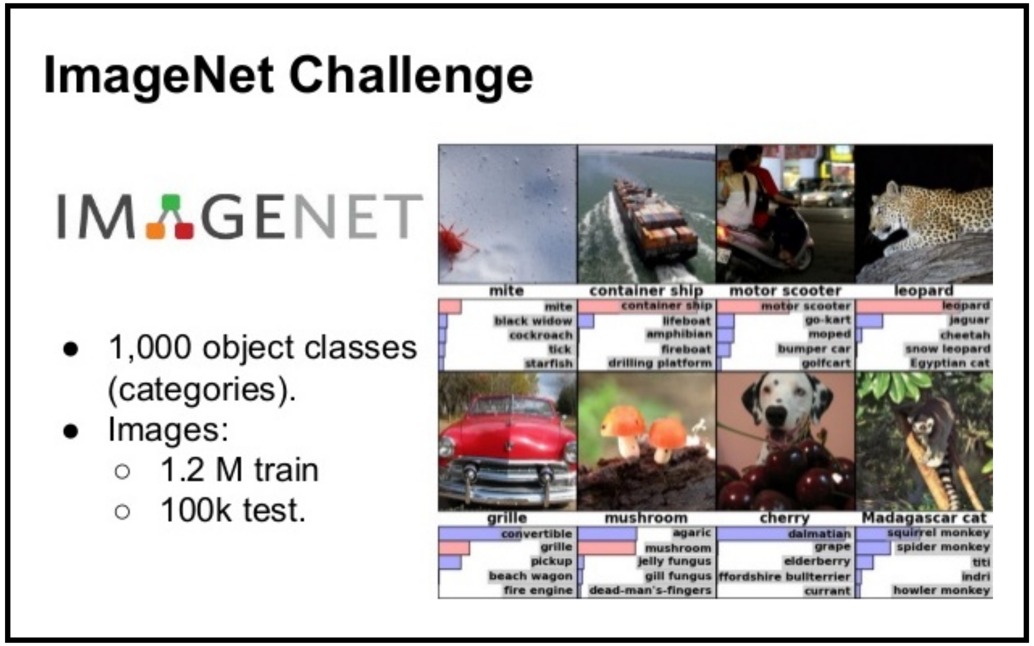
Adversarial Machine Learning *with* MLsploit

🌐 https://mlsploit.github.io/mlsploit-tutorial/

Nilaksh Das, Siwei Li, Chanil Jeon, Jinho Jung, Shang-Tse Chen, Carter Yagemann, Evan Downing, Haekyu Park, Evan Yang, Li Chen, Michael Kounavis, Ravi Sahita, David Durham, Scott Buck, Gokcen Cilingir, Polo Chau, Taesoo Kim, Wenke Lee

AI Advances in Recent Years

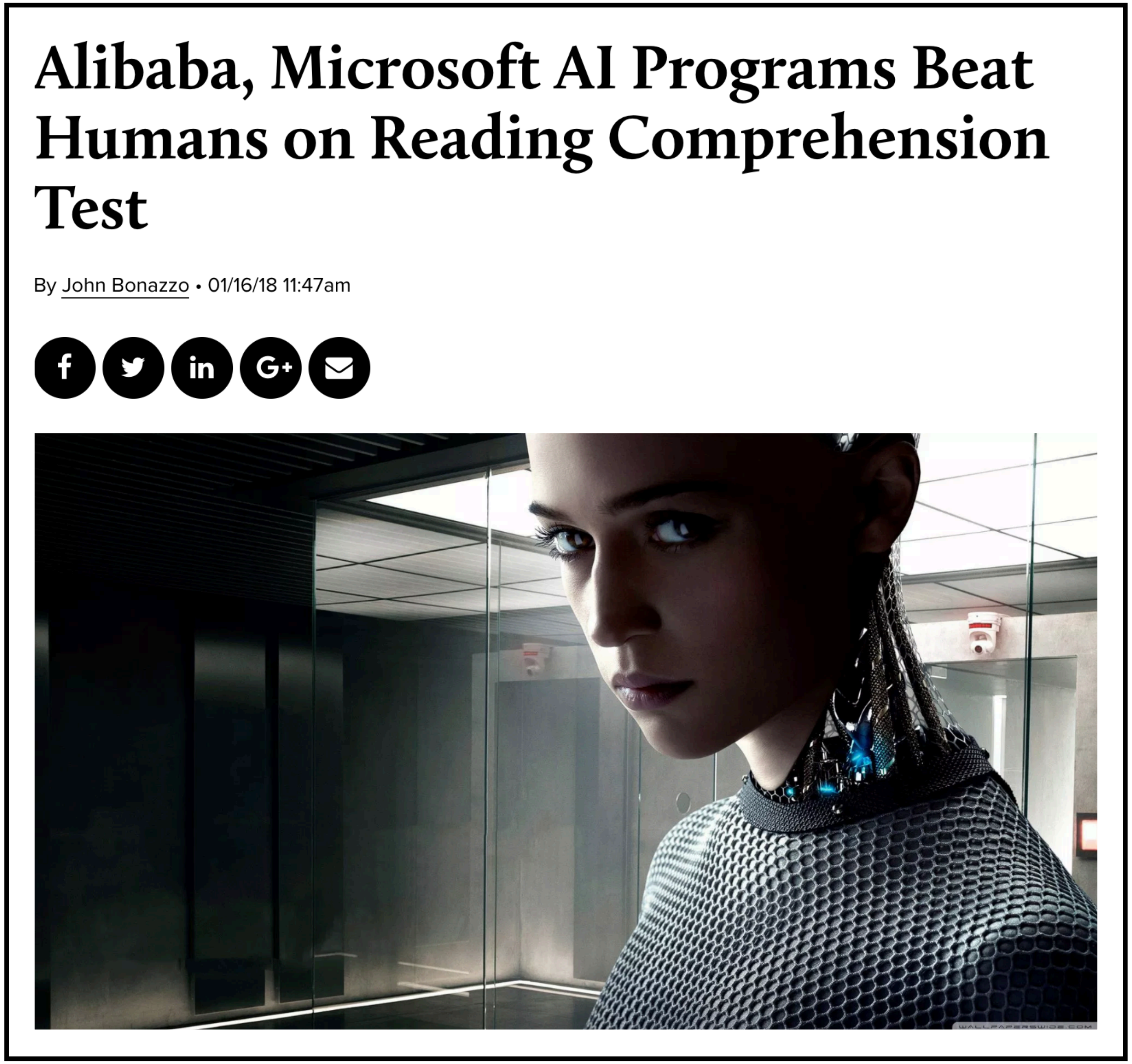


ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

THE ULTIMATE GO CHALLENGE
GAME 3 OF 3
27 MAY 2017
AlphaGo
Winner of Match 3
VS
Ke Jie
RESULT B + Res

