Institute for Network and Security Research

Panel: Machine Learning and Security (and Privacy)

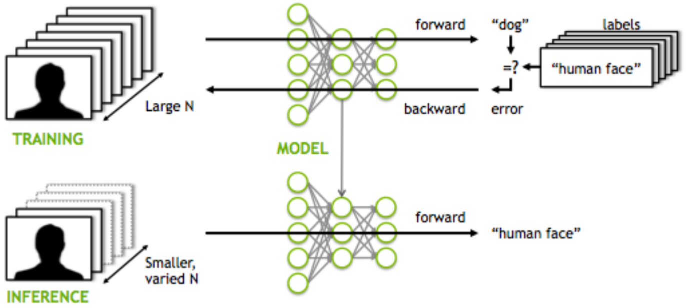
Arlington, VA – January 9-11, 2017

Panelists: David Evans (UVA), Ian Goodfellow (OpenAI), Nicolas Papernot (PSU), Dawn Song (UCB), Michael Wellman (Umich)

Machine Learning

- Perhaps no area of computer science has had more impact on systems and society in the last 5 years than machine learning
  - Analytics
  - Autonomous systems
  - Vision …



Panel: Security and Privacy

- Challenge: what are the security and privacy challenges of the use of machine learning in adversarial settings?
  - Fundamental science:
    - What are the limits of machine learning with respect to accuracy and resilience?
    - What vulnerabilities are general vs. those are a consequence of the techniques used?
    - Can the advantages of ML be realized while preserving privacy?
  - Applied science
    - What countermeasures are likely to be effective in practice?
    - What are the domain specific challenges and safeguards for security and privacy?
    - Ethics:
      - Just because a system may be able to understand environment, should it?
      - Can the advantages of ML be realized fairly without discriminating minorities?
    - Education (what and how to integrate security into machine learning/security courses)

Panelists

- David Evans (UVA)
- Ian Goodfellow (OpenAI)
- Nicolas Papernot (PSU)
- Dawn Song (UCB)
- Michael Wellman (UMich)

The Magic of Machine Learning Isn’t Magic

NSF SaTC Pis Meeting 2017
10 January 2017

David Evans
https://www.cs.virginia.edu/evans

Machine Learning Does Amazing Things!

The Unreasonable Effectiveness of Data

Alon Halevy, Peter Norvig, and Fernando Pereira, *Google*

a cat is sitting on a toilet seat
logprob: -7.79

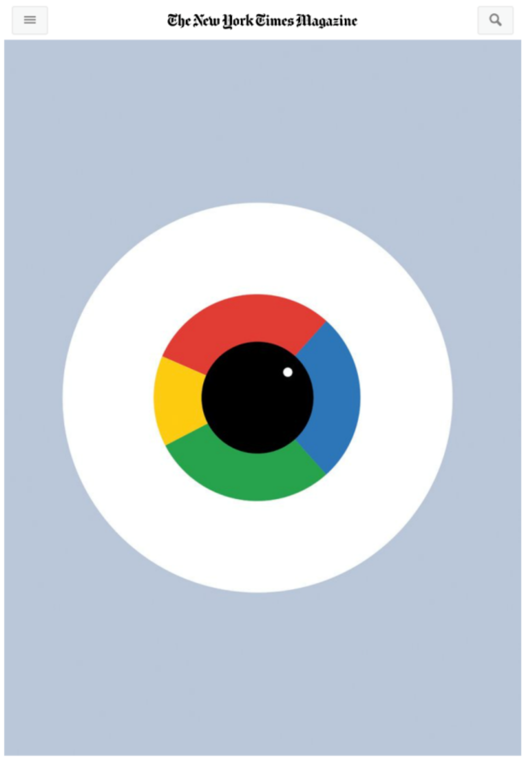
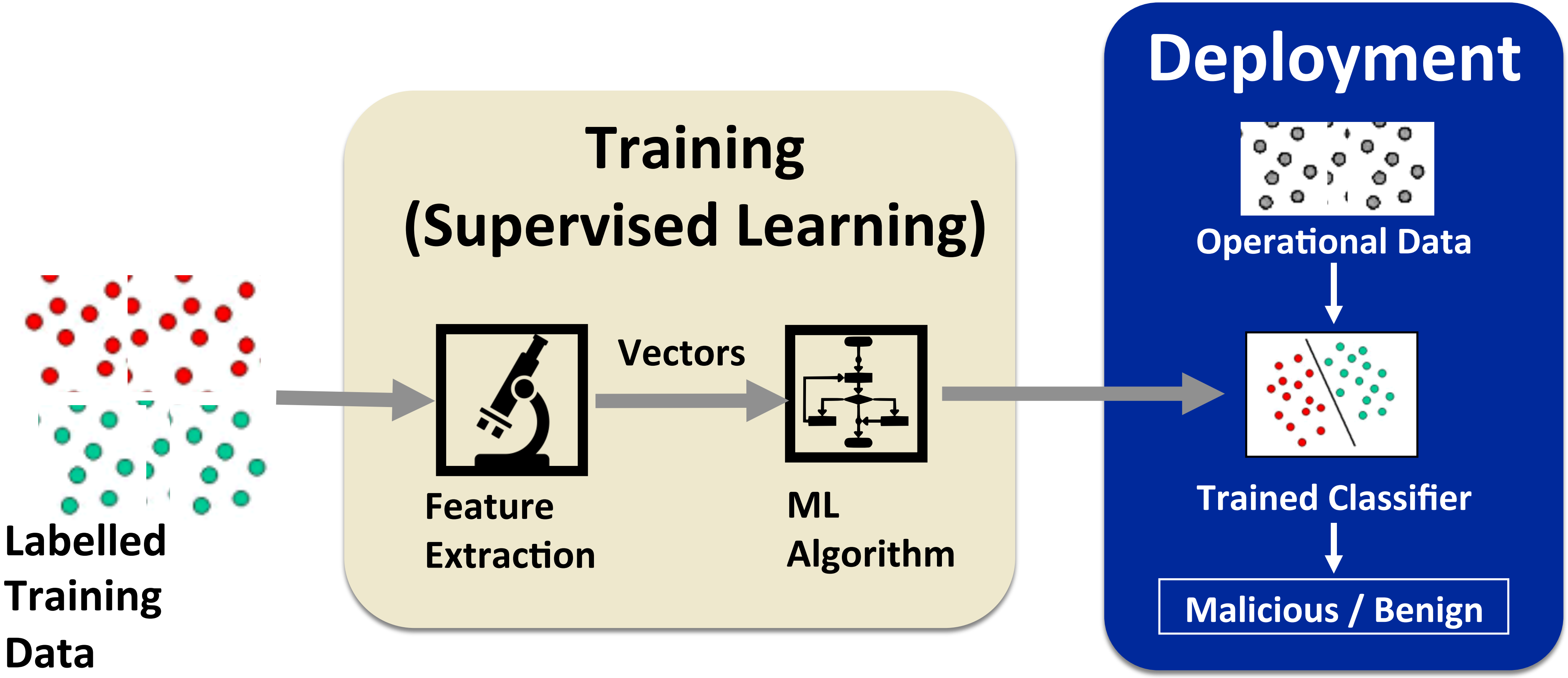


ILLUSTRATION BY PABLO DELCAN

The Great A.I. Awakening

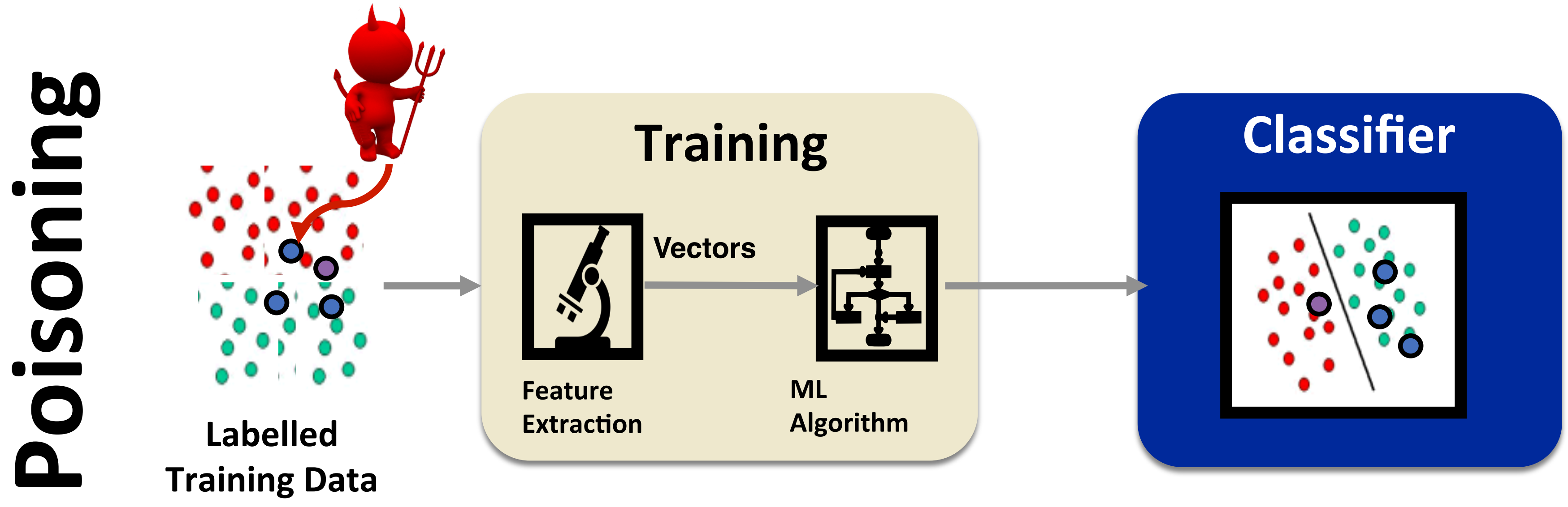
How Google used artificial intelligence to transform Google Translate, one of its more popular services — and how machine learning is poised to reinvent computing itself.

