# 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

**[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
- **DREBIN** [NDSS’14]: Linear SVM, binary feature space
- **Sec-SVM** [TDSC’17]: Feature-space defense for DREBIN (evenly distributes weights)

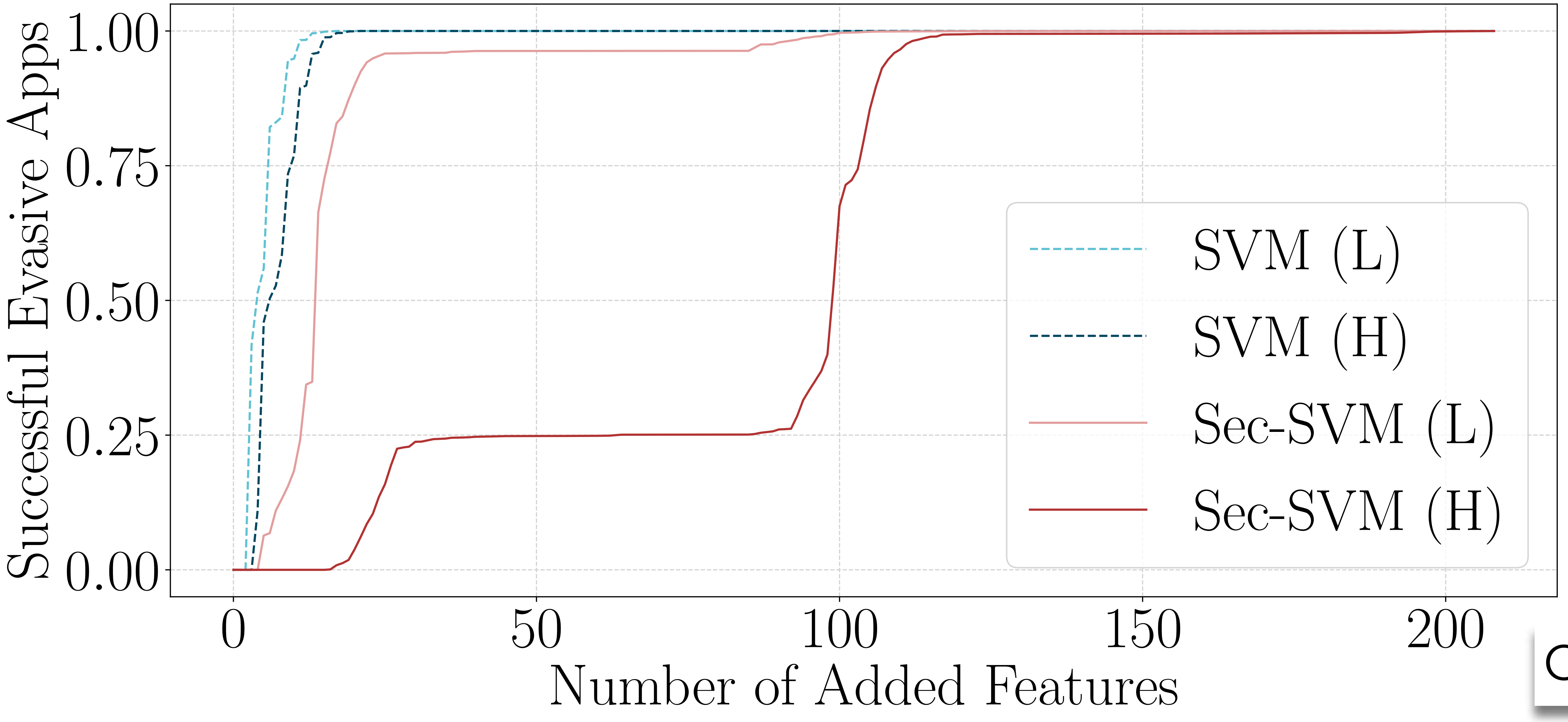
**[IEEE S&P 2020]** Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.cs.ucl.ac.uk/projects/intriguing

# Results: What is the impact on the Feature Space?

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



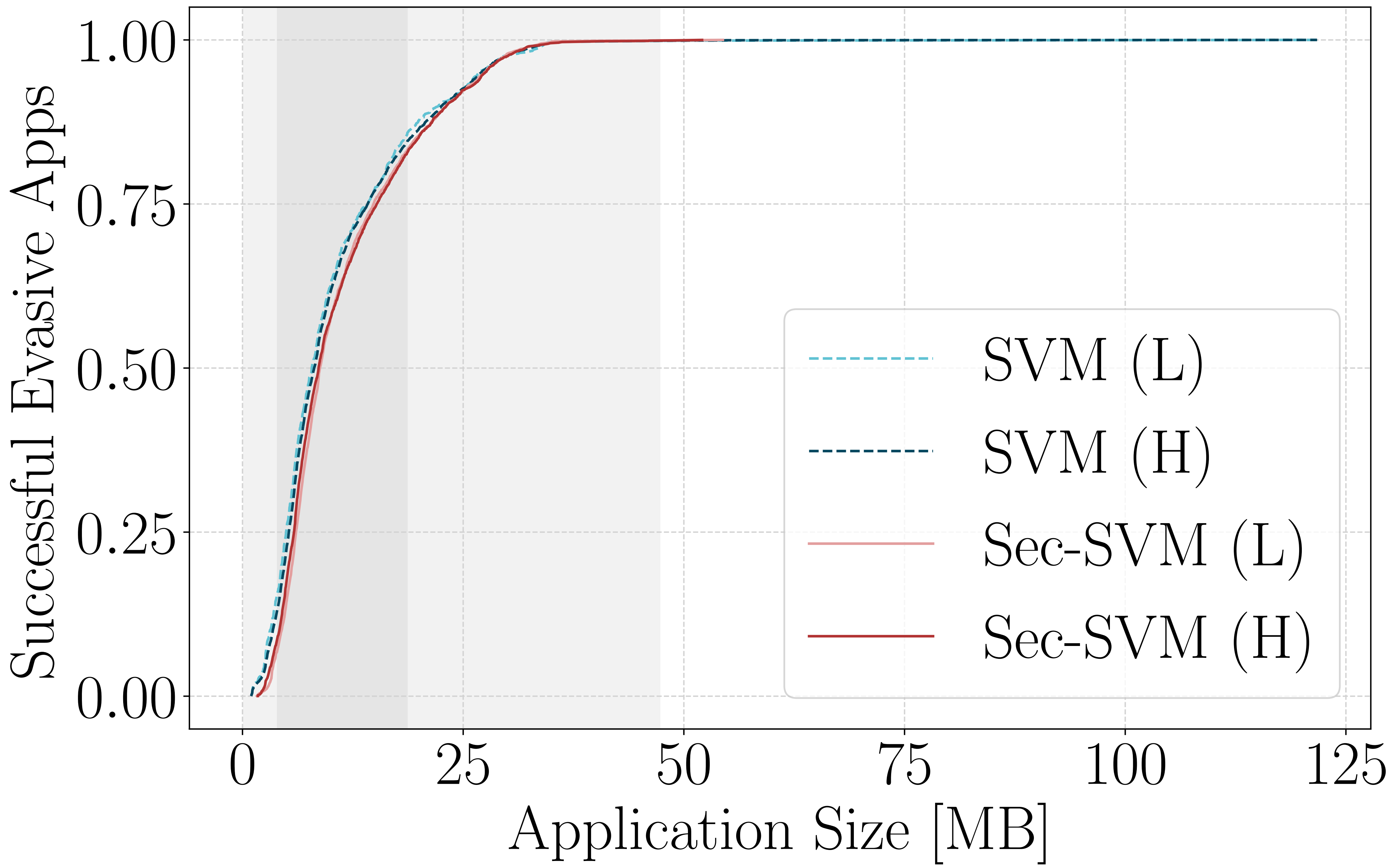

Out of 10K features

**[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

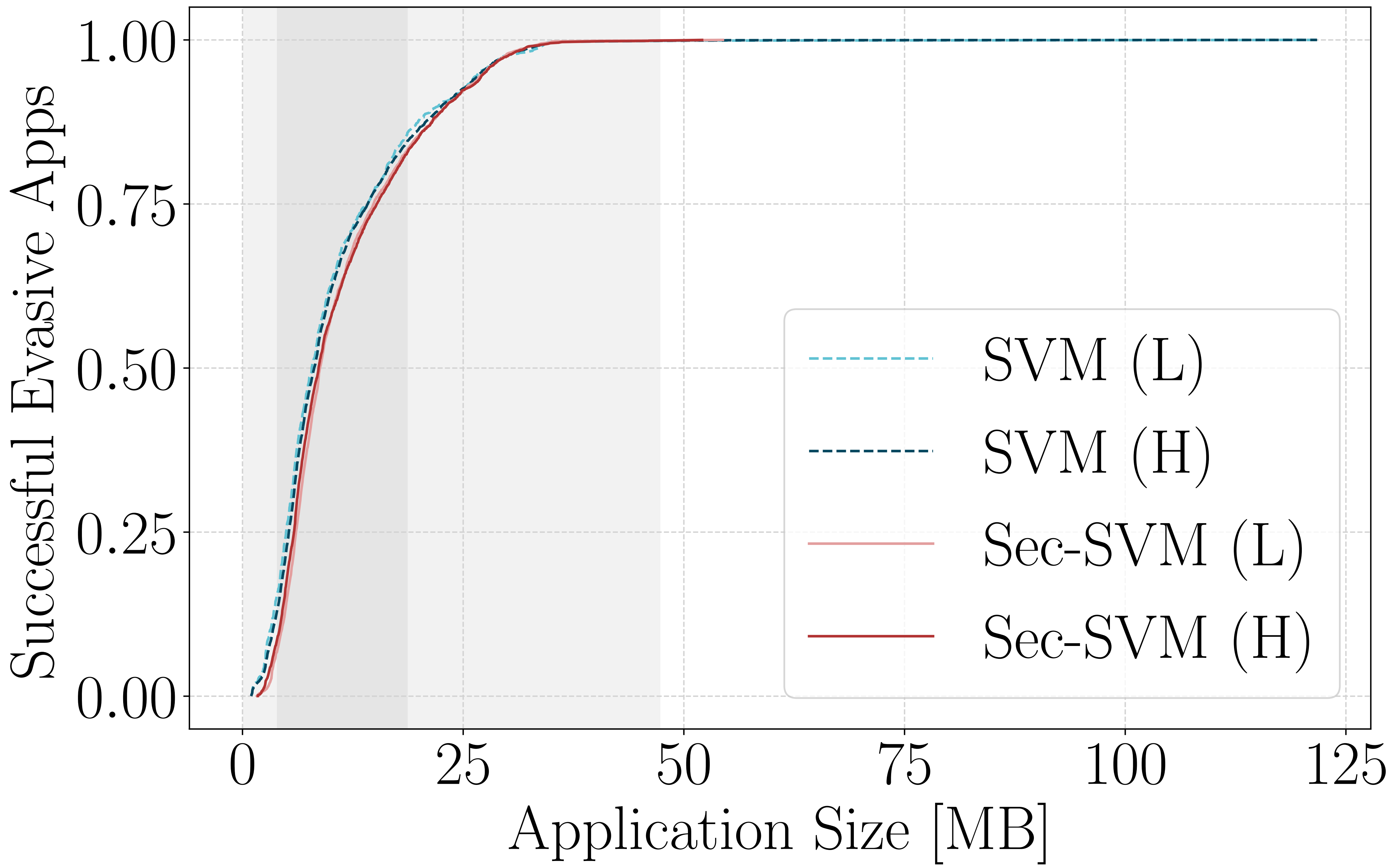
# Results: What is the impact on the Problem Space?




**[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?

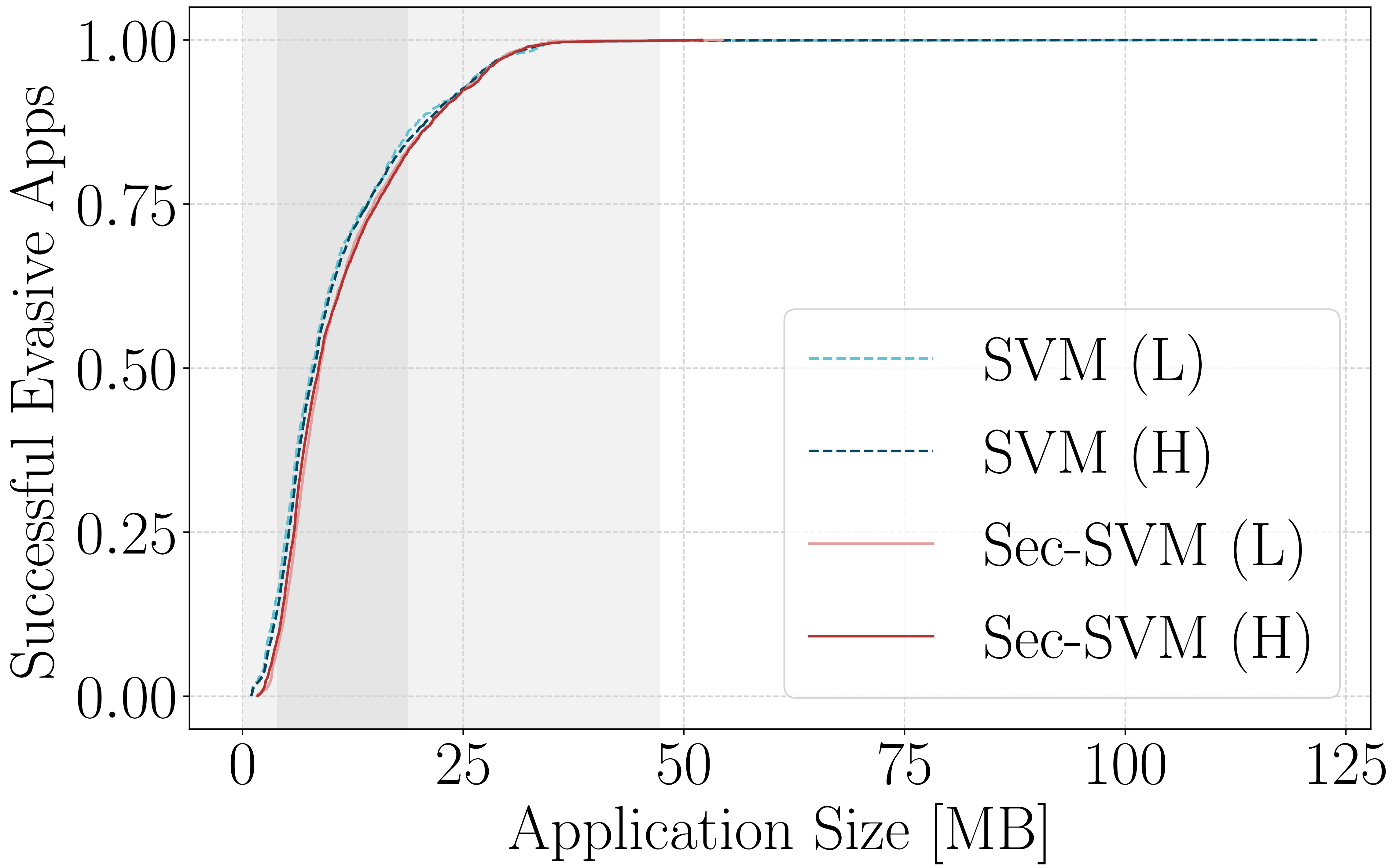
- Adversarial generation < 2 minutes per app




**[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?

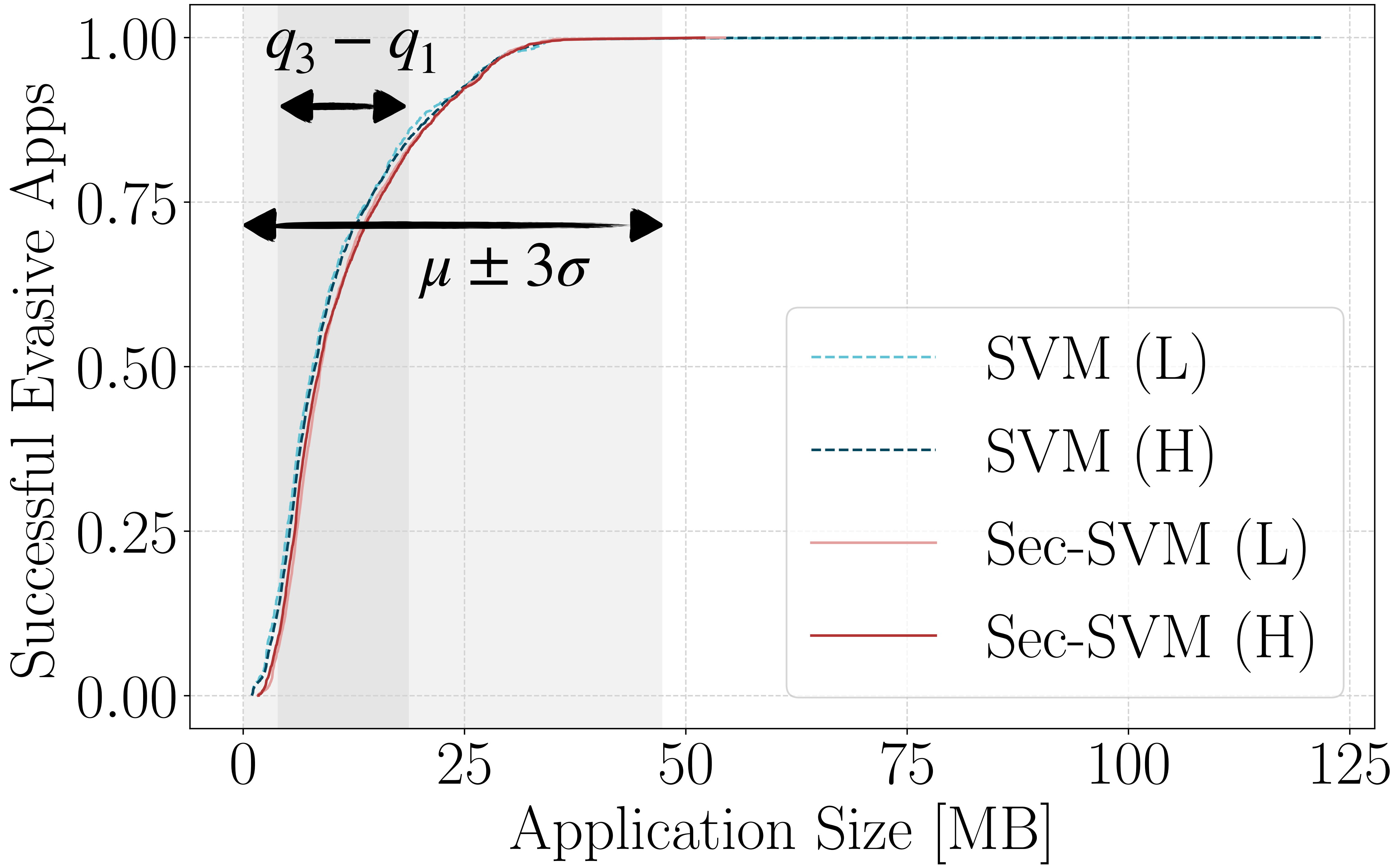
- Adversarial generation < 2 minutes per app
- Restricting feature-space perturbations $\delta$ does not hinder problem-space attack




