP-3: The... 4.3-million-dollar.... Zzzzz.... Zzzzzz...

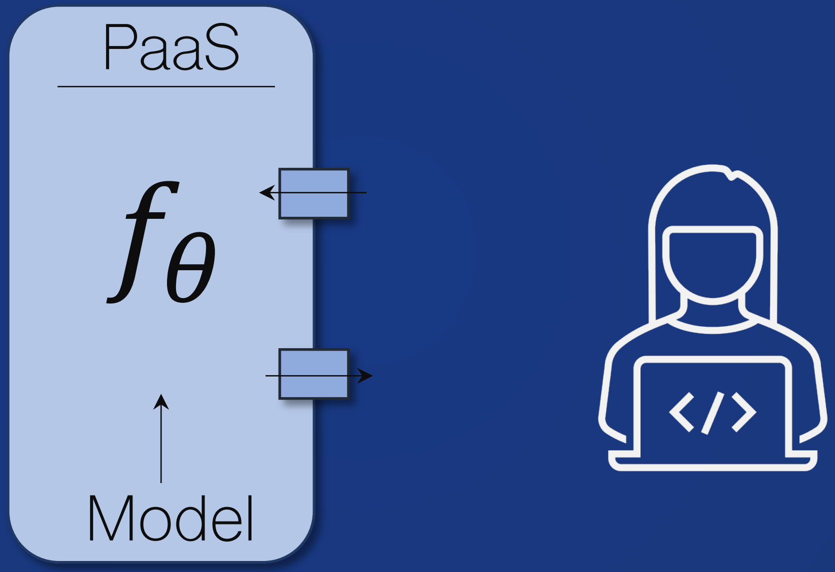
Attacking Availability

What does it mean for machine learning to be "available?"

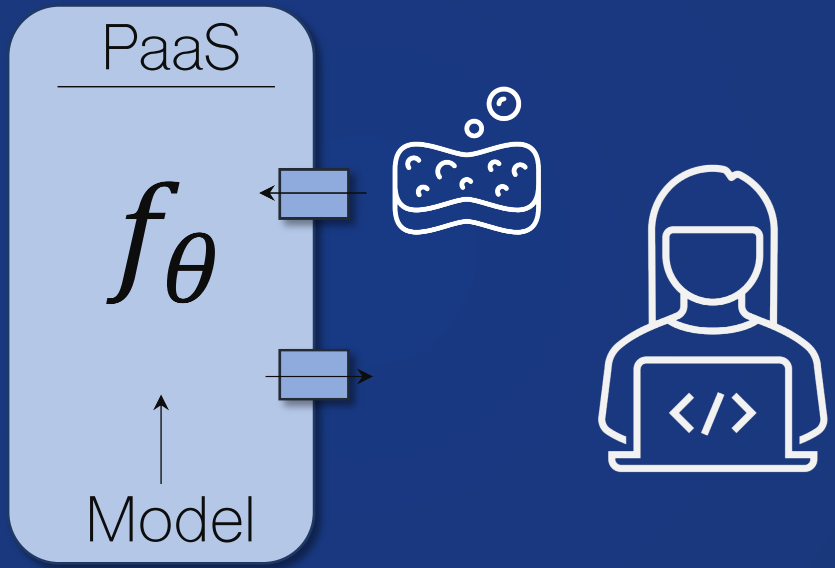


GTP-3: The... 4.3-million-dollar.... Zzzzz.... Zzzzzz...

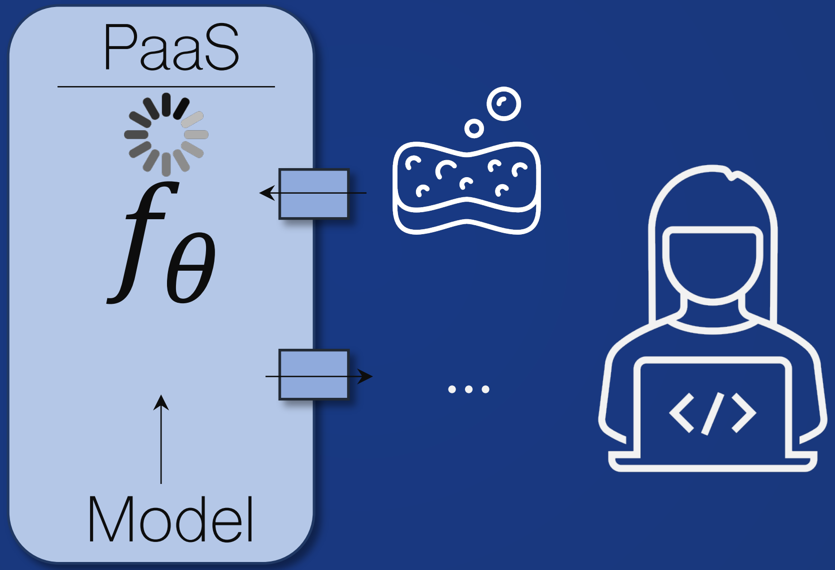
Attacking Availability



Attacking Availability



Attacking Availability



Sponge Attacks

Sponge Examples: Energy-Latency Attacks on Neural Networks

A Preprint

Ilia Shumailov
University of Cambridge
ilia.shumailov@cl.cam.ac.uk

Yiren Zhao
University of Cambridge
yiren.zhao@cl.cam.ac.uk

Daniel Bates
University of Cambridge
daniel.bates@cl.cam.ac.uk

Nicolas Papernot
University of Toronto and Vector Institute
nicolas.papernot@utoronto.ca