By GIDEON LEWIS-KRAUS DECEMBER 14, 2016

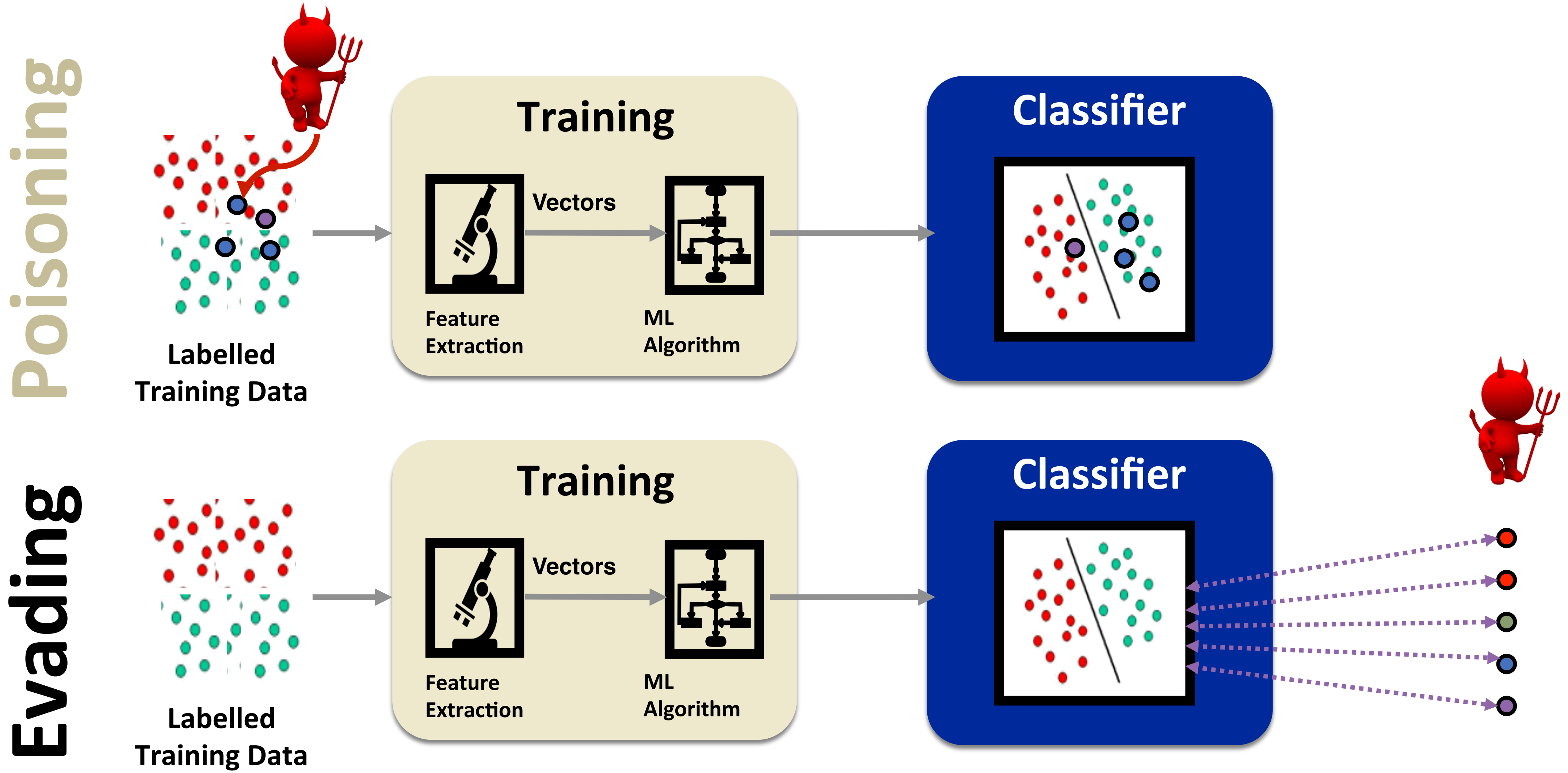


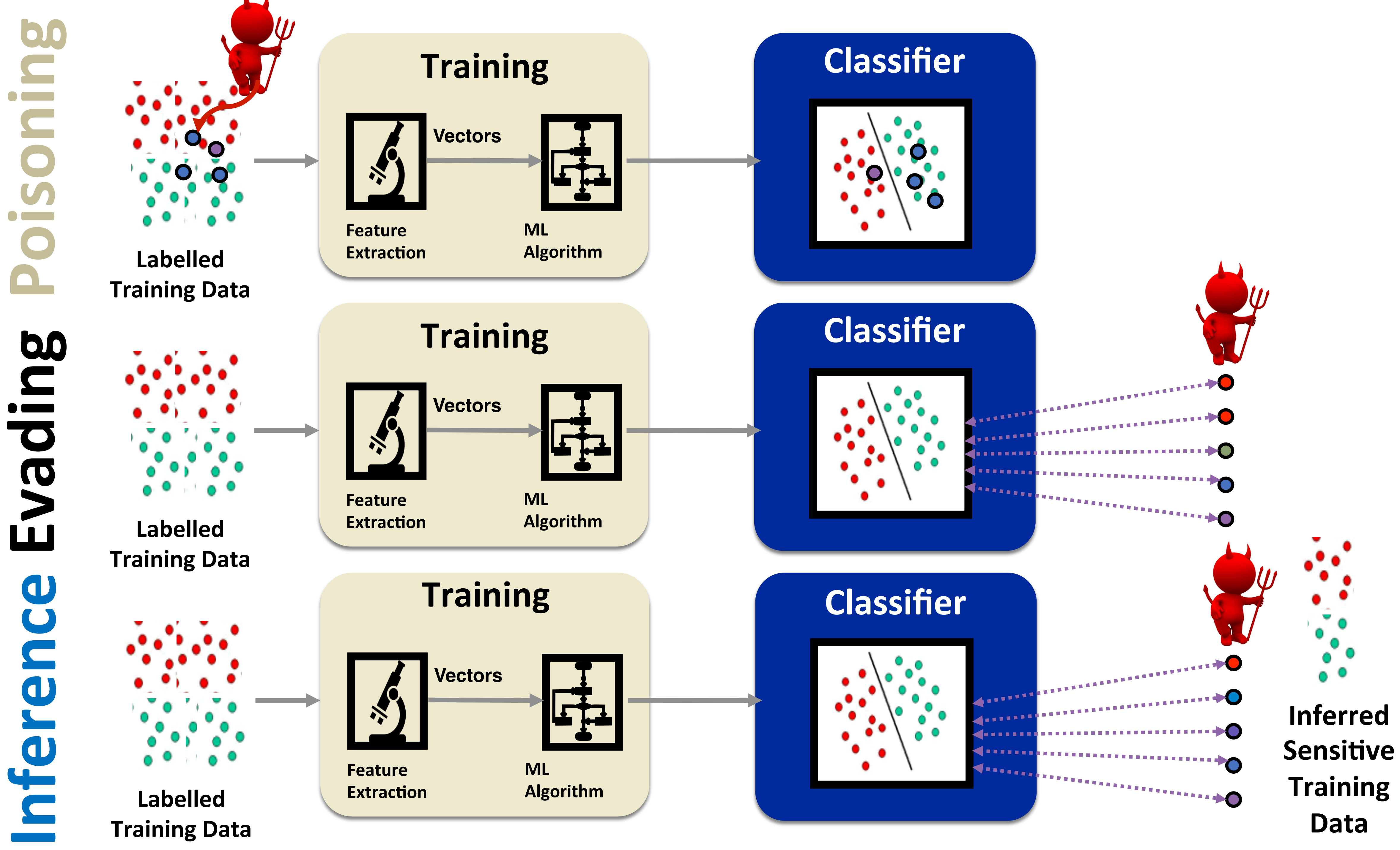
Assumption: Training Data is *Representative*

Adversaries Don't Cooperate

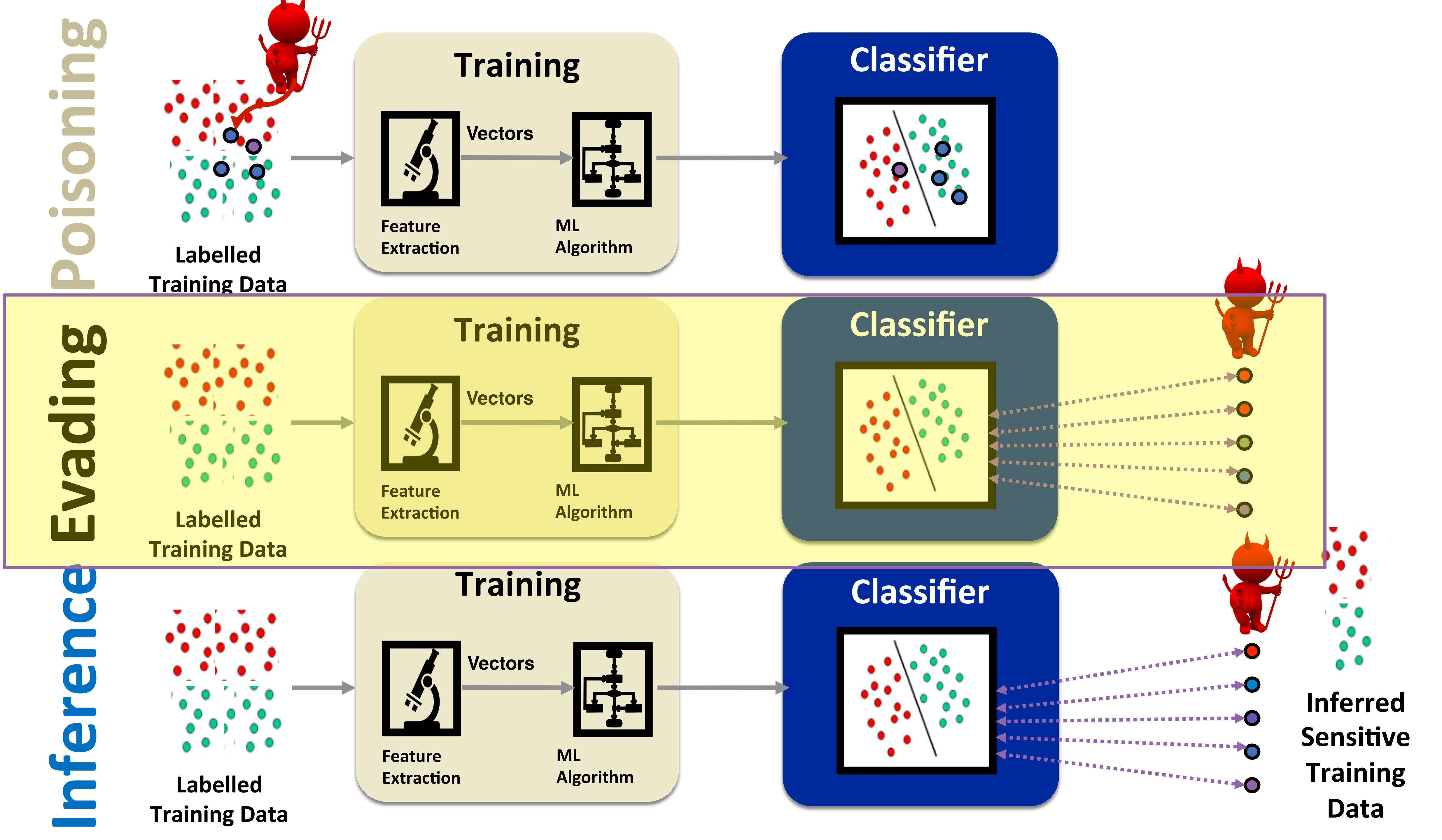


Adversaries Don't Cooperate





Poisoning
Evading
Inference
Labelled
Training Data
Training
Vectors
Feature
Extraction
ML
Algorithm
Classifier
Labelled
Training Data
Training
Vectors
Feature
Extraction
ML
Algorithm
Classifier
Labelled
Training Data
Training
Vectors
Feature
Extraction
ML
Algorithm
Classifier
Inferred
Sensitive
Training
Data



Poisoning
Labelled
Training Data
Training
Vectors
Feature
Extraction
ML
Algorithm
Classifier
Evading
Labelled
Training Data
Training
Vectors
Feature
Extraction
ML
Algorithm
Classifier
Inference
Labelled
Training Data
Training
Vectors
Feature
Extraction
ML
Algorithm
Classifier
Inferred
Sensitive
Training
Data

Focus: **Evasion Attacks**

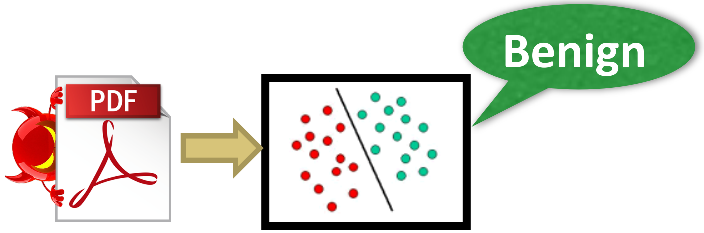
Goal: Automatically **simulate adaptive adversary** against generic classifier

Purpose:

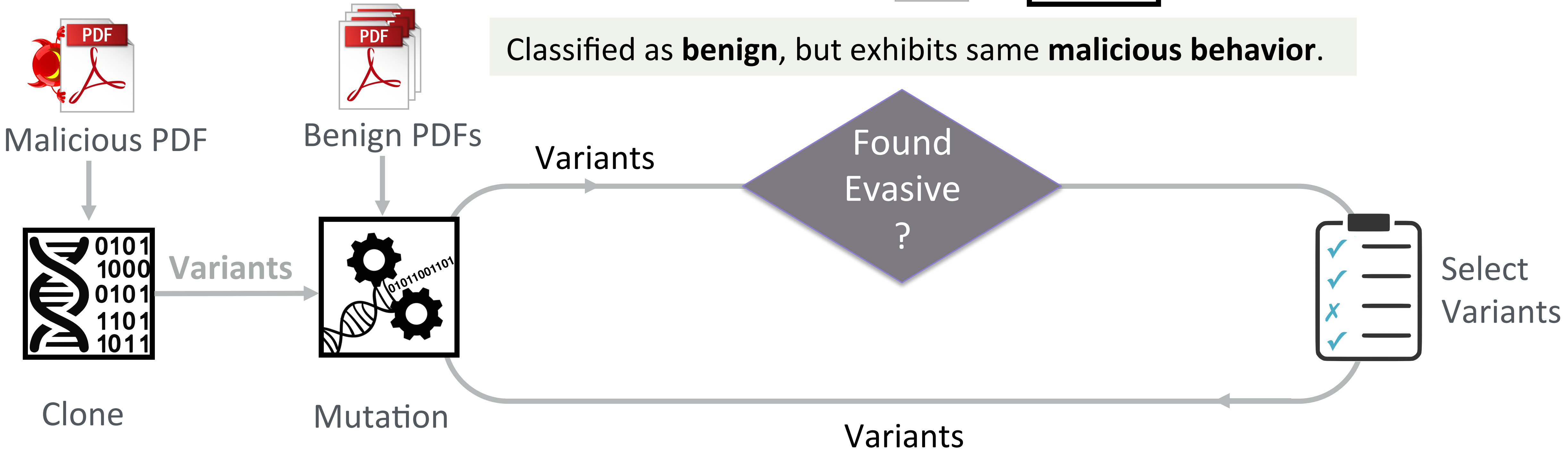
Understand classifier robustness

Build better classifiers (or realize we should give up?)

Automated Evasion using Genetic Programming

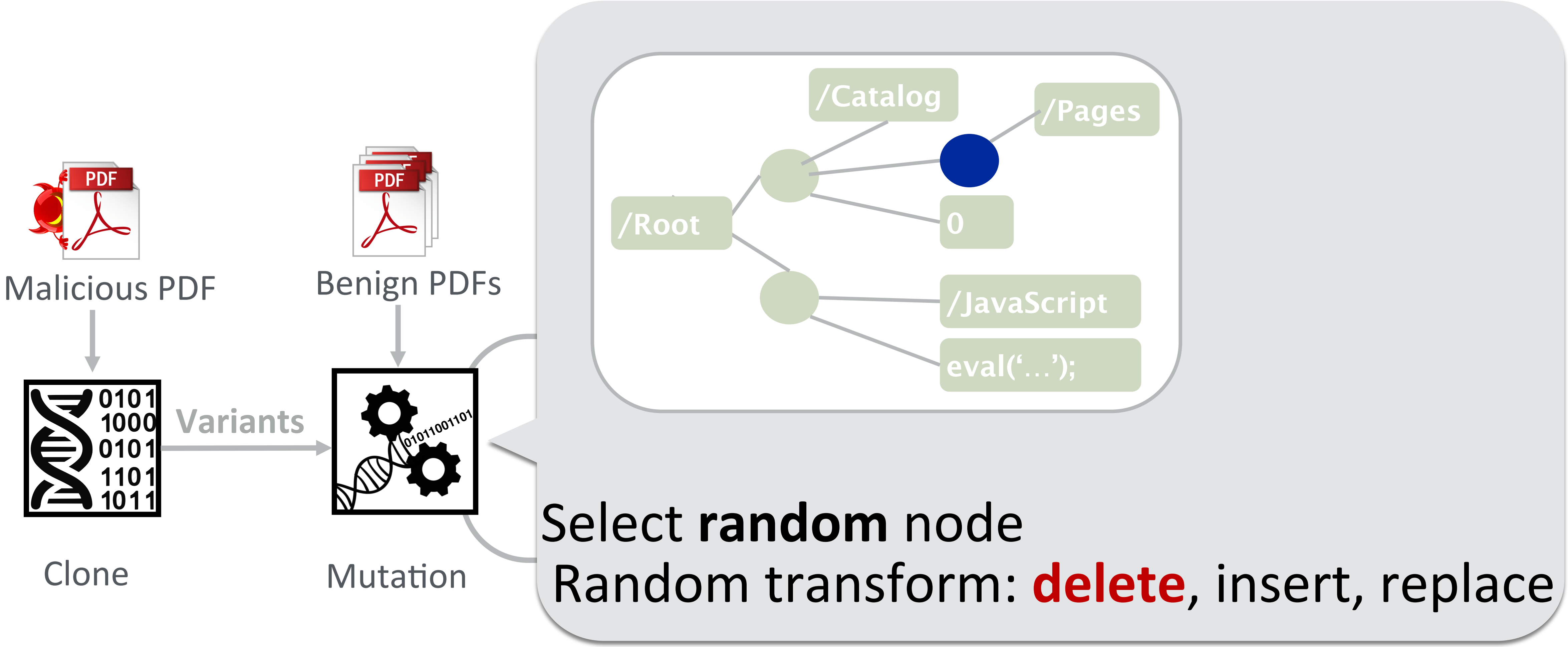
Evasive variant: 

Classified as **benign**, but exhibits same **malicious behavior**.