**[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?

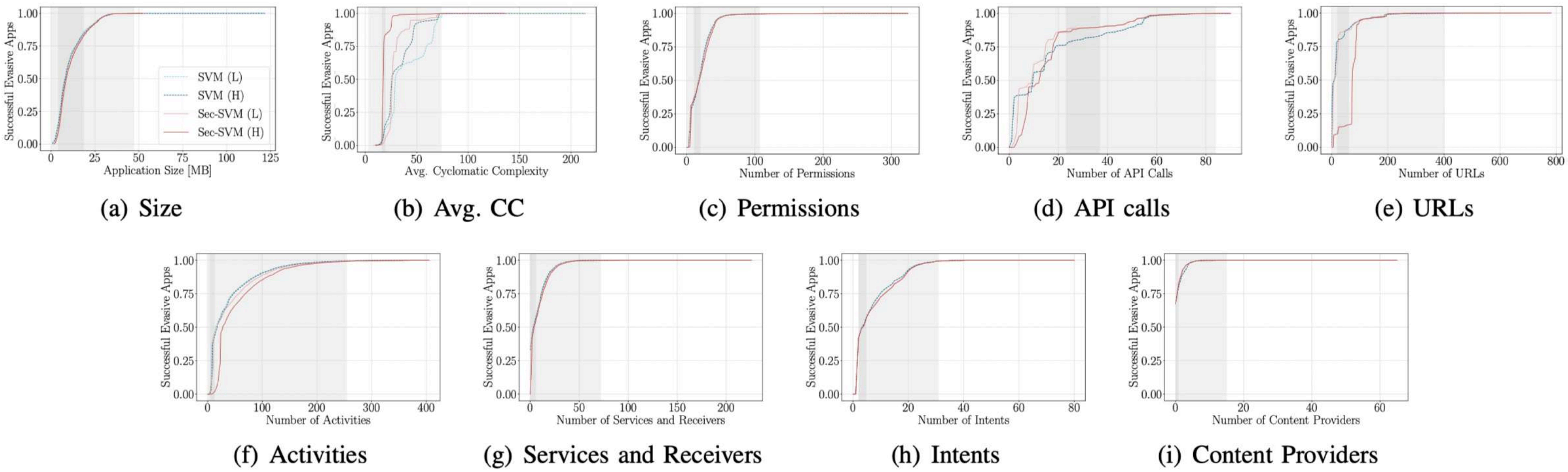
- Adversarial generation < 2 minutes per app
- Restricting feature-space perturbations $\delta$ does not hinder problem-space attack
- App statistics (e.g., size) do not become anomalous after injection




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

# 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 $\delta$ does not affect problem-space attack



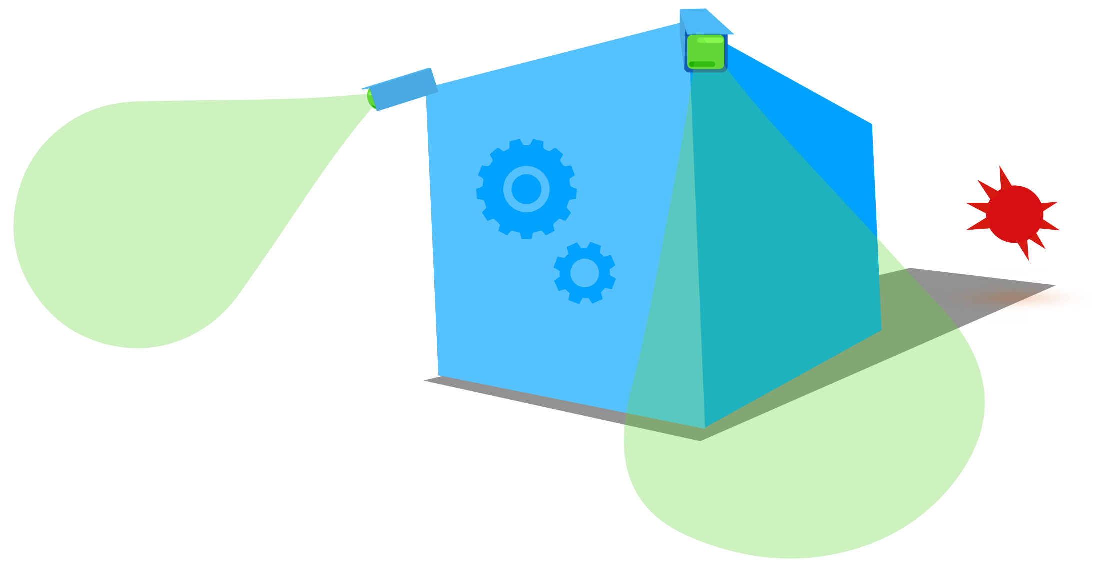

**[IEEE S&P 2020]** Intriguing Properties of Adversarial ML Attacks in the Problem Space
https://s2lab.kcl.ac.uk/projects/intriguing

# 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
- Practical end-to-end automatic adversarial malware as a service — how about defenses?

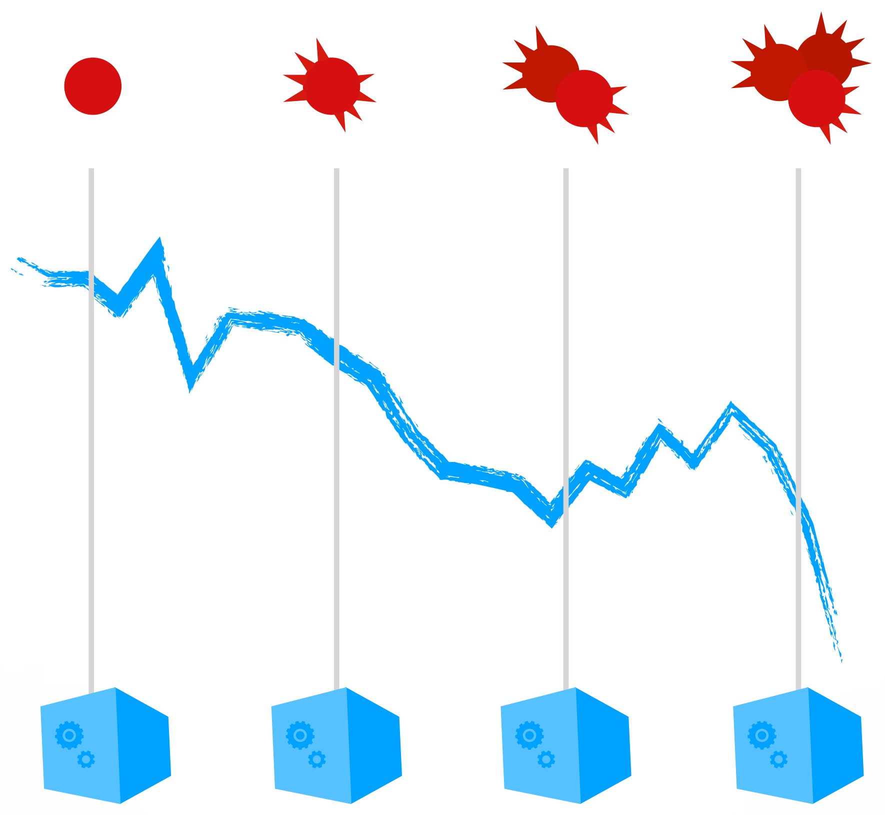


**[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space**

Bigger Picture

## Drifting scenarios caused by threats evolving over time

- How dataset shift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics
- Detecting shifts with abstaining classifiers and classification with rejection



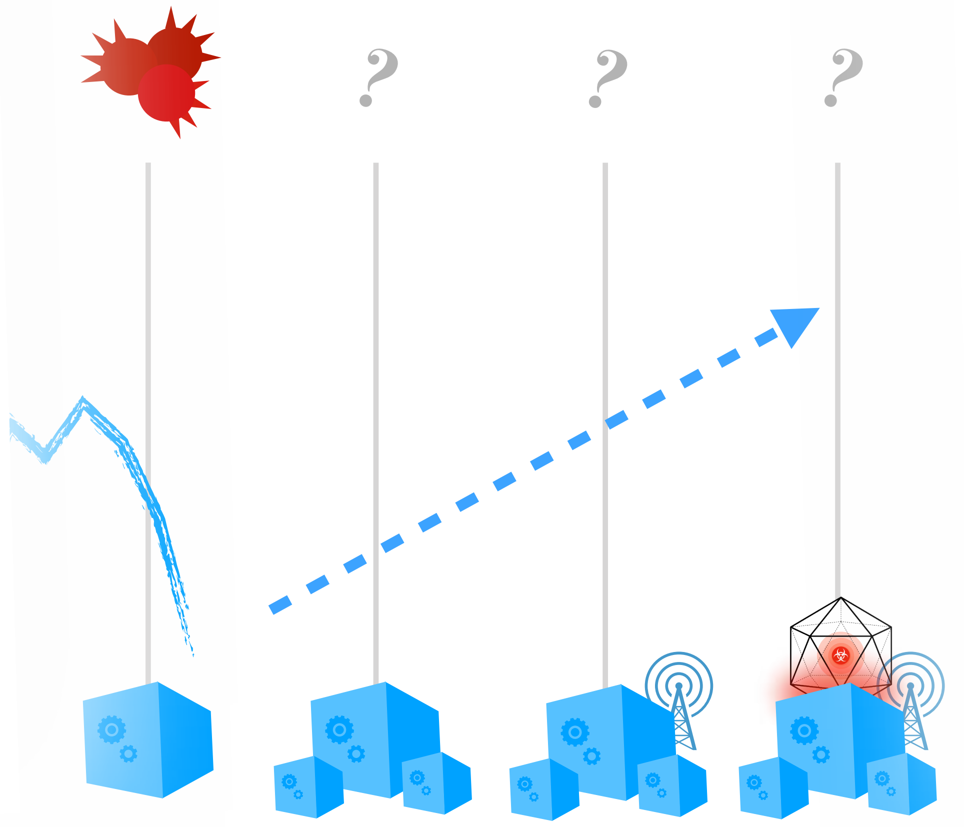
**[USENIX Sec 2017 & IEEE S&P 2022] Transcend: Detecting Concept Drift in Malware Classification Models & Transcending Transcend: Revisiting Malware Classification in the Presence of Concept Drift**

**[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time**

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



**[USENIX Sec 2022] Dos and Don'ts of Machine Learning in Com**

# 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
- Practical end-to-end automatic adversarial malware as a service — how about defenses?

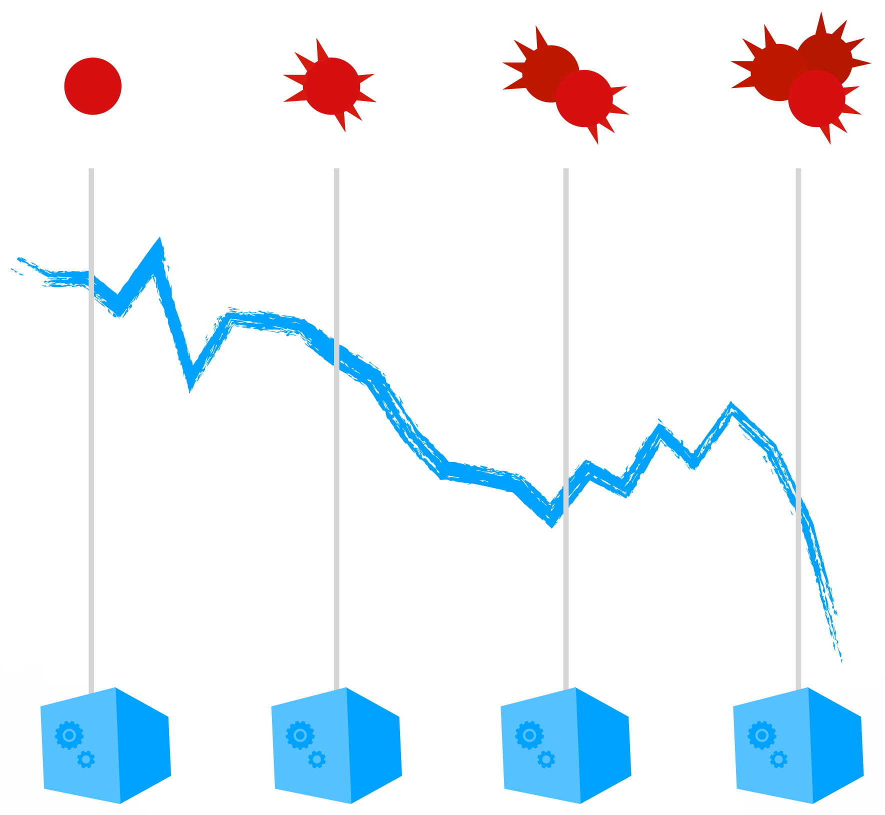


[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space

Bigger Picture

## Drifting scenarios caused by threats evolving over time

- How dataset shift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics
- Detecting shifts with abstaining classifiers and classification with rejection



**[USENIX Sec 2017 & IEEE S&P 2022] Transcend: Detecting Concept Drift in Malware Classification Models & Transcending Transcend: Revisiting Malware Classification in the Presence of Concept Drift**

