Trustworthy AI… for Systems Security

Lorenzo Cavallaro <l.cavallaro@ucl.ac.uk>
@lcavallaro — https://s2lab.cs.ucl.ac.uk

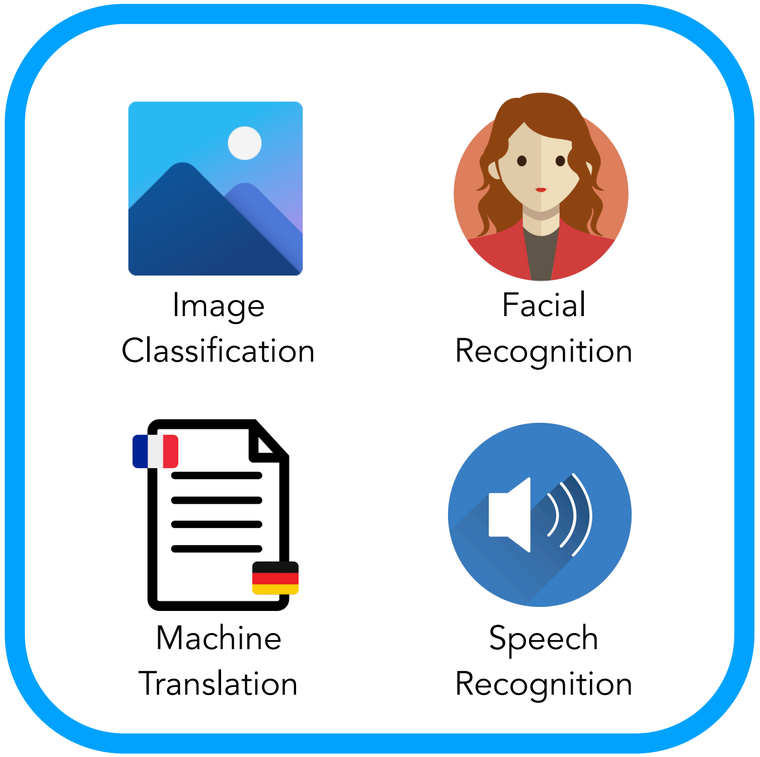
Keynote Talk
NSS-SocialSec 2023

Aug 16, 2023

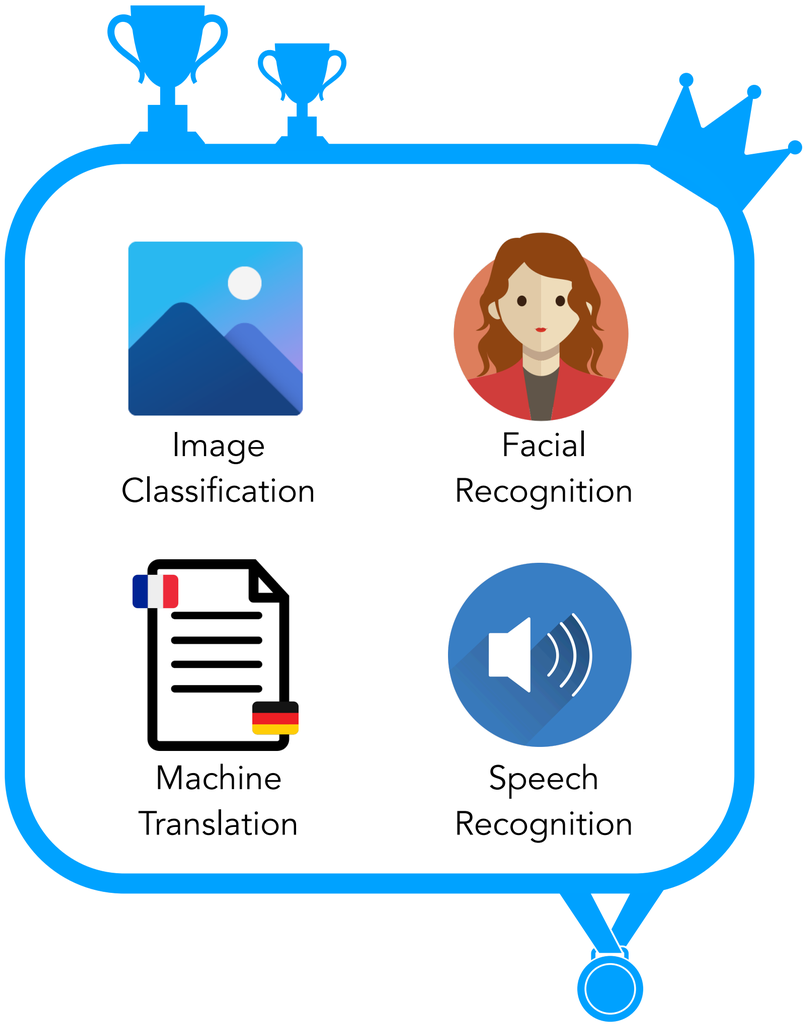
University of Kent, Canterbury, UK

SYSTEMS SECURITY RESEARCH LAB

Machine Learning Revolution



Machine Learning Revolution

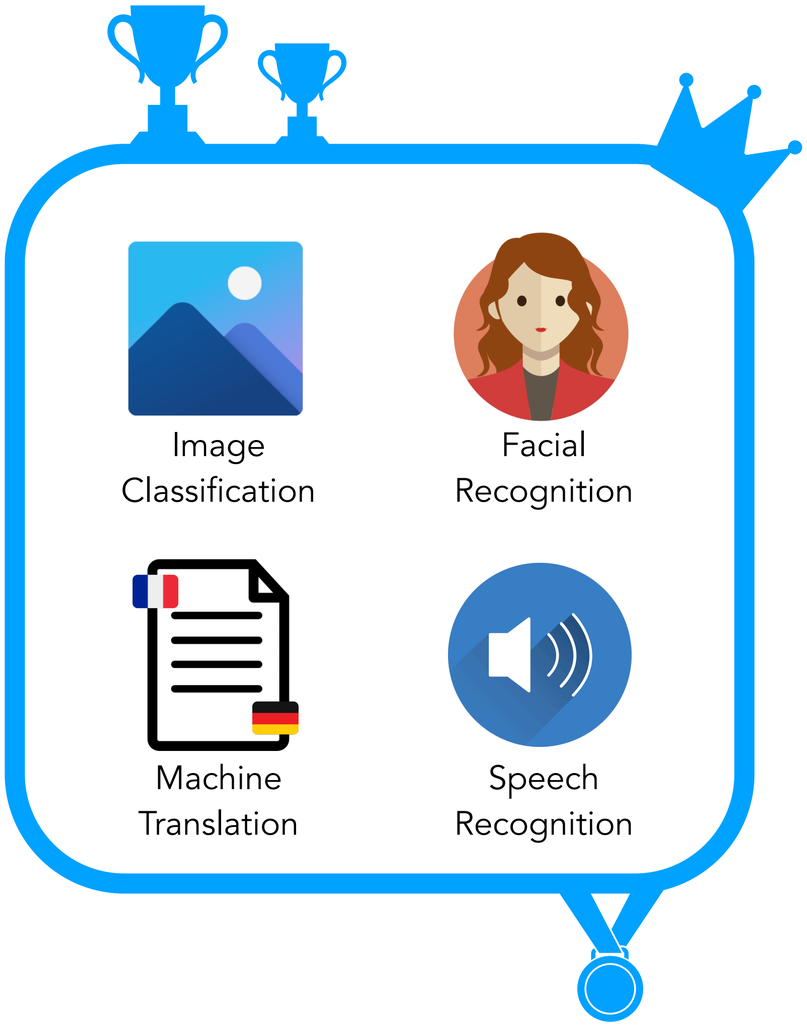


A Photorealistic Medieval Paparazzi of the Year 1100 AD
(Midjourney)

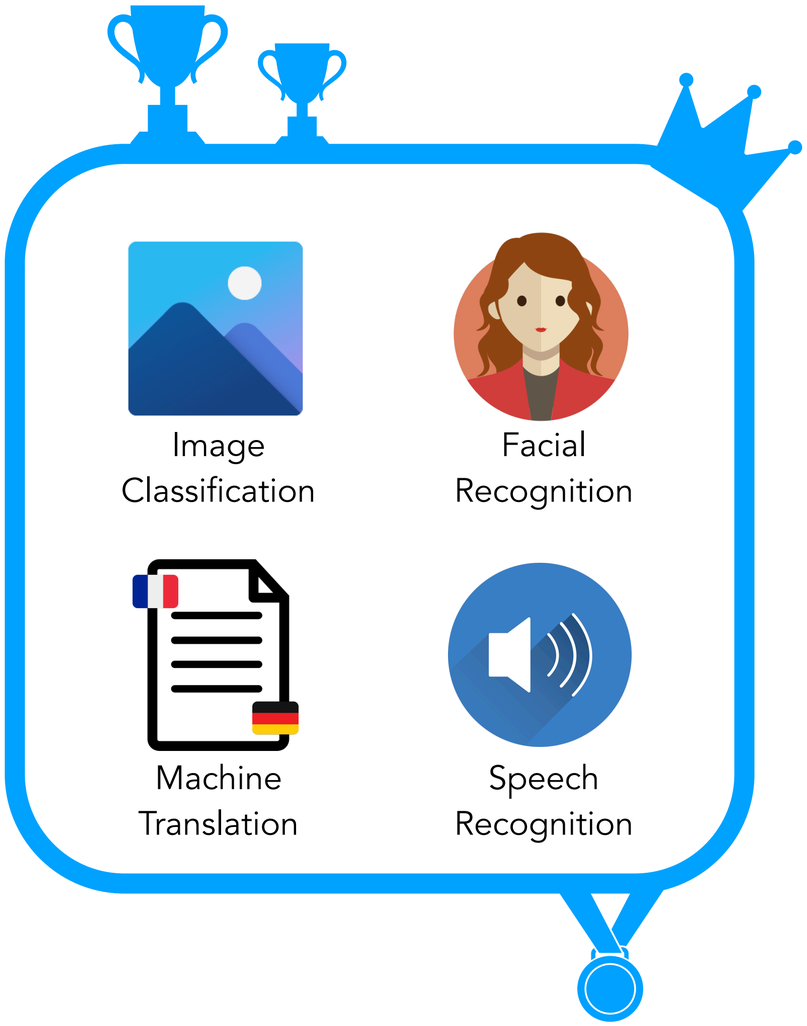
A Photorealistic Medieval Paparazzi of the Year 1100 AD
(Midjourney)

A Photorealistic Medieval Paparazzi of the Year 1100 AD
(Midjourney)

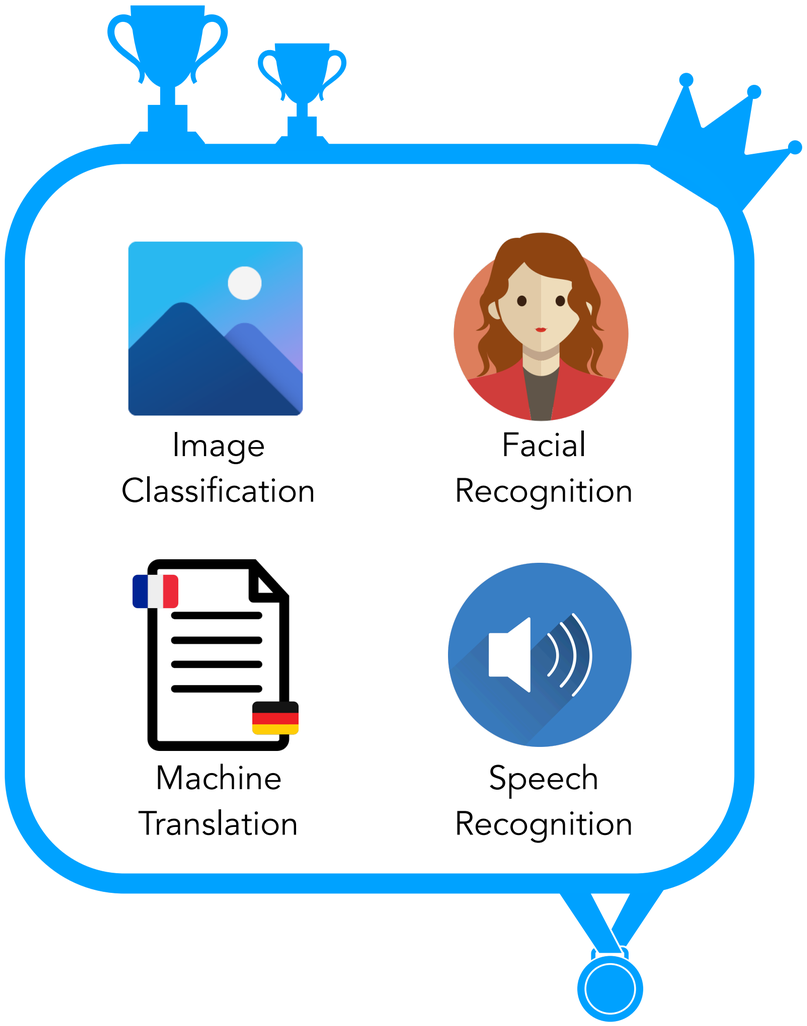
Machine Learning Revolution



Machine Learning Revolution

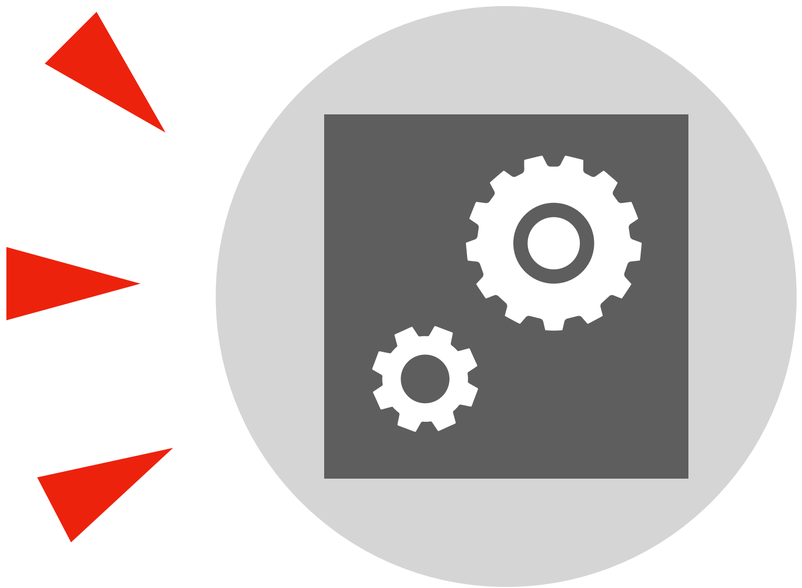


Machine Learning Revolution



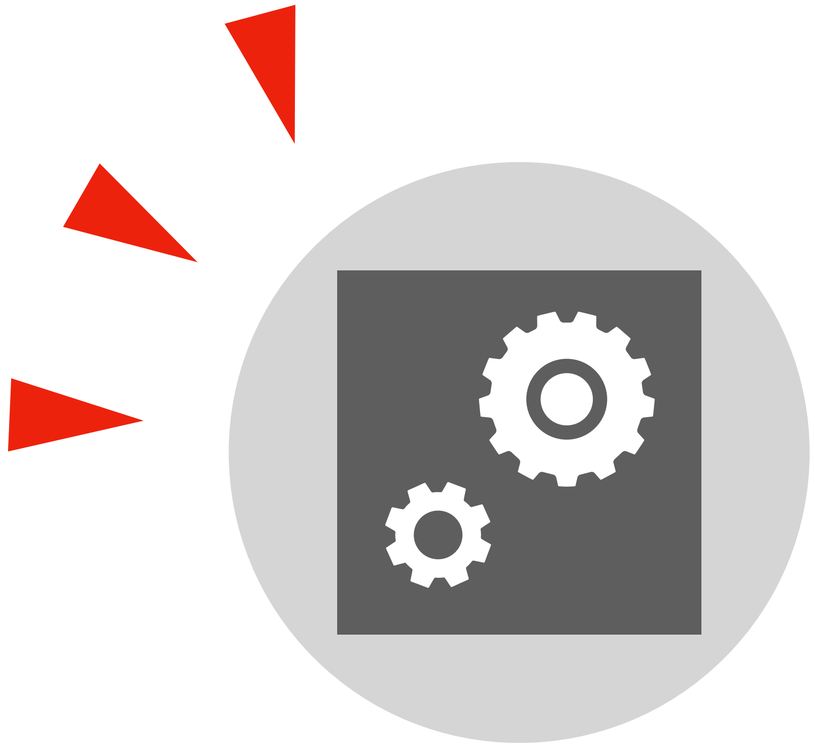
Maybe the conditions aren't ready yet?

Security *is* Adversarial



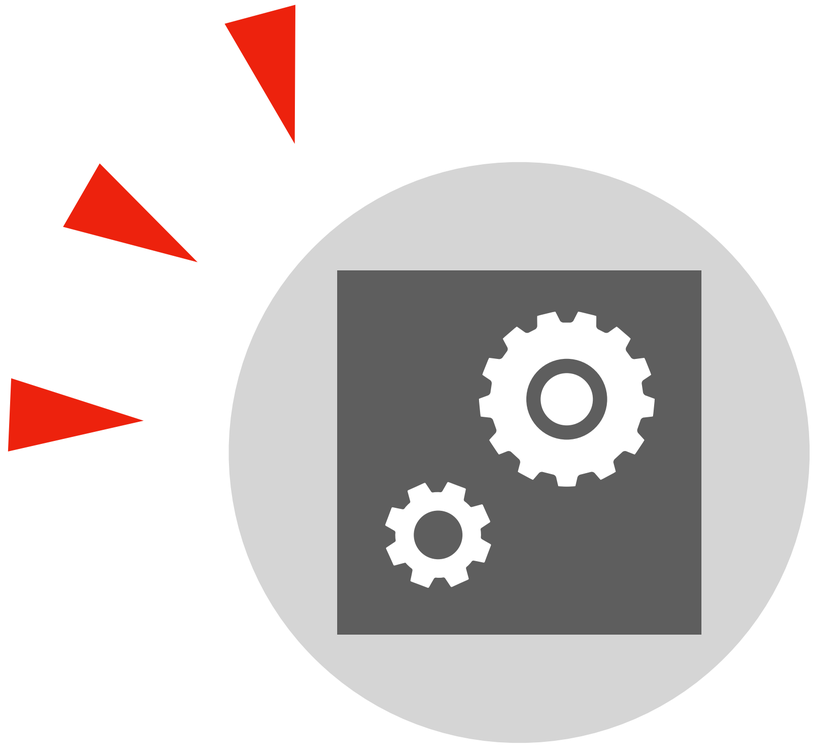
New detection systems trigger
an immediate response…

Security *is* Adversarial

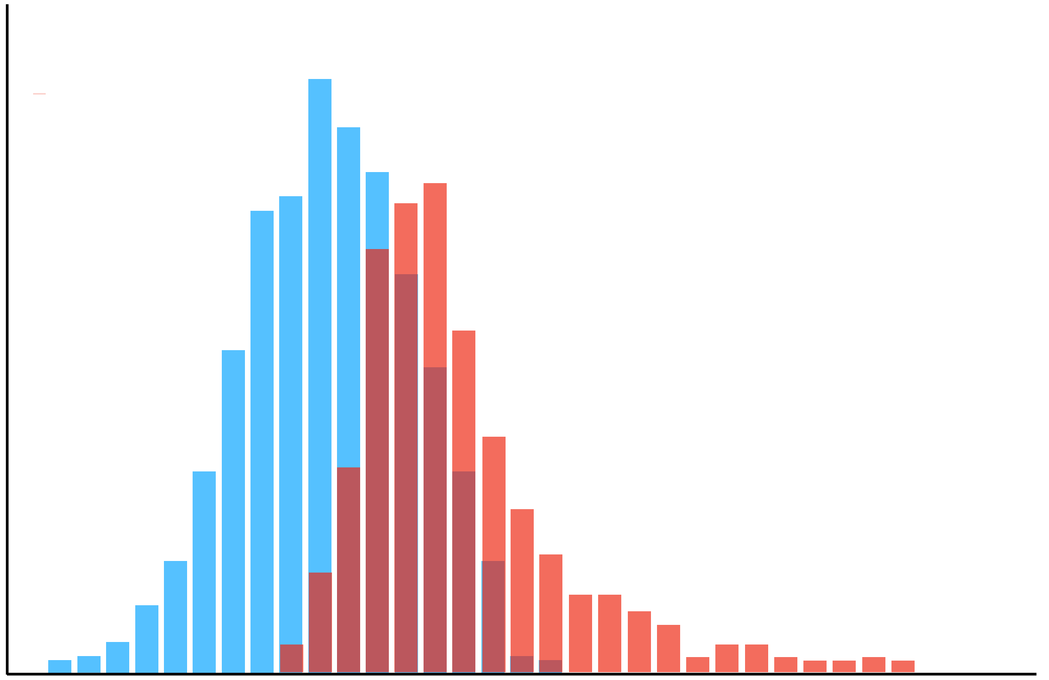


New detection systems trigger
an immediate response…

Security *is* Adversarial

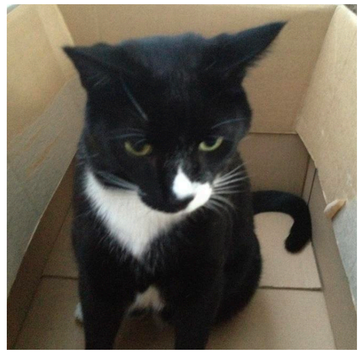


New detection systems trigger an immediate response…



…which causes dataset shifts, often violating the i.i.d. assumption

Security *is* Adversarial







Security *is* Adversarial

RABBIT!

Security *is* Adversarial

RABBIT!

Security *is* Adversarial



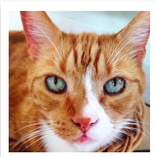

RABBIT!



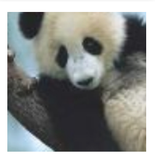
Training on ad fraud…

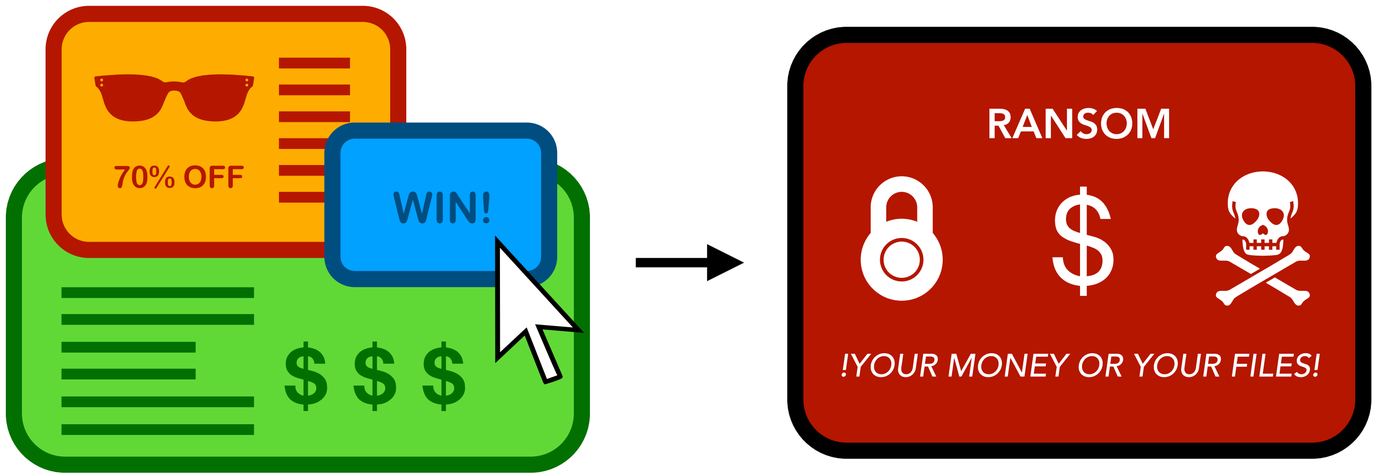
Security *is* Adversarial






RABBIT!

Training on ad fraud…

…attacks evolve at test time…

Security *is* Adversarial




RABBIT!

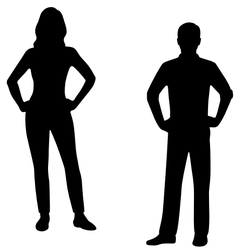


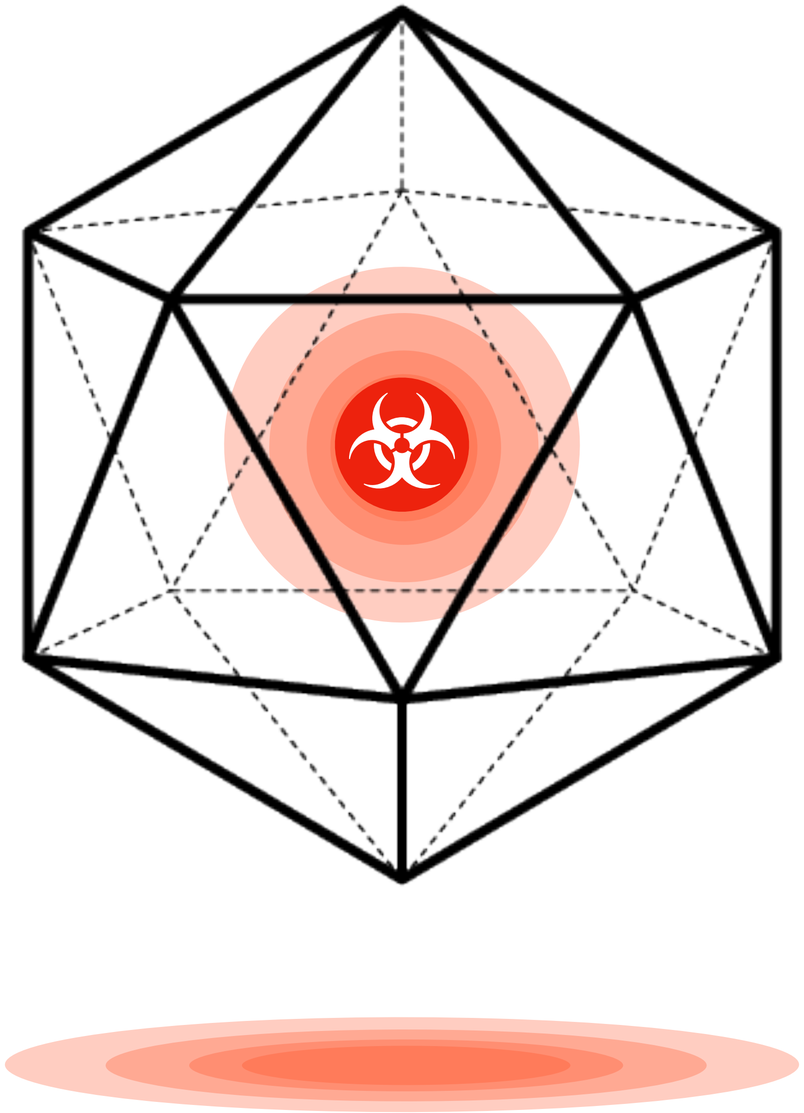
Training on ad fraud…

…attacks evolve at test time…

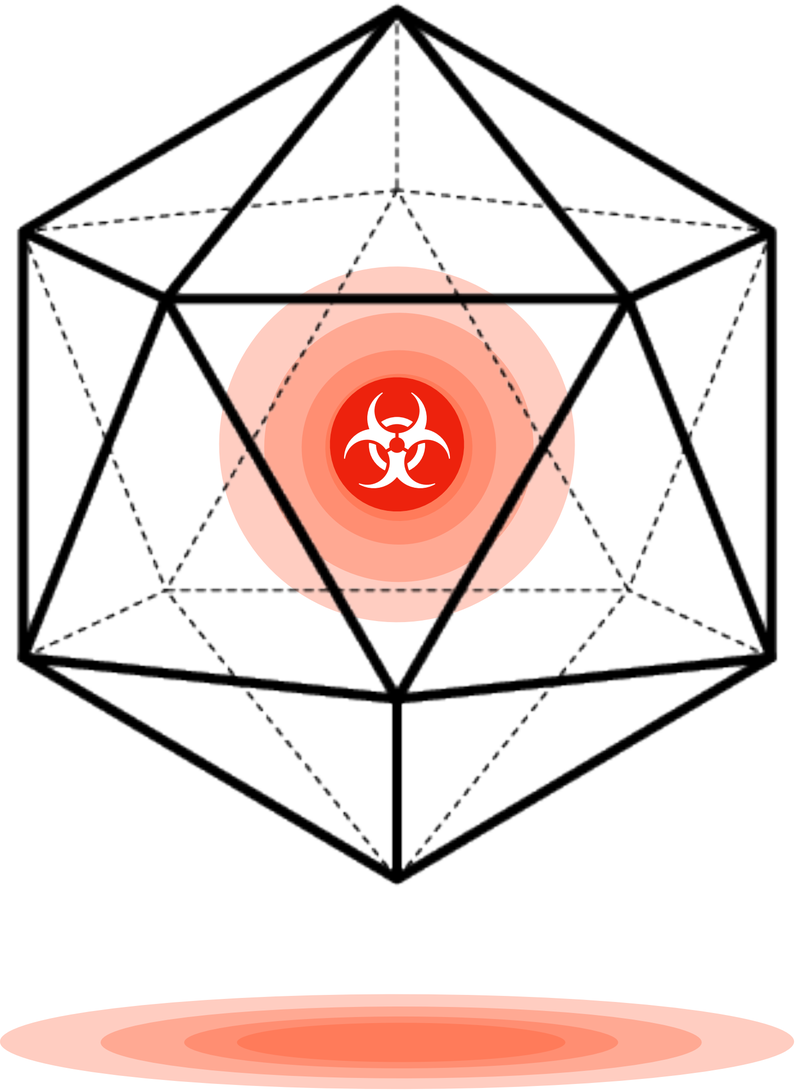
…prepare for the unknown


Security *is* Adversarial

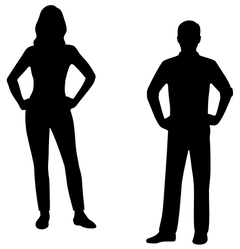




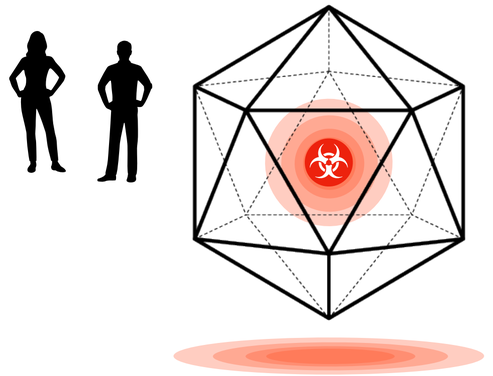
Security *is* Adversarial



Oh yeah, that's malware alright

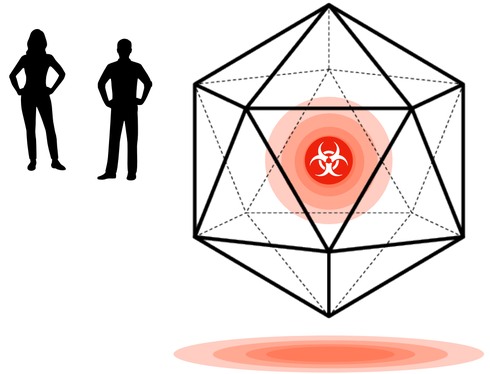


We need...



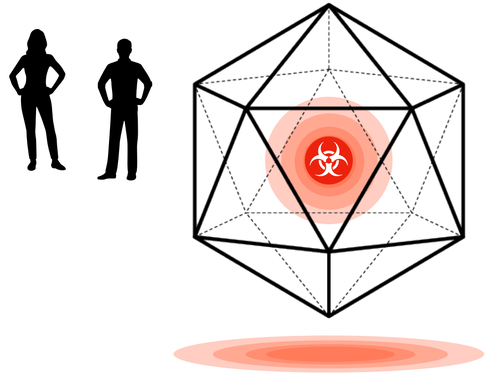
We need...

To understand and improve the effectiveness of machine learning methods for systems security in the presence of adversaries



We need...

To understand and improve the effectiveness of machine learning methods for systems security in the presence of adversaries

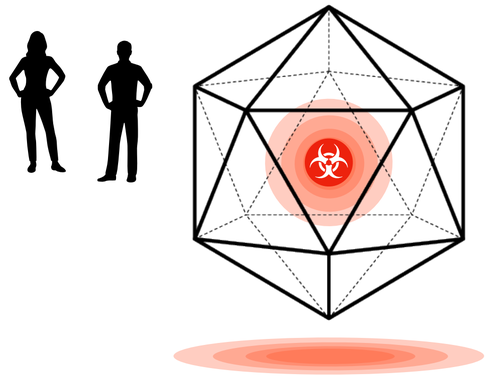


Representation of problem space objects (e.g., programs) results in a **semantic gap**

- It makes designing attacks and defenses more challenging
- It leaves room for adversarial manipulation
- It challenges the identification of causal vs non-causal (spurious) features

We need...

To understand and improve the effectiveness of machine learning methods for systems security in the presence of adversaries



Representation of problem space objects (e.g., programs) results in a **semantic gap**

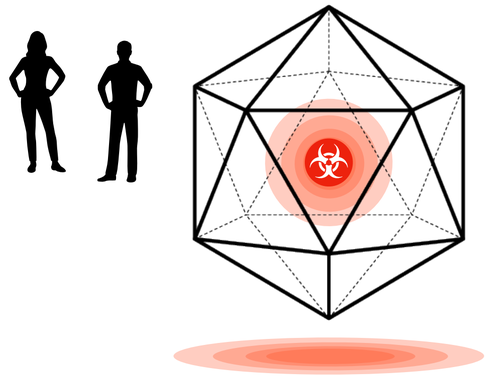
- It makes designing attacks and defenses more challenging
- It leaves room for adversarial manipulation
- It challenges the identification of causal vs non-causal (spurious) features

Effectiveness of ML for systems security is intertwined with the **underlying abstractions**, e.g., program analyses, to represent objects

- This affects robustness to adversarial drift, explainability, costs, and performance

We need...

To understand and improve the effectiveness of machine learning methods for systems security in the presence of adversaries



Representation of problem space objects (e.g., programs) results in a **semantic gap**

- It makes designing attacks and defenses more challenging
- It leaves room for adversarial manipulation
- It challenges the identification of causal vs non-causal (spurious) features

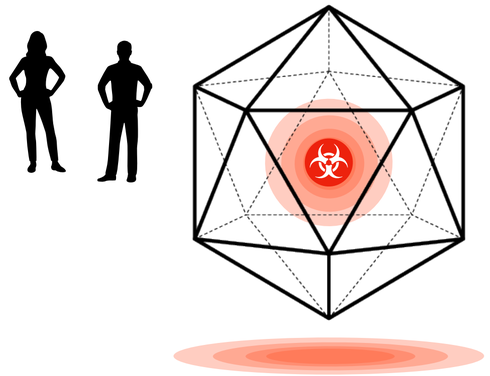
Effectiveness of ML for systems security is intertwined with the **underlying abstractions**, e.g., program analyses, to represent objects

- This affects robustness to adversarial drift, explainability, costs, and performance

Is Trustworthy AI for systems security possible?

We need...

To understand and improve the effectiveness of machine learning methods for systems security in the presence of adversaries



Representation of problem space objects (e.g., programs) results in a **semantic gap**

- It makes designing attacks and defenses more challenging
- It leaves room for adversarial manipulation
- It challenges the identification of causal vs non-causal (spurious) features

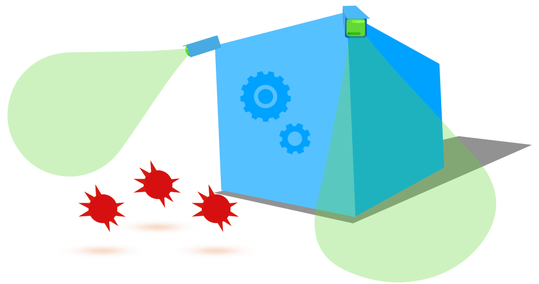
Effectiveness of ML for systems security is intertwined with the **underlying abstractions**, e.g., program analyses, to represent objects

- This affects robustness to adversarial drift, explainability, costs, and performance

Is Trustworthy AI for systems security possible?

We already know the answer!
But wait, there's a plan...

Outline

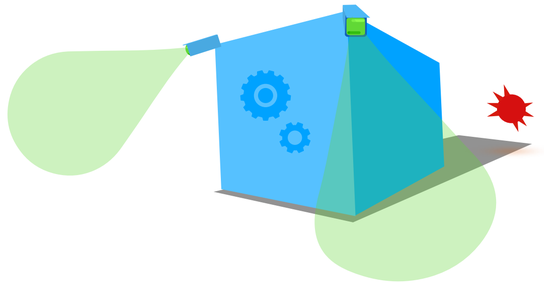


Outline

Focus

Adversarial ML evasion attacks against malware classifiers

- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives

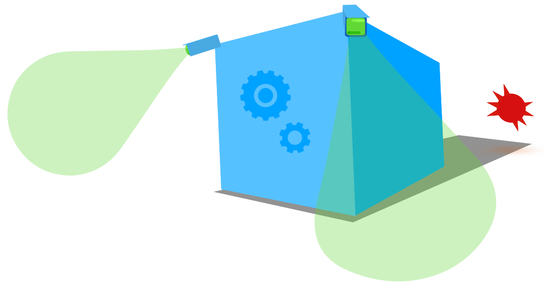


Outline

Focus

Adversarial ML evasion attacks against malware classifiers

- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives

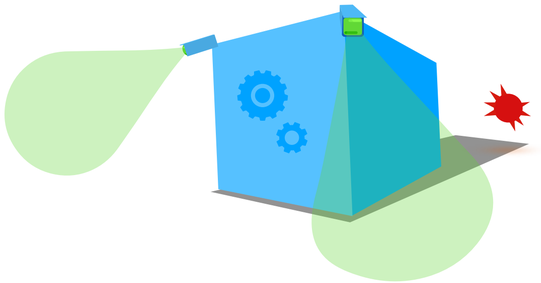


Outline

Focus

Adversarial ML evasion attacks against malware classifiers

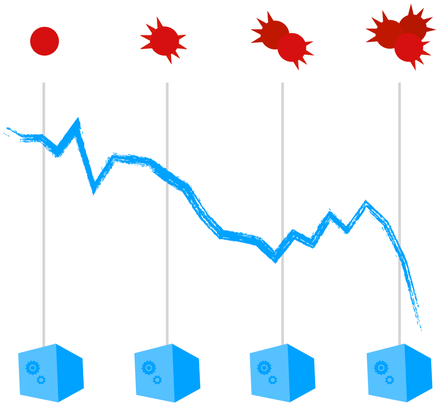
- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives



Bigger Picture

Drifting scenarios caused by threats evolving over time

- How concept drift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics

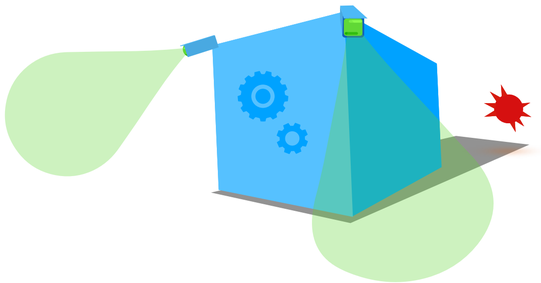


Outline

Focus

Adversarial ML evasion attacks against malware classifiers

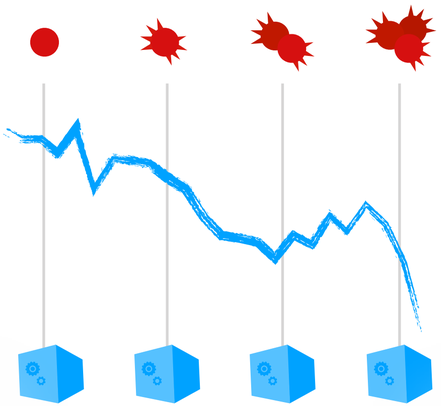
- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives



Bigger Picture

Drifting scenarios caused by threats evolving over time

- How concept drift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics

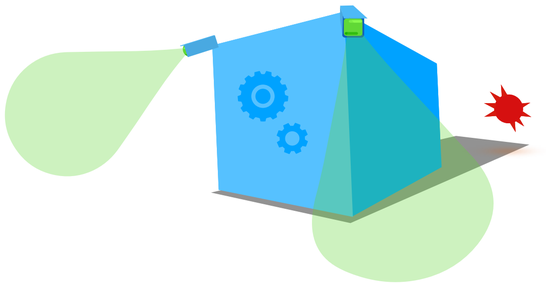


Outline

Focus

Adversarial ML evasion attacks against malware classifiers

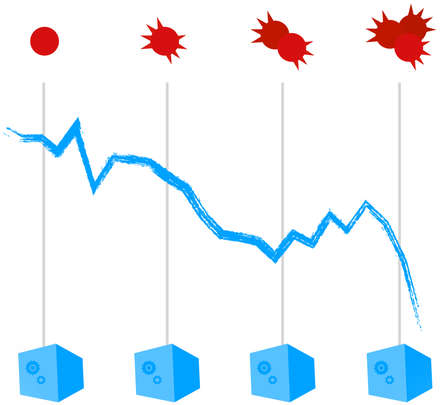
- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives



Bigger Picture

Drifting scenarios caused by threats evolving over time

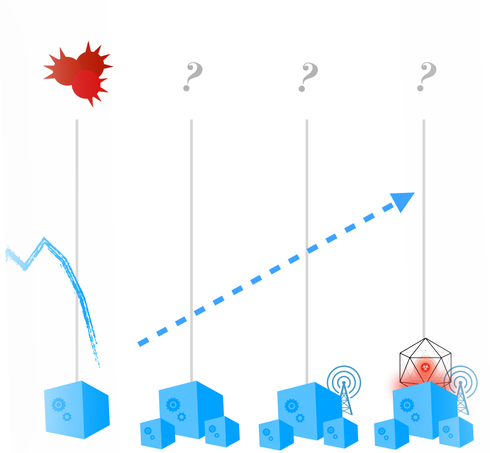
- How concept drift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics



Looking Ahead

Quo vadis?

- Discussion of the future of trustworthy ML for system security
- Robust feature development, universal adversarial perturbations, realizable backdoors, drift forecasting, and the role of abstractions towards the Platonic ideal of interesting behaviors

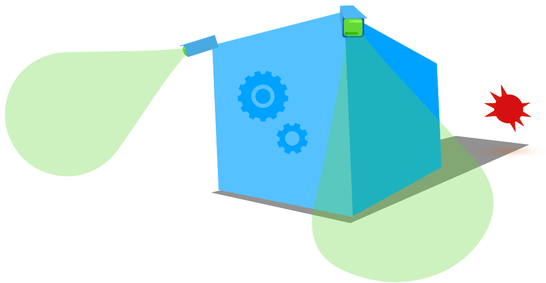


Outline

Focus

Adversarial ML evasion attacks against malware classifiers

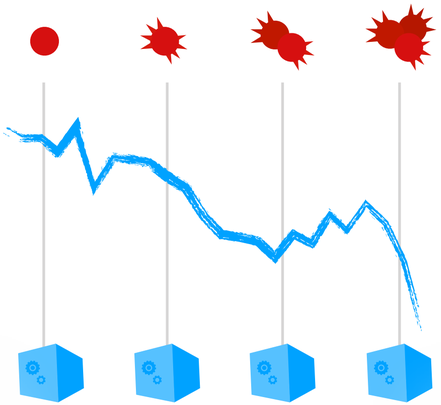
- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives



Bigger Picture

Drifting scenarios caused by threats evolving over time

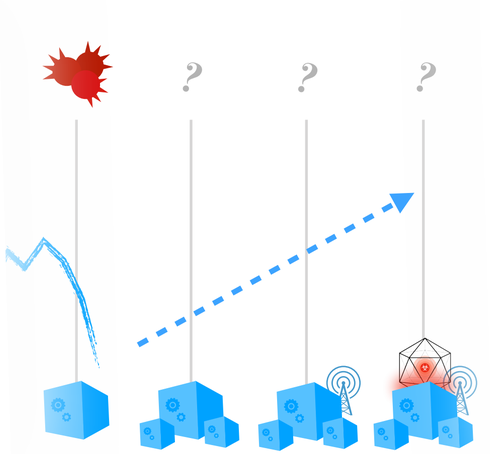
- How concept drift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics



Looking Ahead

Quo vadis?

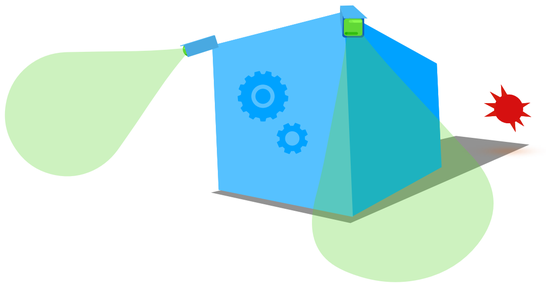
- Discussion of the future of trustworthy ML for system security
- Robust feature development, universal adversarial perturbations, realizable backdoors, drift forecasting, and the role of abstractions towards the Platonic ideal of interesting behaviors



Outline

Focus

Adversarial ML evasion attacks against malware classifiers



- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives

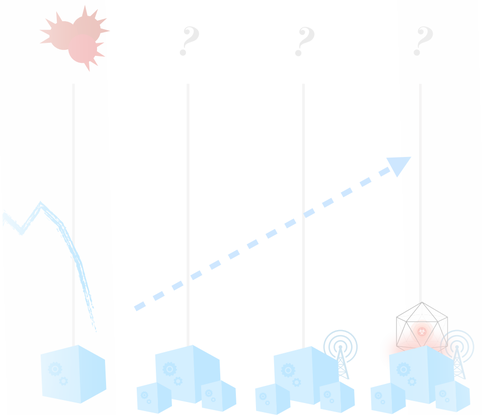
Bigger Picture

Drifting scenarios caused by threats evolving over time

- How concept drift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics

Looking Ahead

Quo vadis?



- Discussion of the future of trustworthy ML for system security
- Robust feature development, universal adversarial perturbations, realizable backdoors, drift forecasting, and the role of abstractions towards the Platonic ideal of interesting behaviors

A Dystopian Future...

Pandas are forbidden!

Guilty of being too cute!

A Dystopian Future...

Pandas are forbidden!

Guilty of being too cute!

A Dystopian Future...

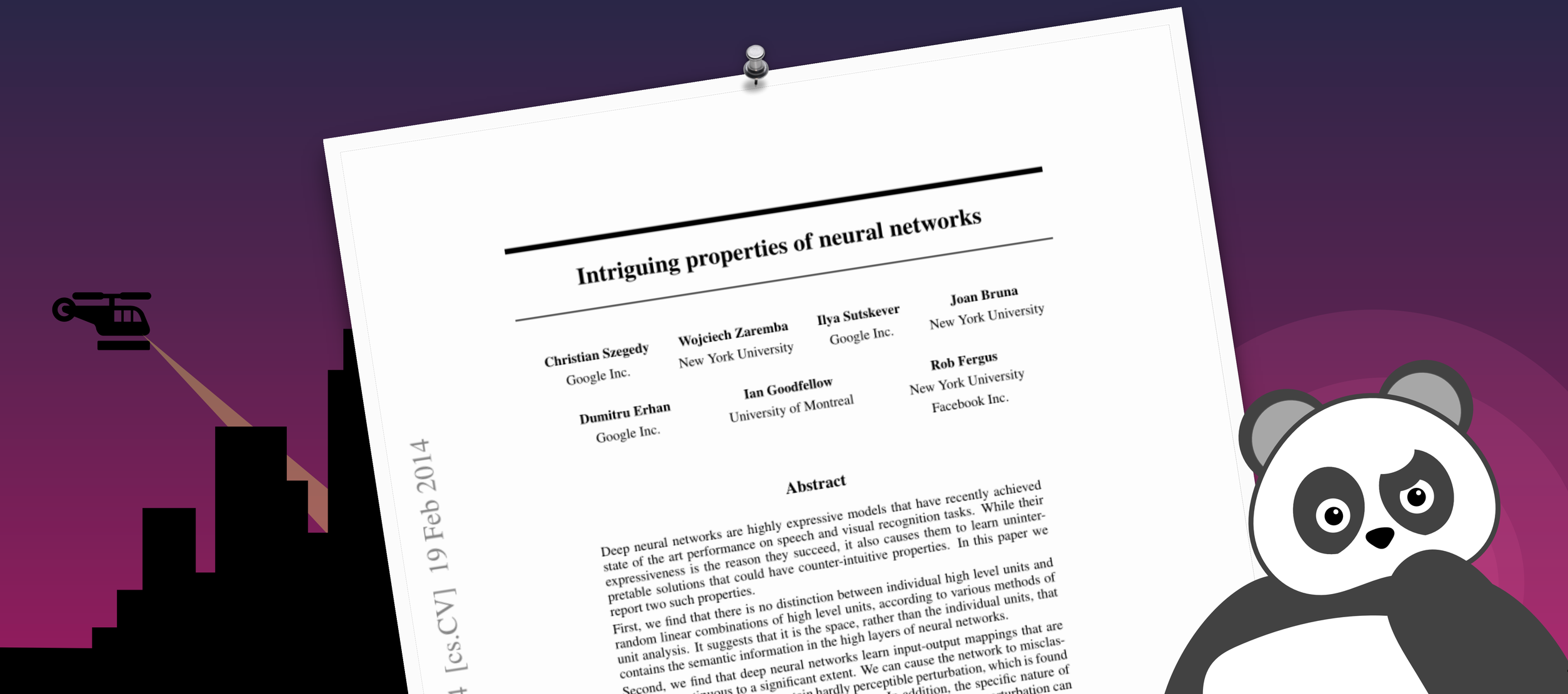
Pandas are forbidden!

Guilty of being too cute!

Luckily, pandas are fluent in math…

Luckily, pandas are fluent in math…

Luckily, pandas are fluent in math…



Intriguing properties of neural networks

Christian Szegedy
Google Inc.

Wojciech Zaremba
New York University

Ilya Sutskever
Google Inc.

Joan Bruna
New York University

Dumitru Erhan
Google Inc.

Ian Goodfellow
University of Montreal

Rob Fergus
New York University
Facebook Inc.

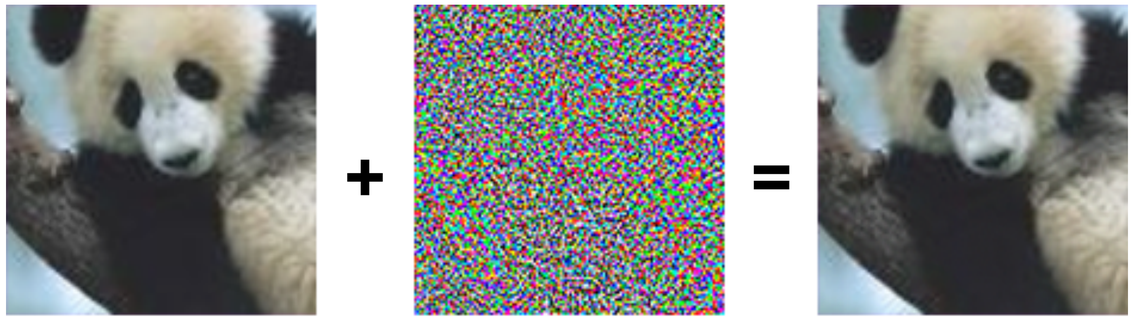
Abstract

Deep neural networks are highly expressive models that have recently achieved state of the art performance on speech and visual recognition tasks. While their expressiveness is the reason they succeed, it also causes them to learn uninterpretable solutions that could have counter-intuitive properties. In this paper we report two such properties.

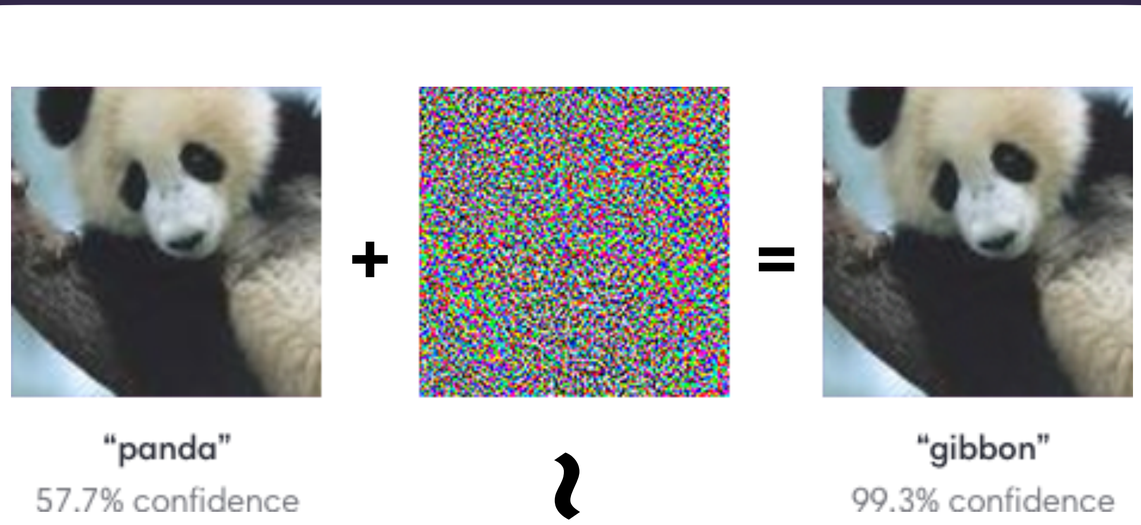
First, we find that there is no distinction between individual high level units and random linear combinations of high level units, according to various methods of unit analysis. It suggests that it is the space, rather than the individual units, that contains the semantic information in the high layers of neural networks.

Second, we find that deep neural networks learn input-output mappings that are

Luckily, pandas are fluent in math…



Luckily, pandas are fluent in math…

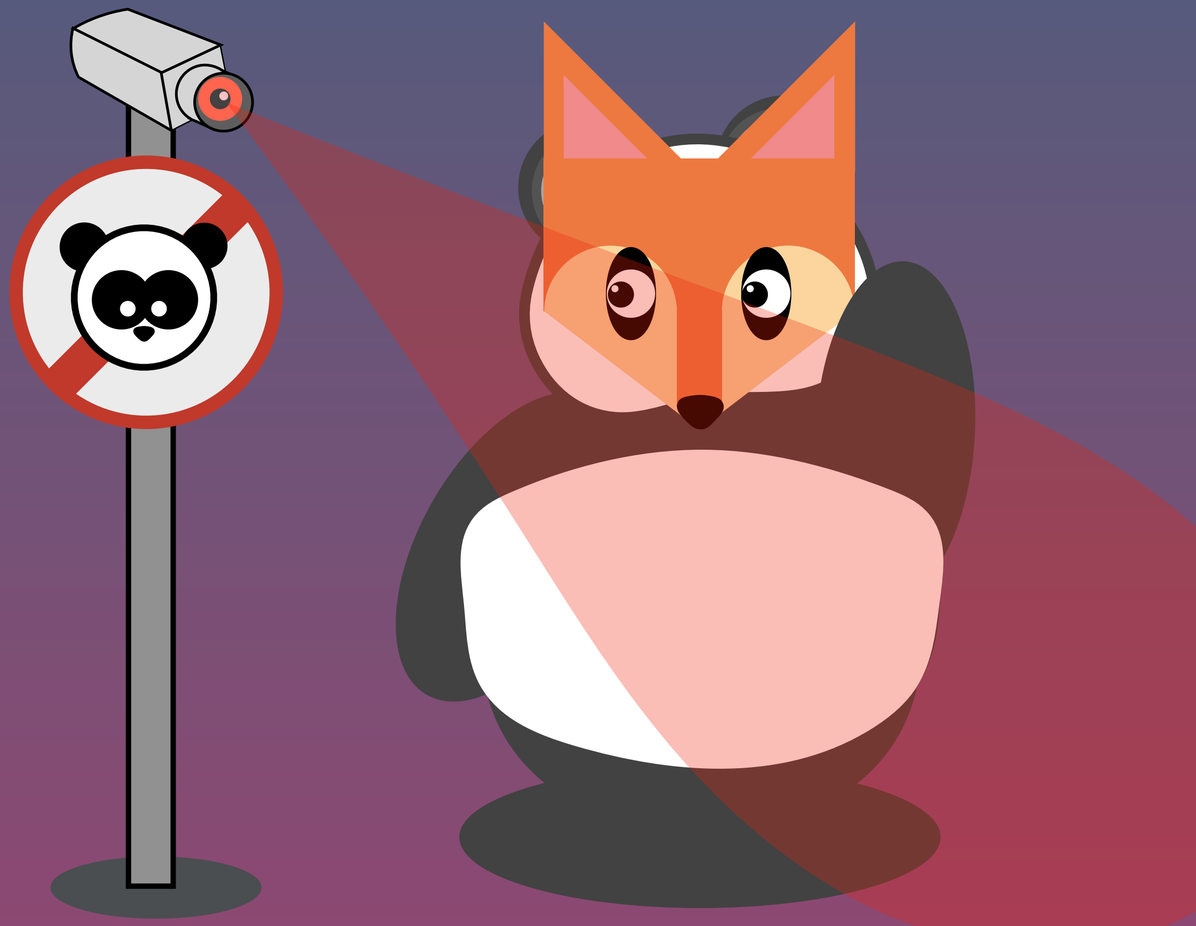


Feature-space
noise mask

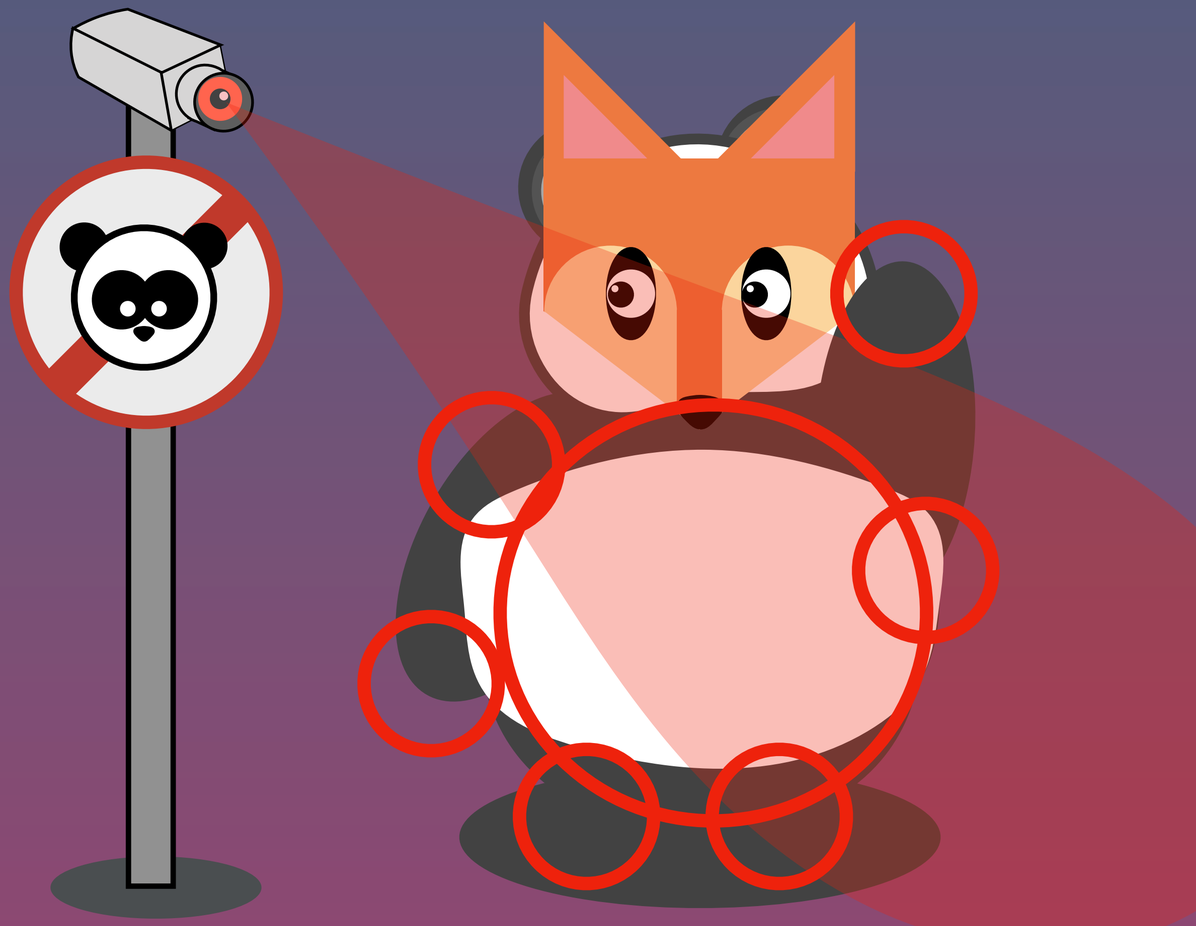
What happens in the **problem space**, i.e., the real world?

What happens in the **problem space**, i.e., the real world?

What happens in the **problem space**, i.e., the real world?



What happens in the **problem space**, i.e., the real world?

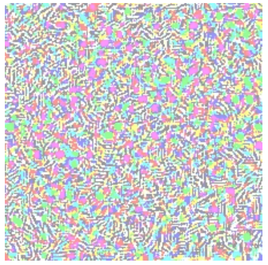


What happens in the **problem space**, i.e., the real world?