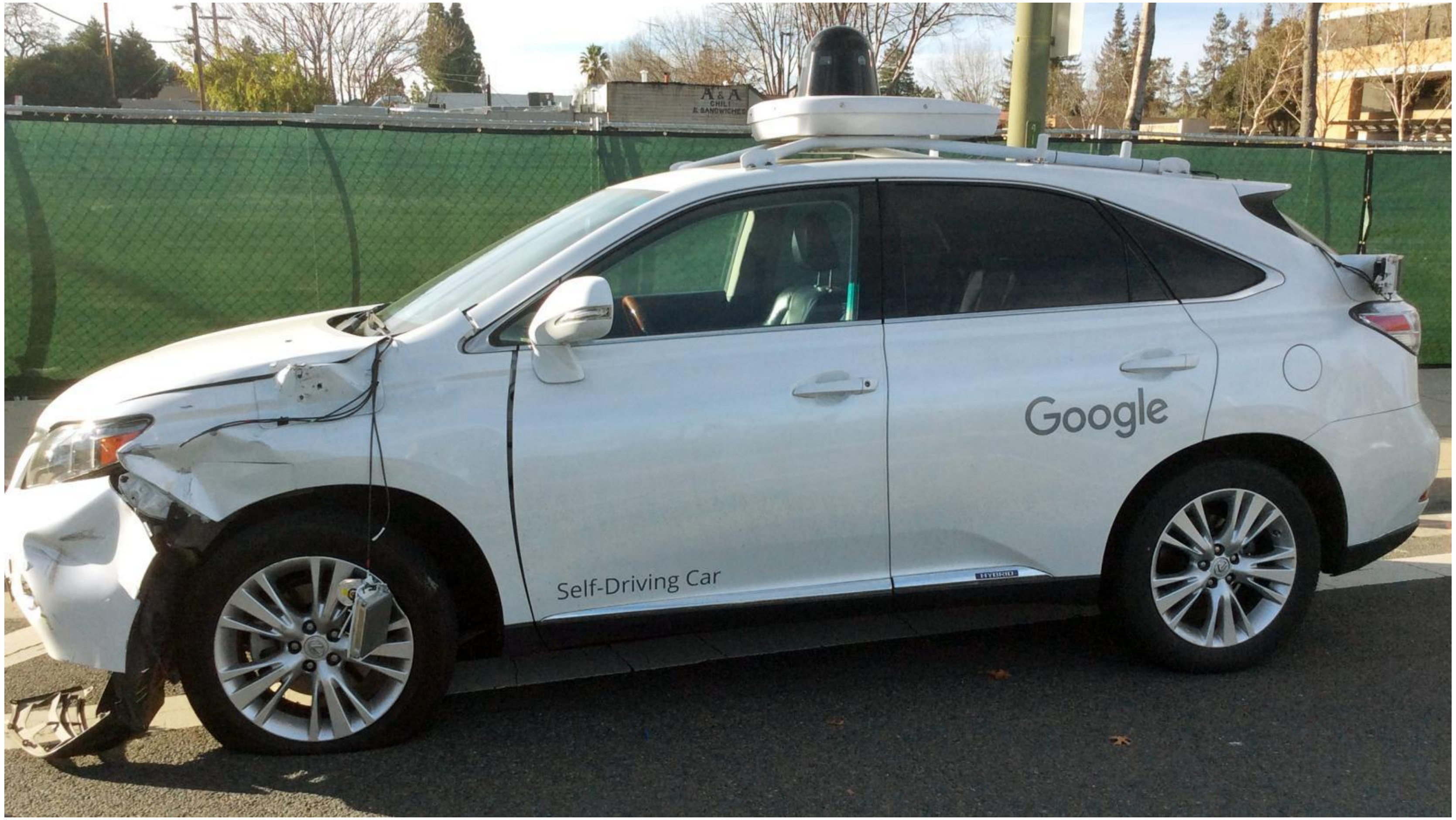


Alibaba, Microsoft AI Programs Beat Humans on Reading Comprehension Test

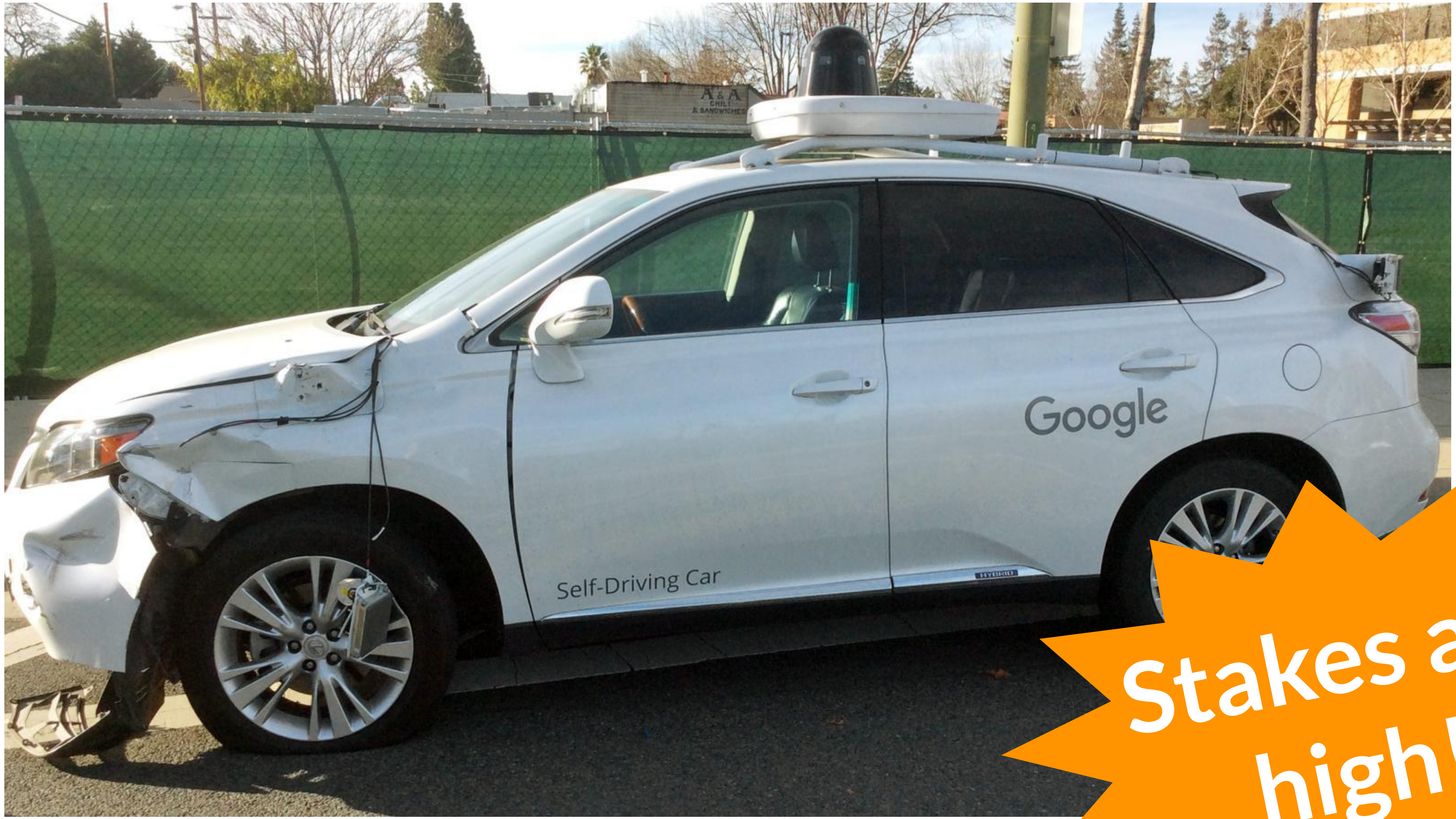
By John Bonazzo • 01/16/18 11:47am

Can we trust AI in real applications?

AI in Safety-Critical Applications



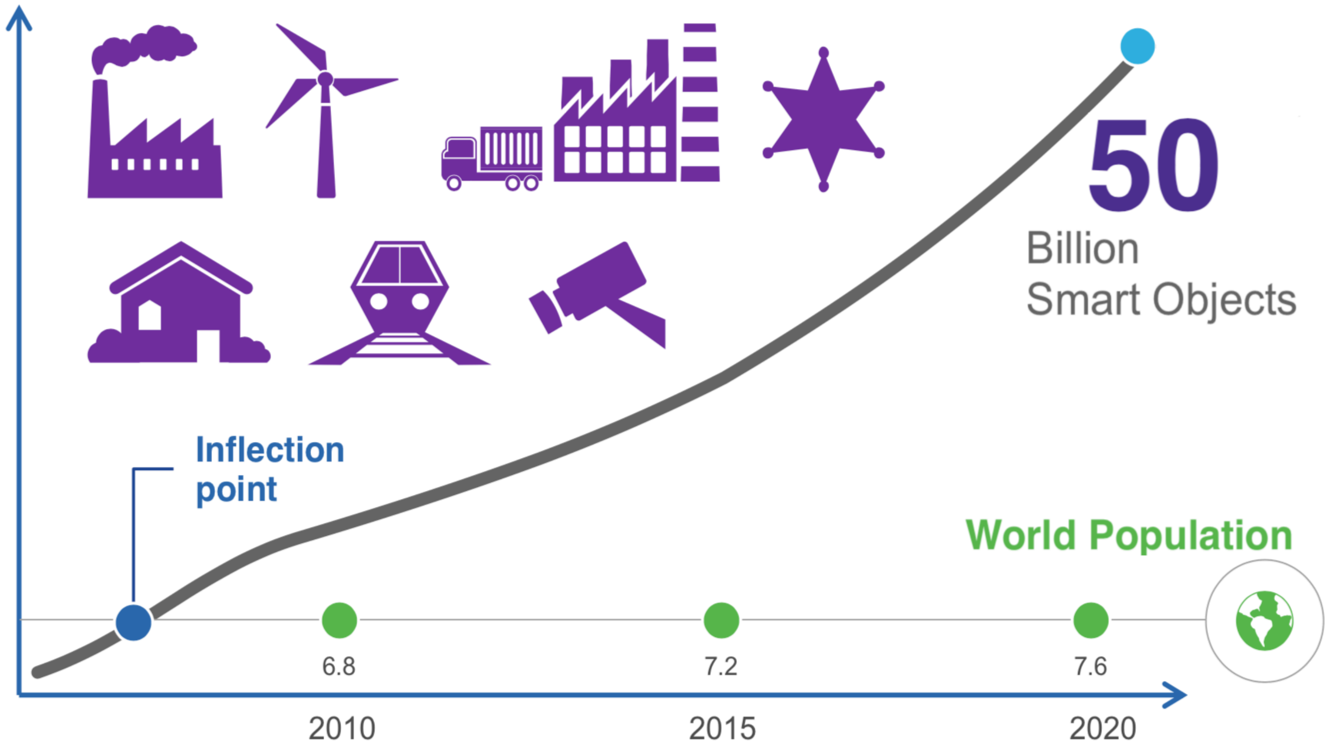
AI in Safety-Critical Applications



Stakes are high!

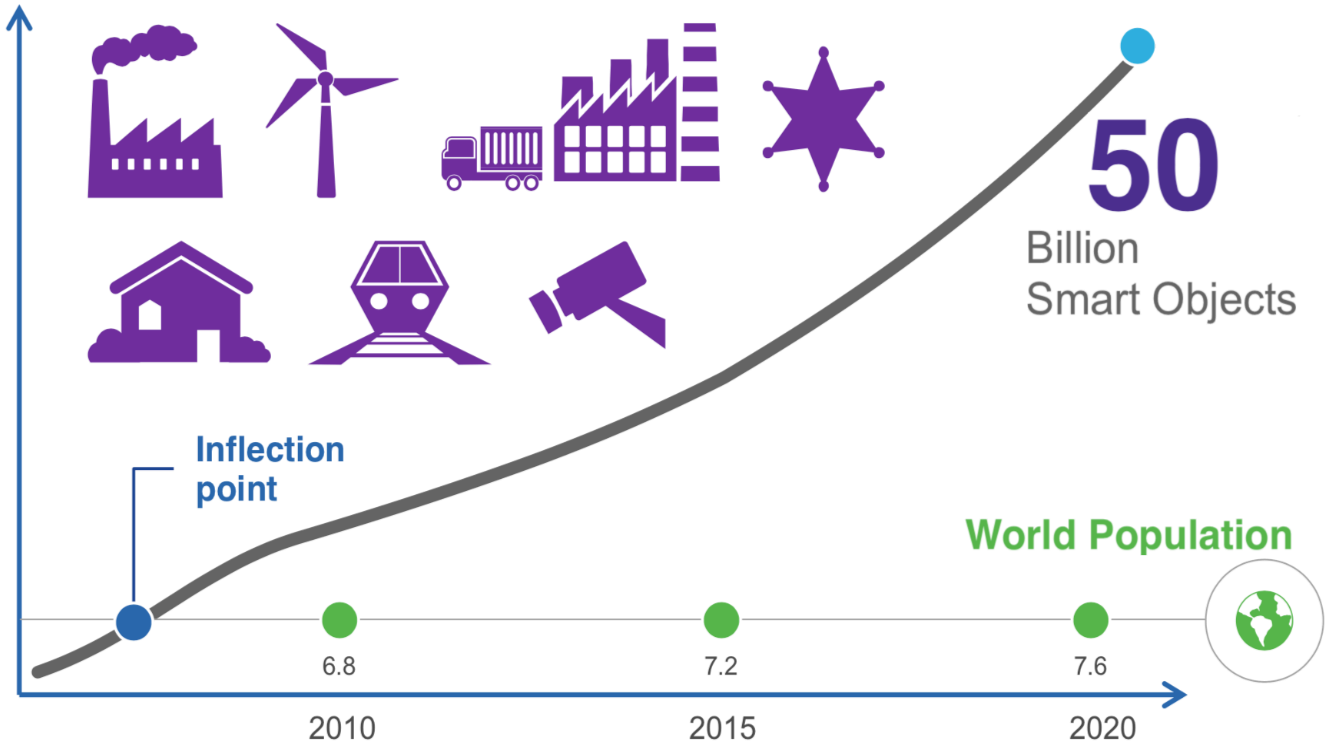
AI Security is becoming increasingly important