**[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time**

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

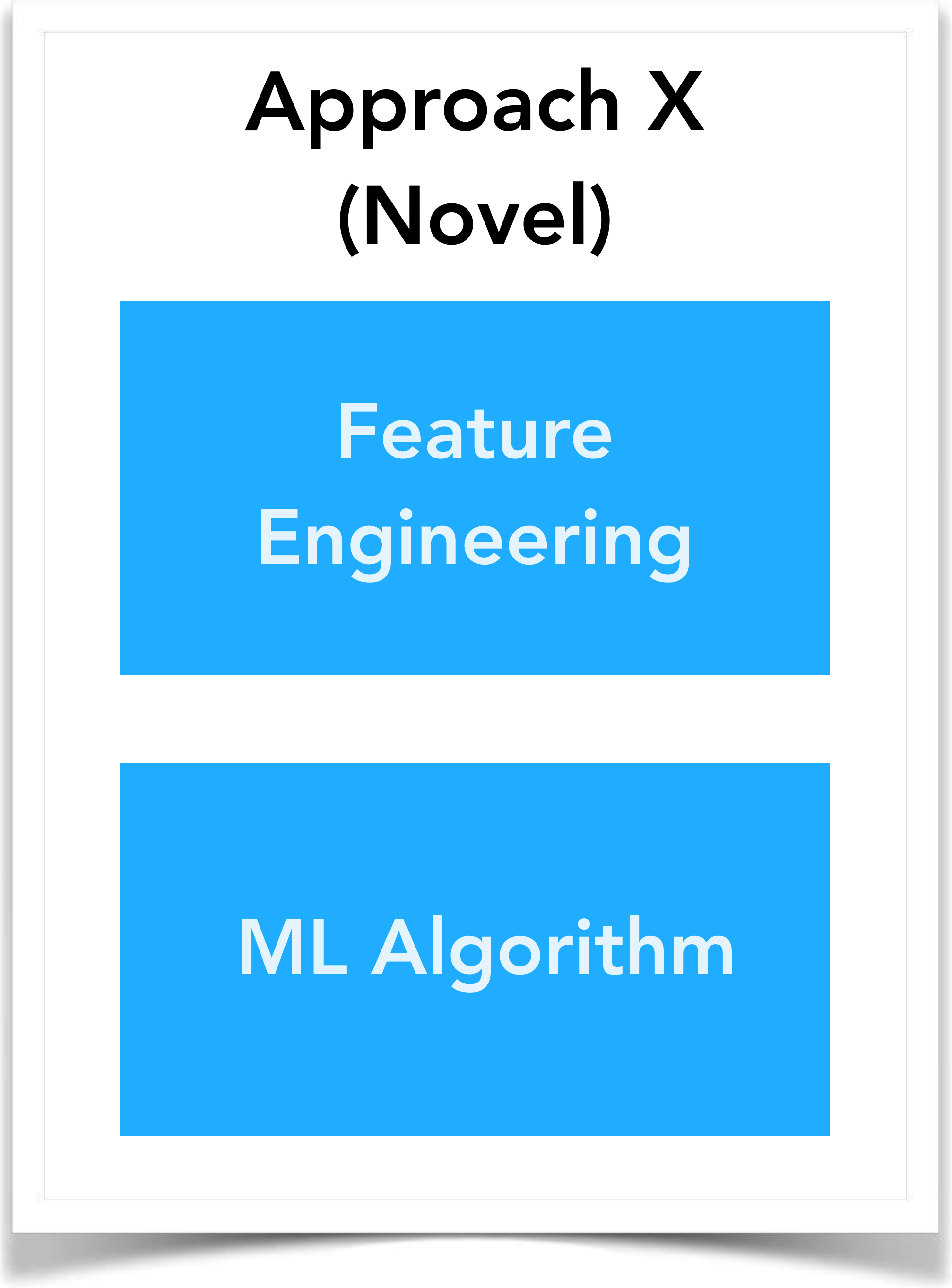


[USENIX Sec 2022] Dos and Don'ts of Machine Learning in Com

# ML for Malware Detection

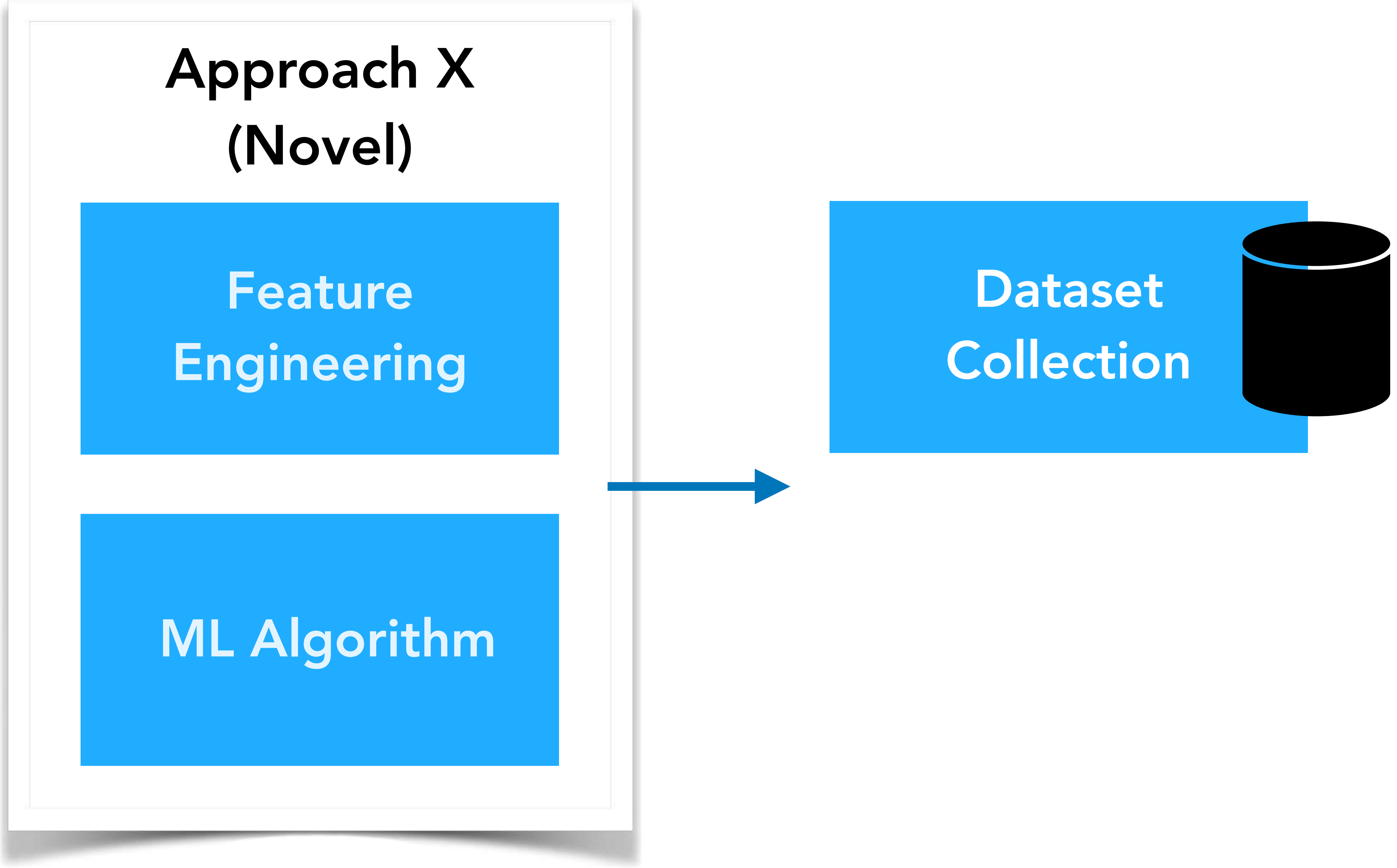
**[USENIX Sec 2019]** **TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time**
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# ML for Malware Detection



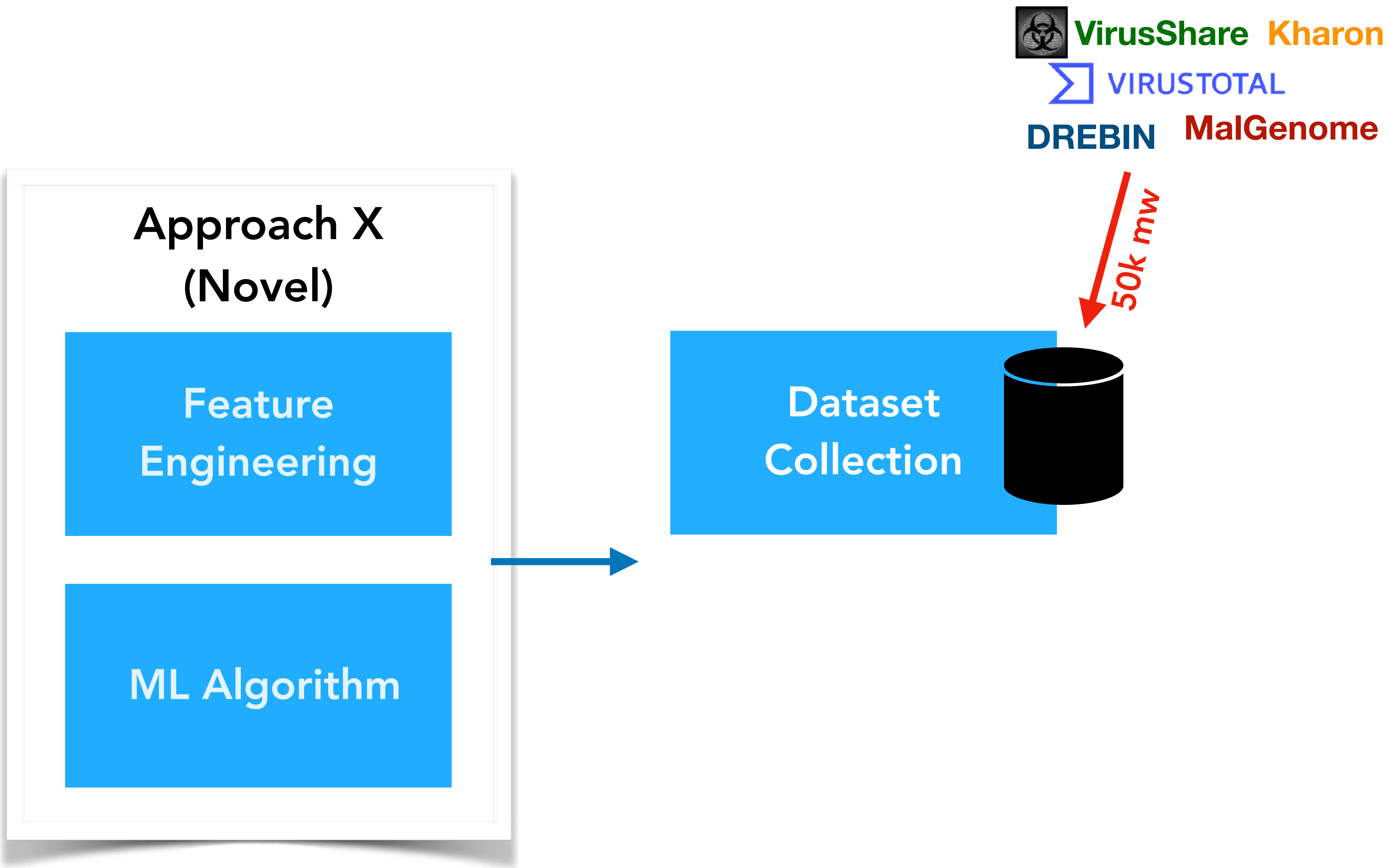

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# ML for Malware Detection



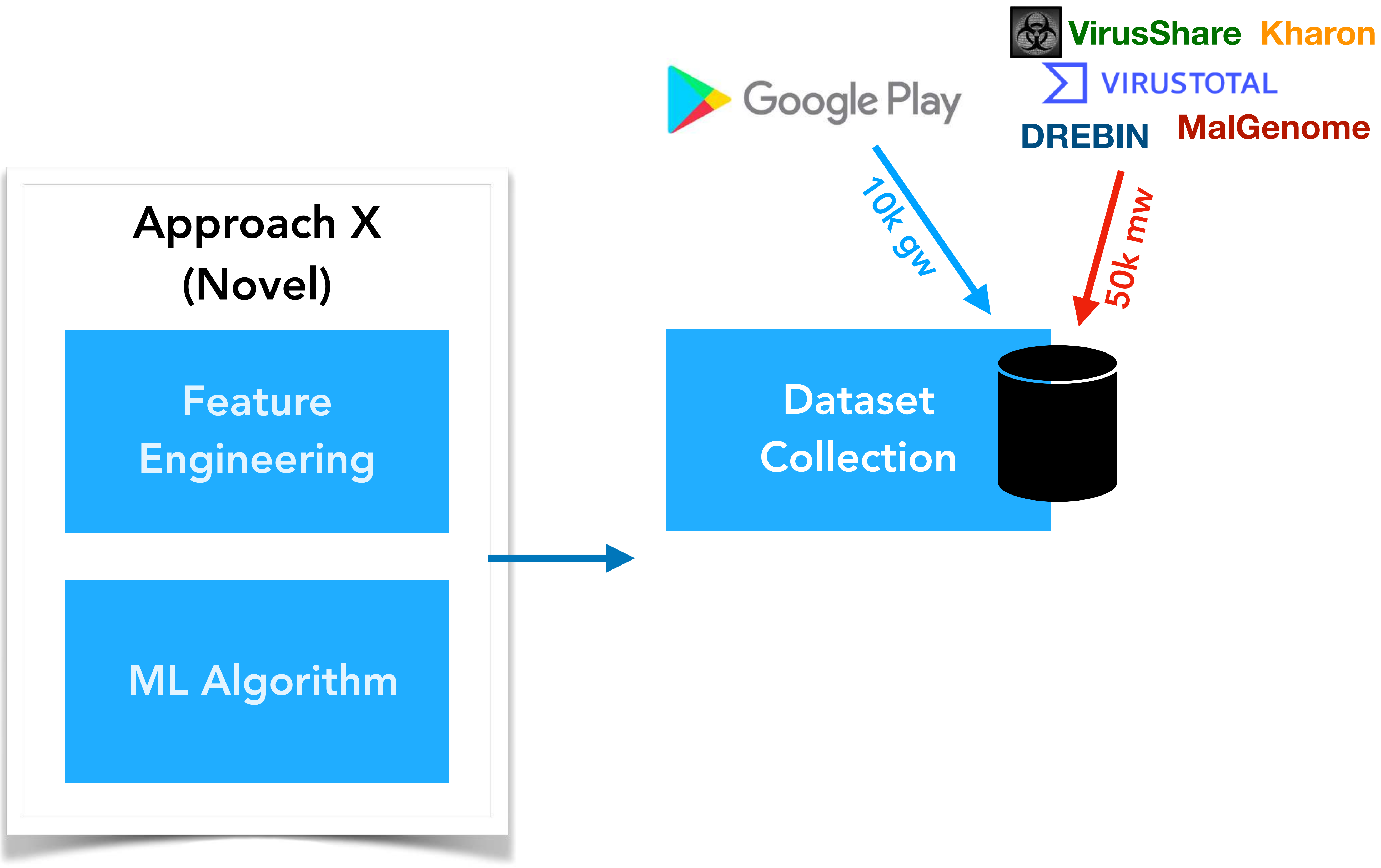

**[USENIX Sec 2019]** **TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time**
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# ML for Malware Detection



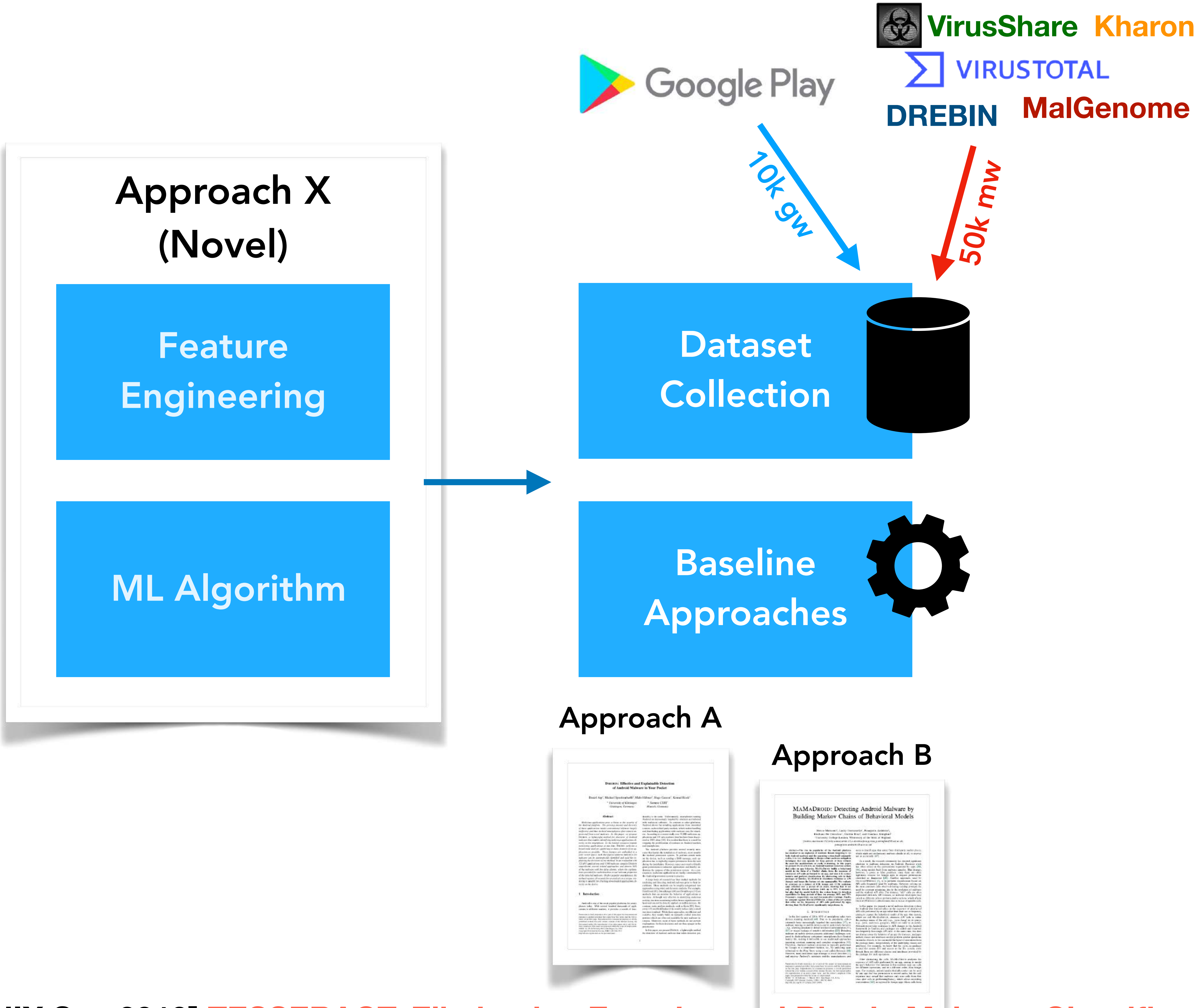

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# ML for Malware Detection



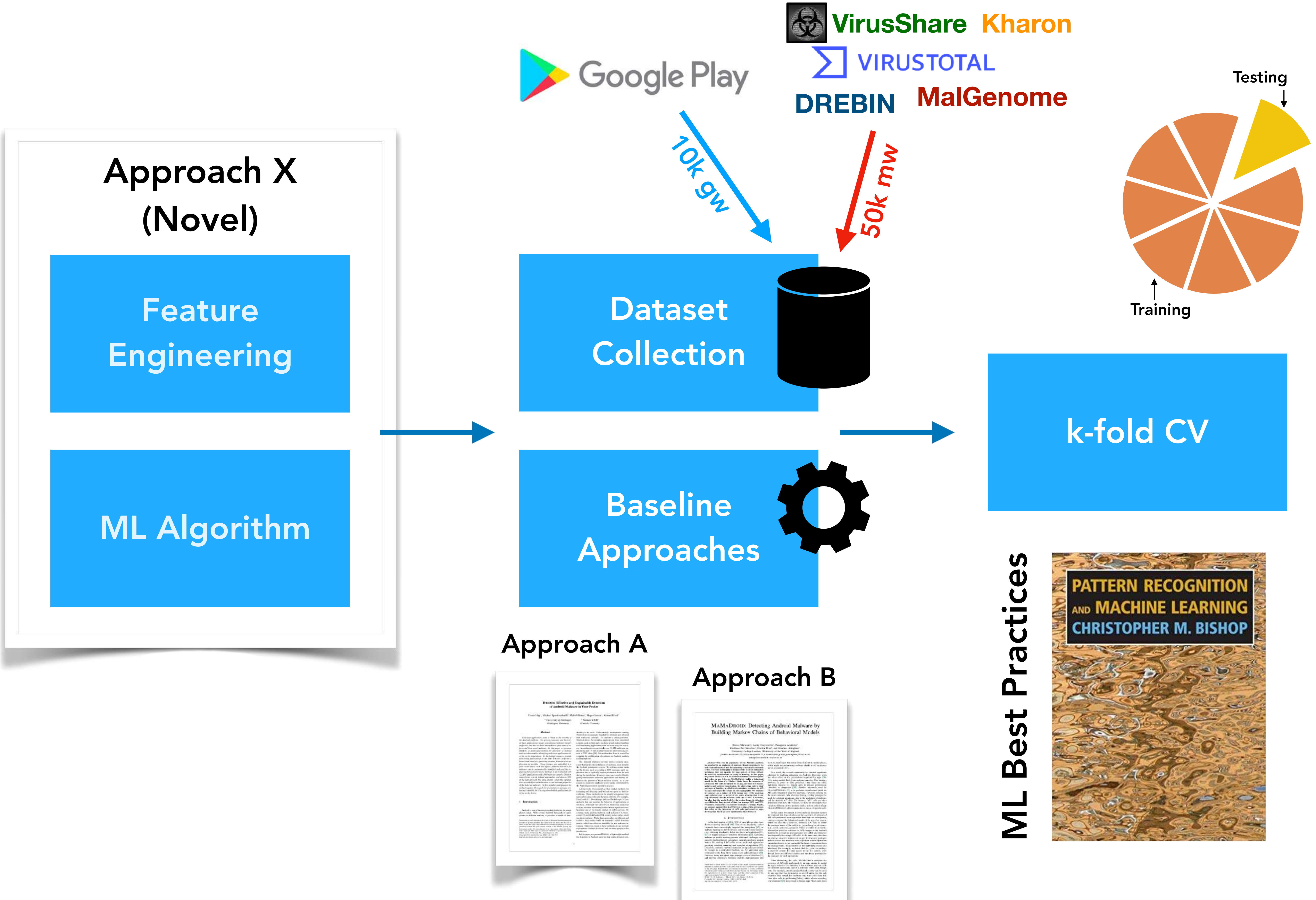

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# ML for Malware Detection




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

# ML for Malware Detection




**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time

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

# ML for Malware Detection

Approach X (Novel)