Let’s Analyze What Happened

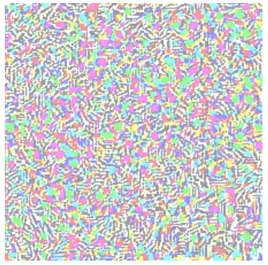
Let’s Analyze What Happened

Feature-Space Attacks

Original Image		Perturbation		Adv. Image
 “panda” 57.7%	+	 *imperceptible noise*	=	 “gibbon” 99.3%
x		δ		$x + \delta$

Let's Analyze What Happened

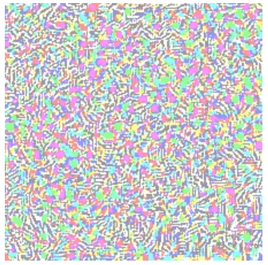
Feature-Space Attacks

Original Image		Perturbation		Adv. Image
 "panda" 57.7%	+	 *imperceptible noise*	=	 "gibbon" 99.3%
x		δ		$x + \delta$

Optimization | $\textbf{minimize}_{\delta} \; ||\delta||_p + c \cdot f(x + \delta)$

Let's Analyze What Happened

Feature-Space Attacks

Original Image		Perturbation		Adv. Image
	+		=	
"panda" 57.7%		*imperceptible noise*		"gibbon" 99.3%
x		δ		$x + \delta$

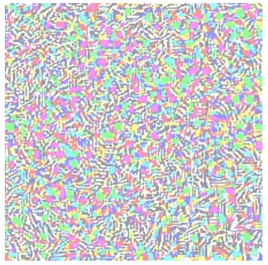
Optimization

$$\text{minimize}_{\delta} \; ||\delta||_p + c \cdot f(x+\delta)$$

Pixel Perturbations

Let's Analyze What Happened

Feature-Space Attacks

Original Image		Perturbation		Adv. Image
	+		=	
"panda" 57.7%		*imperceptible noise*		"gibbon" 99.3%
x		δ		$x + \delta$

Optimization

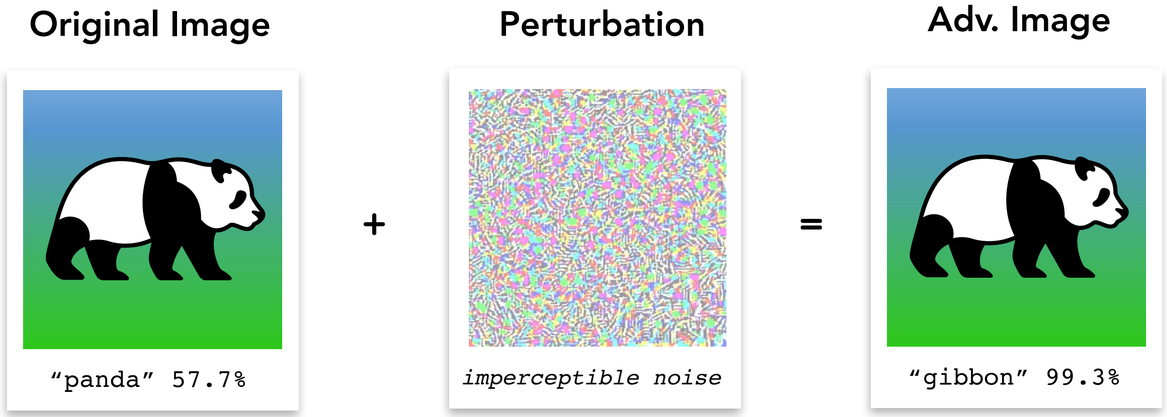
$$\text{minimize}_{\delta} \; ||\delta||_p + c \cdot f(x+\delta)$$

Pixel Perturbations

Loss of Target Class

Let's Analyze What Happened

Feature-Space Attacks



Optimization

$$\text{minimize}_{\delta} \; ||\delta||_p + c \cdot f(x+\delta)$$

Pixel Perturbations

Loss of Target Class

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Let's Analyze What Happened

Problem-Space Attacks

	Original App (z)	"Perturbation"	Adversarial App (z')
Problem Space	 "malware" 57.7%	+  ?	=  "goodware" 95.7%
Feature Space	x	δ	$x + \delta$

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Let's Analyze What Happened

Problem-Space Attacks

Problem Space

Original App (z)		"Perturbation"		Adversarial App (z')
 "malware" 57.7%	+		=	 "goodware" 95.7%

Feature Space

x $\quad$ δ $\quad$ $x + \delta$

Optimization $\quad$ $\textbf{minimize}_{\delta} \; ||\delta||_p + c \cdot f(x+\delta)$

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Let's Analyze What Happened

Problem-Space Attacks

Problem Space

Feature Space

x $\quad$ δ $\quad$ $x + \delta$

Optimization

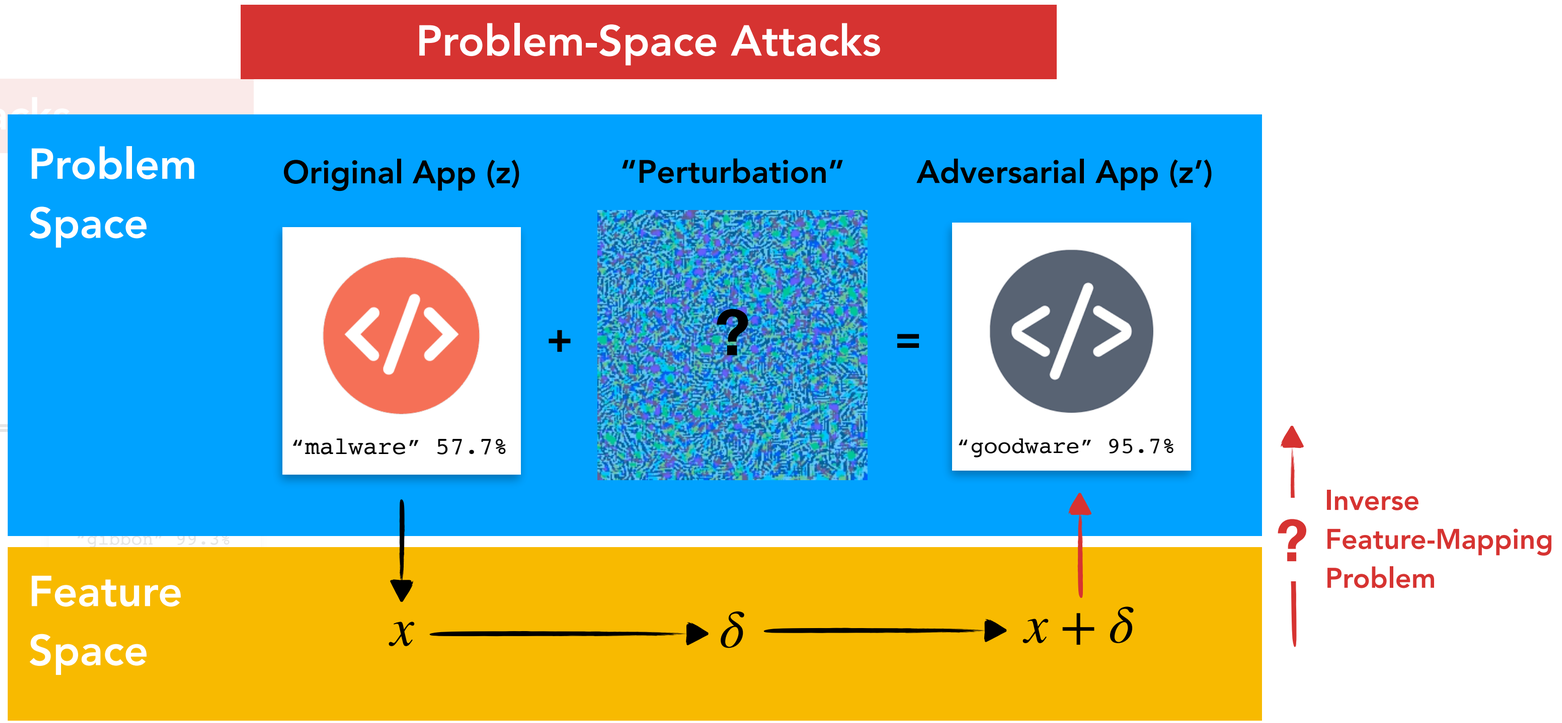
$$\text{minimize}_{\delta} \; ||\delta||_p + c \cdot f(x+\delta)$$

Constraints

- Is it realistic/plausible?
- Does it crash?
- Does it preserve malicious functionality?
- … are there **"general" constraints**?

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Let's Analyze What Happened



Optimization | $\textbf{minimize}_{\delta} \; ||\delta||_p + c \cdot f(x+\delta)$

Constraints

- Is it realistic/plausible?
- Does it crash?
- Does it preserve malicious functionality?
- … are there **"general" constraints**?

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Formalization

Problem-Space Constraints

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Available Transformations

How can you alter problem-space objects?

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Available Transformations

How can you alter problem-space objects?

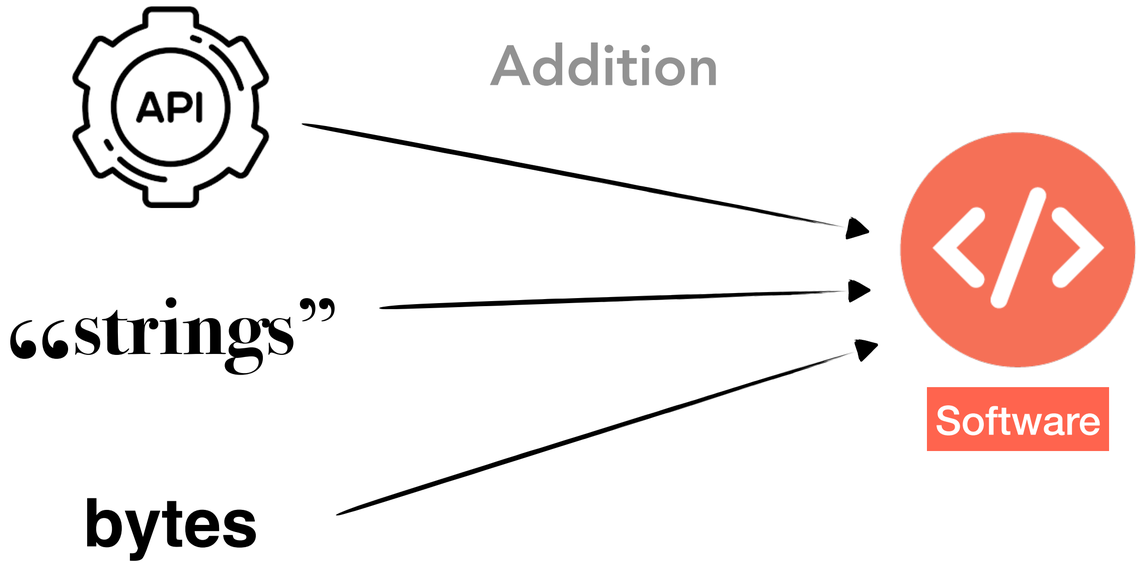
Software

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Available Transformations

How can you alter problem-space objects?

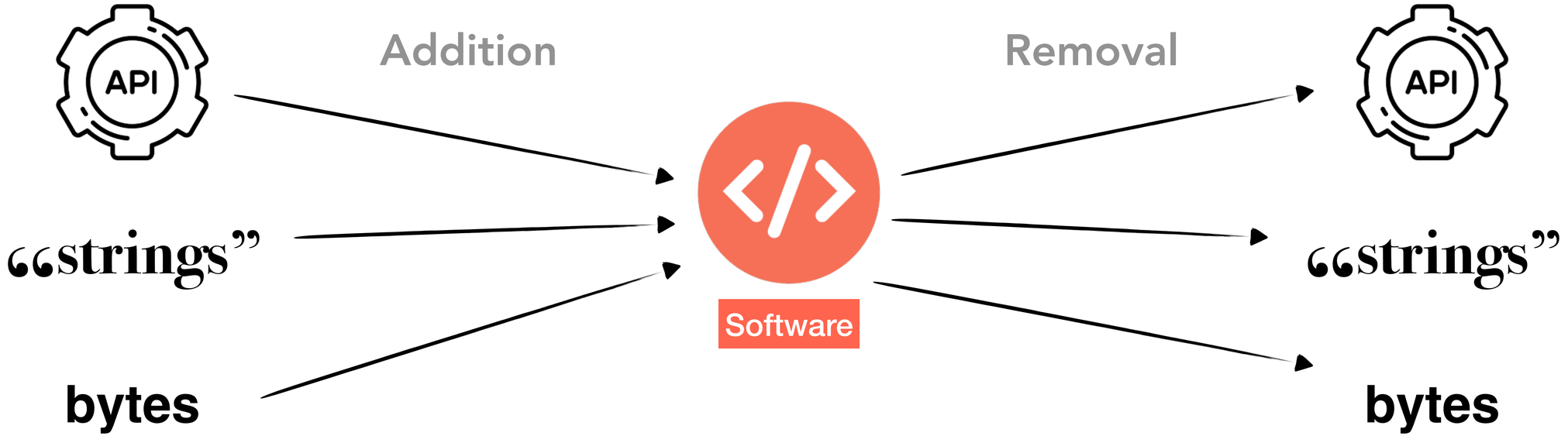


[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Available Transformations

How can you alter problem-space objects?

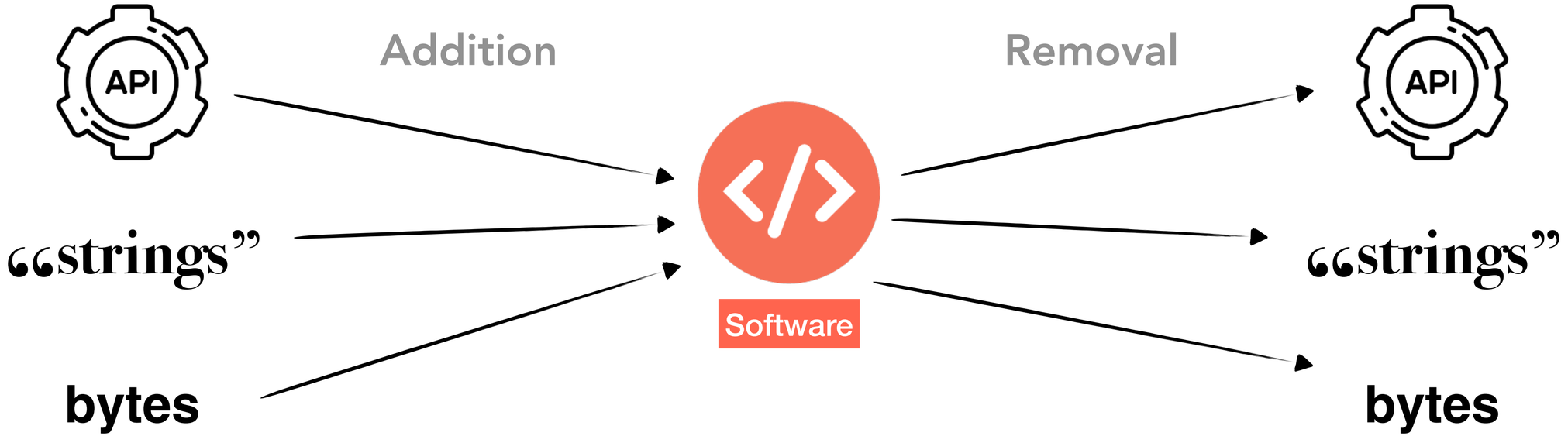


[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Available Transformations

How can you alter problem-space objects?

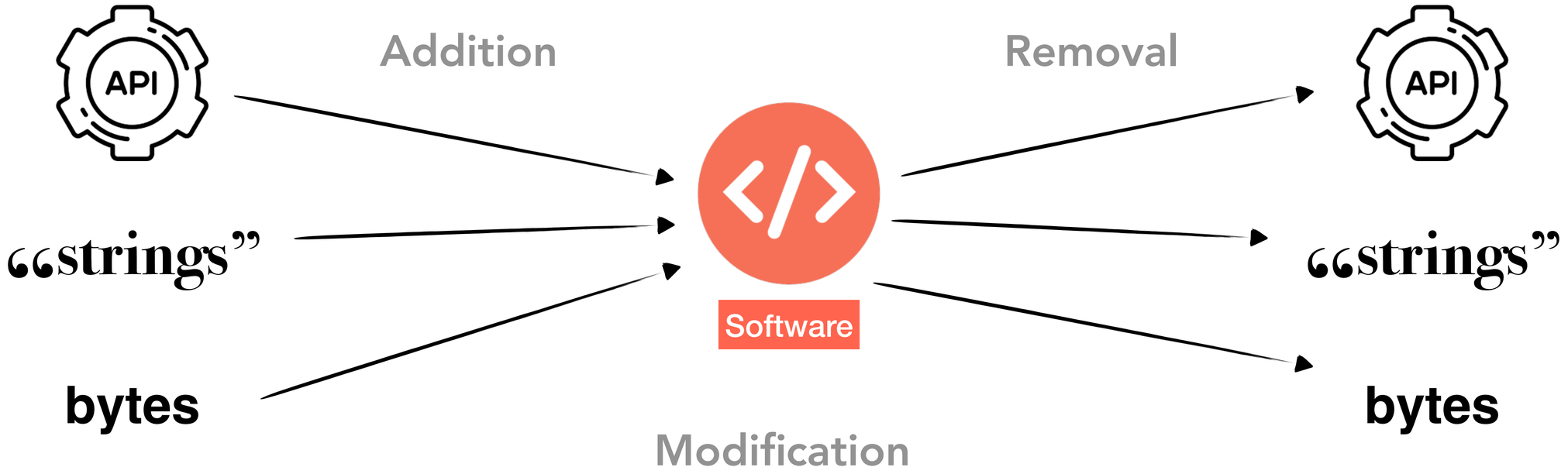


[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Available Transformations

How can you alter problem-space objects?



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Available Transformations

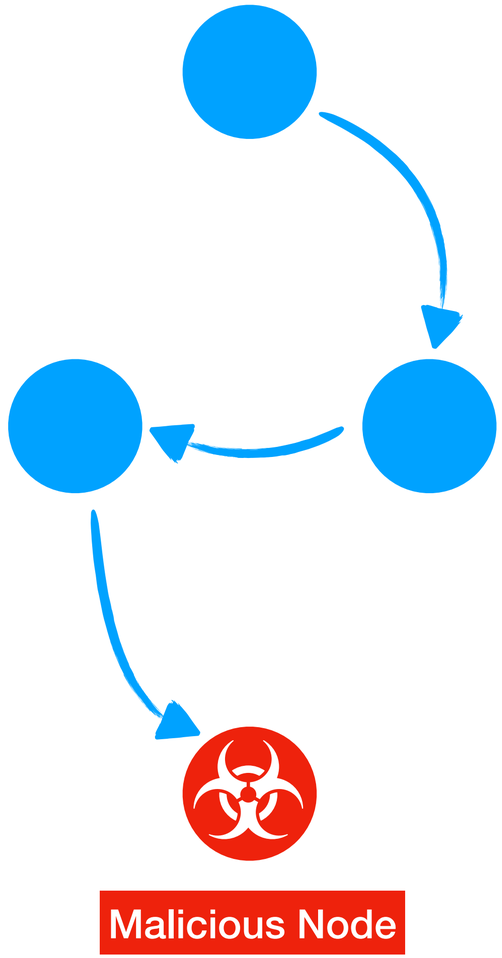
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Preserved Semantics

Available Transformations

Which semantics do you preserve? How?
Which automatic tests can verify it?



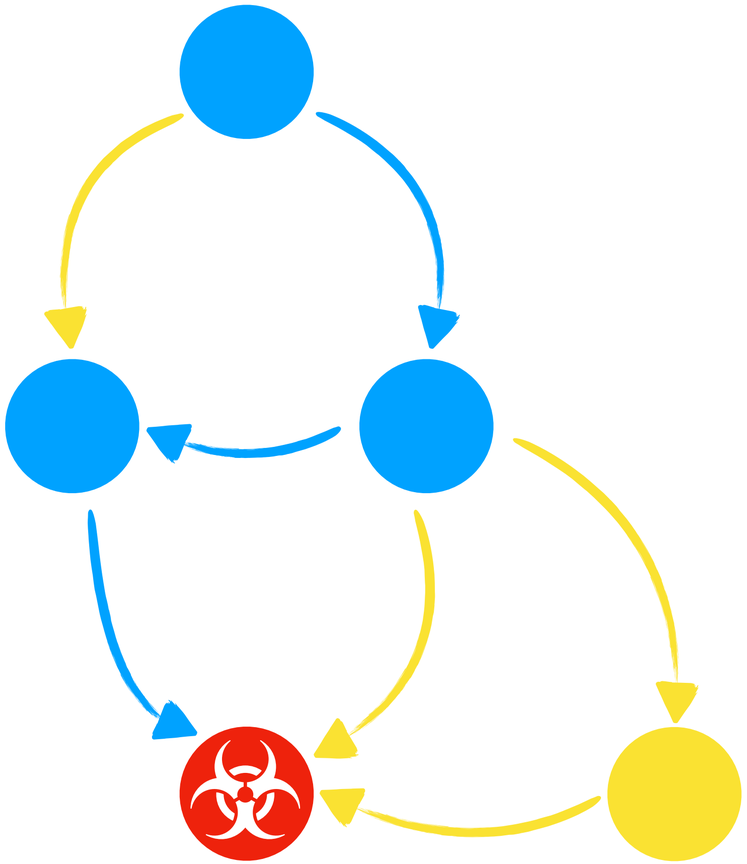
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Preserved Semantics

Available Transformations

Which semantics do you preserve? How?
Which automatic tests can verify it?



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

Preserved Semantics

Available Transformations

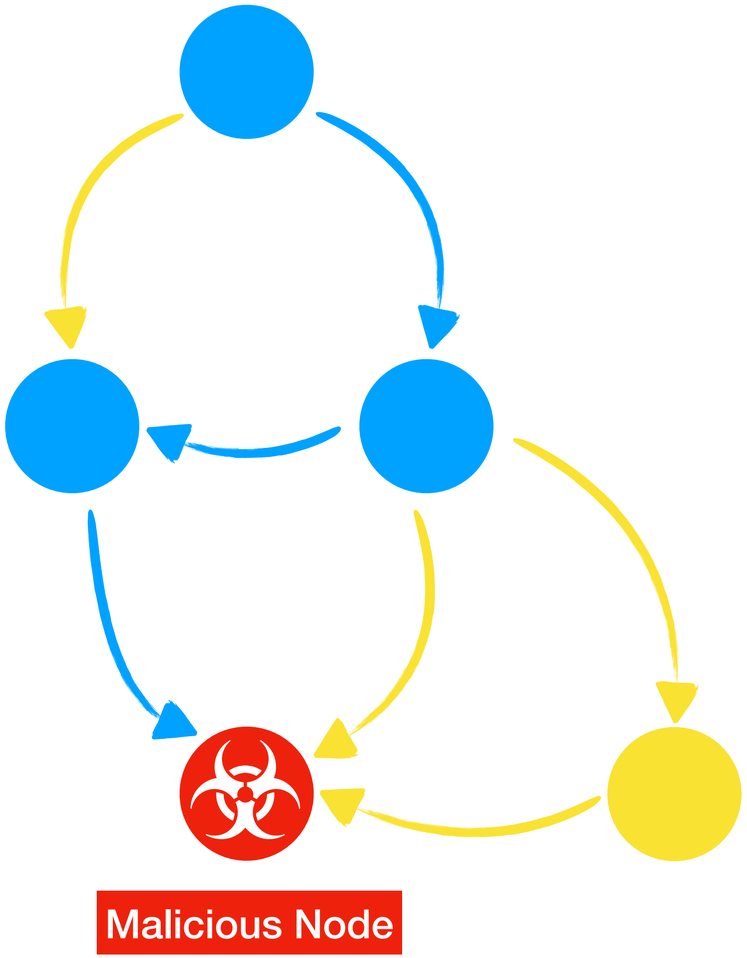
Which semantics do you preserve? How?
Which automatic tests can verify it?

Test Suite

- Does it crash?
- Does it still communicate with CnC?
- Does it still encrypt the /home/ folder?

By Construction

- Add no-op operations
- Ensure it is not executed at runtime



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

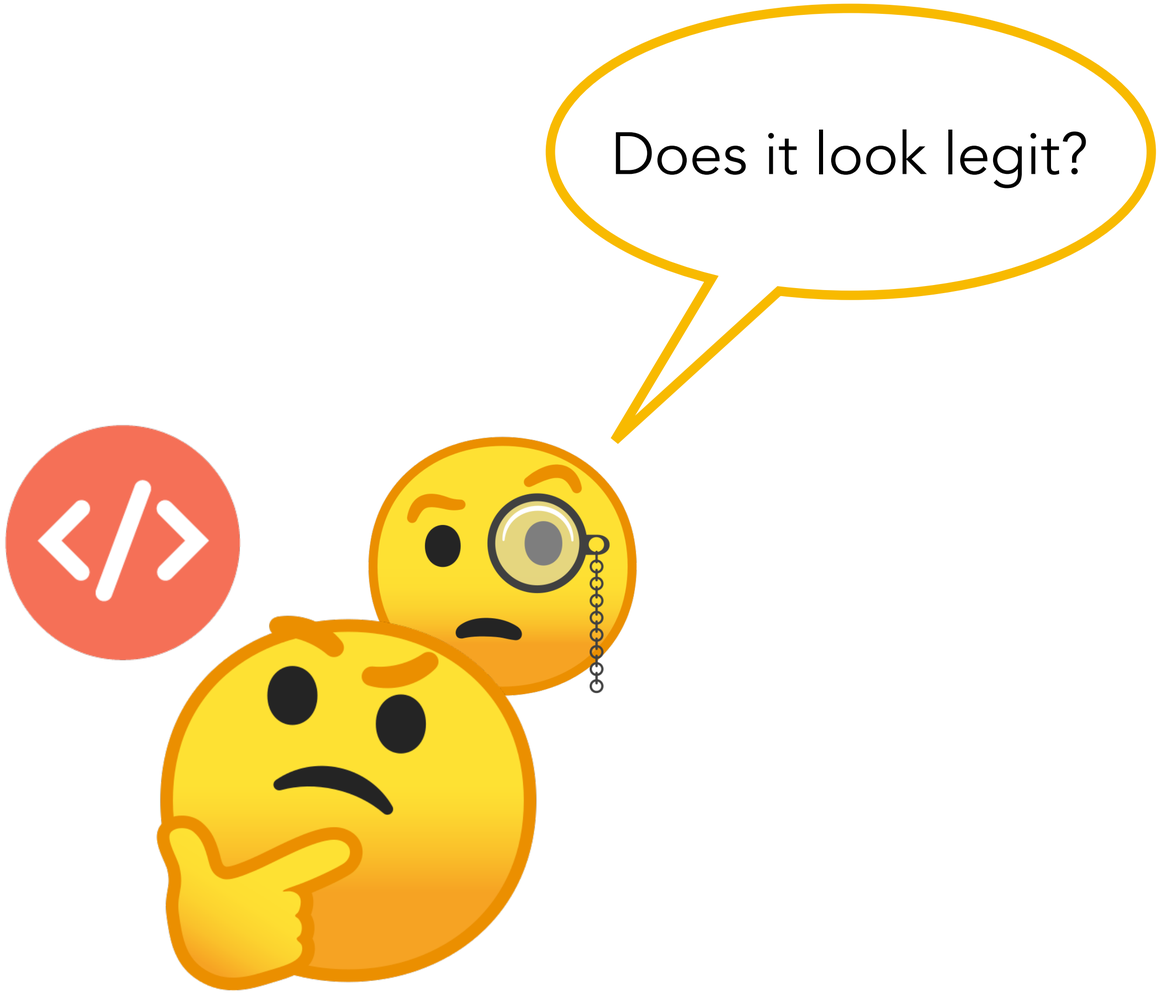
Preserved Semantics

Available Transformations

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

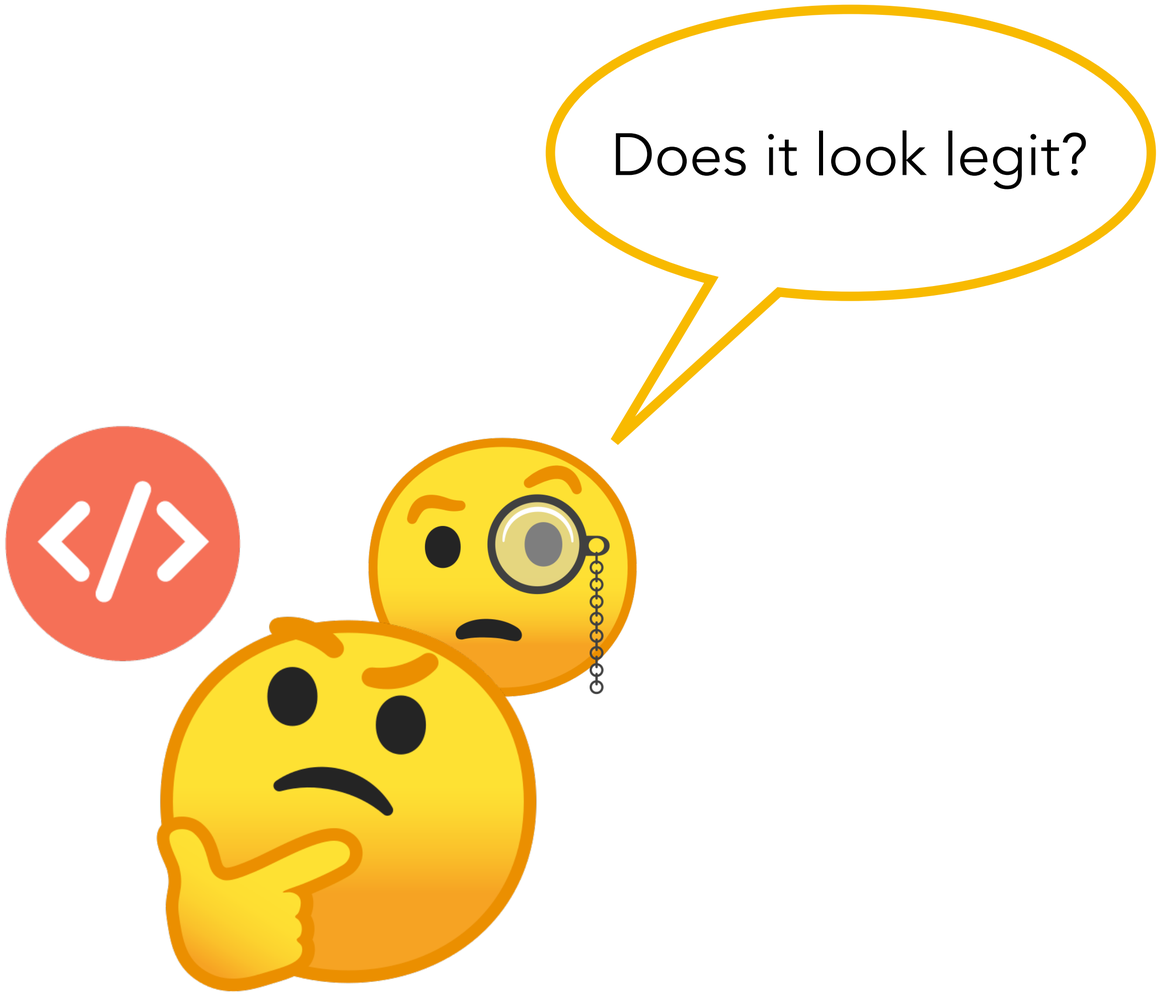
- **Plausibility**
- Preserved Semantics
- Available Transformations



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

- **Plausibility**
- Preserved Semantics
- Available Transformations



Test Suite

- User studies
- Automated heuristics

By Construction

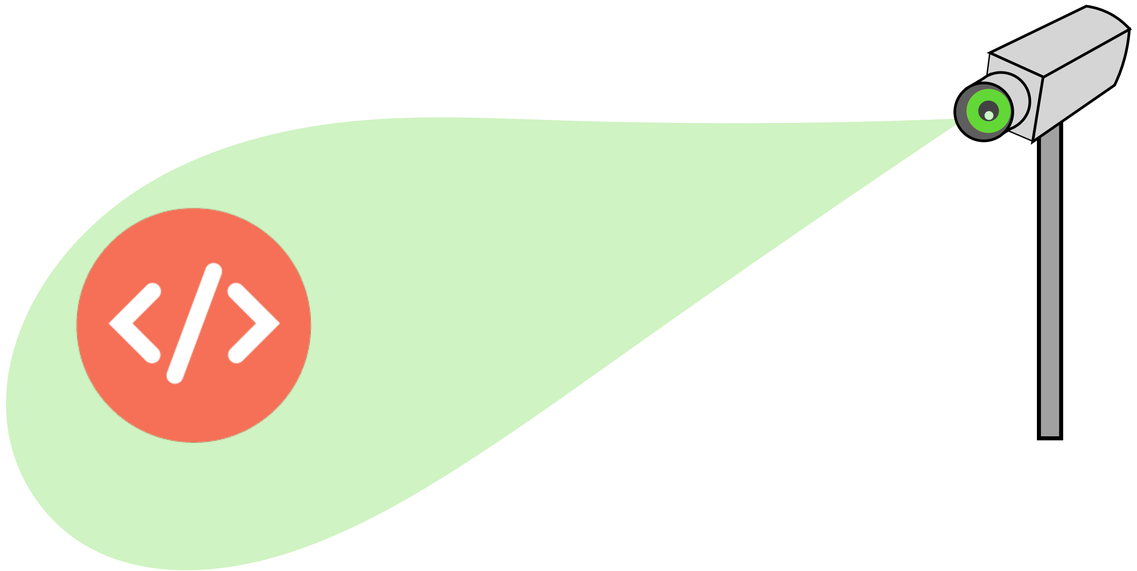
- Taking precautions during mutation

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

- **Plausibility**
- Preserved Semantics
- Available Transformations

Which preprocessing are you considering?

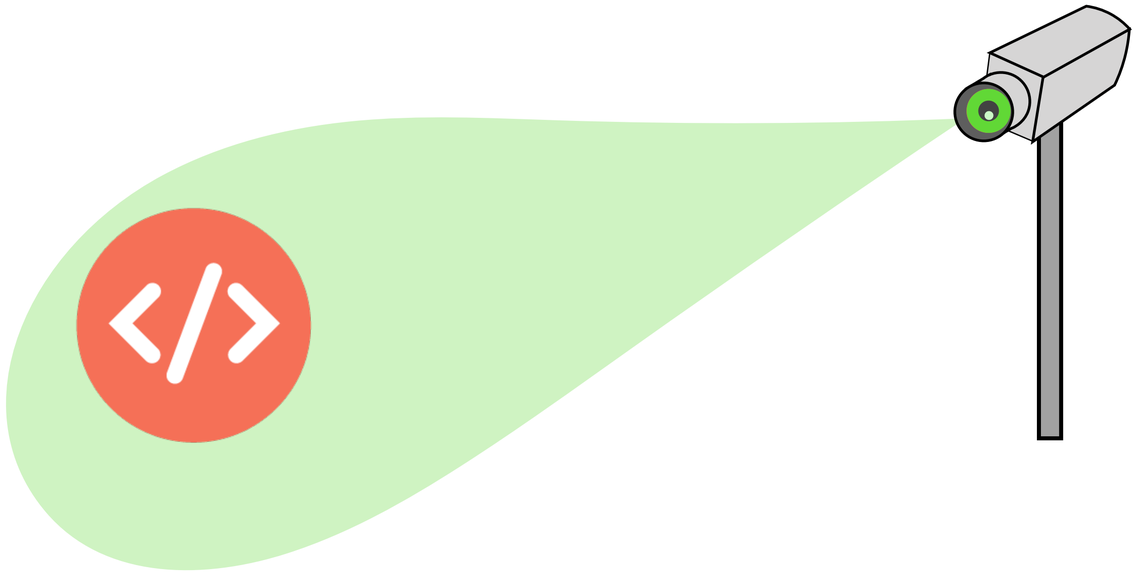


[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

- **Robustness to Preprocessing**
- Plausibility
- Preserved Semantics
- Available Transformations

Which preprocessing are you considering?

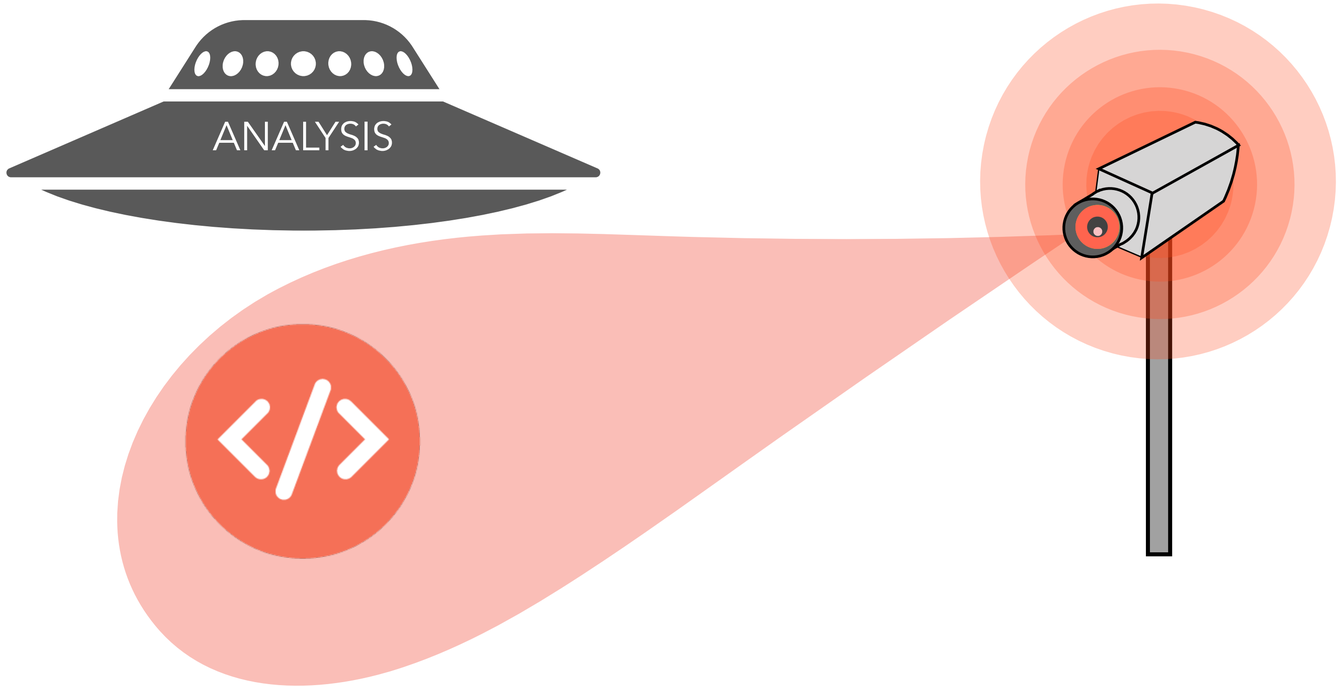


[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem-Space Constraints

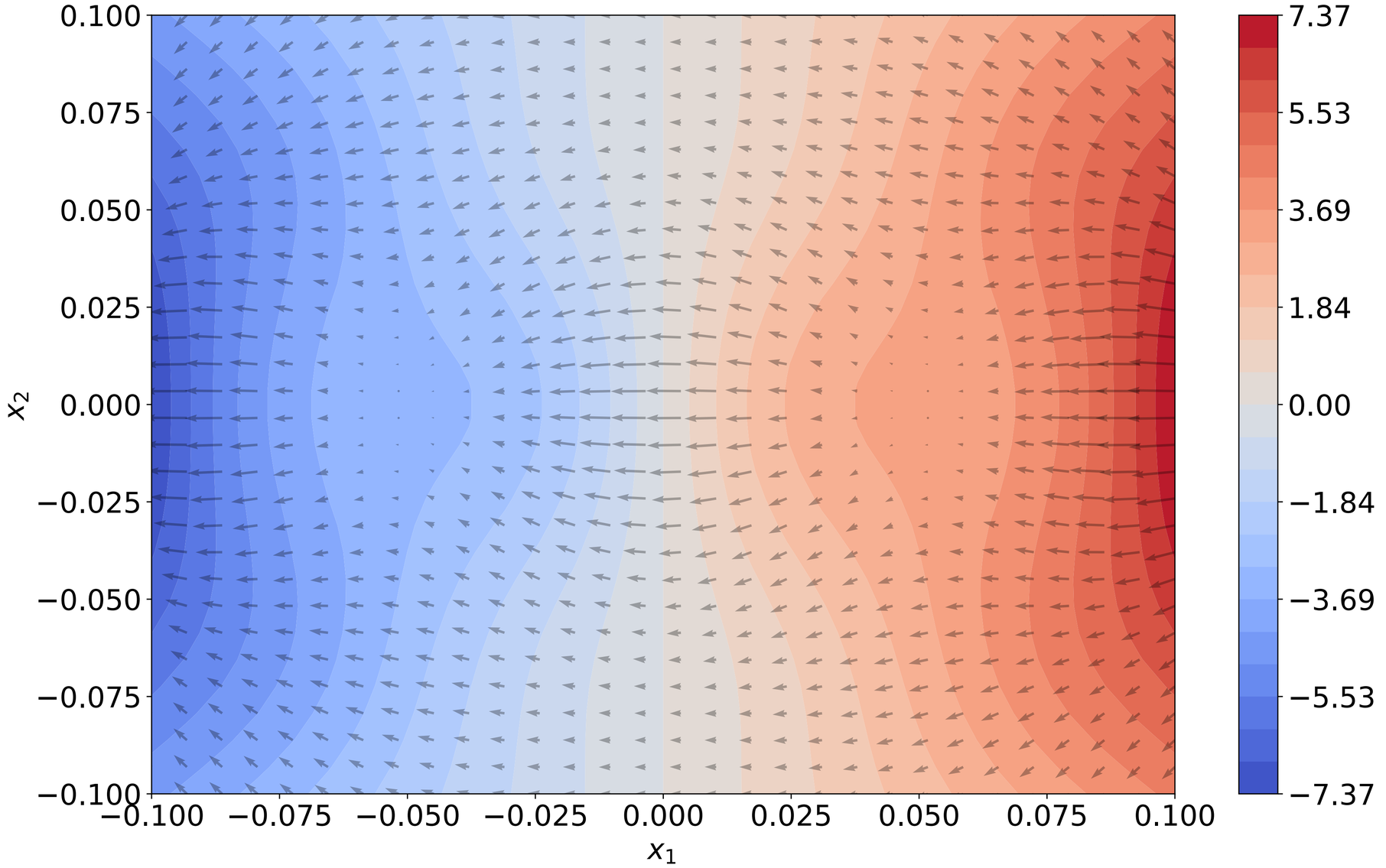
- **Robustness to Preprocessing**
- Plausibility
- Preserved Semantics
- Available Transformations

Which preprocessing are you considering?



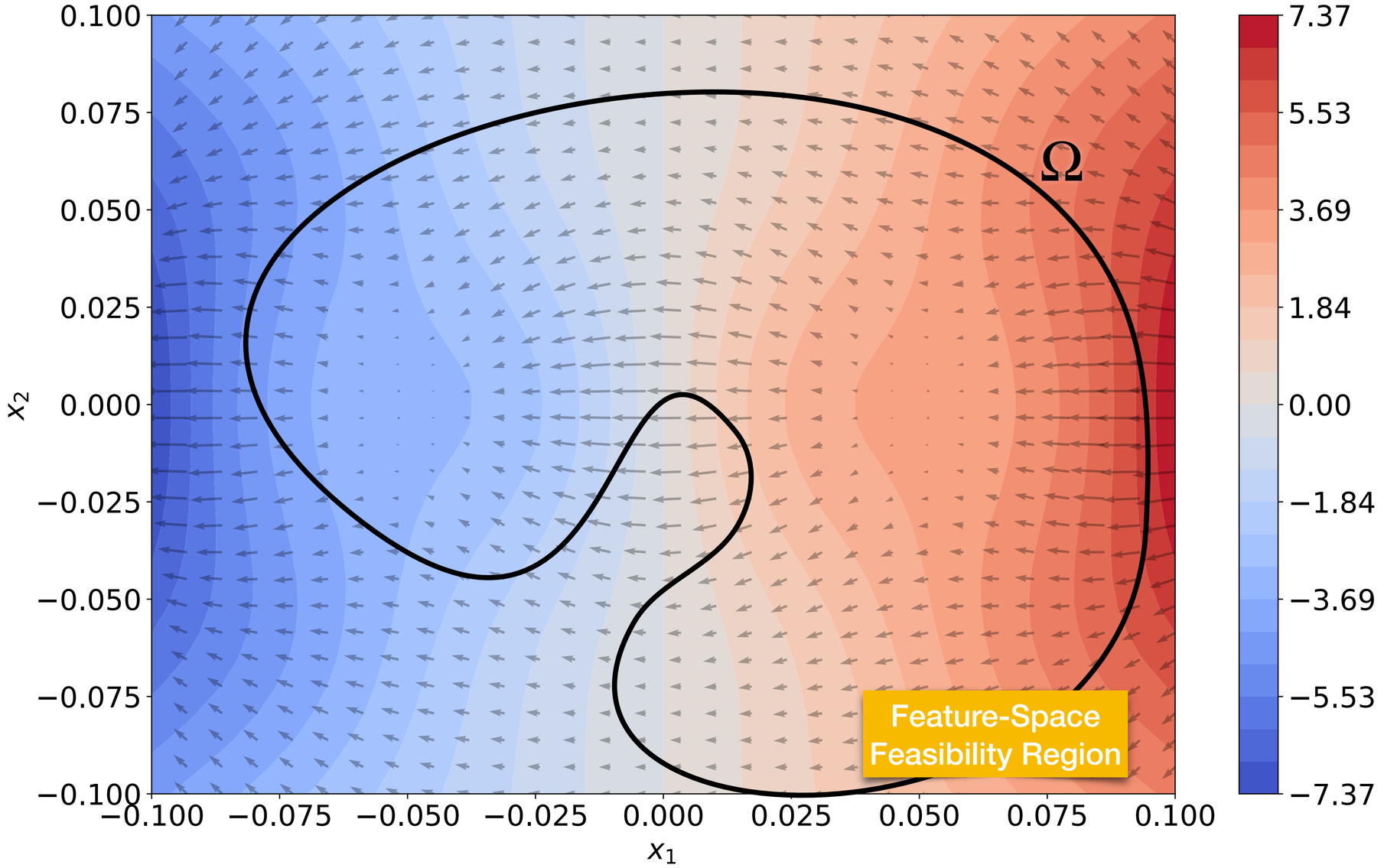
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Side-effect Features: the ONE Slide to Remember



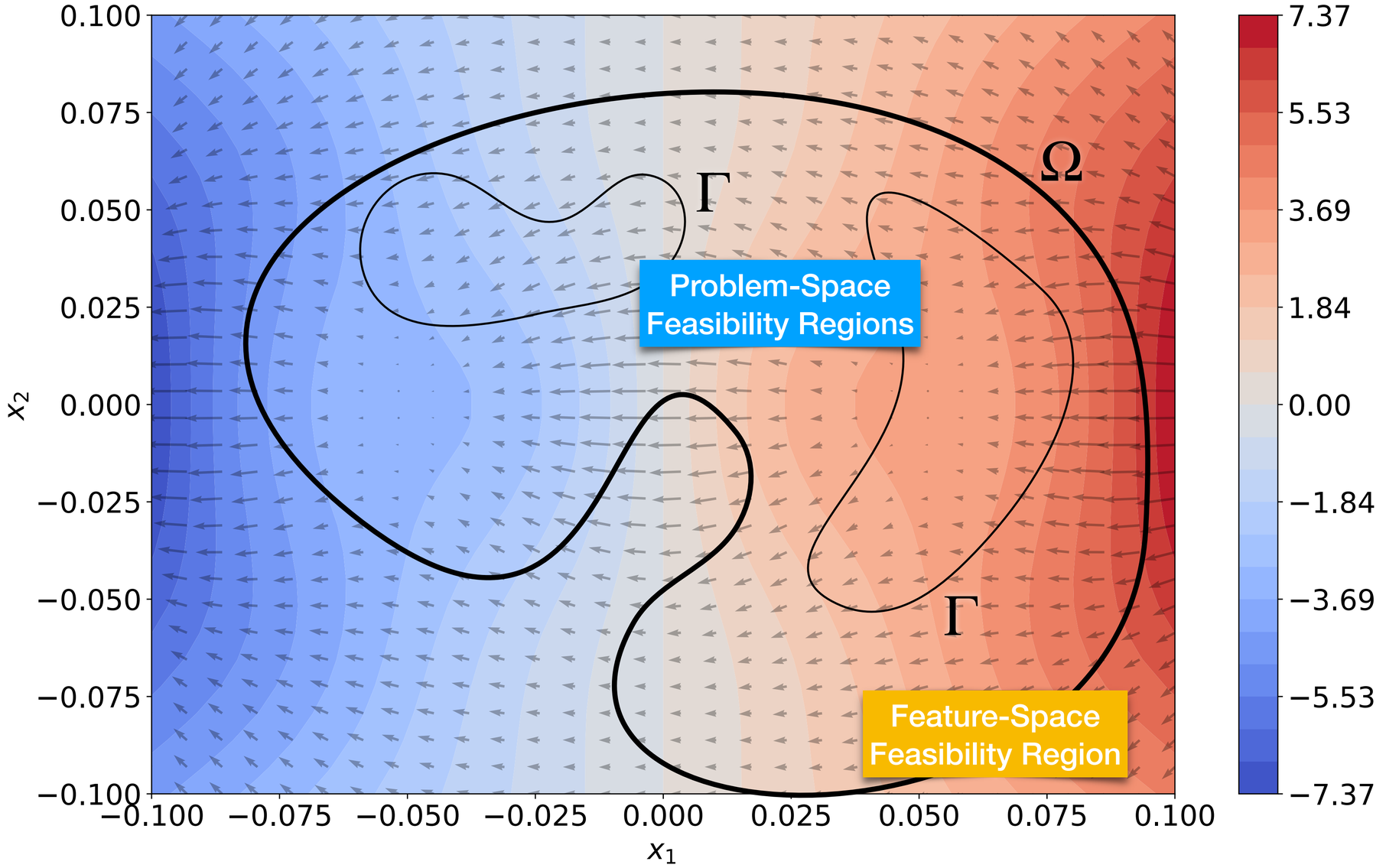
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Side-effect Features: the ONE Slide to Remember



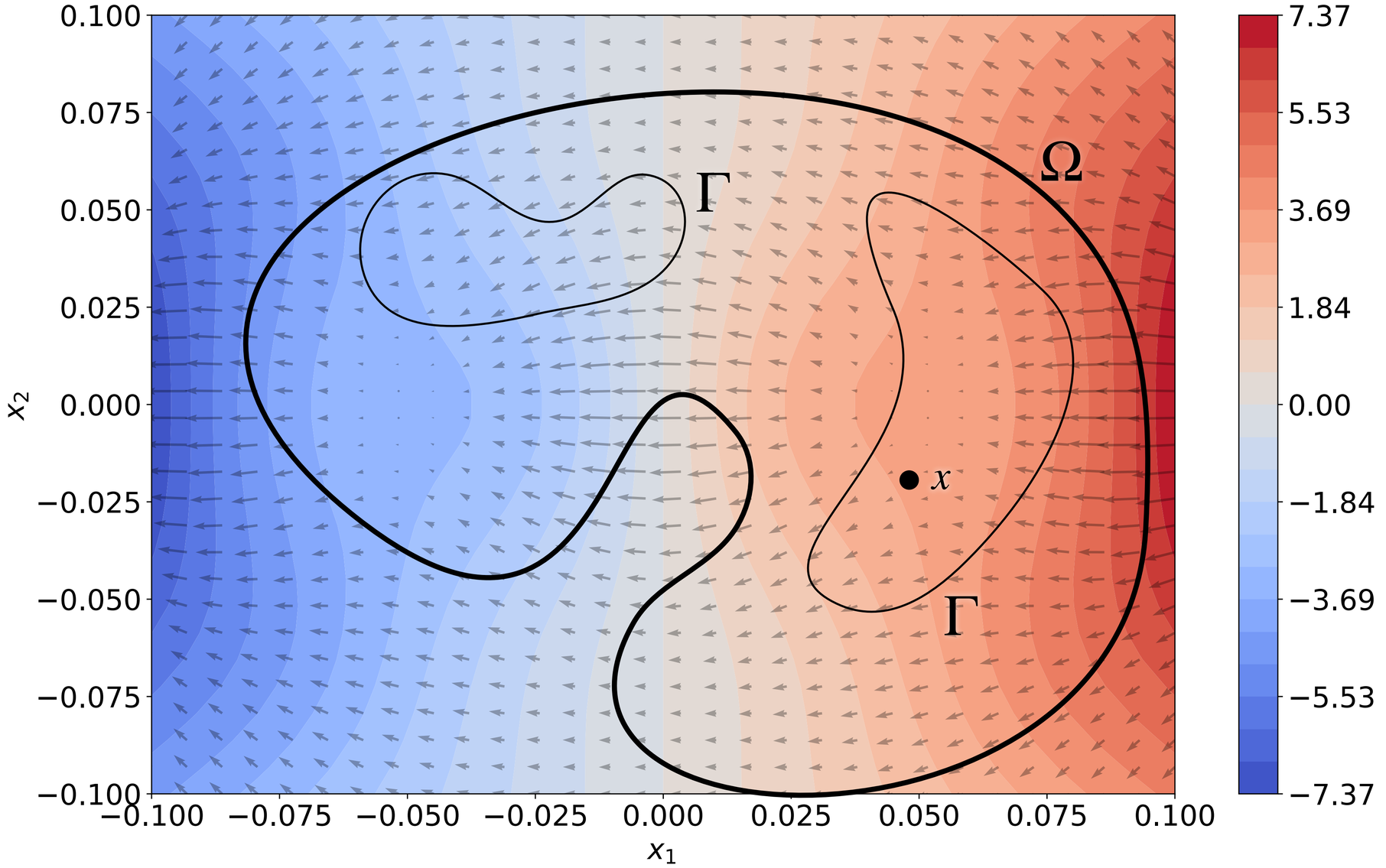
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Side-effect Features: the ONE Slide to Remember



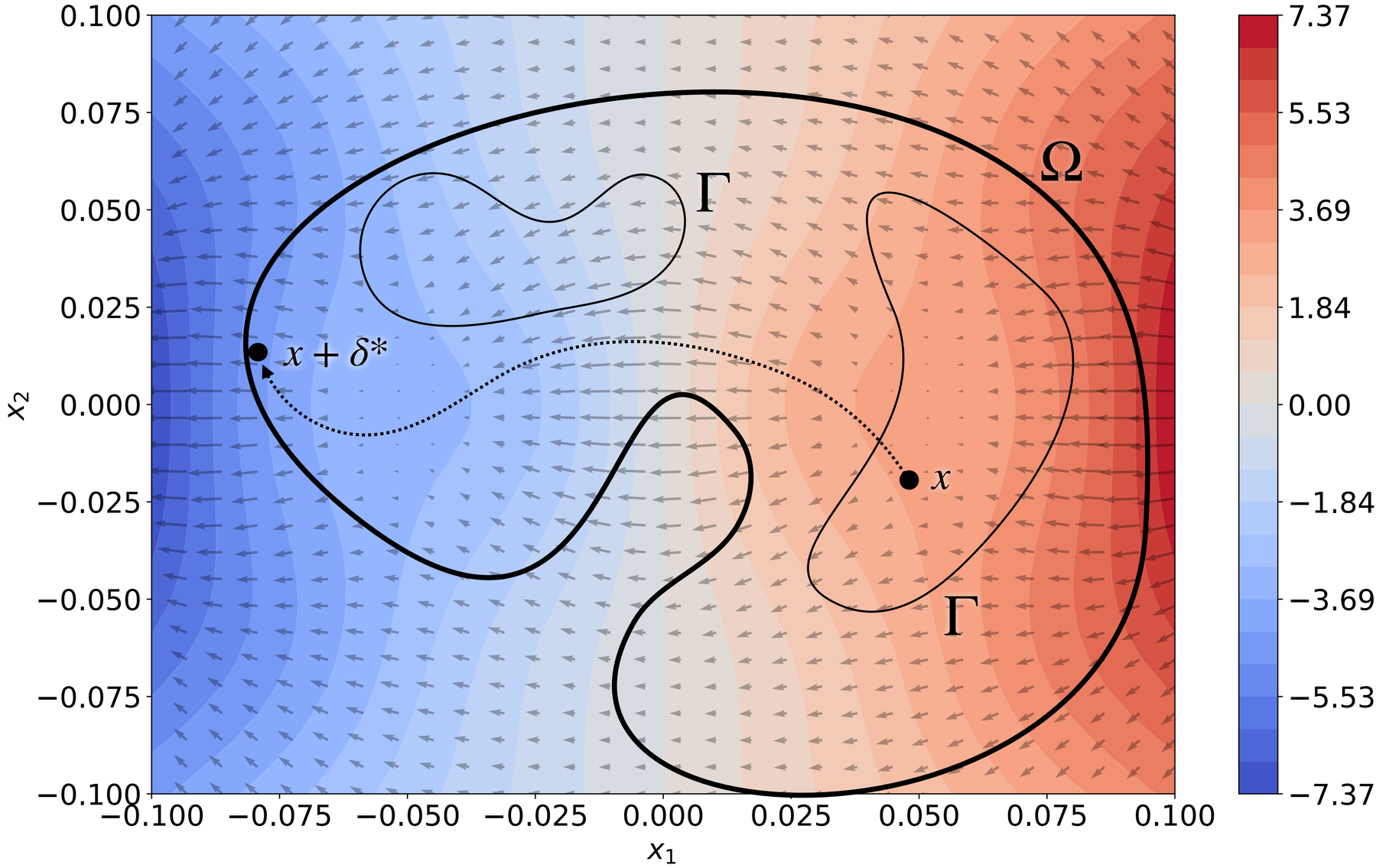
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Side-effect Features: the ONE Slide to Remember



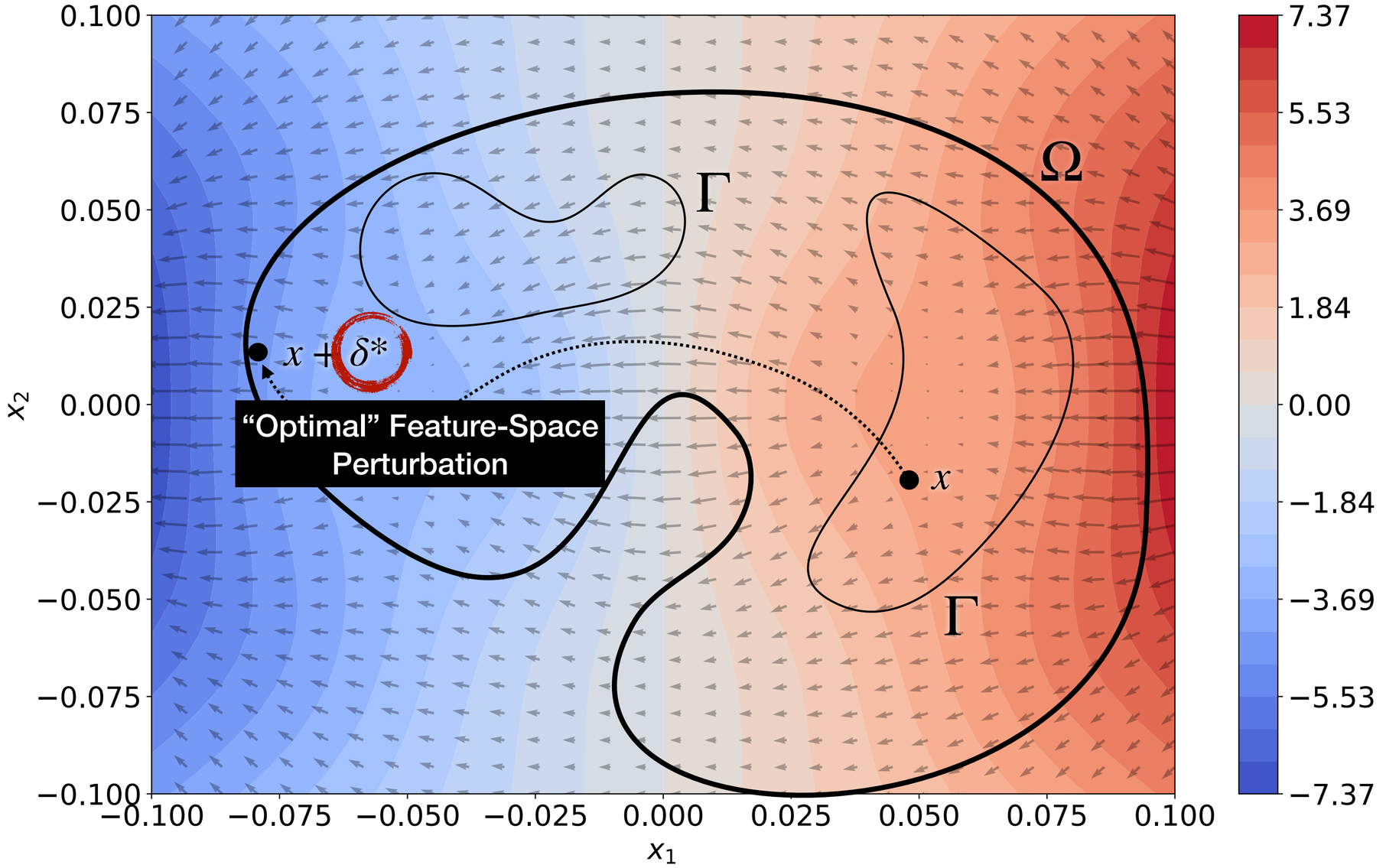
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Side-effect Features: the ONE Slide to Remember