AI Security is becoming increasingly important



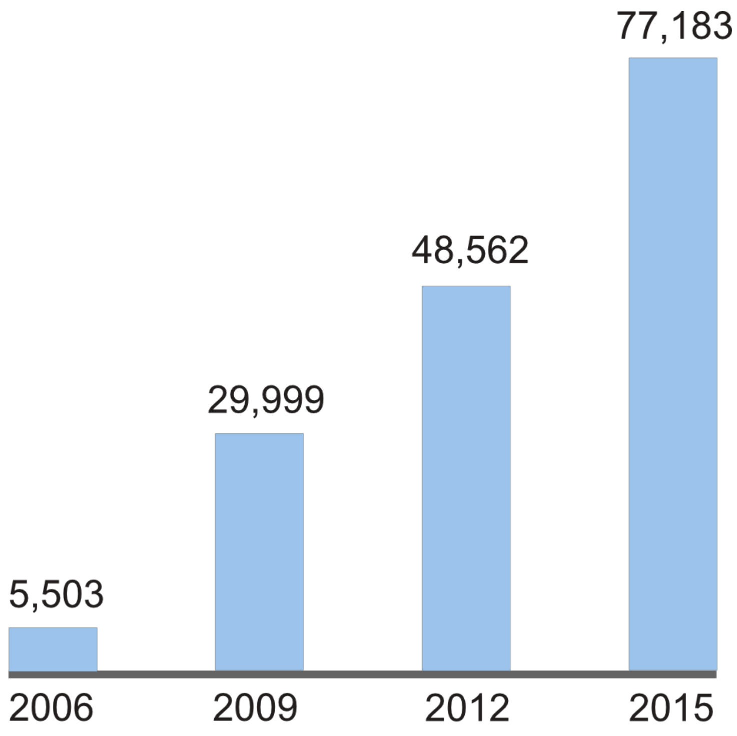
Source: Cisco

AI Security is becoming increasingly important



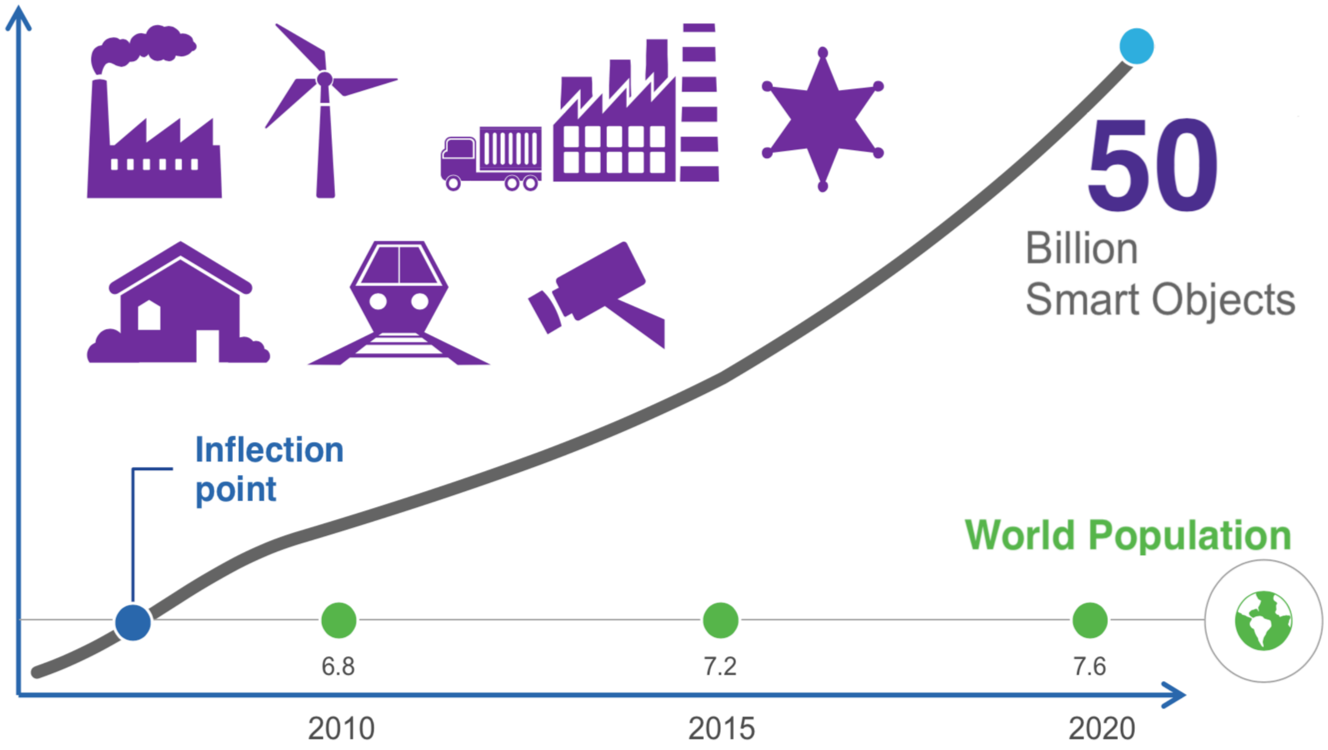
Source: Cisco

# incidents
reported by U.S. federal agencies

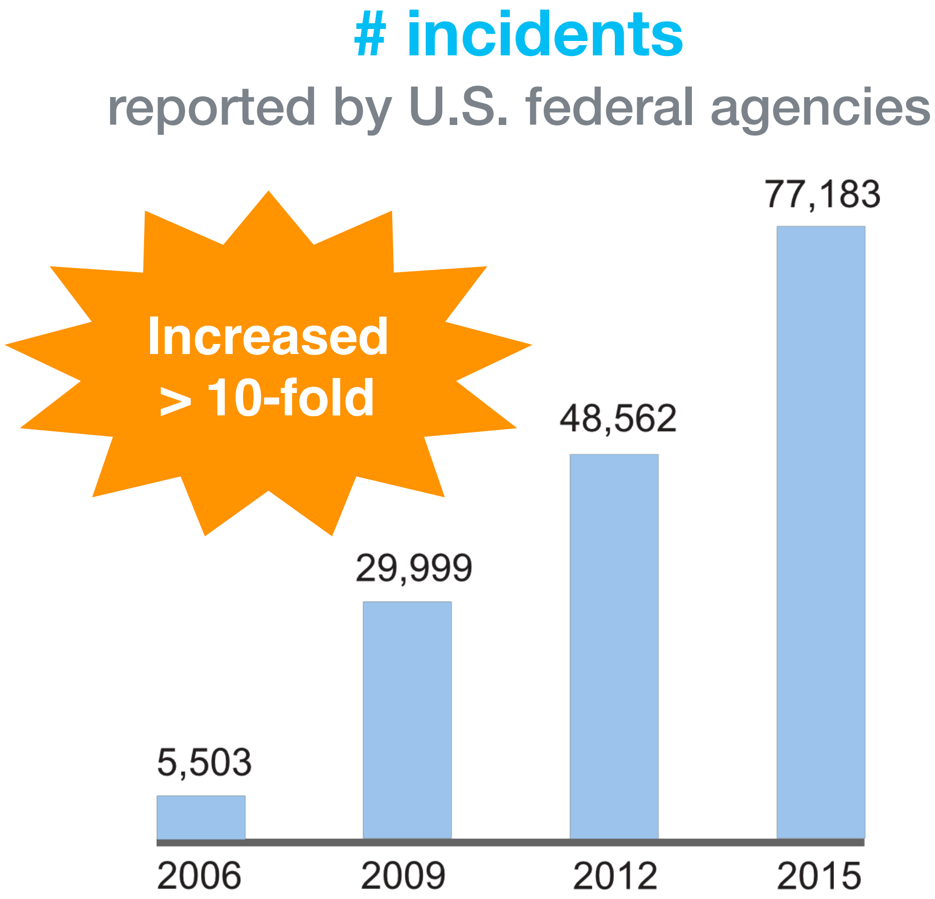


Source: US Department of Homeland Security

AI Security is becoming increasingly important



Source: Cisco

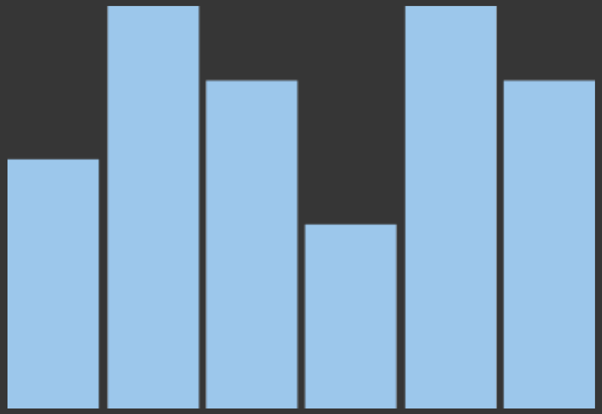


Source: US Department of Homeland Security

MLsploit Goal

Study ML vulnerabilities and develop secure AI for high-stakes problems

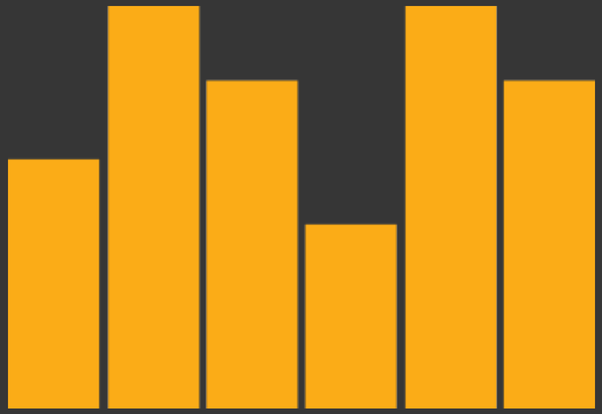
When and why does ML fail?