Feature Engineering

ML Algorithm

Google Play

VirusShare Kharon

VIRUSTOTAL

DREBIN MalGenome

10k gw

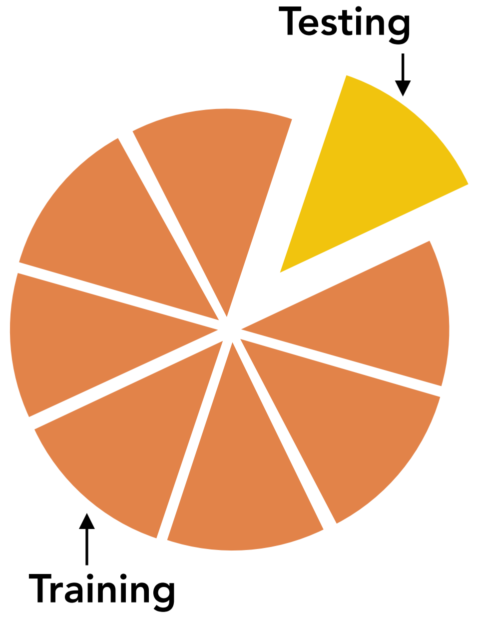
50k mw

Dataset Collection

Baseline Approaches

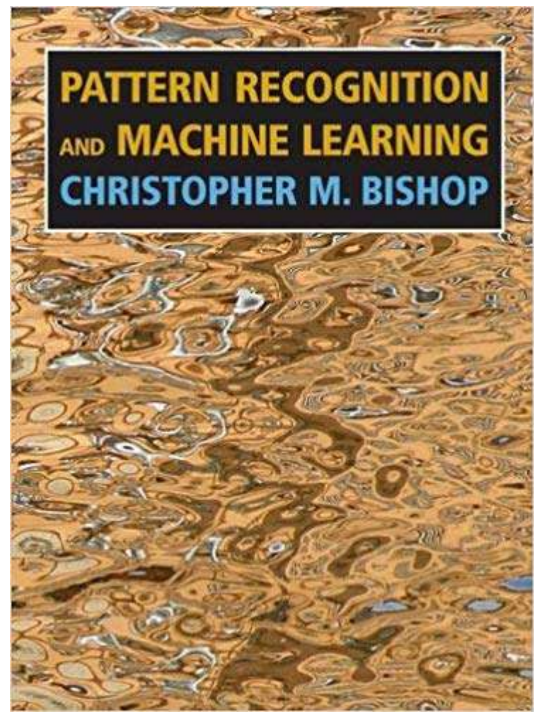
Approach A

Approach B




k-fold CV

ML Best Practices




Compare Results

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

[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time

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

# ML for Malware Detection

Three major unrealistic assumptions

Approach X (Novel)

Feature Engineering

ML Algorithm

Google Play

VirusShare Kharon

VIRUSTOTAL

DREBIN MalGenome

10k gw

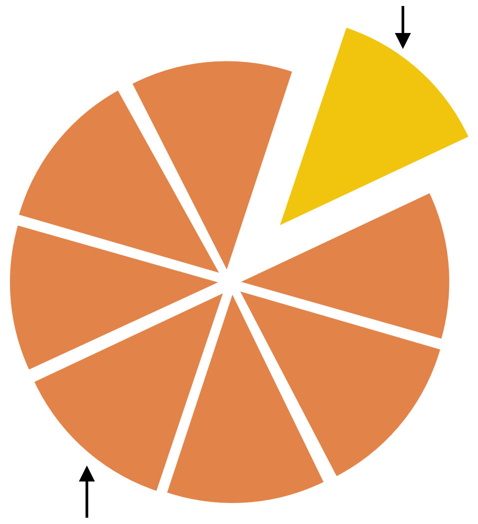
50k mw

Dataset Collection

Baseline Approaches

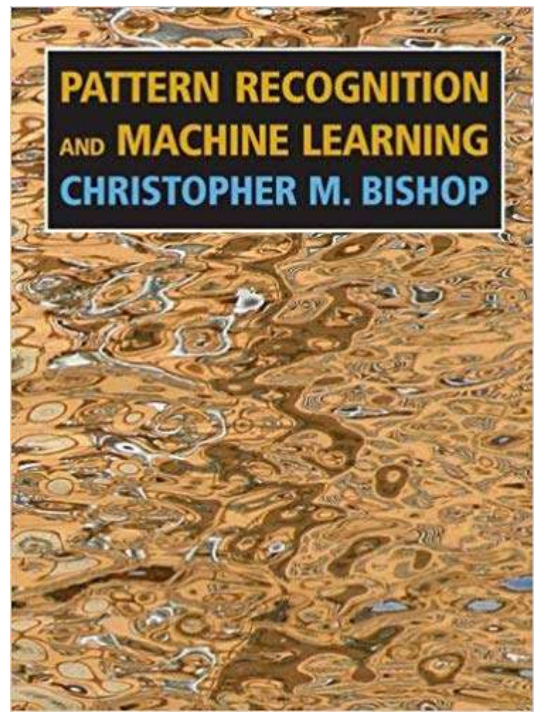
Approach A

Approach B




k-fold CV

ML Best Practices

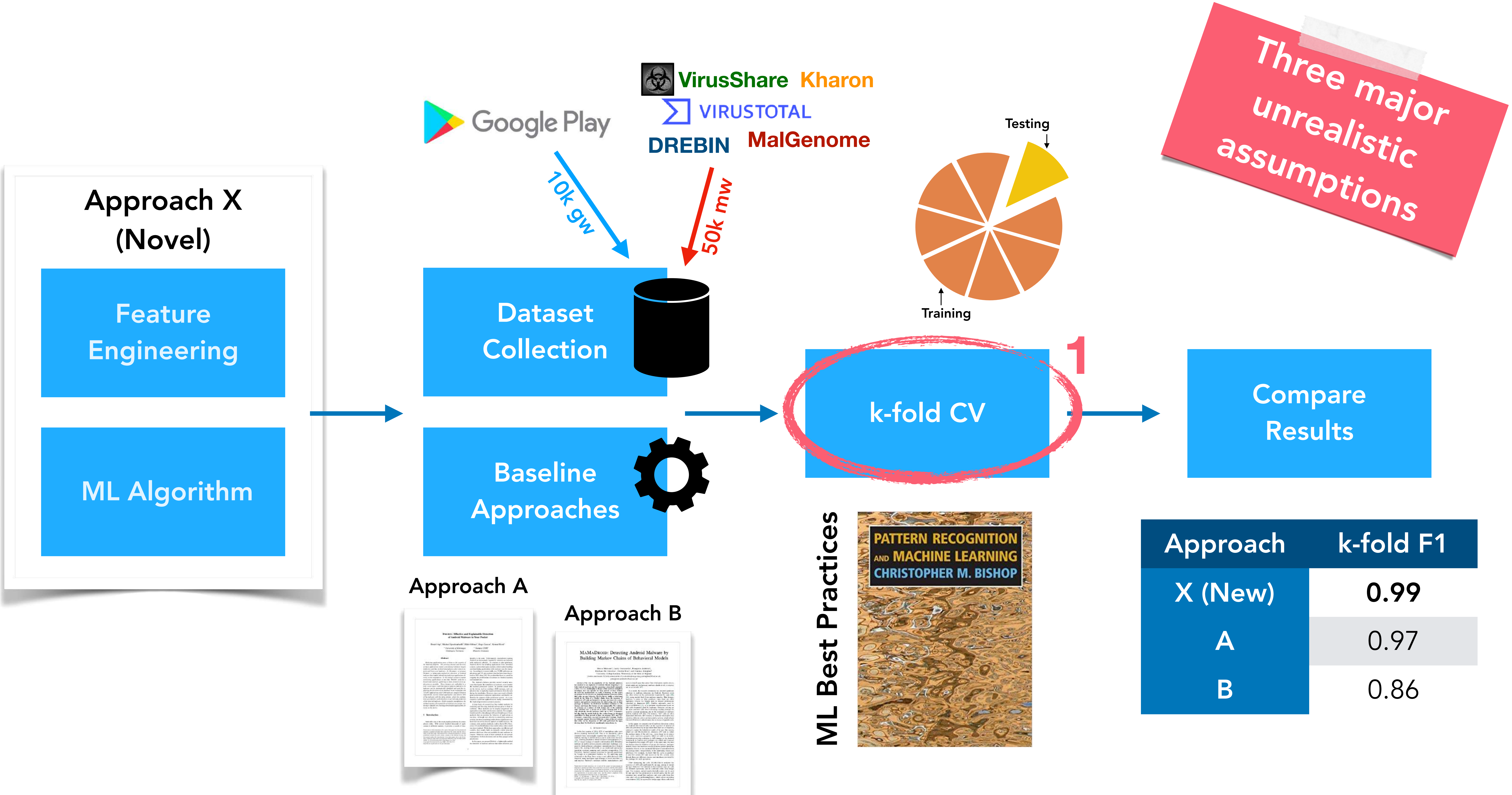



Compare Results

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

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

# ML for Malware Detection




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

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

# ML for Malware Detection

Three major unrealistic assumptions

Approach X (Novel)

Feature Engineering

ML Algorithm

2,3

Google Play

VirusShare Kharon

VIRUSTOTAL

DREBIN MalGenome

10k gw

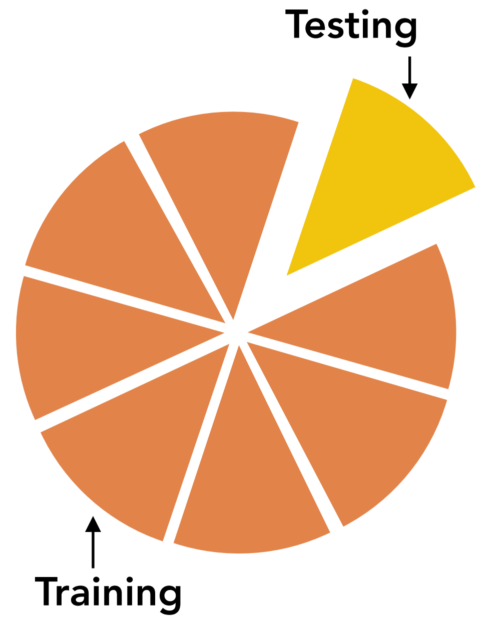
50k mw

Dataset Collection

Baseline Approaches

Approach A

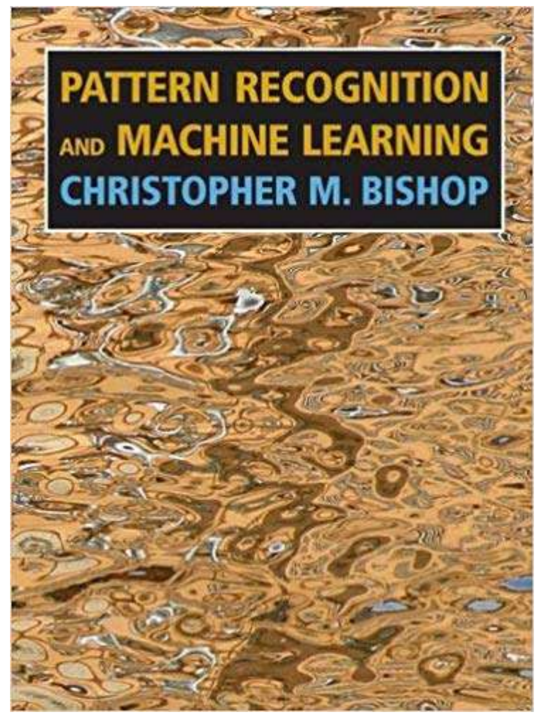
Approach B




1

k-fold CV

ML Best Practices




Compare Results

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

**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time

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

# Sources of Experimental Bias (1/3)

## Temporal Inconsistency in Train/Test Sets

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

# Sources of Experimental Bias (1/3)

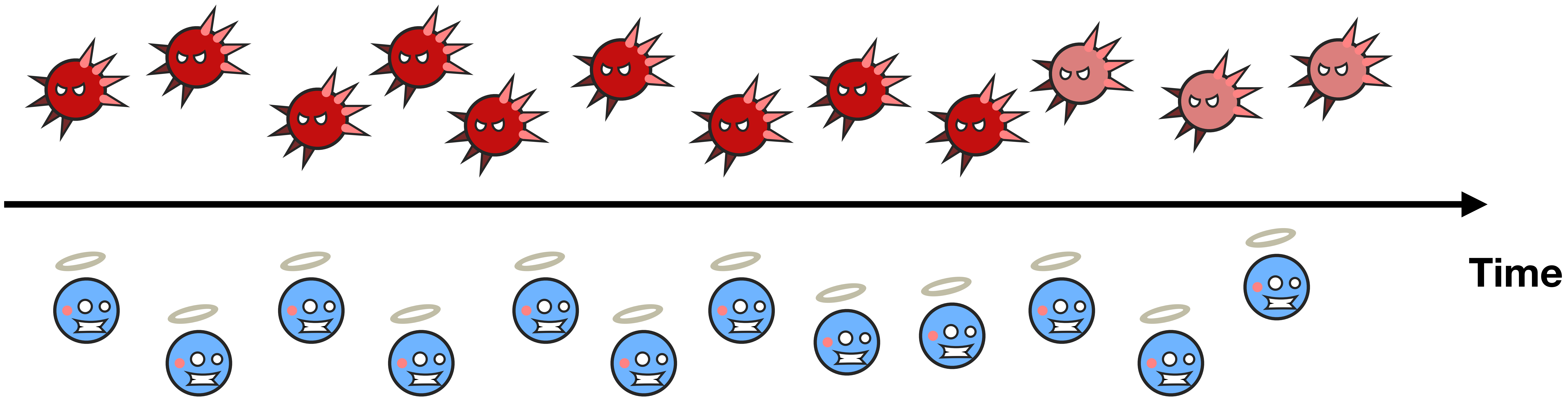
## Temporal Inconsistency in Train/Test Sets

Time

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

# Sources of Experimental Bias (1/3)

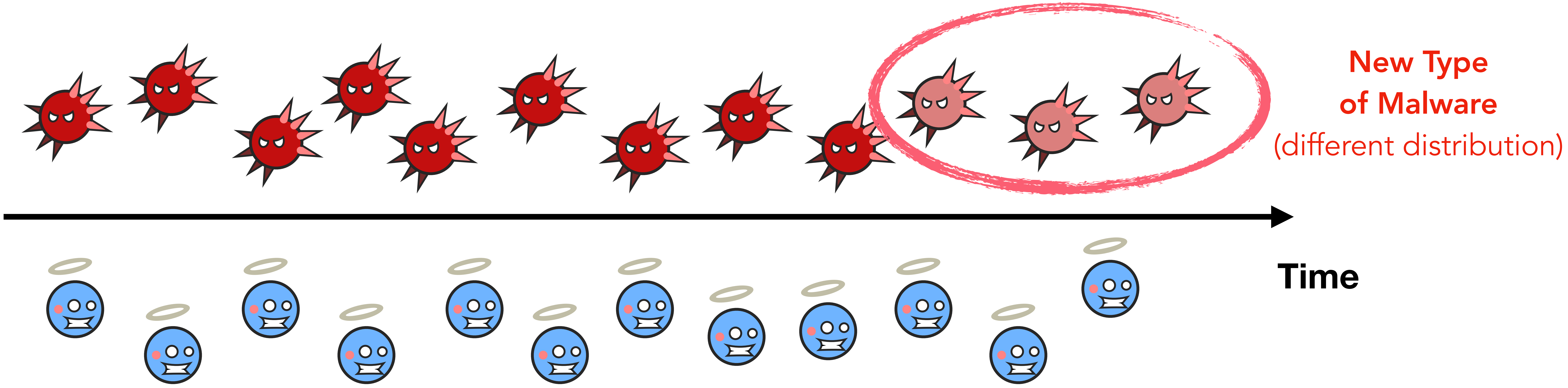
## Temporal Inconsistency in Train/Test Sets