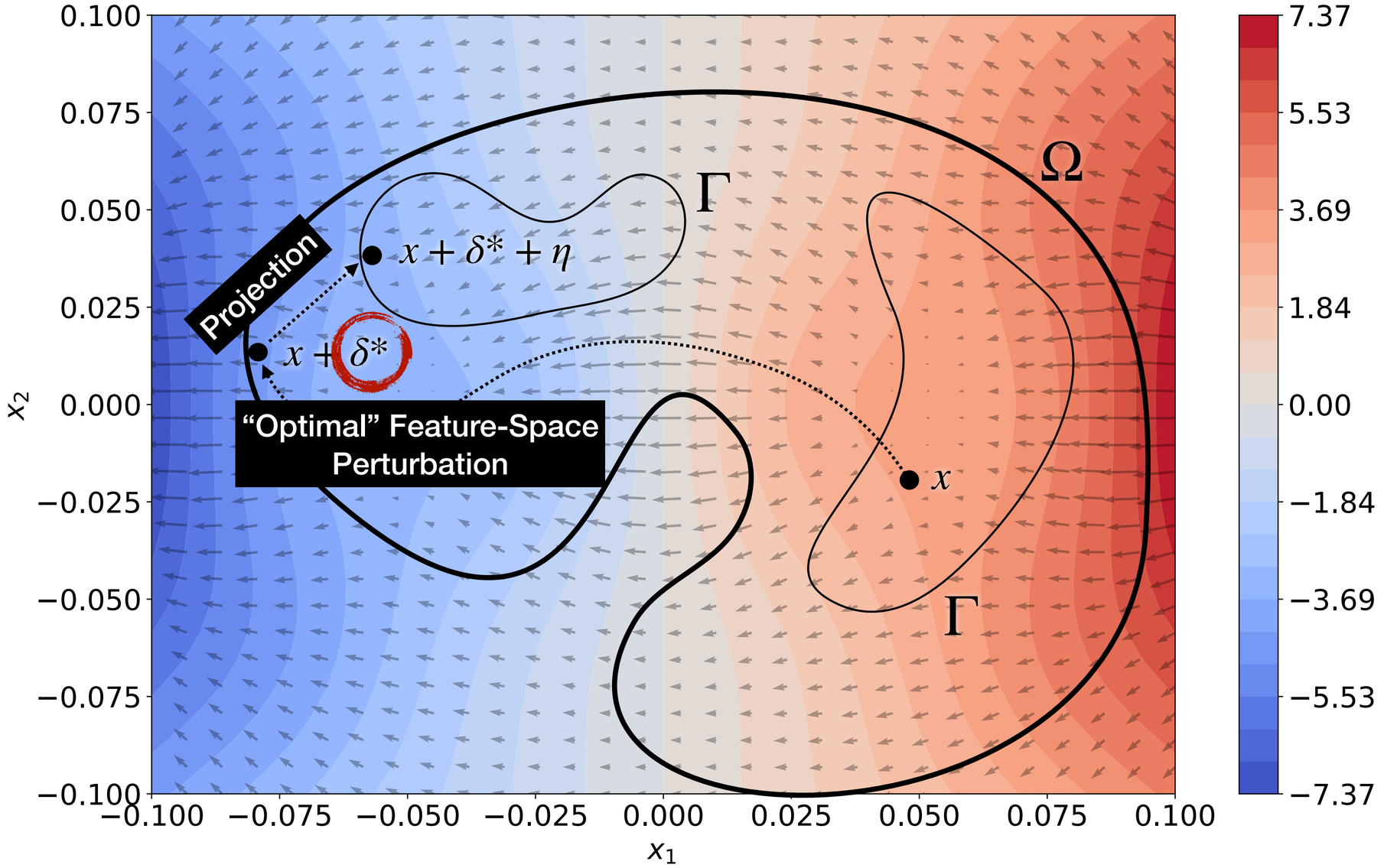
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Side-effect Features: the ONE Slide to Remember



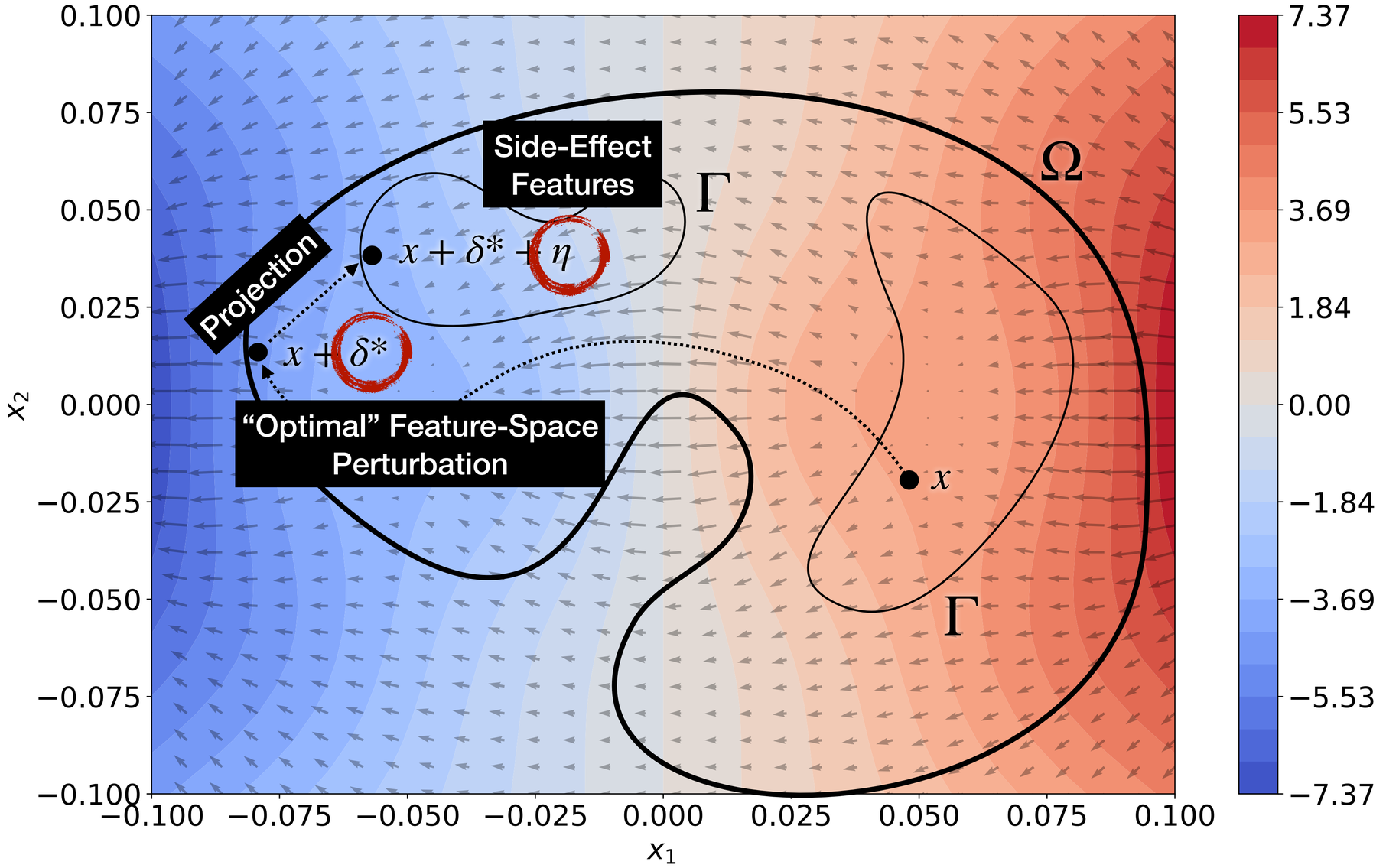
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Side-effect Features: the ONE Slide to Remember



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Side-effect Features: the ONE Slide to Remember



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Feature Space vs. Problem Space: the OTHER Slide to Remember

$$\boldsymbol{\delta}^* = \arg\min_{\boldsymbol{\delta} \in \mathbb{R}^n} \quad f_t(\boldsymbol{x} + \boldsymbol{\delta}) \qquad \text{subject to:} \quad \boldsymbol{\delta} \models \Omega\,.$$

$$\begin{aligned} \operatorname{argmin}_{\mathbf{T} \in \mathcal{T}} \quad & f_t(\varphi(\mathbf{T}(z))) = f_t(\boldsymbol{x} + \boldsymbol{\delta}^* + \boldsymbol{\eta}) \\ \text{subject to:} \quad & [\![z]\!]^\tau = [\![\mathbf{T}(z)]\!]^\tau, \quad \forall \tau \in \Upsilon \\ & \pi(\mathbf{T}(z)) = 1, \quad \forall \pi \in \Pi \\ & \mathbf{A}(\mathbf{T}(z)) = \mathbf{T}(z), \quad \forall \mathbf{A} \in \Lambda \end{aligned}$$

Feature-Space Constraints

- Lp perturbations
- Domain constraints for vectors

Search Strategy

- Gradient-driven

Problem-Space Constraints

- Available Transformations
- Preserved Semantics
- Plausibility
- Robustness to Preprocessing

Search Strategy

- Gradient-driven
- Problem-driven
- Hybrid

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Feature Space vs. Problem Space: the OTHER Slide to Remember

$$\boldsymbol{\delta}^* = \arg\min_{\boldsymbol{\delta}\in\mathbb{R}^n} \quad f_t(\boldsymbol{x}+\boldsymbol{\delta}) \qquad \text{subject to:} \quad \boldsymbol{\delta} \models \Omega\,.$$

$$\operatorname{argmin}_{\mathbf{T}\in\mathcal{T}} \quad f_t(\varphi(\mathbf{T}(z))) = f_t(\boldsymbol{x}+\boldsymbol{\delta}^*+\boldsymbol{\eta})$$
$$\text{subject to:} \quad [\![z]\!]^\tau = [\![\mathbf{T}(z)]\!]^\tau, \quad \forall \tau \in \Upsilon$$
$$\pi(\mathbf{T}(z)) = 1, \quad \forall \pi \in \Pi$$
$$\mathbf{A}(\mathbf{T}(z)) = \mathbf{T}(z), \quad \forall \mathbf{A} \in \Lambda$$

Feature-Space Constraints

- Lp perturbations
- Domain constraints for vectors

Search Strategy

- Gradient-driven

Problem-Space Constraints

- Available Transformations
- Preserved Semantics
- Plausibility
- Robustness to Preprocessing

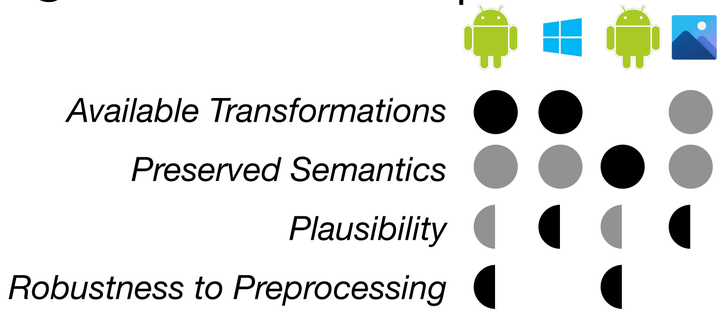
Search Strategy

- Gradient-driven
- Problem-driven
- Hybrid

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Works for Multiple Domains

Compare existing methods & improve SoA

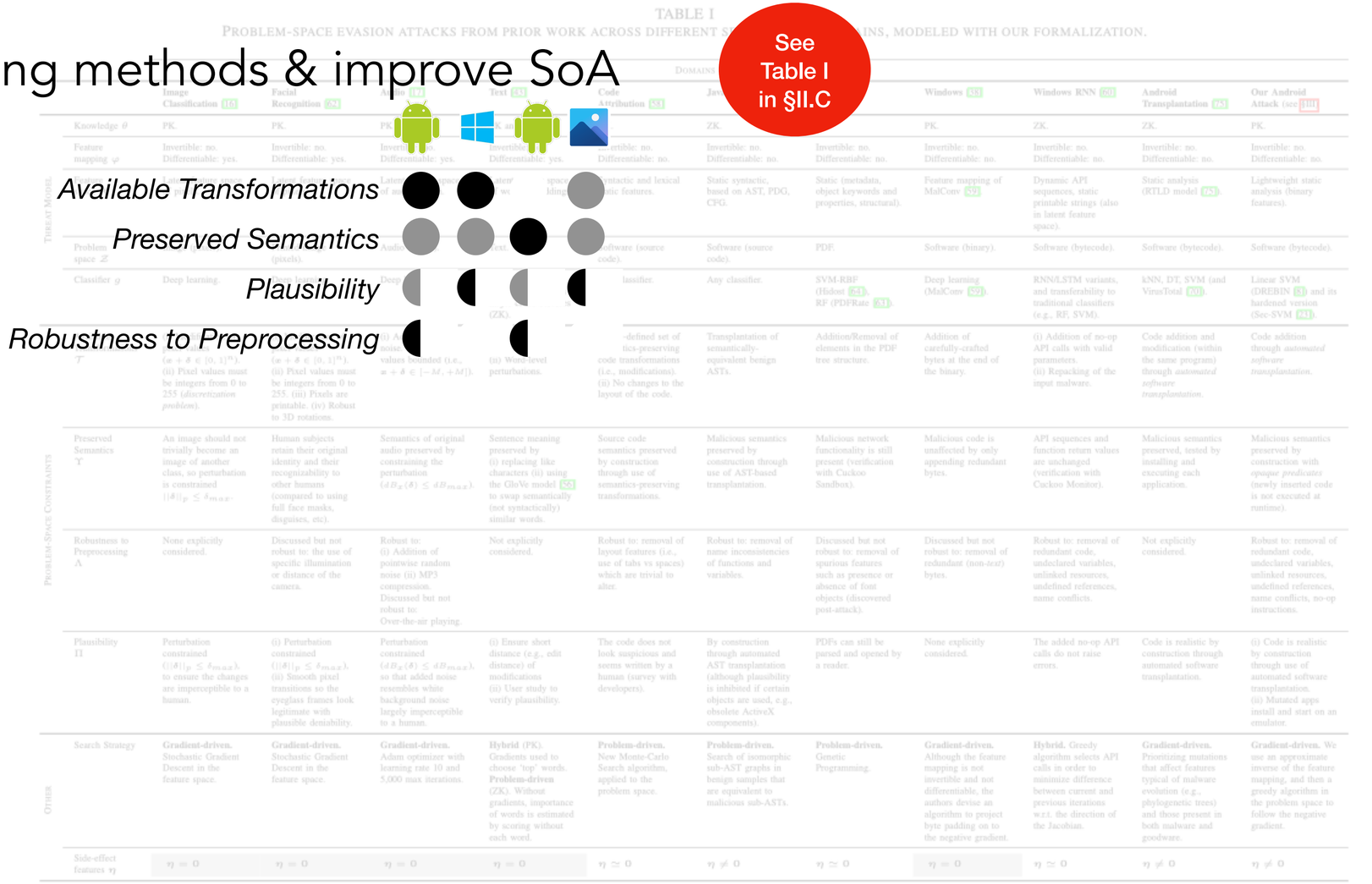


See Table I in §II.C

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Works for Multiple Domains

Compare existing methods & improve SoA



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Works for Multiple Domains

Compare existing methods & improve SoA


Facial Recognition


Speech Recognition


Android Malware

Malicious Javascript


Image Classification


Text Classification

Windows Malware


Code Attribution

PDF Malware

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

How hard could it be?

- **Despite hype on adversarial learning: No suitable work for us** 😢
- **Two tricky challenges**
  - No inverse map from topic space back to problem space
  - Unobtrusive changes lead to side effects in the feature space

Slide 13

Machine Learning and Security

19:31 / 51:04

Konrad Rieck: When Papers Choose their Reviewers: Adversarial ML in Conference Management Systems.

ViSP
34 subscribers
Subscribe
Like
Share
Download
Clip

[USENIX Sec 2023] Eisenhofer et al. No more Reviewer #2: Subverting Automatic Paper-Reviewer Assignment using Adversarial Learning

How hard could it be?

- **Despite hype on adversarial learning: No suitable work for us** 😢
- **Two tricky challenges**
  - No inverse map from topic space back to problem space
  - Unobtrusive changes lead to side effects in the feature space

Slide 13

Machine Learning and Security

19:31 / 51:04

Konrad Rieck: When Papers Choose their Reviewers: Adversarial ML in Conference Management Systems.

ViSP
34 subscribers
Subscribe
Like
Share
Download
Clip

[USENIX Sec 2023] Eisenhofer et al. No more Reviewer #2: Subverting Automatic Paper-Reviewer Assignment using Adversarial Learning

How hard could it be?

- **Despite hype on adversarial learning: No suitable work for us** 😢
- **Two tricky challenges**
  - No inverse map from topic space back to problem space
  - Unobtrusive changes lead to side effects in the feature space

Slide 13

Machine Learning and Security

Problem space reasoning applies and affects (backdoor) poisoning attacks too!

[IEEE S&P 2023] Yang et al. Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers

Android Attack: Experiments

Results: What is the impact on the Feature Space?

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Results: What is the impact on the Feature Space?

- **Dataset**: ~170K Android apps (10% malware) from Jan 2017 to Dec 2018

Results: What is the impact on the Feature Space?

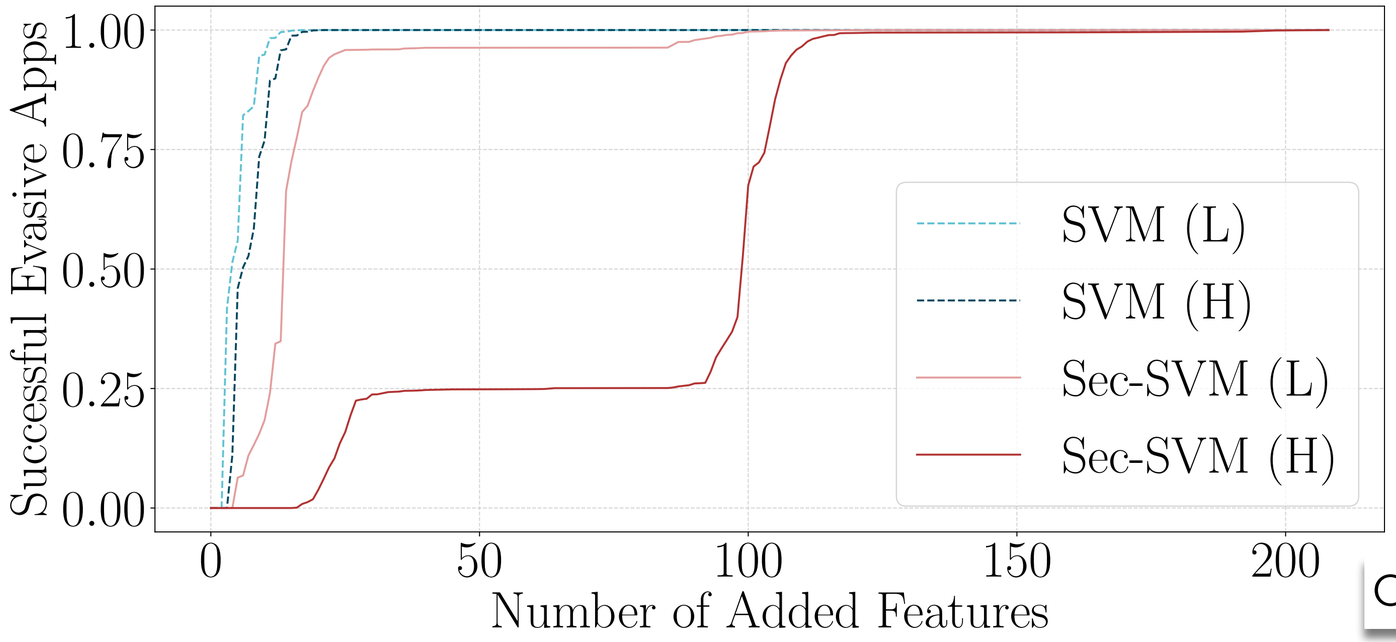
- **Dataset**: ~170K Android apps (10% malware) from Jan 2017 to Dec 2018
- **DREBIN** [NDSS'14]: Linear SVM, binary feature space
- **Sec-SVM** [TDSC'17]: Feature-space defense for DREBIN (evenly distributes weights)

Results: What is the impact on the Feature Space?

- **Dataset**: ~170K Android apps (10% malware) from Jan 2017 to Dec 2018
- **DREBIN** [NDSS'14]: Linear SVM, binary feature space
- **Sec-SVM** [TDSC'17]: Feature-space defense for DREBIN (evenly distributes weights)
- **Low** (L) vs **High** (H) **confidence**: cross decision boundary or cross into Q1 of benign

Results: What is the impact on the Feature Space?

- **Dataset**: ~170K Android apps (10% malware) from Jan 2017 to Dec 2018
- **DREBIN** [NDSS'14]: Linear SVM, binary feature space
- **Sec-SVM** [TDSC'17]: Feature-space defense for DREBIN (evenly distributes weights)
- **Low** (L) vs **High** (H) **confidence**: cross decision boundary or cross into Q1 of benign



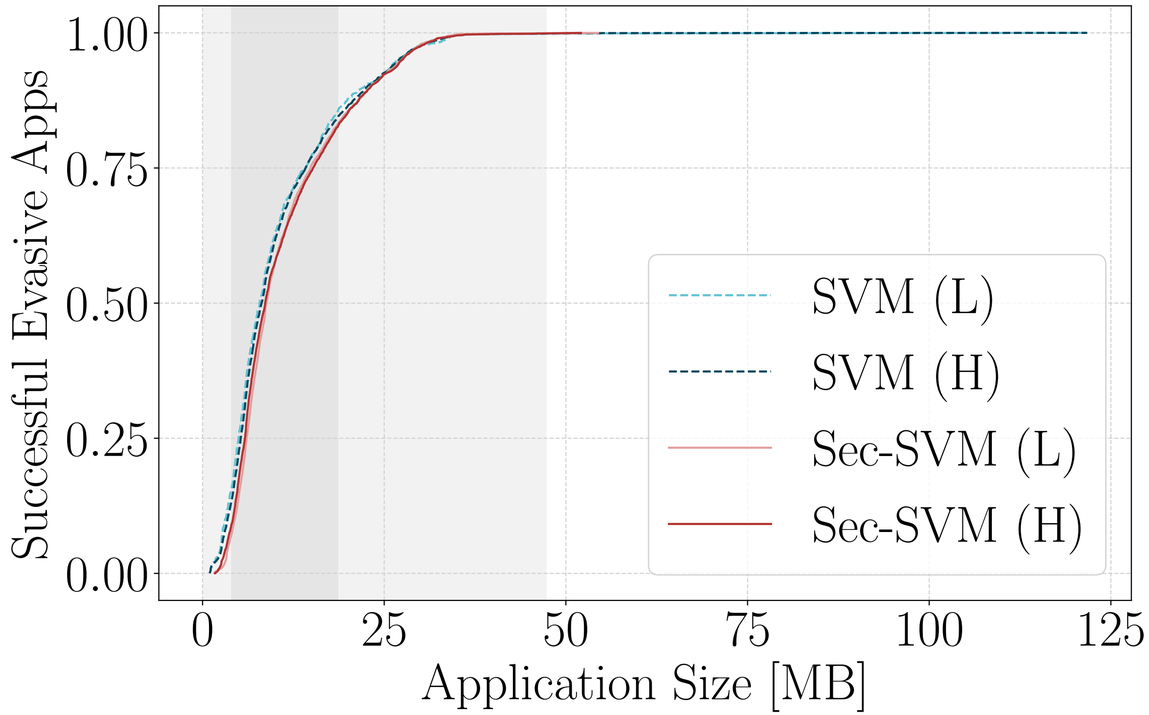
Out of 10K features

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Results: What is the impact on the Problem Space?

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

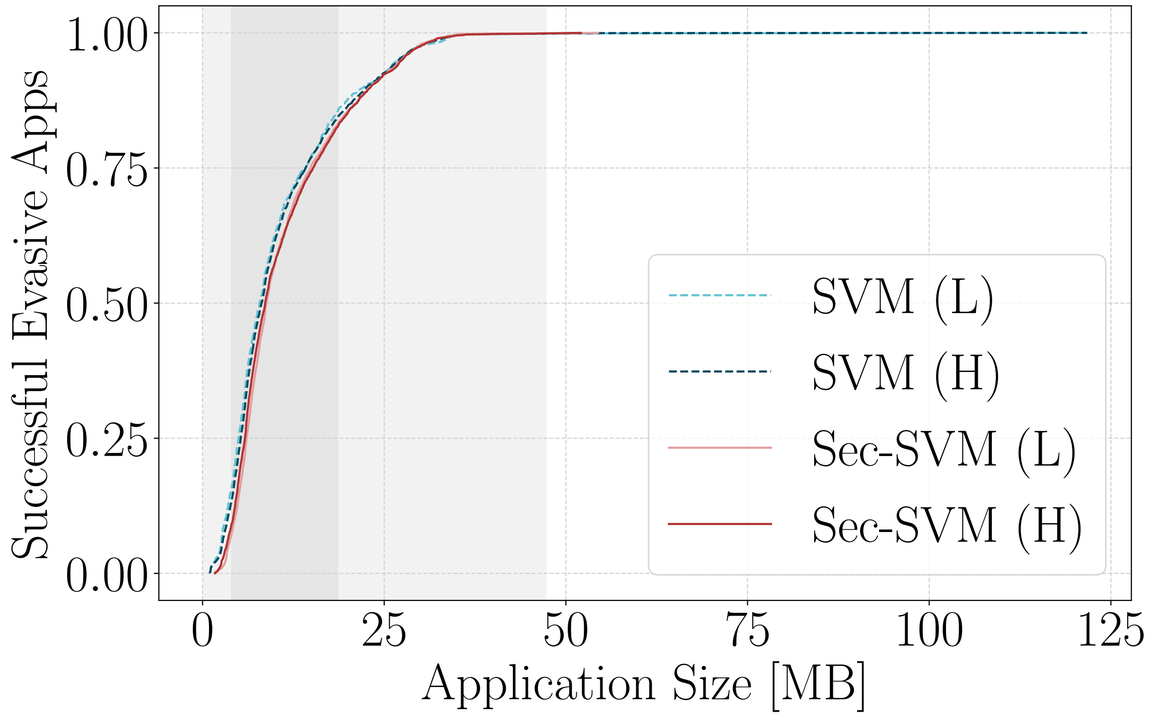
Results: What is the impact on the Problem Space?



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Results: What is the impact on the Problem Space?

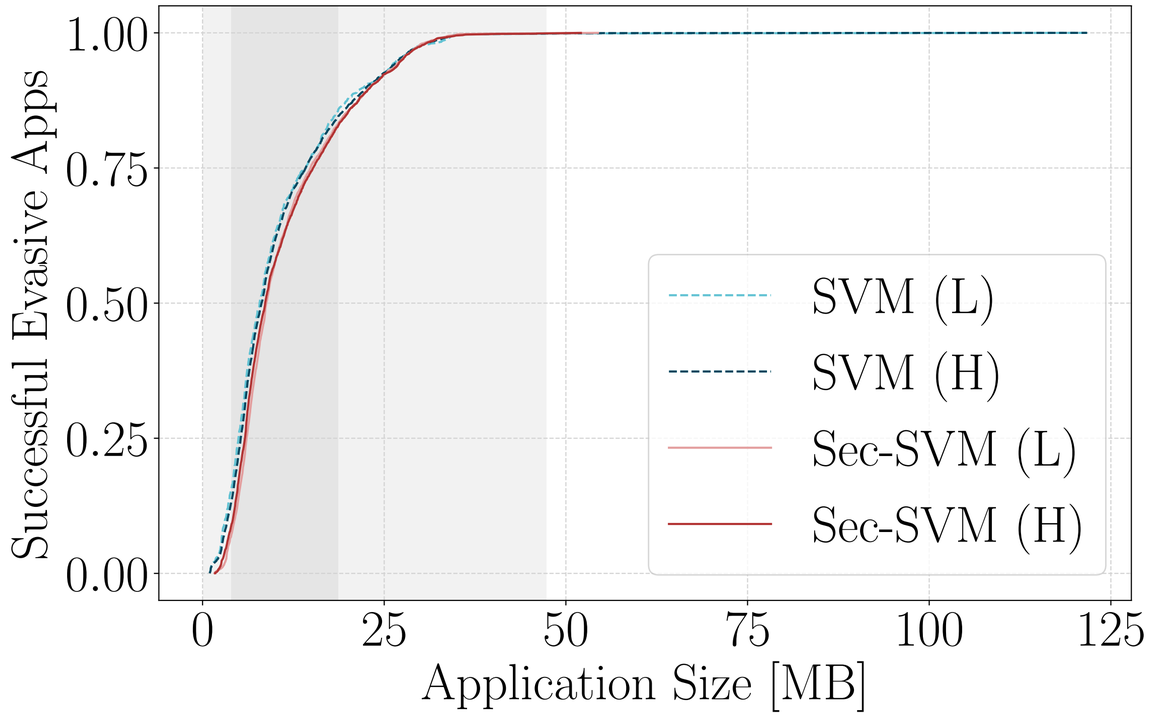
- Adversarial generation < 2 minutes per app



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Results: What is the impact on the Problem Space?

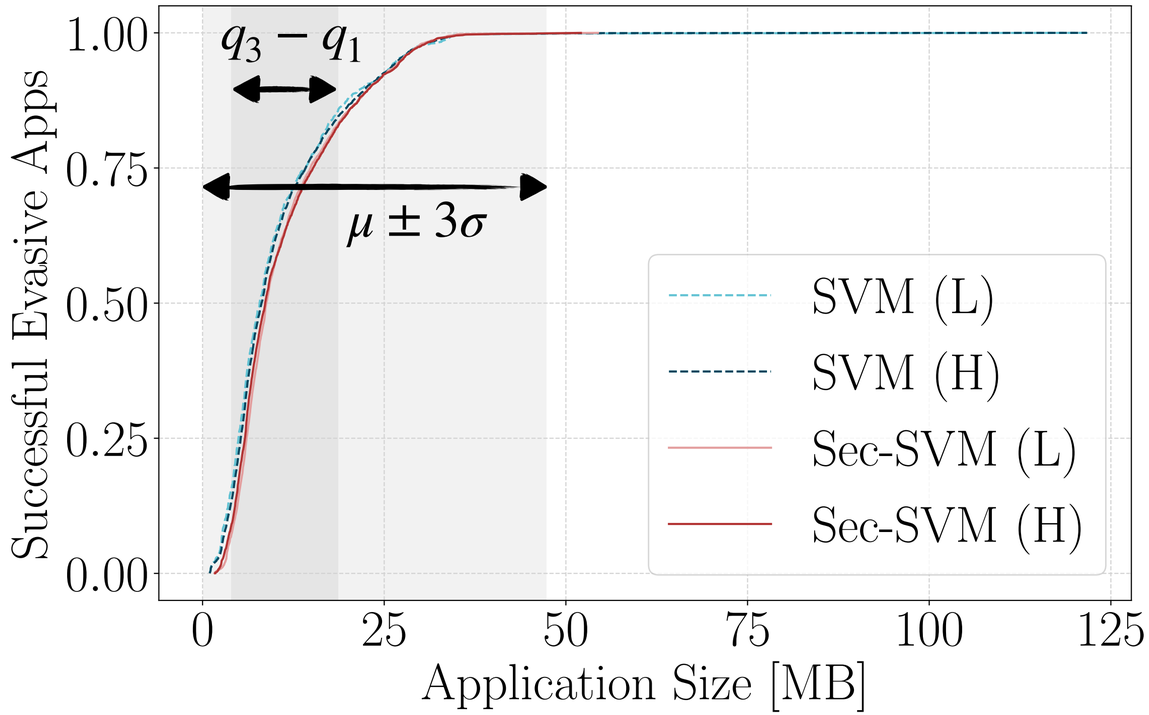
- Adversarial generation < 2 minutes per app
- Restricting feature-space perturbations δ does not hinder problem-space attack



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Results: What is the impact on the Problem Space?

- Adversarial generation < 2 minutes per app
- Restricting feature-space perturbations δ does not hinder problem-space attack
- App statistics (e.g., size) do not become anomalous after injection (other stats in the paper)



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem Space Adversarial Training

Problem vs Feature Space Adversarial Training

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

[IEEE S&P 2023] Dyrmishi et al. On The Empirical Effectiveness of Unrealistic Adversarial Hardening Against Realistic Adversarial Attacks

Problem vs Feature Space Adversarial Training

- Exciting work [IEEE S&P 2023] on Text, Botnet Traffic, Windows Malware Classification Tasks
  - Text: Problem Space AT 16.94% more effective than Feature Space AT
  - Botnet Traffic: Problem Space AT robustness ~= Feature Space AT
  - Windows Malware: Problems Space AT outperforms Feature Space AT robustness

[IEEE S&P 2020] **Intriguing Properties of Adversarial ML Attacks in the Problem Space**
https://s2lab.cs.ucl.ac.uk/projects/intriguing

[IEEE S&P 2023] Dyrmishi et al. **On The Empirical Effectiveness of Unrealistic Adversarial Hardening Against Realistic Adversarial Attacks**

Problem vs Feature Space Adversarial Training

- Exciting work [IEEE S&P 2023] on Text, Botnet Traffic, Windows Malware Classification Tasks
  - **(Marginal)** Text: Problem Space AT 16.94% more effective than Feature Space AT
  - **(Not Required)** Botnet Traffic: Problem Space AT robustness ~= Feature Space AT
  - **(Required)** Windows Malware: Problems Space AT > Feature Space AT robustness
- It may seem a task-dependent result… let's fix some variables

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

[IEEE S&P 2023] Dyrmishi et al. On The Empirical Effectiveness of Unrealistic Adversarial Hardening Against Realistic Adversarial Attacks

Problem vs Feature Space Adversarial Training

- Exciting work [IEEE S&P 2023] on Text, Botnet Traffic, Windows Malware Classification Tasks
  - **(Marginal)** Text: Problem Space AT 16.94% more effective than Feature Space AT
  - **(Not Required)** Botnet Traffic: Problem Space AT robustness ~= Feature Space AT
  - **(Required)** Windows Malware: Problems Space AT > Feature Space AT robustness
- It may seem a task-dependent result… let's fix some variables

- **Dataset**: ~170K Android apps (10% malware) from Jan 2017 to Dec 2018
- **DREBIN** [NDSS'14]: Linear SVM, binary feature space
- **Sec-SVM** [TDSC'17]: Feature-space defense for DREBIN (evenly distributes weights)
- Increasing level of perturbation budget and [IEEE S&P 2020] Android attack

[IEEE S&P 2020] **Intriguing Properties of Adversarial ML Attacks in the Problem Space**
https://s2lab.cs.ucl.ac.uk/projects/intriguing

[IEEE S&P 2023] Dyrmishi et al. **On The Empirical Effectiveness of Unrealistic Adversarial Hardening Against Realistic Adversarial Attacks**

Problem vs Feature Space Adversarial Training (SVM)

- **Dataset**: ~170K Android apps (10% malware) from Jan 2017 to Dec 2018
- **DREBIN** [NDSS'14]: Linear SVM, binary feature space
- **Sec-SVM** [TDSC'17]: Feature-space defense for DREBIN (evenly distributes weights)
- Increasing level of perturbation budget and [IEEE S&P 2020] Android attack

SVM Adversarial Training

Budget (%)	FS ASR	PS ASR	FS F1 Clean	PS F1 Clean
10	0.22	0.02	0.9	0.9
20	0.14	0.01	0.9	0.9
30	0.12	0	0.9	0.9
40	0.1	0	0.9	0.9

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem vs Feature Space Adversarial Training (SecSVM)

- **Dataset**: ~170K Android apps (10% malware) from Jan 2017 to Dec 2018
- **DREBIN** [NDSS'14]: Linear SVM, binary feature space
- **Sec-SVM** [TDSC'17]: Feature-space defense for DREBIN (evenly distributes weights)
- Increasing level of perturbation budget and [IEEE S&P 2020] Android attack

SecSVM Adversarial Training

Budget (%)	FS ASR	PS ASR	FS F1 Clean	PS F1 Clean
10	0.95	0.17	0.81	0.81
20	0.91	0.12	0.81	0.81
30	0.92	0.1	0.81	0.81
40	0.89	0.07	0.81	0.81

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem vs Feature Space Adversarial Training (SecSVM)

SecSVM Adversarial Training

Budget (%)	FS ASR	PS ASR	FS F1 Clean	PS F1 Clean
10	0.95	0.17	0.81	0.81
20	0.91	0.12	0.81	0.81
30	0.92	0.1	0.81	0.81
40	0.89	0.07	0.81	0.81

SVM Adversarial Training

Budget (%)	FS ASR	PS ASR	FS F1 Clean	PS F1 Clean
10	0.22	0.02	0.9	0.9
20	0.14	0.01	0.9	0.9
30	0.12	0	0.9	0.9
40	0.1	0	0.9	0.9

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Problem vs Feature Space Adversarial Training

SecSVM Adversarial Training

Budget (%)	FS ASR	PS ASR	FS F1 Clean	PS F1 Clean
10	0.95	0.17	0.81	0.81
20	0.91	0.12	0.81	0.81
[illegible]	[illegible]	[illegible]	[illegible]	0.81
40	0.89	0.07	0.81	0.81

SVM Adversarial Training

Budget (%)	FS ASR	PS ASR	FS F1 Clean	PS F1 Clean
[illegible]	0.22	0.02	0.9	0.9
20	0.14	0.01	0.9	0.9
30	0.12	0	0.9	0.9
40	0.1	0	0.9	0.9

Perhaps not task-dependent but affected by

- Program abstractions
- Feature representations
- ML models

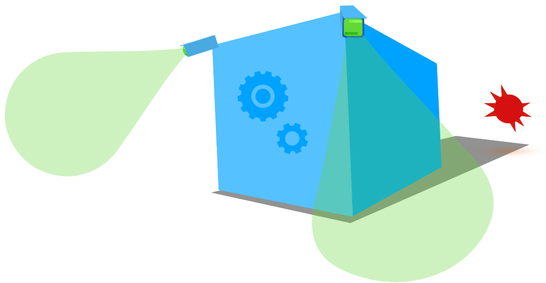
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Outline

Focus

Adversarial ML evasion attacks against malware classifiers

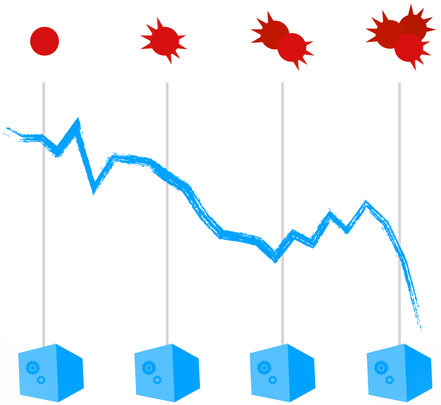
- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives



Bigger Picture

Drifting scenarios caused by threats evolving over time

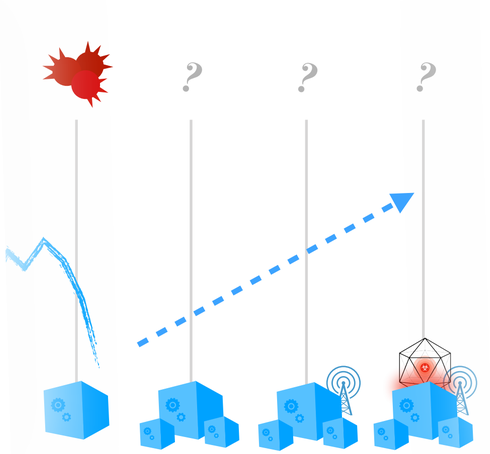
- How concept drift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics



Looking Ahead

Quo vadis?

- Discussion of the future of trustworthy ML for system security
- Robust feature development, universal adversarial perturbations, realizable backdoors, drift forecasting, and the role of abstractions towards the Platonic ideal of interesting behaviors



Outline

Focus

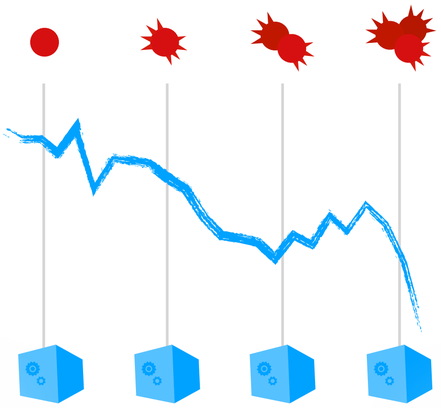
Adversarial ML evasion attacks against malware classifiers

- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives

Bigger Picture

Drifting scenarios caused by threats evolving over time

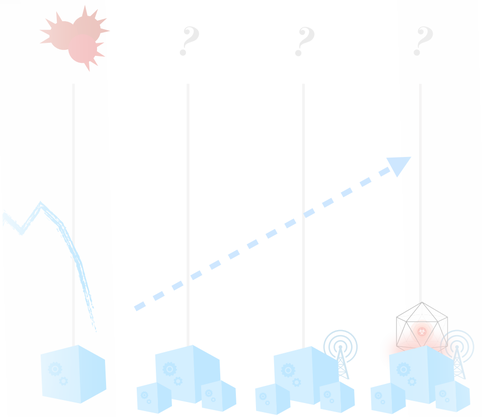
- How concept drift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics



Looking Ahead

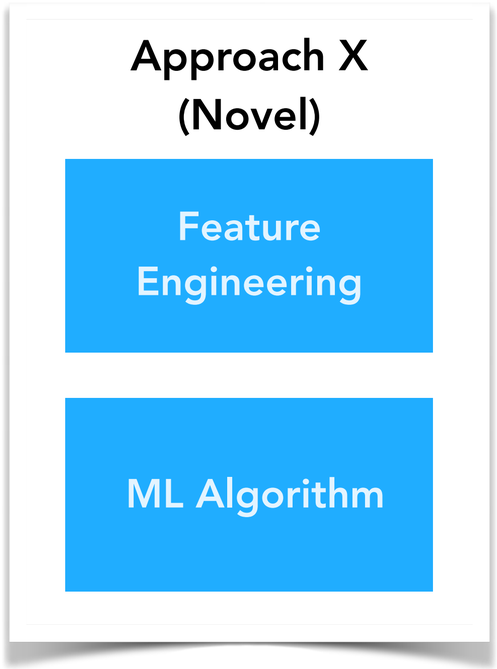
Quo vadis?

- Discussion of the future of trustworthy ML for system security
- Robust feature development, universal adversarial perturbations, realizable backdoors, drift forecasting, and the role of abstractions towards the Platonic ideal of interesting behaviors

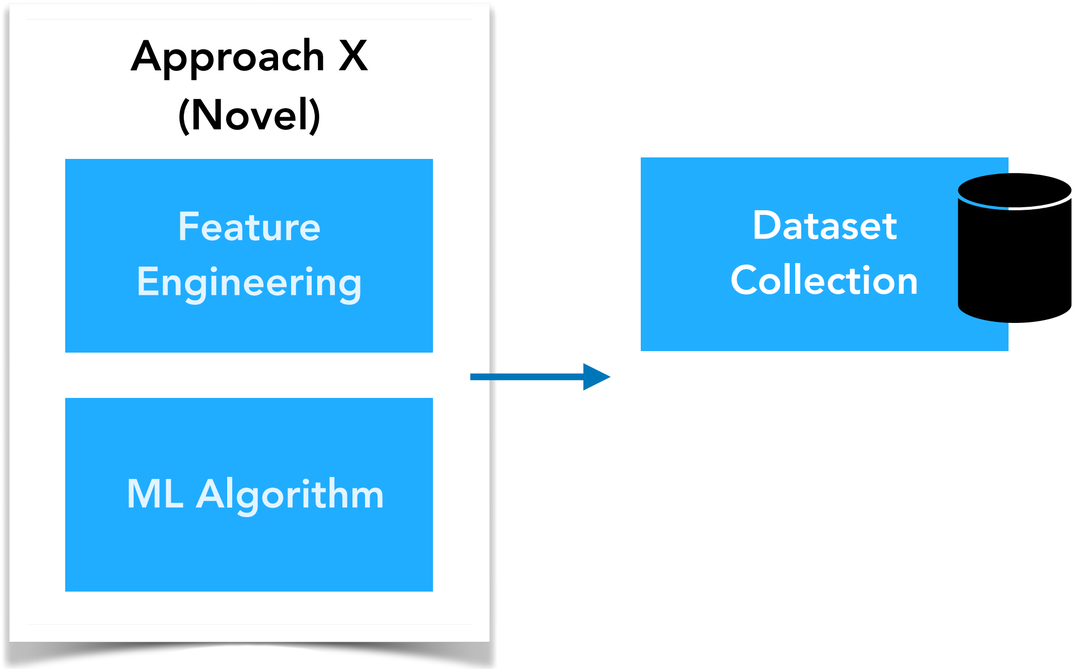


ML for Malware Detection 101

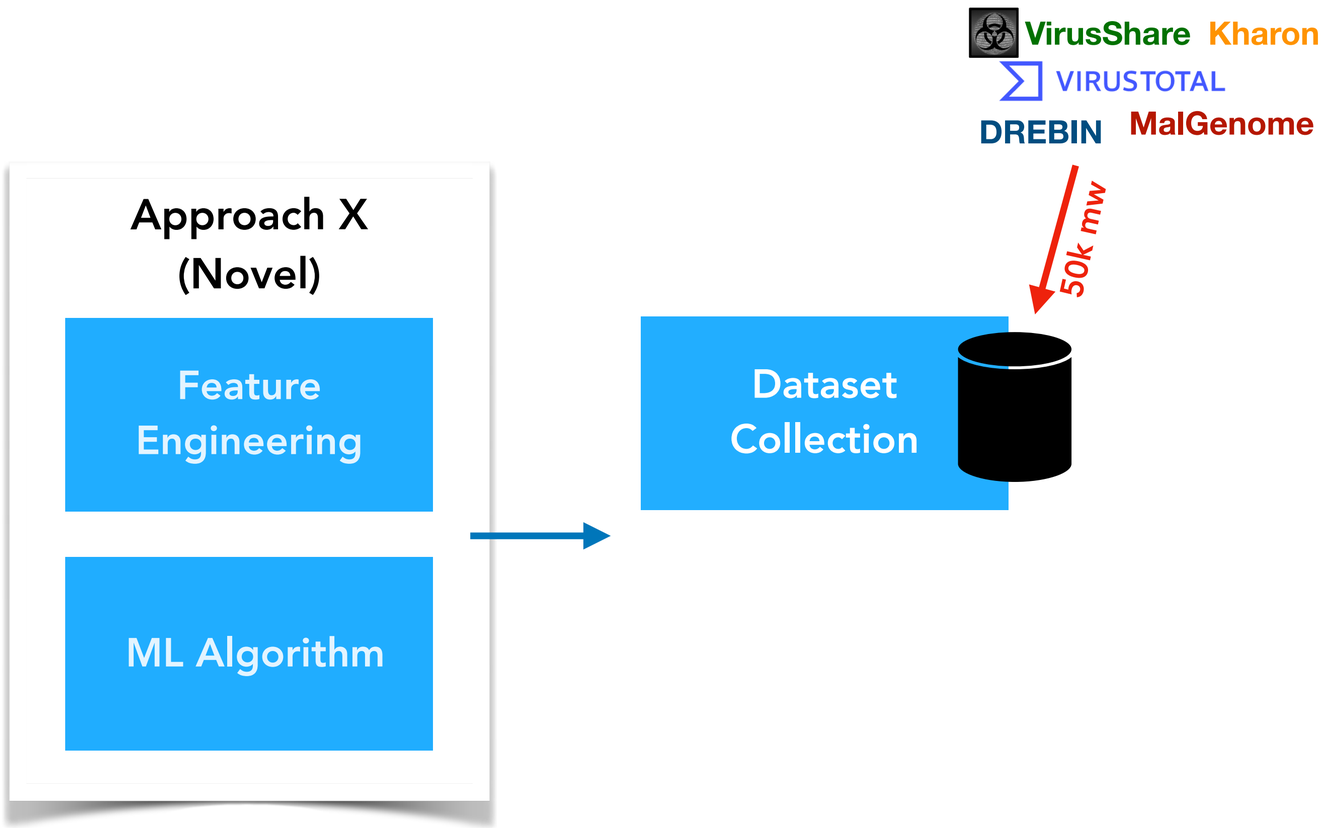
ML for Malware Detection 101



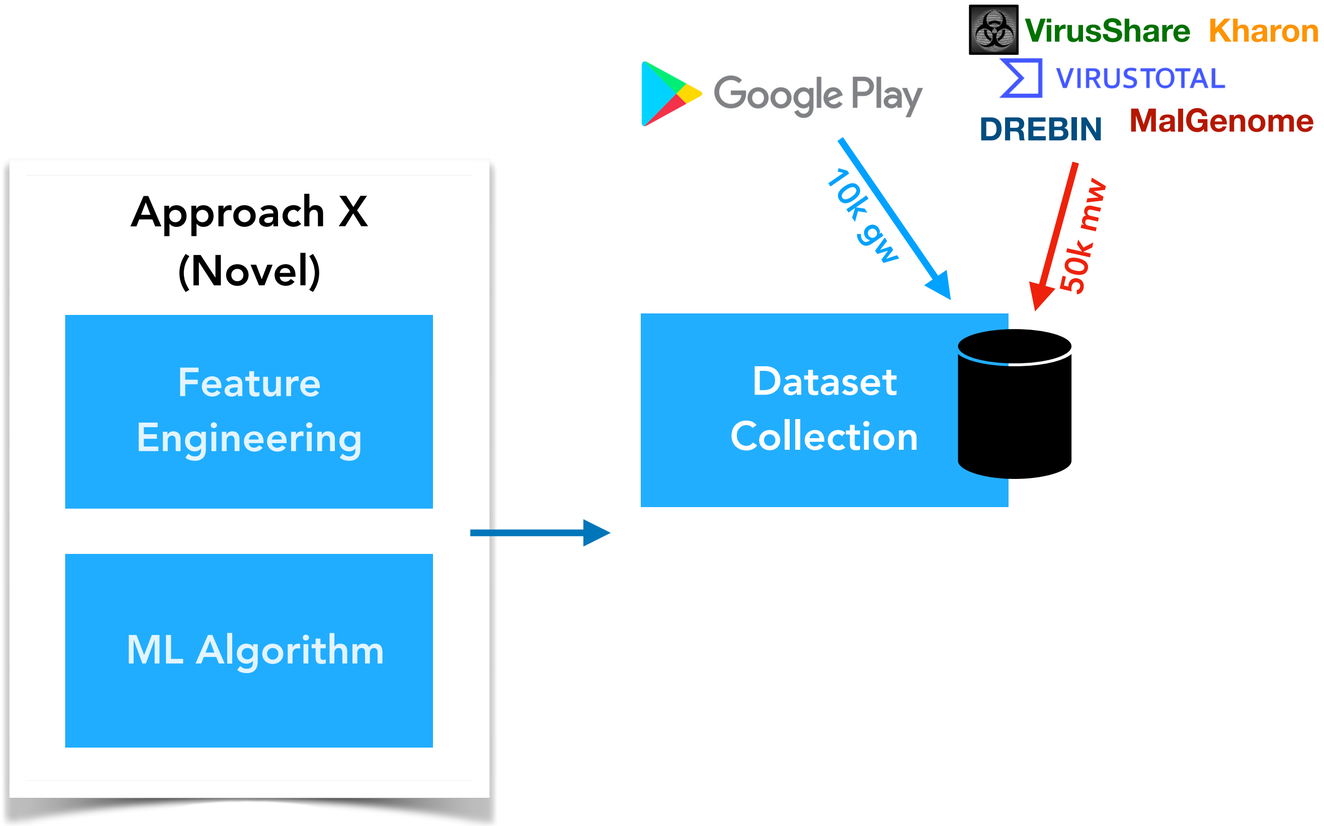
ML for Malware Detection 101



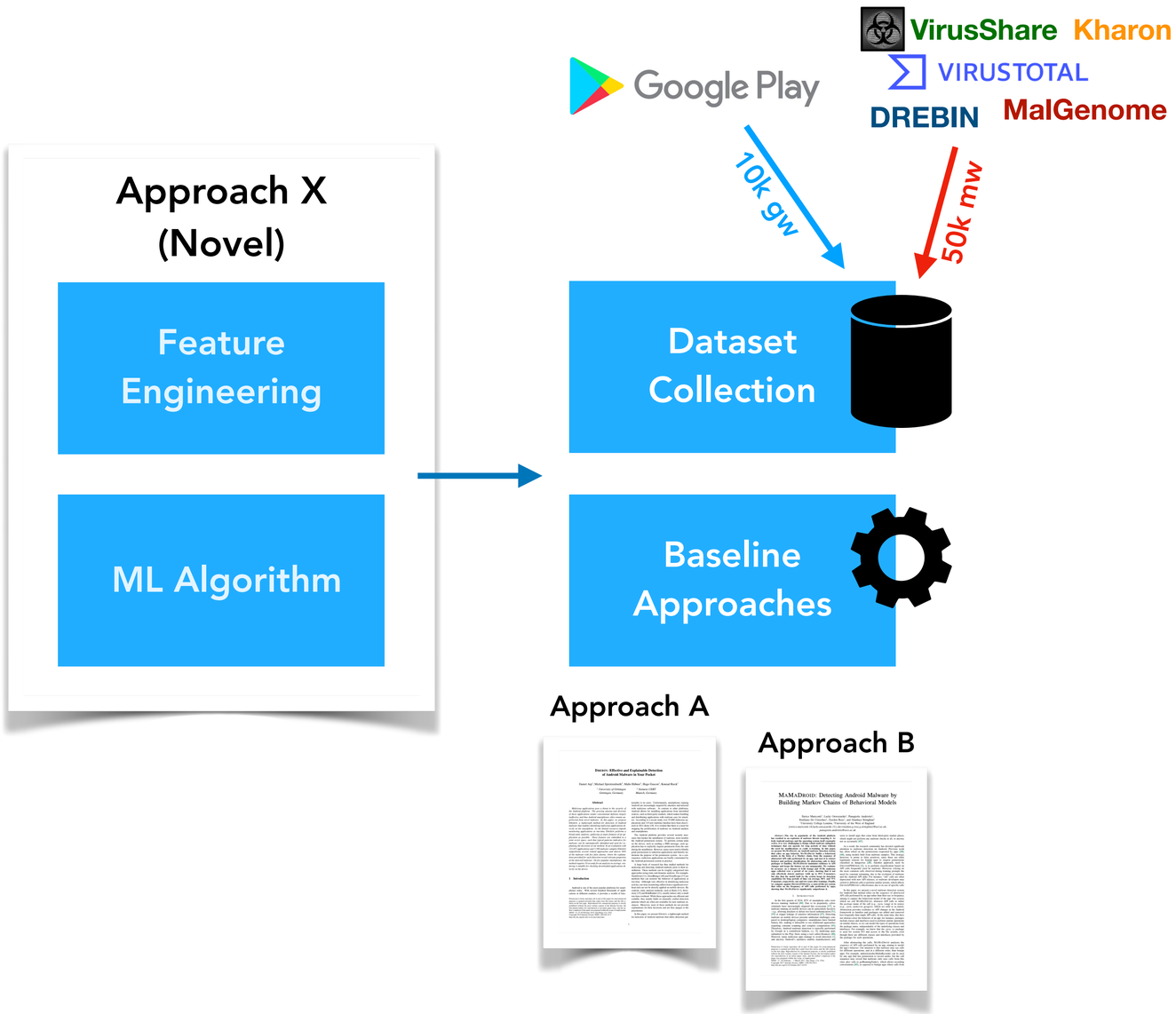
ML for Malware Detection 101



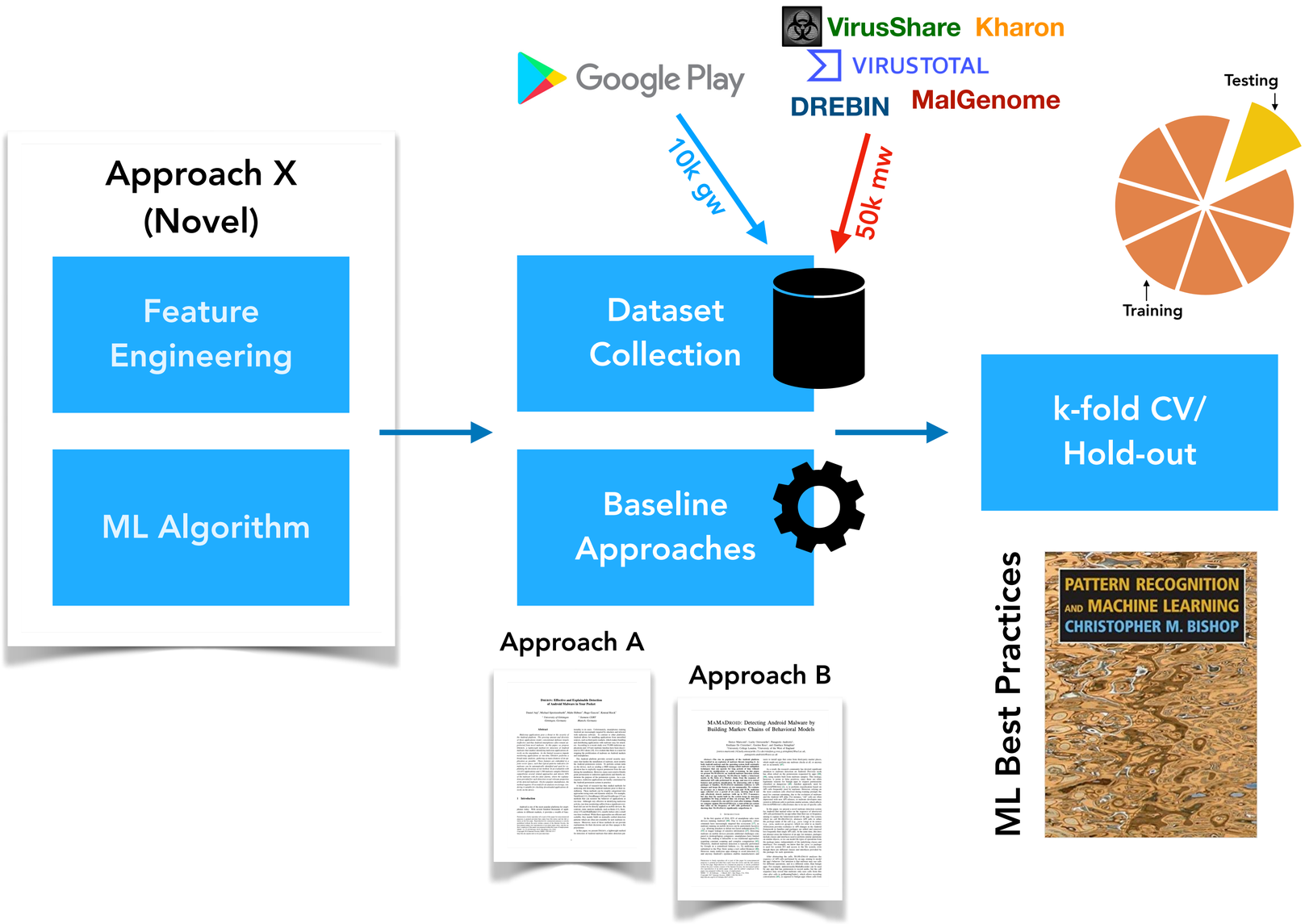
ML for Malware Detection 101



ML for Malware Detection 101



ML for Malware Detection 101



ML for Malware Detection 101

Approach X
(Novel)

Feature Engineering

ML Algorithm

Google Play

VirusShare Kharon

VIRUSTOTAL

DREBIN MalGenome

10k gw

50k mw

Dataset Collection

Baseline Approaches

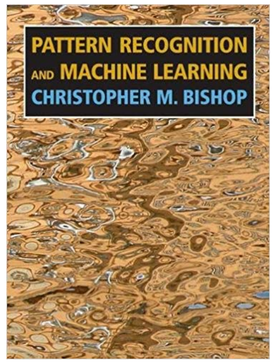
k-fold CV/
Hold-out

Compare Results

Approach A

Approach B

ML Best Practices



Approach	k-fold F1
X (New)	**0.99**
A	0.97
B	0.86

ML for Malware Detection 101

Three major unrealistic assumptions*

Approach X (Novel)

Feature Engineering

ML Algorithm

Google Play

VirusShare Kharon

VIRUSTOTAL

DREBIN MalGenome

10k gw

50k mw

Dataset Collection

Baseline Approaches

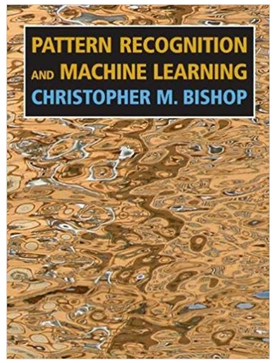
k-fold CV/ Hold-out

Compare Results

Approach A

Approach B

ML Best Practices



Approach	k-fold F1
X (New)	**0.99**
A	0.97
B	0.86

*** "Pitfalls" expanded in [USENIX Sec 2022]** Dos and Don'ts of Machine Learning for Computer Security
https://s2lab.cs.ucl.ac.uk/downloads/dodo.pdf

ML for Malware Detection 101

Three major unrealistic assumptions*

Approach X (Novel)