Training Data

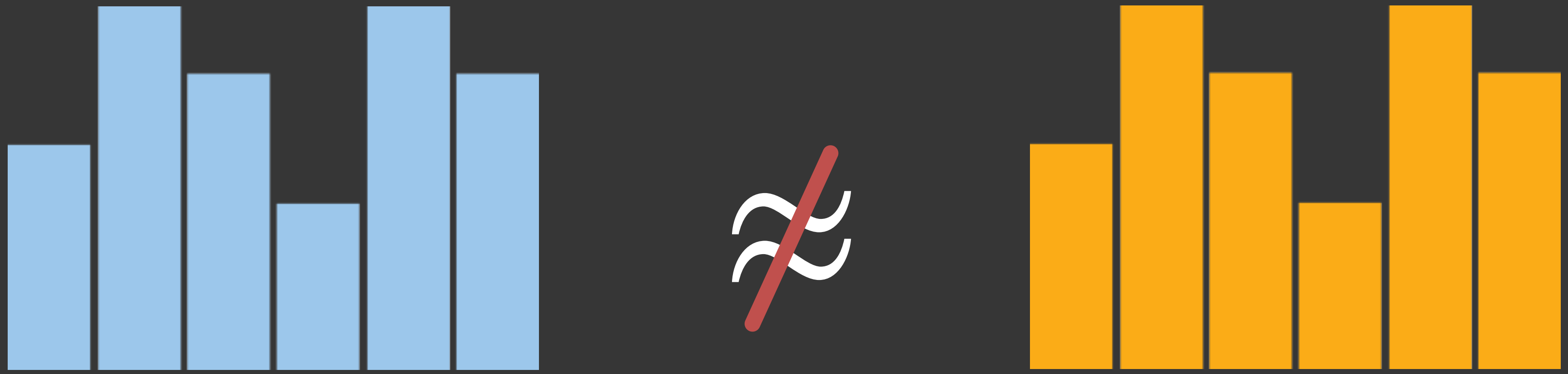
Common assumption

≈



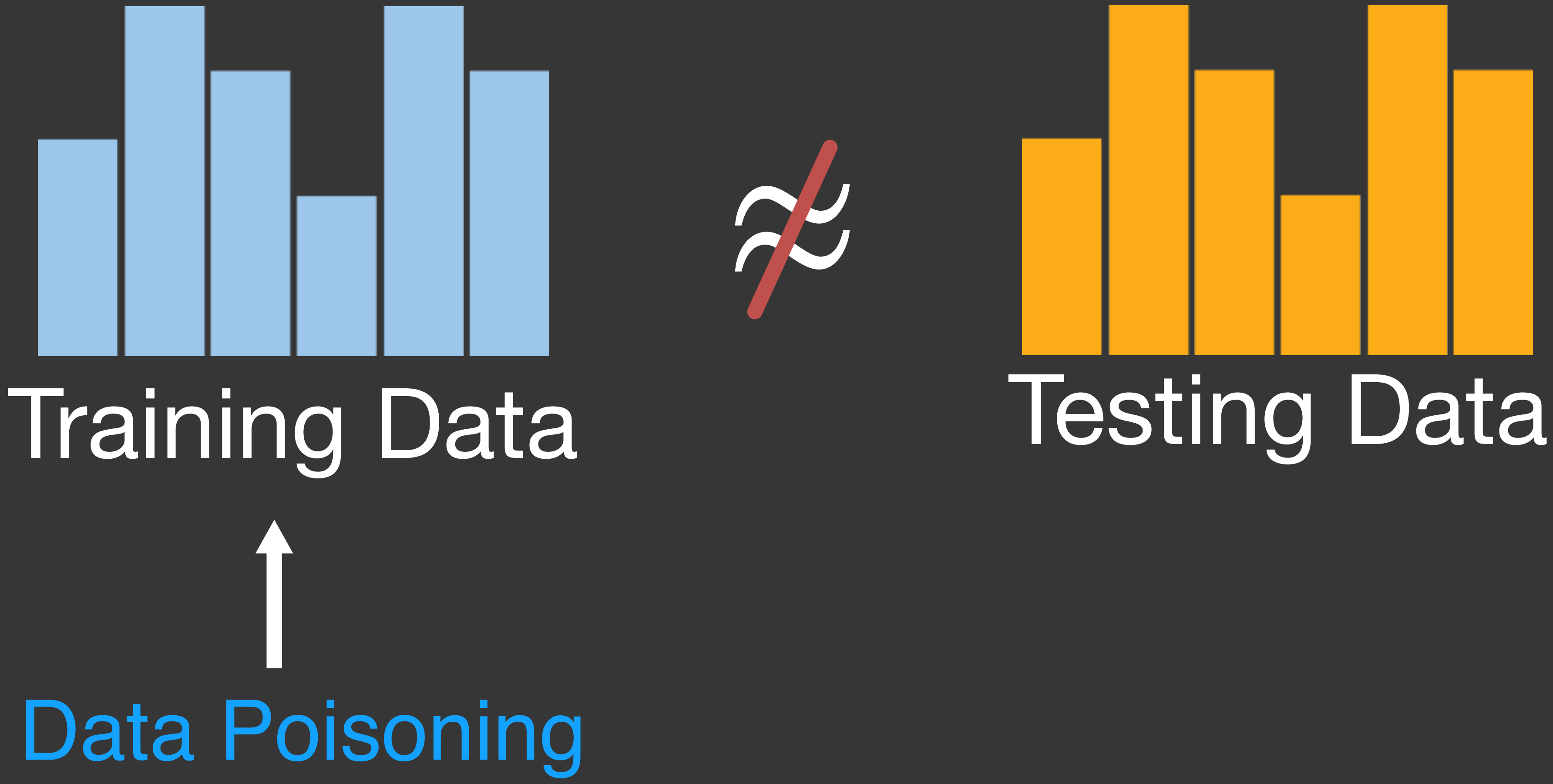
Testing Data

When and why does ML fail?



Training Data ≉ Testing Data

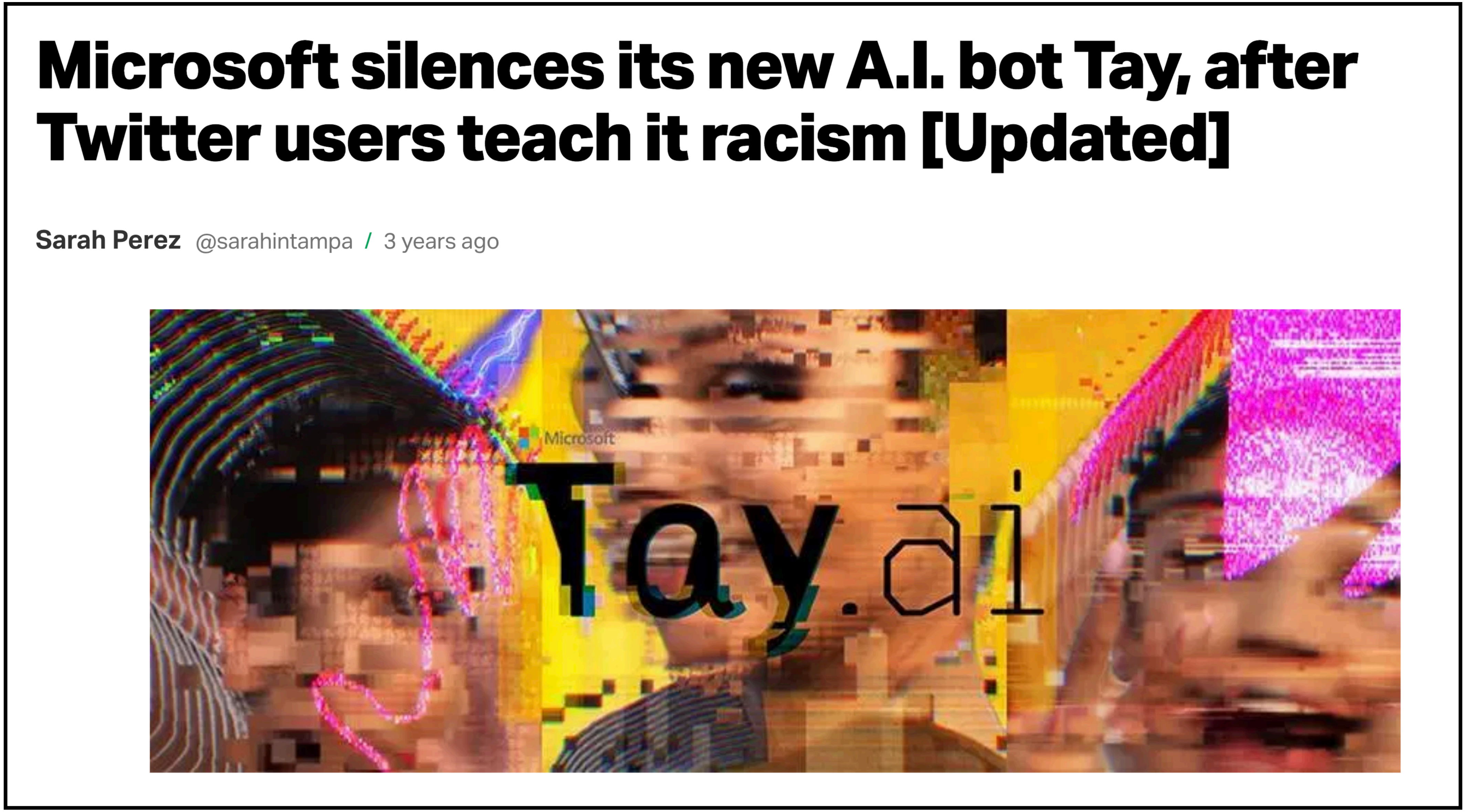
When and why does ML fail?



Training Data ≉ Testing Data

Data Poisoning

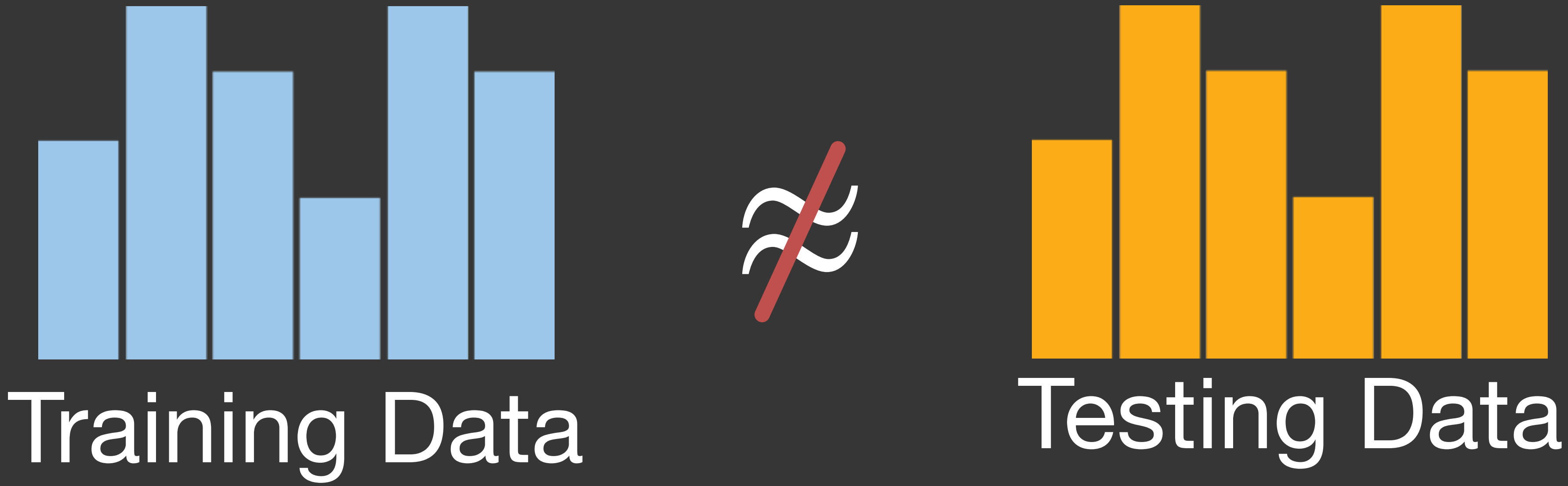
Data Poisoning in Real World



Microsoft silences its new A.I. bot Tay, after Twitter users teach it racism [Updated]

Sarah Perez @sarahintampa / 3 years ago

When and why does ML fail?



Training Data ≉ Testing Data

When and why does ML fail?