Robert Mullins
University of Cambridge
robert.mullins@cl.cam.ac.uk

Ross Anderson
University of Cambridge
ross.anderson@cl.cam.ac.uk

May 13, 2021

Abstract

The high energy costs of neural network training and inference led to the use of acceleration hardware such as GPUs and TPUs. While such devices enable us to train large-scale neural networks in

Sponge Attacks

Sponge Examples: Energy-Latency Attacks on Neural Networks

A Preprint

Ilia Shumailov
University of Cambridge
ilia.shumailov@cl.cam.ac.uk

Yiren Zhao
University of Cambridge
yiren.zhao@cl.cam.ac.uk

Daniel Bates
University of Cambridge
daniel.bates@cl.cam.ac.uk

Nicolas Papernot
University of Toronto and Vector Institute
nicolas.papernot@utoronto.ca

Robert Mullins
University of Cambridge
robert.mullins@cl.cam.ac.uk

Ross Anderson
University of Cambridge
ross.anderson@cl.cam.ac.uk

May 13, 2021

Abstract

The high energy costs of neural network training and inference led to the use of acceleration hardware such as GPUs and TPUs. While such devices enable us to train large-scale neural networks in

4.2 The Energy Gap

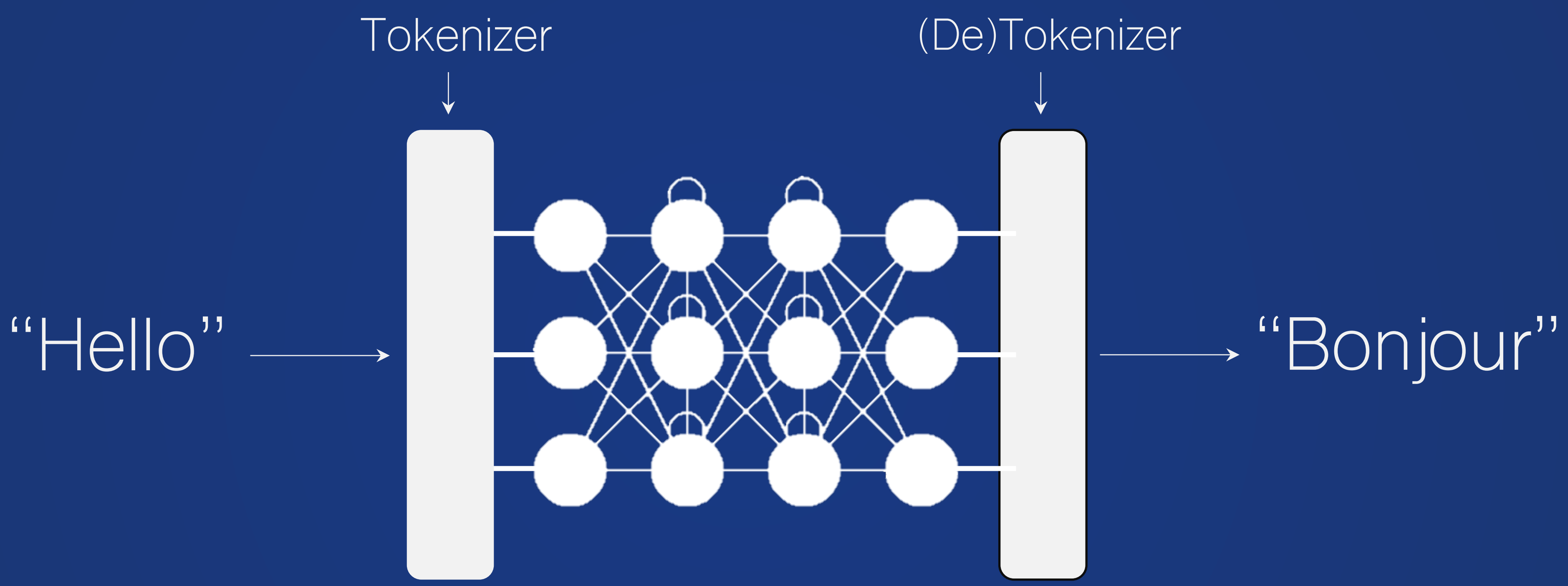
The *Energy Gap* is the performance gap between average-case and worst-case performance, and is the target for our sponge attacks. To better understand the cause of this gap, we tested three hardware platforms: a CPU, a GPU and an ASIC simulator. The amount of energy consumed by one inference pass (*i.e.* a forward pass in a neural network) depends primarily on [45]:

- the overall number of arithmetic operations required to process the inputs; and
- the number of memory accesses *e.g.* to the GPU DRAM.