Feature Engineering

ML Algorithm

Google Play

VirusShare Kharon

VIRUSTOTAL

DREBIN MalGenome

10k gw

50k mw

Dataset Collection

Baseline Approaches

k-fold CV/ Hold-out

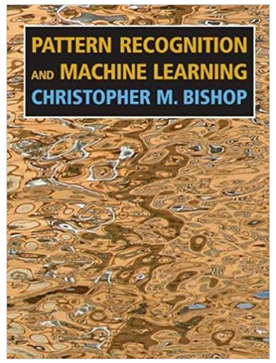
1

Compare Results

Approach A

Approach B

ML Best Practices



Approach	k-fold F1
X (New)	**0.99**
A	0.97
B	0.86

*** "Pitfalls" expanded in [USENIX Sec 2022]** Dos and Don'ts of Machine Learning for Computer Security
https://s2lab.cs.ucl.ac.uk/downloads/dodo.pdf

ML for Malware Detection 101

Three major unrealistic assumptions*

Approach X (Novel)

Feature Engineering

ML Algorithm

Google Play

VirusShare Kharon

VIRUSTOTAL

DREBIN MalGenome

10k gw

50k mw

2,3

Dataset Collection

Baseline Approaches

k-fold CV/ Hold-out

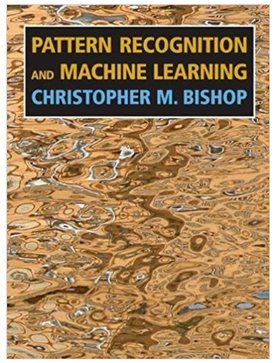
1

Compare Results

Approach A

Approach B

ML Best Practices



Approach	k-fold F1
X (New)	**0.99**
A	0.97
B	0.86

*** "Pitfalls" expanded in [USENIX Sec 2022]** Dos and Don'ts of Machine Learning for Computer Security
https://s2lab.cs.ucl.ac.uk/downloads/dodo.pdf

TESSERACT Framework

[USENIX Sec 2019] **TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time**
https://s2lab.cs.ucl.ac.uk/projects/tesseract

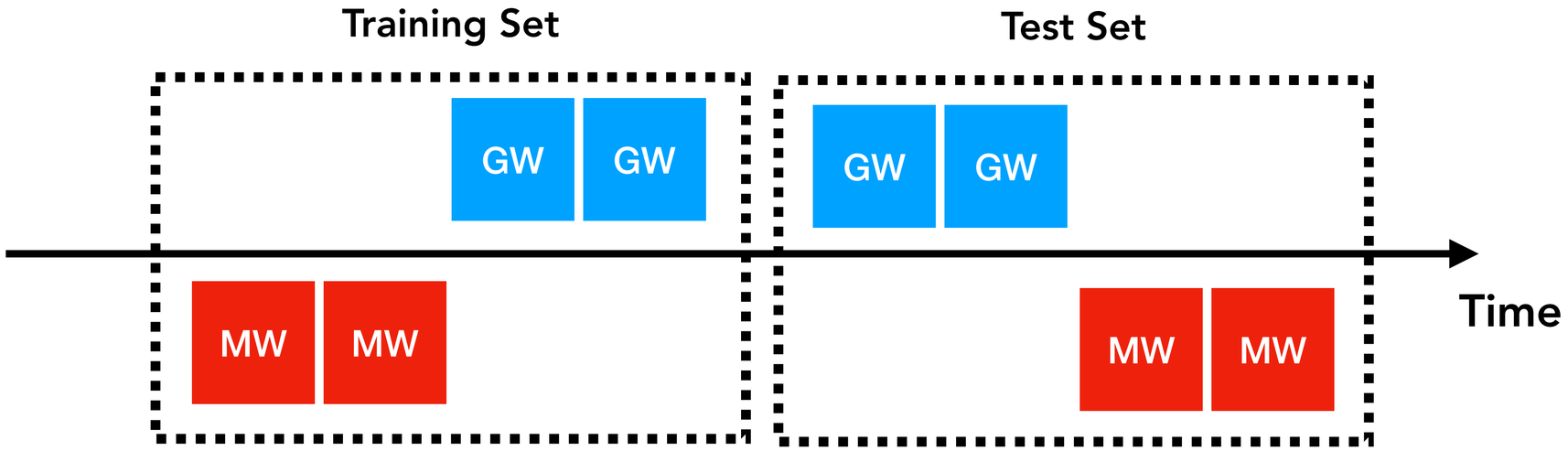
TESSERACT Framework

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Framework

Experimental Constraints

C1 Temporal training consistency → time(training) < time(testing)



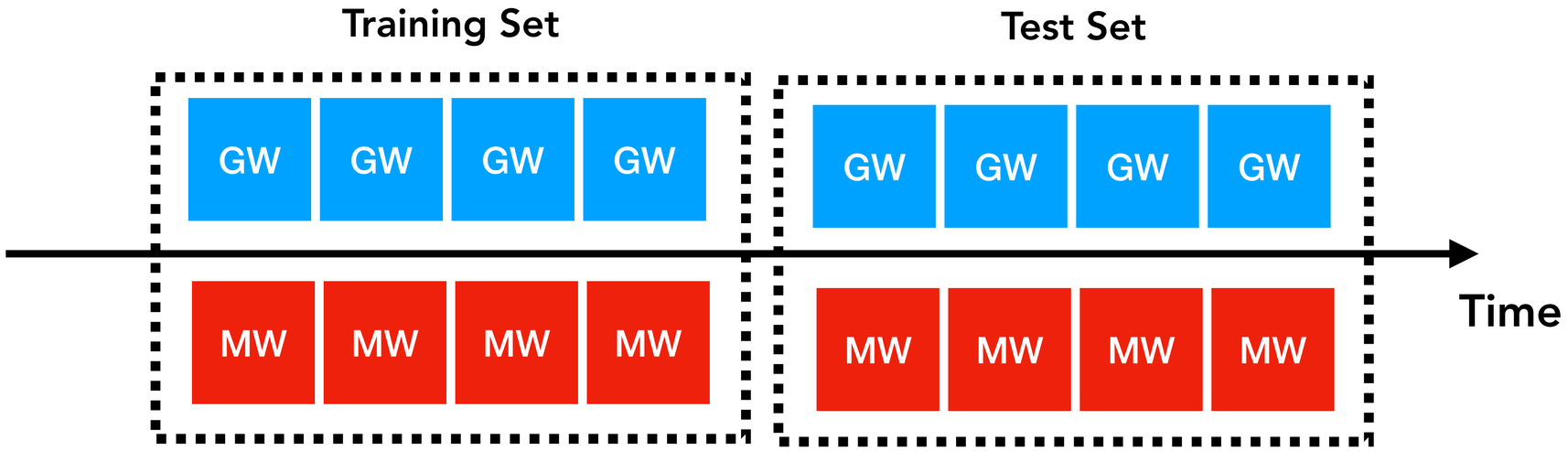
[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Framework

Experimental Constraints

C1 Temporal training consistency → time(training) < time(testing)

C2 {good|mal}ware temporal consistency → time(gw) = time(mw)

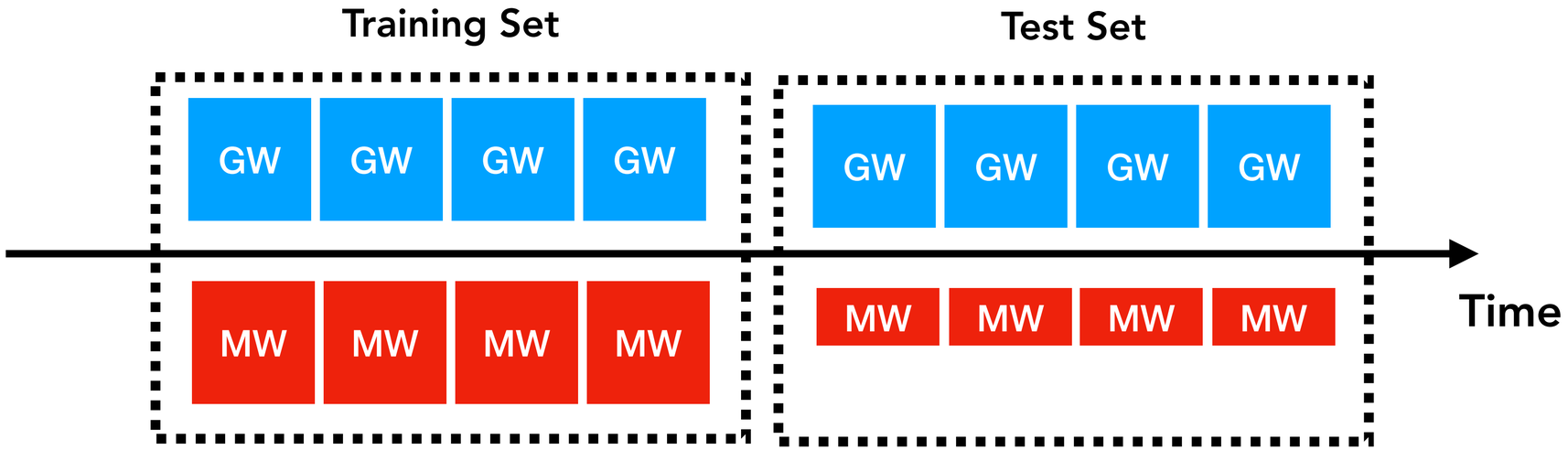


[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Framework

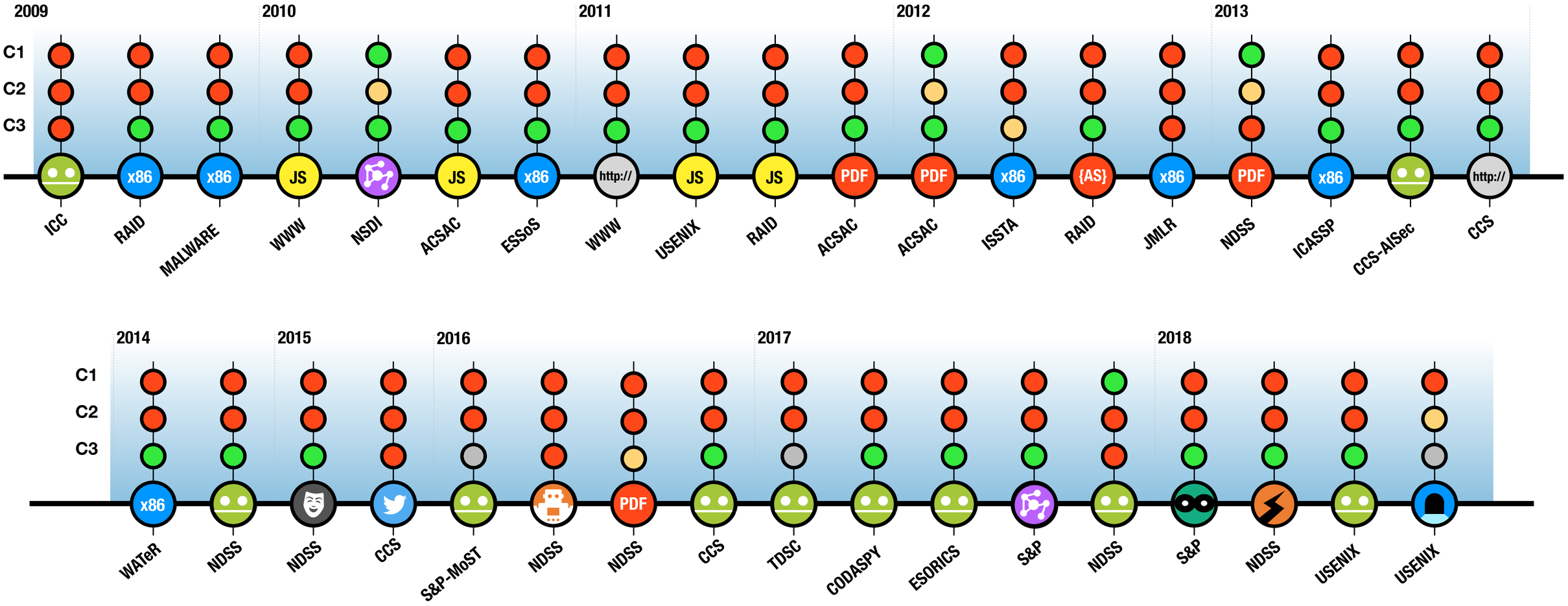
Experimental Constraints

C1	Temporal training consistency	→ time(training) < time(testing)
C2	{good|mal}ware temporal consistency	→ time(gw) = time(mw)
C3	Realistic testing classes ratio	→ realistic %mw in test



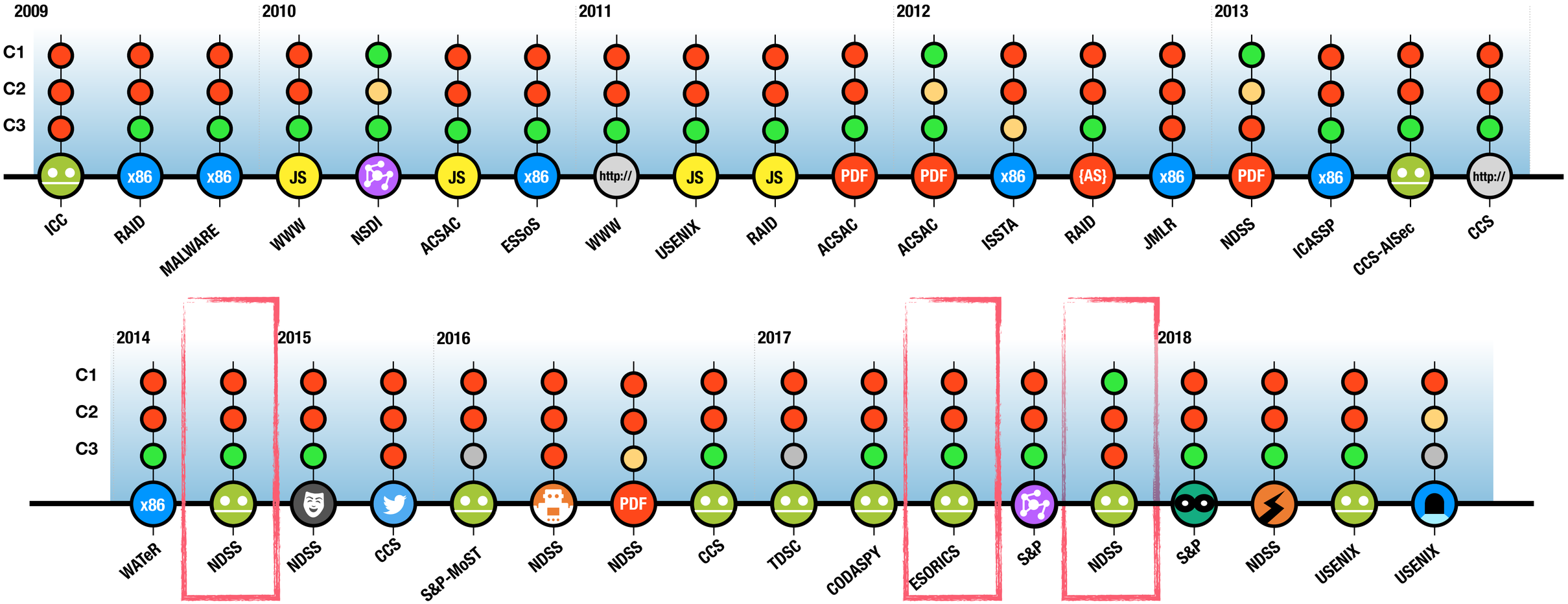
[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

Endemic Problem



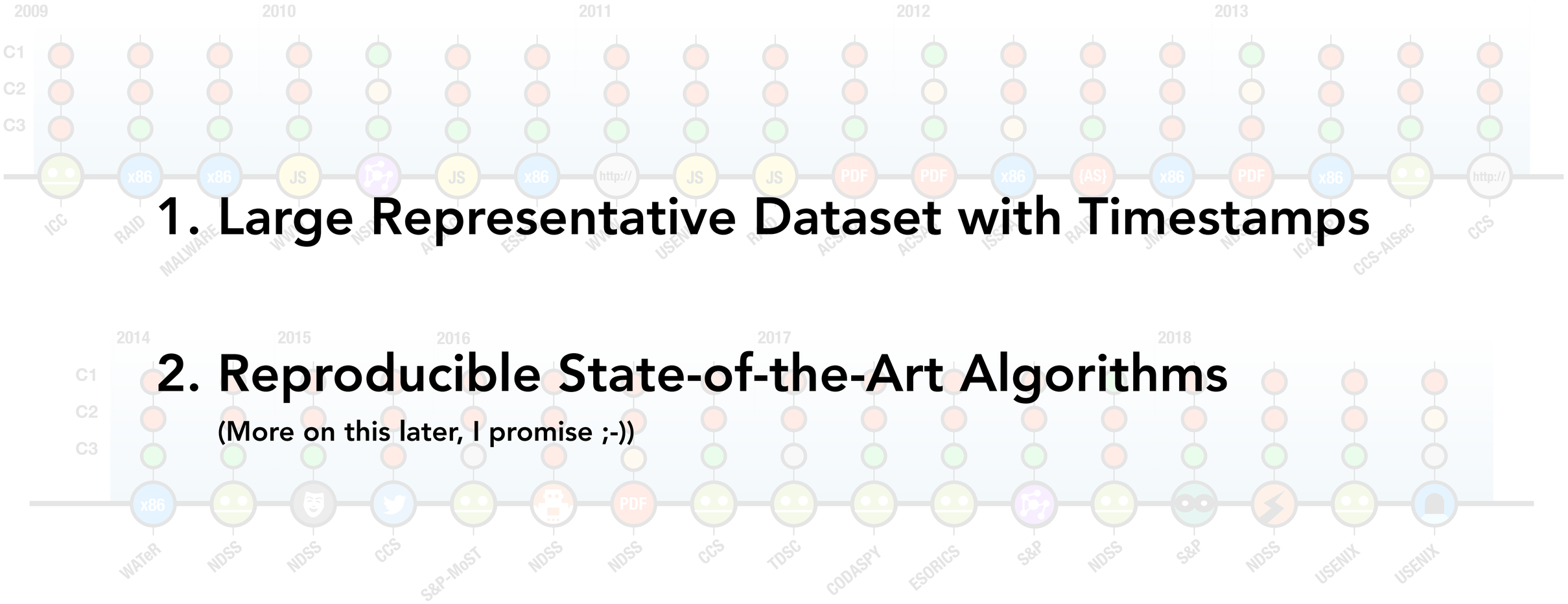
[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

Endemic Problem



[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

Endemic Problem



1. Large Representative Dataset with Timestamps

2. Reproducible State-of-the-Art Algorithms

   (More on this later, I promise ;-))

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

Dataset

- **129,729** Android applications from **AndroZoo**
- **10%** malware
- Covering **3 years** (2014 to 2016)

TESSERACT Evaluations

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Evaluations

Experimental Constraints

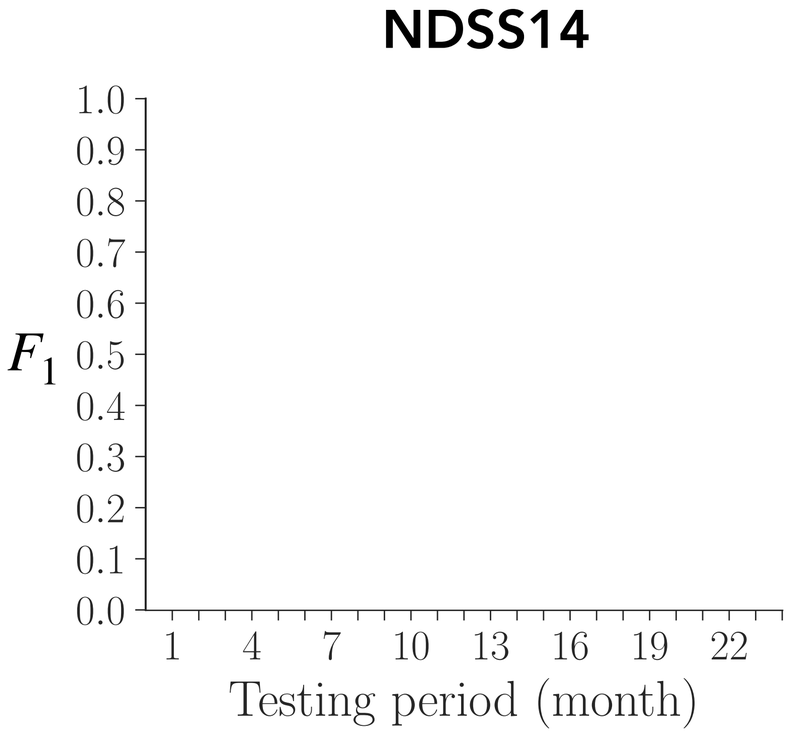
C1	Temporal training consistency
C2	{good|mal}ware temporal consistency
C3	Realistic testing classes ratio

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Evaluations

Experimental Constraints

C1	Temporal training consistency
C2	{good|mal}ware temporal consistency
C3	Realistic testing classes ratio

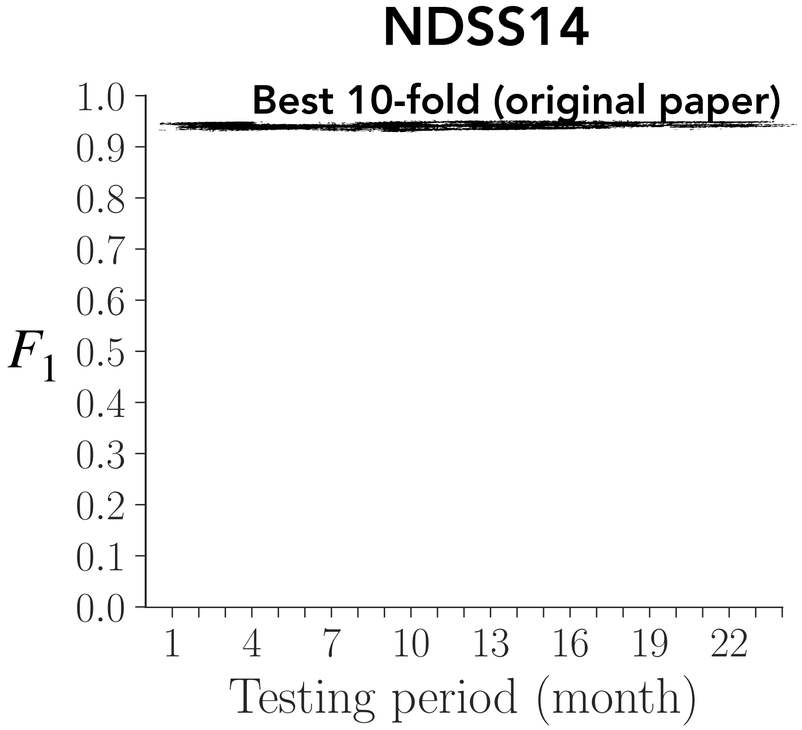


[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Evaluations

Experimental Constraints

C1	Temporal training consistency
C2	{good|mal}ware temporal consistency
C3	Realistic testing classes ratio

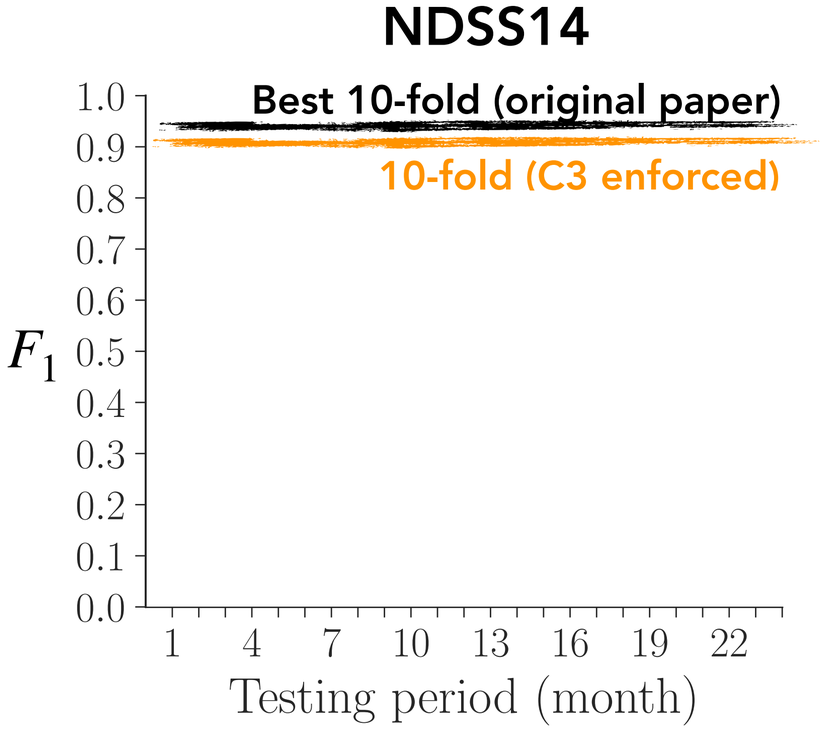


[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Evaluations

Experimental Constraints

C1	Temporal training consistency
C2	{good|mal}ware temporal consistency
C3	Realistic testing classes ratio

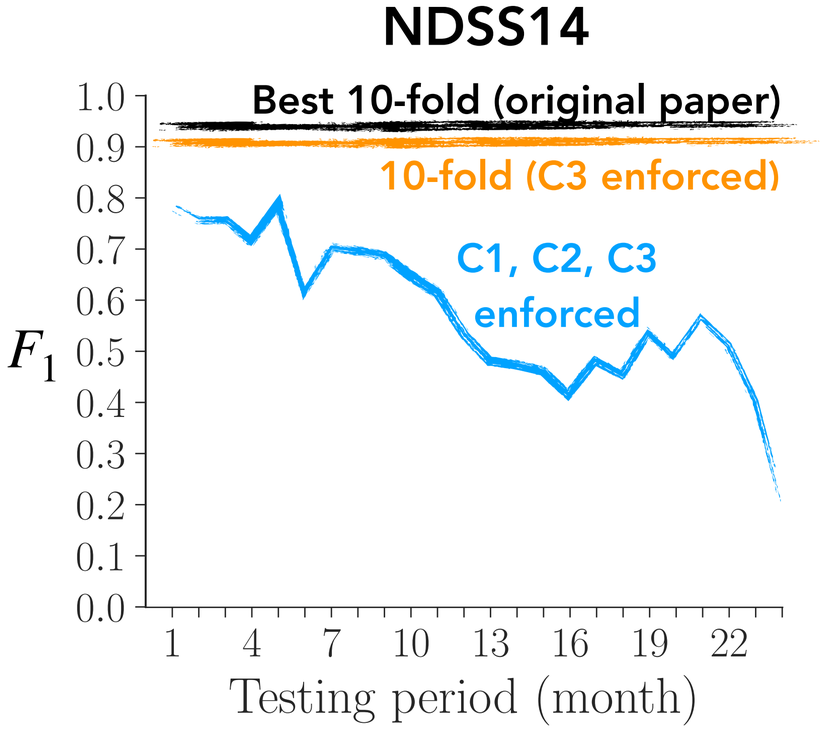


[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Evaluations

Experimental Constraints

C1	Temporal training consistency
C2	{good|mal}ware temporal consistency
C3	Realistic testing classes ratio

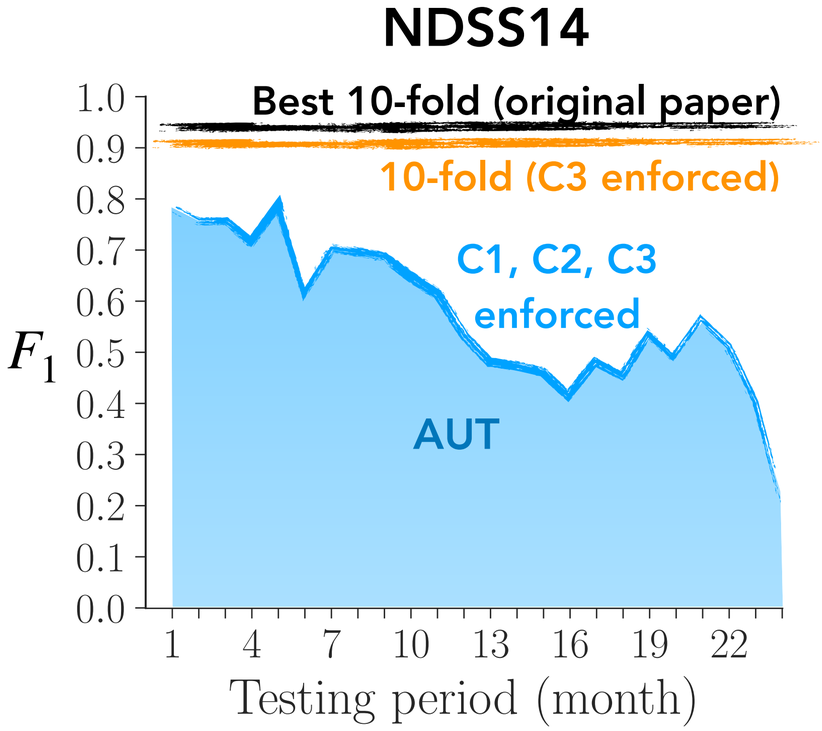


[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Evaluations

Experimental Constraints

C1	Temporal training consistency
C2	{good|mal}ware temporal consistency
C3	Realistic testing classes ratio



$$AUT(F_1, 24m) = 0.58$$

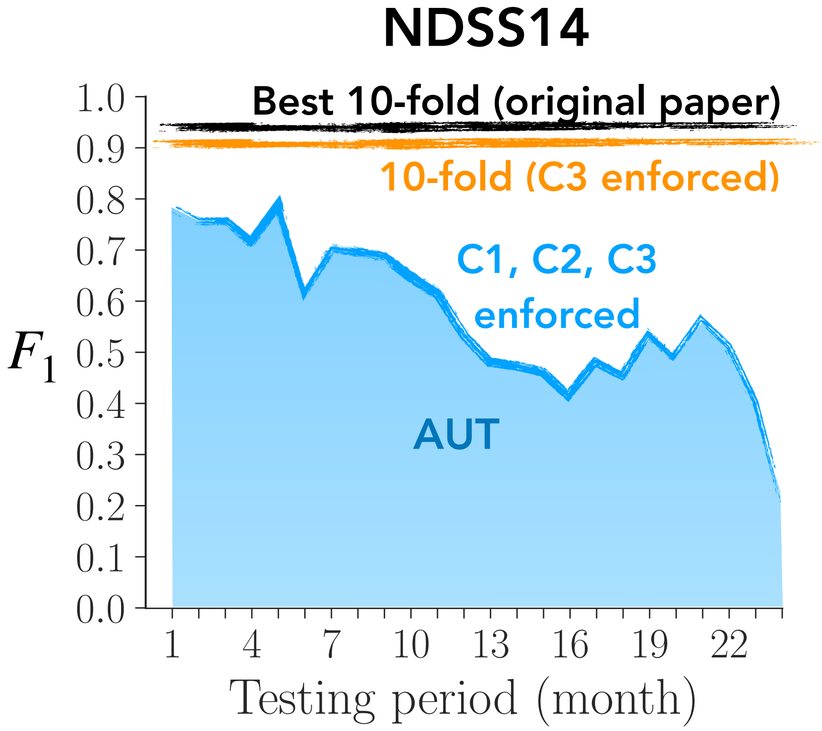
Area Under Time
AUT(Metric, Period)

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT Evaluations

Experimental Constraints

C1	Temporal training consistency
C2	{good|mal}ware temporal consistency
C3	Realistic testing classes ratio



$$AUT(F_1, 24m) = 0.58$$

Area Under Time
AUT(Metric, Period)

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

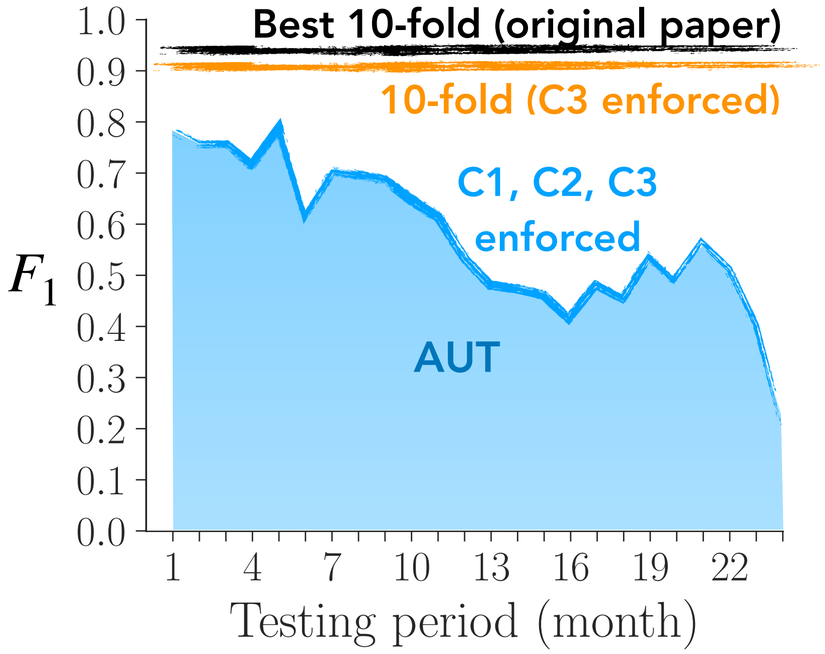
TESSERACT Evaluations

Experimental Constraints

- **C1** Temporal training consistency
- **C2** {good|mal}ware temporal consistency
- **C3** Realistic testing classes ratio

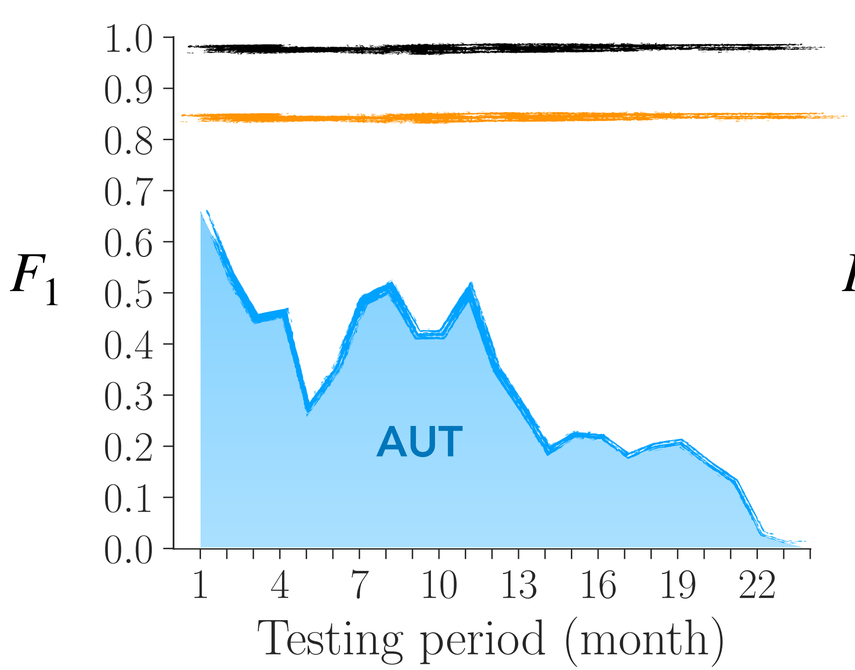
Area Under Time
AUT(Metric, Period)

NDSS14



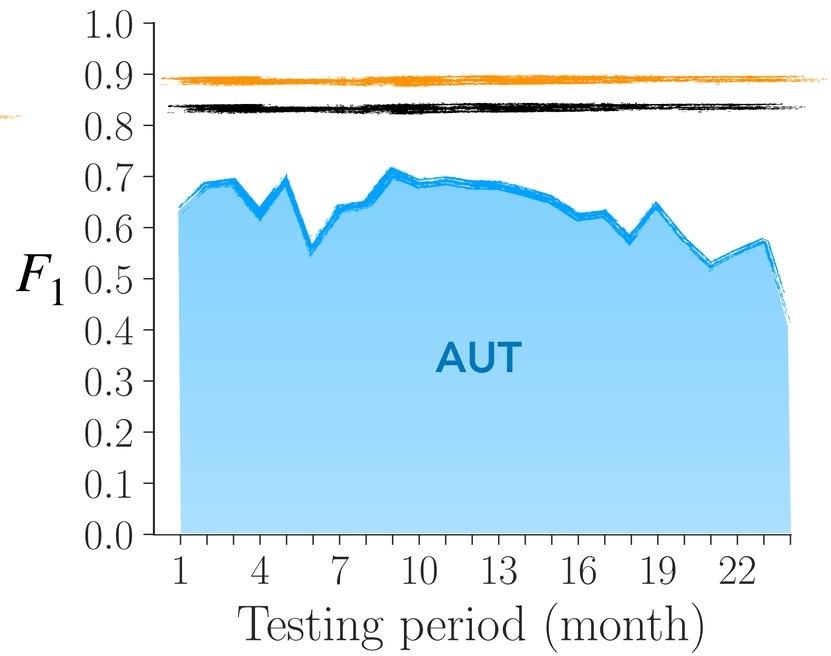
$$AUT(F_1, 24m) = 0.58$$

NDSS17



$$AUT(F_1, 24m) = 0.32$$

ESORICS17



$$AUT(F_1, 24m) = 0.64$$

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

TESSERACT: Actionable Points

TESSERACT: Actionable Points

Realistic Evaluations

- Reveals performance in more realistic setting
- Removes space-time experimental bias
- **Practitioners:** Choose Best Solution
- **Researchers**: Evaluate New Solutions

TESSERACT: Actionable Points

Realistic Evaluations

- Reveals performance in more realistic setting
- Removes space-time experimental bias
- **Practitioners:** Choose Best Solution
- **Researchers**: Evaluate New Solutions

Performance-Cost Trade Offs

- **Detection Performance** (e.g., AUT F_1)
- **Labeling Cost** for retraining (e.g., manpower)
- **Quarantine Cost** for rejection (e.g., low-confidence decisions)

Incremental Retraining Active Learning Online Learning **Rejection***

TESSERACT: Actionable Points

Realistic Evaluations

- Reveals performance in more realistic setting
- Removes space-time experimental bias
- **Practitioners:** Choose Best Solution
- **Researchers**: Evaluate New Solutions

Performance-Cost Trade Offs

- **Detection Performance** (e.g., AUT F_1)
- **Labeling Cost** for retraining (e.g., manpower)
- **Quarantine Cost** for rejection (e.g., low-confidence decisions)

Incremental Retraining Active Learning Online Learning **Rejection***

- Incremental retraining costly?*
- Active learning and query strategies* **
- Online learning and the risk of self-poisoning+

*[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
**[arXiv 2023] Chen et al. Continuous Learning for Android Malware Detection

+[AISec 2021] Investigating Labelless Drift Adaptation for Malware Detection — https://s2lab.cs.ucl.ac.uk/downloads/aisec47-kanA.pdf

TESSERACT: Actionable Points

Realistic Evaluations

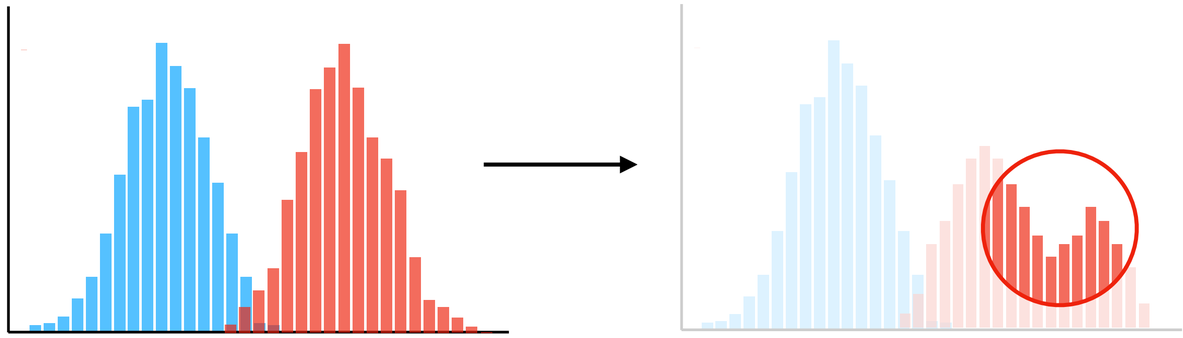
- Reveals performance in more realistic setting
- Removes space-time experimental bias
- **Practitioners:** Choose Best Solution
- **Researchers**: Evaluate New Solutions

Performance-Cost Trade Offs

- **Detection Performance** (e.g., AUT F_1)
- **Labeling Cost** for retraining (e.g., manpower)
- **Quarantine Cost** for rejection (e.g., low-confidence decisions)

Incremental Retraining Active Learning Online Learning **Rejection***

As well as measuring the overall effect of drift we can **identify** specific aspects of the drift and **reject** objects that are likely to be misclassified.



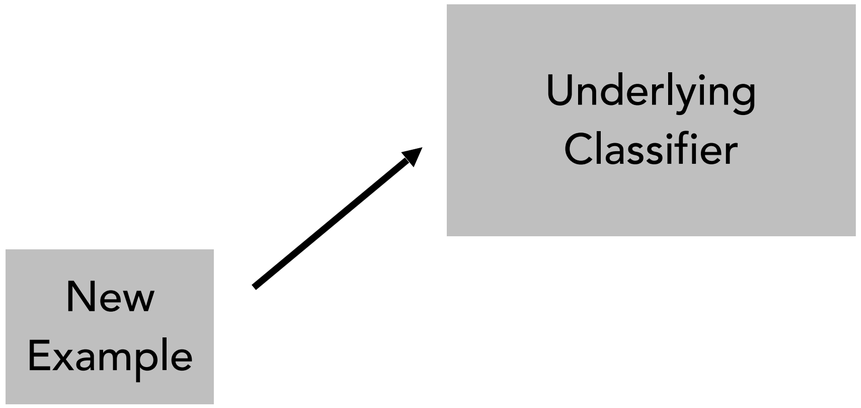
*** [USENIX Sec 2017]** Transcend: Detecting Concept Drift in Malware Classification Models
*** [IEEE S&P 2022]** Transcending Transcend: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend

Revisiting Classification in the Presence of Concept Drift

Revisiting Classification in the Presence of Concept Drift

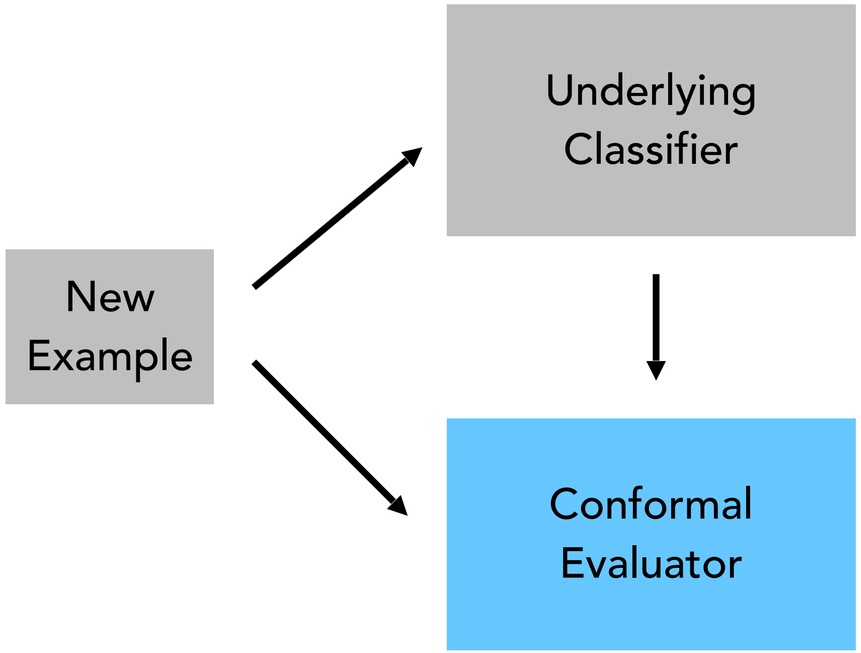
Covariate Shift: Change in feature distribution	$P(x \in X)$
Prior-probability Shift: Change in class base rate	$P(y \in Y)$
Concept Drift: Change in ground truth definition	$P(y \in Y \mid x \in X)$

Transcend at Test Time



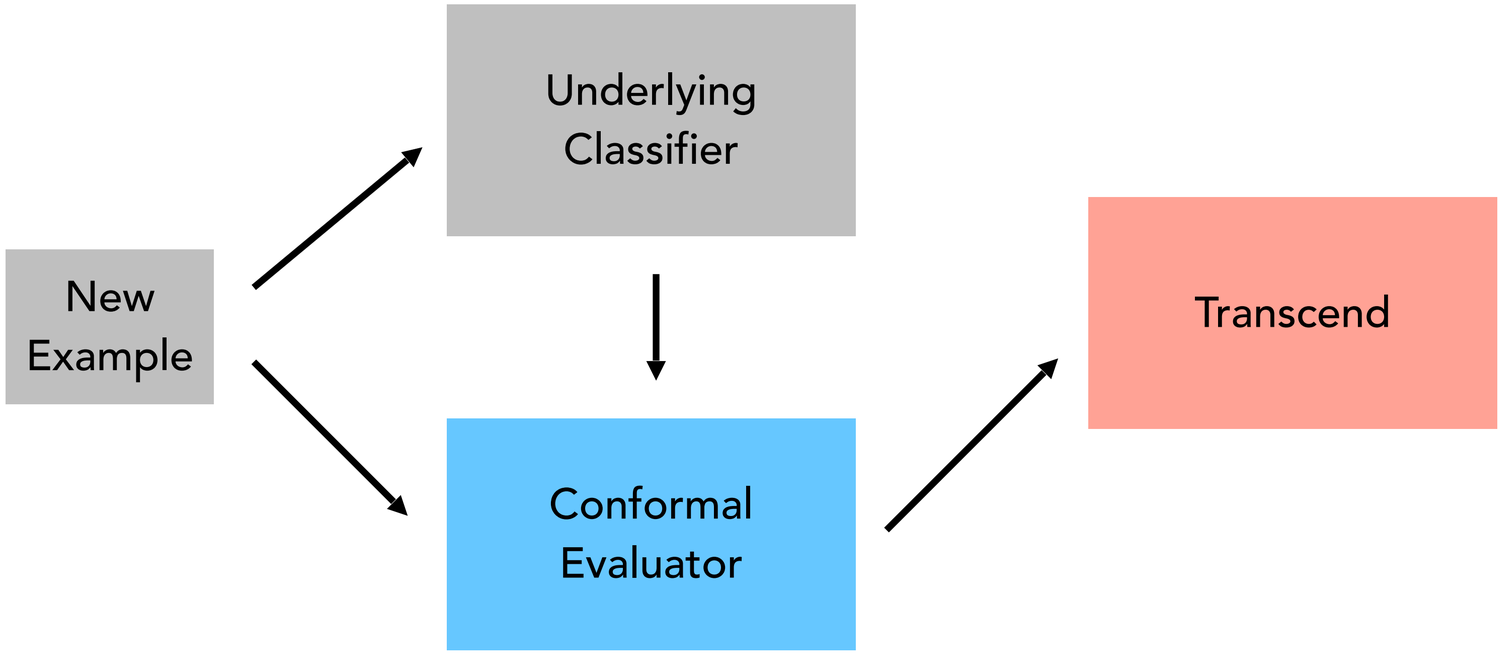
[USENIX Sec 2017] Transcend: Detecting Concept Drift in Malware Classification Models
https://s2lab.cs.ucl.ac.uk/projects/transcend/

Transcend at Test Time



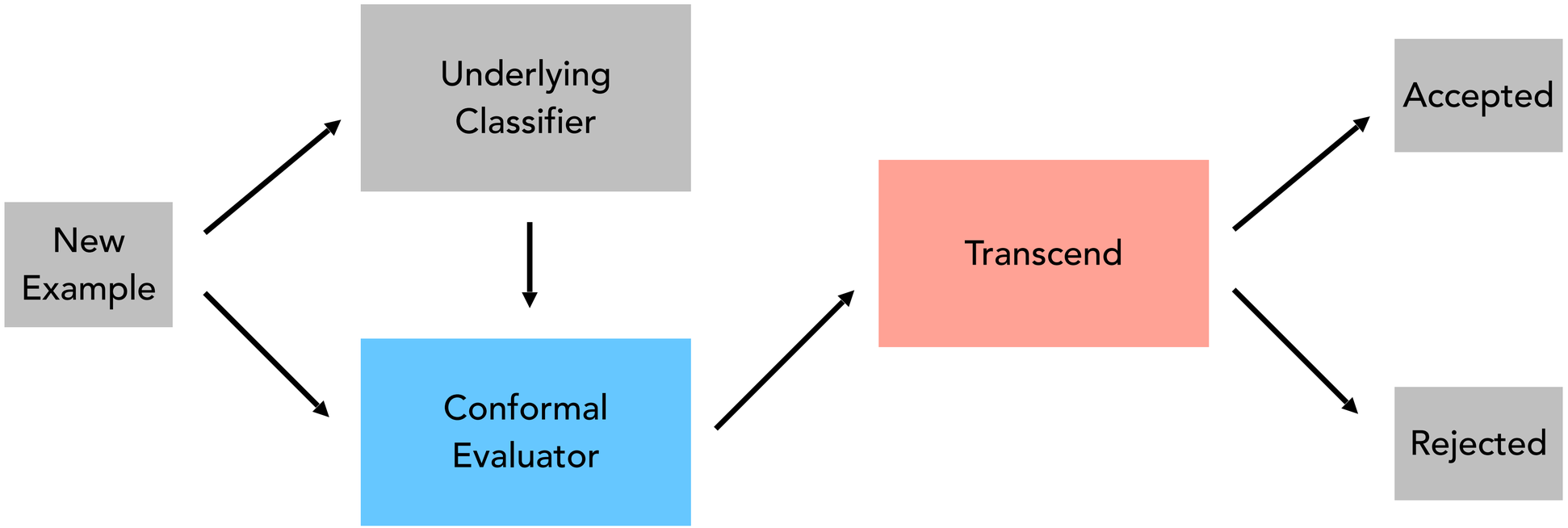
[USENIX Sec 2017] Transcend: Detecting Concept Drift in Malware Classification Models
https://s2lab.cs.ucl.ac.uk/projects/transcend/

Transcend at Test Time



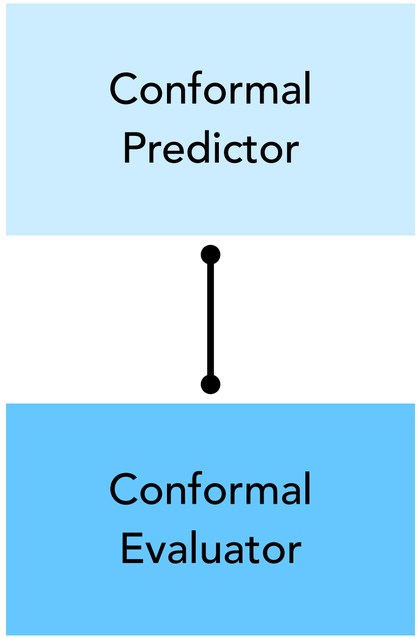
[USENIX Sec 2017] Transcend: Detecting Concept Drift in Malware Classification Models
https://s2lab.cs.ucl.ac.uk/projects/transcend/

Transcend at Test Time



[USENIX Sec 2017] Transcend: Detecting Concept Drift in Malware Classification Models
https://s2lab.cs.ucl.ac.uk/projects/transcend/

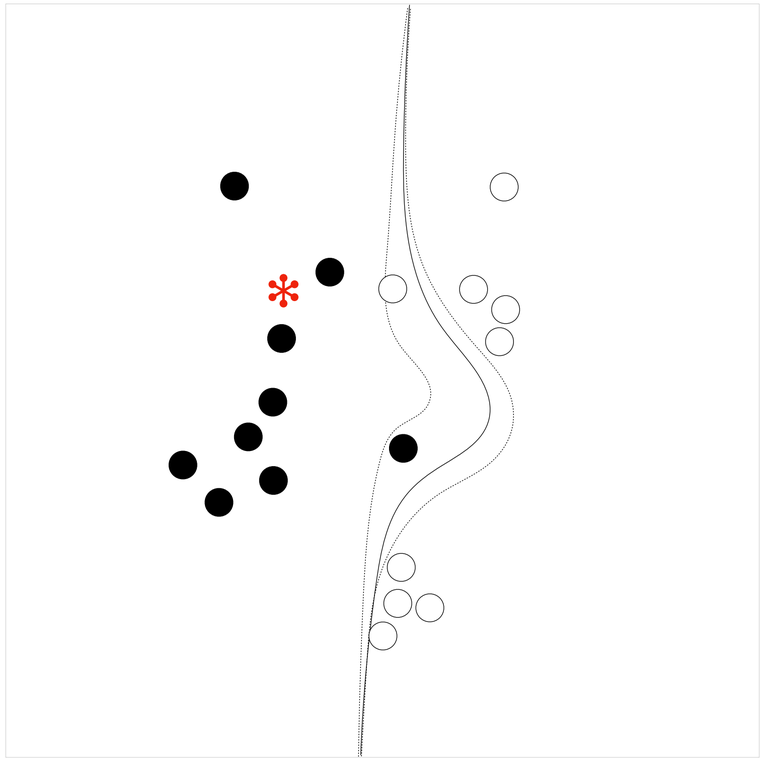
Conformal Prediction and Evaluation



- CP theory lays foundation for CE
- CPs outputs prediction sets with guaranteed confidence 1 - ε
- CPs rely on two assumptions:
  - **Exchangeability**: Generalization of i.i.d.
  - **Closed-world**: Fixed label space

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

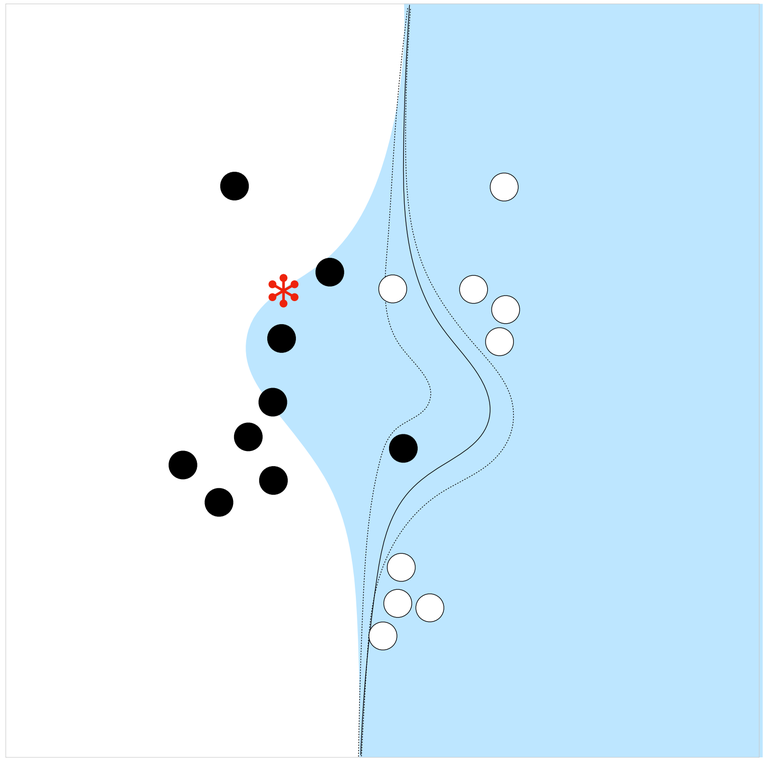
Conformal Prediction and Non-Conformity Measure (NCM)



SVM Polynomial

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

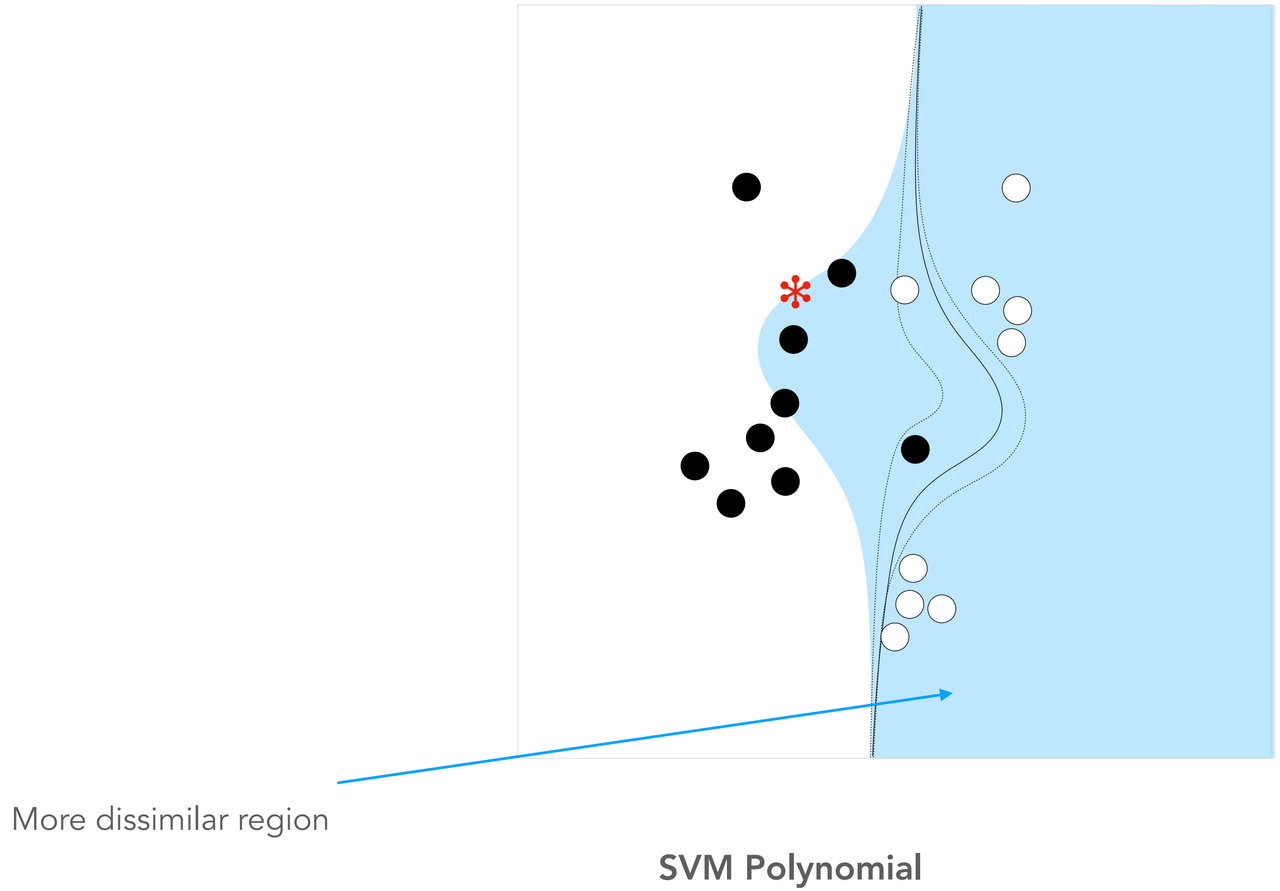
Conformal Prediction and Non-Conformity Measure (NCM)



SVM Polynomial

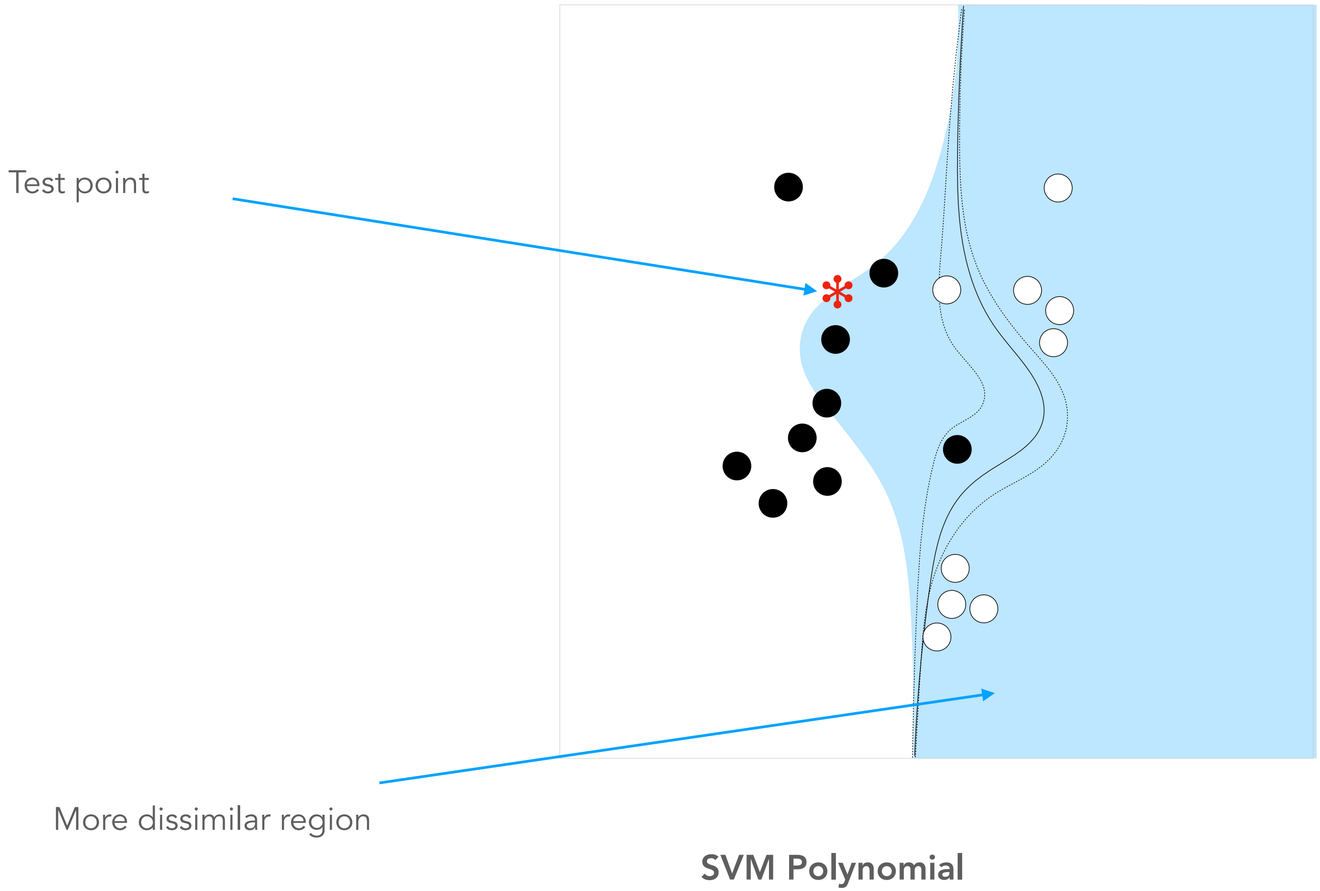
[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