Training Data ≉ Testing Data

↑

Adversarial Examples

Adversarial Examples

Input Image

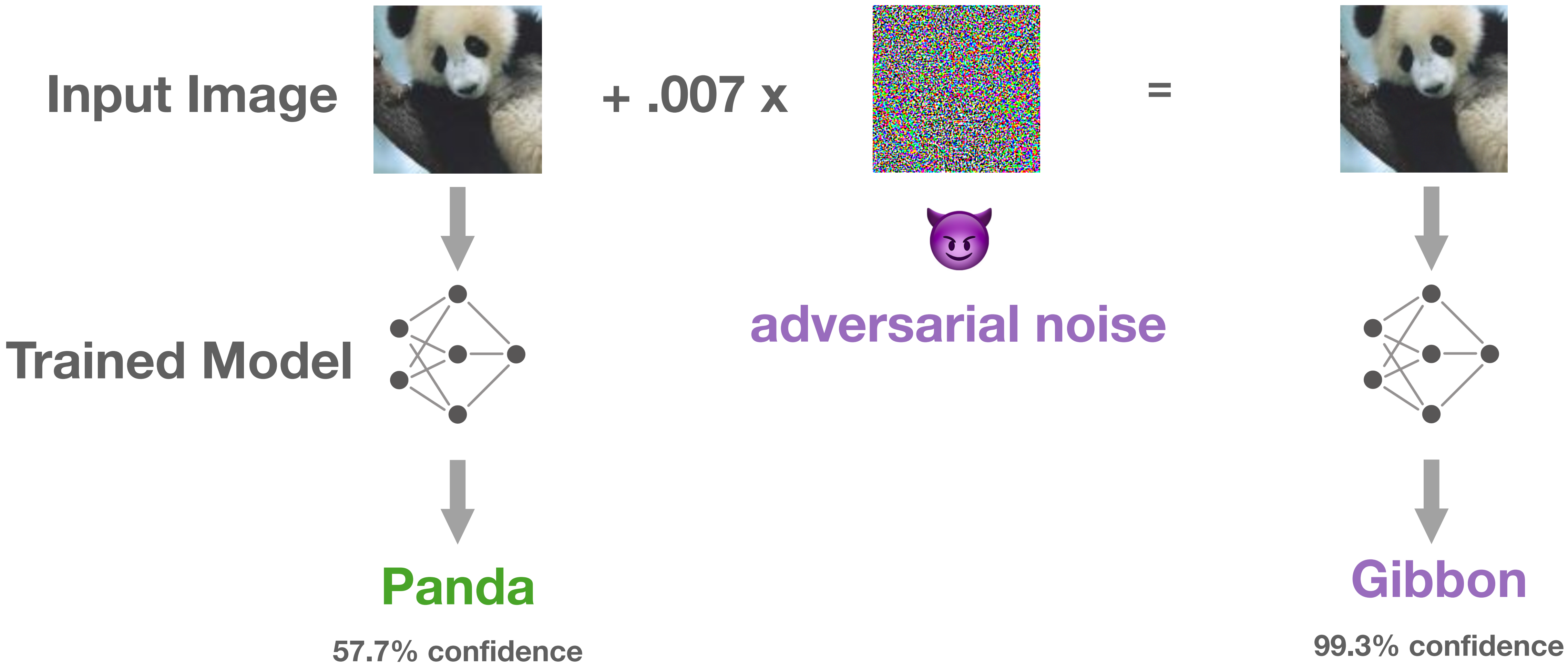
Trained Model

Panda

57.7% confidence

[Goodfellow et al. ICLR 2015]

Adversarial Examples



Input Image

+ .007 x

=

Trained Model

😈

adversarial noise

Panda

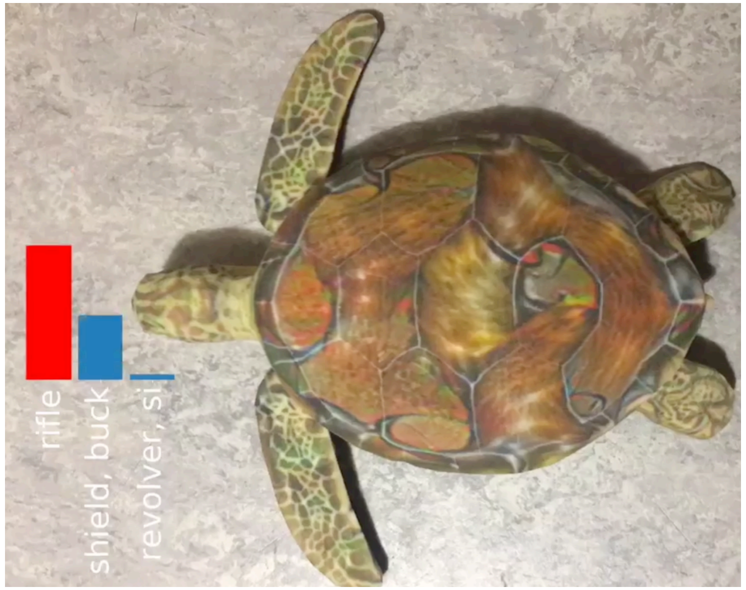
57.7% confidence

Gibbon

99.3% confidence

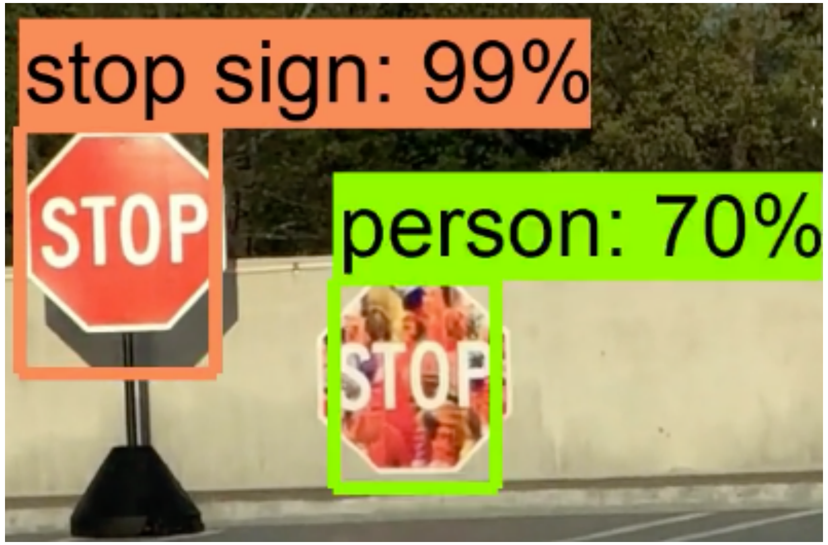
[Goodfellow et al. ICLR 2015]

Why is Adversarial Example a Threat?



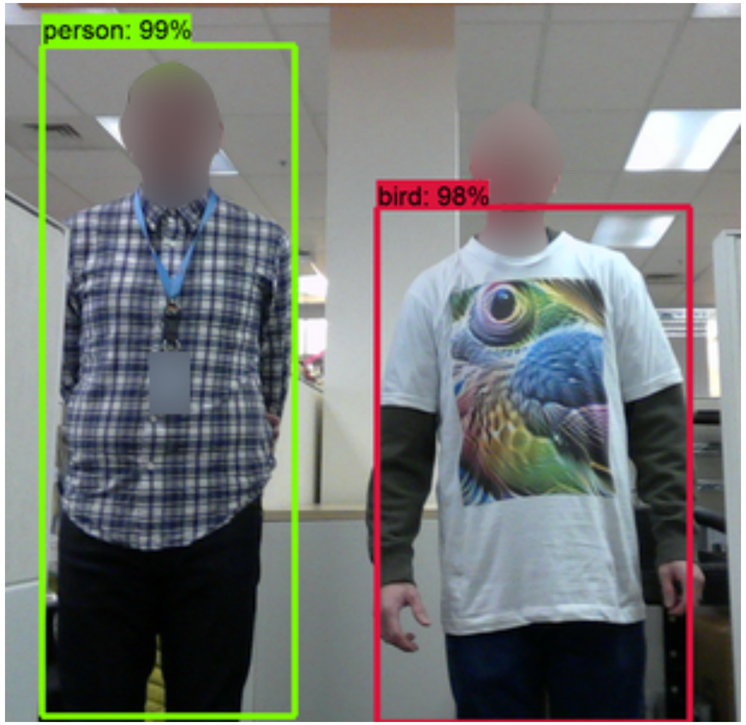
3D-printed object that fools an image classifier

[Athalye et al. ICML’18]



Physical stop sign that fools traffic sign recognition

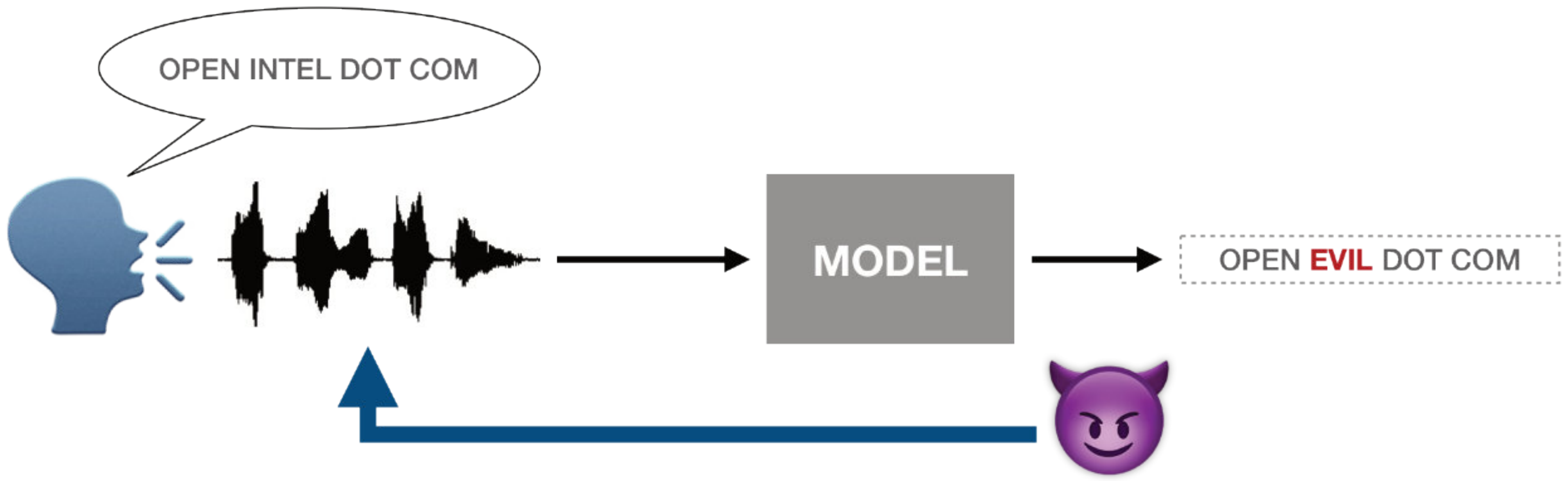
[Chen et al. ECML-PKDD’18]



Physical t-shirt that fools security camera

[Cornelius et al. DSML’19]