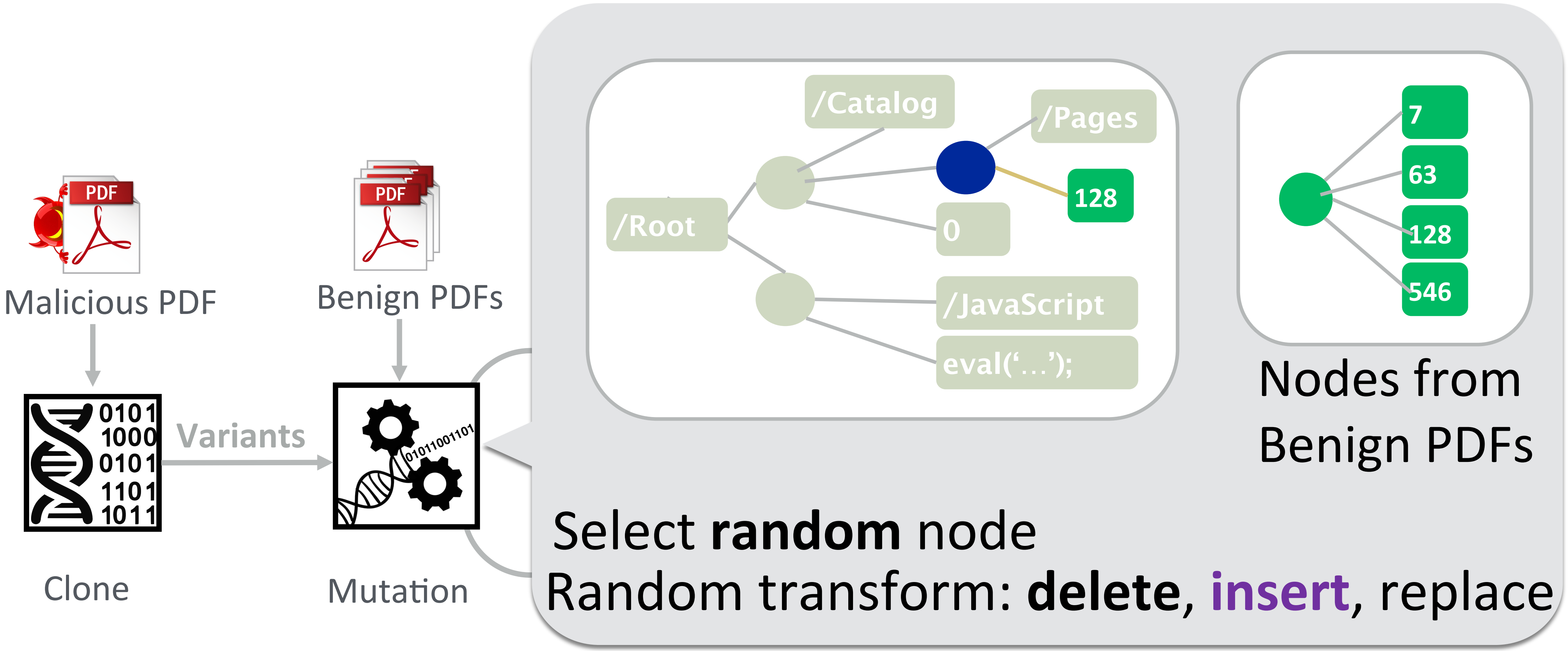


NSF SaTC-funded work with **Weilin Xu** and **Yanjun Qi** (University of Virginia)