Conformal Prediction and Non-Conformity Measure (NCM)



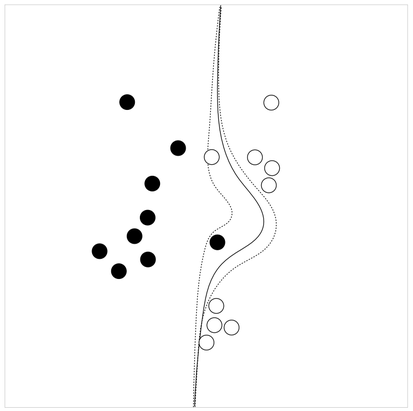
[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

Conformal Prediction and Non-Conformity Measure (NCM)

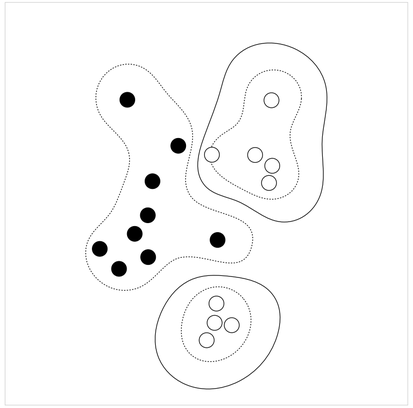


[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

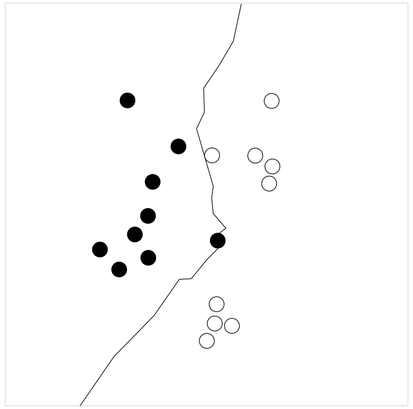
Conformal Prediction and Non-Conformity Measure (NCM)



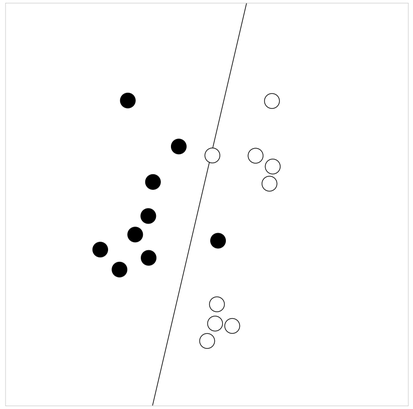
SVM Polynomial



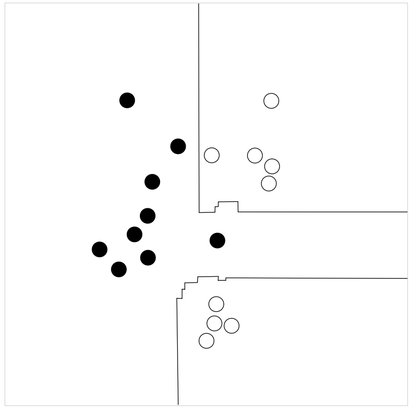
SVM RBF



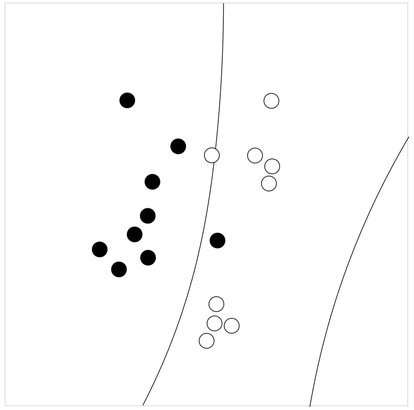
3NN



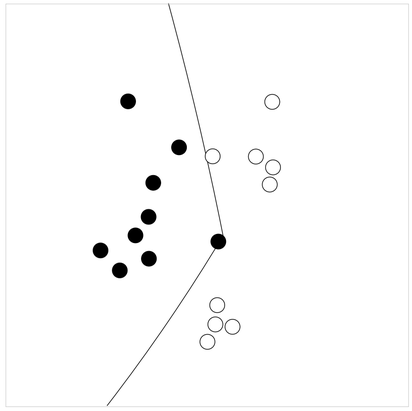
Nearest Centroid



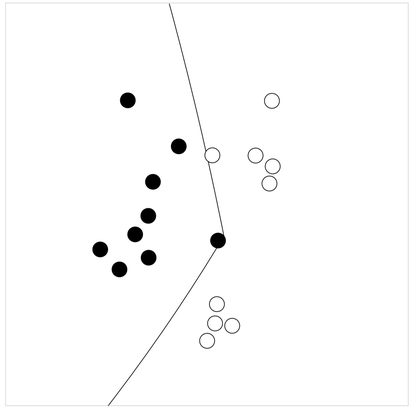
Random Forests



QDA

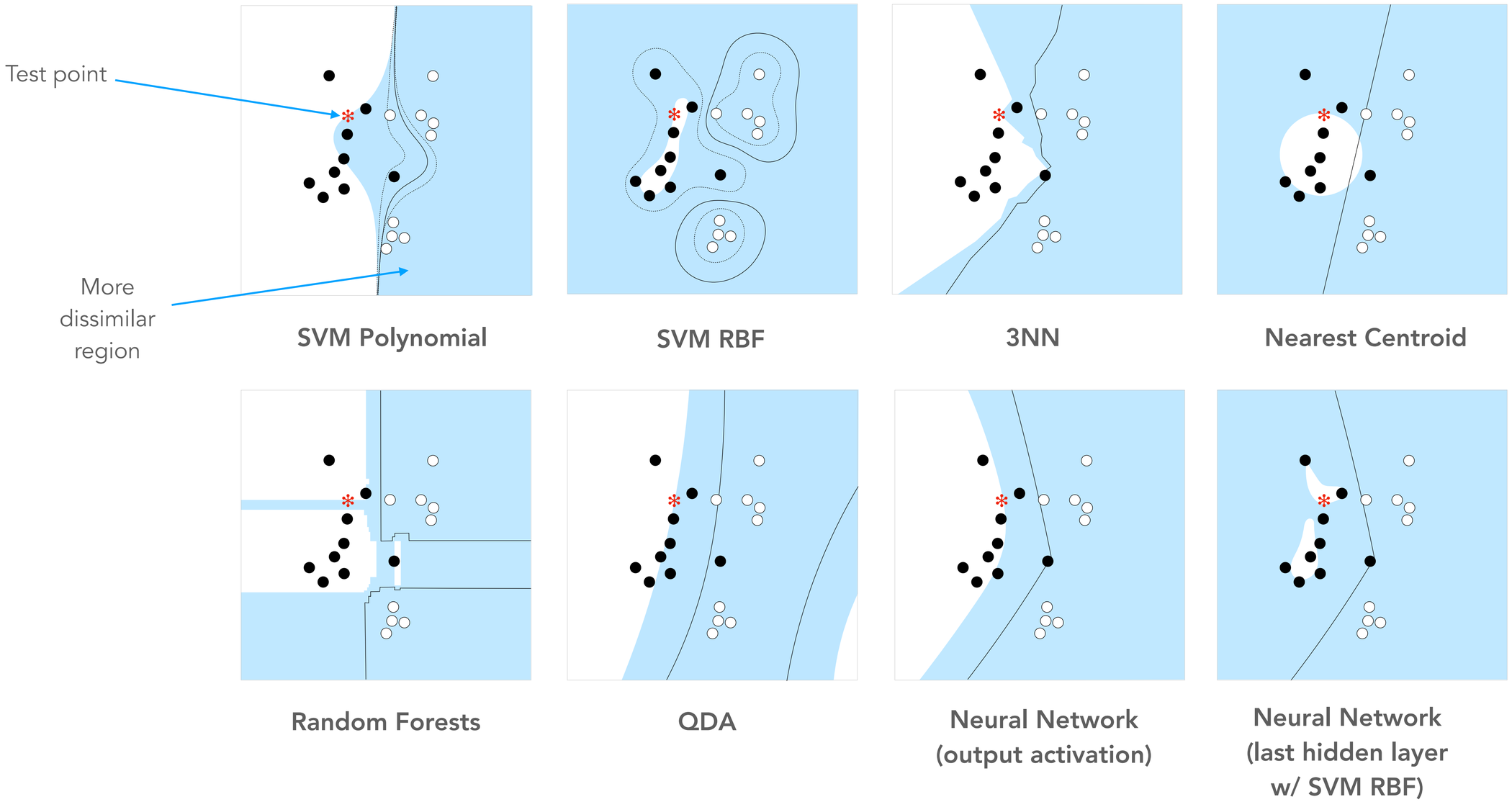


Neural Network
(output activation)



Neural Network
(last hidden layer
w/ SVM RBF)

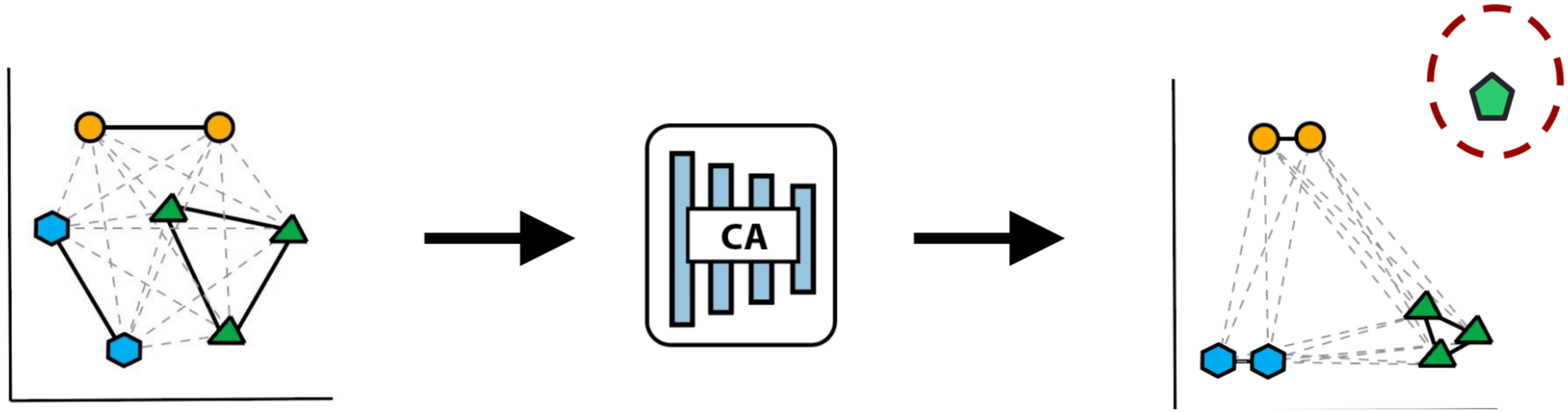
Conformal Prediction and Non-Conformity Measure (NCM)



Contrastive Learning Representation and Transcendent's NCM

Aside

- Use contrastive learning to learn a compressed representation of the training data by contrasting with existing samples



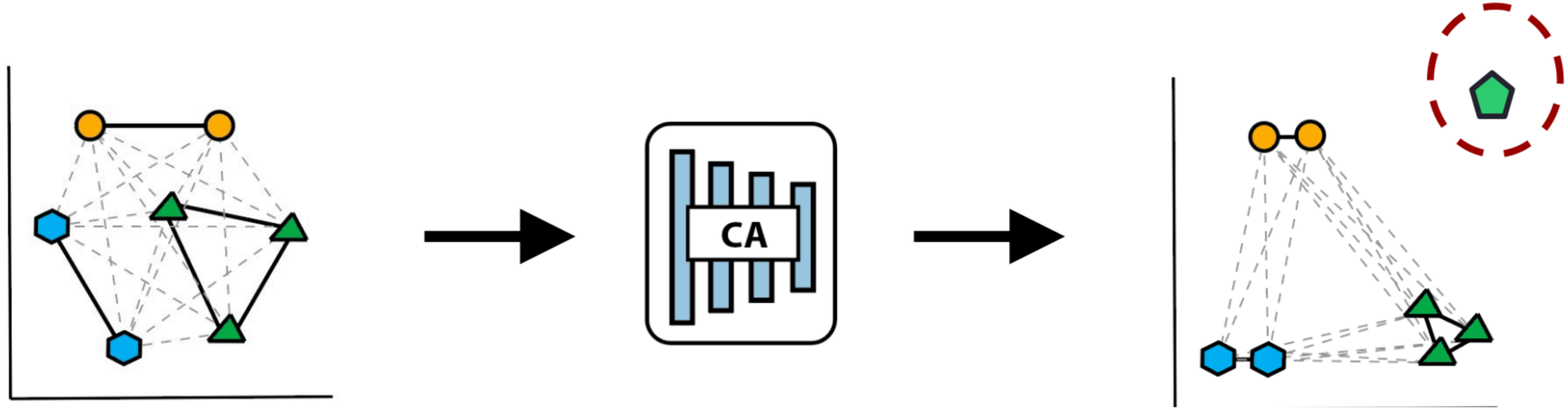
Yang et al. CADE: Detecting and Explaining Concept Drift Samples for Security Applications. USENIX Sec 2021

(Slide courtesy of Gang Wang)

Contrastive Learning Representation and Transcendent's NCM

Aside

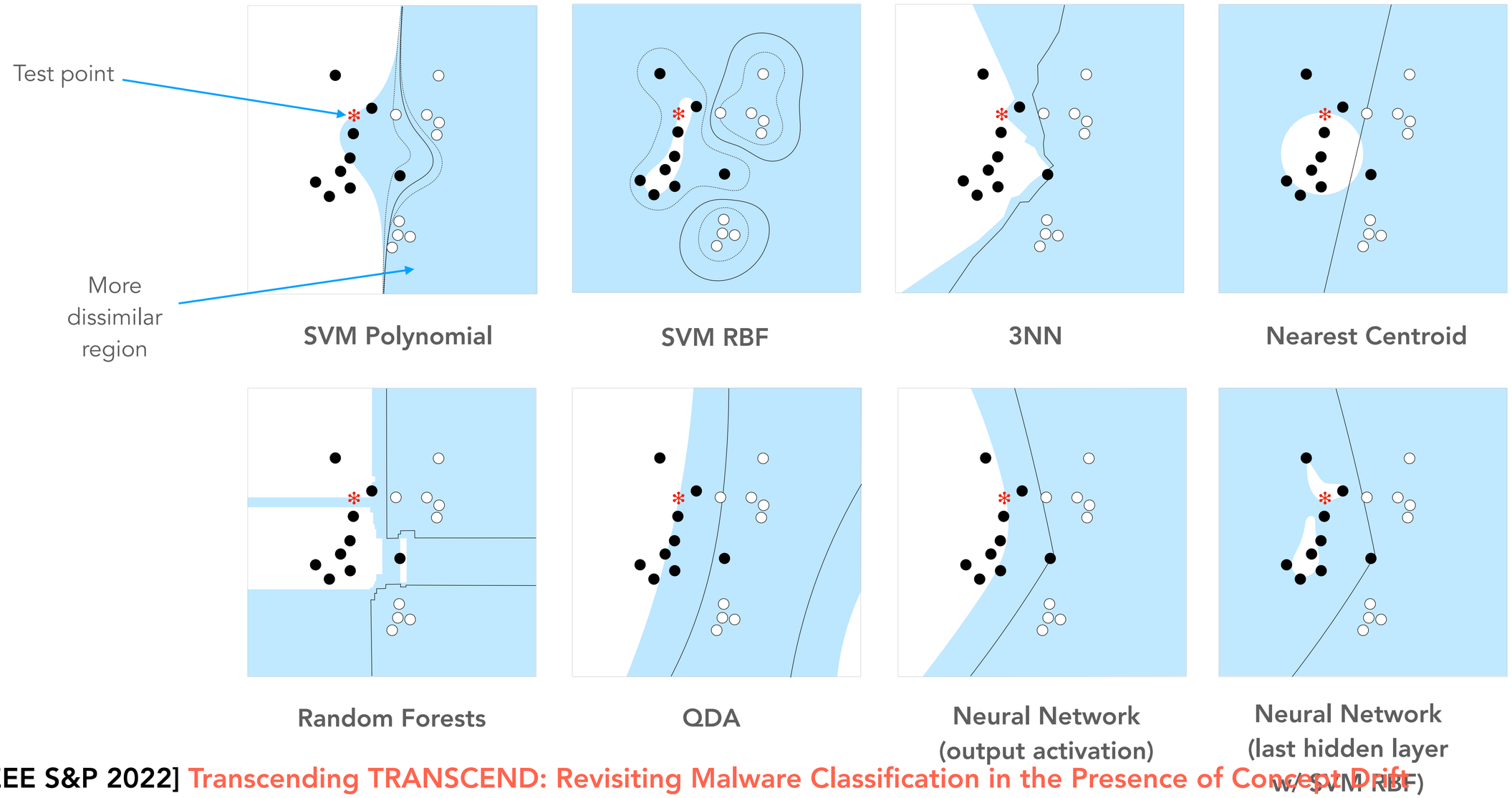
- Use contrastive learning to learn a compressed representation of the training data by contrasting with existing samples



Yang et al. CADE: Detecting and Explaining Concept Drift Samples for Security Applications. USENIX Sec 2021
(Slide courtesy of Gang Wang)

- CL seems ideal to be used as a non-conformity measure (!)

Conformal Prediction and Non-Conformity Measure (NCM)

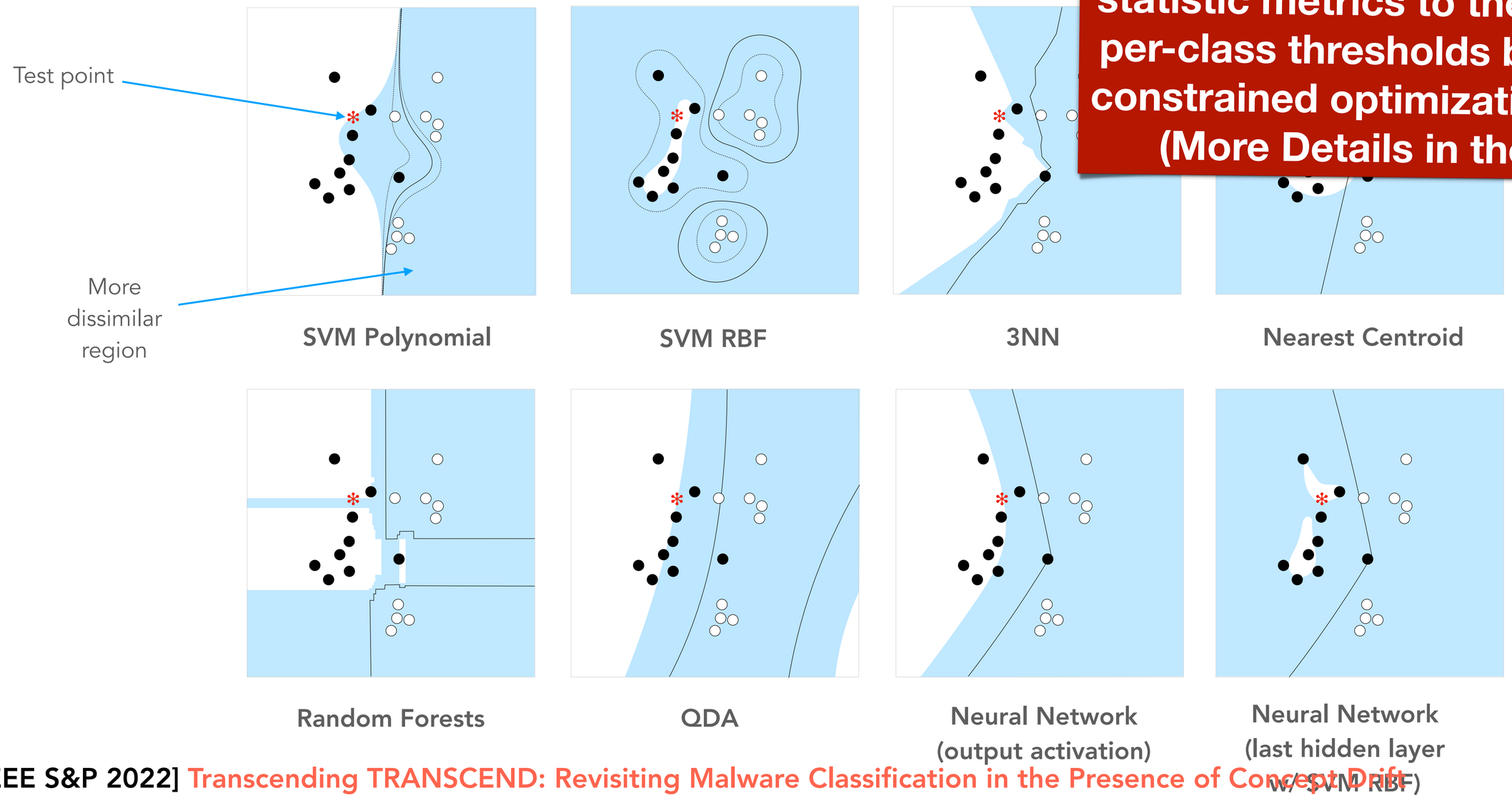


[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift

https://s2lab.cs.ucl.ac.uk/projects/transcend/

Conformal Prediction and Non-Conformity Measure (NCM)

From NCMs to p-values and statistic metrics to then compute per-class thresholds by solving a constrained optimization problem (More Details in the Paper)



[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift

https://s2lab.cs.ucl.ac.uk/projects/transcend/

Experimental Setup

Experimental Setup

Android

- DREBIN w/ ~260K apps (Jan 2014 - Dec 2018)
- Linear SVM, binary feature space

Experimental Setup

Android

- DREBIN w/ ~260K apps (Jan 2014 - Dec 2018)
- Linear SVM, binary feature space

Windows PE

- EMBER v2 w/ ~117K apps (Jan 2017 - Dec 2017)
- Gradient Boosted Decision Tree (GBDT)

Experimental Setup

Android

- DREBIN w/ ~260K apps (Jan 2014 - Dec 2018)
- Linear SVM, binary feature space

Windows PE

- EMBER v2 w/ ~117K apps (Jan 2017 - Dec 2017)
- Gradient Boosted Decision Tree (GBDT)

PDF

- Hidost w/ ~189k apps (Aug 2017 - Sep 2017)
- Random Forest, features robust to drift

Experimental Setup

Android

- DREBIN w/ ~260K apps (Jan 2014 - Dec 2018)
- Linear SVM, binary feature space

Windows PE

- EMBER v2 w/ ~117K apps (Jan 2017 - Dec 2017)
- Gradient Boosted Decision Tree (GBDT)

PDF

- Hidost w/ ~189k apps (Aug 2017 - Sep 2017)
- Random Forest, features robust to drift

Thresholding Optimization

- Constraints: minimum F1 of 0.9 for kept elements @ rejection rate < 15%

Experimental Setup

Android

- DREBIN w/ ~260K apps (Jan 2014 - Dec 2018)
- Linear SVM, binary feature space

Windows PE

- EMBER v2 w/ ~117K apps (Jan 2017 - Dec 2017)
- Gradient Boosted Decision Tree (GBDT)

PDF

- Hidost w/ ~189k apps (Aug 2017 - Sep 2017)
- Random Forest, features robust to drift

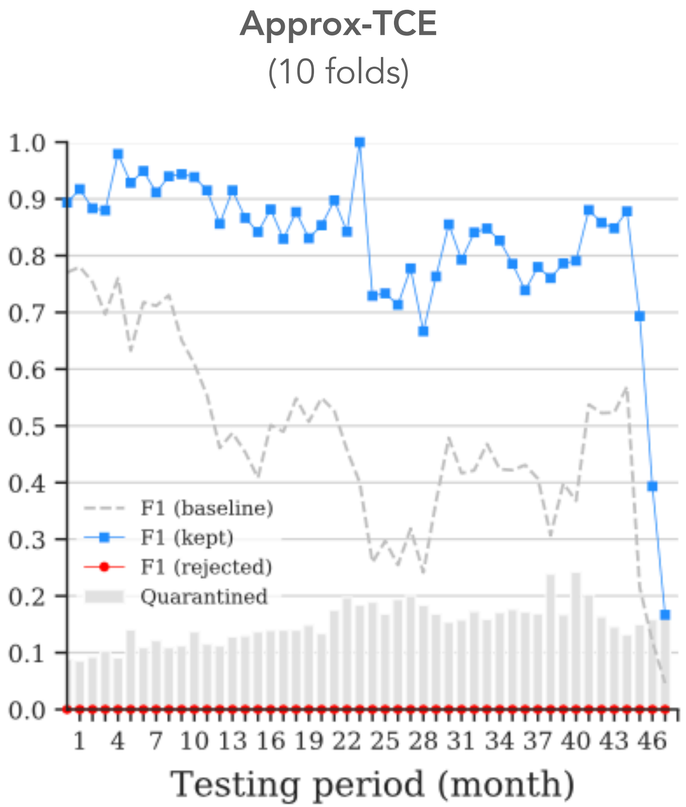
Thresholding Optimization

- Constraints: minimum F1 of 0.9 for kept elements @ rejection rate < 15%

Results: Rejection Performance

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

Results: Rejection Performance

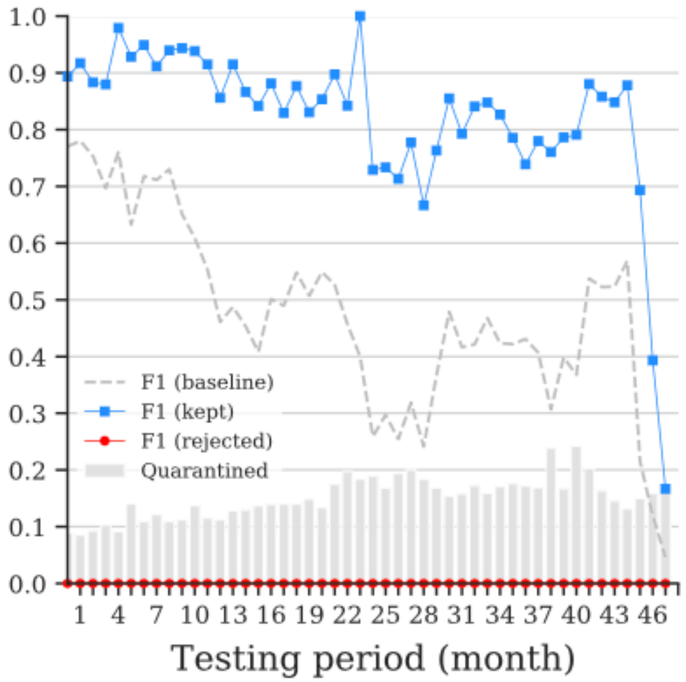



```
AUT(F1, kept):       0.82
AUT(F1, rejected):   0.00
CPU hours:           46.1
```

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

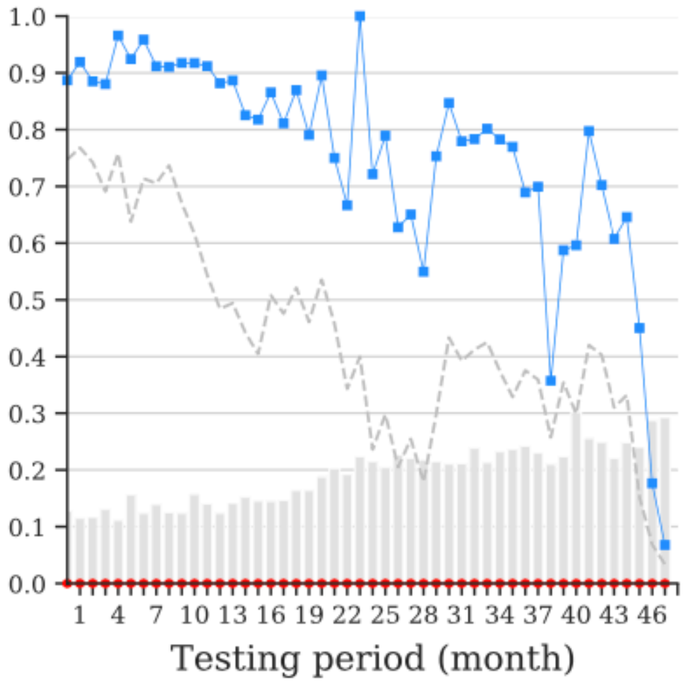
Results: Rejection Performance

Approx-TCE
(10 folds)




```
AUT(F1, kept):       0.82
AUT(F1, rejected):   0.00
CPU hours:           46.1
```

ICE
(0.33 calibration split)

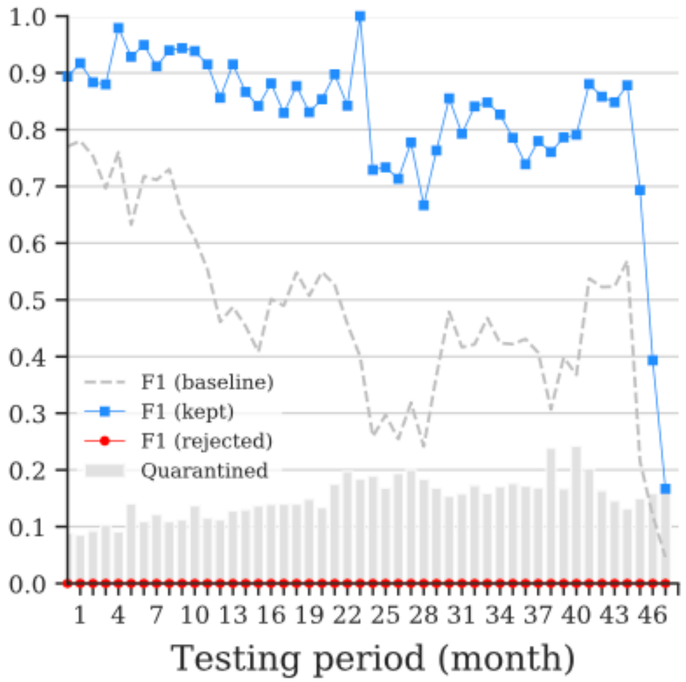



```
AUT(f1, kept):       0.76
AUT(f1, rejected):   0.00
CPU hours:           11.5
```

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

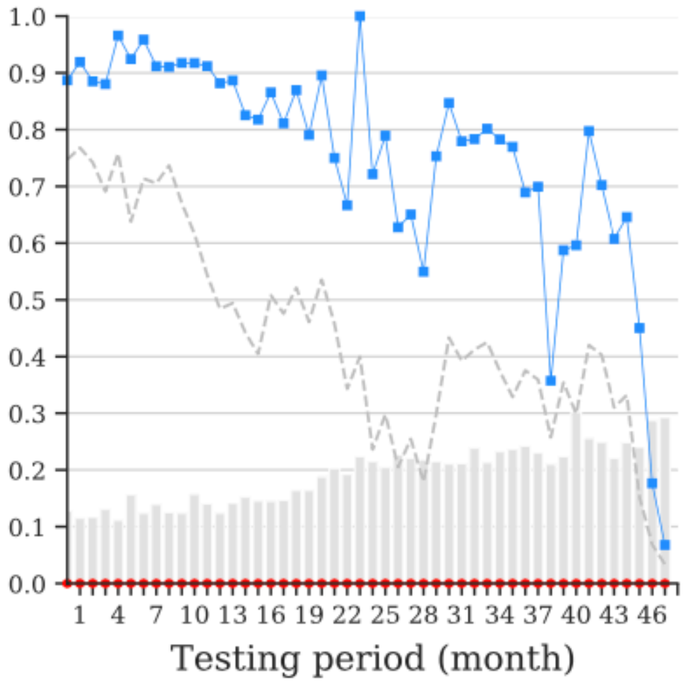
Results: Rejection Performance

Approx-TCE
(10 folds)



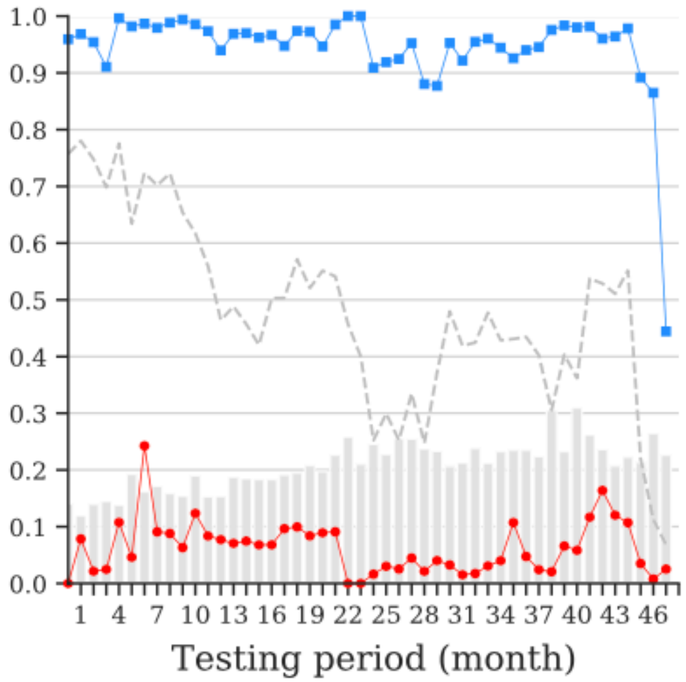

```
AUT(F1, kept):       0.82
AUT(F1, rejected):   0.00
CPU hours:           46.1
```

ICE
(0.33 calibration split)




```
AUT(f1, kept):       0.76
AUT(f1, rejected):   0.00
CPU hours:           11.5
```

CCE
(10 folds)



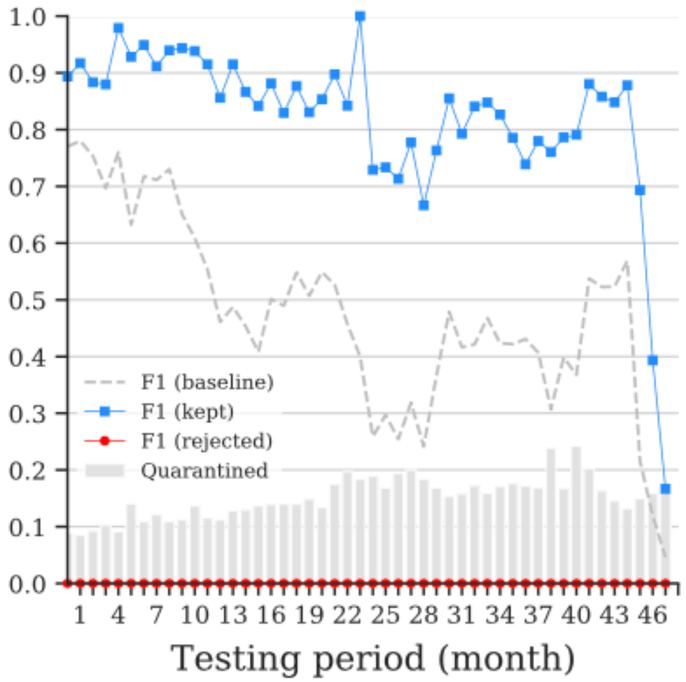

```
AUT(f1, kept):       0.94
AUT(f1, rejected):   0.06
CPU hours:           35.6
```

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

Results: Rejection Performance

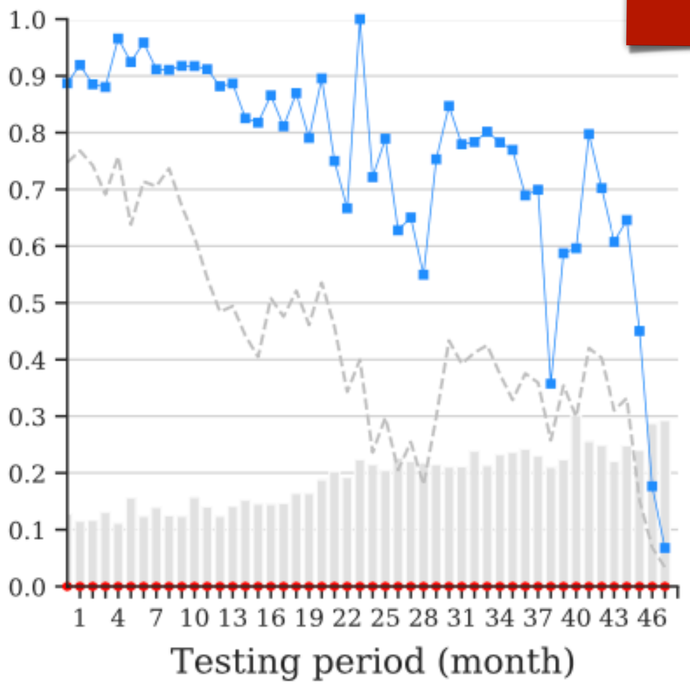
More experiments on different datasets, SOTA, and optimization in the paper

Approx-TCE
(10 folds)

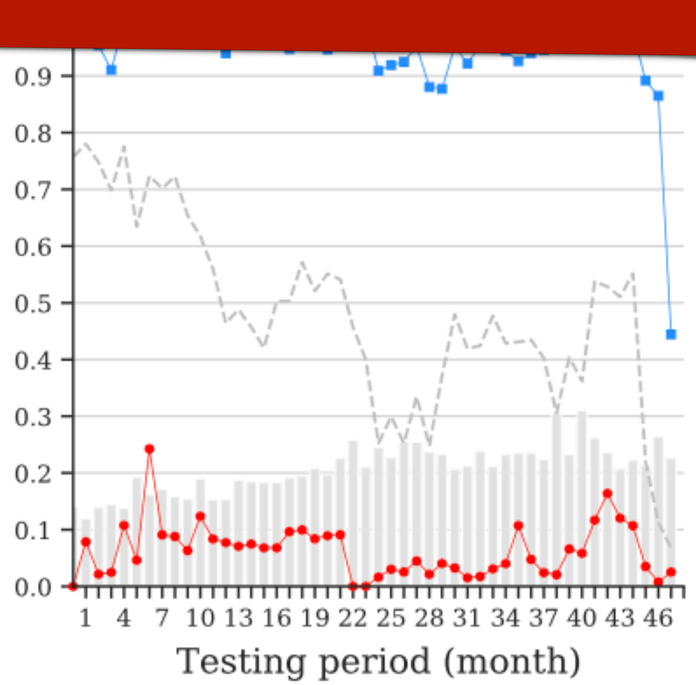



```
AUT(F1, kept):       0.82
AUT(F1, rejected):   0.00
CPU hours:           46.1
```

ICE
(0.33 calibration split)




```
AUT(f1, kept):       0.76
AUT(f1, rejected):   0.00
CPU hours:           11.5
```




```
AUT(f1, kept):       0.94
AUT(f1, rejected):   0.06
CPU hours:           35.6
```

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

Reproducibility Crisis

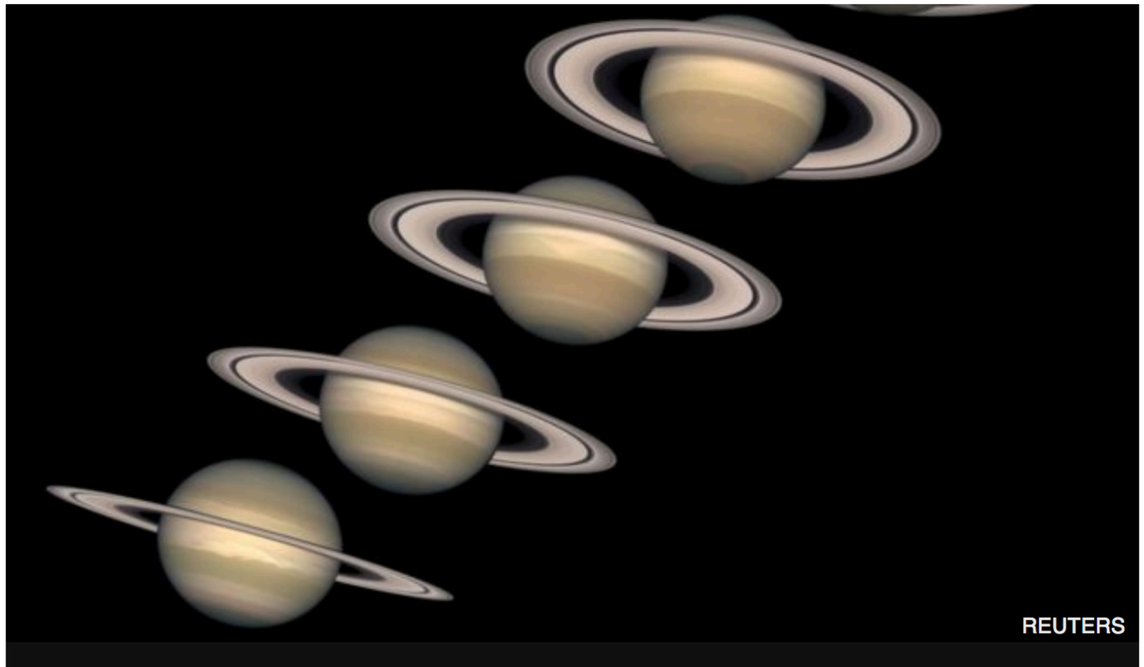
February 2019

AAAS: Machine learning 'causing science crisis'

By Pallab Ghosh
Science correspondent, BBC News, Washington

16 February 2019

AAAS meeting



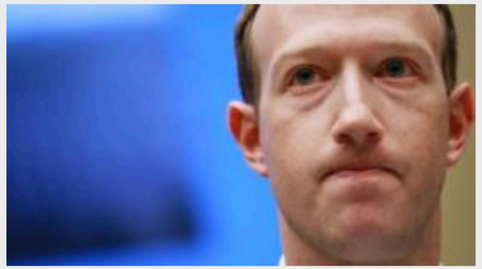
Top Stories

MPs demand 'urgent' Facebook regulation
The House of Commons publishes its report into fake news with some strong criticism of Facebook.
7 hours ago

UK cyber-security: Huawei risk manageable
58 minutes ago

Several Labour MPs 'set to quit party'
2 hours ago

Features

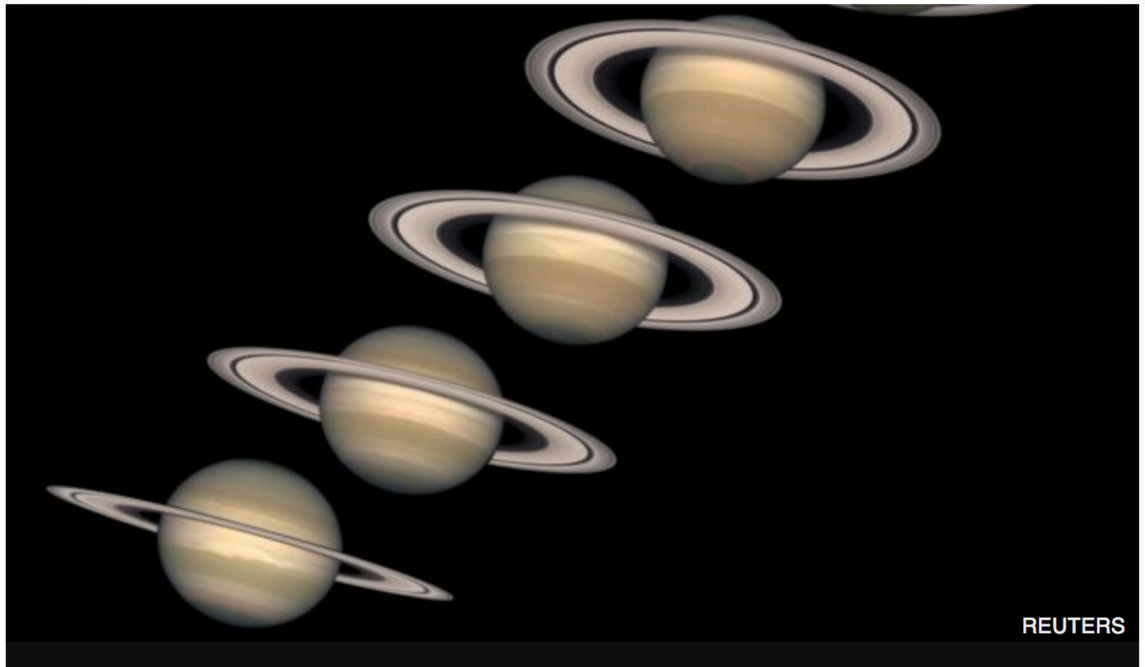


In a hurry? Here's what you need

February 2019

AAAS: Machine learning 'causing science crisis'

Machine learning techniques used by thousands of scientists to analyse data are producing results that are misleading and often completely wrong



Top Stories

MPs demand 'urgent' Facebook regulation

Commons publishes its
e news with some strong
acebook.

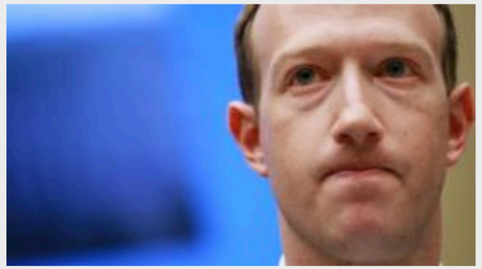
ecurity: Huawei risk
e

58 minutes ago

Several Labour MPs 'set to quit party'

2 hours ago

Features



In a hurry? Here's what you need

Leakage and the Reproducibility Crisis in ML-based Science

We argue that there is a reproducibility crisis in ML-based science. We compile evidence of this crisis across fields, identify data leakage as a pervasive cause of reproducibility failures, conduct our own reproducibility investigations using in-depth code-review, and propose a solution.

PRINCETON UNIVERSITY

Draft paper

July '22 online workshop

Context

Many quantitative science fields are adopting the paradigm of predictive modeling using machine learning. We welcome this development. At the same time, as researchers whose interests include the strengths and limits of machine learning, we have concerns about reproducibility and overoptimism.

There are many reasons for caution:

- Performance evaluation is notoriously tricky in machine learning.
- ML code tends to be complex and as yet lacks standardization.
- Subtle pitfalls arise from the differences between explanatory and predictive modeling.
- The hype and overoptimism about commercial AI may spill over into ML-based scientific research.
- Pressures and publication biases that have led to past reproducibility crises are also present in ML-based science.

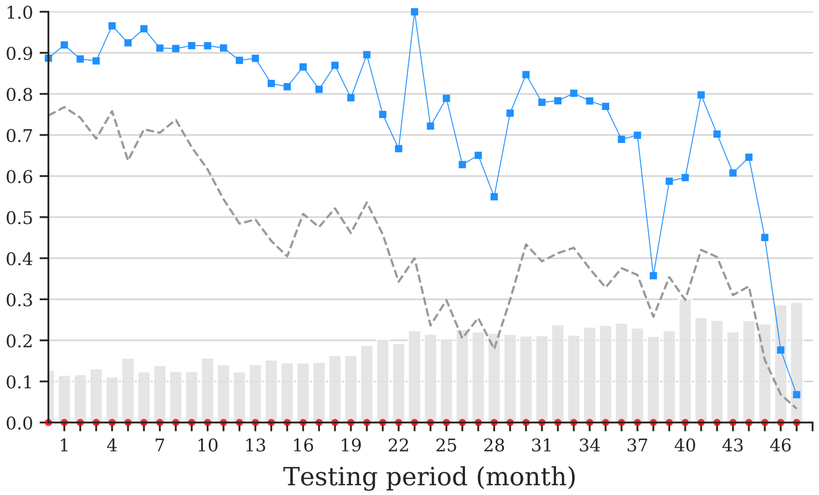
Given these reasons, we view reproducibility difficulties as the expected state of affairs until best practices become better established and understood. The spate of reproducibility failures (that we have compiled below) highlights the immaturity of ML-based science, the critical need for ongoing work on methods and best practices, and the importance of treating the results from this body of work with caution.

Unintentional Data Snooping

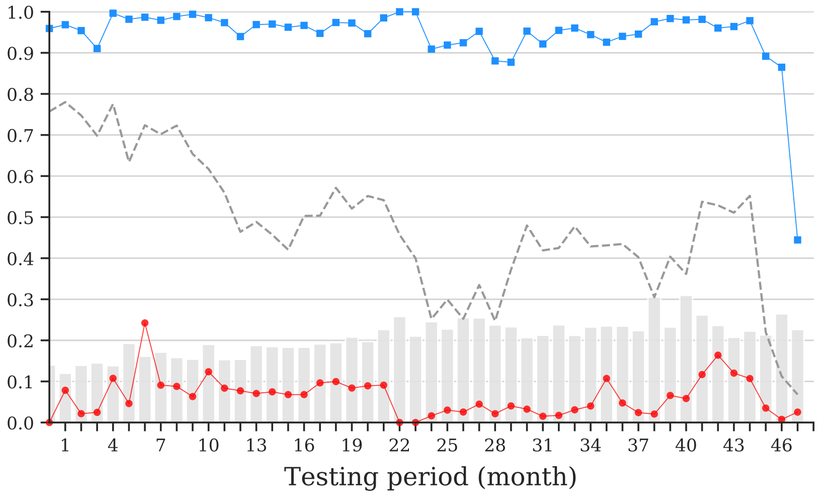
[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift https://s2lab.cs.ucl.ac.uk/projects/transcend/

Conformal Evaluation

ICE
(0.33 calibration split)



CCE
(10 folds)



Unintentional Data Snooping

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift https://s2lab.cs.ucl.ac.uk/projects/transcend/

Conformal Evaluation

BUGFIX: fixed data snooping using ground truth labels when computing ...

...SVM scores for pvals

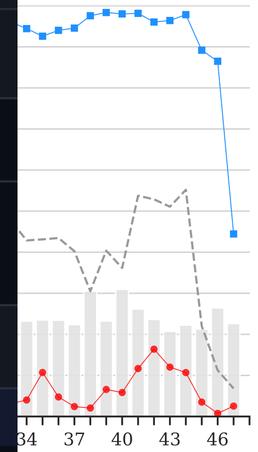
main

lcavallaro committed on Dec 8, 2022

Showing **1 changed file** with **1 addition** and **1 deletion**.

2 ice.py

```
@@ -282,7 +282,7 @@ def main():
282 282                         args.pval_consider))
283 283
284 284             logging.info('Getting NCMs for test ({})...'.format(argstest))
285     -           ncms_full_test = scores.get_svm_ncms(svm, X_test, y_test)
    285 +           ncms_full_test = scores.get_svm_ncms(svm, X_test, pred_test)
286 286
287 287             p_val_test_dict = scores.compute_p_values_cred_and_conf(
288 288                 train_ncms=ncms,
```



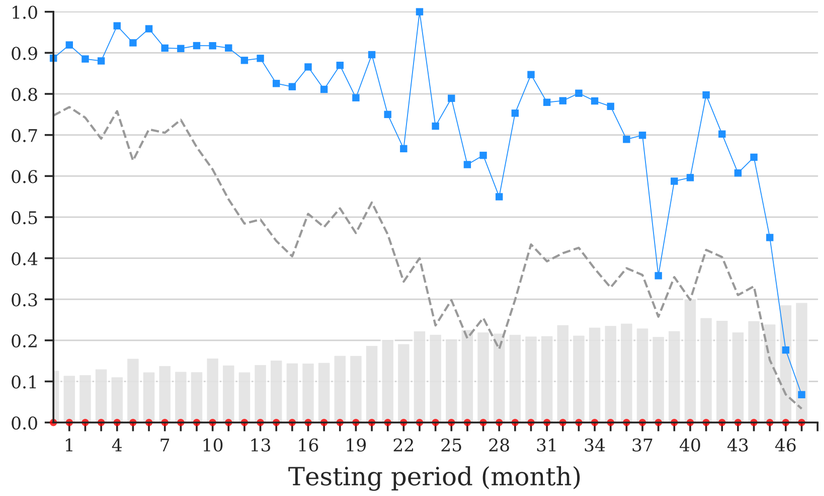
Unintentional Data Snooping

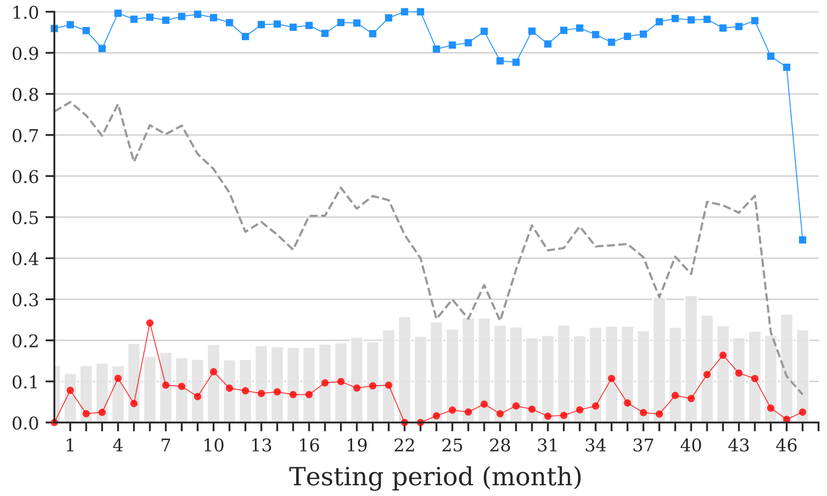
[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift https://s2lab.cs.ucl.ac.uk/projects/transcend/

ICE
(0.33 calibration split)

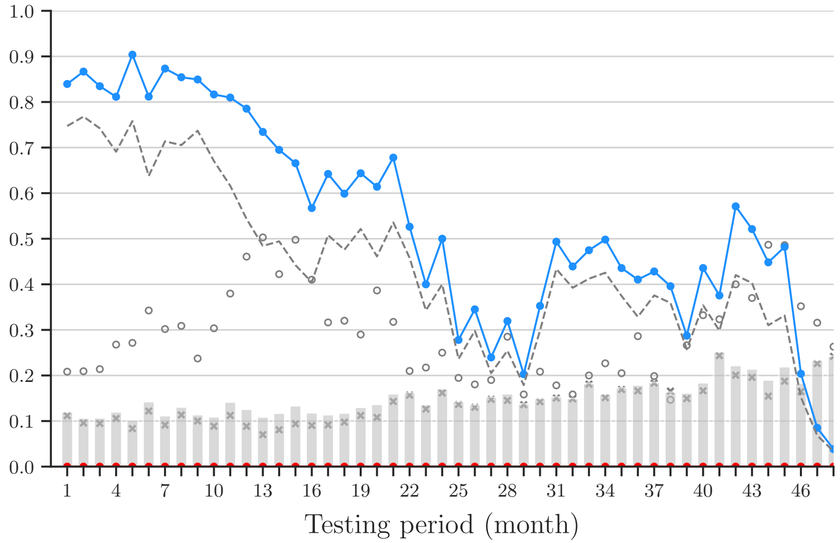
CCE
(10 folds)

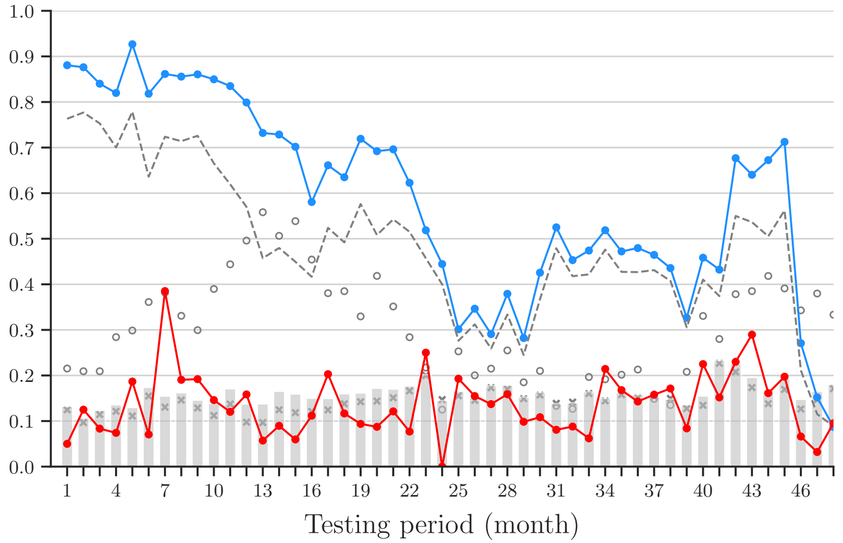
Conformal Evaluation





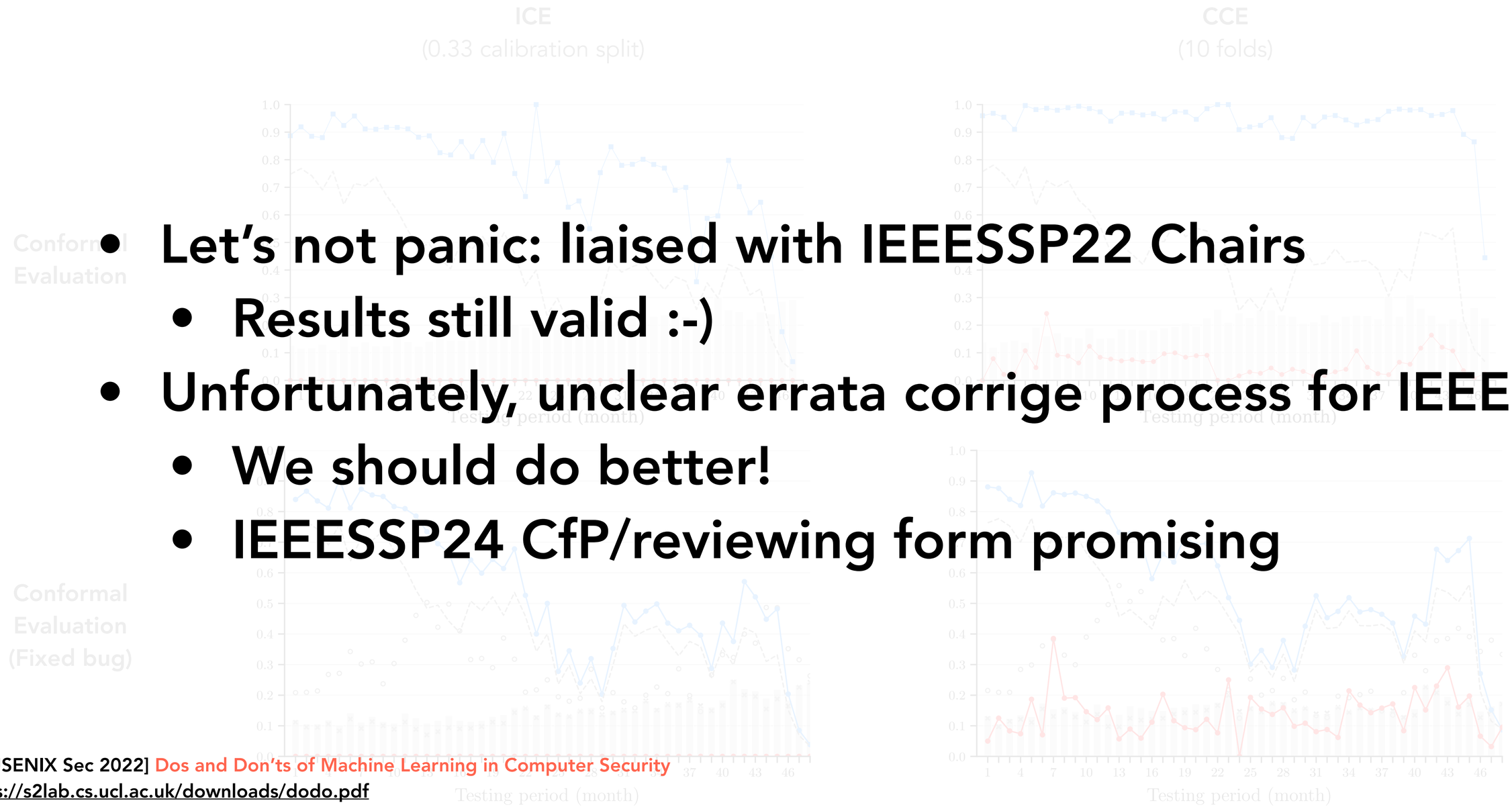
Conformal Evaluation (Fixed bug)





Unintentional Data Snooping(*)

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift https://s2lab.cs.ucl.ac.uk/projects/transcend/