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

# Sources of Experimental Bias (1/3)

## Temporal Inconsistency in Train/Test Sets

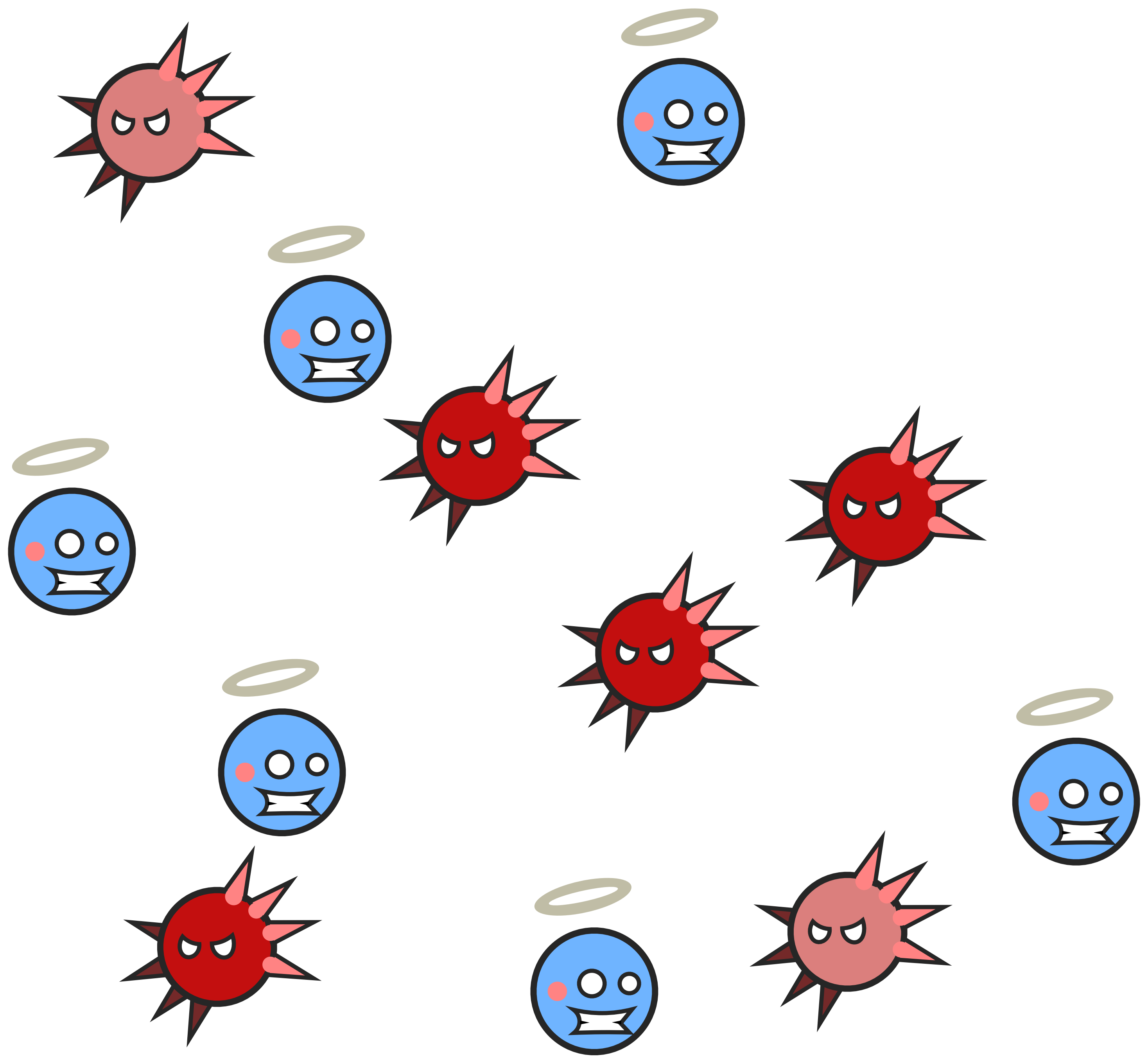



[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

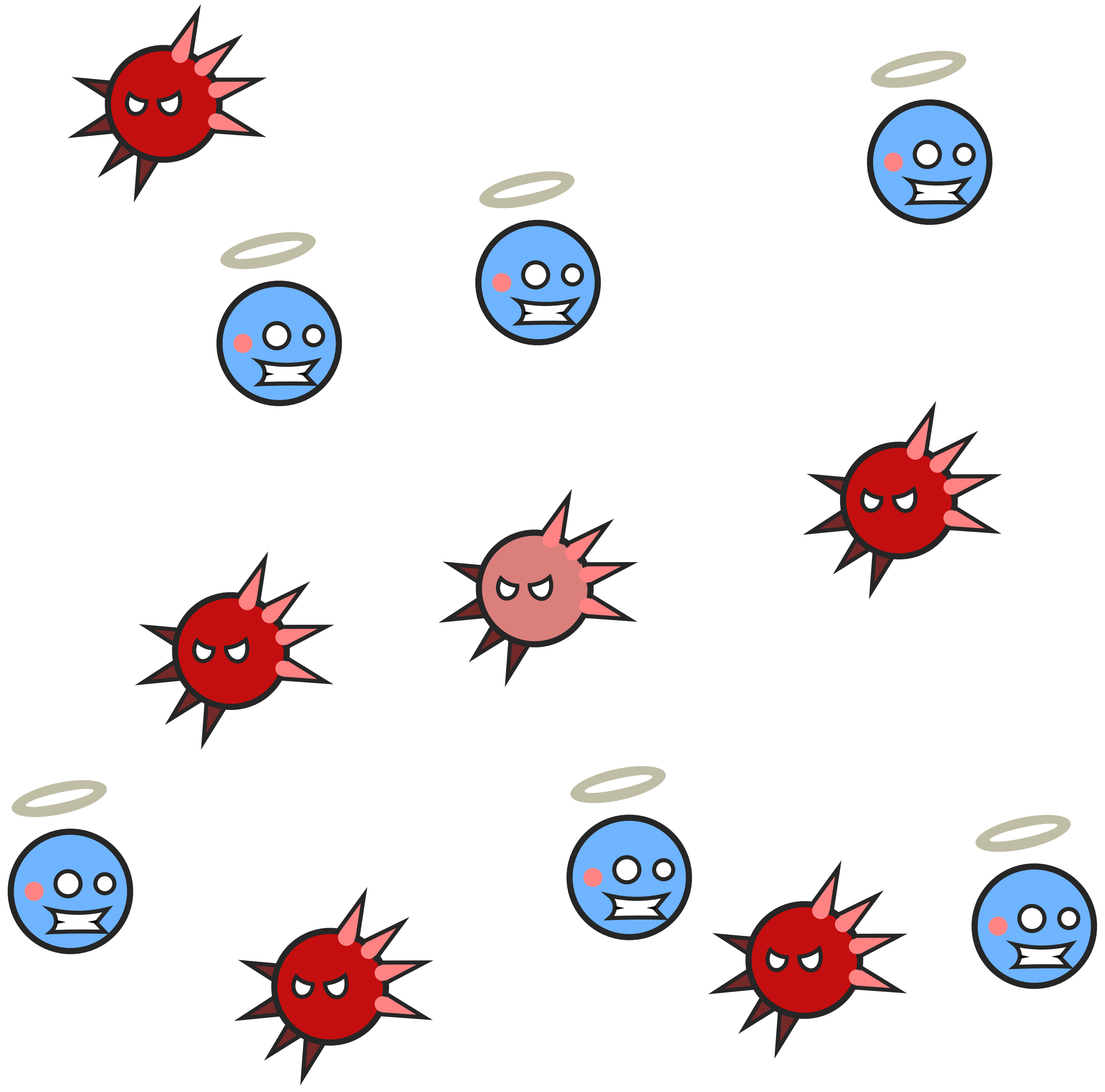
# Sources of Experimental Bias (1/3)

## Temporal Inconsistency in Train/Test Sets

Training



Testing



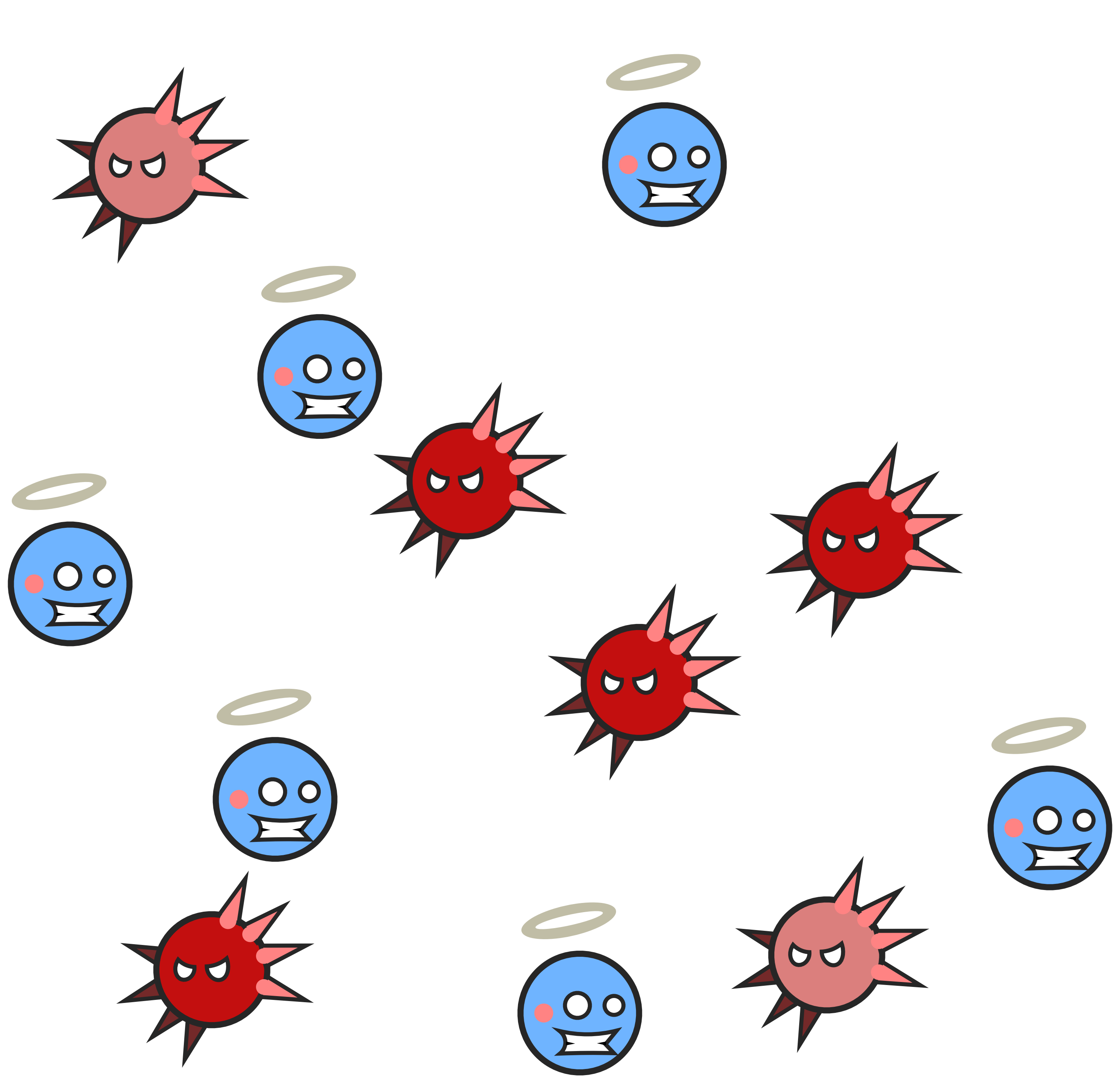
**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# Sources of Experimental Bias (1/3)

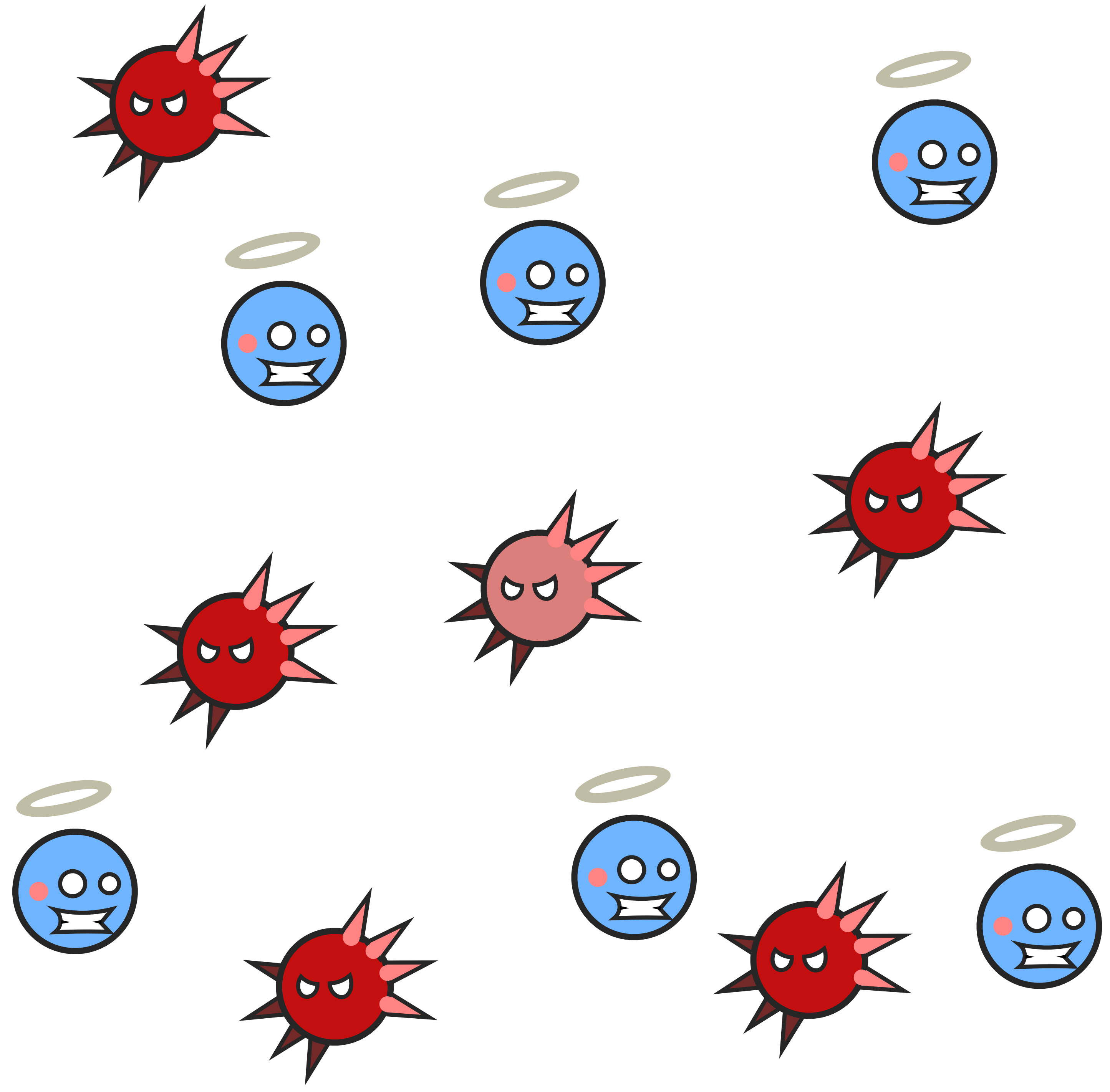
## Temporal Inconsistency in Train/Test Sets

**Violations** use future knowledge in training

**Training**



**Testing**



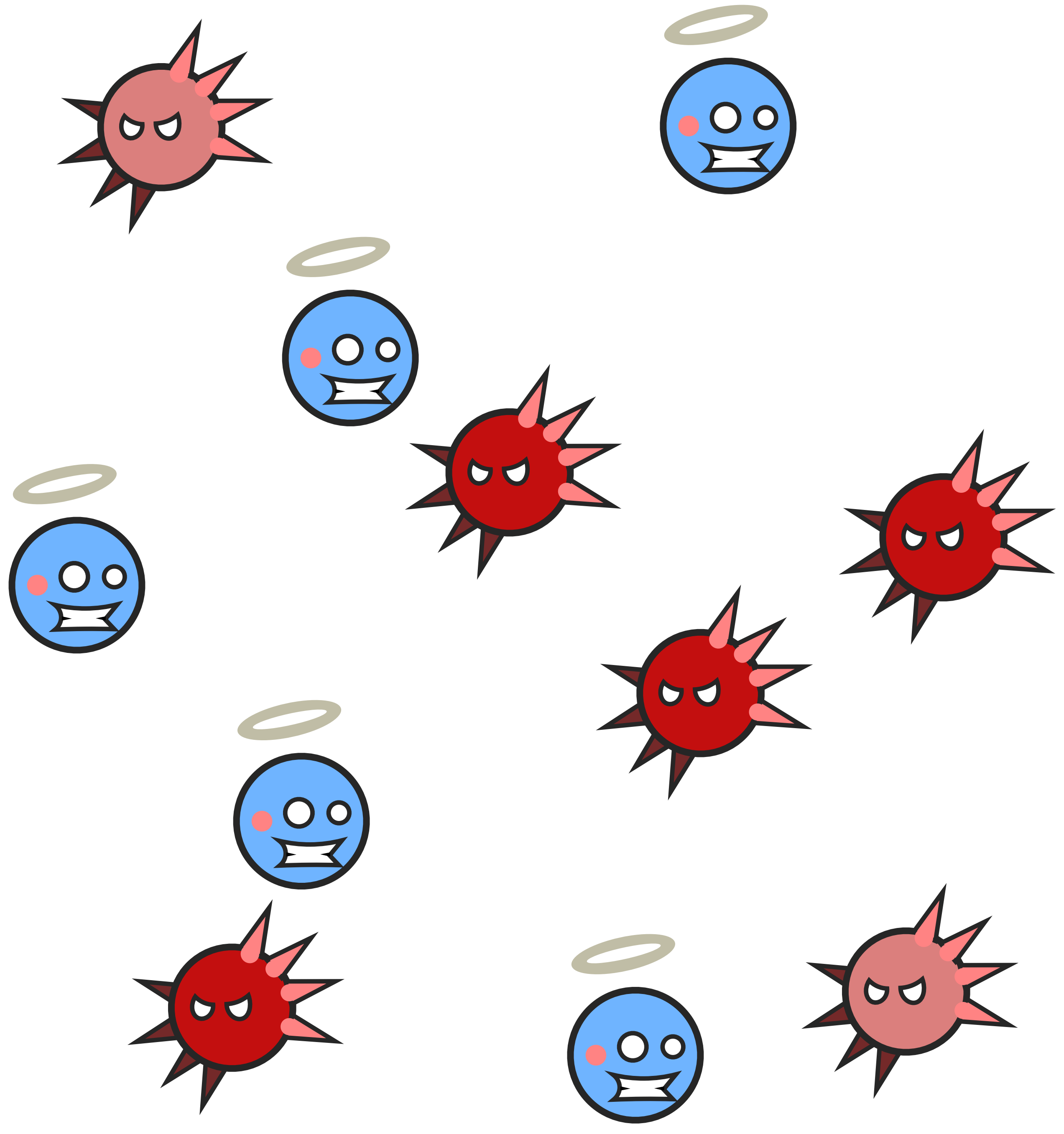
**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# 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