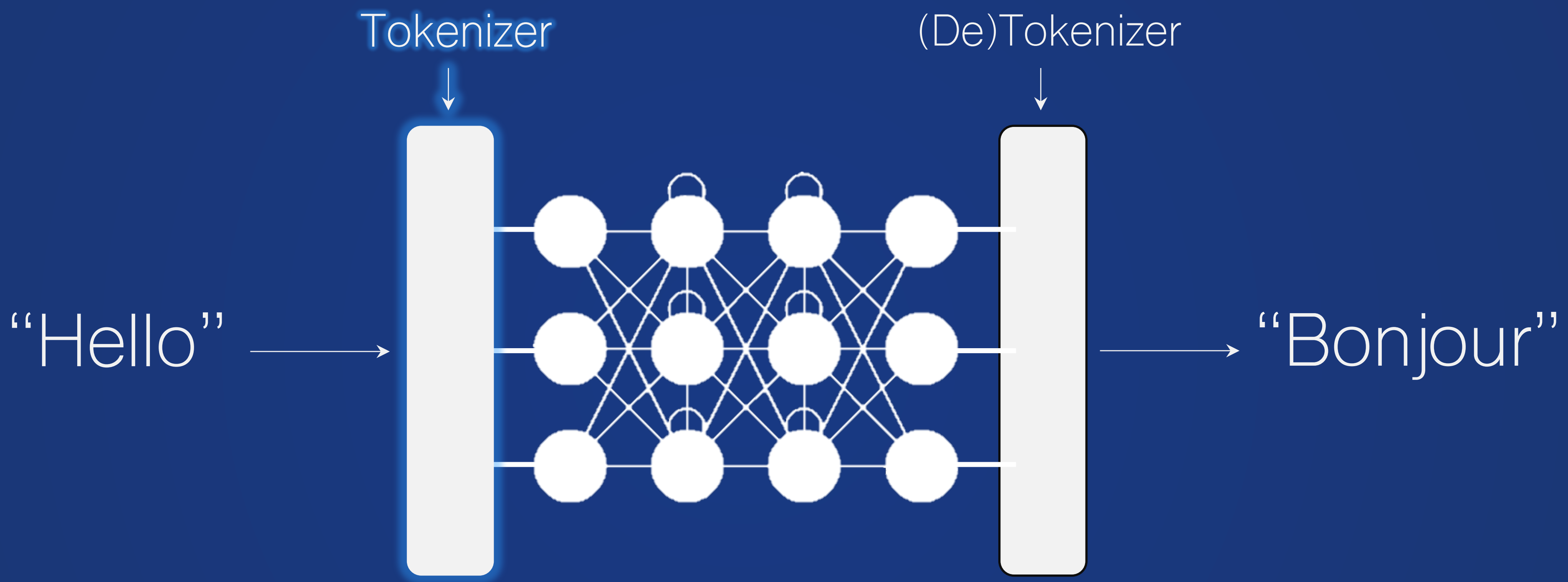
The intriguing question now is:

is there a significant gap in energy consumption for different model inputs of the same dimension?

Sponge Attacks



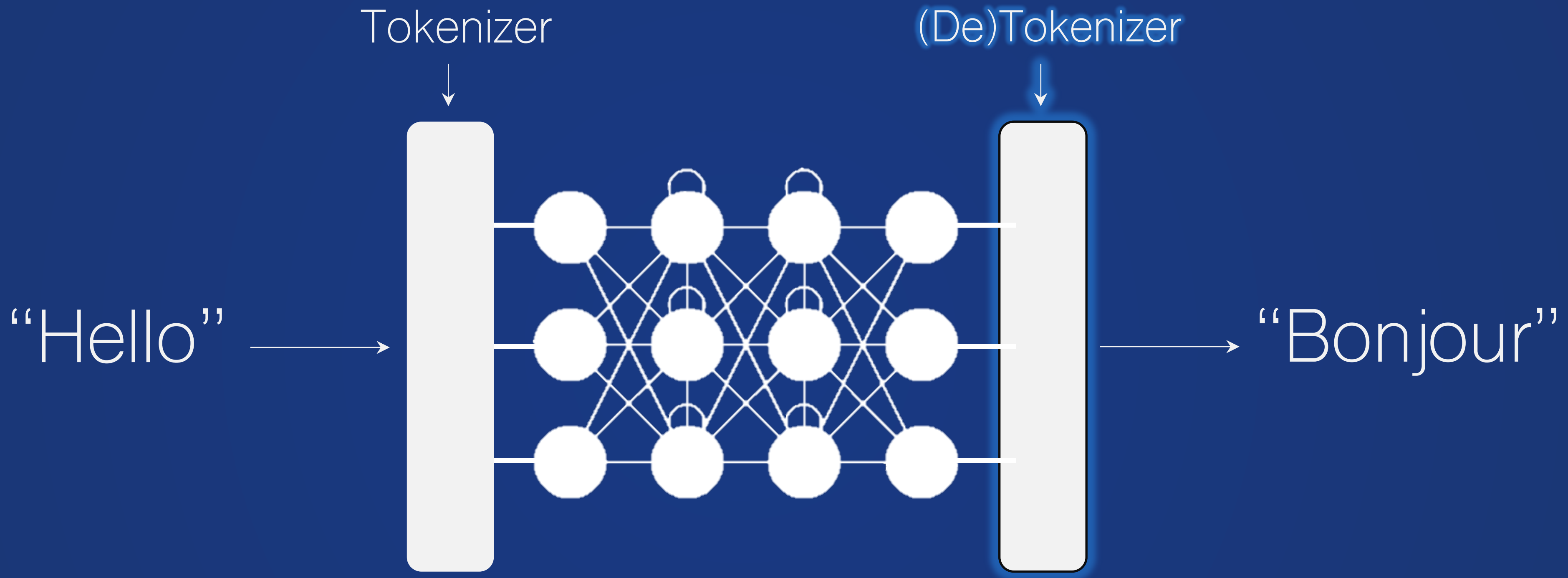
Sponge Attacks



Observations:

1. Lots of (uncommon) input tokens = lots of compute

Sponge Attacks



Observations:

1. Lots of (uncommon) input tokens = lots of compute
2. Maximize output sequence length = lots of compute

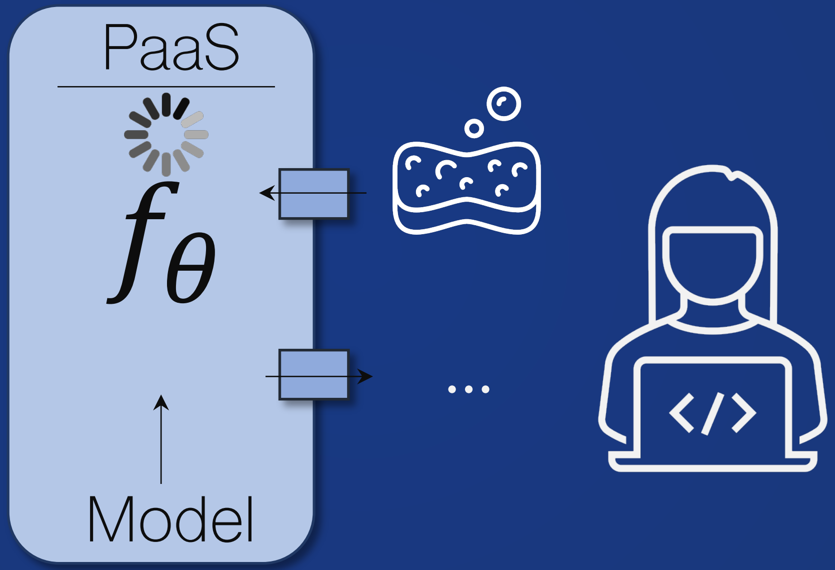
Sponge Attacks

From	To		ASIC Energy [mJ]	GPU Time [S]	GPU Energy [mJ]	CPU Time [S]	CPU Energy [mJ]
White-box							
$\mathrm{WMT16}_{en \to de}$ [64]	$\mathrm{WMT16}_{en \to de}$ [64]	Sponge	48447.093	2.414	260187.900	13.615	781758.680
		Natural	1360.118	0.056	6355.620	0.520	23262.311
			35.62×	**42.98**×	**40.94**×	26.20×	33.61×

https://arxiv.org/pdf/2006.03463.pdf

Attacking Availability

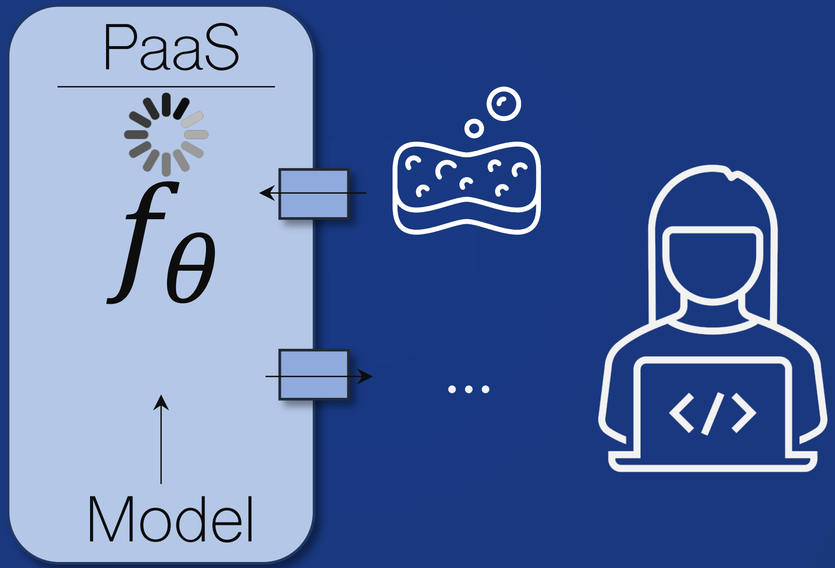
Under this threat model:



Attacking Availability

Under this threat model:

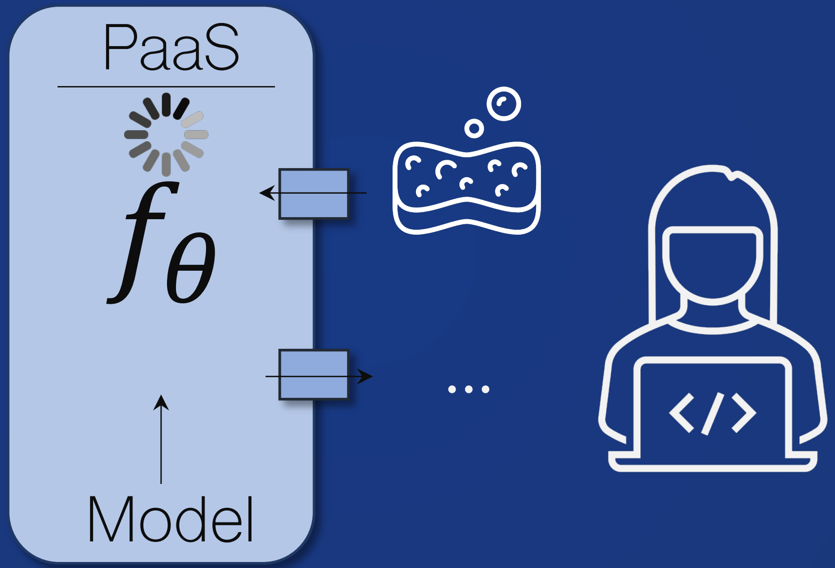
- *Threat:* An adversary can query arbitrary inputs