- Let's not panic: liaised with IEEESSP22 Chairs
  - Results still valid :-)
- Unfortunately, unclear errata corrige process for IEEE
  - We should do better!
  - IEEESSP24 CfP/reviewing form promising

(*) [USENIX Sec 2022] Dos and Don'ts of Machine Learning in Computer Security https://s2lab.cs.ucl.ac.uk/downloads/dodo.pdf

Unintentional Data Snooping(*)

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift https://s2lab.cs.ucl.ac.uk/projects/transcend/

IEEE S&P 2024 Review Form (Excerpt)

Scientific Contribution

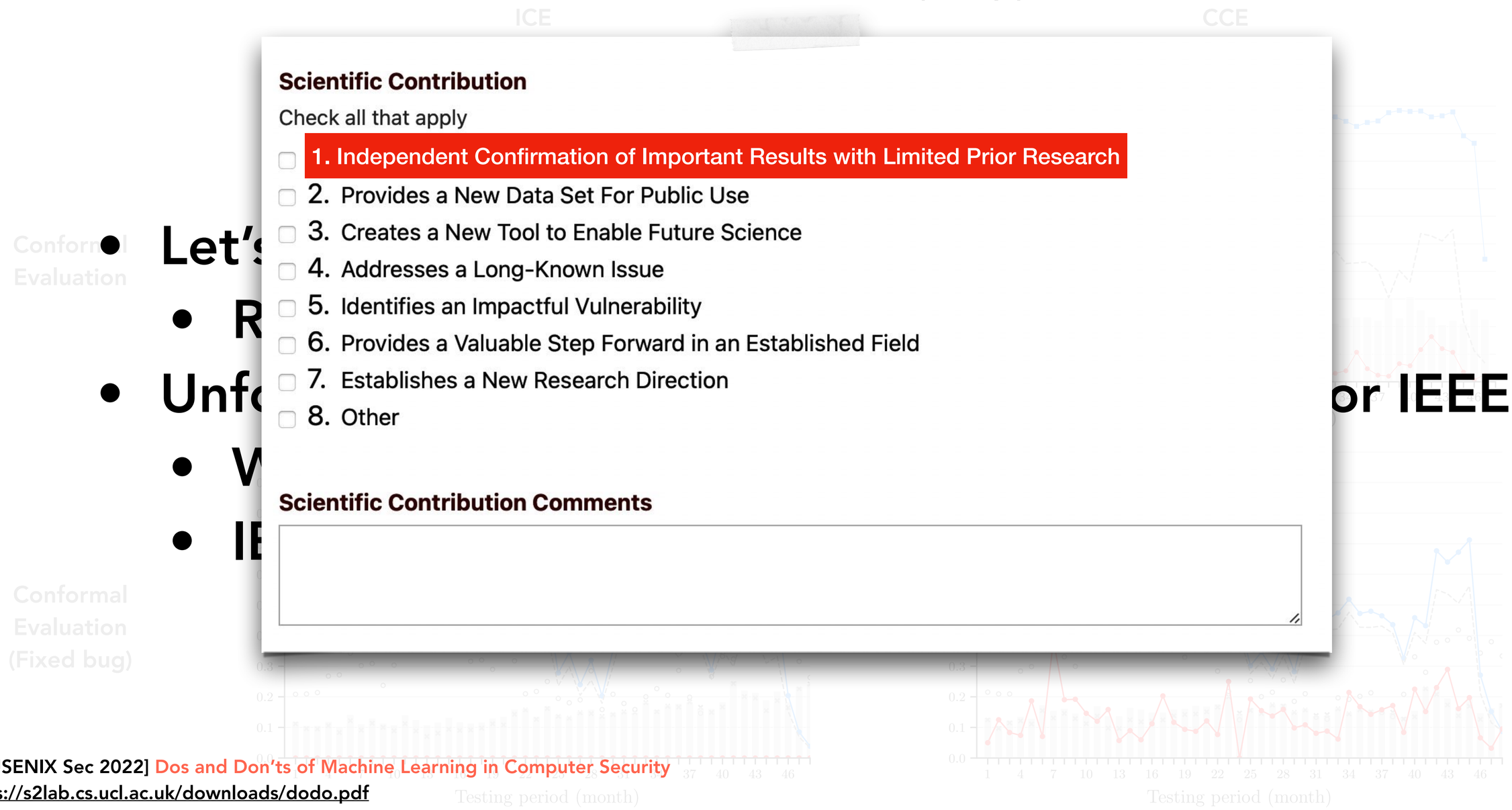
Check all that apply

- [ ] 1. Independent Confirmation of Important Results with Limited Prior Research
- [ ] 2. Provides a New Data Set For Public Use
- [ ] 3. Creates a New Tool to Enable Future Science
- [ ] 4. Addresses a Long-Known Issue
- [ ] 5. Identifies an Impactful Vulnerability
- [ ] 6. Provides a Valuable Step Forward in an Established Field
- [ ] 7. Establishes a New Research Direction
- [ ] 8. Other

Scientific Contribution Comments

- Let's
  - R
- Unfo
  - W
  - IE

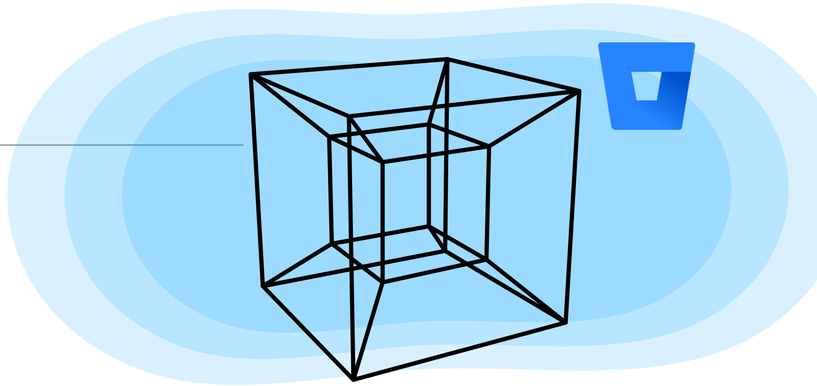
or IEEE



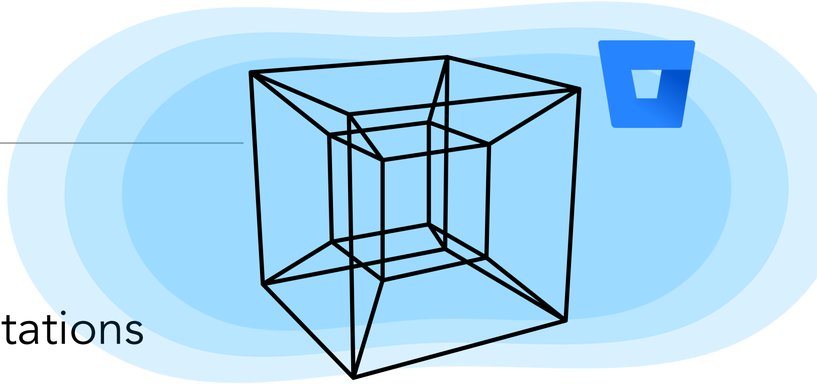
(*) [USENIX Sec 2022] Dos and Don'ts of Machine Learning in Computer Security https://s2lab.cs.ucl.ac.uk/downloads/dodo.pdf

Take Aways

Take Aways



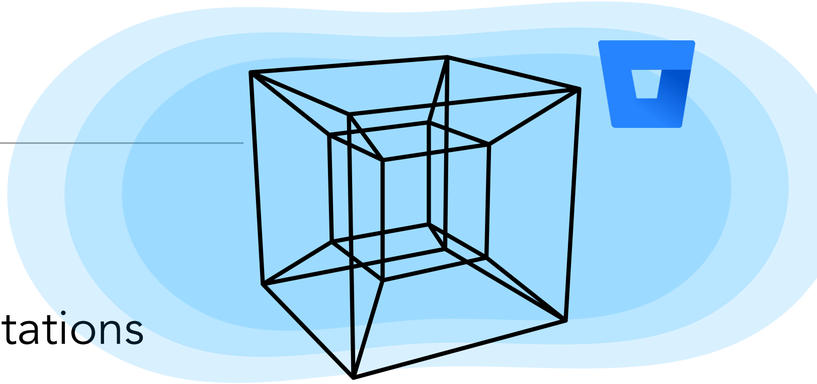
Take Aways



- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations [1], and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning** [1], **online learning** [6]

[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malware classification across space and time**, USENIX Security 2019
[2] Pierazzi et al., **Intriguing Properties of Adversarial ML attacks in the problem psace**, IEEE S&P 2020
[3] Jordaney et al., **Transcend: Detecting concept drift in malware classification models**, USENIX Security 2017
[4] Barbero et al., **Transcending Transcend: Revisiting malware classification in the presence of concept drift**, IEEE S&P 2022
[5] Arp et al., **Dos and Dont's of Machine Learning in Security**, USENIX Security 2022
[6] Kan et al. **Investigating Labelless Drift Adaptation for Malware Detection**, AISec 2021
[7] Yang et al. **Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers**. IEEE S&P 2023

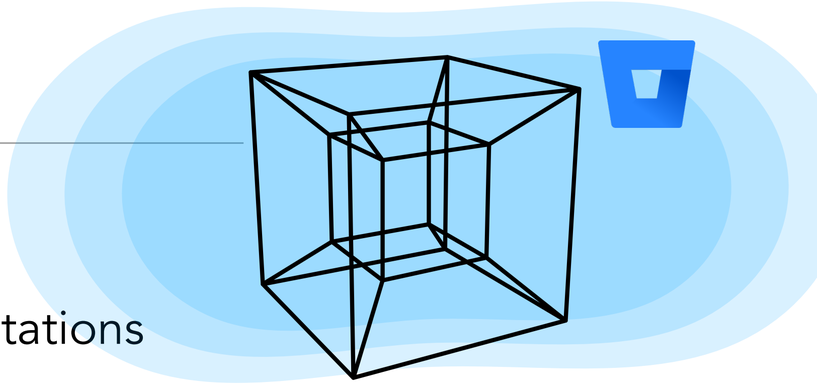
Take Aways



- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations [1], and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning** [1], **online learning** [6]
- Reason about problem space **(relizable) adversarial attacks and defenses** [2, 7]

[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malware classification across space and time**, USENIX Security 2019
[2] Pierazzi et al., **Intriguing Properties of Adversarial ML attacks in the problem psace**, IEEE S&P 2020
[3] Jordaney et al., **Transcend: Detecting concept drift in malware classification models**, USENIX Security 2017
[4] Barbero et al., **Transcending Transcend: Revisiting malware classification in the presence of concept drift**, IEEE S&P 2022
[5] Arp et al., **Dos and Dont's of Machine Learning in Security**, USENIX Security 2022
[6] Kan et al. **Investigating Labelless Drift Adaptation for Malware Detection**, AISec 2021
[7] Yang et al. **Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers**. IEEE S&P 2023

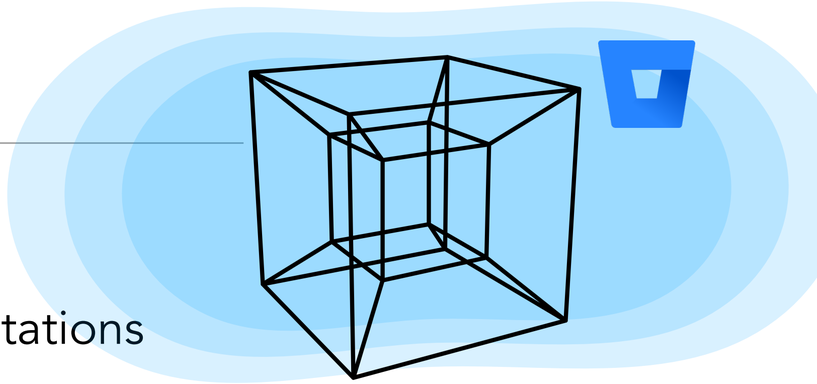
Take Aways



- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations [1], and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning** [1], **online learning** [6]
- Reason about problem space **(relizable) adversarial attacks and defenses** [2, 7]
- Reason about the **relationship** between **adversarial ML** and **dataset shifts**

[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malware classification across space and time**, USENIX Security 2019
[2] Pierazzi et al., **Intriguing Properties of Adversarial ML attacks in the problem psace**, IEEE S&P 2020
[3] Jordaney et al., **Transcend: Detecting concept drift in malware classification models**, USENIX Security 2017
[4] Barbero et al., **Transcending Transcend: Revisiting malware classification in the presence of concept drift**, IEEE S&P 2022
[5] Arp et al., **Dos and Dont's of Machine Learning in Security**, USENIX Security 2022
[6] Kan et al. **Investigating Labelless Drift Adaptation for Malware Detection**, AISec 2021
[7] Yang et al. **Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers**. IEEE S&P 2023

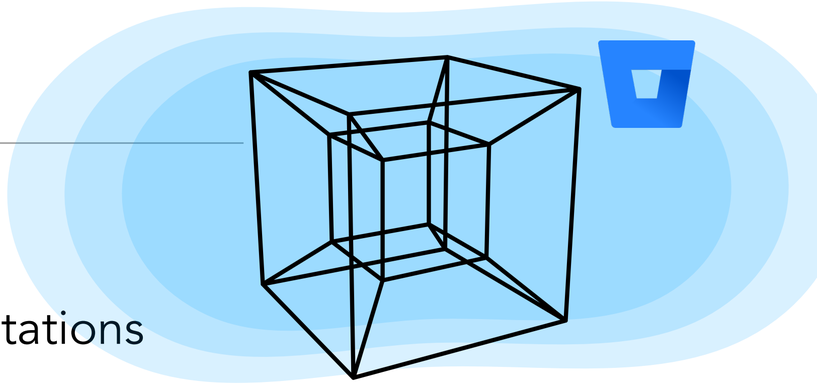
Take Aways



- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations [1], and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning** [1], **online learning** [6]
- Reason about problem space **(relizable) adversarial attacks and defenses** [2, 7]
- Reason about the **relationship** between **adversarial ML** and **dataset shifts**
- Reason about **abstractions** and **representations** and their effect on the entire ML pipeline

[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malware classification across space and time**, USENIX Security 2019
[2] Pierazzi et al., **Intriguing Properties of Adversarial ML attacks in the problem psace**, IEEE S&P 2020
[3] Jordaney et al., **Transcend: Detecting concept drift in malware classification models**, USENIX Security 2017
[4] Barbero et al., **Transcending Transcend: Revisiting malware classification in the presence of concept drift**, IEEE S&P 2022
[5] Arp et al., **Dos and Dont's of Machine Learning in Security**, USENIX Security 2022
[6] Kan et al. **Investigating Labelless Drift Adaptation for Malware Detection**, AISec 2021
[7] Yang et al. **Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers**. IEEE S&P 2023

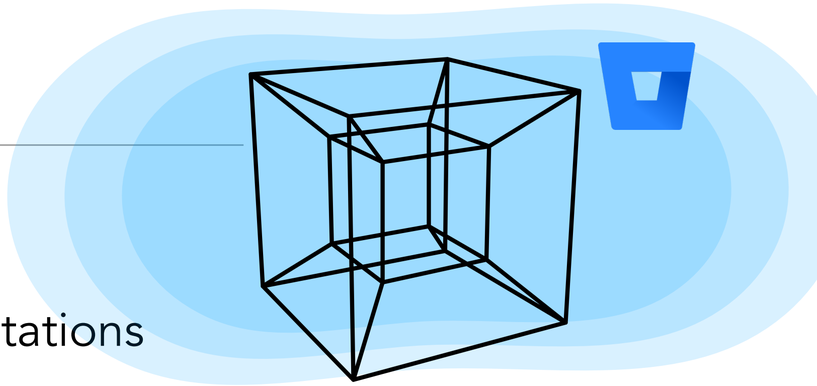
Take Aways



- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations [1], and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning** [1], **online learning** [6]
- Reason about problem space **(relizable) adversarial attacks and defenses** [2, 7]
- Reason about the **relationship** between **adversarial ML** and **dataset shifts**
- Reason about **abstractions** and **representations** and their effect on the entire ML pipeline
- Bridging the academia-industry gap
  - See https://s2lab.cs.ucl.ac.uk for access

[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malware classification across space and time**, USENIX Security 2019
[2] Pierazzi et al., **Intriguing Properties of Adversarial ML attacks in the problem psace**, IEEE S&P 2020
[3] Jordaney et al., **Transcend: Detecting concept drift in malware classification models**, USENIX Security 2017
[4] Barbero et al., **Transcending Transcend: Revisiting malware classification in the presence of concept drift**, IEEE S&P 2022
[5] Arp et al., **Dos and Dont's of Machine Learning in Security**, USENIX Security 2022
[6] Kan et al. **Investigating Labelless Drift Adaptation for Malware Detection**, AISec 2021
[7] Yang et al. **Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers**. IEEE S&P 2023

Take Aways



- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations [1], and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning** [1], **online learning** [6]
- Reason about problem space **(relizable) adversarial attacks and defenses** [2, 7]
- Reason about the **relationship** between **adversarial ML** and **dataset shifts**
- Reason about **abstractions** and **representations** and their effect on the entire ML pipeline
- Bridging the academia-industry gap
  - See https://s2lab.cs.ucl.ac.uk for access

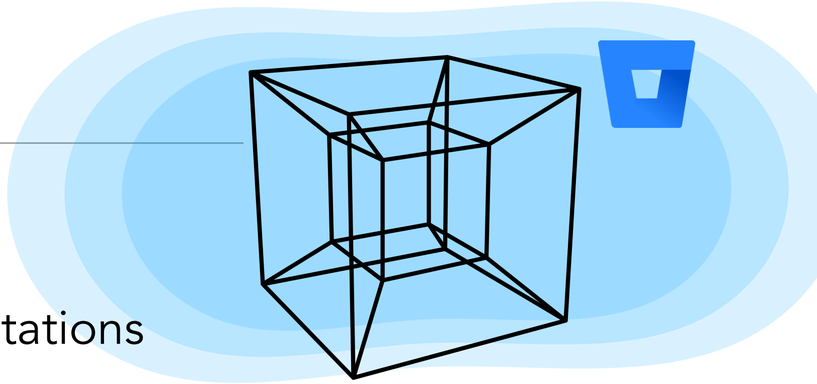
https://twitter.com/joshua_saxe/status/1550545466072264704

[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malw**
[2] Pierazzi et al., **Intriguing Properties of Adversarial ML attacks in the p**
[3] Jordaney et al., **Transcend: Detecting concept drift in malware classif**
[4] Barbero et al., **Transcending Transcend: Revisiting malware classificat**
[5] Arp et al., **Dos and Dont's of Machine Learning in Security**, USENIX S
[6] Kan et al. **Investigating Labelless Drift Adaptation for Malware Detection**, AISec 2021
[7] Yang et al. **Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers**. IEEE S&P 2023

Our Open-Source Libraries

- Requested access by 120+ organizations, including (honorable mentions):

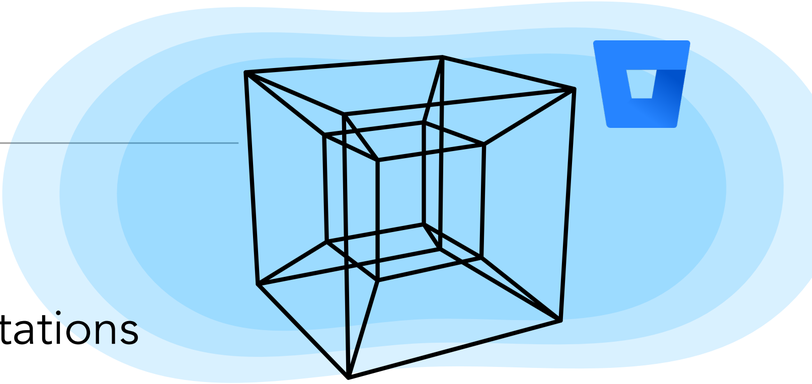
Take Aways



- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations [1], and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning** [1], **online learning** [6]
- Reason about problem space **(relizable) adversarial attacks and defenses** [2, 7]
- Reason about the **relationship** between **adversarial ML** and **dataset shifts**
- Reason about **abstractions** and **representations** and their effect on the entire ML pipeline
- Bridging the academia-industry gap
  - See https://s2lab.cs.ucl.ac.uk for access

[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malware classification across space and time**, USENIX Security 2019
[2] Pierazzi et al., **Intriguing Properties of Adversarial ML attacks in the problem psace**, IEEE S&P 2020
[3] Jordaney et al., **Transcend: Detecting concept drift in malware classification models**, USENIX Security 2017
[4] Barbero et al., **Transcending Transcend: Revisiting malware classification in the presence of concept drift**, IEEE S&P 2022
[5] Arp et al., **Dos and Dont's of Machine Learning in Security**, USENIX Security 2022
[6] Kan et al. **Investigating Labelless Drift Adaptation for Malware Detection**, AISec 2021
[7] Yang et al. **Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers**. IEEE S&P 2023

Take Aways



- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations [1], and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning** [1], **online learning** [6]
- Reason about problem space **(relizable) adversarial attacks and defenses** [2, 7]
- Reason about the **relationship** between **adversarial ML** and ~~**dataset shifts**~~
- Reason about **abstractions** and **representations** and their effect on the entire ML pipeline
- Bridging the academia-industry gap
  - See https://s2lab.cs.ucl.ac.uk for access

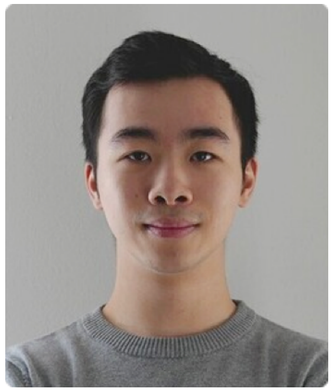
Exciting Recent Effort [8]

[8] Pei et al. **Symmetry-Preserving Program Representations for Learning Code Semantics** Collaboration with Columbia University — https://arxiv.org/abs/2308.03312

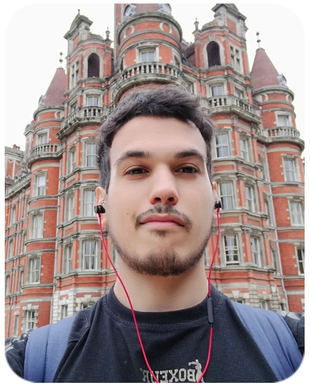
[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malware classification across space and time**, USENIX Security 2019
[2] Pierazzi et al., **Intriguing Properties of Adversarial ML attacks in the problem psace**, IEEE S&P 2020
[3] Jordaney et al., **Transcend: Detecting concept drift in malware classification models**, USENIX Security 2017
[4] Barbero et al., **Transcending Transcend: Revisiting malware classification in the presence of concept drift**, IEEE S&P 2022
[5] Arp et al., **Dos and Dont's of Machine Learning in Security**, USENIX Security 2022
[6] Kan et al. **Investigating Labelless Drift Adaptation for Malware Detection**, AISec 2021
[7] Yang et al. **Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers**. IEEE S&P 2023

Core Team

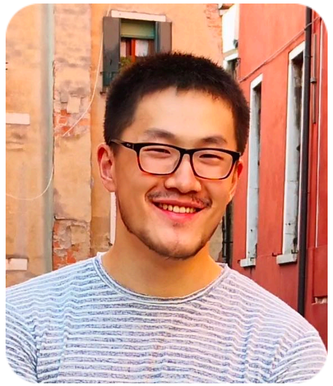
Ph.D. Students



Theo



Jacopo



Mark

Giulio

Ph.D. Alumni

Feargus

Team-ups

Fabio Pierazzi

Current Research Collaborators

Columbia University in the City of New York

University of Illinois at Urbana-Champaign

Imperial College London

Zhejiang University 1897

Berkeley University of California

avast

King's College London

Università degli Studi di Milano

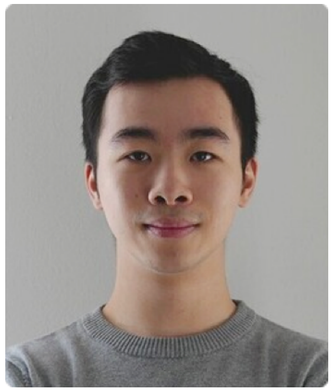
Universität der Bundeswehr München

University of Cambridge

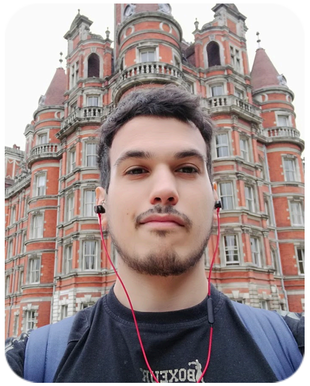
Technische Universität Braunschweig

Core Team

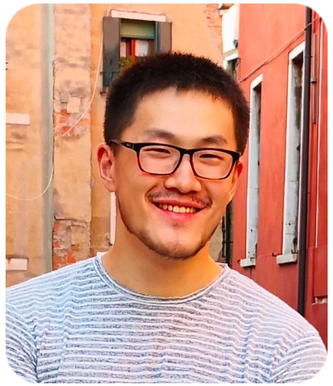
Ph.D. Students



Theo



Jacopo



Mark

Giulio

Ph.D. Alumni

Feargus

Team-ups

Fabio Pierazzi

Current Research Collaborators

Columbia University in the City of New York

University of Illinois at Urbana-Champaign

Imperial College London

Zhejiang University 1897

Berkeley University of California

avast

King's College London

Università degli Studi di Milano

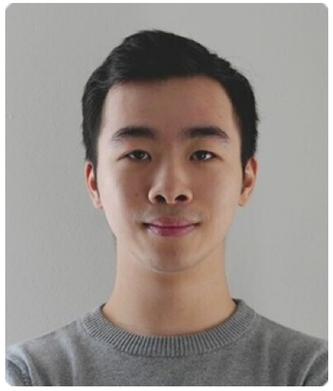
Universität der Bundeswehr München

University of Cambridge

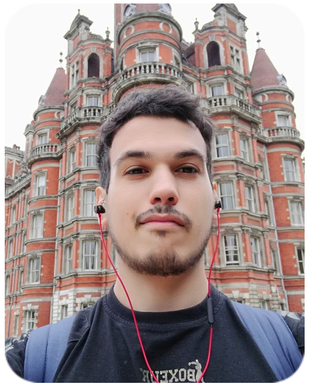
Technische Universität Braunschweig

Core Team

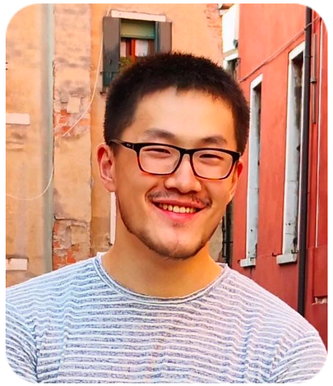
Ph.D. Students



Theo



Jacopo



Mark

Giulio

Ph.D. Alumni

Feargus

Team-ups

Fabio Pierazzi

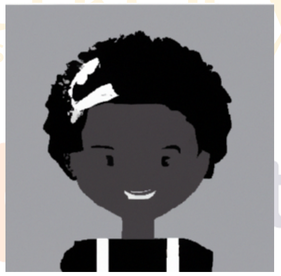
KING'S College LONDON

Current Research Collaborators

I am hiring at UCL! :-)

PhD1

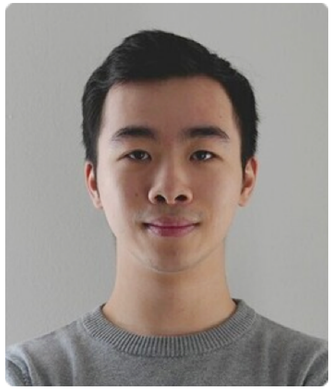
PhD2



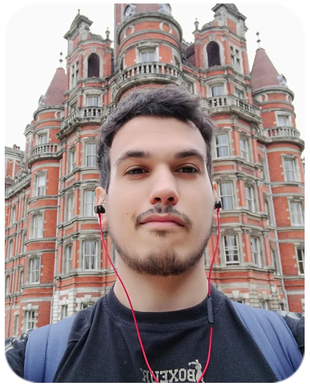
PostDoc

Core Team

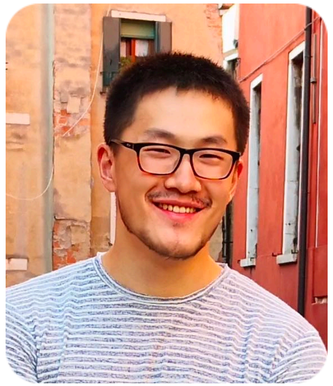
Ph.D. Students



Theo



Jacopo



Mark

Giulio

Ph.D. Alumni

Feargus

Team-ups

Fabio Pierazzi

I am hiring at UCL! :-)

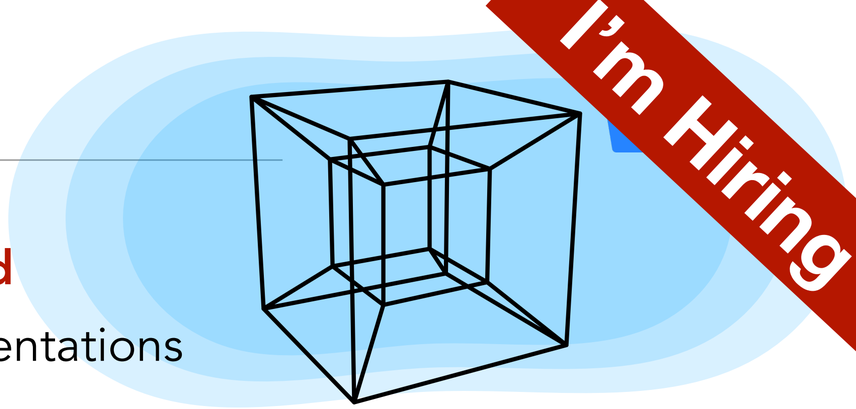
Current Research Collaborators

Xinran Zheng

Xingzhi Qiang

PostDoc

Take Aways



- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations [1], and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning** [1], **online learning** [6]
- Reason about problem space **(relizable) adversarial attacks and defenses** [2, 7]
- Reason about the **relationship** between **adversarial ML** and ~~**dataset shifts**~~
- Reason about **abstractions** and **representations** and their effect on the entire ML pipeline
- Bridging the academia-industry gap
  - See https://s2lab.cs.ucl.ac.uk for access

Exciting Recent Effort [8]

[8] Pei et al. **Symmetry-Preserving Program Representations for Learning Code Semantics** Collaboration with Columbia University — https://arxiv.org/abs/2308.03312

[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malware classification across space and time**, USENIX Security 2019
[2] Pierazzi et al., **Intriguing Properties of Adversarial ML attacks in the problem psace**, IEEE S&P 2020
[3] Jordaney et al., **Transcend: Detecting concept drift in malware classification models**, USENIX Security 2017
[4] Barbero et al., **Transcending Transcend: Revisiting malware classification in the presence of concept drift**, IEEE S&P 2022
[5] Arp et al., **Dos and Dont's of Machine Learning in Security**, USENIX Security 2022
[6] Kan et al. **Investigating Labelless Drift Adaptation for Malware Detection**, AISec 2021
[7] Yang et al. **Jigsaw Puzzle: Selective Backdoor Attack to Subvert Malware Classifiers**. IEEE S&P 2023

Backup Slides

Sources of Experimental Bias (1/3)

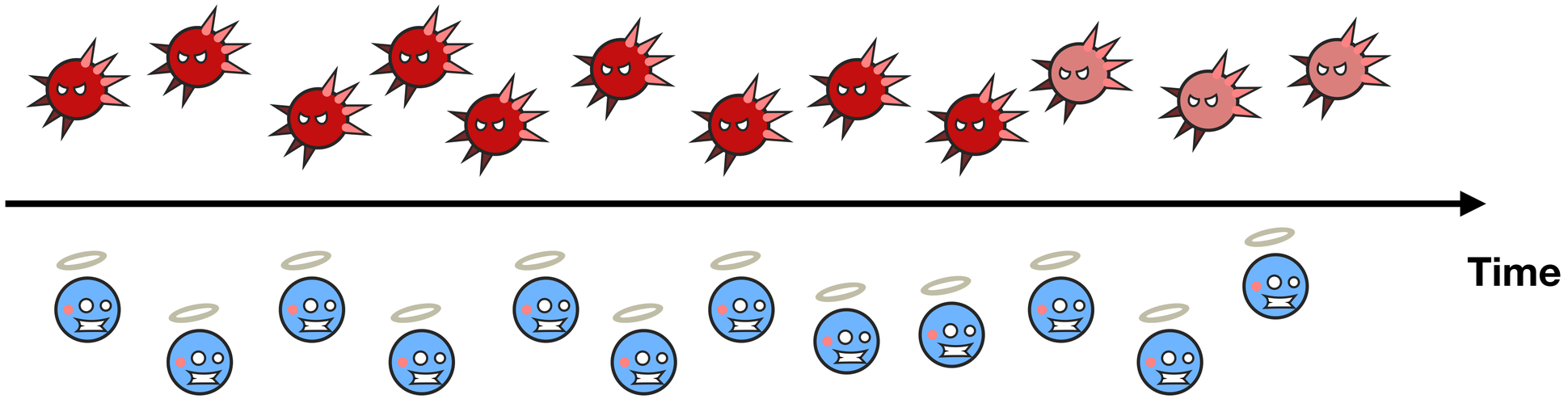
Temporal Inconsistency in Train/Test Sets

Sources of Experimental Bias (1/3)

Temporal Inconsistency in Train/Test Sets

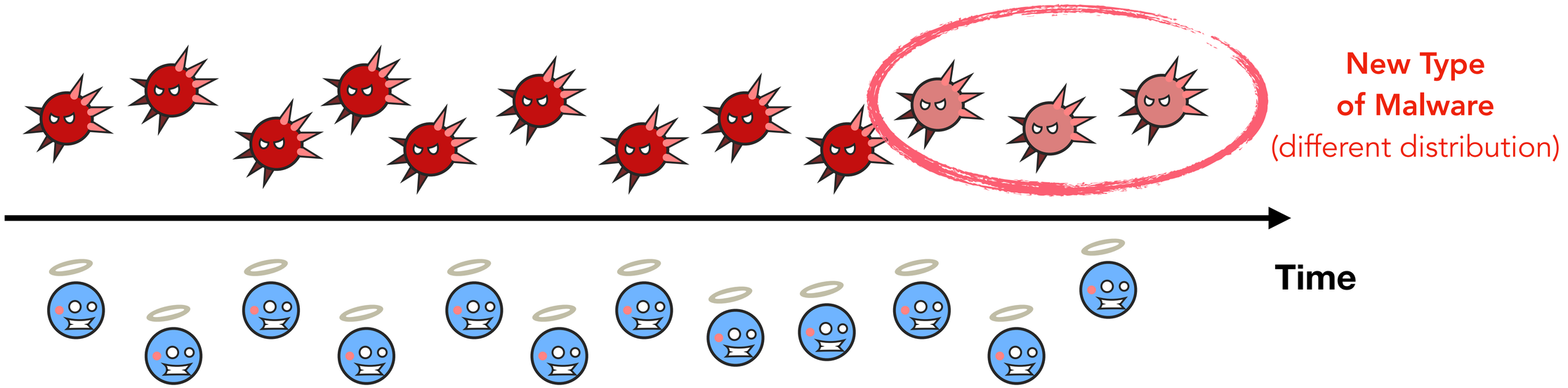
Sources of Experimental Bias (1/3)

Temporal Inconsistency in Train/Test Sets



Sources of Experimental Bias (1/3)

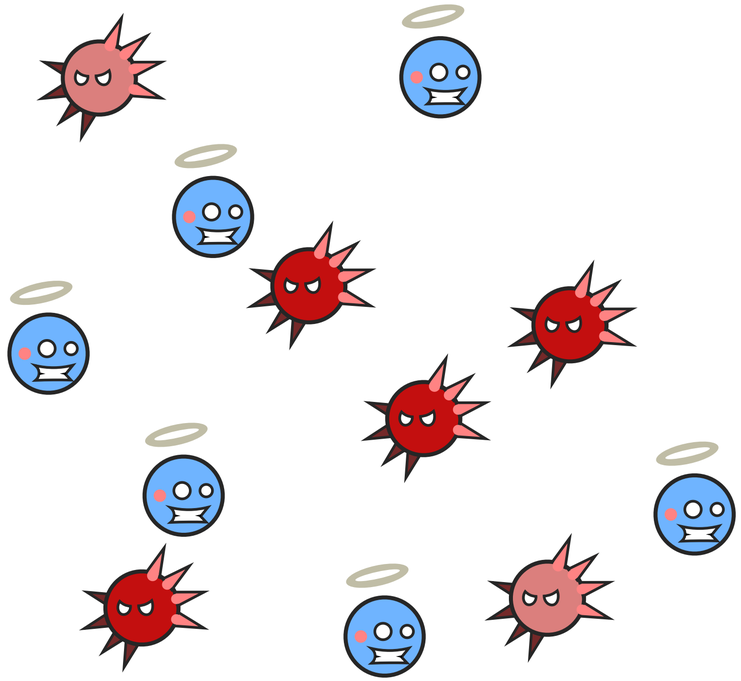
Temporal Inconsistency in Train/Test Sets



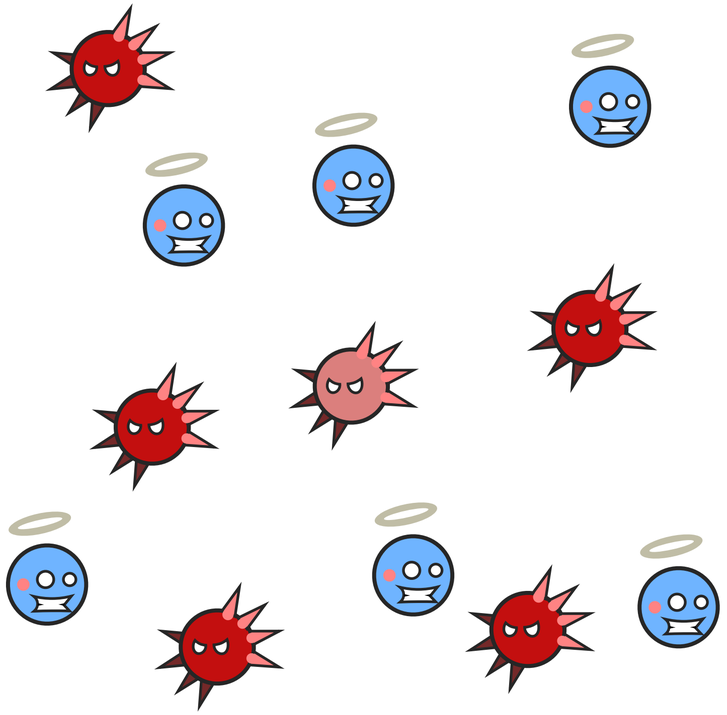
Sources of Experimental Bias (1/3)

Temporal Inconsistency in Train/Test Sets

Training



Testing

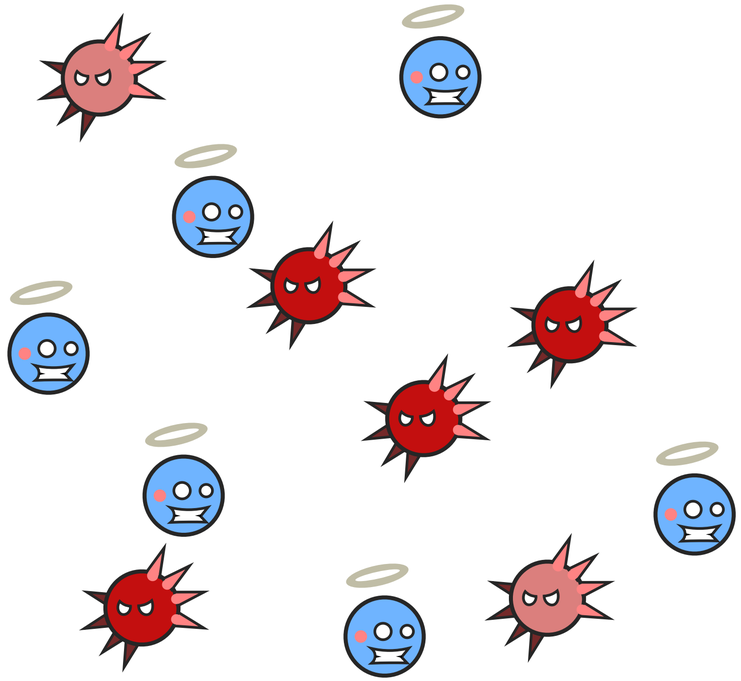


Sources of Experimental Bias (1/3)

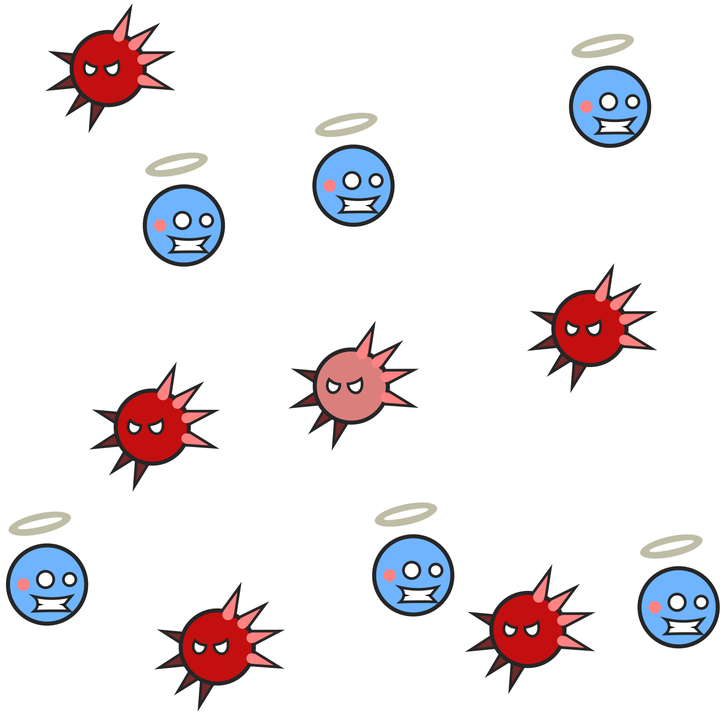
Temporal Inconsistency in Train/Test Sets

Violations use future knowledge in training

Training



Testing

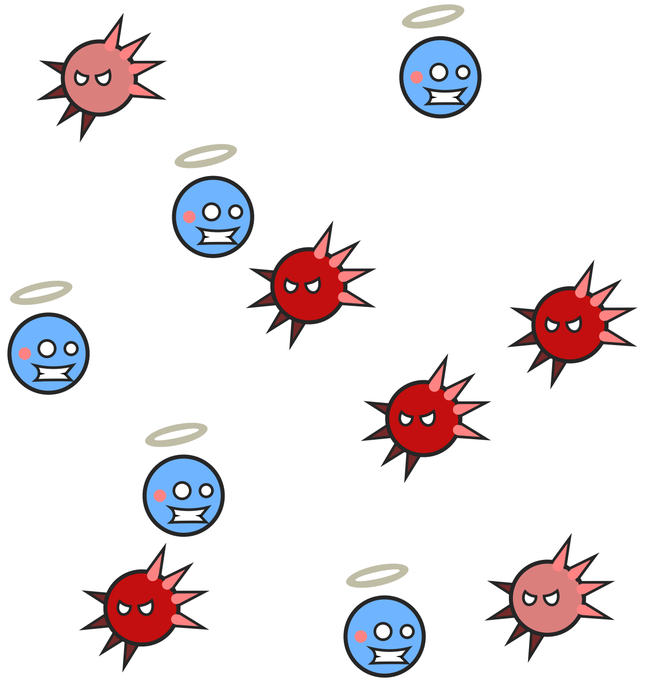


Sources of Experimental Bias (1/3)

Temporal Inconsistency in Train/Test Sets

Violations use future knowledge in training

Training



Kevin Allix et al. [ESSoS 2016]

Are Your Training Datasets Yet Relevant?
An Investigation into the Importance of Timeline in Machine Learning-based Malware Detection

Kevin Allix, Tegawendé F. Bissyandé, Jacques Klein, and Yves Le Traon

SnT - University of Luxembourg

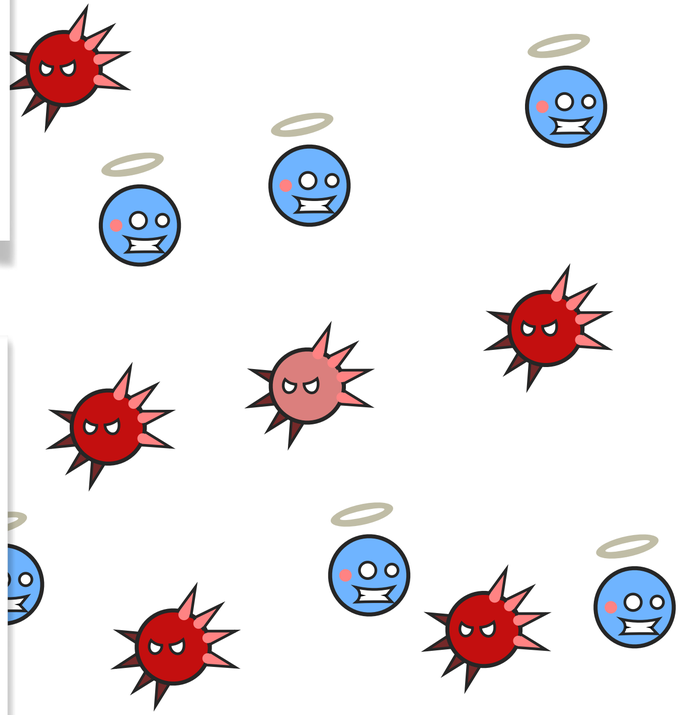
Brad Miller et al. [DIMVA 2016]

Reviewer Integration and Performance Measurement for Malware Detection

Brad Miller[1†], Alex Kantchelian[2], Michael Carl Tschantz[3], Sadia Afroz[3], Rekha Bachwani[4‡], Riyaz Faizullabhoy[2], Ling Huang[5], Vaishaal Shankar[2], Tony Wu[2], George Yiu[6†], Anthony D. Joseph[2], and J. D. Tygar[2]

[1] Google Inc. bradmiller@google.com
[2] UC Berkeley {akant,riyazdf,vaishaal,tony.wu,adj,tygar}@cs.berkeley.edu
[3] International Computer Science Institute {mct,sadia}@icsi.berkeley.edu
[4] Netflix rbachwani@netflix.com
[5] DataVisor ling.huang@datavisor.com
[6] Pinterest george@pinterest.com

Testing

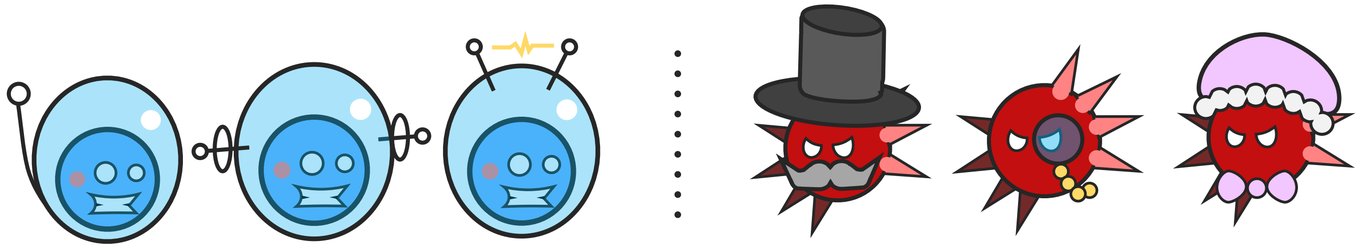


Sources of Experimental Bias (2/3)

Temporal {good|mal}ware inconsistency

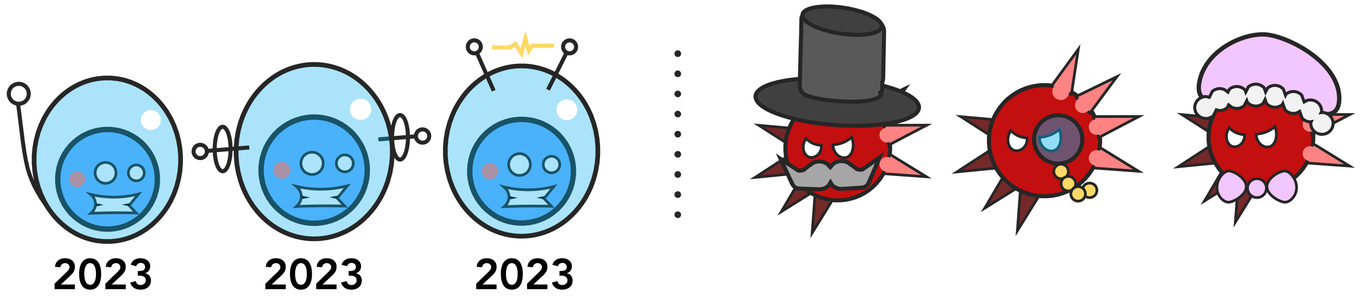
Sources of Experimental Bias (2/3)

Temporal {good|mal}ware inconsistency



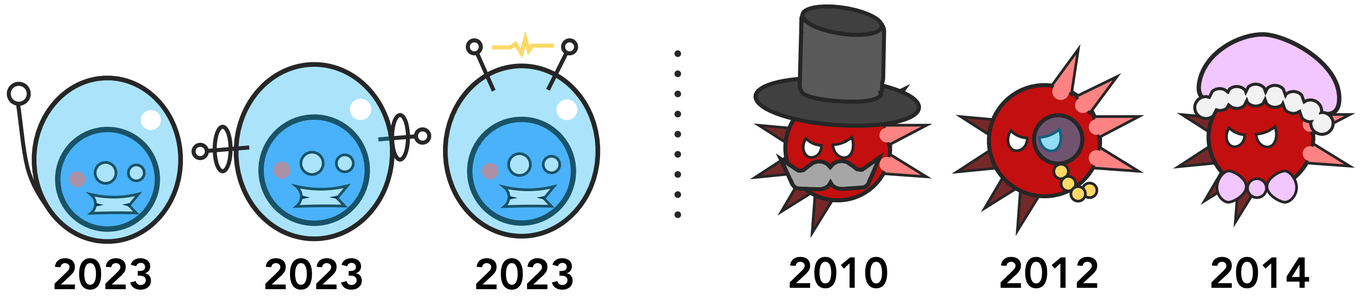
Sources of Experimental Bias (2/3)

Temporal {good|mal}ware inconsistency



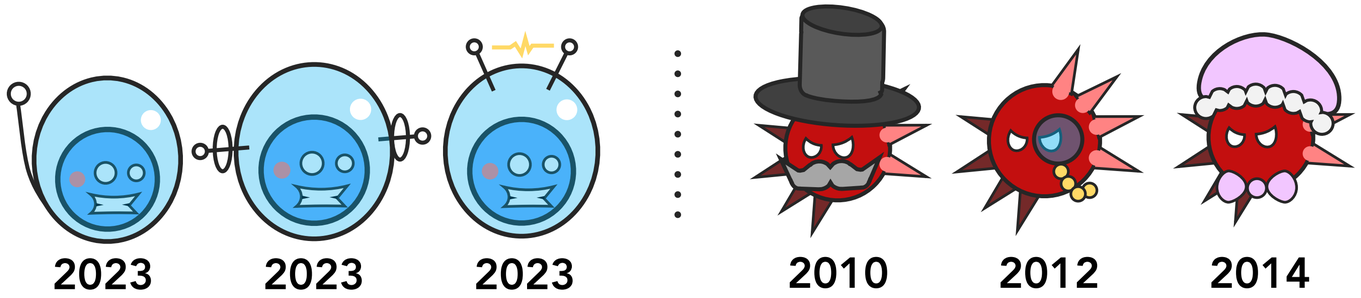
Sources of Experimental Bias (2/3)

Temporal {good|mal}ware inconsistency



Sources of Experimental Bias (2/3)

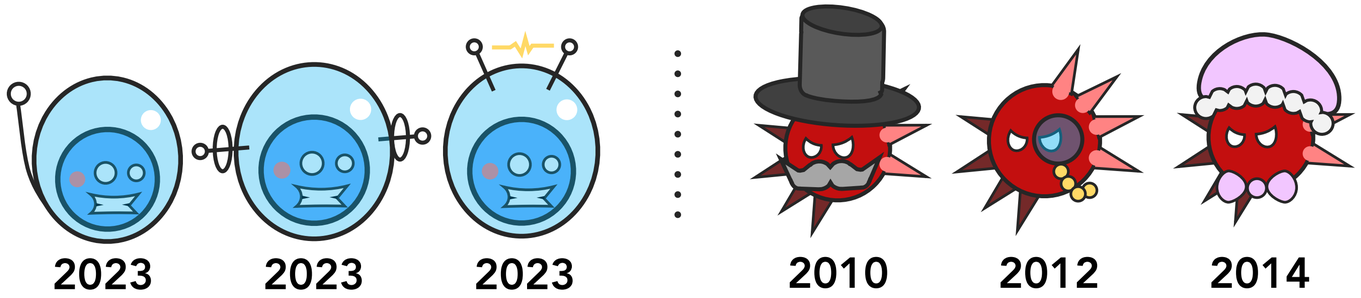
Temporal {good|mal}ware inconsistency




```
new_method()
```

Sources of Experimental Bias (2/3)

Temporal {good|mal}ware inconsistency

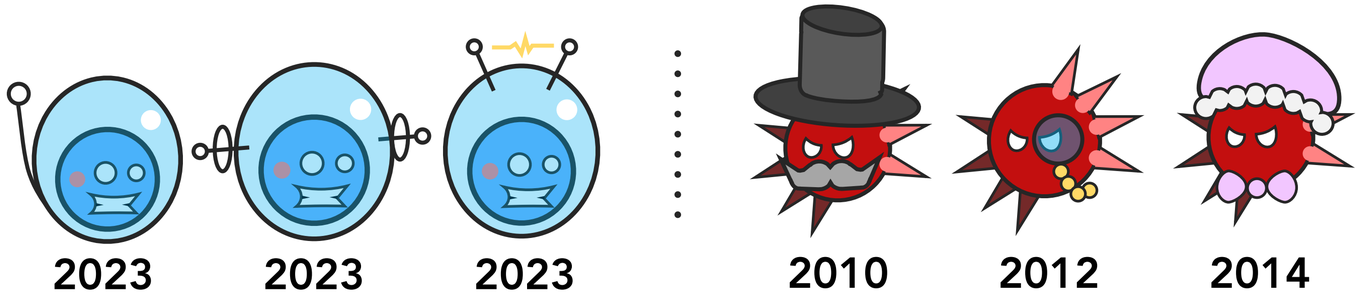


`new_method()`

Sources of Experimental Bias (2/3)

Temporal {good|mal}ware inconsistency

Violations may learn artifacts



`new_method()`

Sources of Experimental Bias (3/3)

Unrealistic Test Class Ratio

Sources of Experimental Bias (3/3)

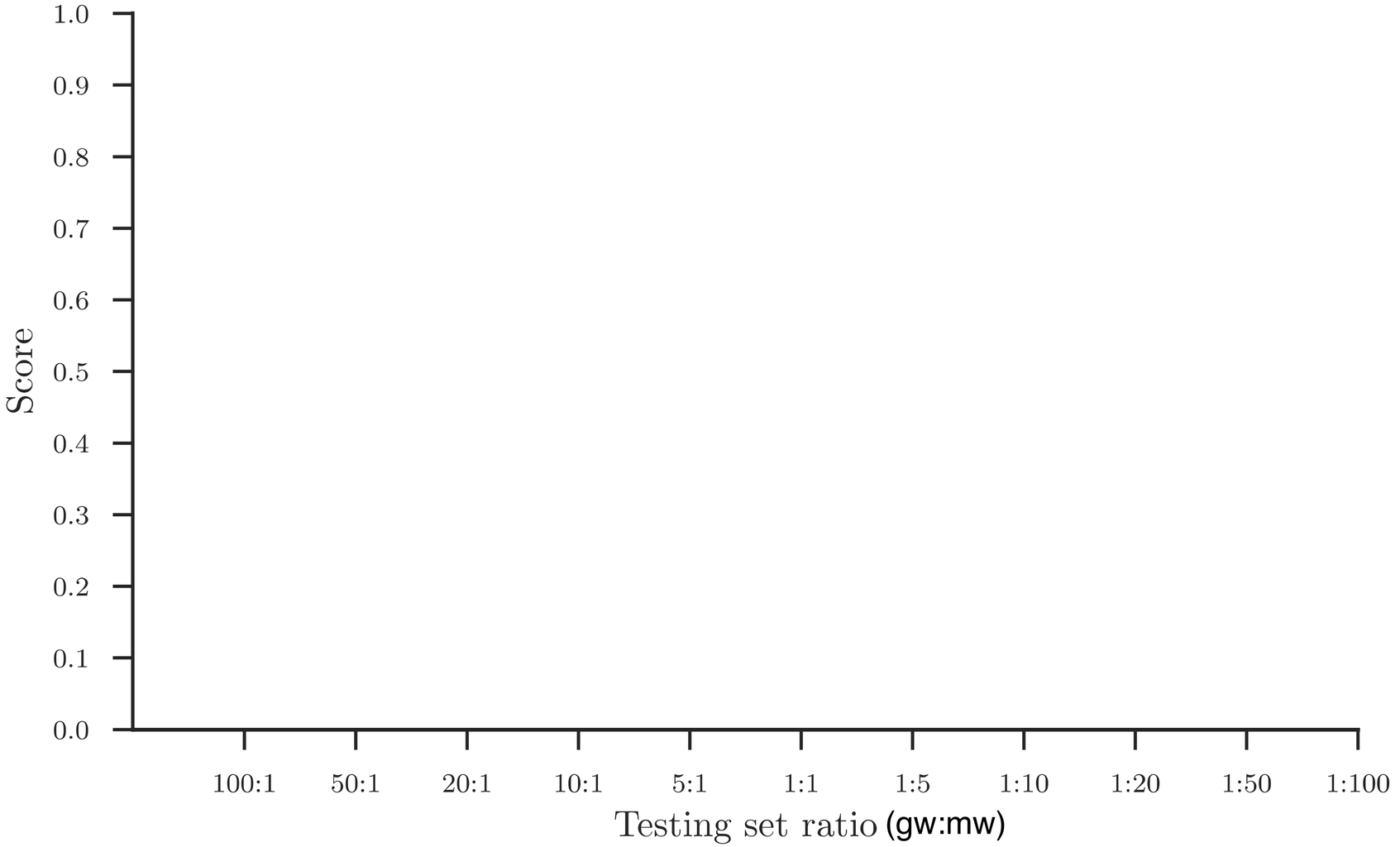
Unrealistic Test Class Ratio

- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)

Sources of Experimental Bias (3/3)

Unrealistic Test Class Ratio

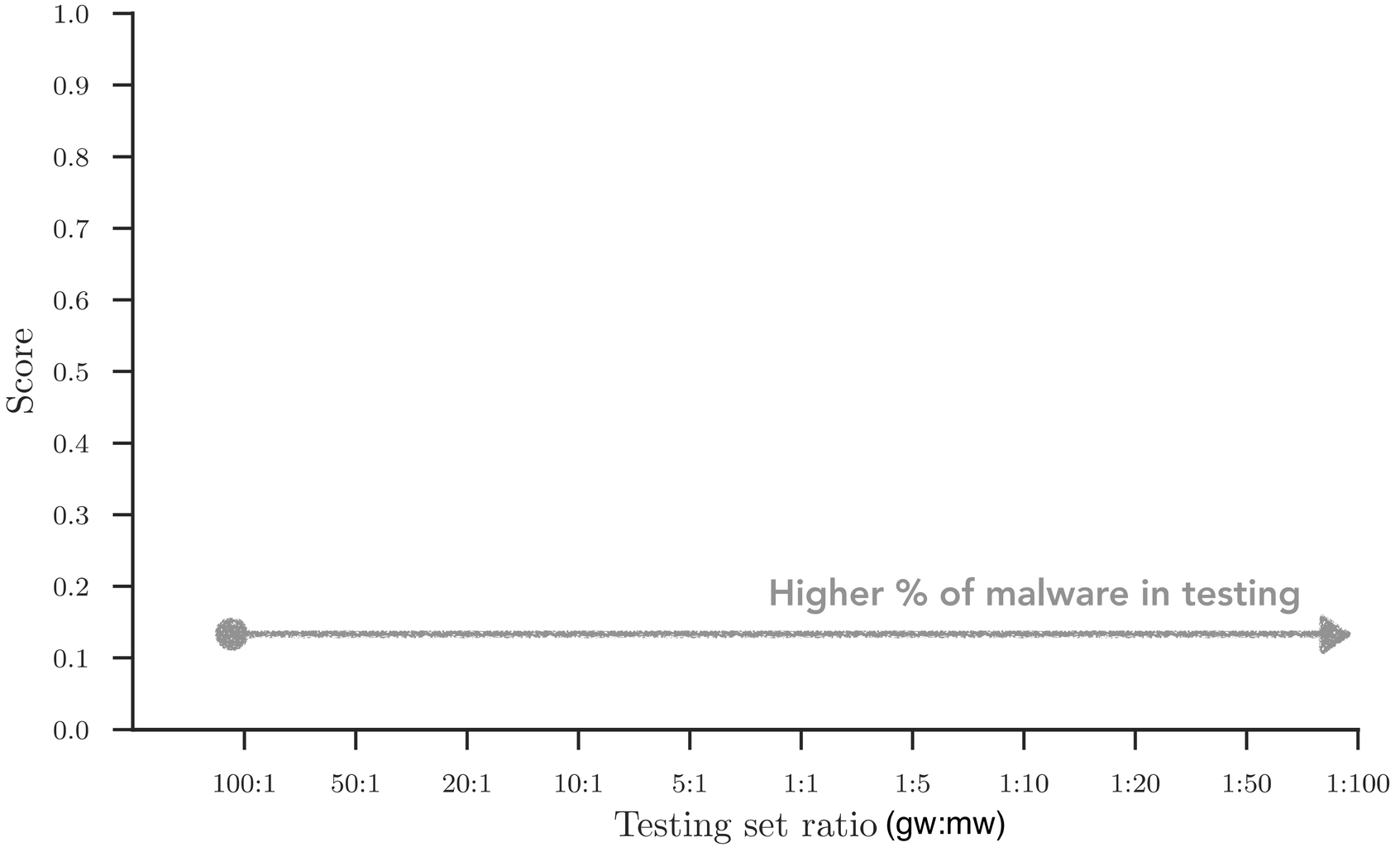
- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)