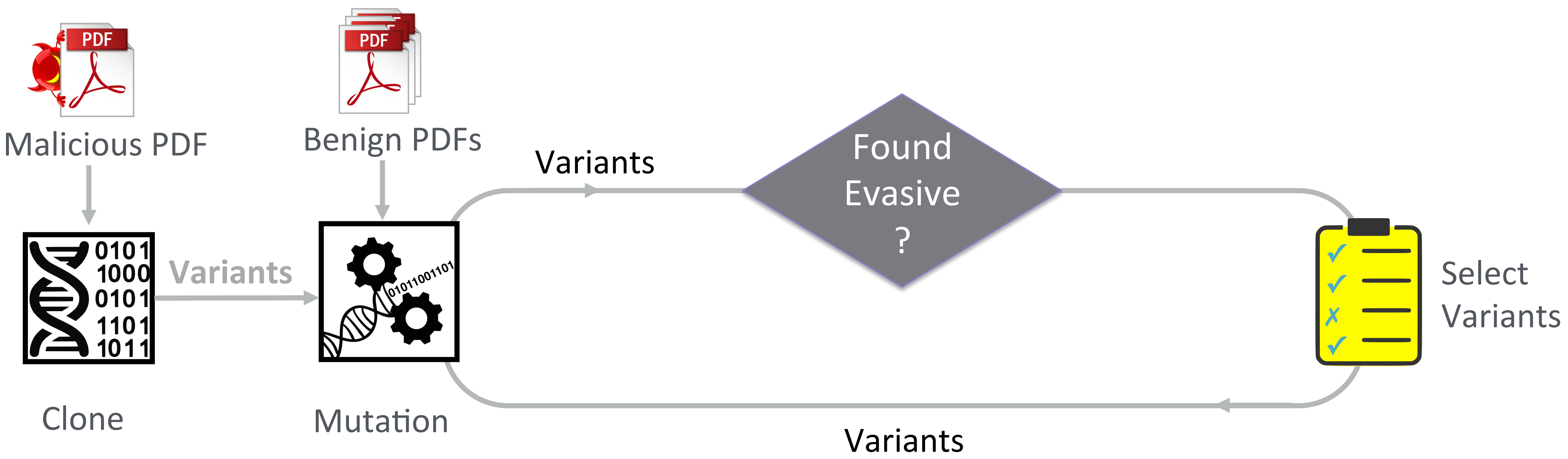
Generating Variants



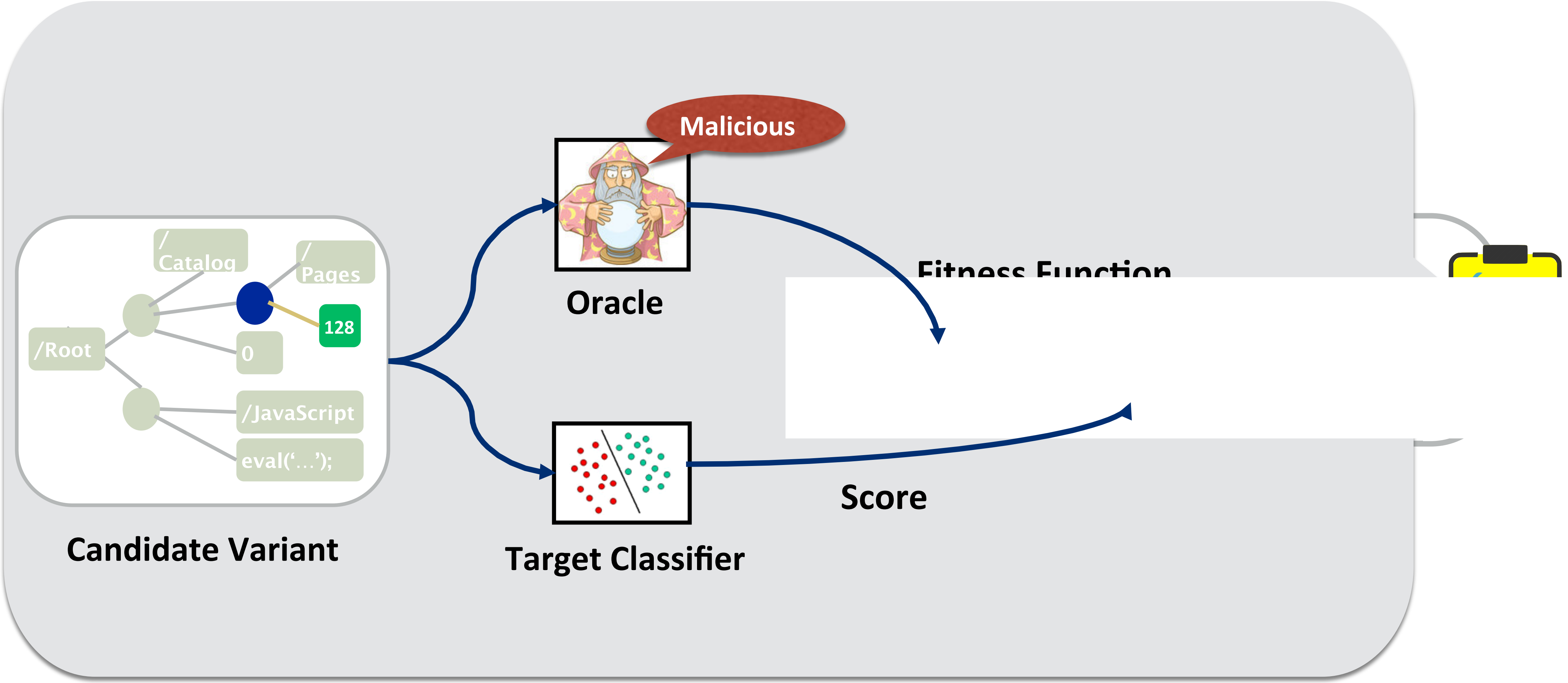
Generating Variants

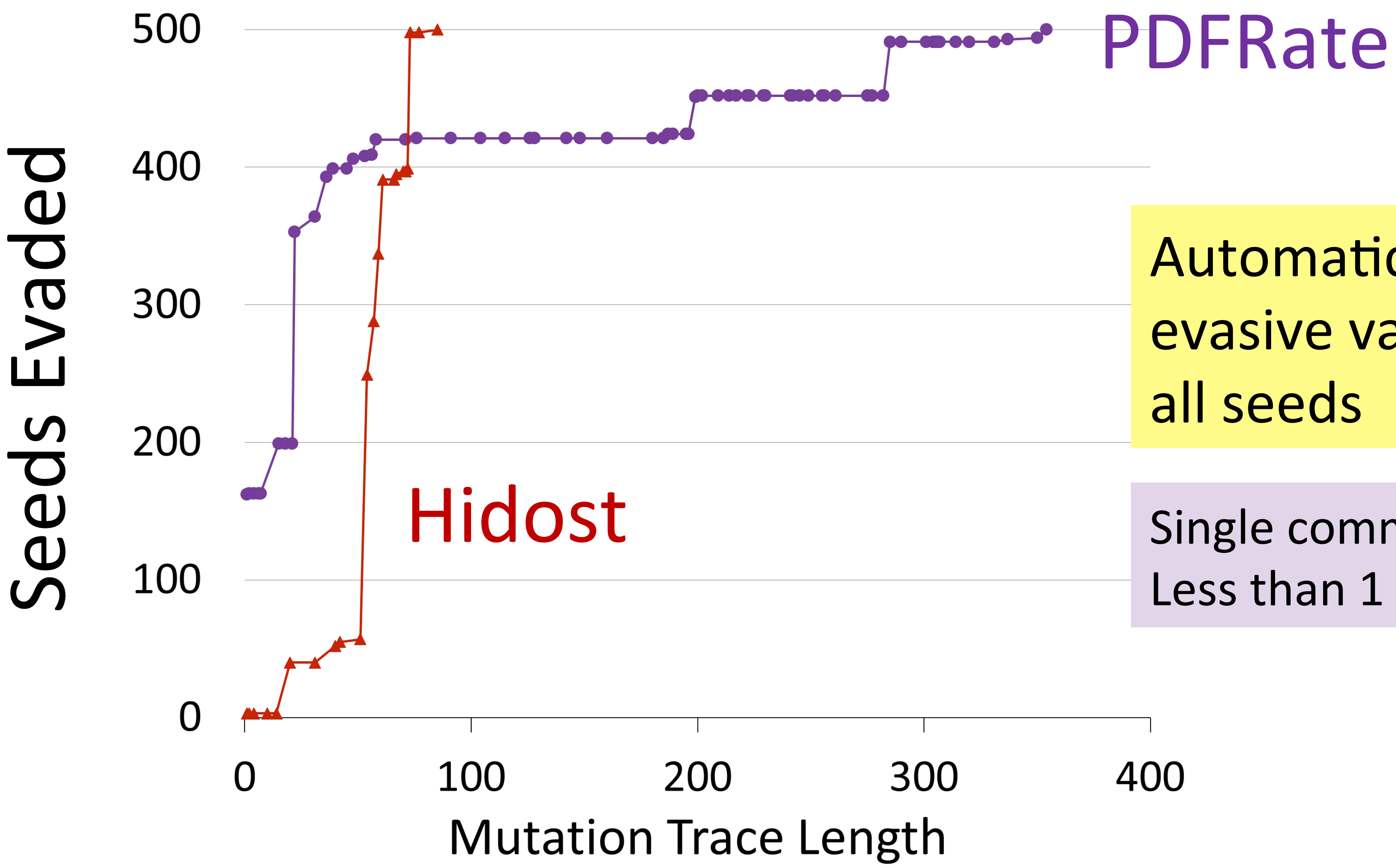


Selecting Promising Variants



Selecting Promising Variants

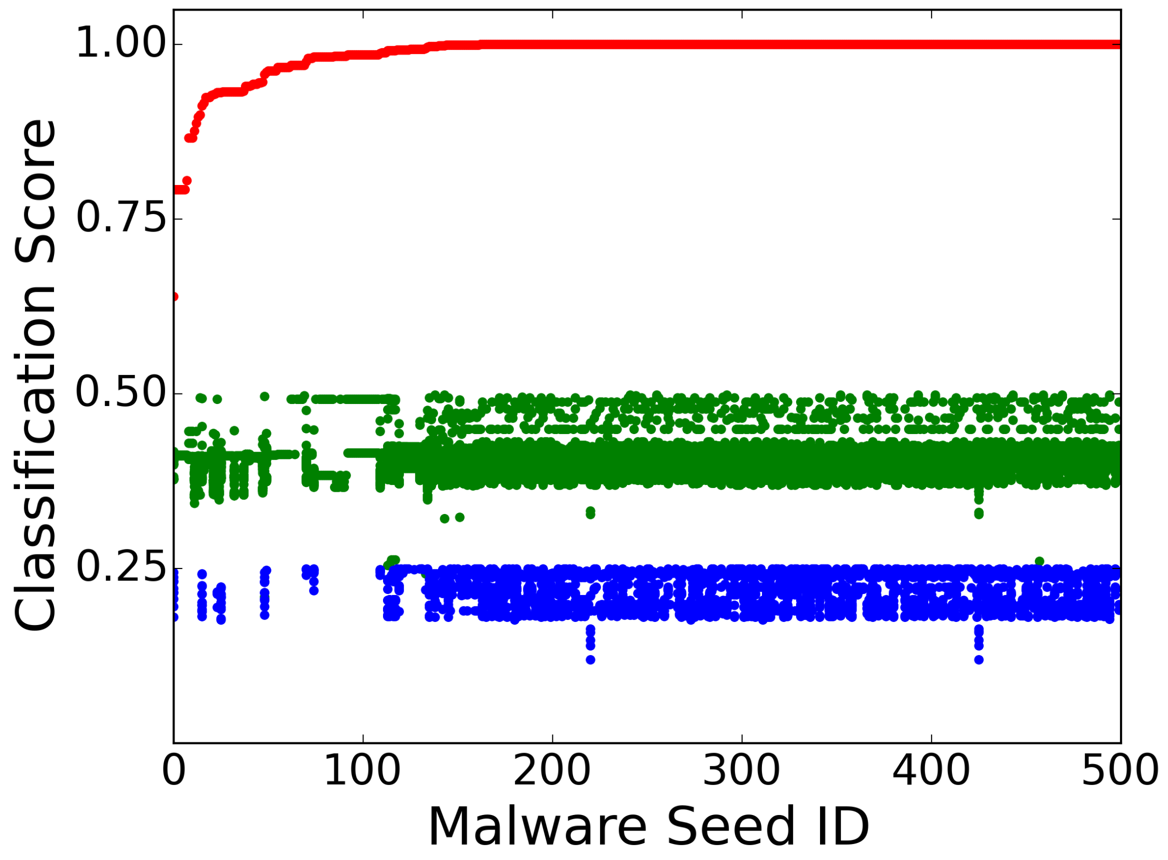




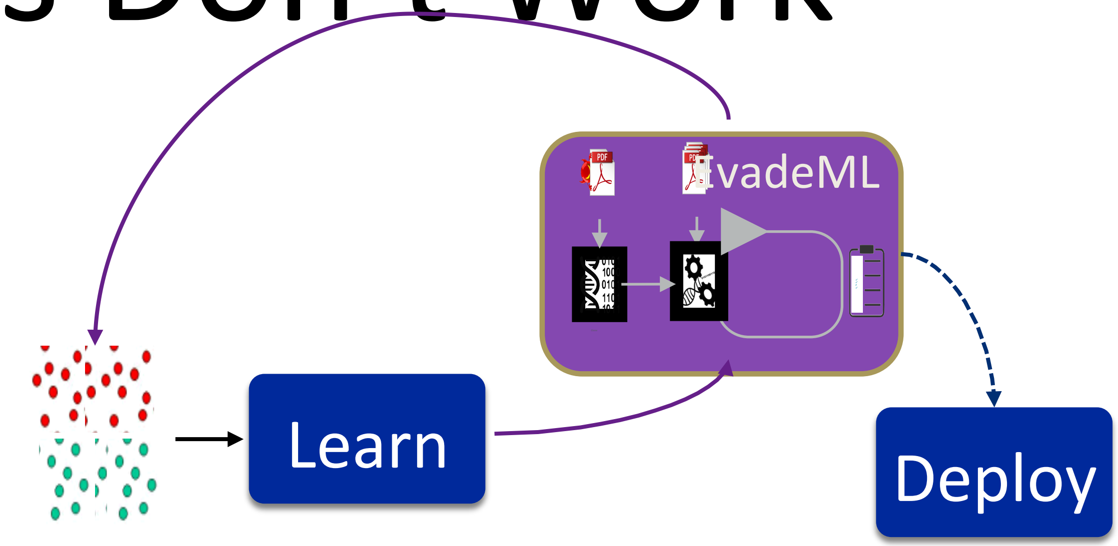
Automatically finds evasive variants for all seeds

Single commodity PC
Less than 1 week

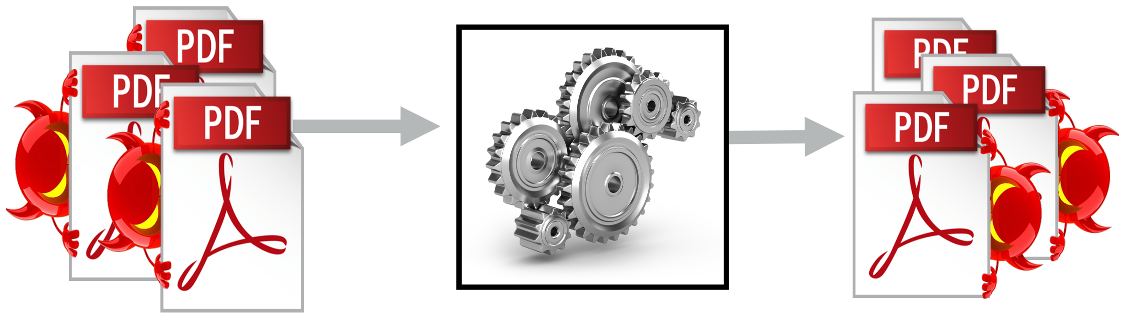
Simple Defenses Don't Work



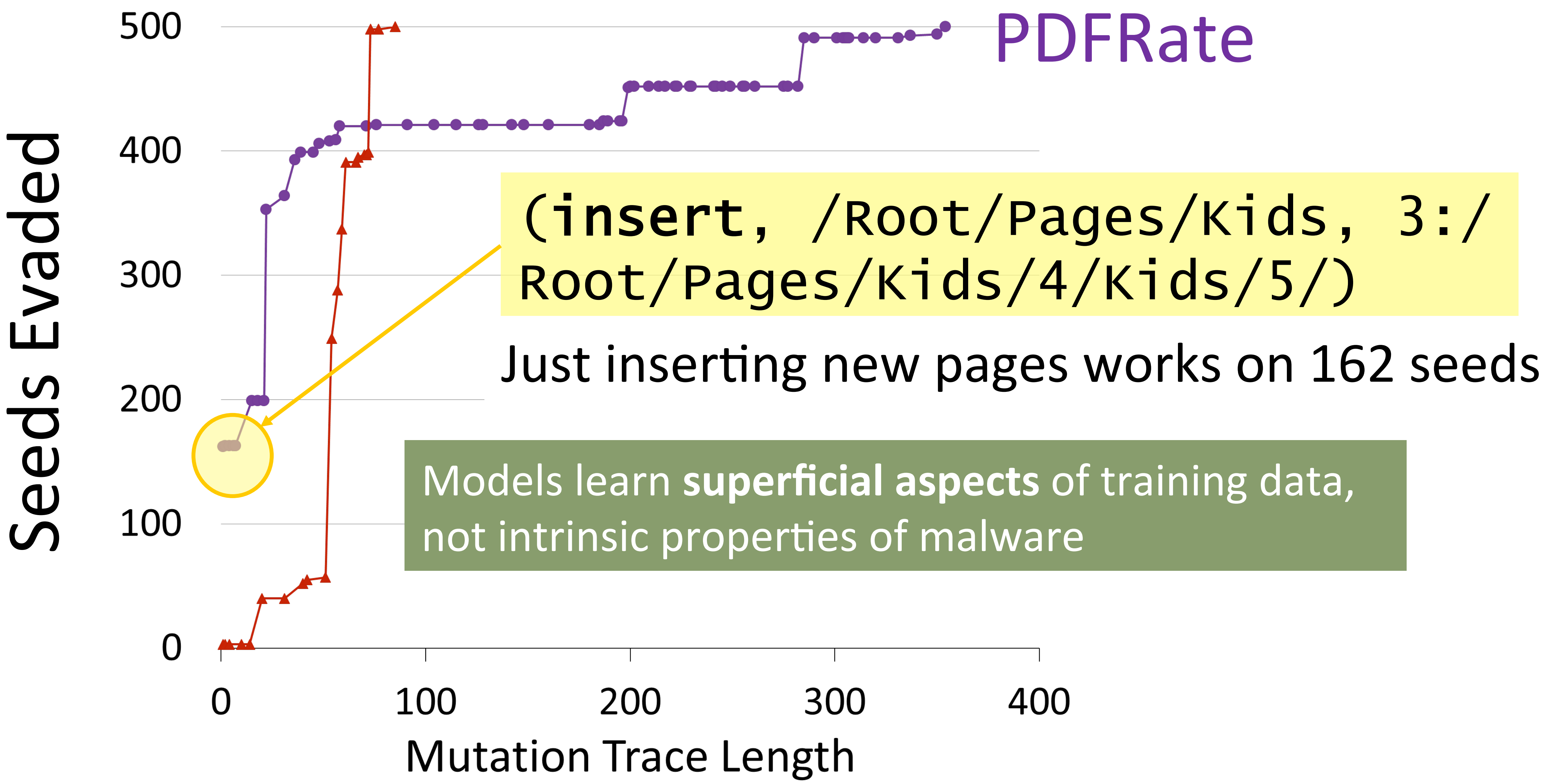
Adjusting Maliciousness Threshold



Retraining with Evasive Variants



Hiding Classifier: Cross-Evasion


PDFRate
(insert, /Root/Pages/Kids, 3:/Root/Pages/Kids/4/Kids/5/)
Just inserting new pages works on 162 seeds
Models learn **superficial aspects** of training data, not intrinsic properties of malware
Seeds Evaded
Mutation Trace Length
0
100
200
300
400
500

Missing Magic

- ML is just learning a function: $f(X) \rightarrow y$
- If input (features) are not real signals of y, function learned is just artifact of training data
- If we know feature that is real signals for y,

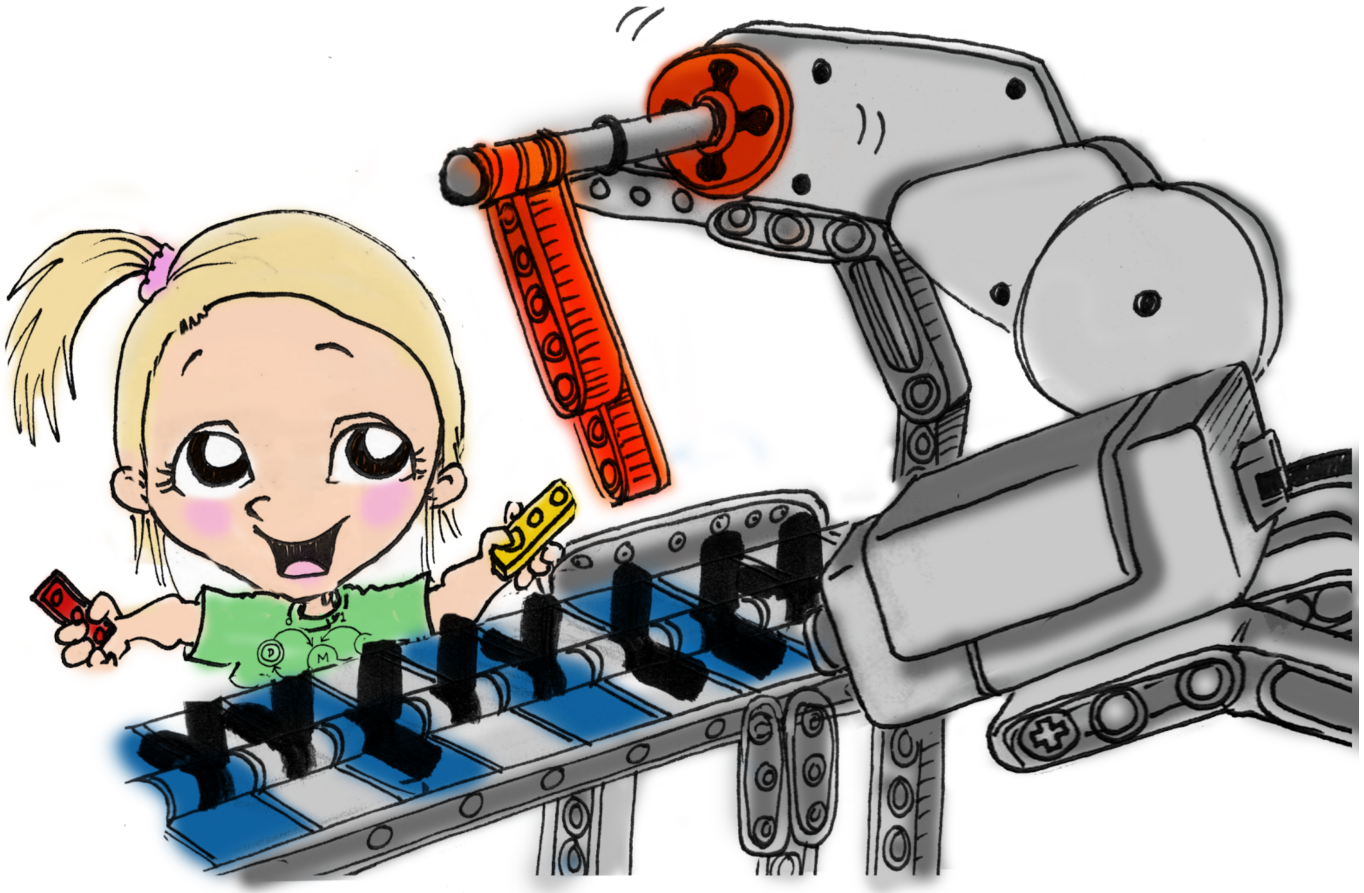
Real goal for ML in security classification should be to learn the **real signals** so that ML is not needed any more.