Attacking Availability

Under this threat model:

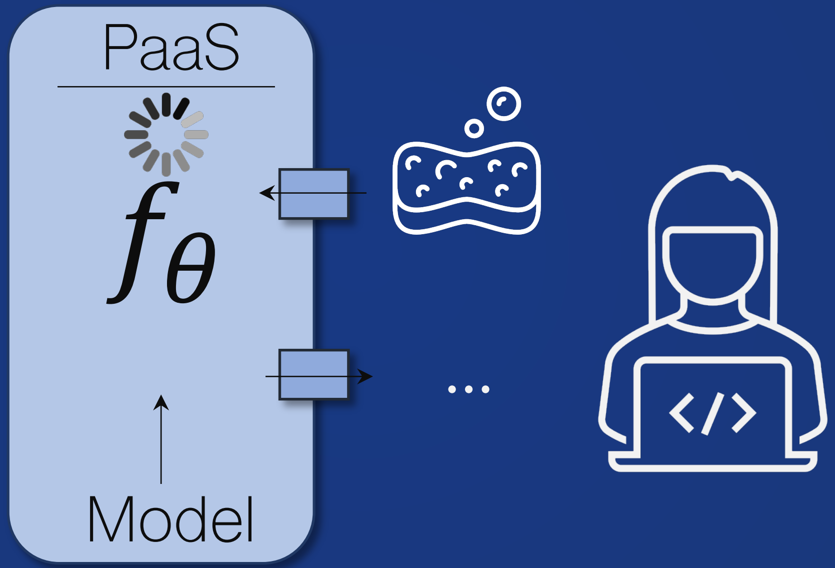
- *Threat:* An adversary can query arbitrary inputs
- *Vulnerability:* Model throughput is input-specific



Attacking Availability

Under this threat model:

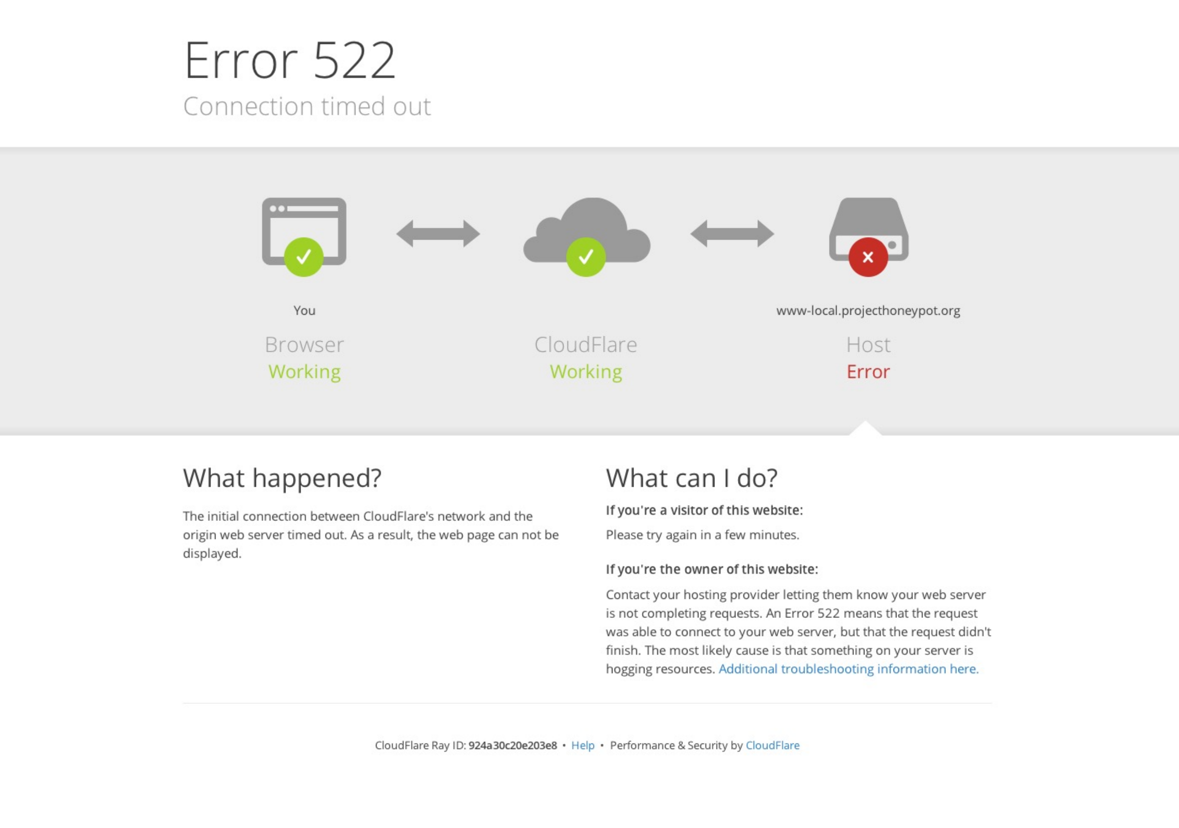
- *Threat:* An adversary can query arbitrary inputs
- *Vulnerability:* Model throughput is input-specific
- *Exploit:* ?