Sources of Experimental Bias (3/3)

Unrealistic Test Class Ratio

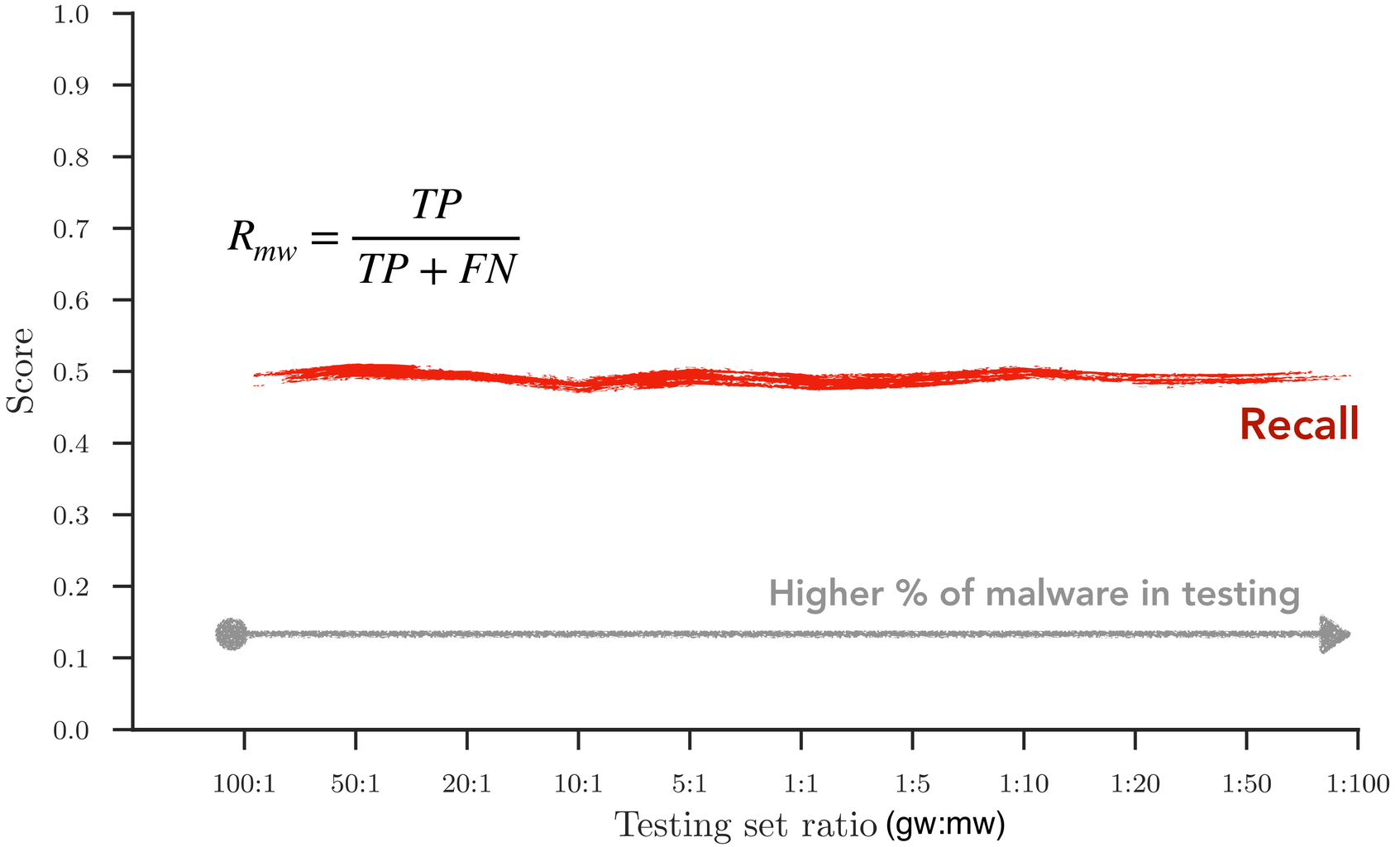
- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)



Sources of Experimental Bias (3/3)

Unrealistic Test Class Ratio

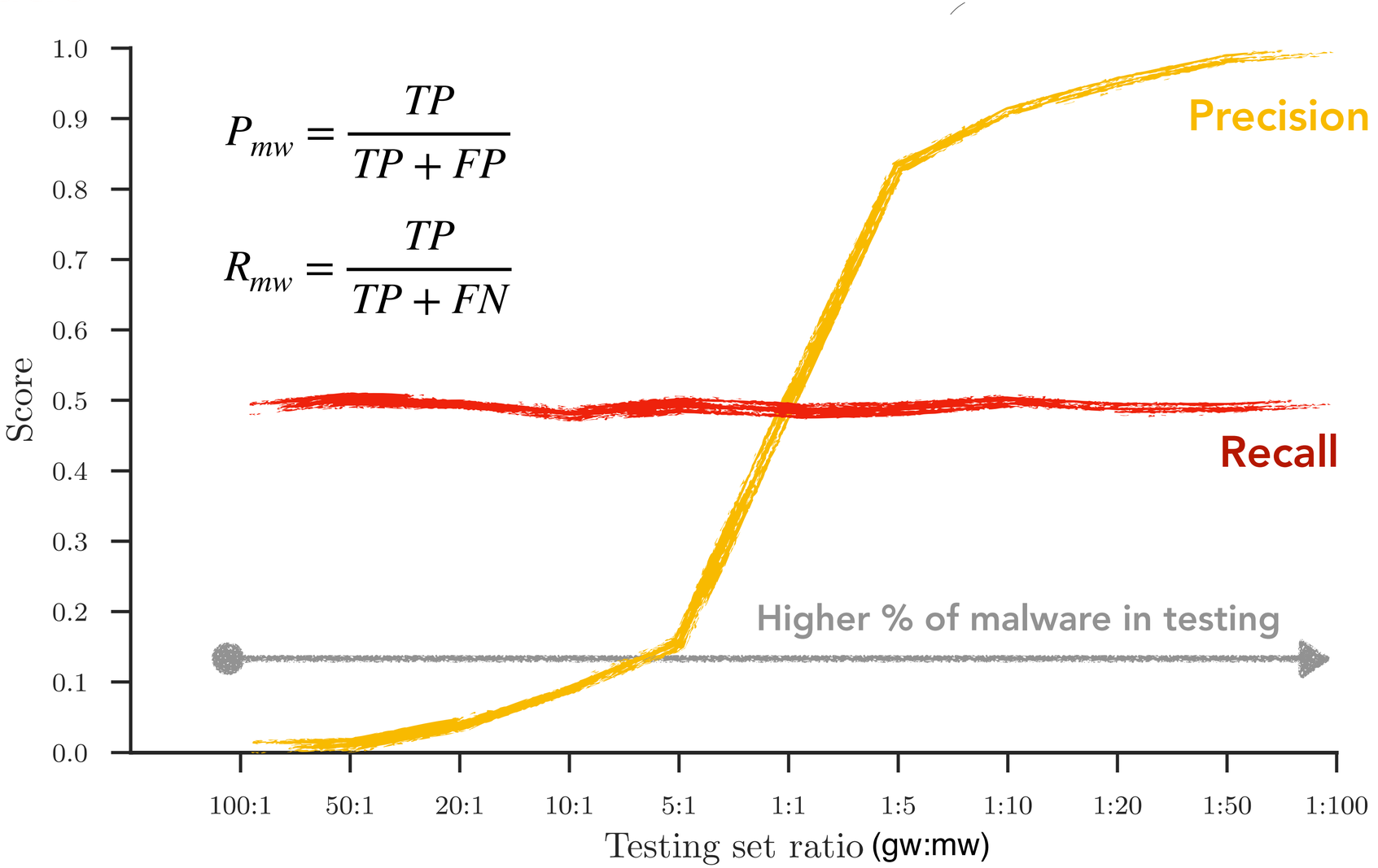
- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)



Sources of Experimental Bias (3/3)

Unrealistic Test Class Ratio

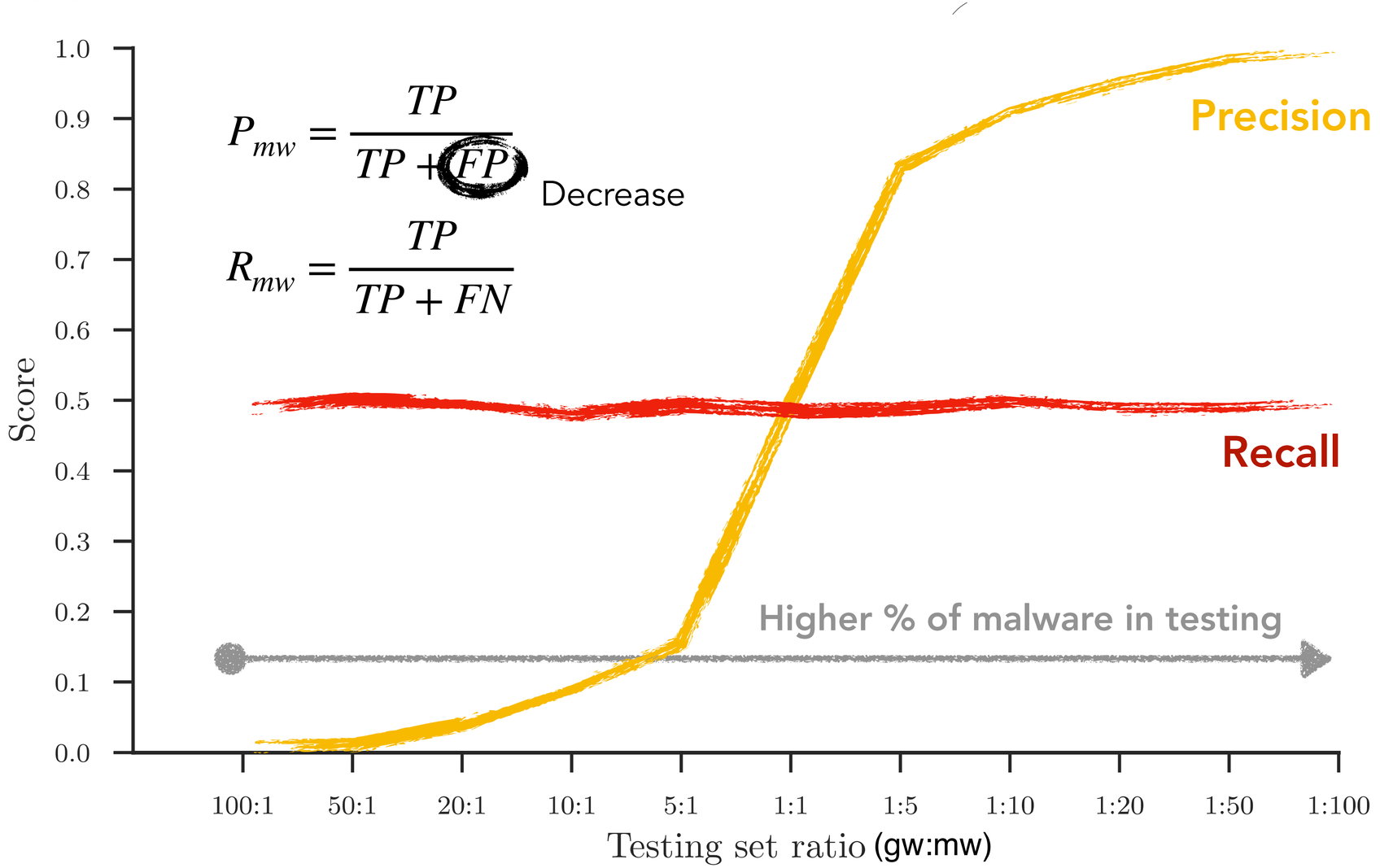
- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)



Sources of Experimental Bias (3/3)

Unrealistic Test Class Ratio

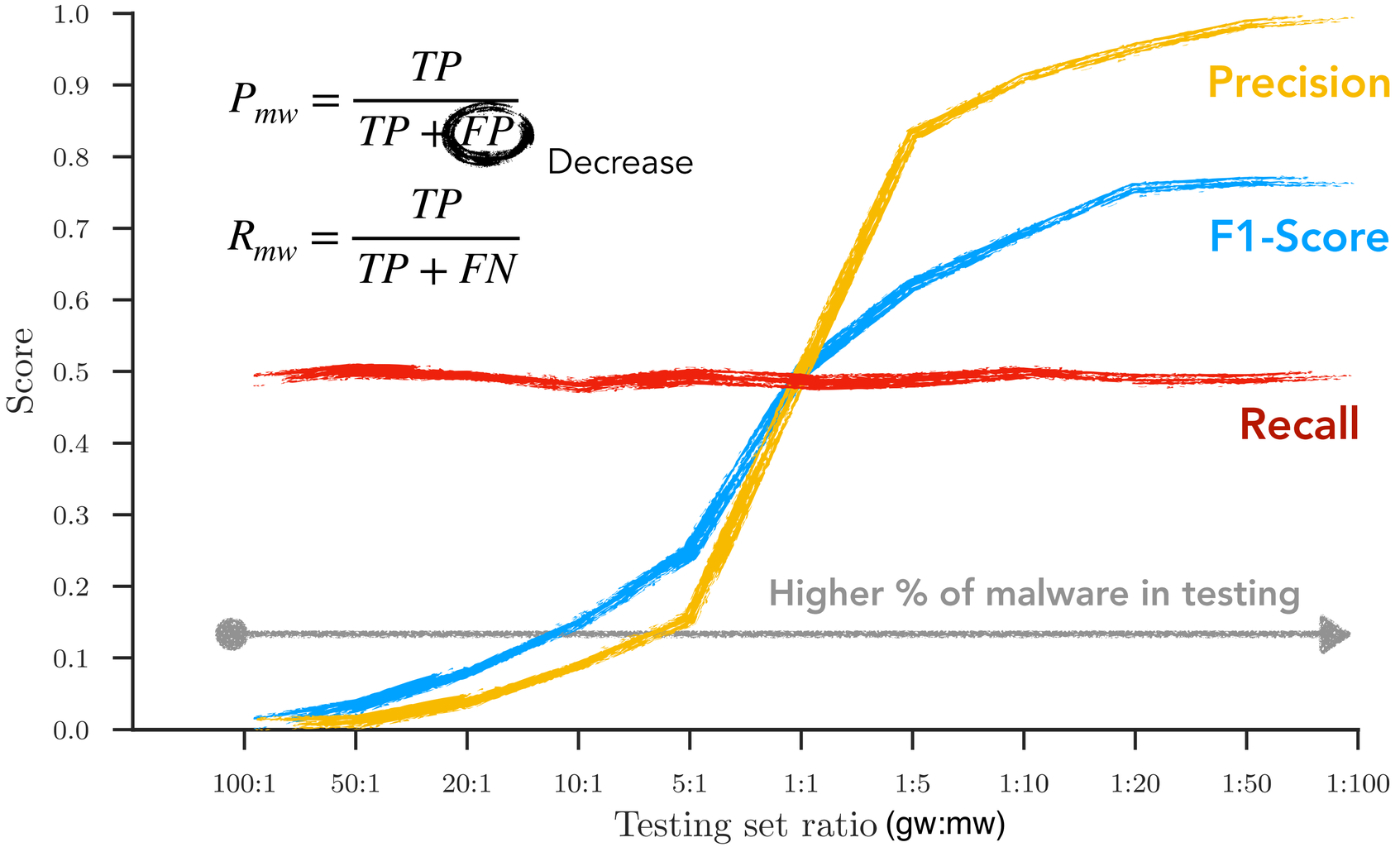
- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)



Sources of Experimental Bias (3/3)

Unrealistic Test Class Ratio

- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)

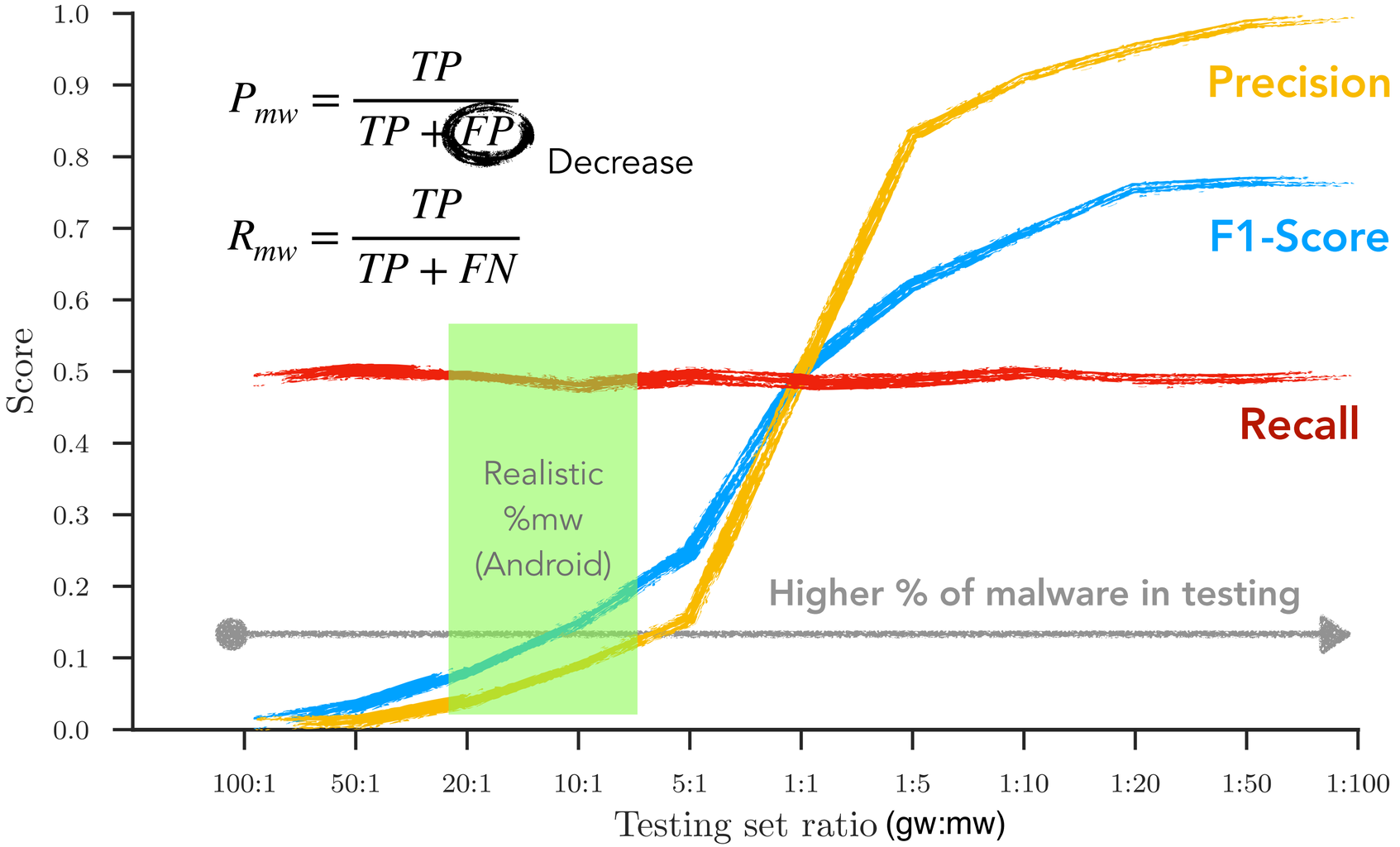


Sources of Experimental Bias (3/3)

Unrealistic Test Class Ratio

Violations produce unrealistic results

- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)



Prior Work on Adv. Malware

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Prior Work on Adv. Malware

Prior work was fundamental to initially explore problem-space attacks.
We propose a principled approach that supports reasoning.

Available Transformations

- Limiting #features modified [ESORICS'17]

Robustness to Preprocessing

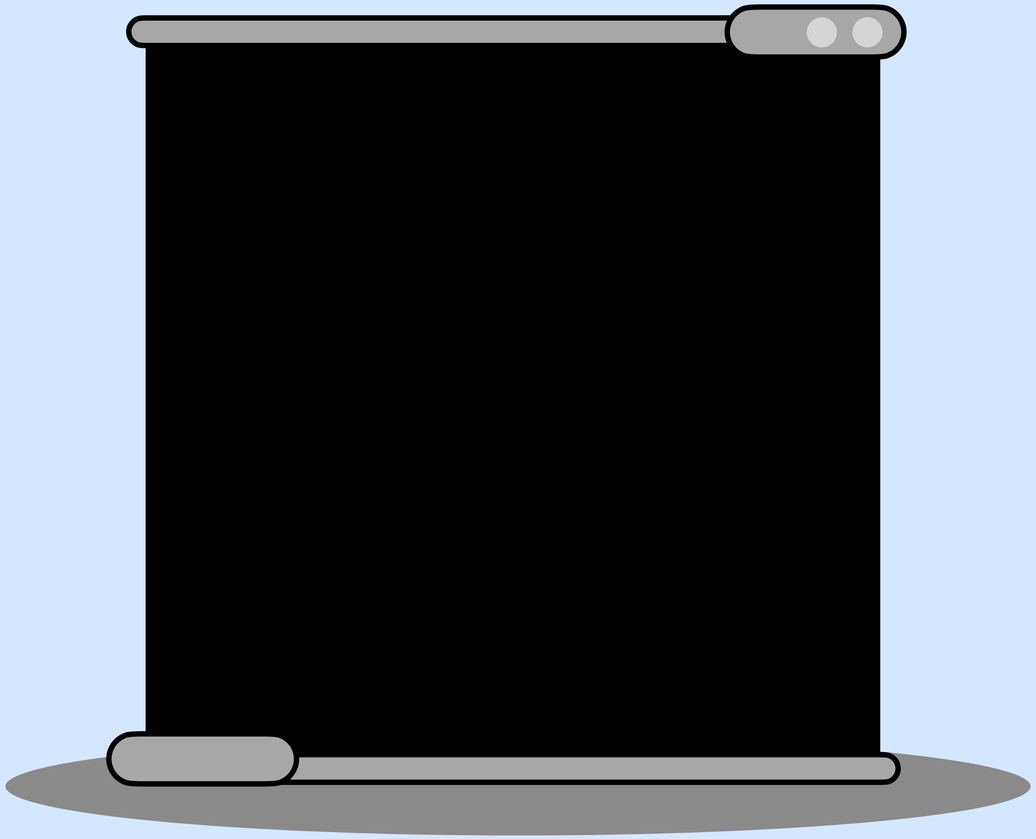
- Removable unused permissions [ESORICS'17]
- Removal code (unreachable, no-op) [EUSIPCO'18, RAID'18]
- Unclear [ACSAC'18]

Preserved Semantics

- Highly unstable transformations [ACSAC'18]

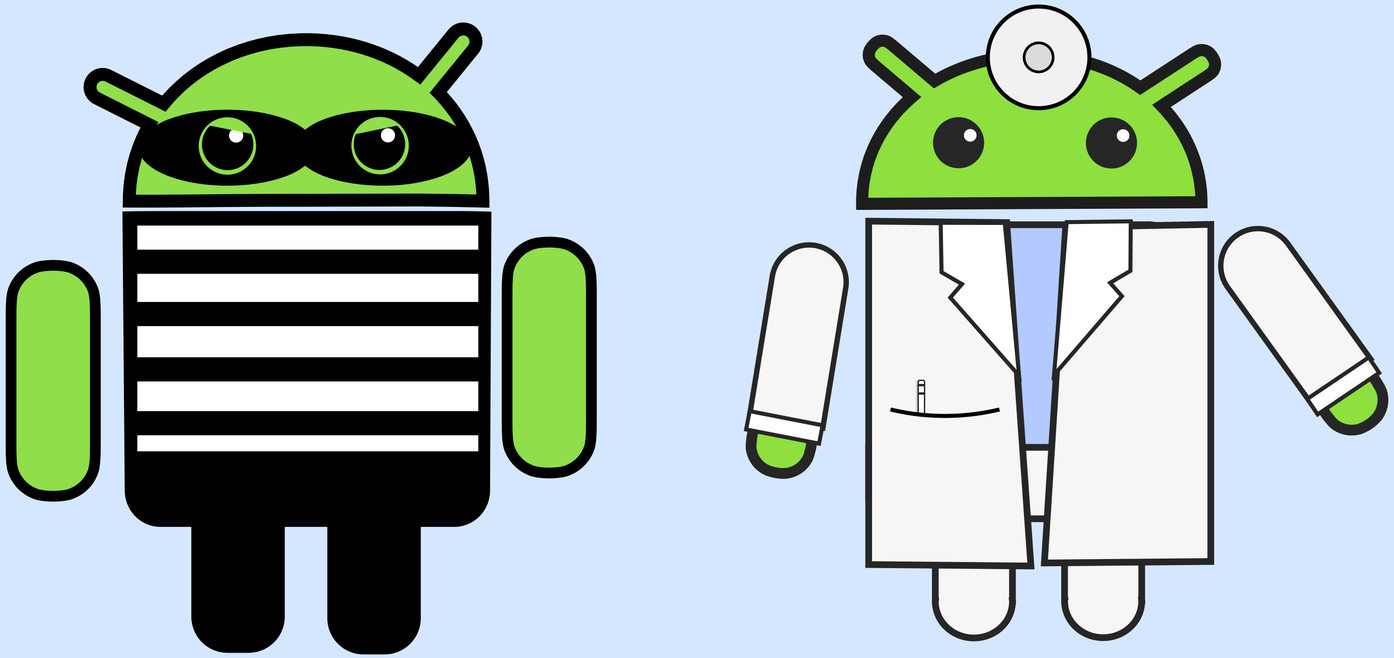
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Our Android Attack



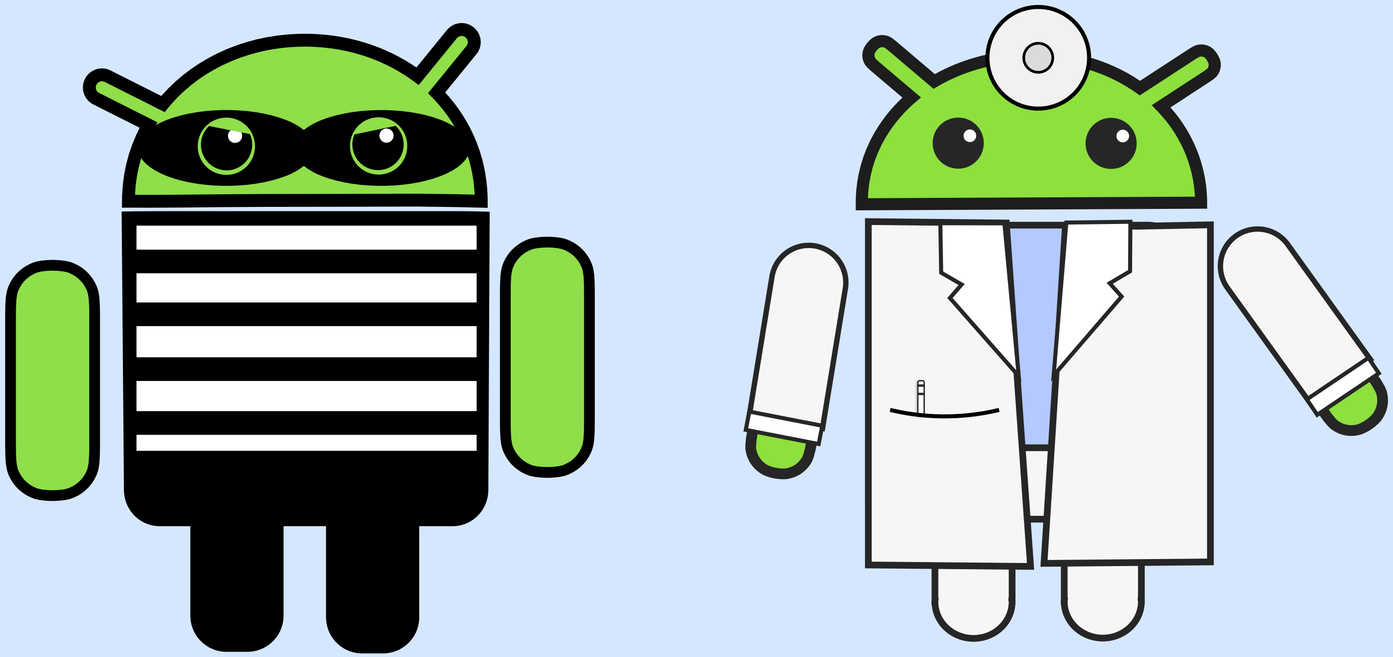
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Our Android Attack



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

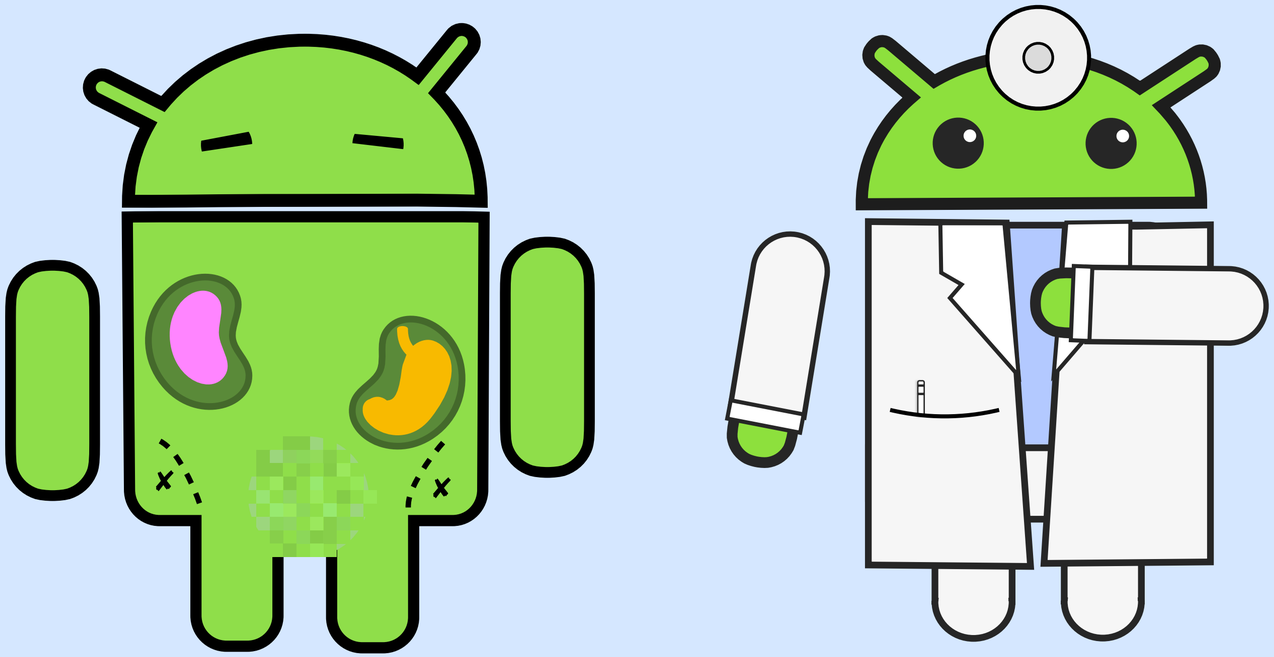
Our Android Attack



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Our Android Attack

Available Transformations
Code addition through automated software transplantation.

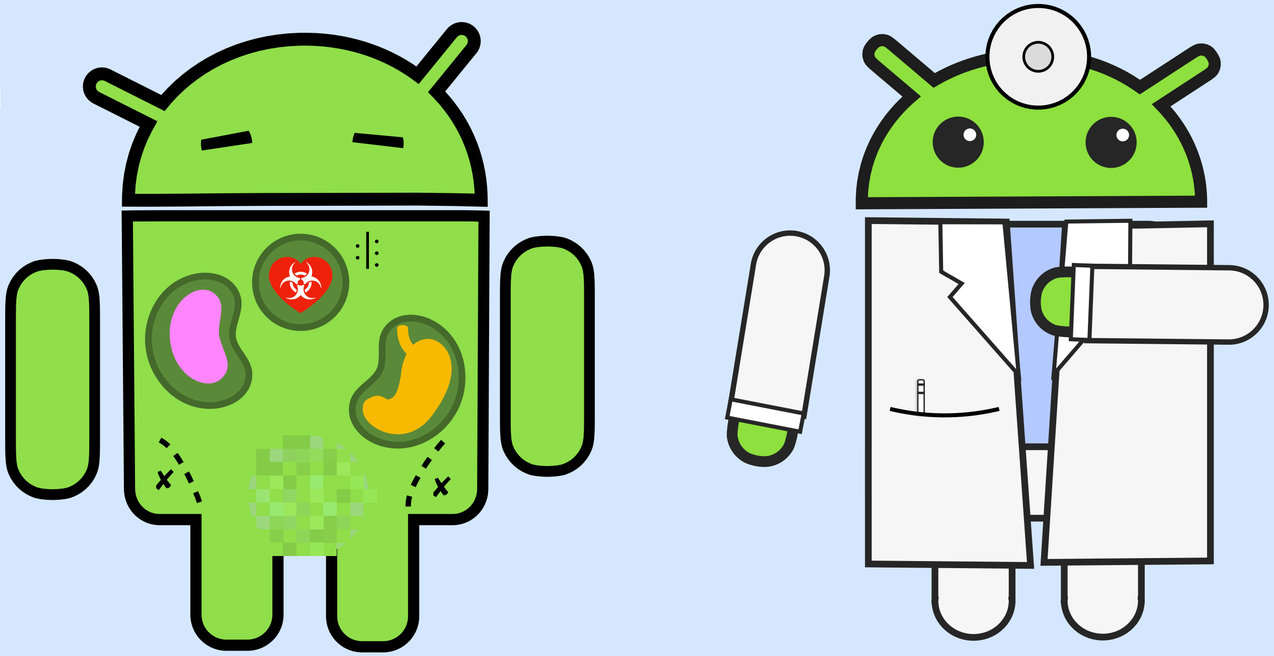


[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Our Android Attack

Available Transformations
Code addition through automated software transplantation.

Preserved Semantics
Malicious semantics preserved by construction using opaque predicates (inserted code is not executed at runtime).



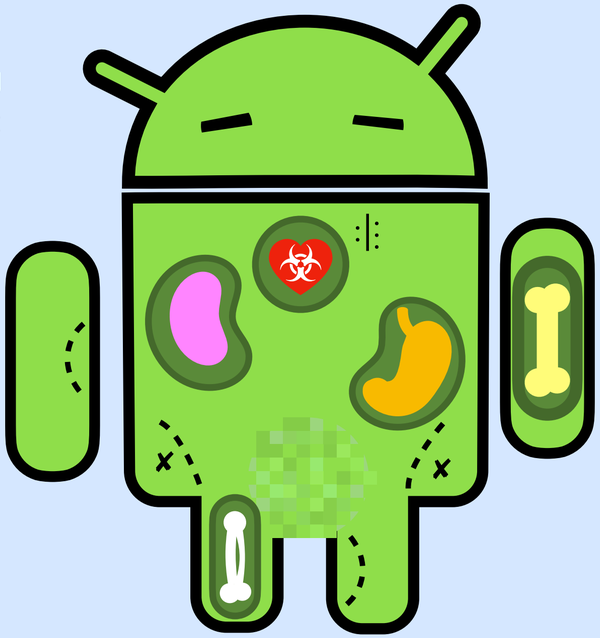
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

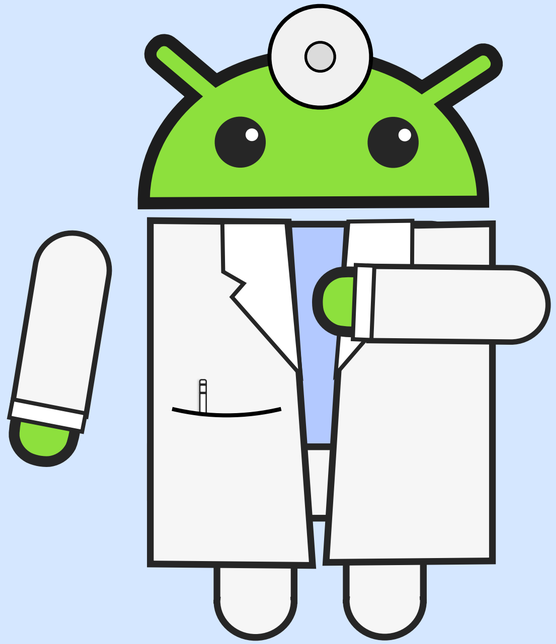
Our Android Attack

Available Transformations
Code addition through automated software transplantation.

Preserved Semantics
Malicious semantics preserved by construction using opaque predicates (inserted code is not executed at runtime).

Robustness to Preprocessing
We're robust to:
- removal of redundant code
- undeclared variables
- unlinked resources
- undefined references
- naming conflicts
- no-op instructions.





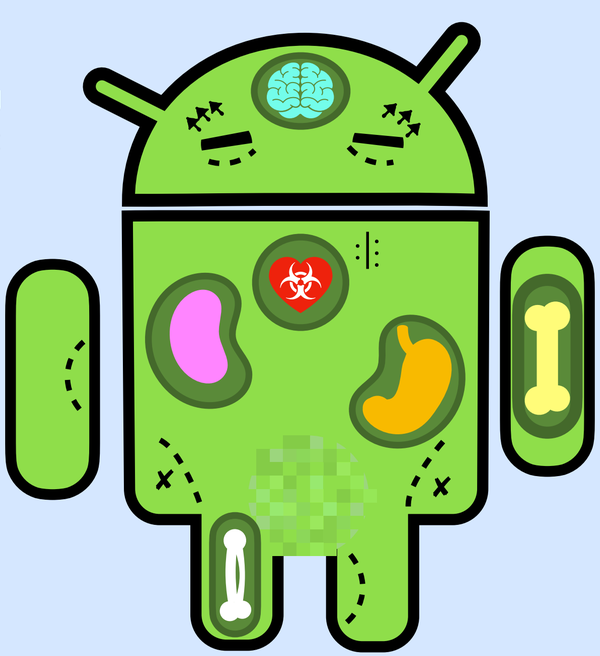
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

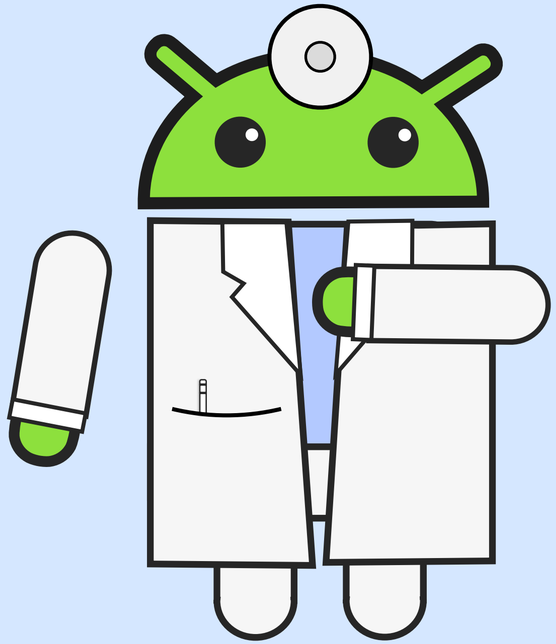
Our Android Attack

Available Transformations
Code addition through automated software transplantation.

Preserved Semantics
Malicious semantics preserved by construction using opaque predicates (inserted code is not executed at runtime).

Robustness to Preprocessing
We're robust to:
- removal of redundant code
- undeclared variables
- unlinked resources
- undefined references
- naming conflicts
- no-op instructions.

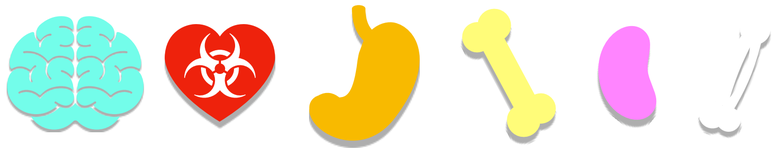




Plausibility
Only realistic code is injected (rather than orphaned urls, api calls, etc.)
Mutated apps install and start on an emulator.

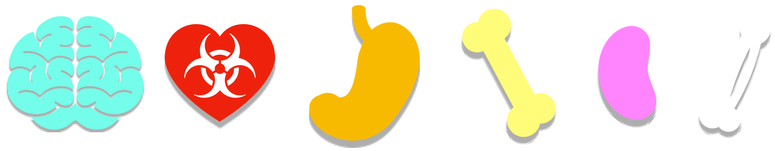
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting

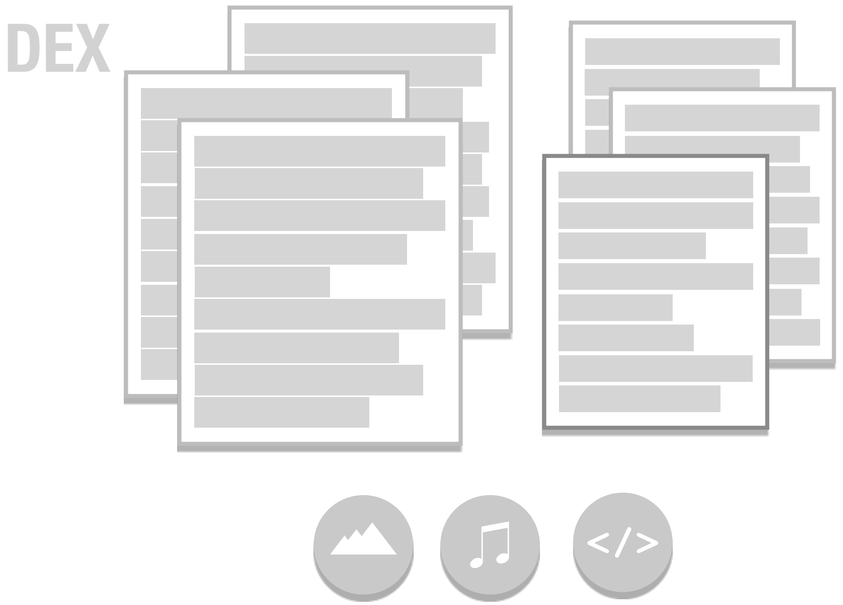


[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



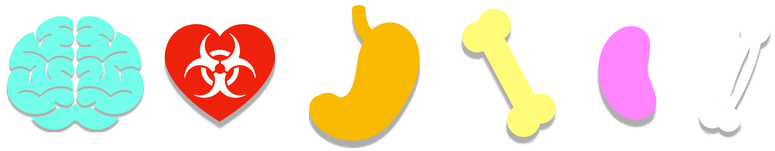
1. Identify feature entry point



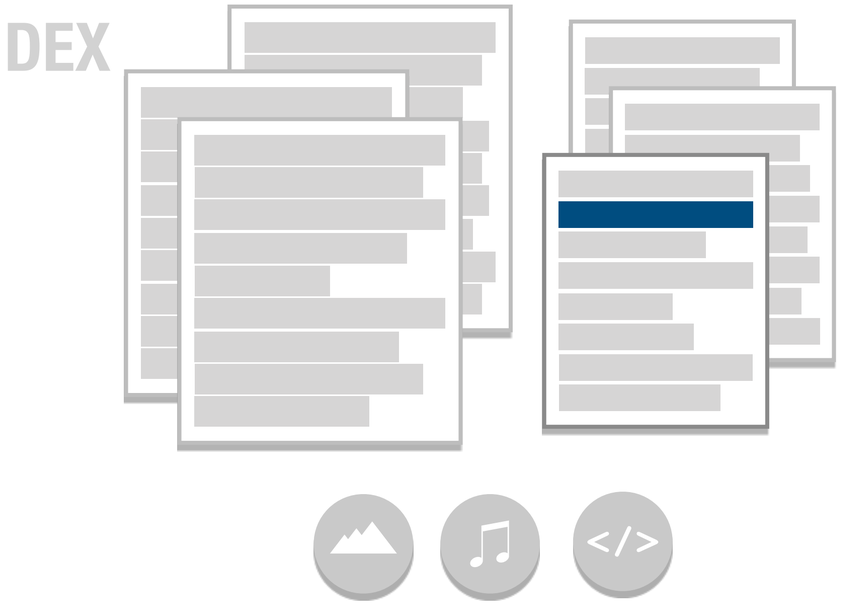
Identify activity in dex

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



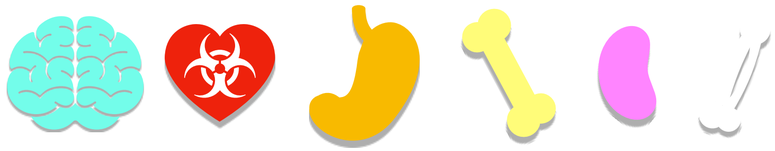
1. Identify feature entry point



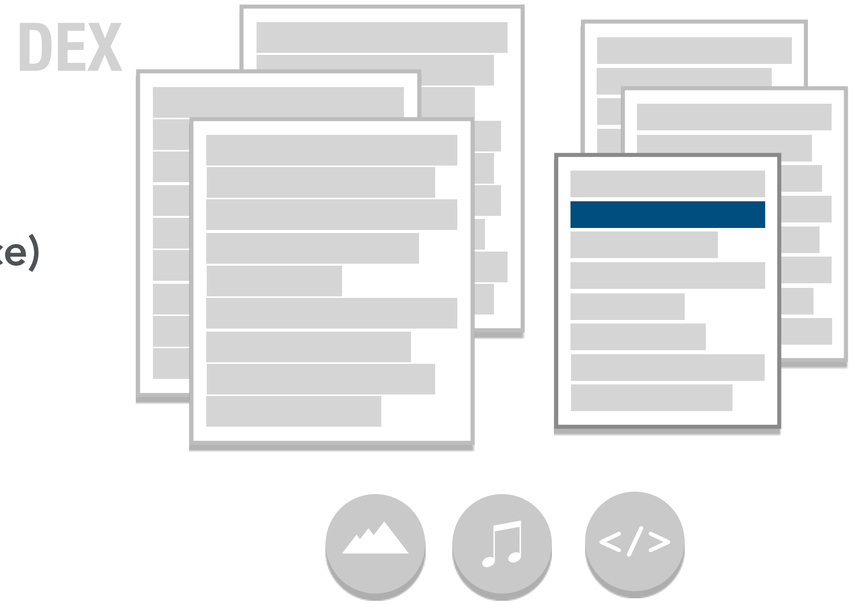
Identify activity in dex

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



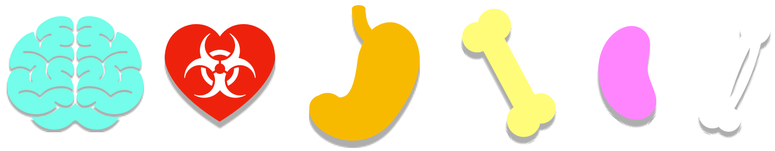
2 Choose any vein (backward slice)



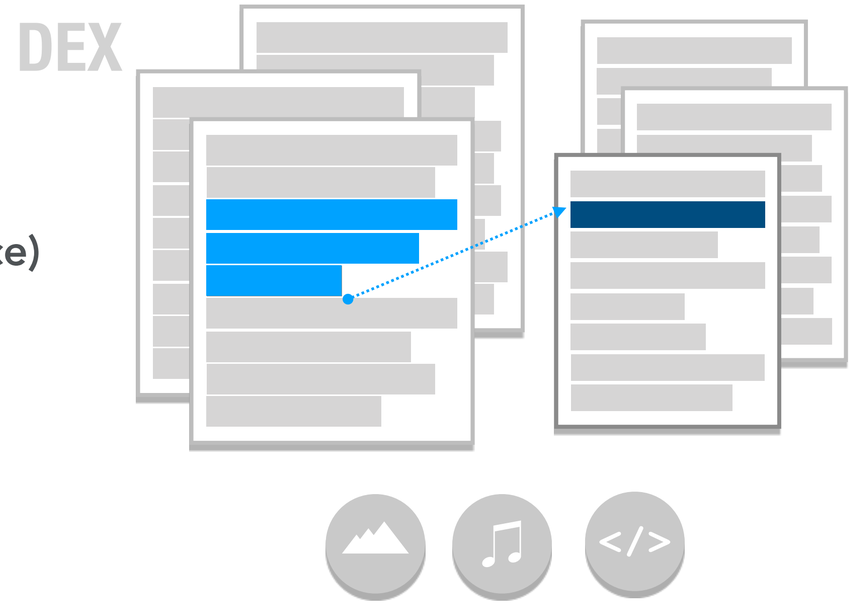
Extract intent creation and startActivity()

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



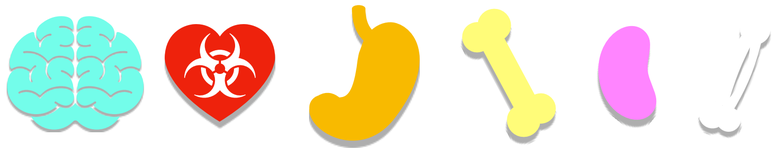
2 Choose any vein (backward slice)



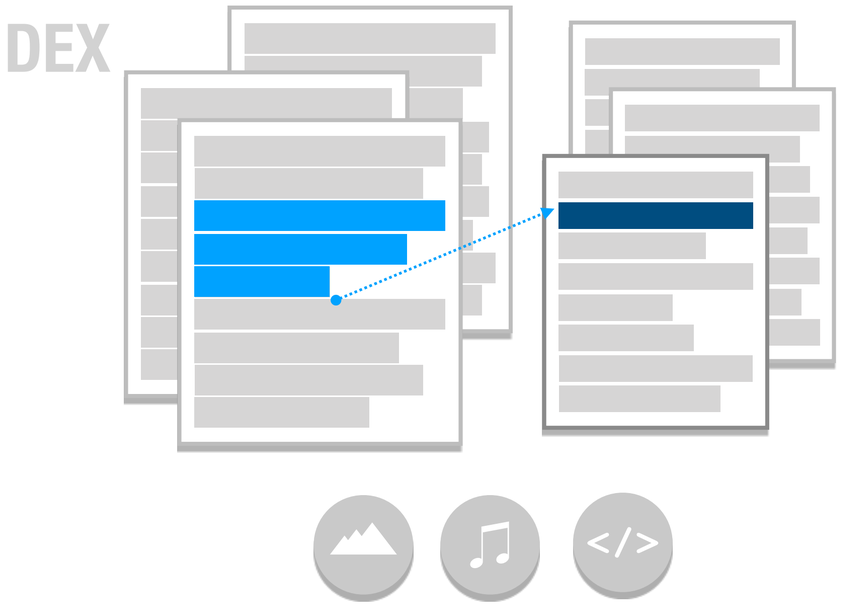
Extract intent creation and startActivity()

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



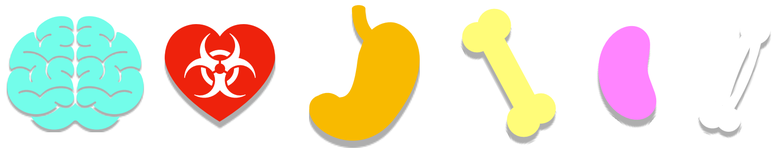
3 Collect organ (forward slice)



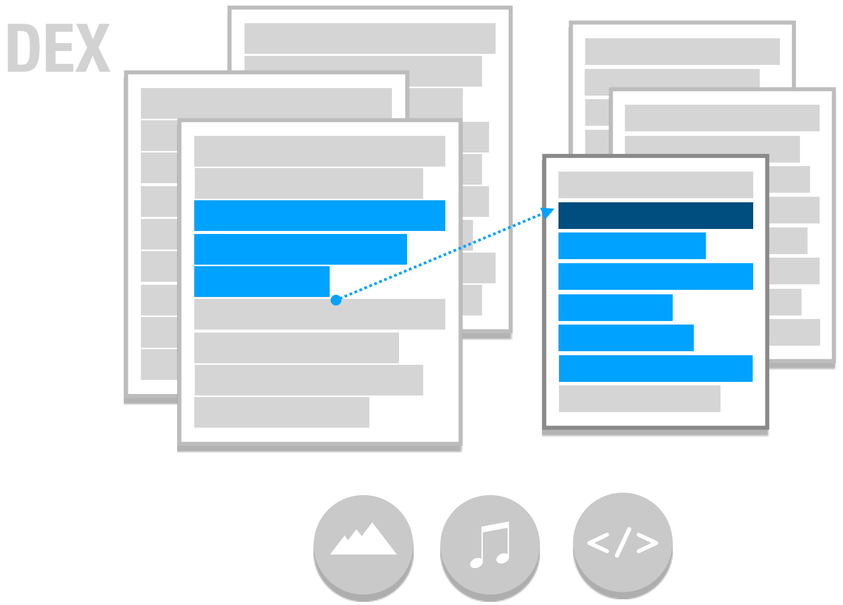
Gather activity definition

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



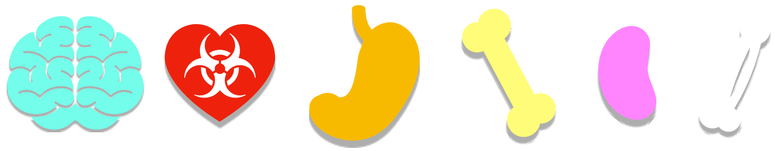
3 Collect organ (forward slice)



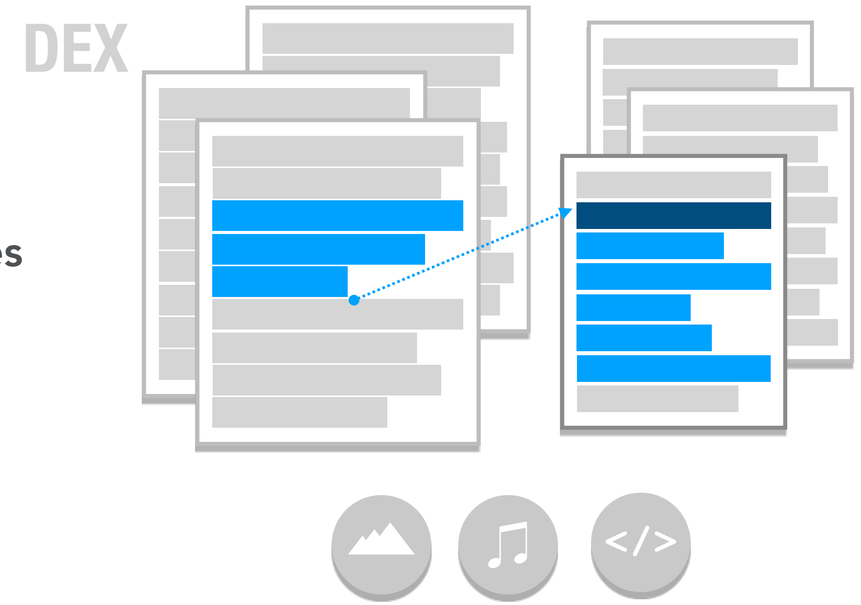
Gather activity definition

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



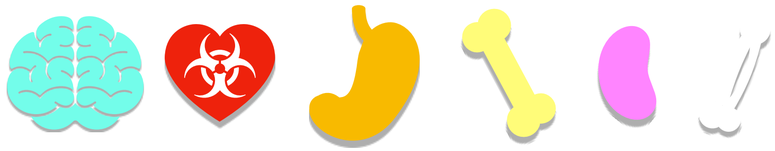
4 Include transitive dependencies



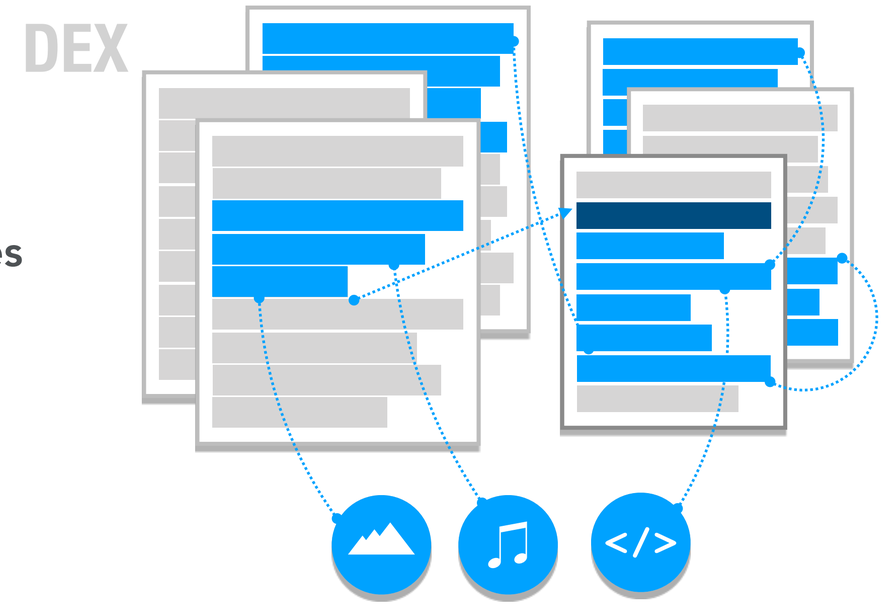
Recursively collect dependencies

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



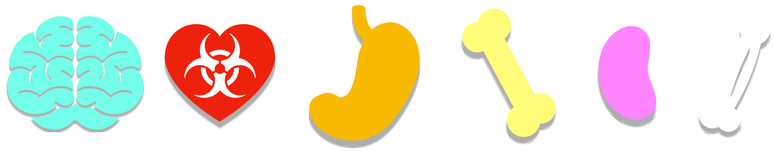
4 Include transitive dependencies



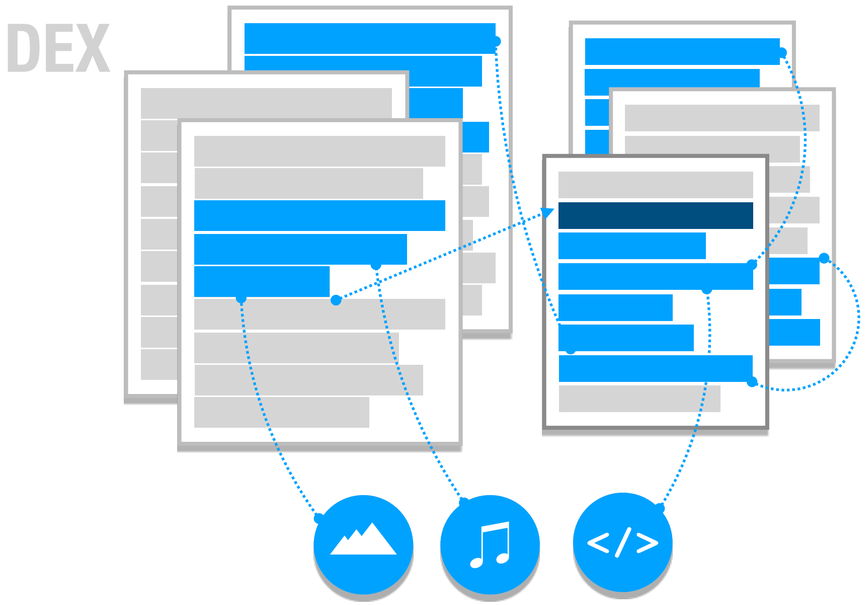
Recursively collect dependencies

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



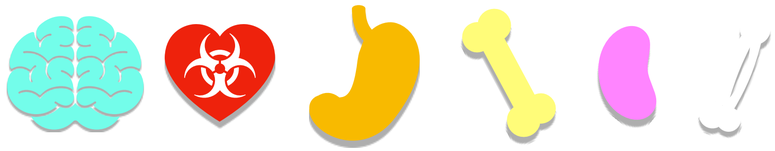
5 Store organ in an “ice box”