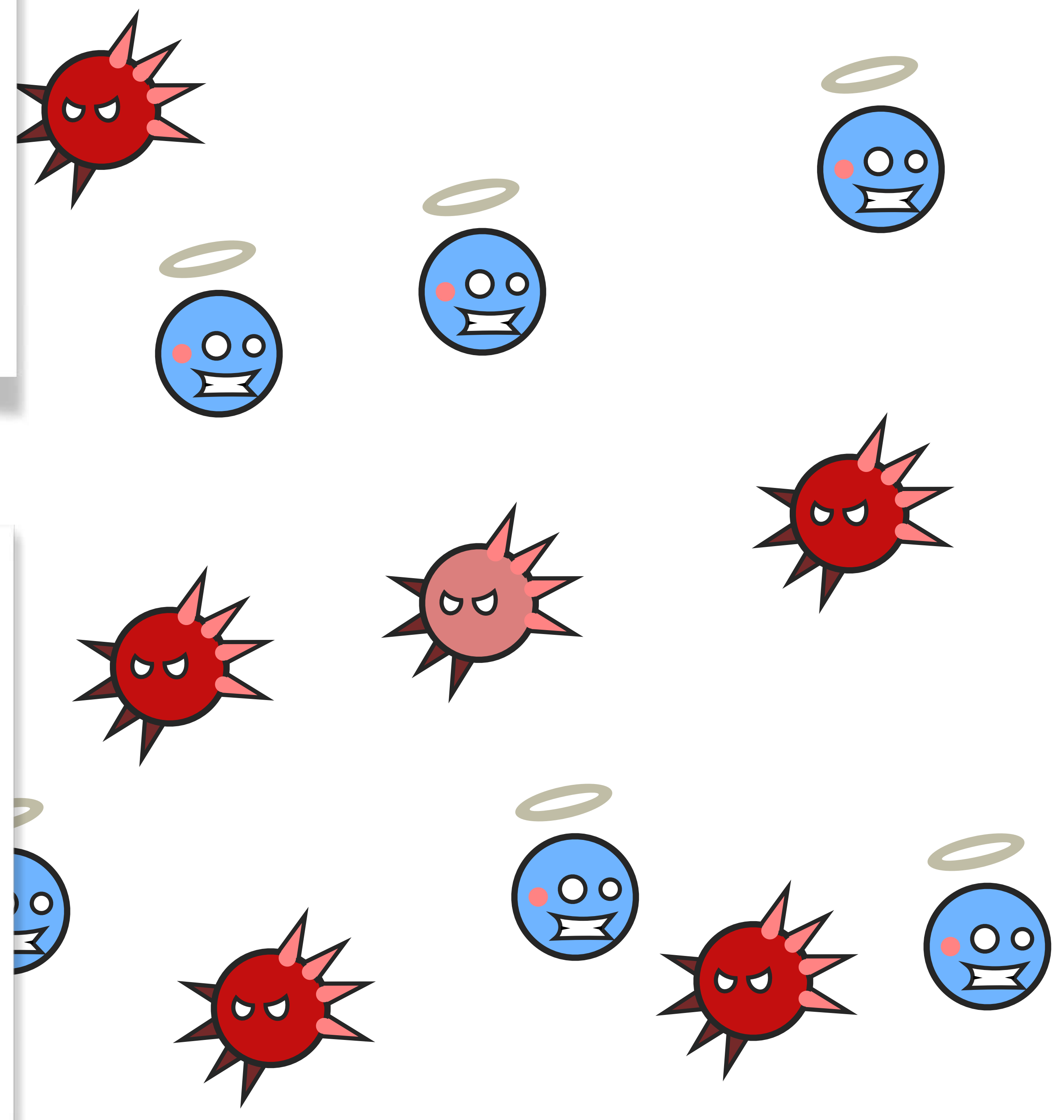
**Brad Miller et al. [DIMVA 2016]**

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**



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

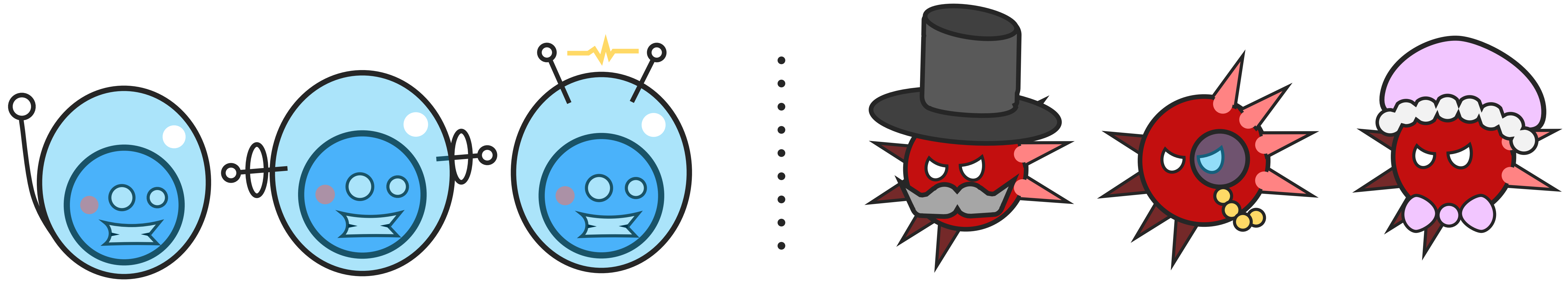
# Sources of Experimental Bias (2/3)

## Temporal {good|mal}ware inconsistency

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

# Sources of Experimental Bias (2/3)

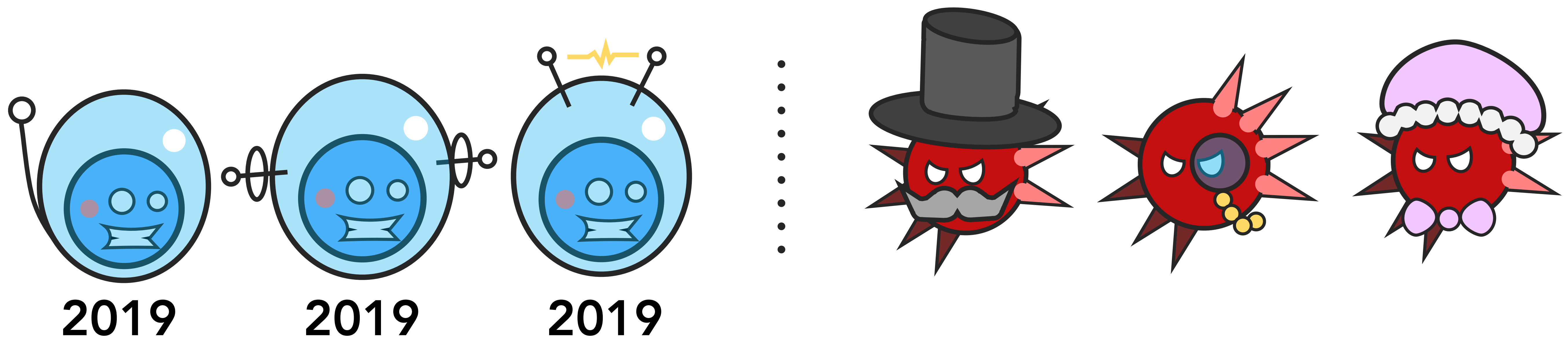
## Temporal {good|mal}ware inconsistency



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

# Sources of Experimental Bias (2/3)

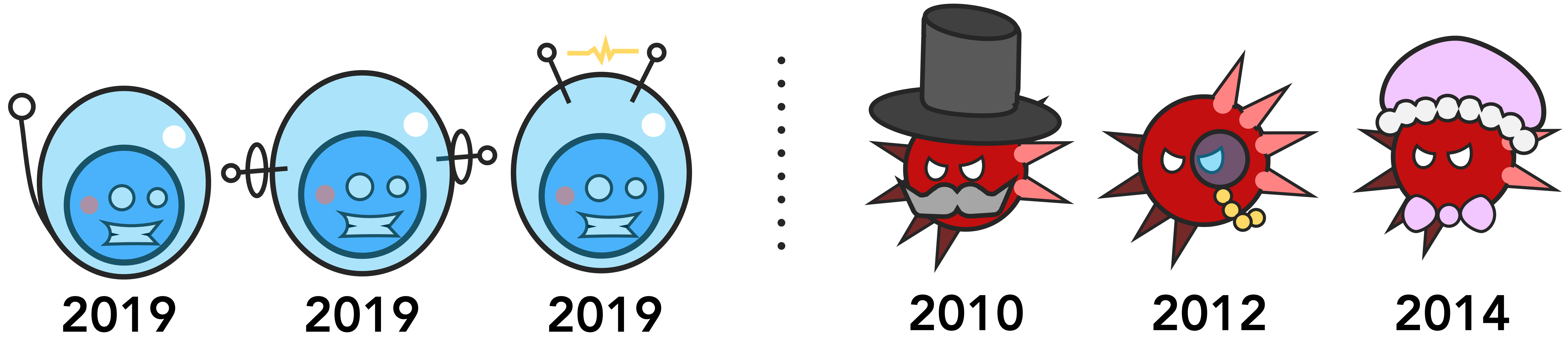
## Temporal {good|mal}ware inconsistency




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

# Sources of Experimental Bias (2/3)

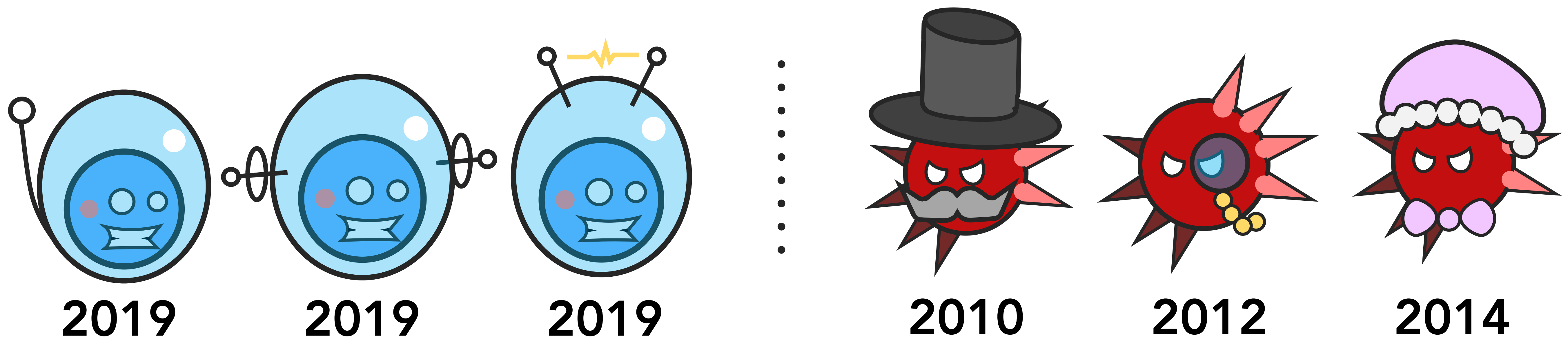
## Temporal {good|mal}ware inconsistency




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

# Sources of Experimental Bias (2/3)

## Temporal {good|mal}ware inconsistency

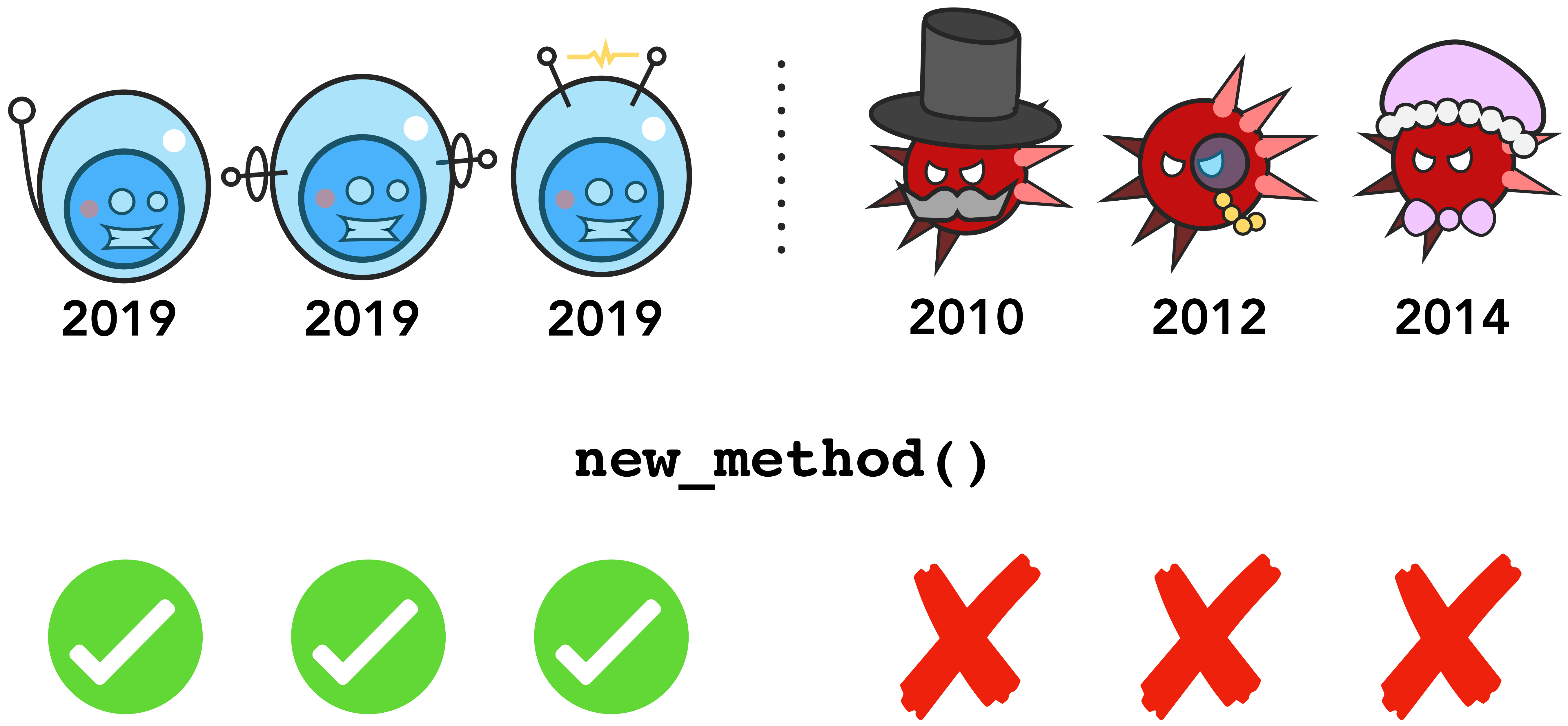



`new_method()`

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

# Sources of Experimental Bias (2/3)

## Temporal {good|mal}ware inconsistency

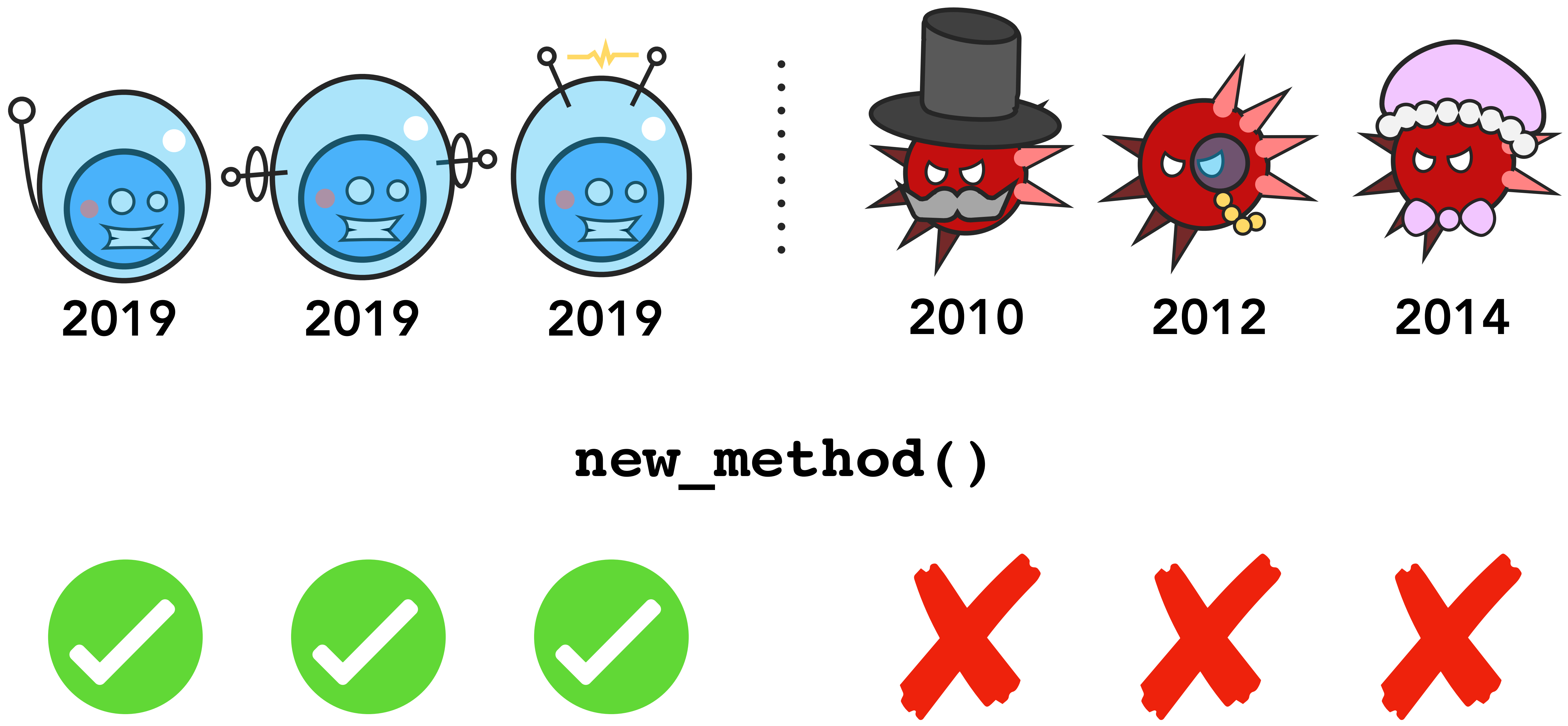



[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# Sources of Experimental Bias (2/3)