Adversarial Examples Beyond Vision



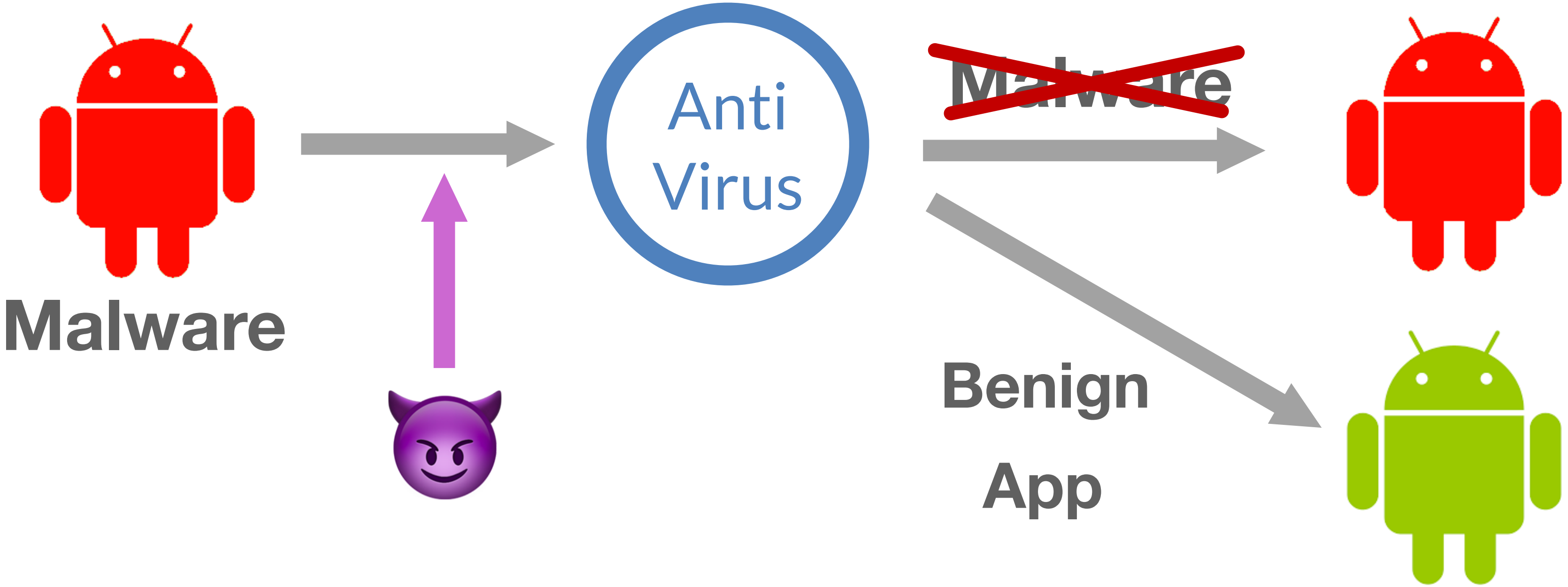
OPEN INTEL DOT COM

MODEL

OPEN EVIL DOT COM

Audio Attack

[Carlini & Wagner. DLS 2018]



Malware

Anti Virus

~~Malware~~

Benign App

Android Malware

[Jung et al. Black Hat 2017]

MLsploit

github.com/mlsploit

A Framework for Interactive Experimentation with Adversarial Machine Learning Research

Contributors from *Intel Science and Technology Center for Adversary-Resilient Security Analytics*:
Nilaksh Das, Siwei Li, Chanil Jeon, Jinho Jung*, Shang-Tse Chen*, Carter Yagemann*, Evan Downing*, Haekyu Park, Evan Yang, Li Chen, Michael Kounavis, Ravi Sahita, David Durham, Scott Buck, Polo Chau, Taesoo Kim, Wenke Lee
(*equal contribution)

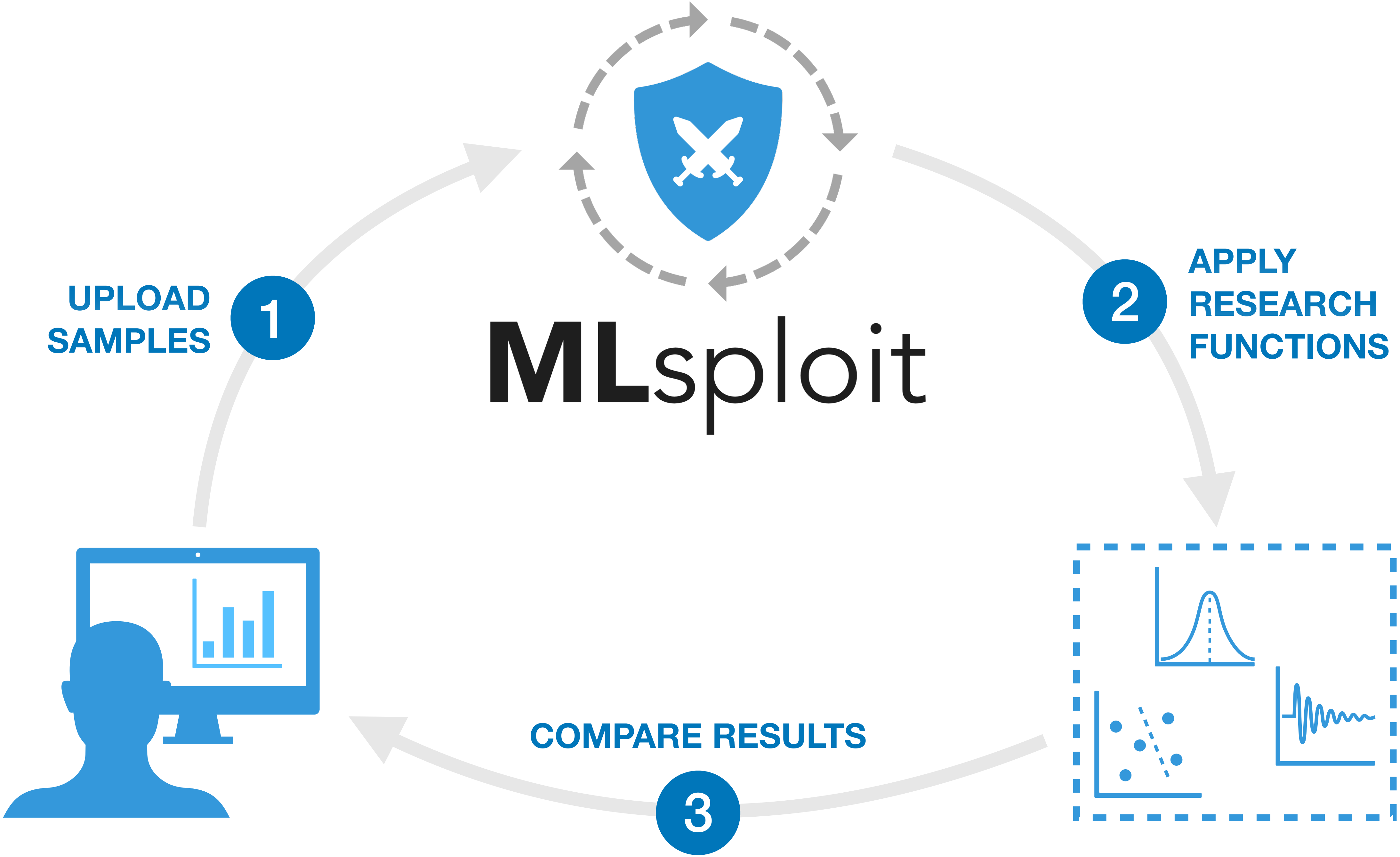
MLsploit

- ★ **Research modules** for adversarial ML
  - ✳ Enables **comparison** of attacks and defenses
- ★ **Interactive experimentation** with ML research
- ★ Researchers can **easily integrate** novel research into an intuitive and seamless **user interface**

MLsploit

- **AVPass** (leaking and bypassing Android malware detection systems)
- **ELF** (bypassing Linux malware detection with API perturbation)
- **PE** (create and attack ML models for detecting Windows PE malware)
- **Intel®-Software Guard Extensions**

  (privacy preserving adversarial ML as a service)
- **SHIELD** (attack and defend state-of-the-art image classification models)
  - Attacks: **FGSM, DeepFool, Carlini-Wagner**
  - Defenses: **SLQ, JPEG, Median Filter, TV-Bregman**