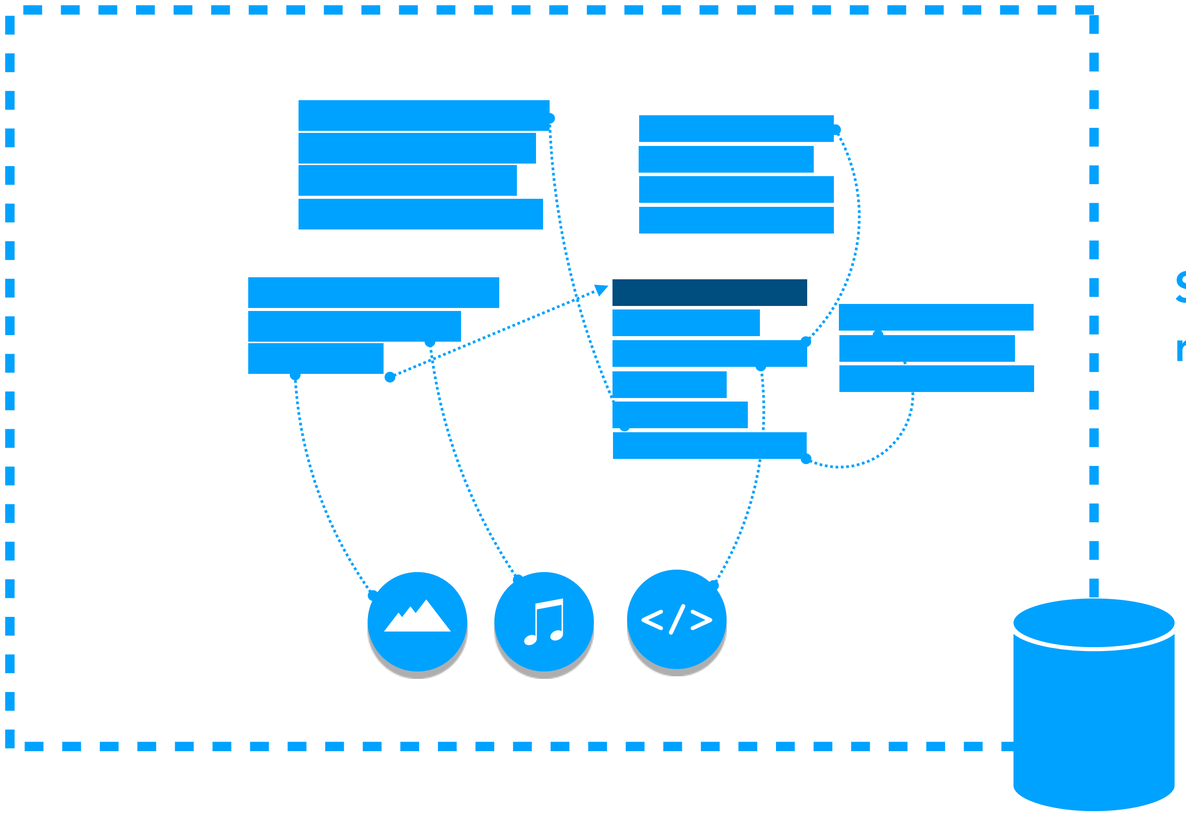
Save gadget to a database ready for the attack

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Organ Harvesting



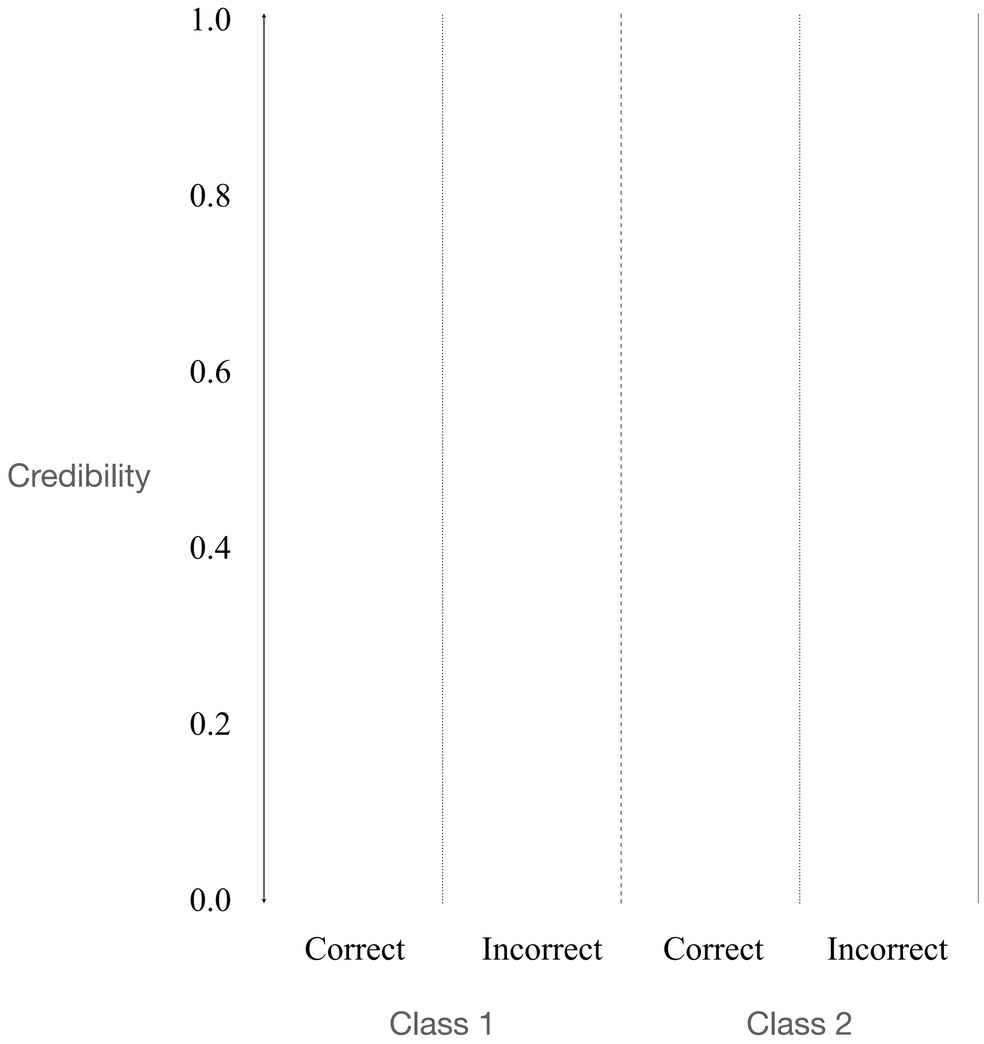
5 Store organ in an “ice box”



Save gadget to a database ready for the attack

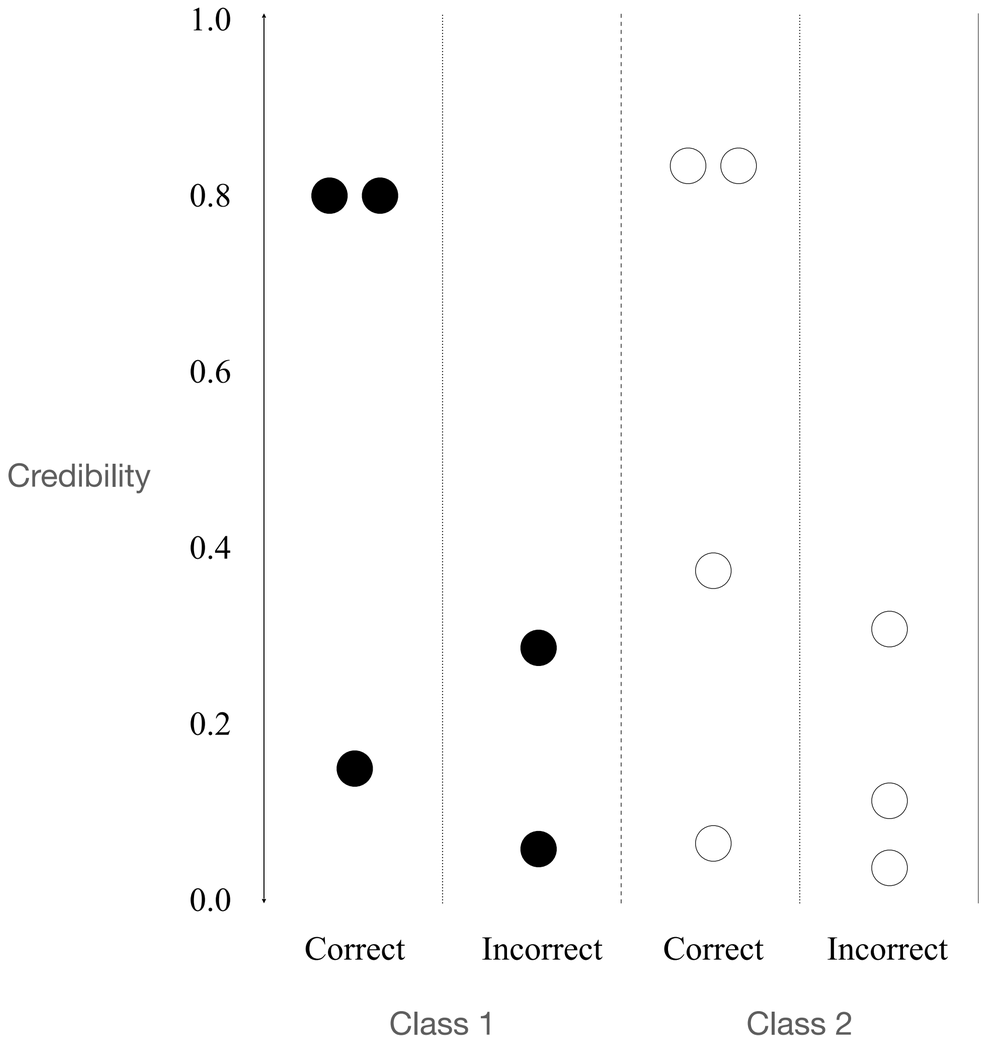
[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

Transcend Calibration



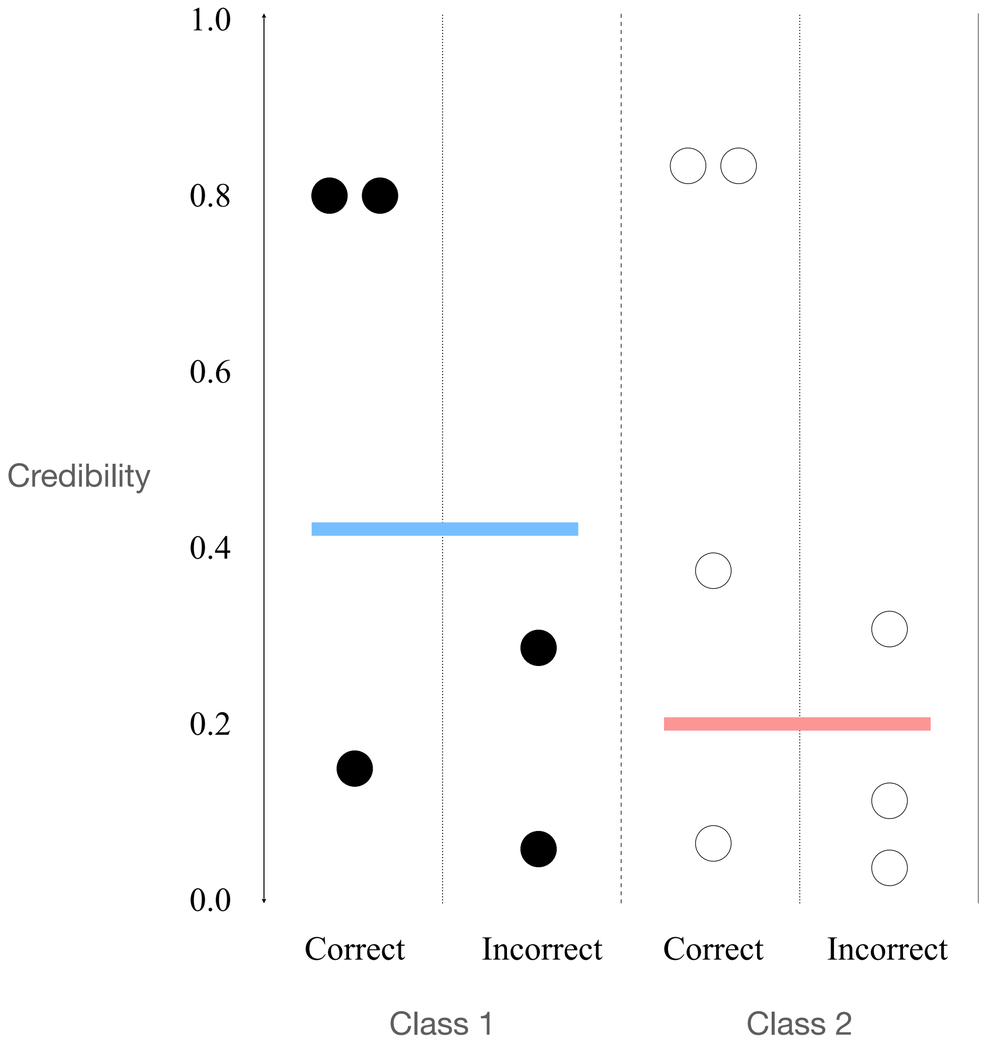
- **How much drift is too much?**
- Produce a threshold for each class
- Optimize cost vs performance on training and calibration sets
- Maximise separation between credibility of correct and incorrect decisions

Transcend Calibration



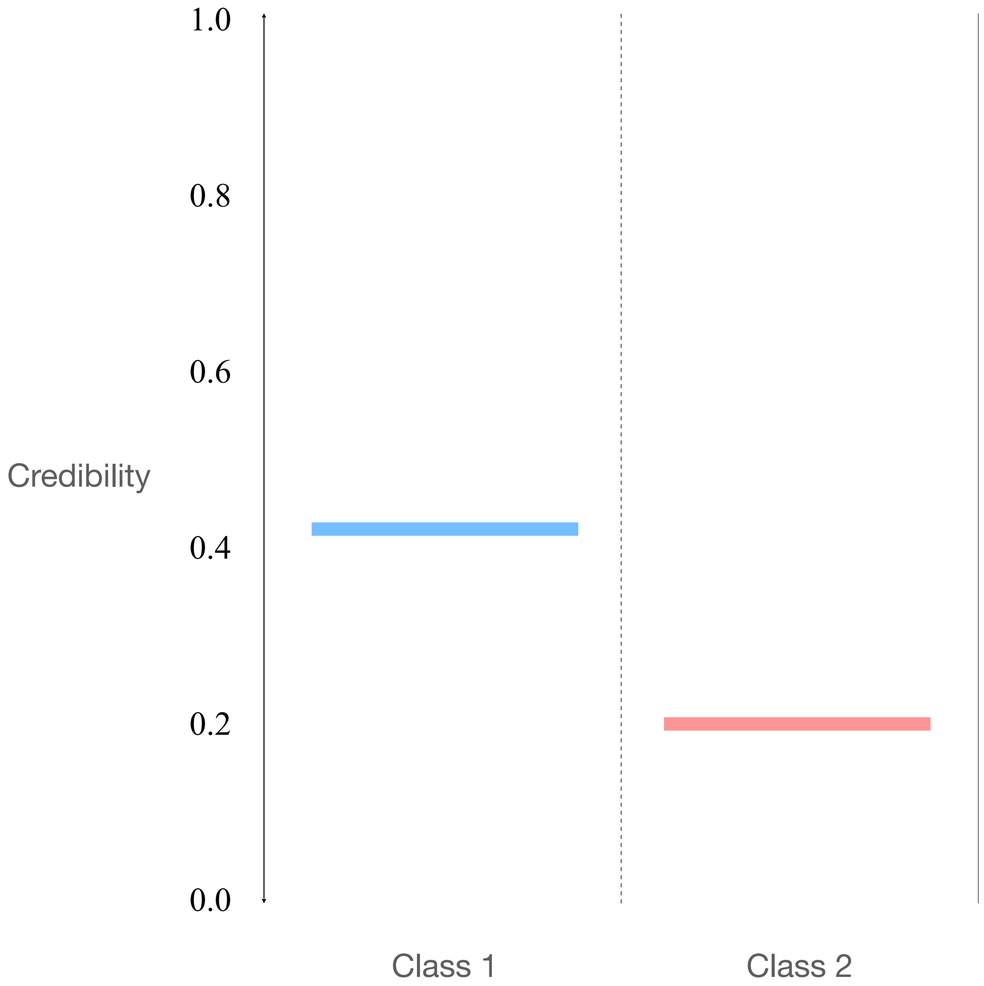
- **How much drift is too much?**
- Produce a threshold for each class
- Optimize cost vs performance on training and calibration sets
- Maximise separation between credibility of correct and incorrect decisions

Transcend Calibration



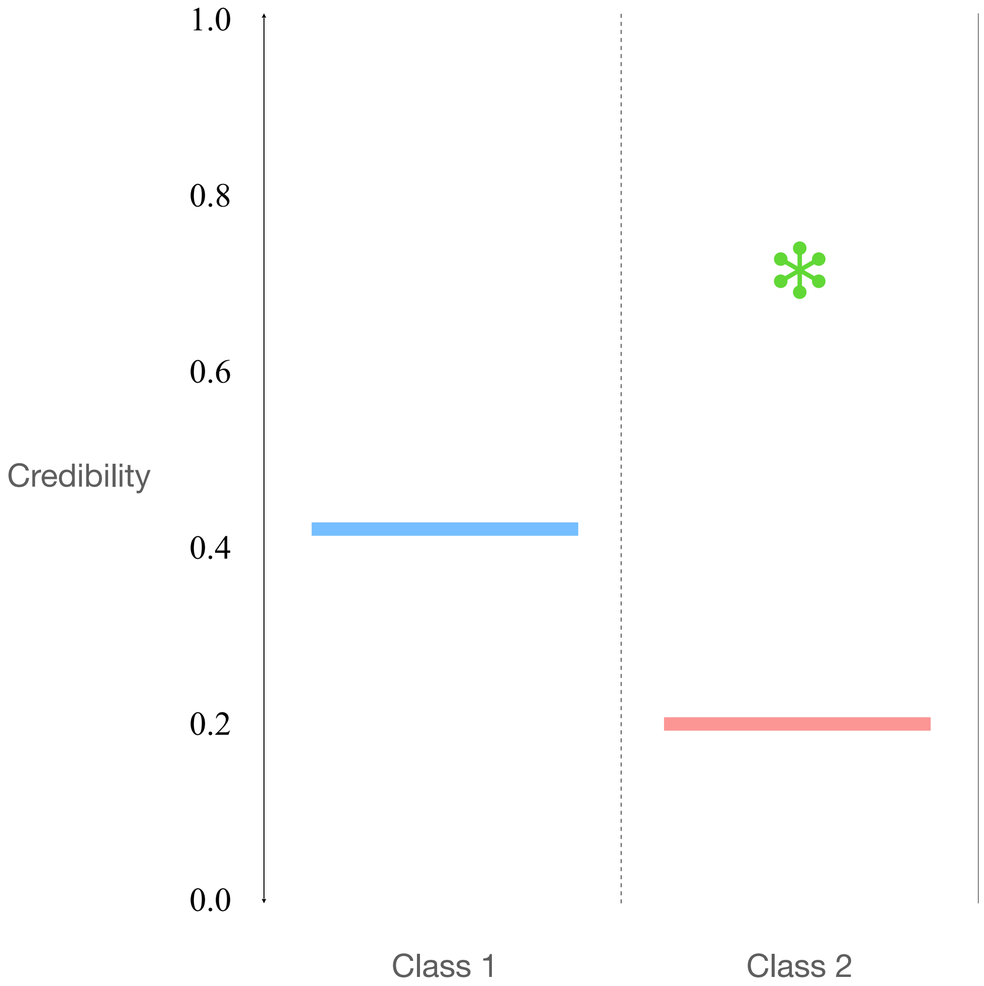
- **How much drift is too much?**
- Produce a threshold for each class
- Optimize cost vs performance on training and calibration sets
- Maximise separation between credibility of correct and incorrect decisions

Transcend at Test Time



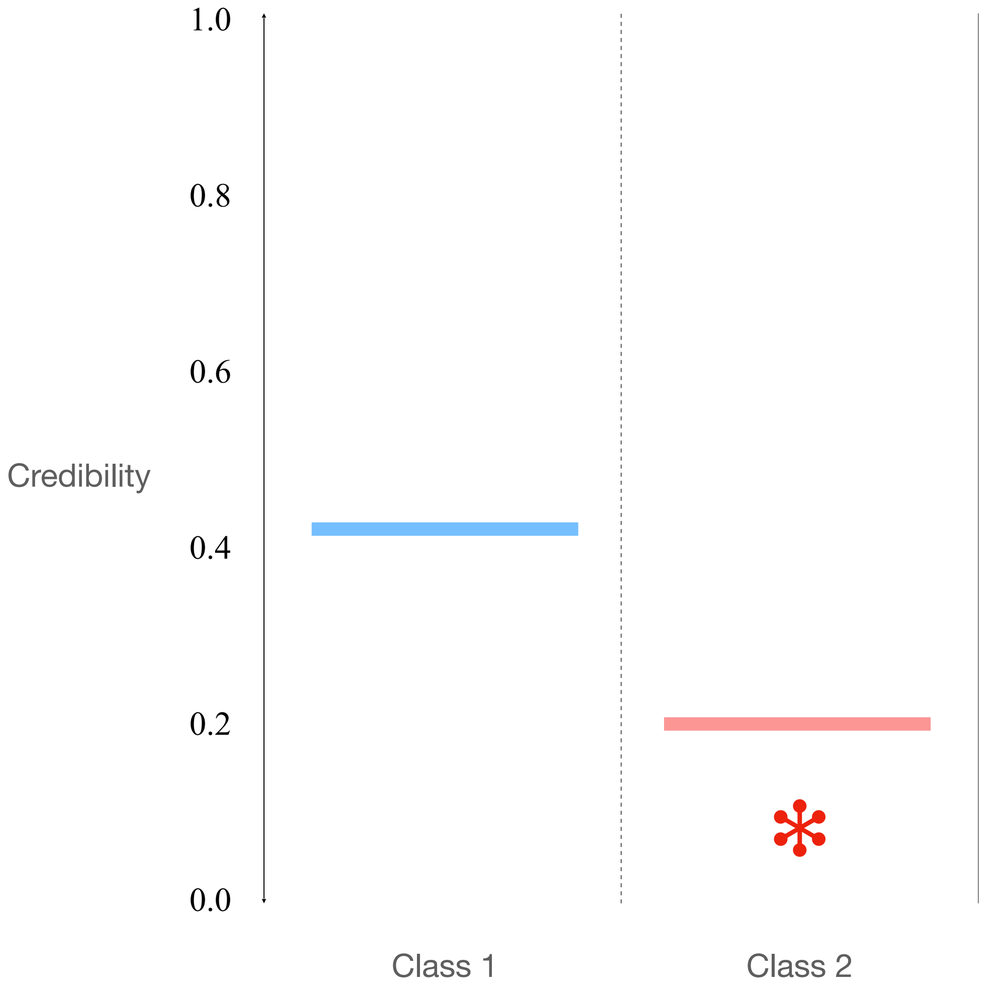
- Credibilities of new examples are compared against the threshold of their predicted class
- Above = keep the prediction
- Below = reject the prediction

Transcend at Test Time



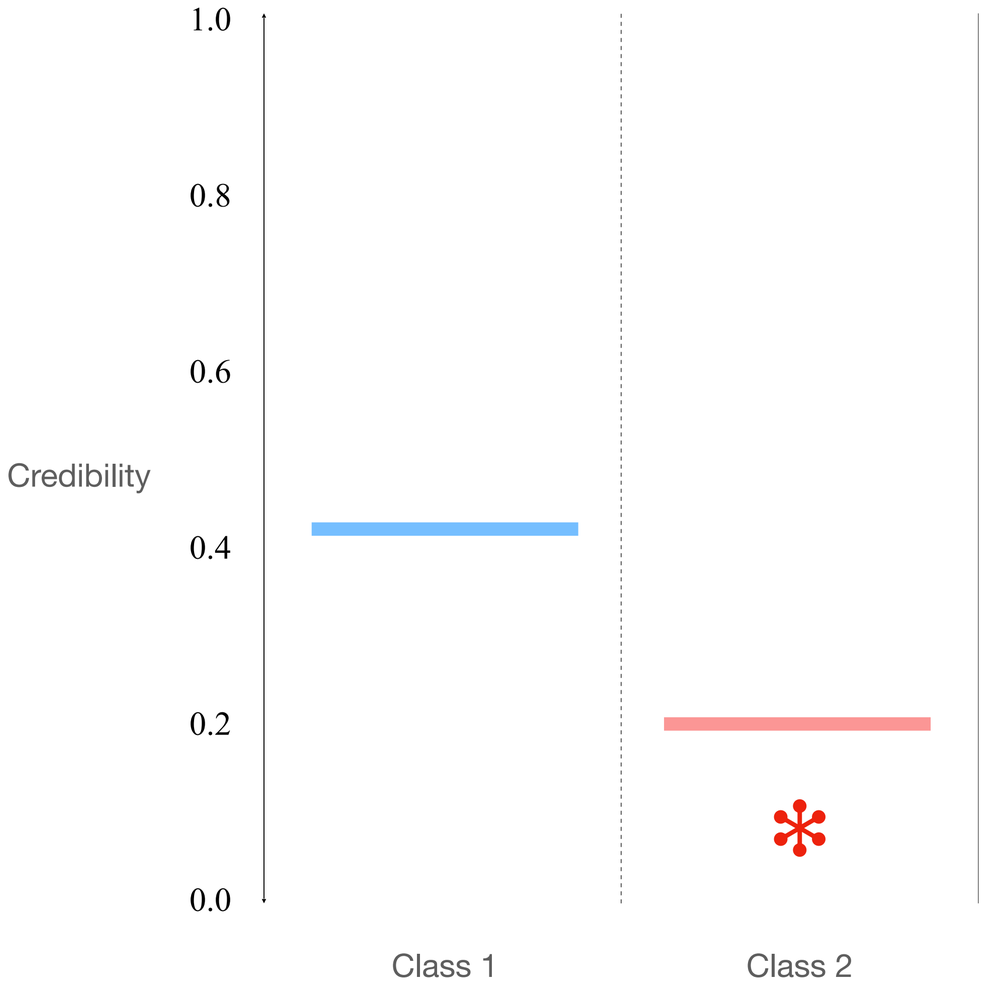
- Credibilities of new examples are compared against the threshold of their predicted class
- Above = keep the prediction
- Below = reject the prediction

Transcend at Test Time



- Credibilities of new examples are compared against the threshold of their predicted class
- Above = keep the prediction
- Below = reject the prediction

Rejection Cost

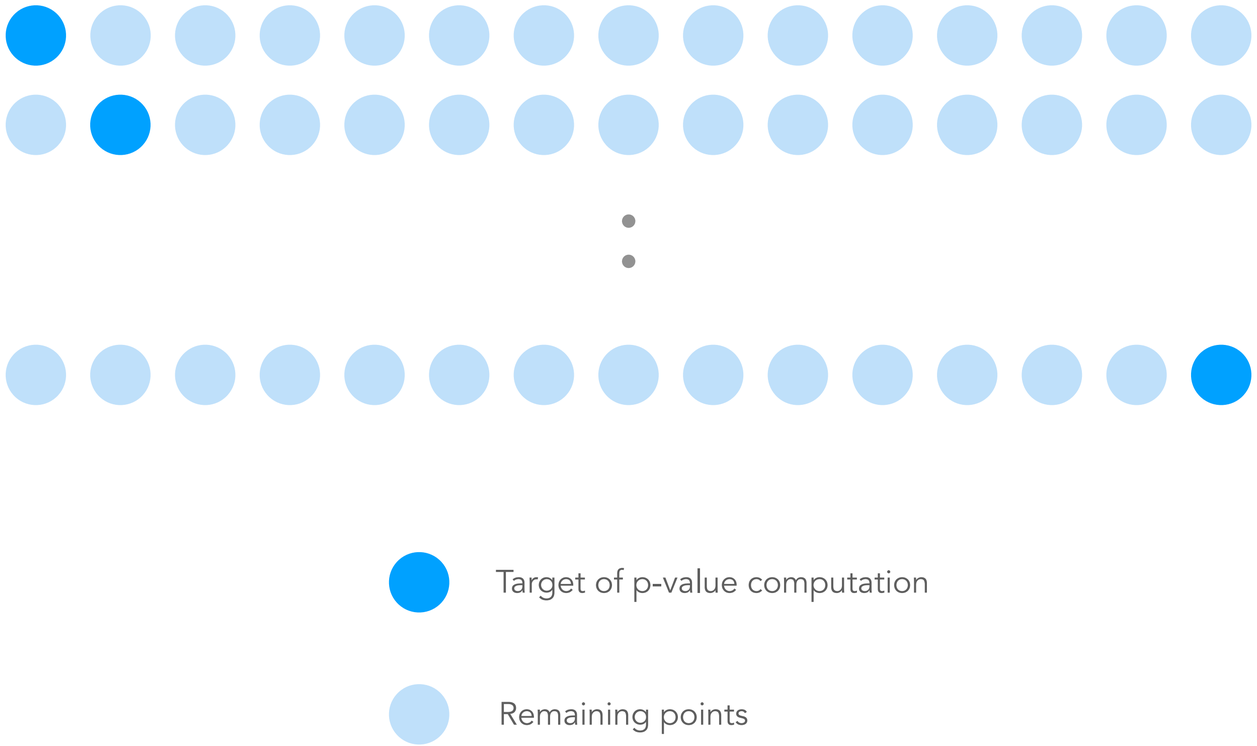


Rejection Cost

- Actions for rejected points *:
  - Manual inspection
  - Downstream analysis
  - Quarantine
  - Exemption

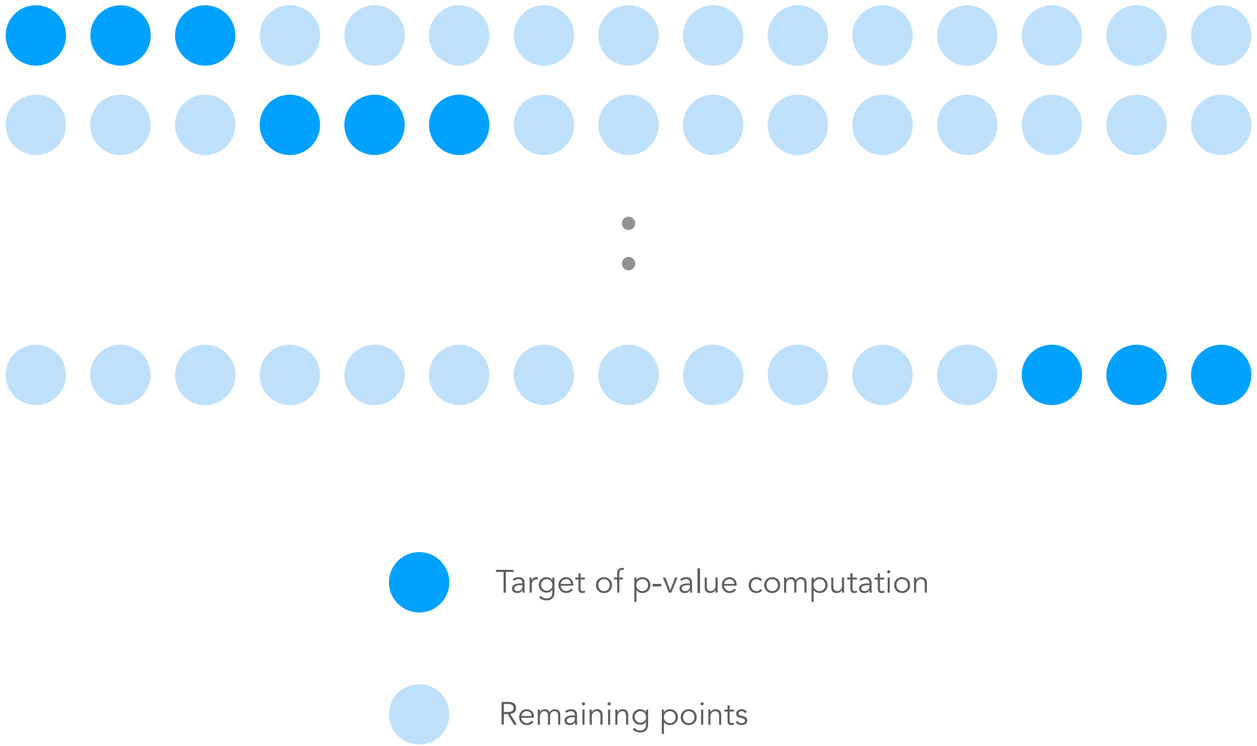
* [AISec 2021] Investigating Labelless Drift Adaptation for Malware Detection
* [AISec 2021] INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection

The Cost of Transductive Conformal Evaluators



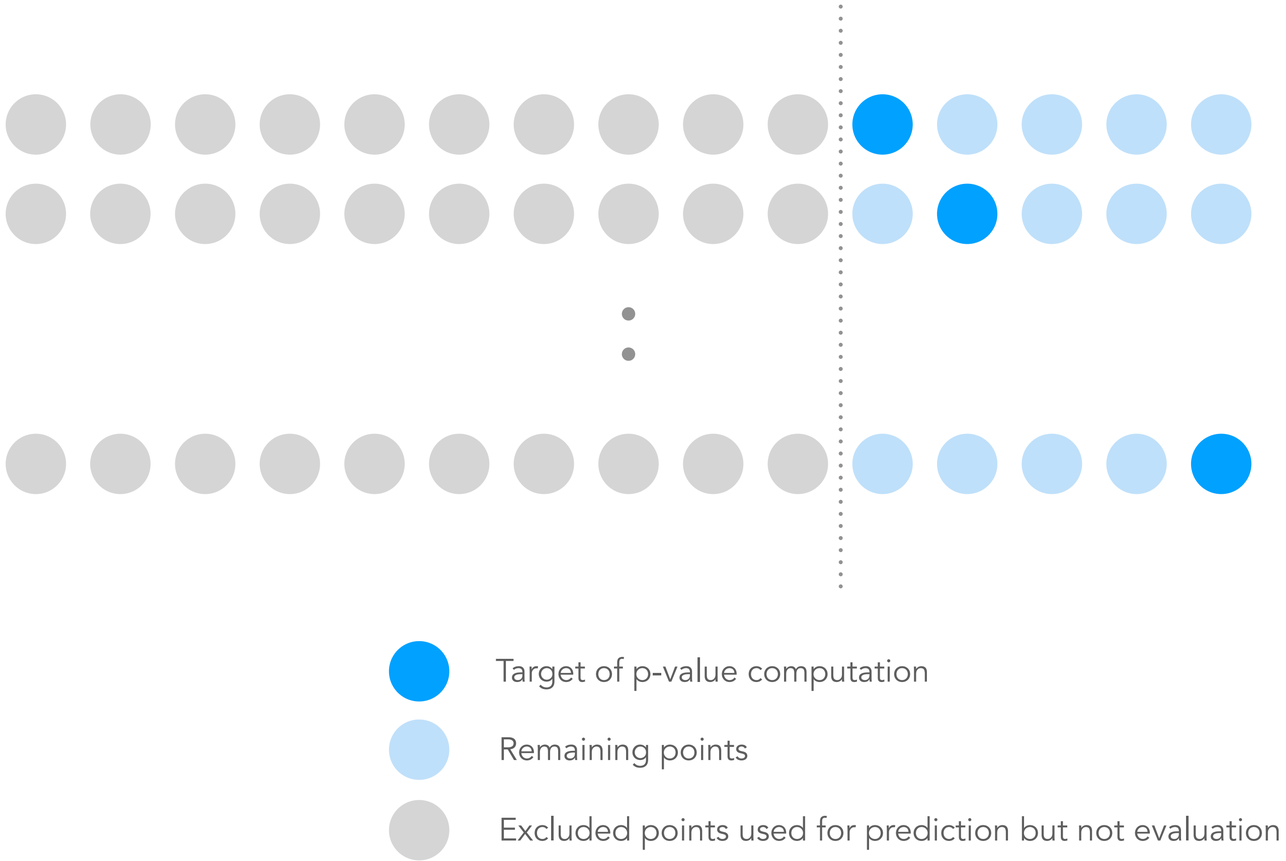
- Underlying classifier retrained for every training point
- Rooted in CP theory
- Often computationally infeasible

Approximate TCE



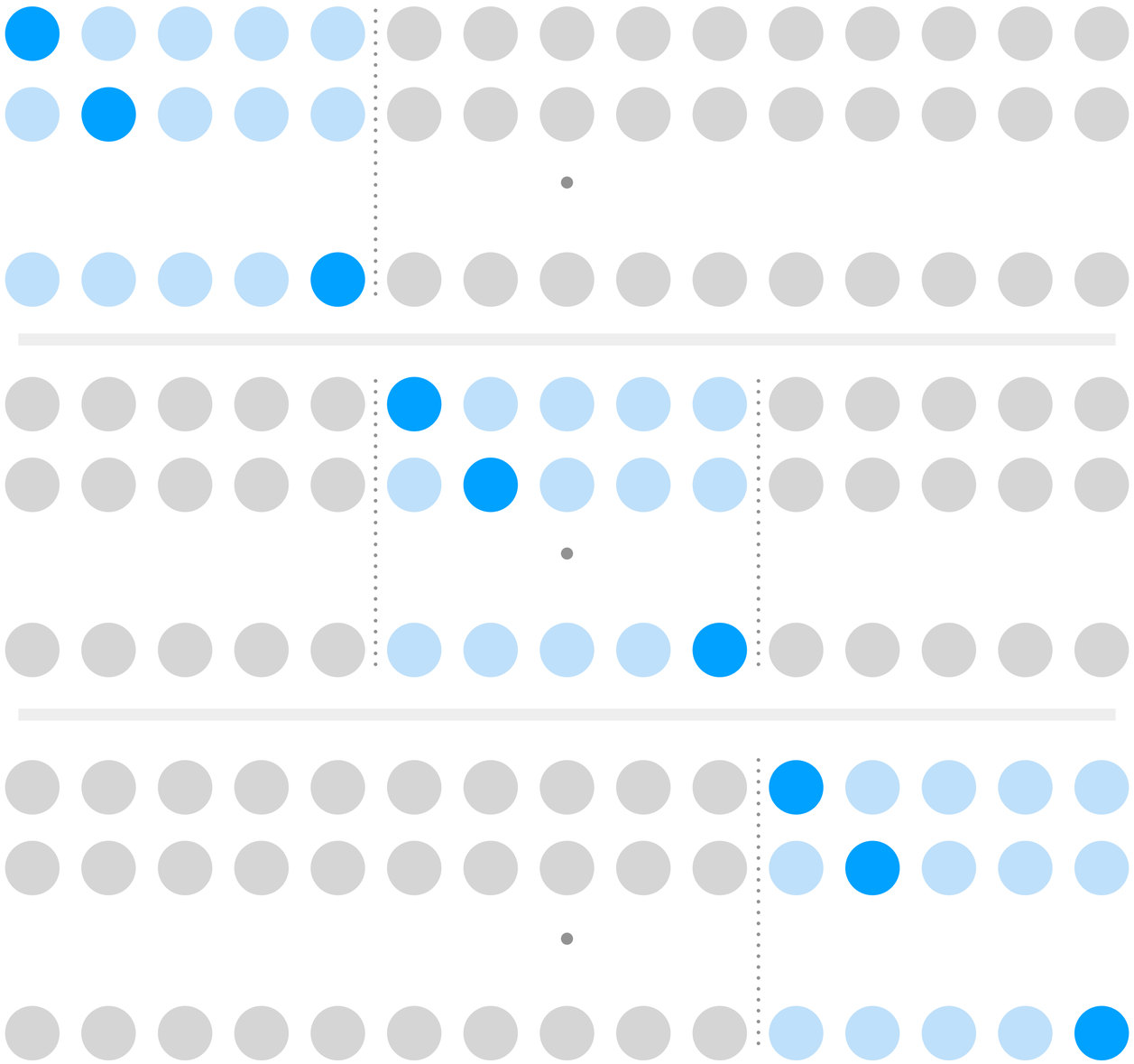
- First attempt to improve on the TCE
- P-values computed in batches
- Relies on unsound assumption

Inductive Conformal Evaluator (ICE)



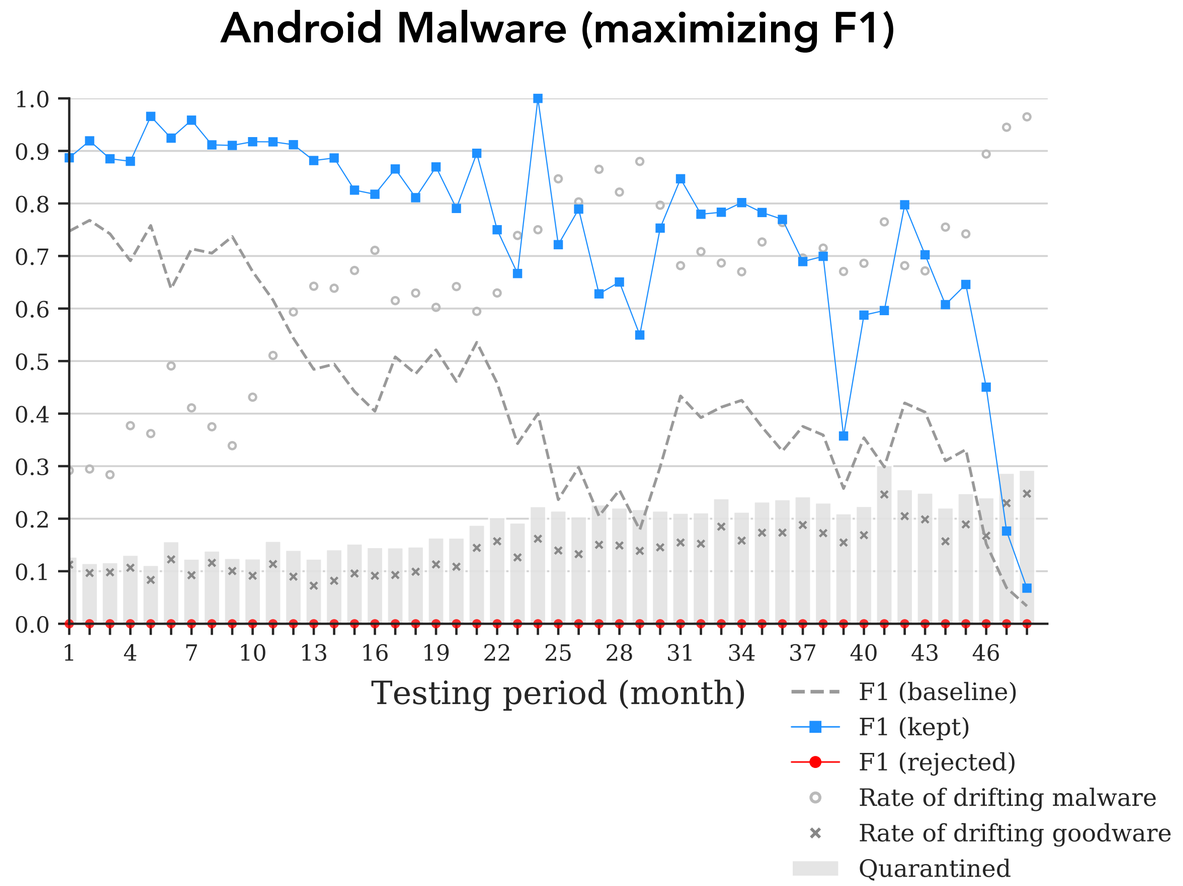
- Increase speed by splitting into training and calibration sets
- Rooted in CP theory
- Computationally efficient
- Informationally inefficient

Cross-Conformal Evaluator (CCE)



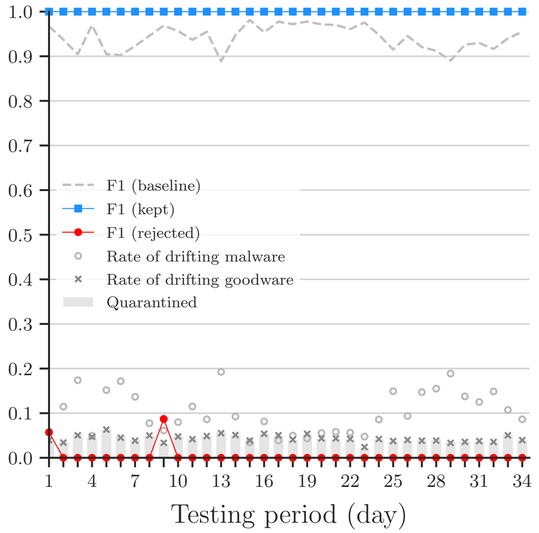
- Inspired by cross validation - multiple ICEs in parallel vote on evaluation
- Rooted in CP theory
- Computationally efficient
- Informationally efficient

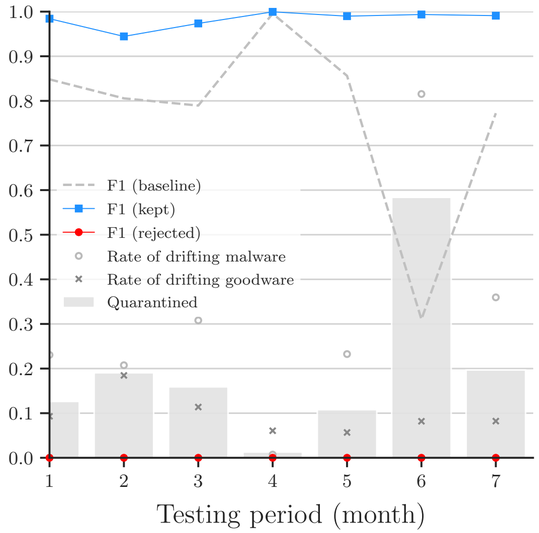
Results: Rejection Performance — Drift Rate

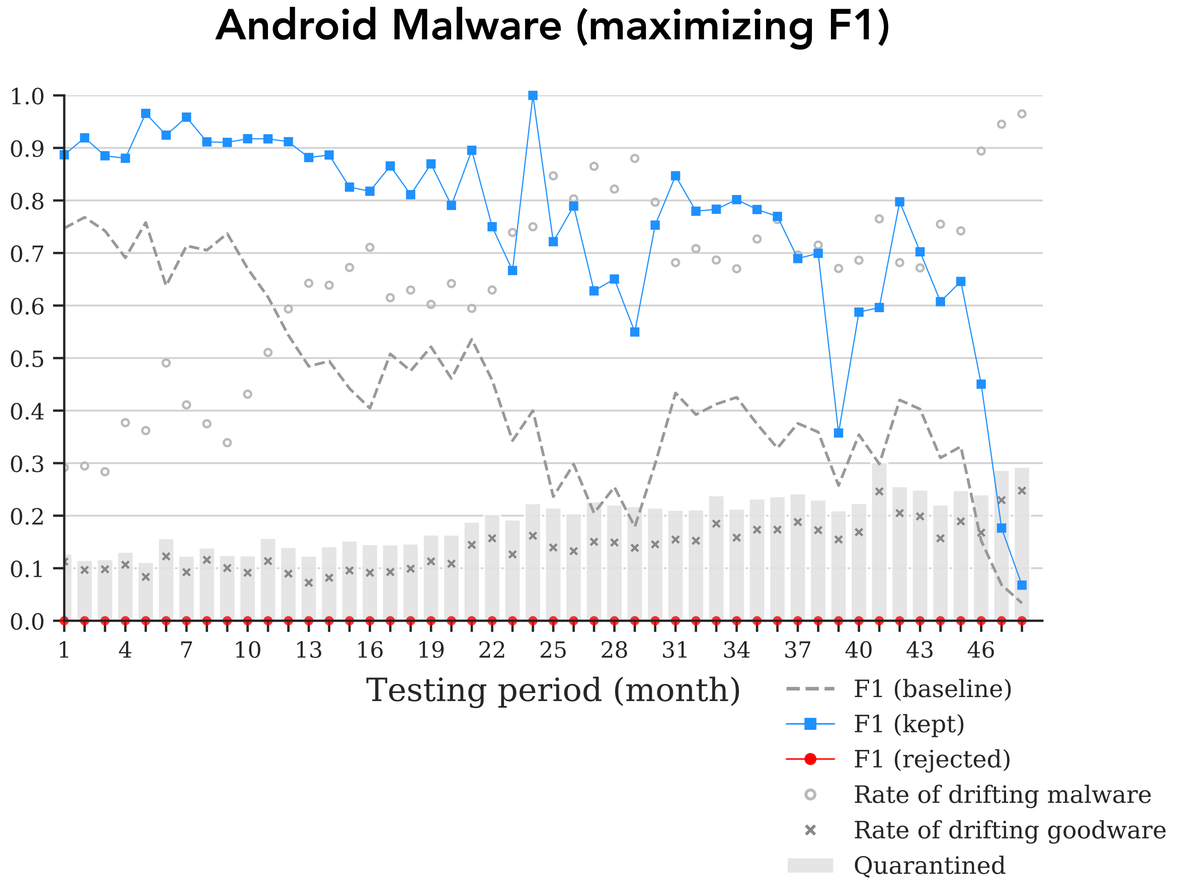


[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

Results: Rejection Performance — Drift Rate



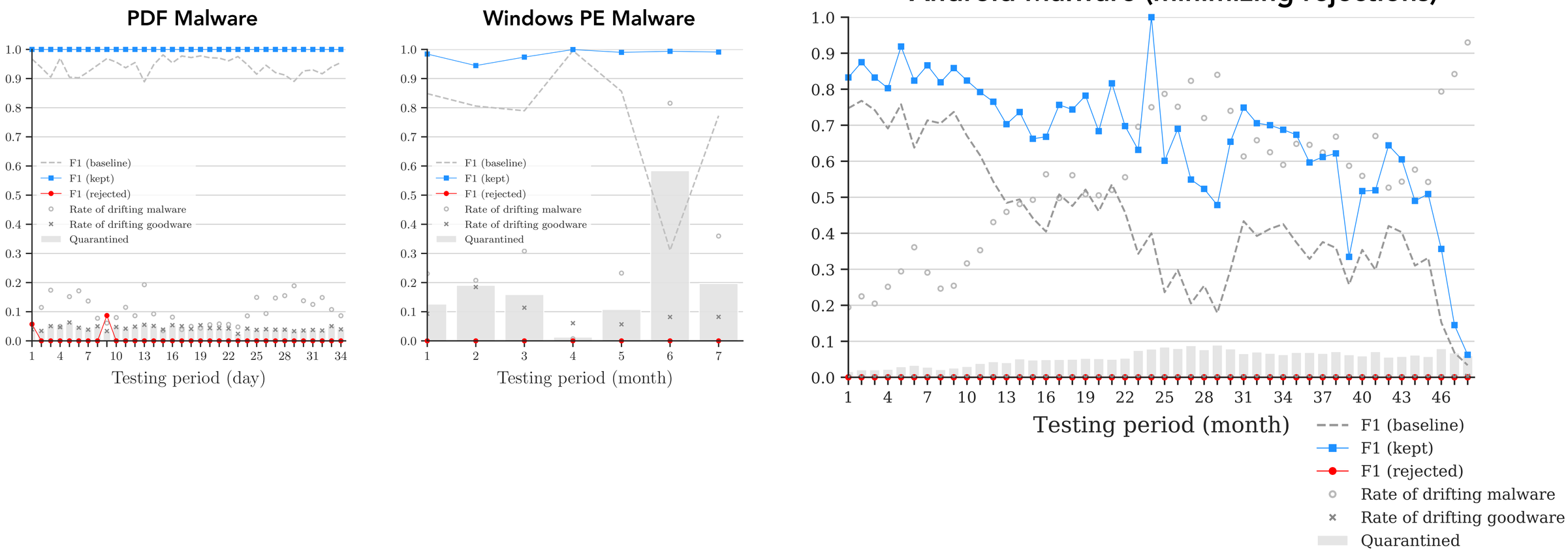




[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift

https://s2lab.cs.ucl.ac.uk/projects/transcend/

Results: Rejection Performance — Drift Rate

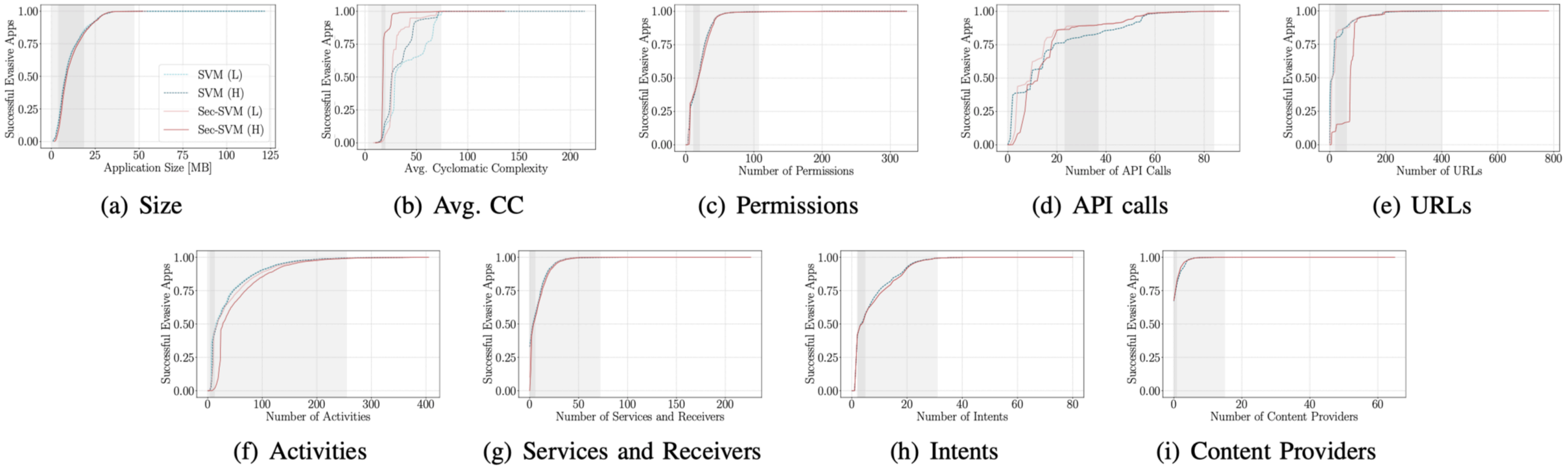


[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift

https://s2lab.cs.ucl.ac.uk/projects/transcend/

Results: How much are app statistics affected?

- Adding all these features (+ side-effect features), what does it do to app statistics?
  - › Limiting feature-space perturbations δ does not affect problem-space attack



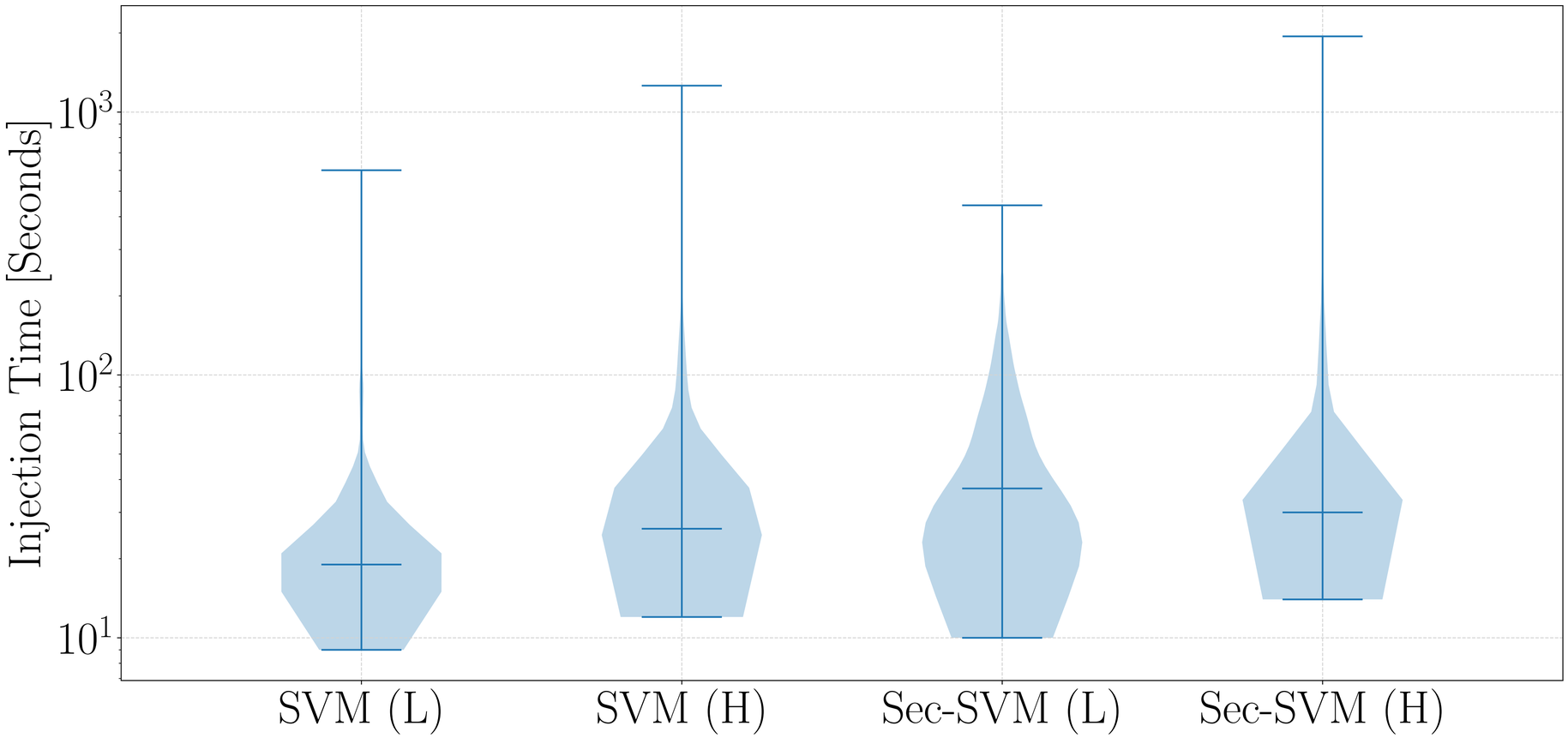
(a) Size (b) Avg. CC (c) Permissions (d) API calls (e) URLs

(f) Activities (g) Services and Receivers (h) Intents (i) Content Providers

[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.kcl.ac.uk/projects/intriguing

Results: How much time does an attack take?

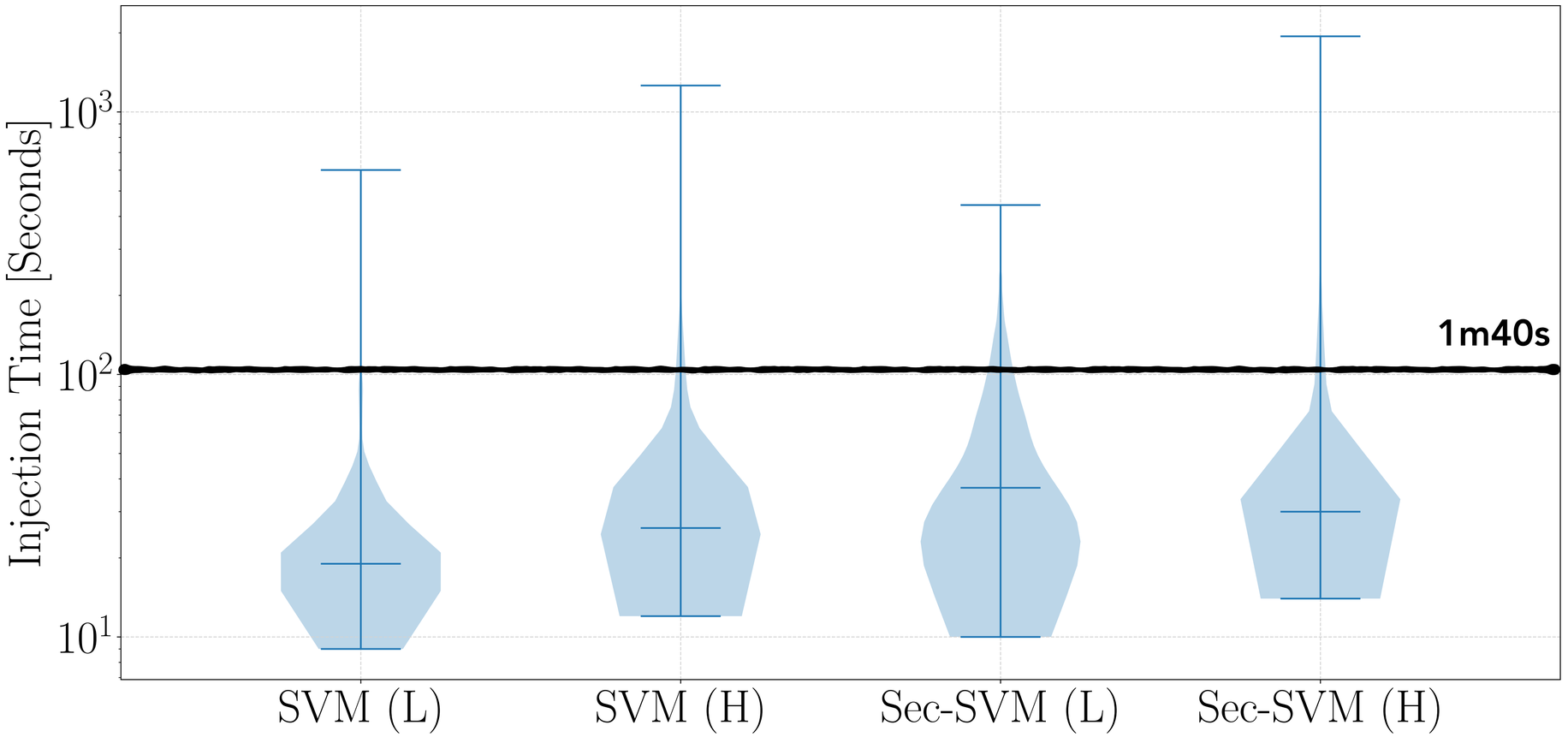
- In most cases, **less than 2 minutes** to create an adversarial example



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.kcl.ac.uk/projects/intriguing

Results: How much time does an attack take?

- In most cases, **less than 2 minutes** to create an adversarial example



[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.kcl.ac.uk/projects/intriguing