Big Research Challenges

- How can we **understand** and **reason about** learned models?
- How can we **test** an ML model for *robustness* and *fairness*?
- How can we make useful models from **sensitive data**?

Farnam: "We don't do the easy stuff well, and the hard stuff is getting harder."

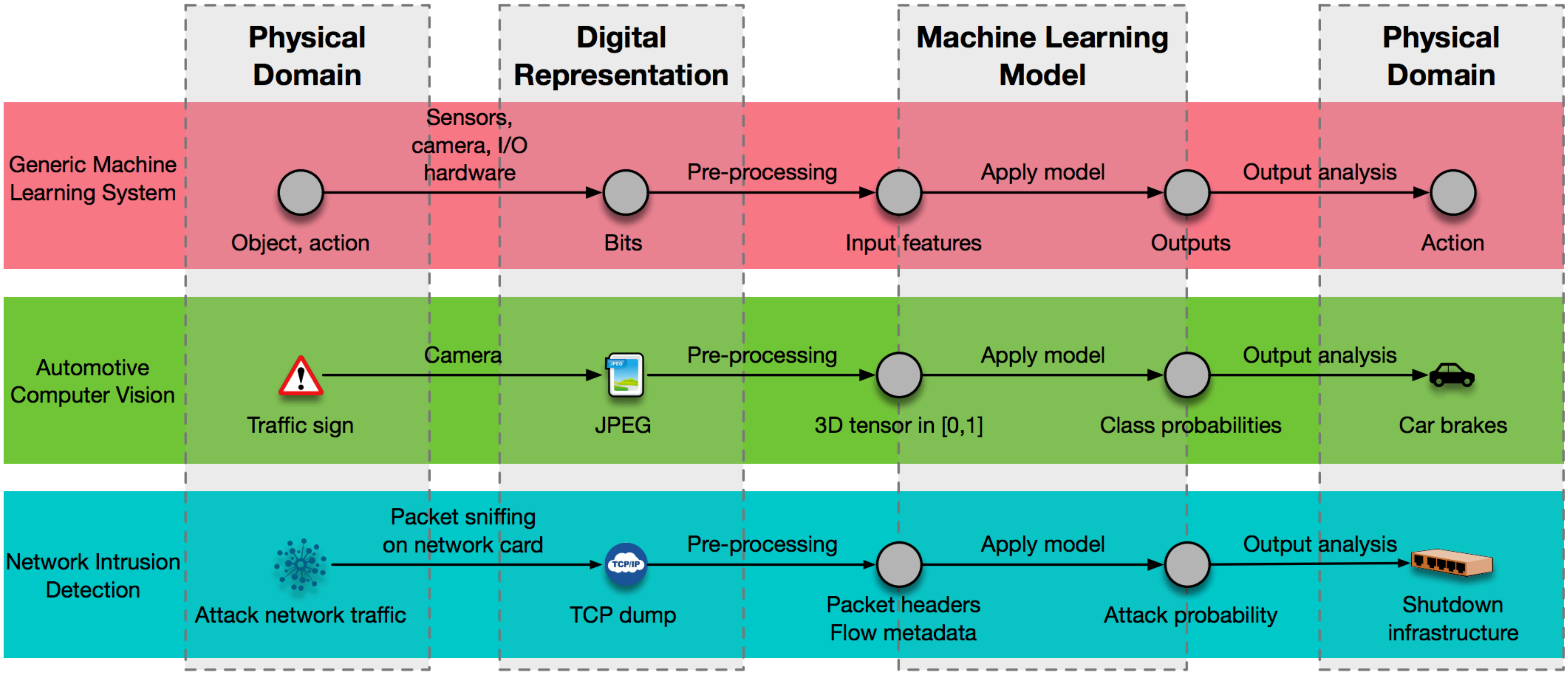
We can't even make computing systems work right when we know what they are supposed to do and have human-written code, *how can we possibly make them work right when we don't?*



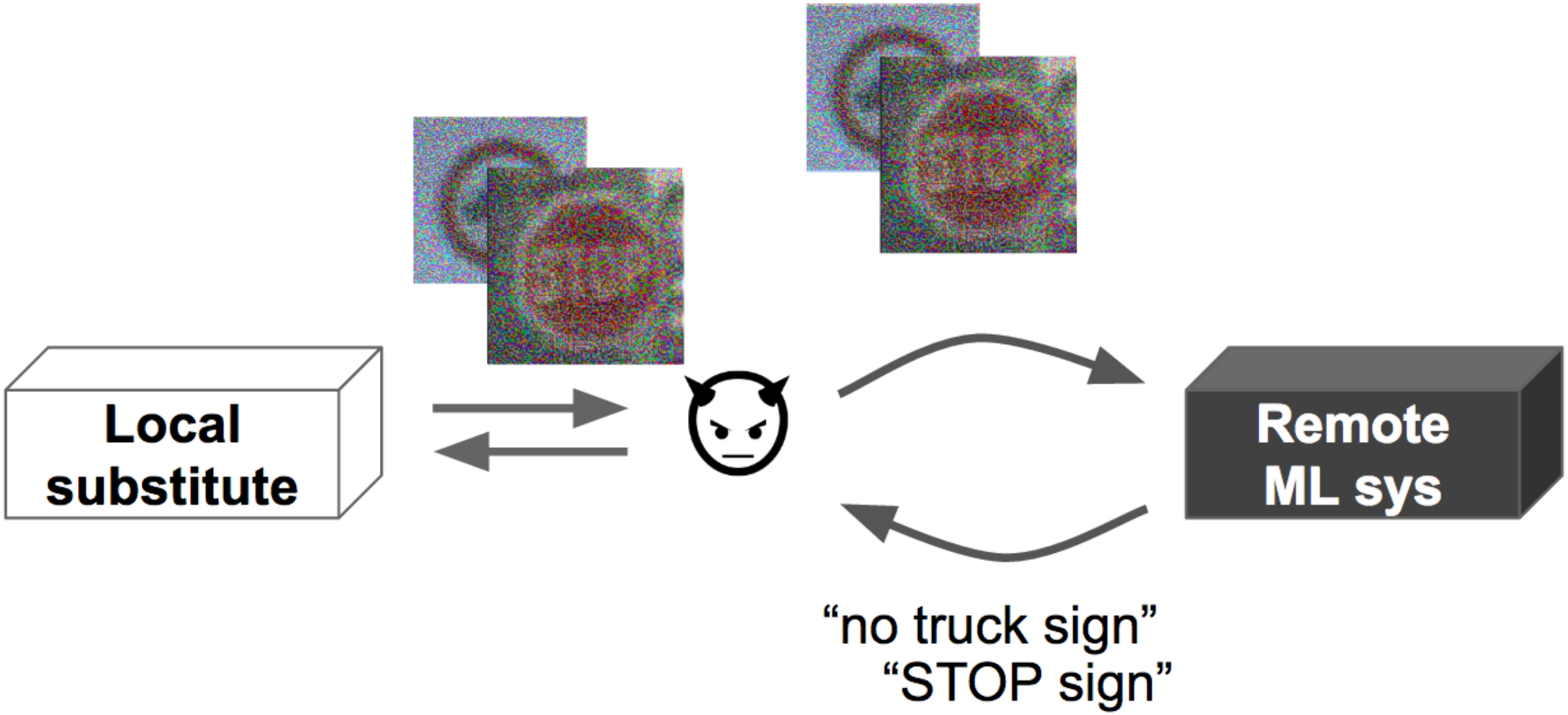
David Evans
evans@virginia.edu

EvadeML.org

Nicolas Papernot



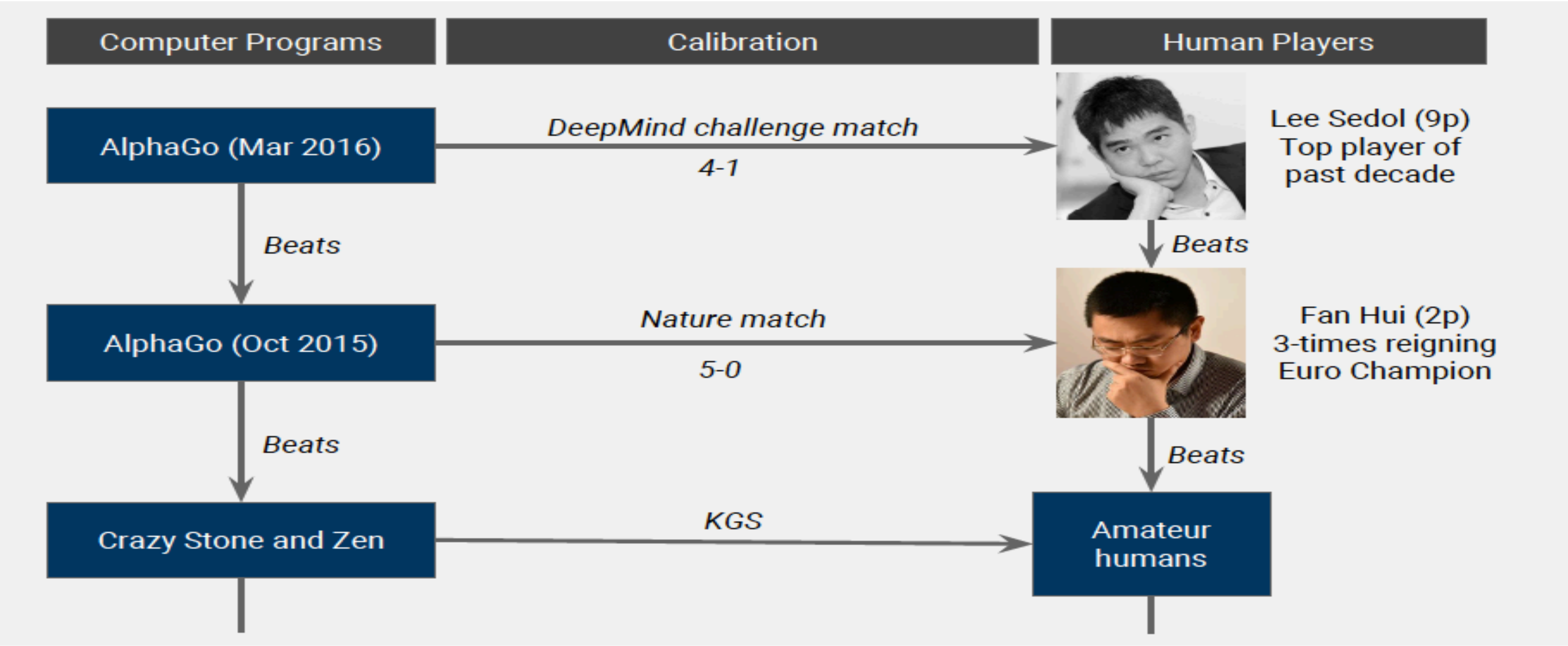
Nicolas Papernot (ngp5056@cse.psu.edu)