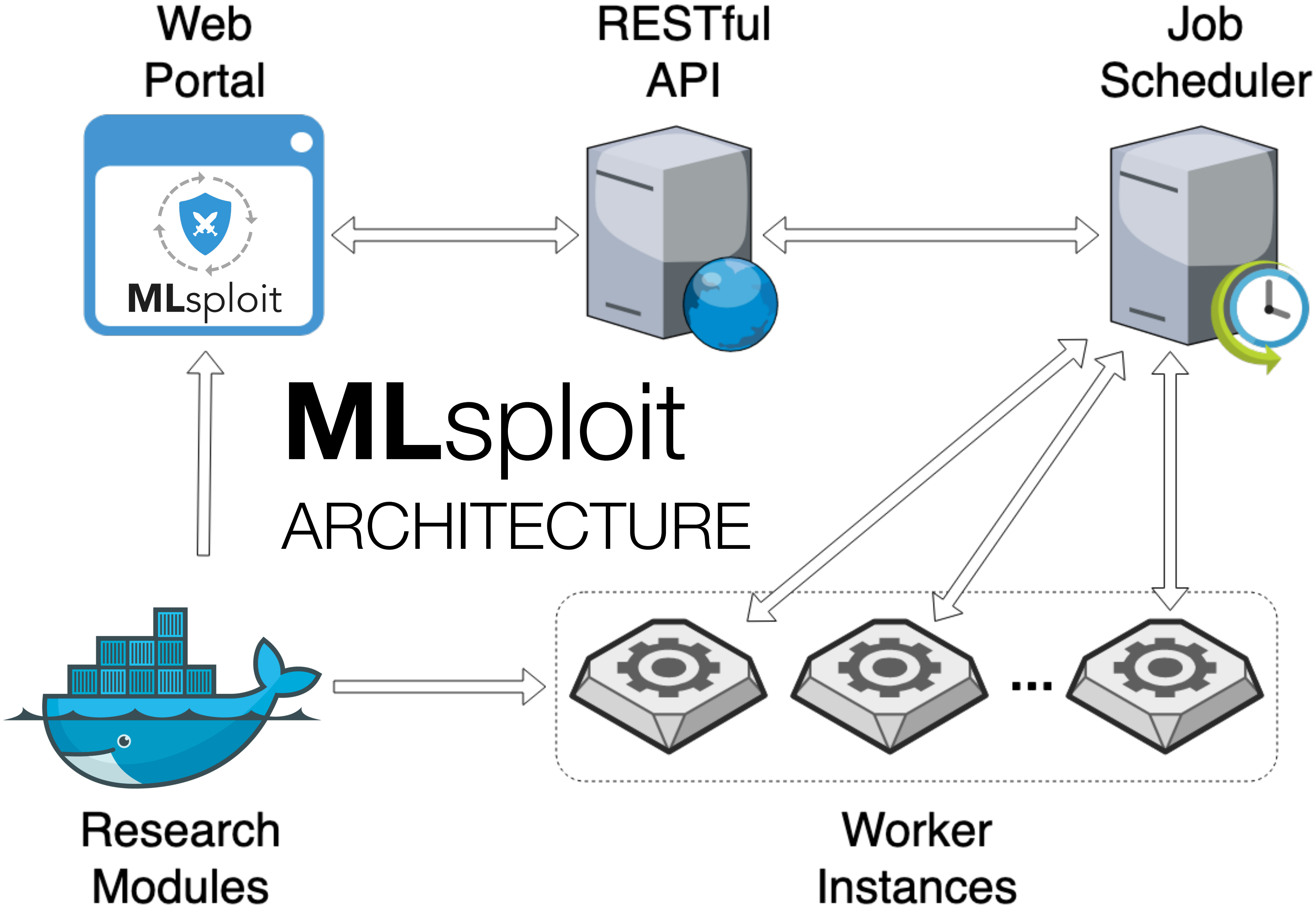
MLsploit

1 UPLOAD SAMPLES

2 APPLY RESEARCH FUNCTIONS

3 COMPARE RESULTS

MLsploit ARCHITECTURE



Web Portal

MLsploit

RESTful API

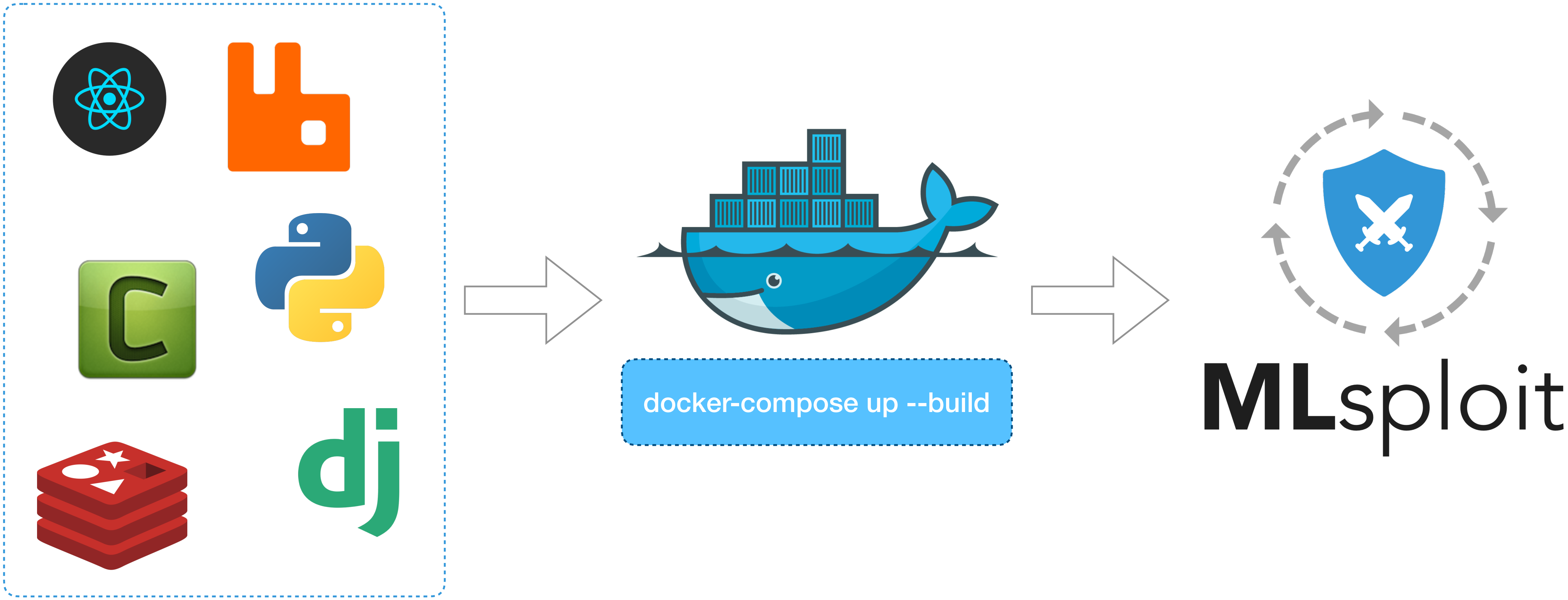
Job Scheduler

Research Modules

...