## Temporal {good|mal}ware inconsistency

**Violations** may learn artifacts




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

# Sources of Experimental Bias (3/3)

## Unrealistic Test Class Ratio

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

# Sources of Experimental Bias (3/3)

## Unrealistic Test Class Ratio

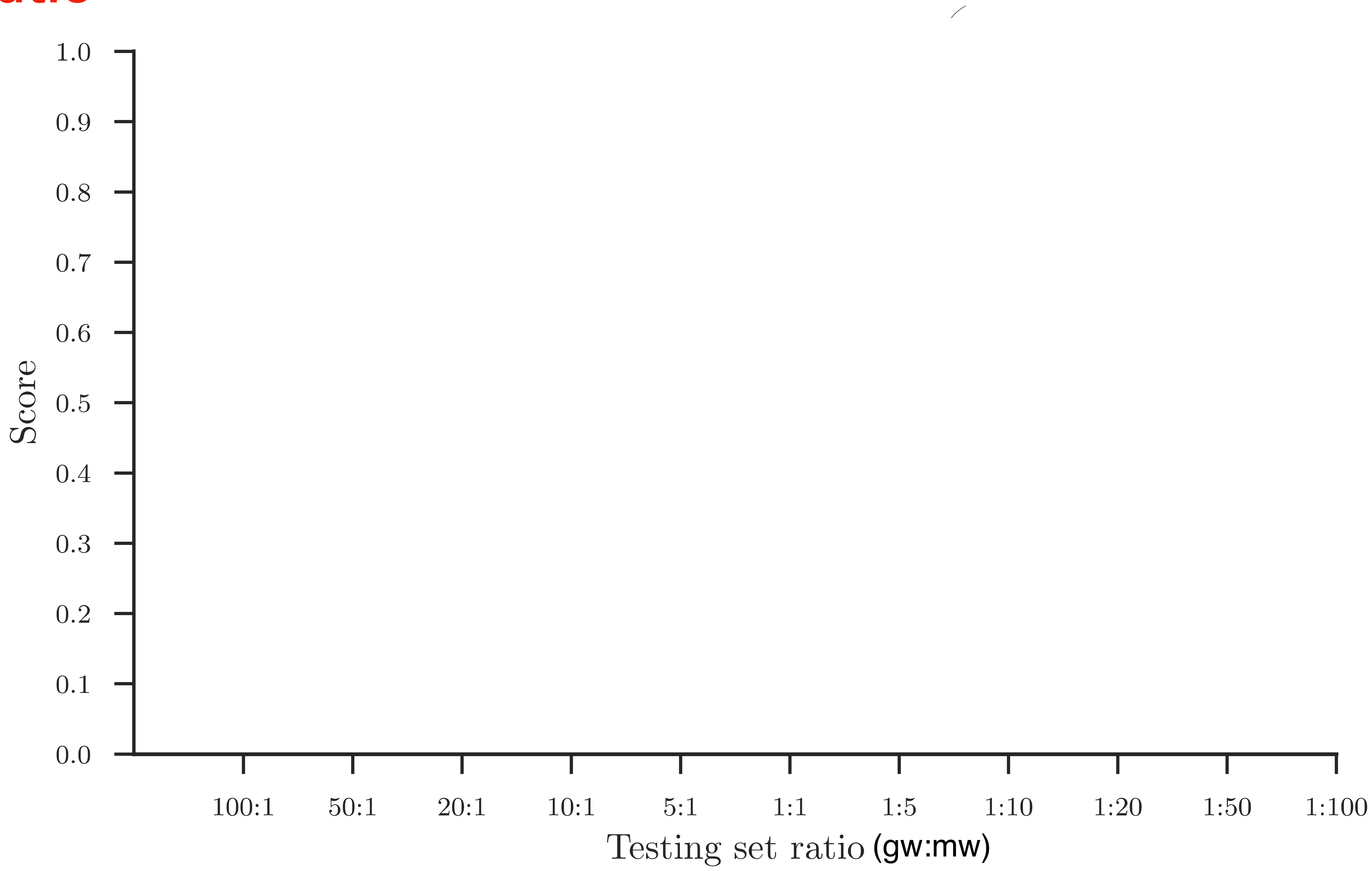
- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)

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

# Sources of Experimental Bias (3/3)

## Unrealistic Test Class Ratio

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

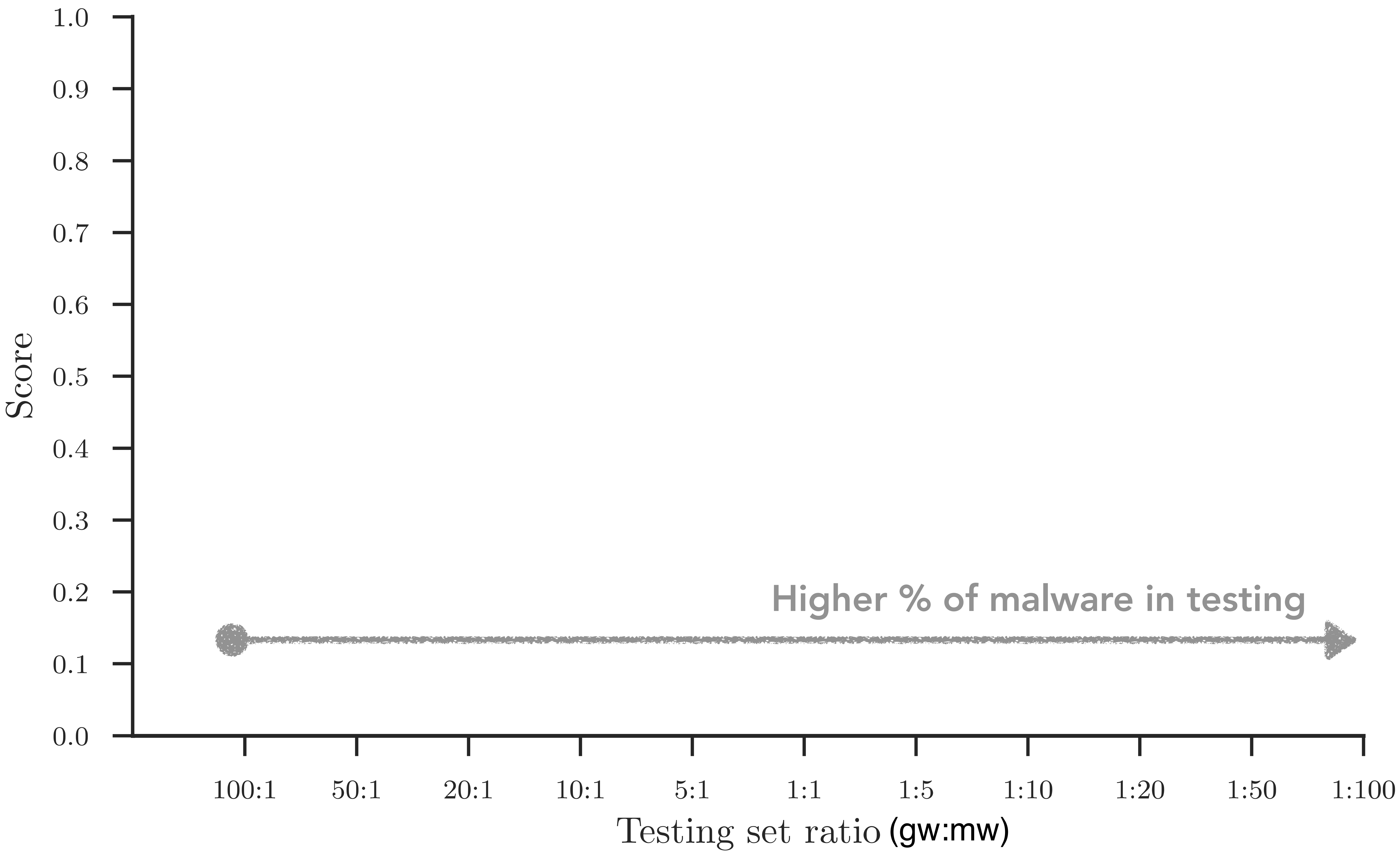



**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# Sources of Experimental Bias (3/3)

## Unrealistic Test Class Ratio

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

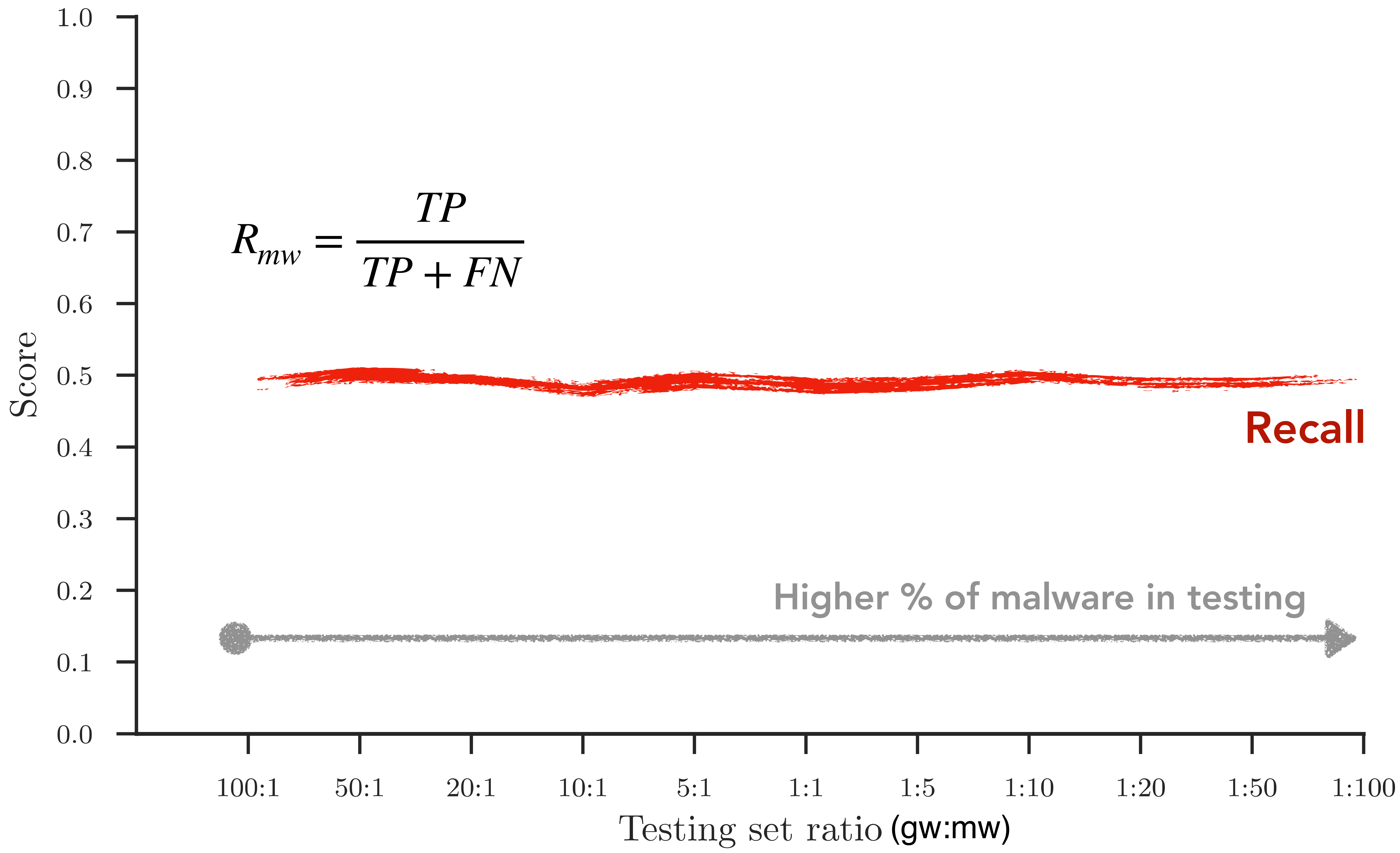



**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# Sources of Experimental Bias (3/3)

## Unrealistic Test Class Ratio

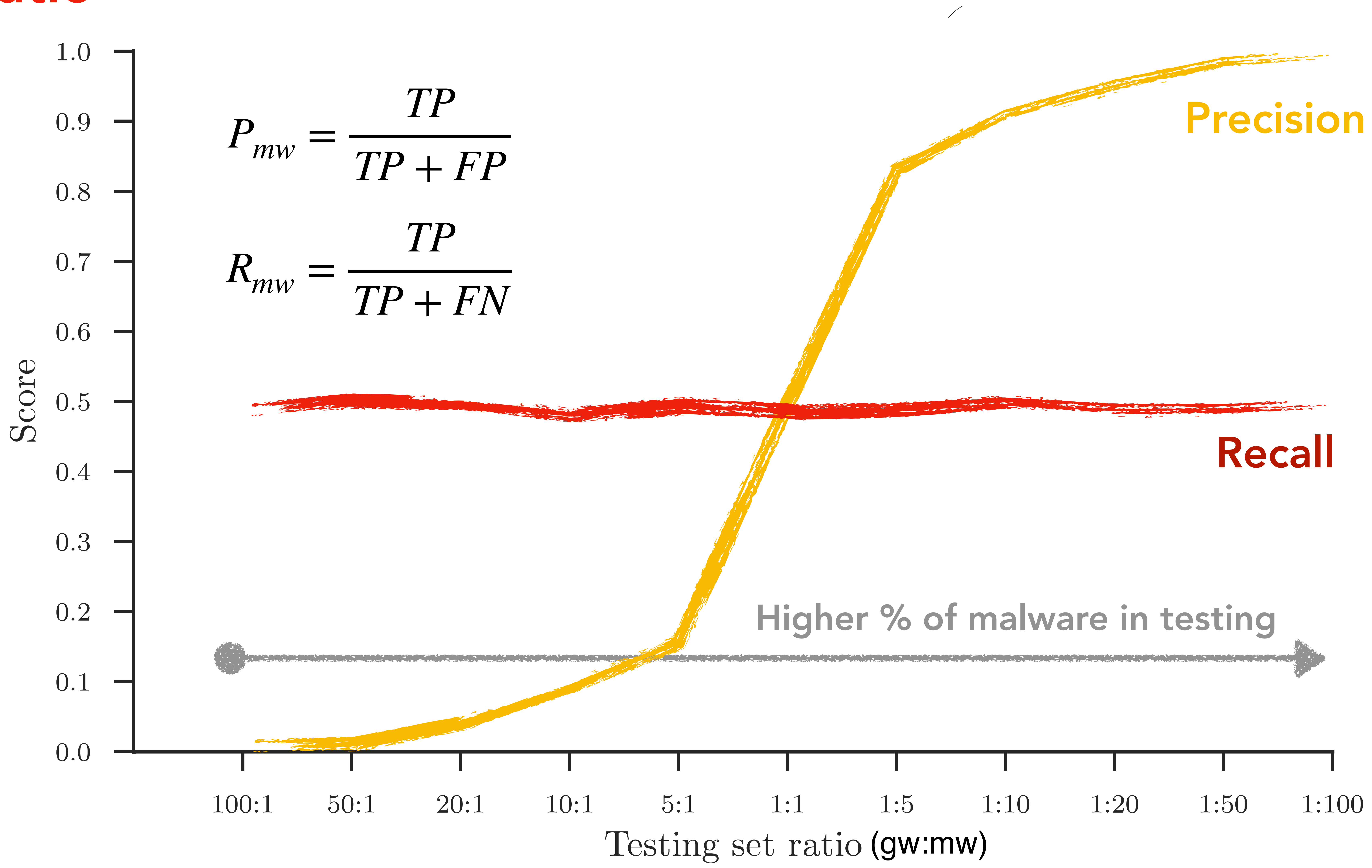
- **Training set**: Fixed
- **Testing set**: Varying % of mw (by downsampling gw)