Open Challenges in Making AI Systems Secure

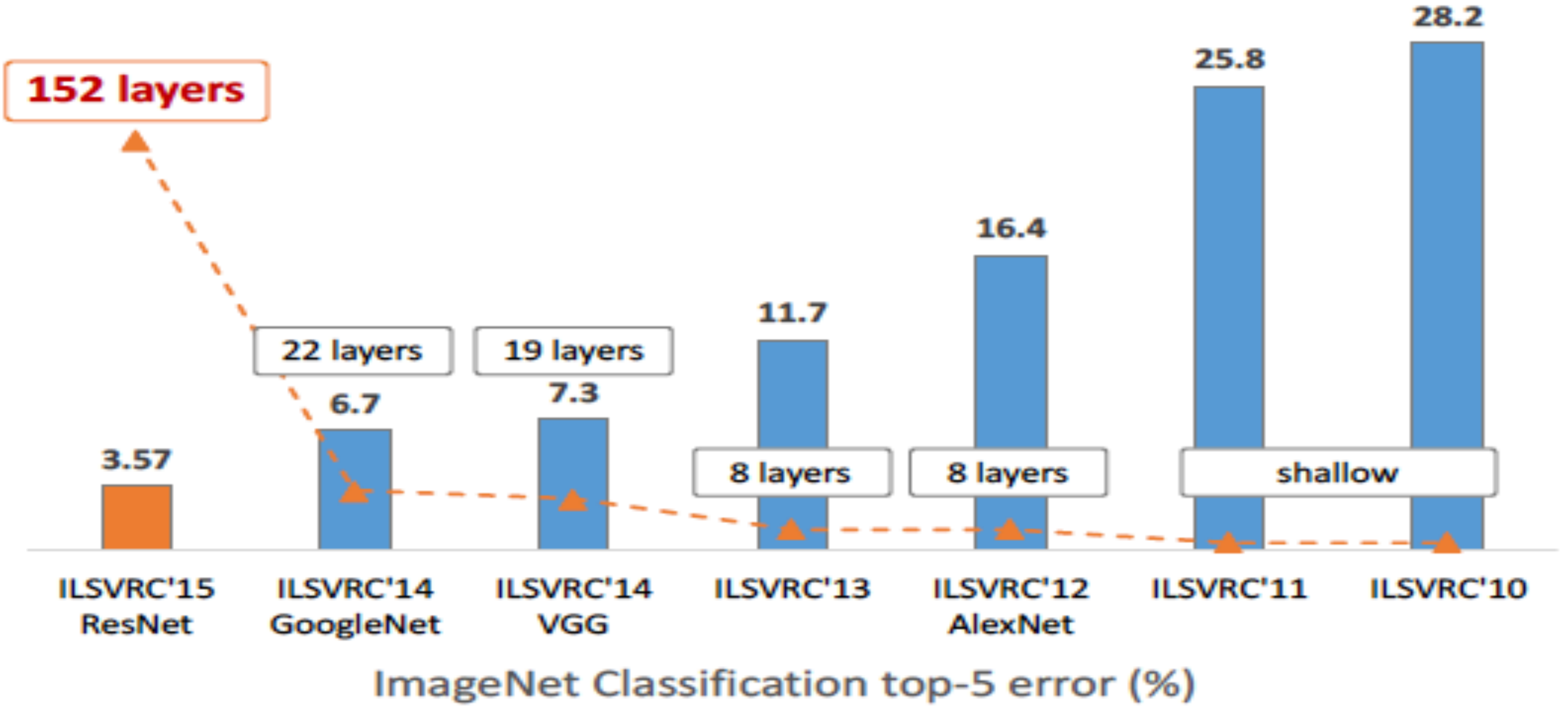
Dawn Song
UC Berkeley

AlphaGo: Winning over World Champion



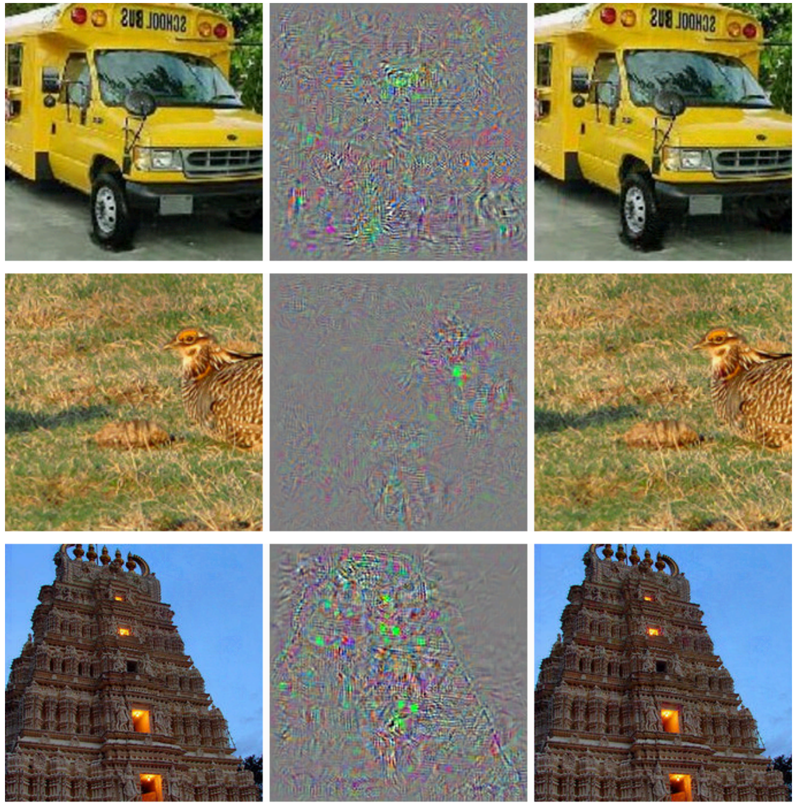
Source: David Silv

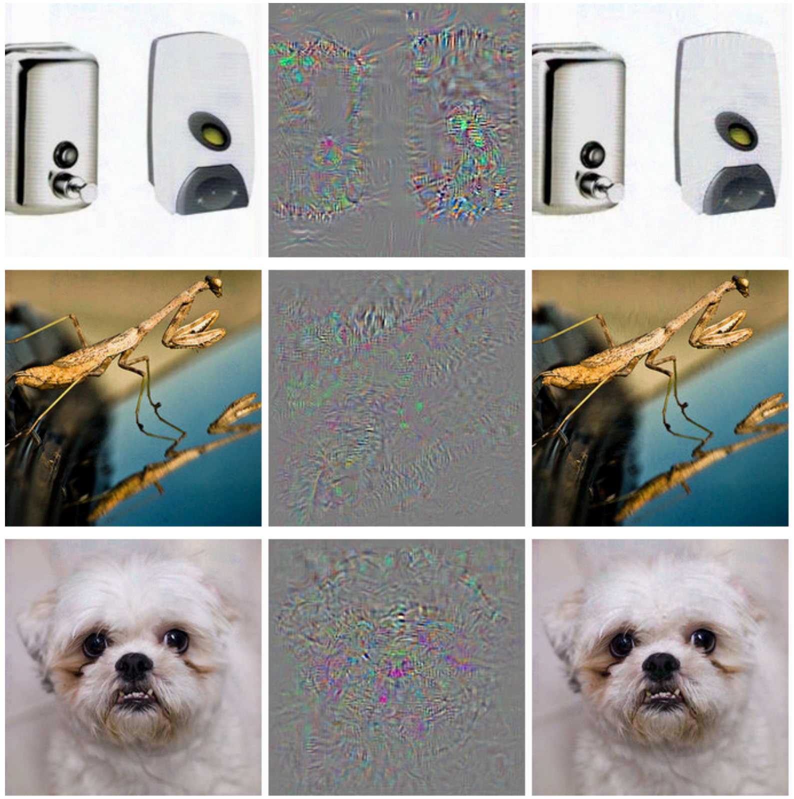
Achieving Human-Level Performance on ImageNet Classification



Source: Kaiming He

Deep Learning Systems Are Easily Fooled





$$\frac{\partial \text{output}}{\partial \text{pixels}}$$

ostrich

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. Intriguing properties of neural networks. ICLR 2014.

Literature:
Image Classification;
Model is known

Weaker threat model:
The machine learning model is a black-box

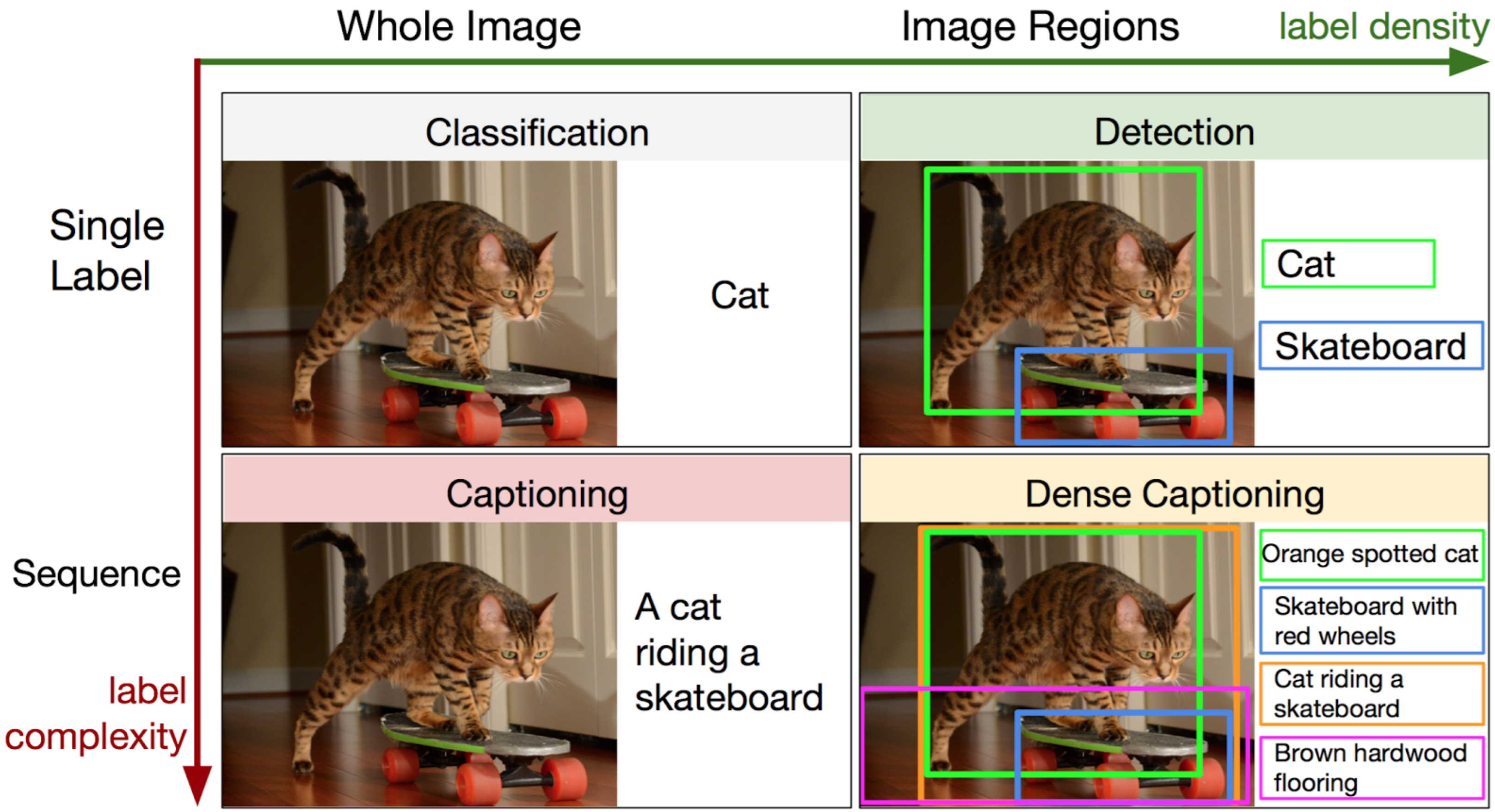
Other model classes:
Generative model

Other tasks:
Object detection;
Captioning

Going Beyond Image Classification

Object Detection

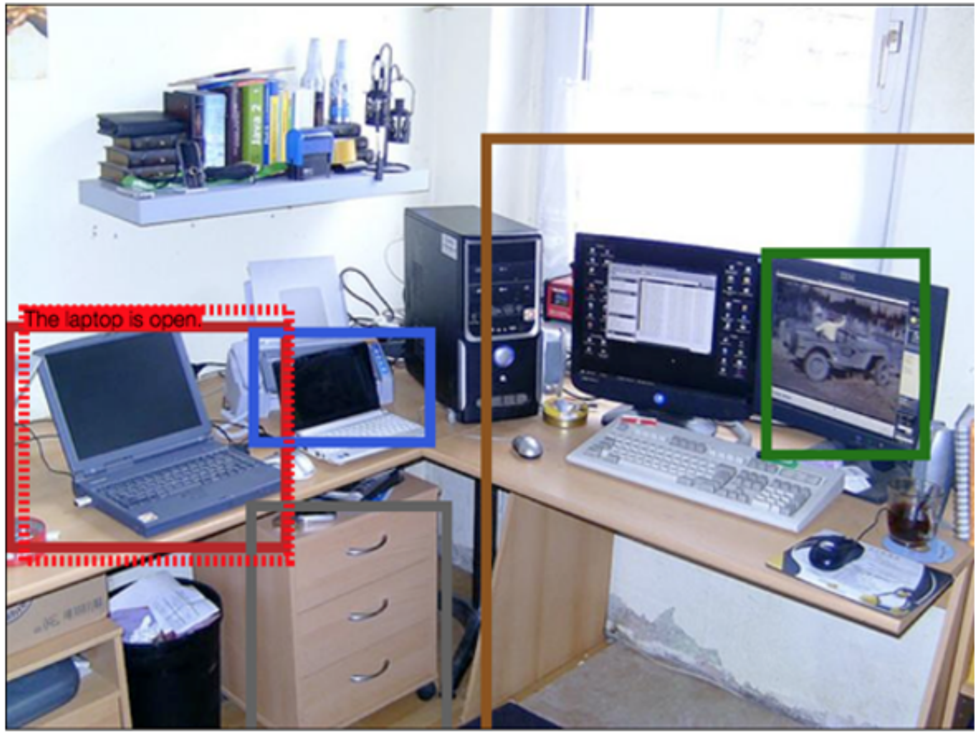
Captioning



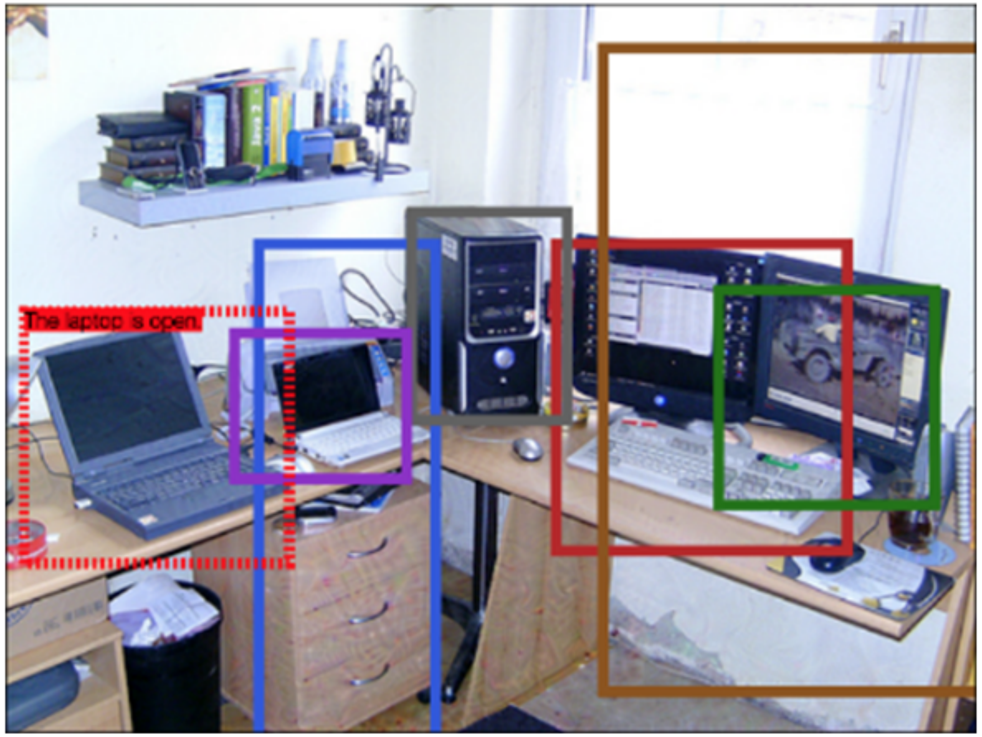
Justin Johnson, Andrej Karpathy, and Fei-Fei, L. Densecap: Fully convolutional localization networks for dense captioning. *CVPR (2016)*