Attacking Availability

Under this threat model:

- *Threat:* An adversary c
  query arbitrary inputs
- *Vulnerability:* Model
  throughput is input-spe
- *Exploit:* ?



An unusable Predictions-as-a-Service platform

Overview

1. Machine Learning $\min_{\theta} L(f_{\theta}(x), y)$

Overview

1. Machine Learning $\min_{\theta} L(f_\theta(x), y)$

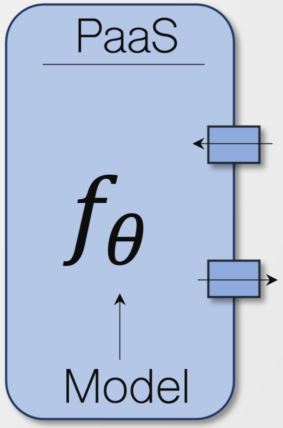
2. Integrity
   I. $\min_{\theta} L(f_\theta(x), y)$
   II. $\min_{\cancel{\theta} \to x} L(f_\theta(x), y_{\text{airplane}})$

Overview

1. Machine Learning $\min_{\theta} L(f_\theta(x), y)$

2. Integrity I. $\min_{\theta} L(f_\theta(x), y)$ II. $\min_{\cancel{\theta} \to x} L(f_\theta(x), y_{\text{airplane}})$

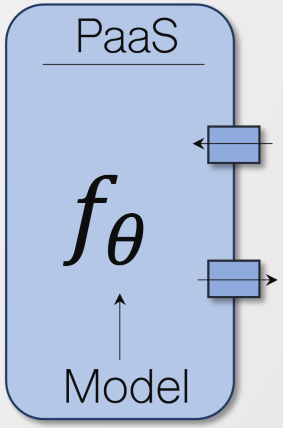
3. Confidentiality



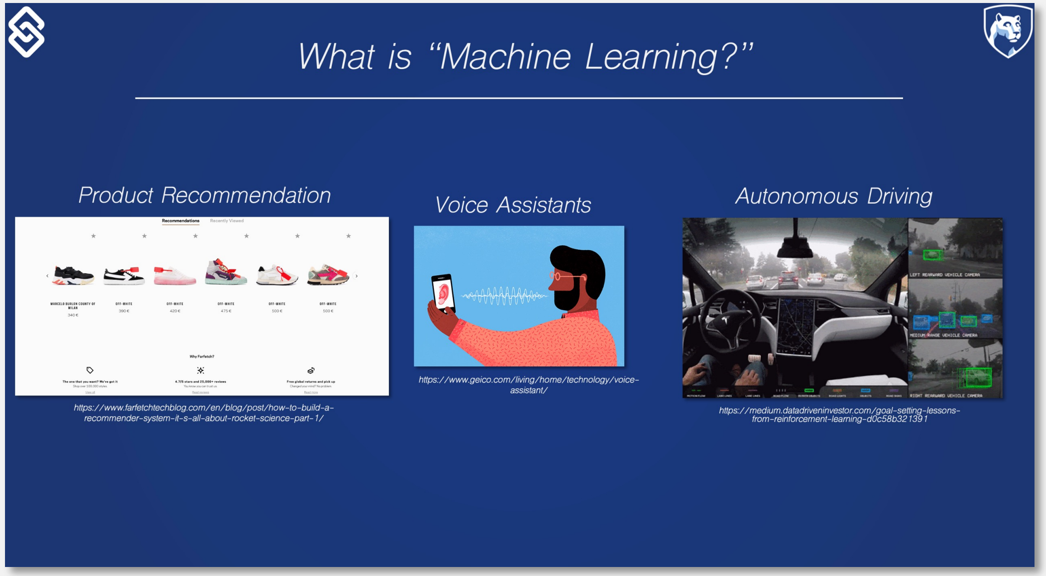
Overview

1. Machine Learning $\min_{\theta} L(f_\theta(x), y)$
2. Integrity I. $\min_{\theta} L(f_\theta(x), y)$ II. $\min_{\cancel{\theta}\to x} L(f_\theta(x), y_{\text{airplane}})$
3. Confidentiality