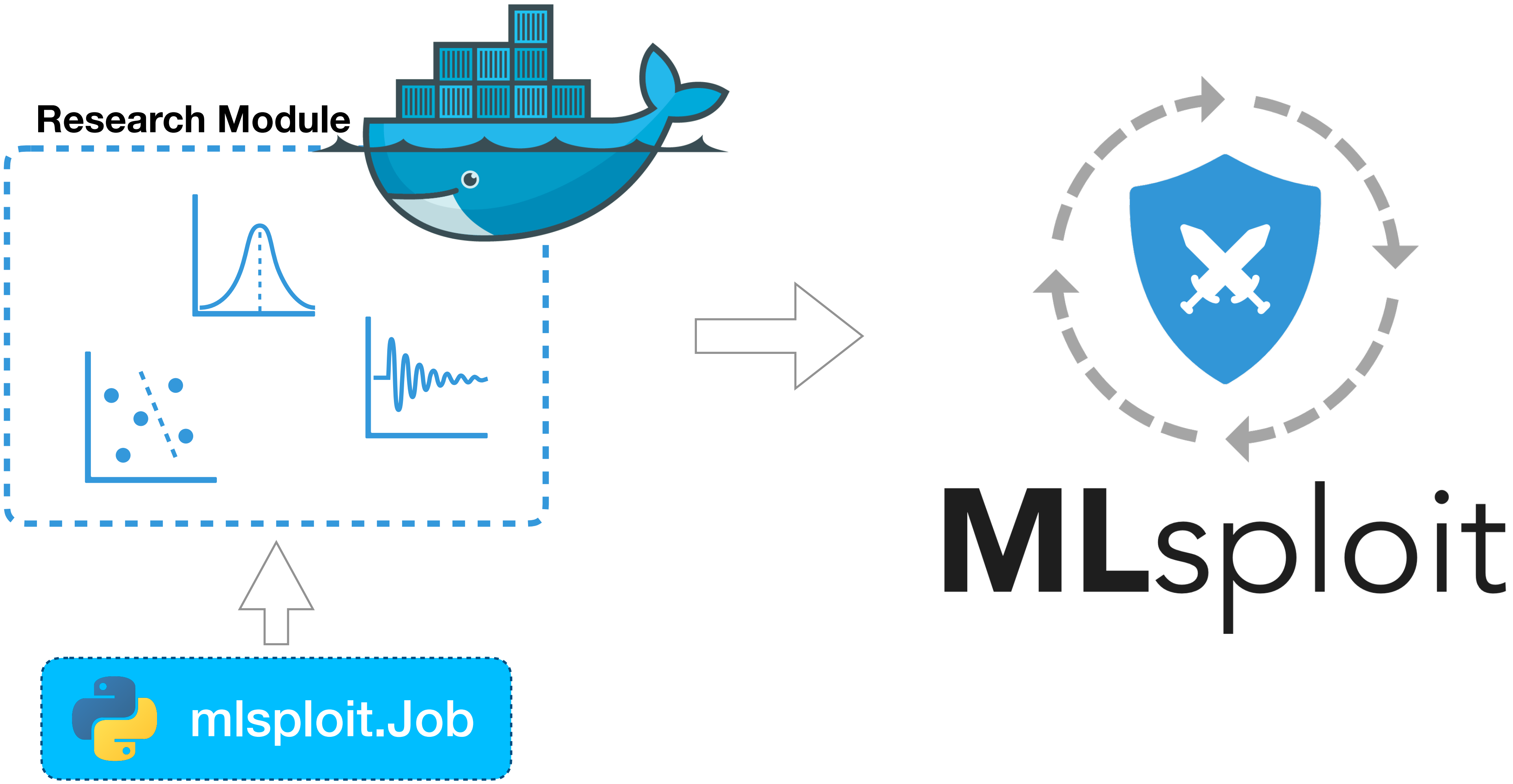
Worker Instances

ONE-STEP INSTALLATION



```
docker-compose up --build
```

EASY INTEGRATION OF RESEARCH

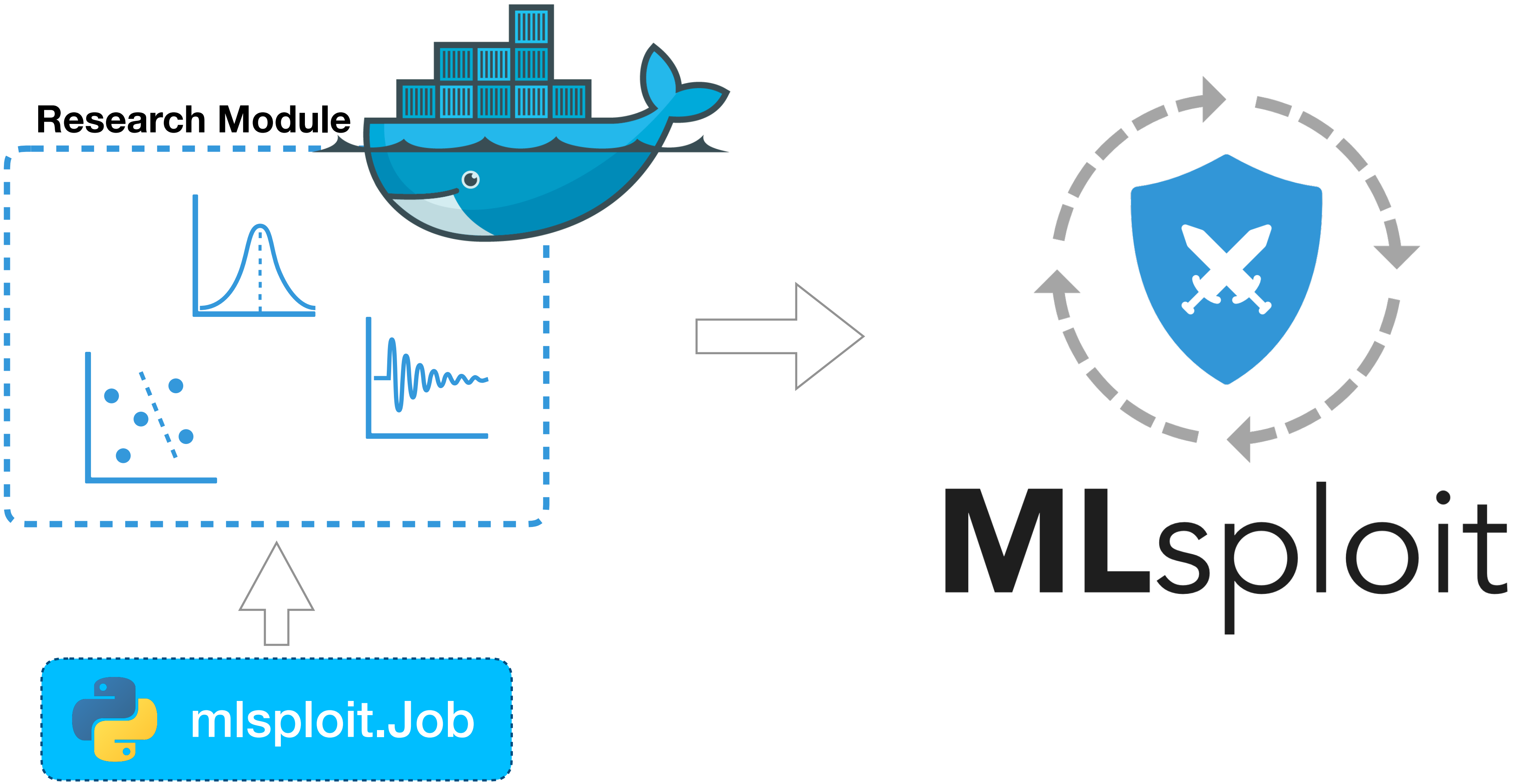


Research Module

mlsploit.Job

MLsploit

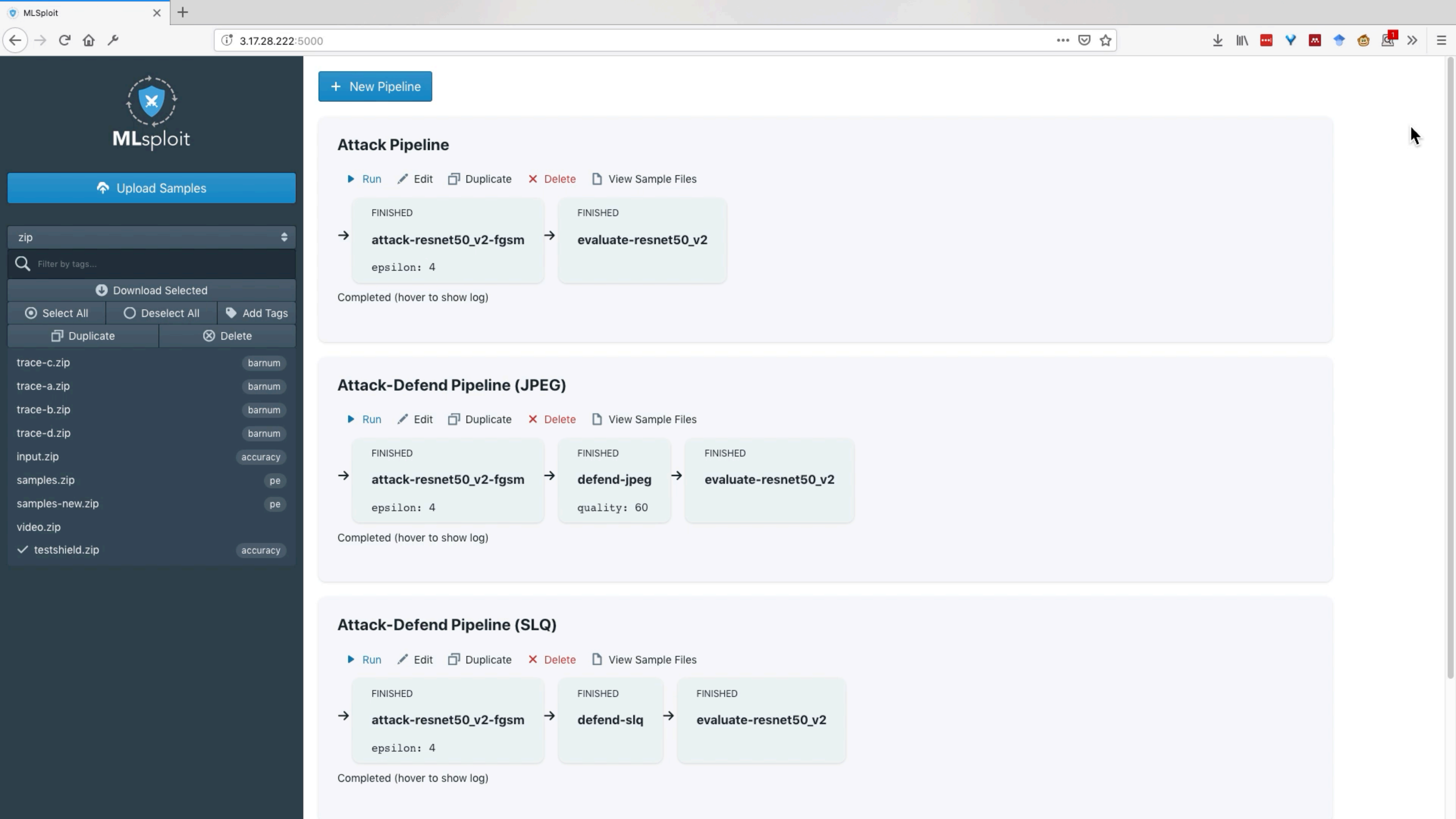
EASY INTEGRATION OF RESEARCH



Research Module

mlsploit.Job

MLsploit



trace-c.zip barnum
trace-a.zip barnum
trace-b.zip barnum
trace-d.zip barnum
input.zip accuracy
samples.zip pe
samples-new.zip pe
video.zip
✓ testshield.zip accuracy

Attack Pipeline

→ FINISHED **attack-resnet50_v2-fgsm** `epsilon: 4` → FINISHED **evaluate-resnet50_v2**

Completed (hover to show log)

Attack-Defend Pipeline (JPEG)

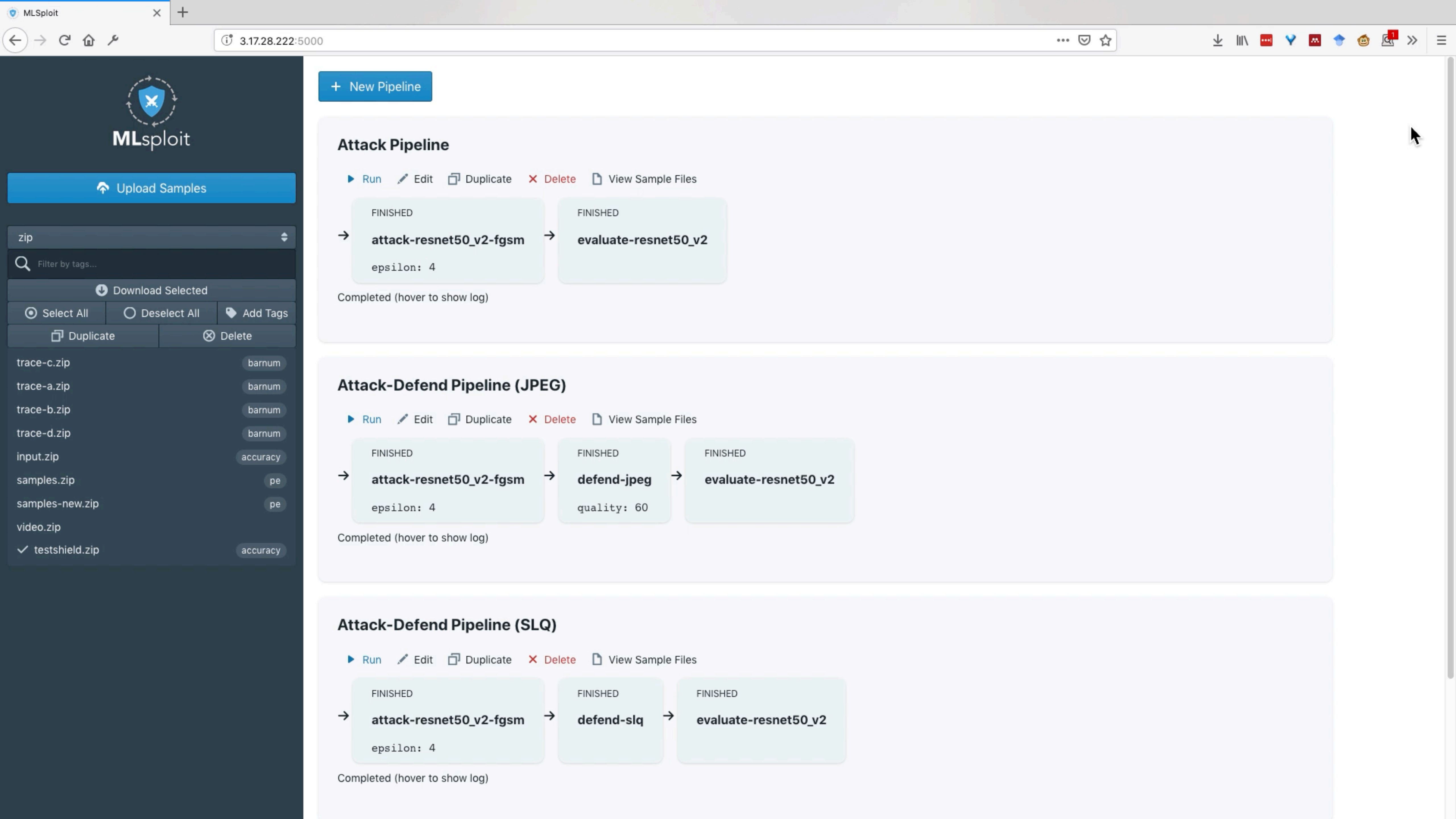
→ FINISHED **attack-resnet50_v2-fgsm** `epsilon: 4` → FINISHED **defend-jpeg** `quality: 60` → FINISHED **evaluate-resnet50_v2**

Completed (hover to show log)

Attack-Defend Pipeline (SLQ)

→ FINISHED **attack-resnet50_v2-fgsm** `epsilon: 4` → FINISHED **defend-slq** → FINISHED **evaluate-resnet50_v2**

Completed (hover to show log)



trace-c.zip	barnum
trace-a.zip	barnum
trace-b.zip	barnum
trace-d.zip	barnum
input.zip	accuracy
samples.zip	pe
samples-new.zip	pe
video.zip	
✓ testshield.zip	accuracy

Attack Pipeline

→ FINISHED attack-resnet50_v2-fgsm `epsilon: 4` → FINISHED evaluate-resnet50_v2

Completed (hover to show log)

Attack-Defend Pipeline (JPEG)

→ FINISHED attack-resnet50_v2-fgsm `epsilon: 4` → FINISHED defend-jpeg `quality: 60` → FINISHED evaluate-resnet50_v2

Completed (hover to show log)

Attack-Defend Pipeline (SLQ)

→ FINISHED attack-resnet50_v2-fgsm `epsilon: 4` → FINISHED defend-slq → FINISHED evaluate-resnet50_v2

Completed (hover to show log)

Georgia Tech

intel

MLsploit

github.com/mlsploit

A Framework for Interactive Experimentation with Adversarial Machine Learning Research

Contributors from *Intel Science and Technology Center for Adversary-Resilient Security Analytics*:
Nilaksh Das, Siwei Li, Chanil Jeon, Jinho Jung*, Shang-Tse Chen*, Carter Yagemann*, Evan Downing*, Haekyu Park, Evan Yang, Li Chen, Michael Kounavis, Ravi Sahita, David Durham, Scott Buck, Polo Chau, Taesoo Kim, Wenke Lee
(*equal contribution)