Adversarial Example for Object Detection

Original Image with Detected Objects



Adversarial Image



Adversarial Examples for Captioning

Original Image

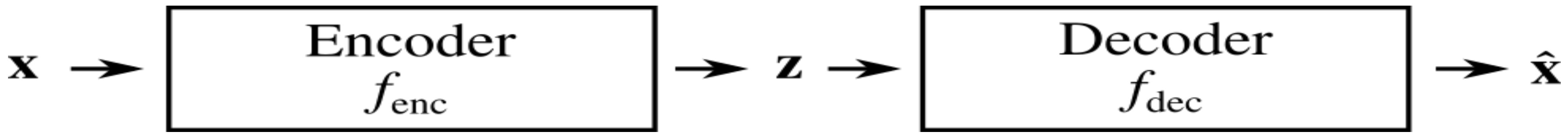
a towel hanging on a rack
a trash can on the floor
a mirror on the wall
a white bathtub
white cabinets under sink

Adversarial Image

a white and red cup
front window of a bus
a dog in a window
a large mirror on the wall
a sign on the side of the bus

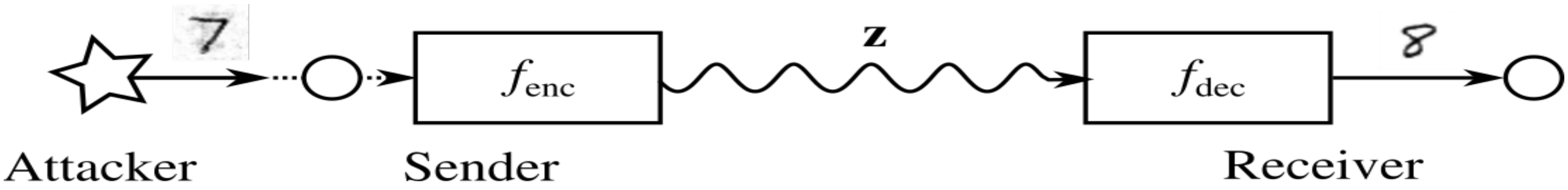
Generative models

- VAE-like models (VAE, VAE-GAN) use an intermediate latent representation
- An **encoder**: maps a high-dimensional input into lower-dimensional latent representation **z**.
- A **decoder:** maps the latent representation back to a high-dimensional reconstruction.



Adversarial Examples in Generative Models

- An example attack scenario:

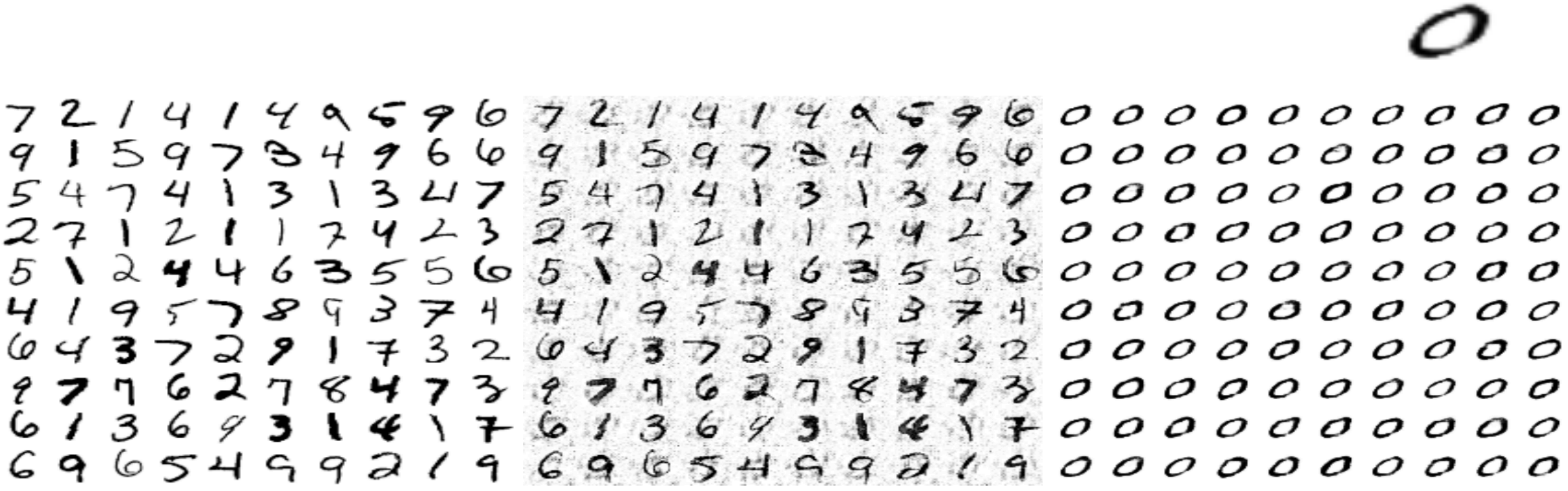


- The generative model: used as a compression scheme.

- Attacker's goal: for the receiver to reconstruct a different image from the one that the sender sees.

Adversarial Examples in MNIST

Target Image



Original images

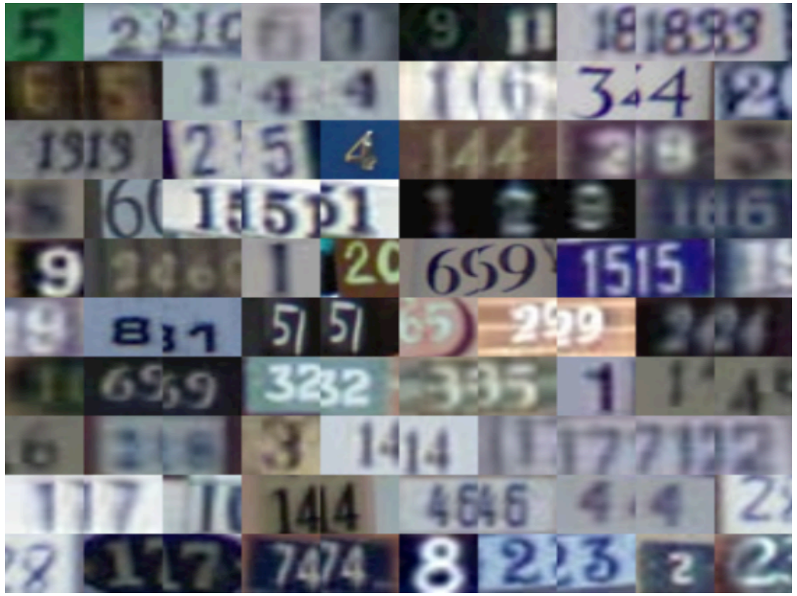
Adversarial examples

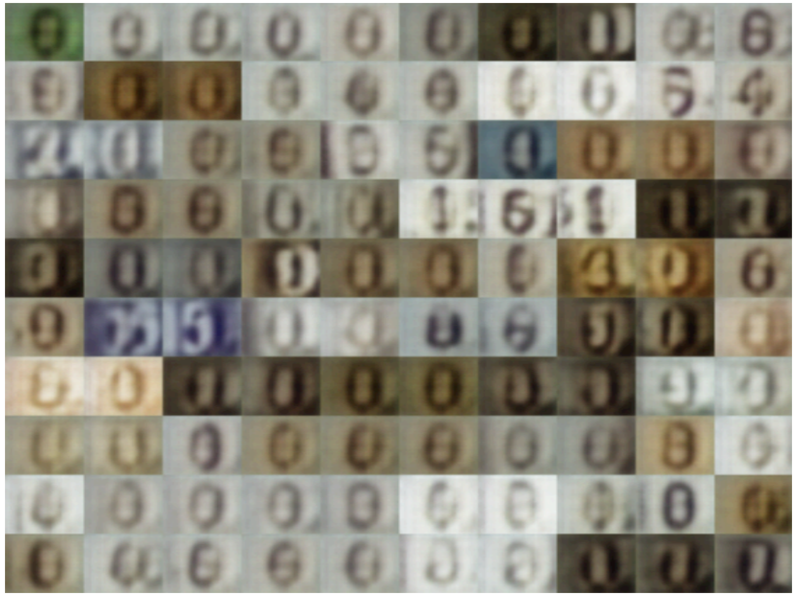
VAE-GAN reconstruction of adversarial examples