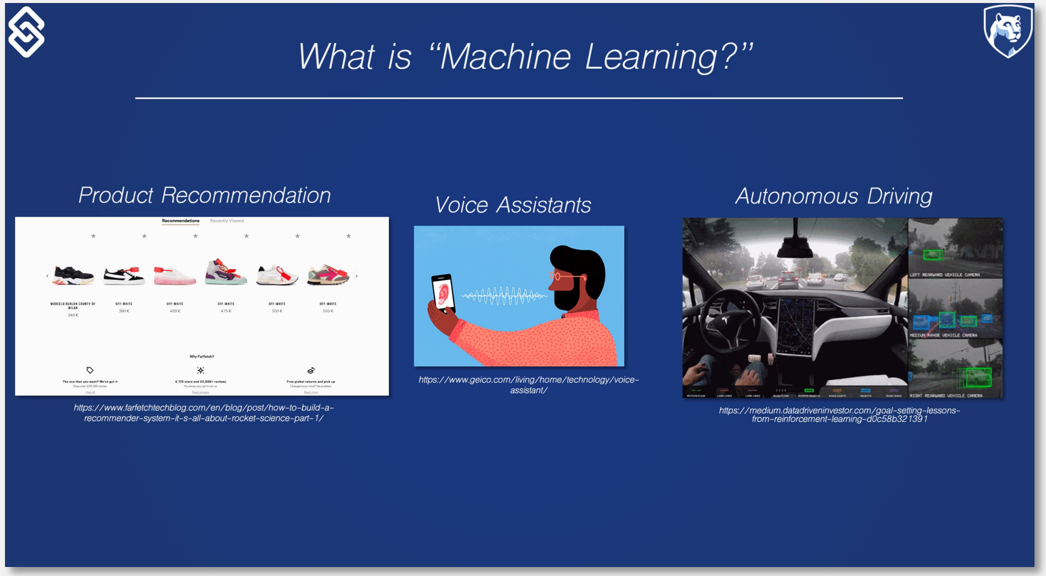
4. Availability



Machine Learning: The Bottom Line

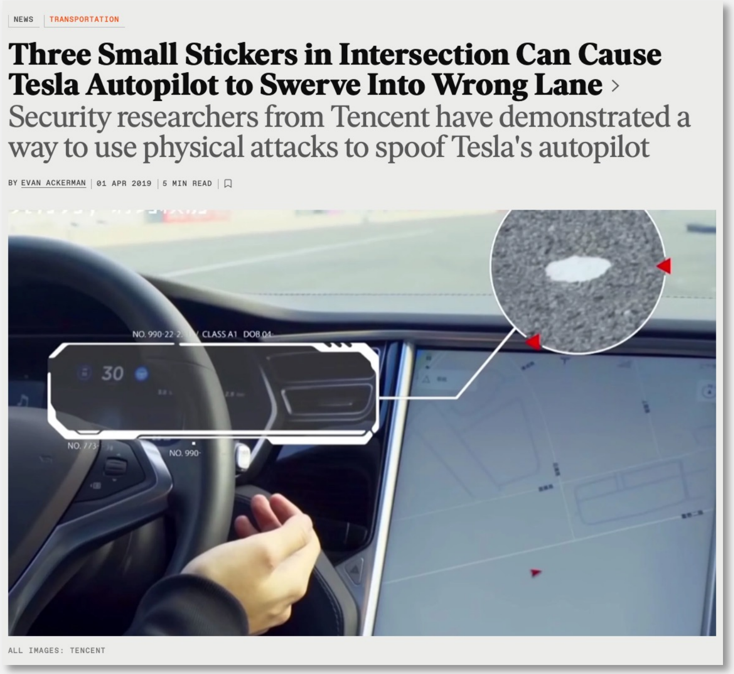


This tech is here to stay…



This tech is here to stay...

... and we'll get it wrong at first


NEWS | TRANSPORTATION

Three Small Stickers in Intersection Can Cause Tesla Autopilot to Swerve Into Wrong Lane

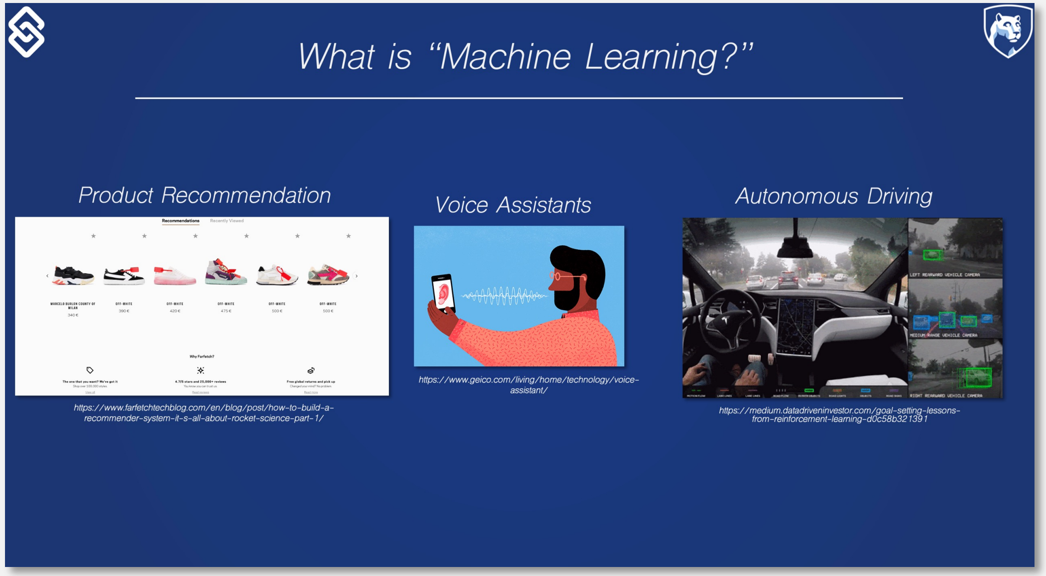
Security researchers from Tencent have demonstrated a way to use physical attacks to spoof Tesla's autopilot