Kos, Ian Fischer, Dawn Song: Adversarial Examples for Generative M

Adversarial Examples in SVHN

Target Image





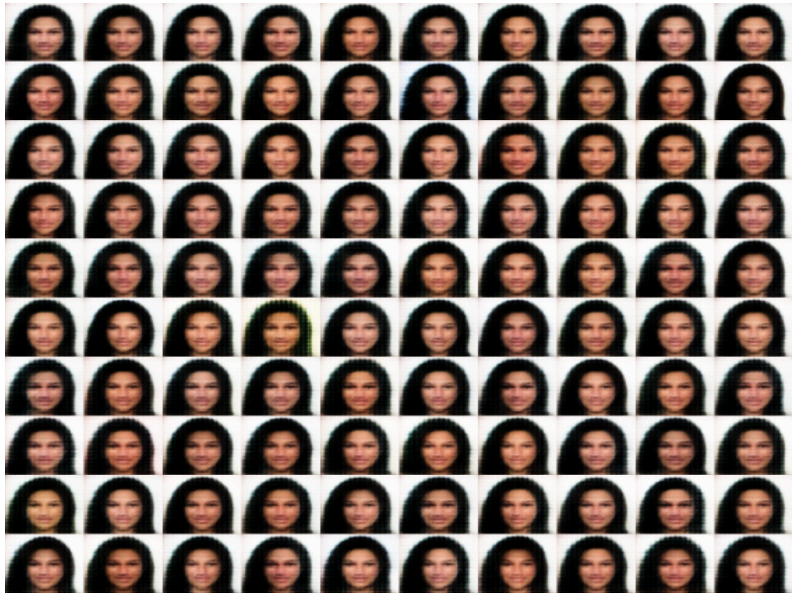
Original images

Adversarial examples

VAE-GAN reconstructio
of adversarial examples

Adversarial Examples in CELEB-A Faces

Target Image

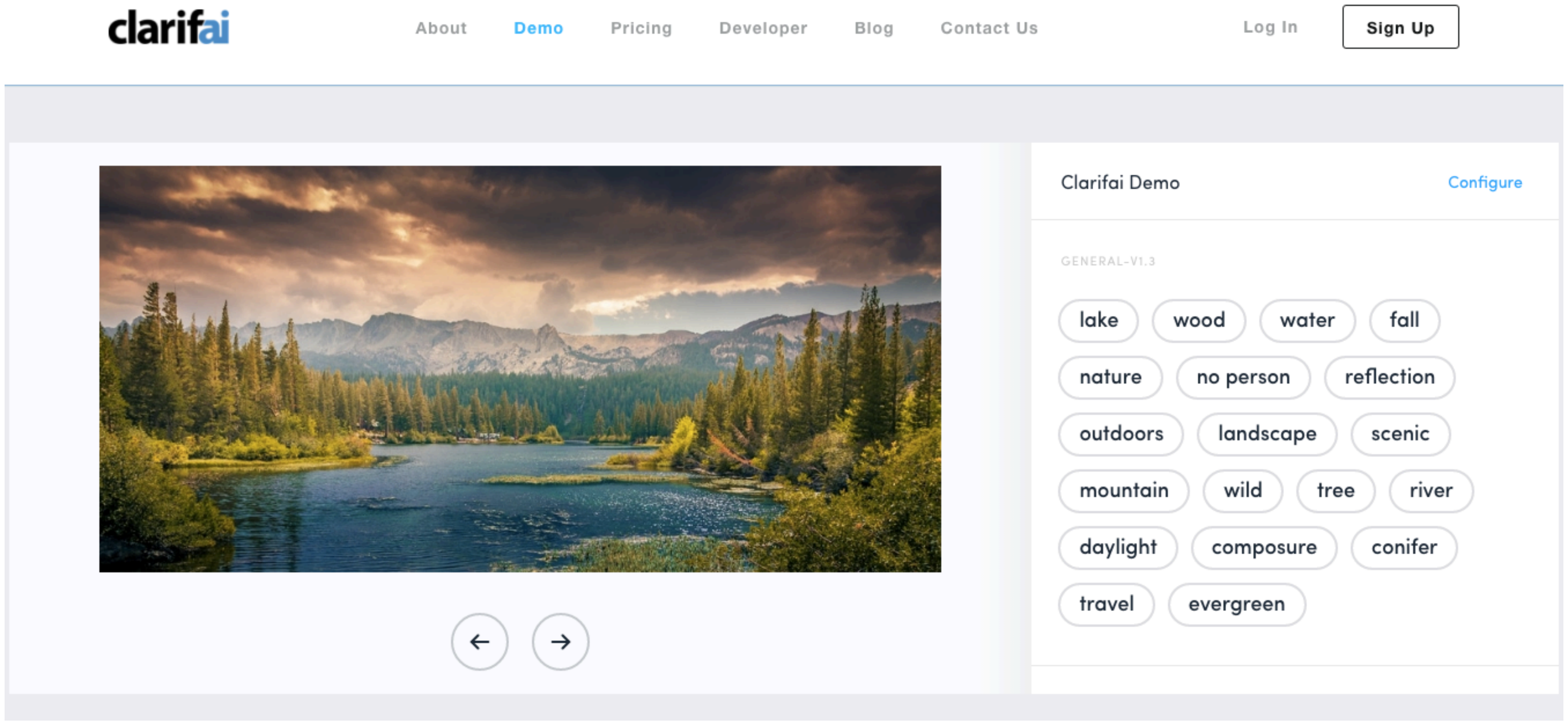


Original images

Adversarial examples

VAE-GAN reconstruction of adversarial examples

Adversarial examples for a black-box system

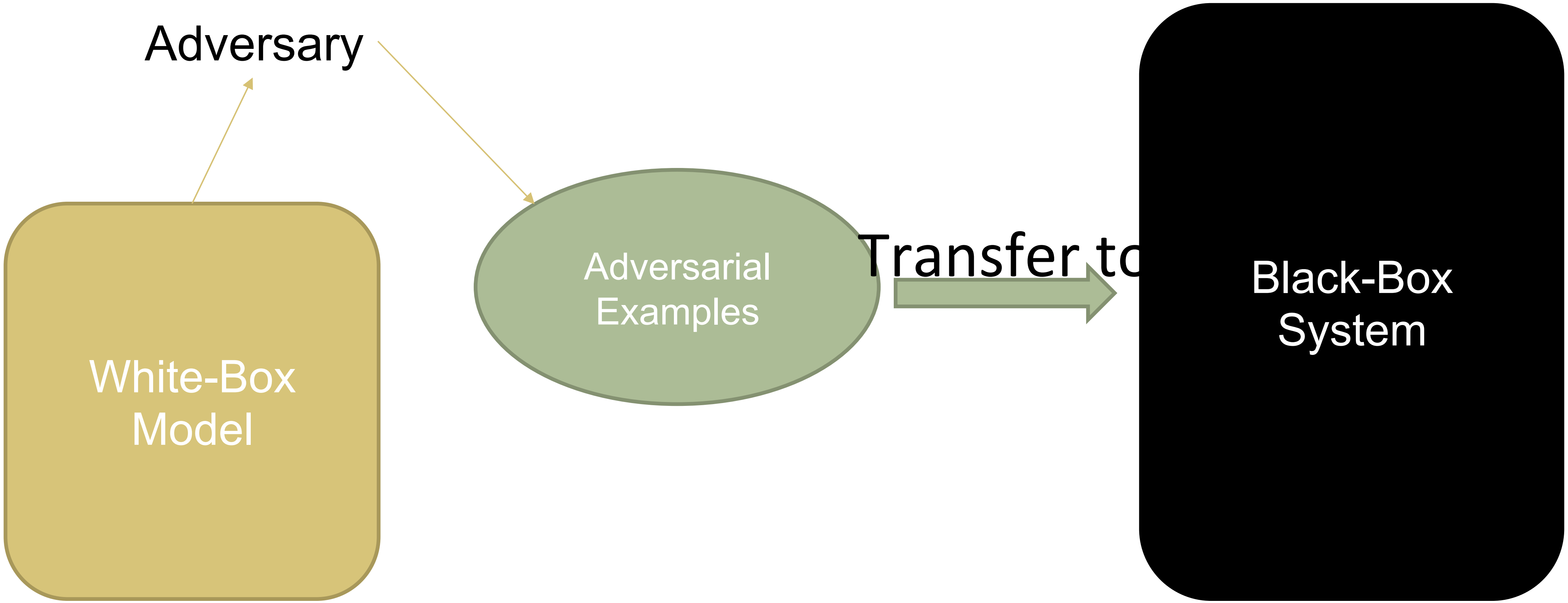


Unknown:

- Model
- Training data
- Label set

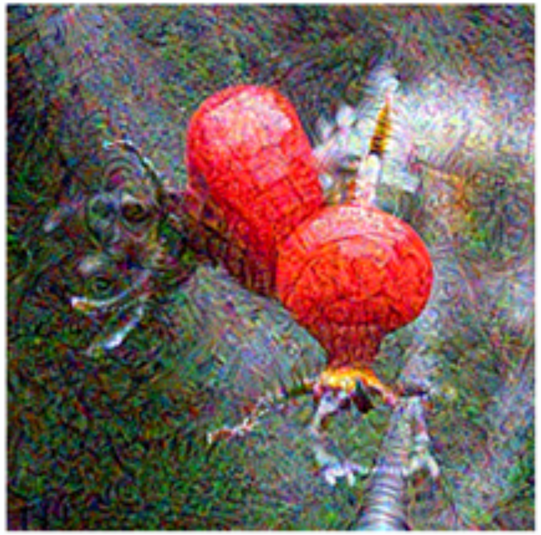
Yanpei Liu, Xinyun Chen, ***Chang Liu***, Dawn Song. Delving into Transferable Adversarial

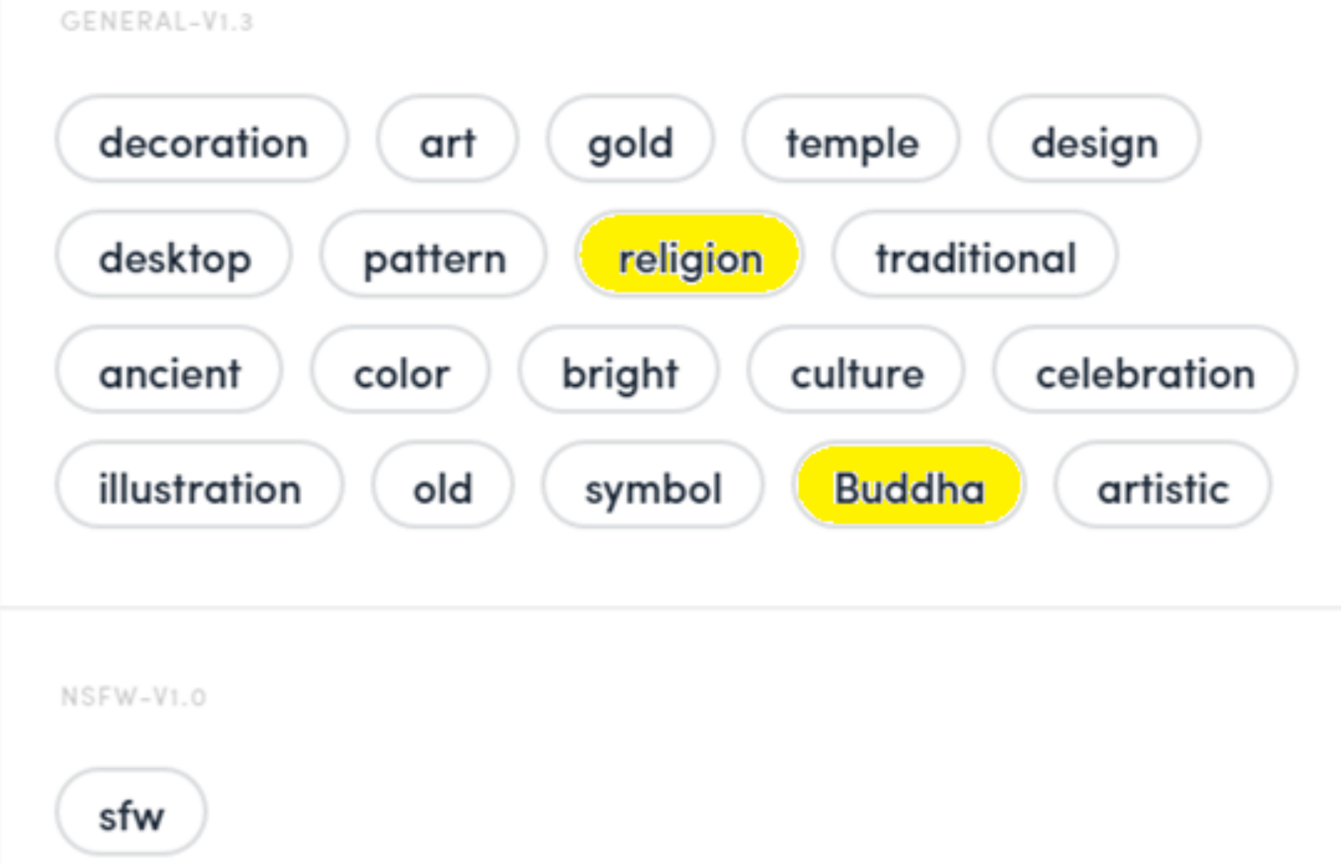
Black-box Attacks Based On Transferability



Adversarial Examples for Clarifai.com

- Ground truth: rosehip
- Original label: fruit
- Target label: **stupa**





Security will be one of the biggest challenges in Deploying AI

Security of Learning Systems

- Software level
- Learning level
- Distributed level

Security of Learning Systems

- Software level
  - No software vulnerabilities such as buffer overflows
  - Existing software security/formal verification techniques apply

- Learning level

- Distributed level

Challenges for Security at Learning Level

- Need to evaluate system under adversarial events, not just normal events

Regression Testing vs. Security Testing in Traditional Software System

	Regression Testing	**Security Testing**
Operation	Run program on **normal** inputs	Run program on **abnormal/adversarial** inputs
Goal	Prevent **normal** users from encountering errors	Prevent **attackers** from encountering **exploitable** errors

Regression Testing vs. Security Testing in Learning System

	Regression Testing	**Security Testing**
Training	Train on noisy training data: Estimate resiliency against noisy training inputs	Train on poisoned training data: Estimate resiliency against poisoned training inputs
Testing	Test on **normal** inputs: Estimate generalization error	Test on **abnormal/adversarial** inputs: Estimate resiliency against adversarial inputs

Challenges for Security at Learning Level

- Need to evaluate system under adversarial events, not just normal events
- Need to reason about complex, non-symbolic programs

No Sufficient Tools to Reason about Non-Symbolic Programs

- Symbolic programs:
  - Semantics defined by logic
  - Decades of techniques & tools developed for logic/symbolic reasoning
    - Theorem provers, SMT solvers
    - Abstract interpretation
- Non-symbolic programs:
  - No precisely specified properties & goals
  - No good understanding of how system works
  - Traditional symbolic reasoning techniques do not apply

Need to Understand Better When System Works/Breaks

- Understand the assumptions/conditions required for system to work
  - Go beyond test error evaluation
  - Evaluate assumptions & conditions in real-world application
  - Especially important for critical-applications
    - Self-driving cars, etc.

Can We Provide Provable Guarantees for Learning Systems?

- **Example problem**:
  - Neural architectures that learn programs currently do not generalize well (e.g., to problems of longer input length)
  - No provable guarantees about the generalization of the learned programs
- **Approach:**
  - Introduce notion of recursion to neural programs: ***Recursive neural programs***
    - Using recursion, a problem is reduced to *subproblems*
      - Base cases and reduction rules
- **Proof of Generalization**:
  - Recursion enables provable guarantees about neural programs
  - Prove perfect generalization of a learned recursive program via a verification procedure, by explicitly testing on all possible base cases and reduction rules

Accuracy on Randomly Generated Problems for Topological Sort

Number of Vertices	Non-Recursive	Recursive
5	6.7%	100%
6	6.7%	100%
7	3.3%	100%
8	0%	100%
70	0%	100%

Making Neural Programming Architectures Generalize via Recursion [Jonathon Cai, Richard Shin, and Dawn Song]

Challenges for Security at Learning Level

- Need to evaluate system under adversarial events, not just normal events
- Need to reason about complex, non-symbolic programs
- Need to reason about how to compose components

Compositional Reasoning

- Building large, complex systems require compositional reasoning
  - Each component provides abstraction
    - E.g., pre/post conditions
  - Hierarchical, compositional reasoning proves properties of whole system
- How to do abstraction, compositional reasoning for non-symbolic programs?

Challenges for Security at Learning Level

- Need to evaluate system under adversarial events, not just normal events
- Need to reason about complex, non-symbolic programs
- Need to reason about how to compose components
- Need to develop new defense approaches

Security of Learning Systems

- Software level
- Learning level
- Distributed level
  - Each agent makes local decisions; how to make good local decisions achieve good global decision?

Summary

- Security will be one of the biggest challenges for deploying AI
- Traditional program analysis and verification approaches are insufficient for learning systems
- Need new ways
  - Define & reason about security
  - Build defense
  - Ensure fairness and other properties

dawnsong@cs.berkeley.edu

Let's tackle the big challenges together!

Strategic Considerations for Learning Agents

Michael Wellman
University of Michigan

What is Special about Machine Learning?

- Increasingly serves key functional role in deployed information systems
  - Including autonomous cyber-defense systems
- Behavior *designed* to be influenced by experience
  - Opens up new areas in attack surface
  - More difficult to specify desired behavior in advance, detect deviations
- Novel modes of attack, defense
- Inherent tradeoffs in learning performance and resilience

Strategic Reasoning

- <u>Definition</u>: Analysis of situations where decision outcomes depend on actions of other agents
  - (essentially all security and trust-relevant environments)
- Except in zero-sum interactions (rare!), worst-case analysis does not apply
- Realistic attack scenarios are iterative/interactive, not one-shot
- Formally, situations are *games*
  - Complex: dynamics, uncertainty…
  - Computational game-theoretic methods can provide insights even when games are analytically intractable

AI Safety and Control

- In many respects, AI safety faces same problems as SaTC generally
- With rapid advances in AI and autonomous systems, increasing attention on the *control problem*
  - How to ensure that AI systems faithfully pursue intended human objectives?
- Solutions themselves typically entail learning
- Need much more flow betw AI and SaTC communities