BY EVAN ACKERMAN | 01 APR 2019 | 5 MIN READ

ALL IMAGES: TENCENT

Security

Machine Learning: The Bottom Line

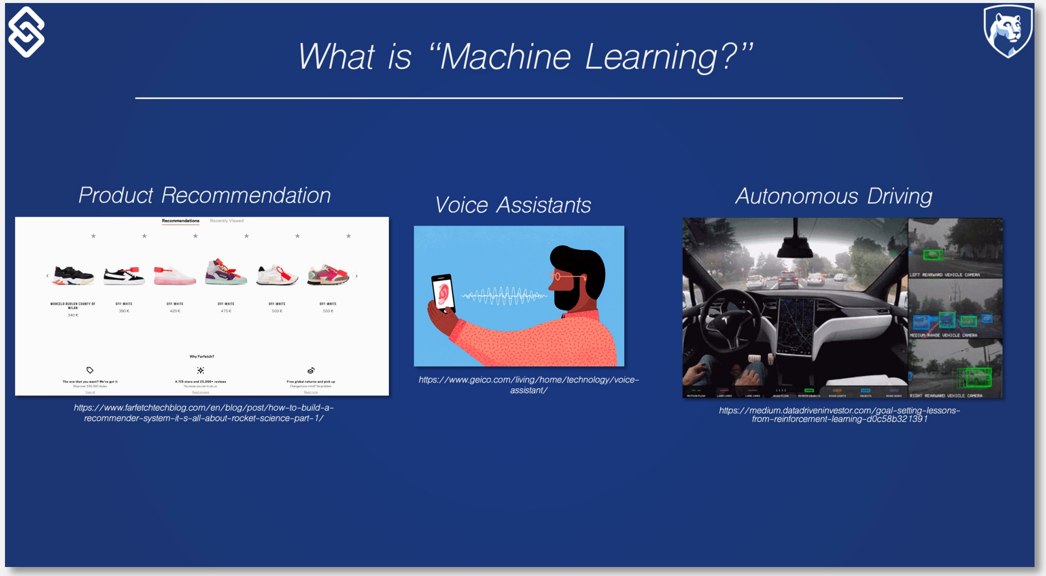


This tech is here to stay…

… and we'll get it wrong at first

Privacy

Machine Learning: The Bottom Line



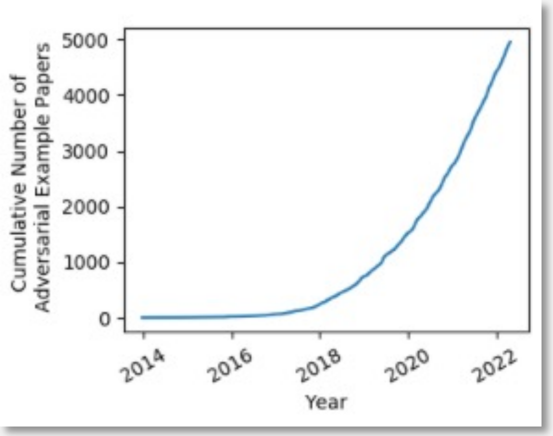
This tech is here to stay…

… and we'll get it wrong at first

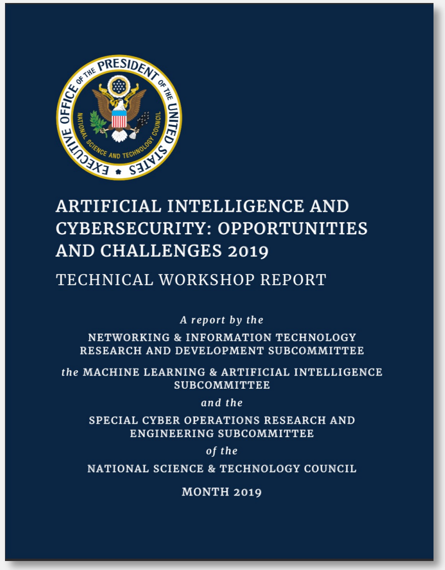
Fairness

Machine Learning: The Bottom Line

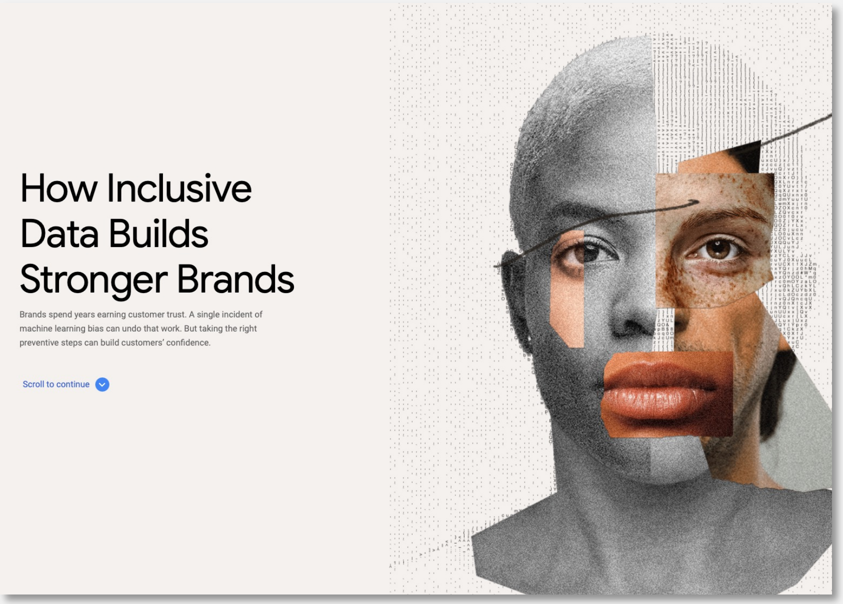
Academia



Policy



Awareness



It is our duty to take this by storm

Systems and Internet Infrastructure Security Laboratory

PennState

Enjoy your last week of the semester!

Ryan Sheatsley

sheatsley@psu.edu

https://sheatsley.me

@RyanSheatsley