# Sources of Experimental Bias (3/3)

## Unrealistic Test Class Ratio

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

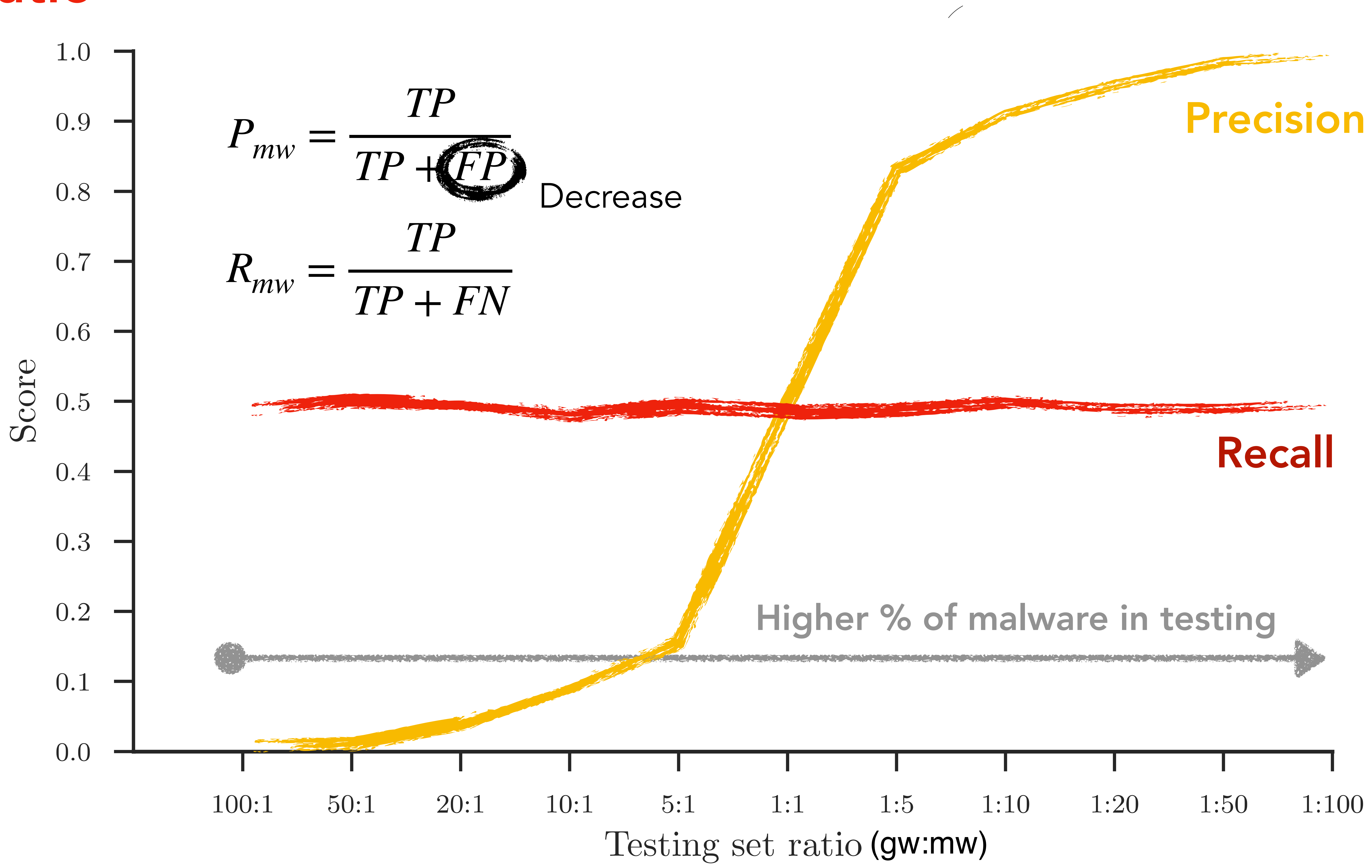



**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# Sources of Experimental Bias (3/3)

## Unrealistic Test Class Ratio

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

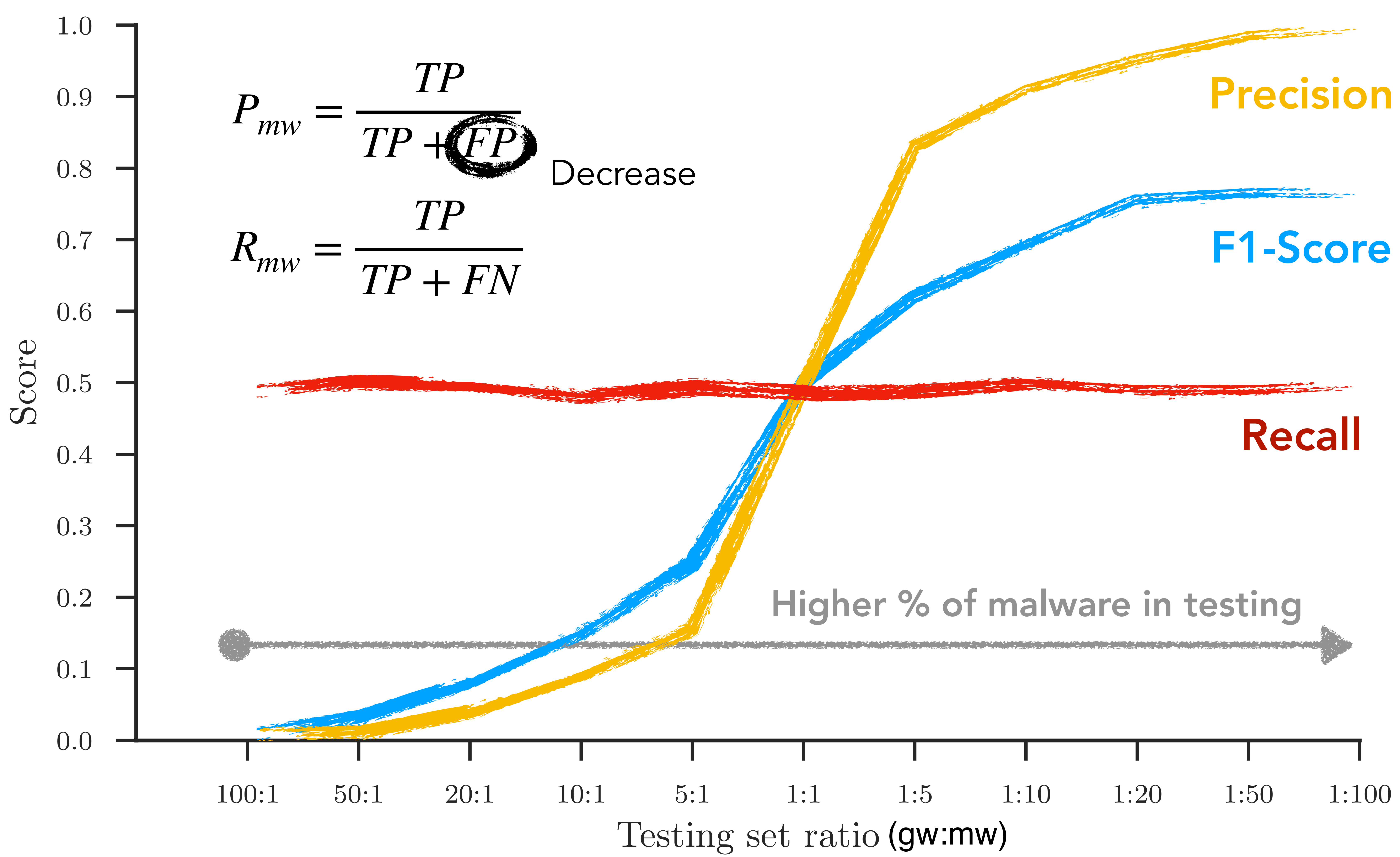



**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# Sources of Experimental Bias (3/3)

## Unrealistic Test Class Ratio

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



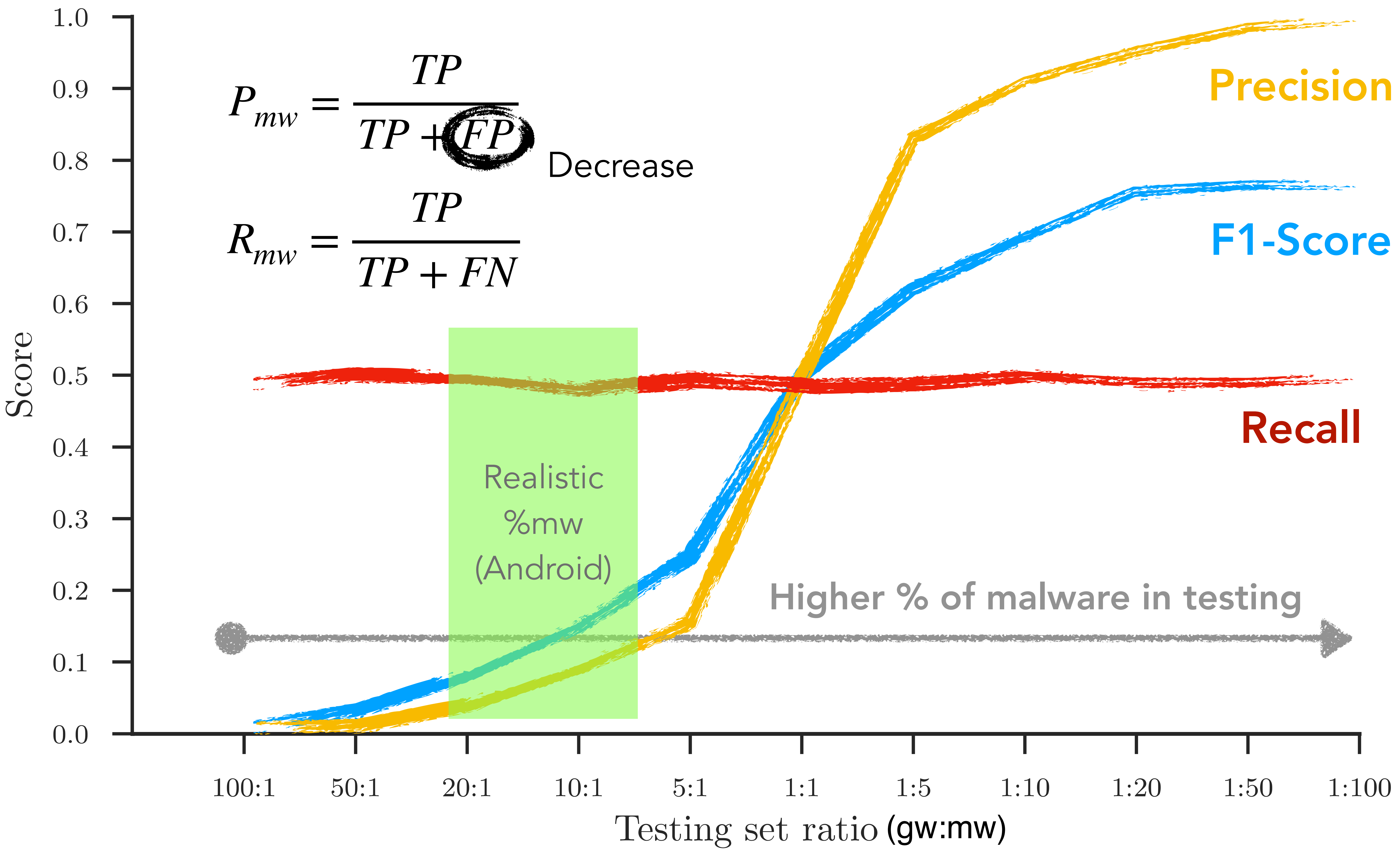

**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# Sources of Experimental Bias (3/3)

## Unrealistic Test Class Ratio

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

**Violations** produce unrealistic results




**[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

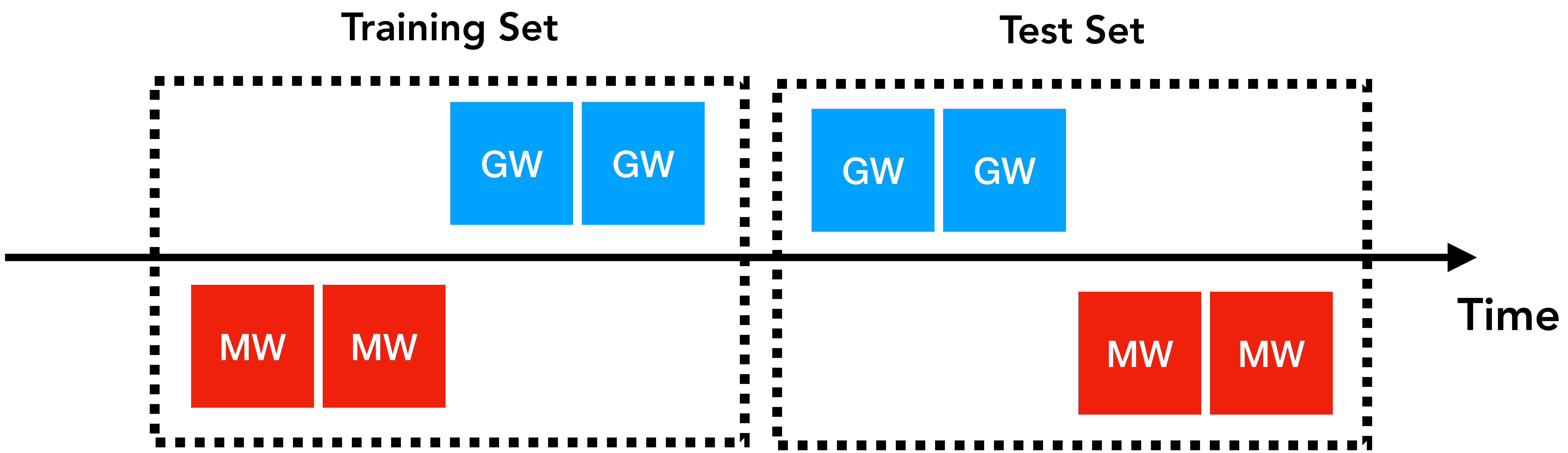
Time

**[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)

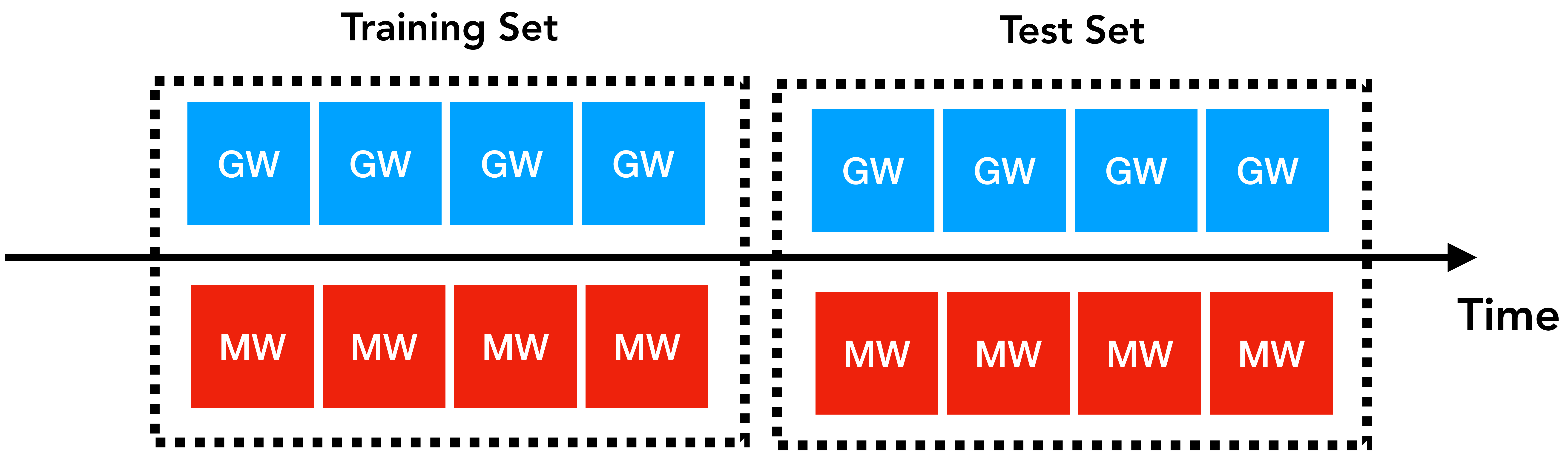



**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# TESSERACT Framework

Experimental Constraints

| | | |
|---|---|---|
| **C1** | Temporal training consistency | → time(training) < time(testing) |
| **C2** | {good\|mal}ware temporal consistency | → time(gw) = time(mw) |

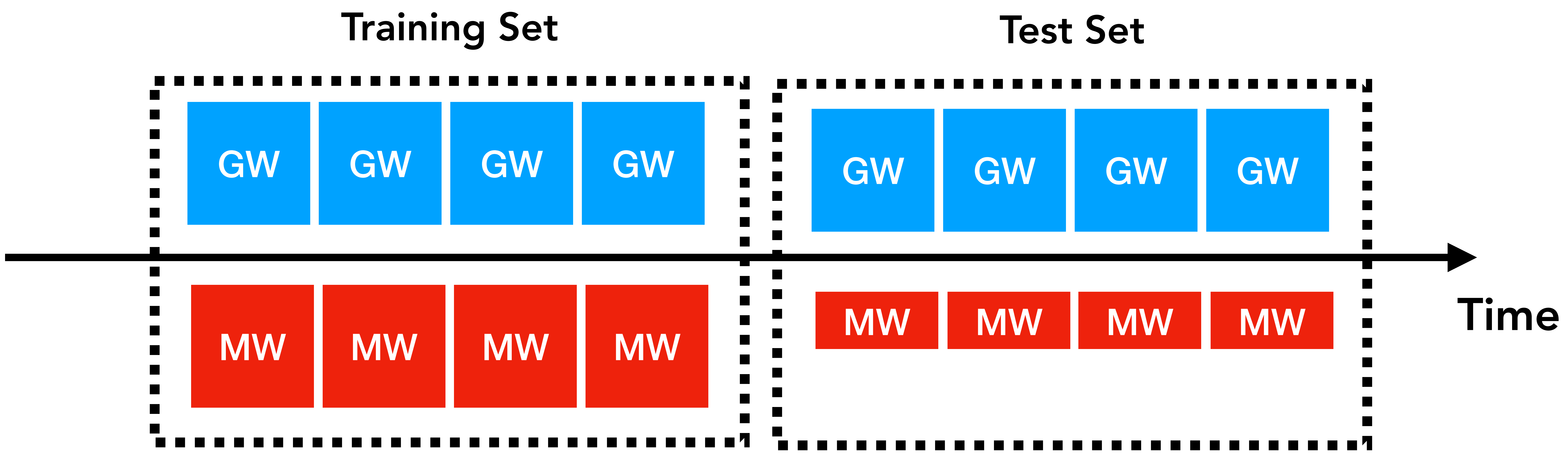



**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# TESSERACT Framework

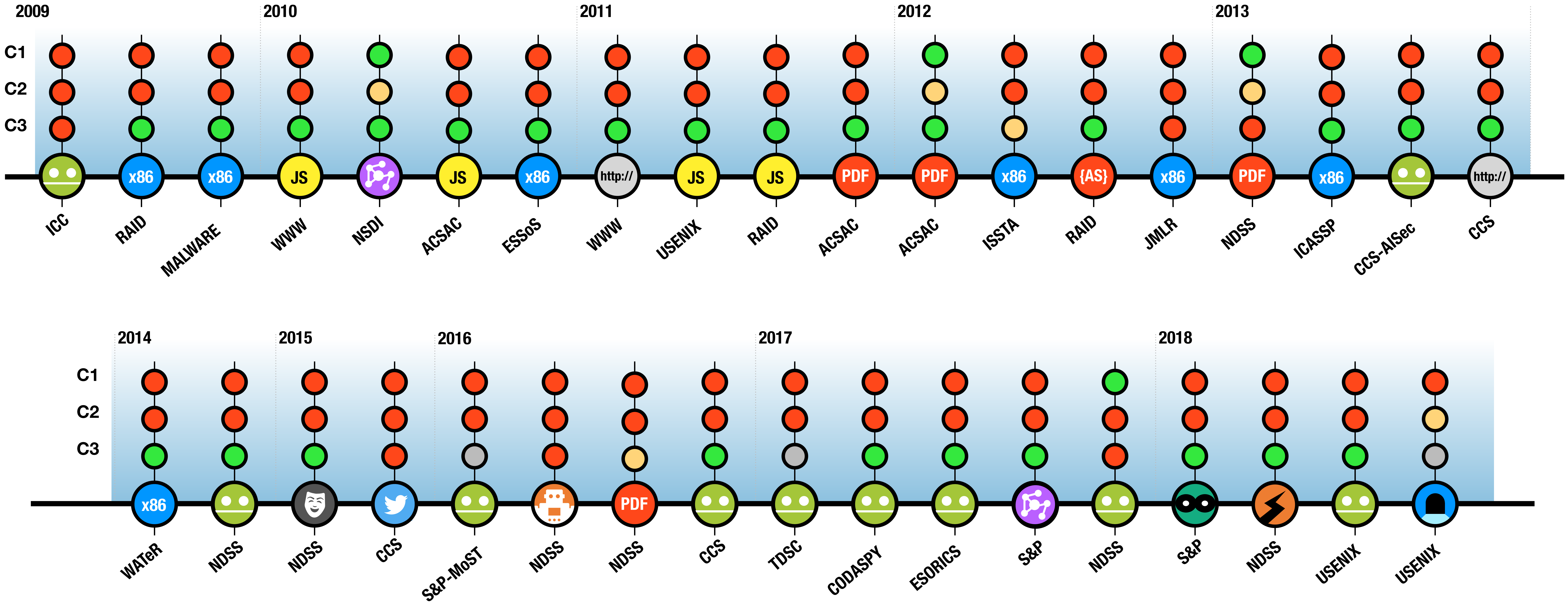
**Experimental Constraints**

| | | | |
|---|---|---|---|
| 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

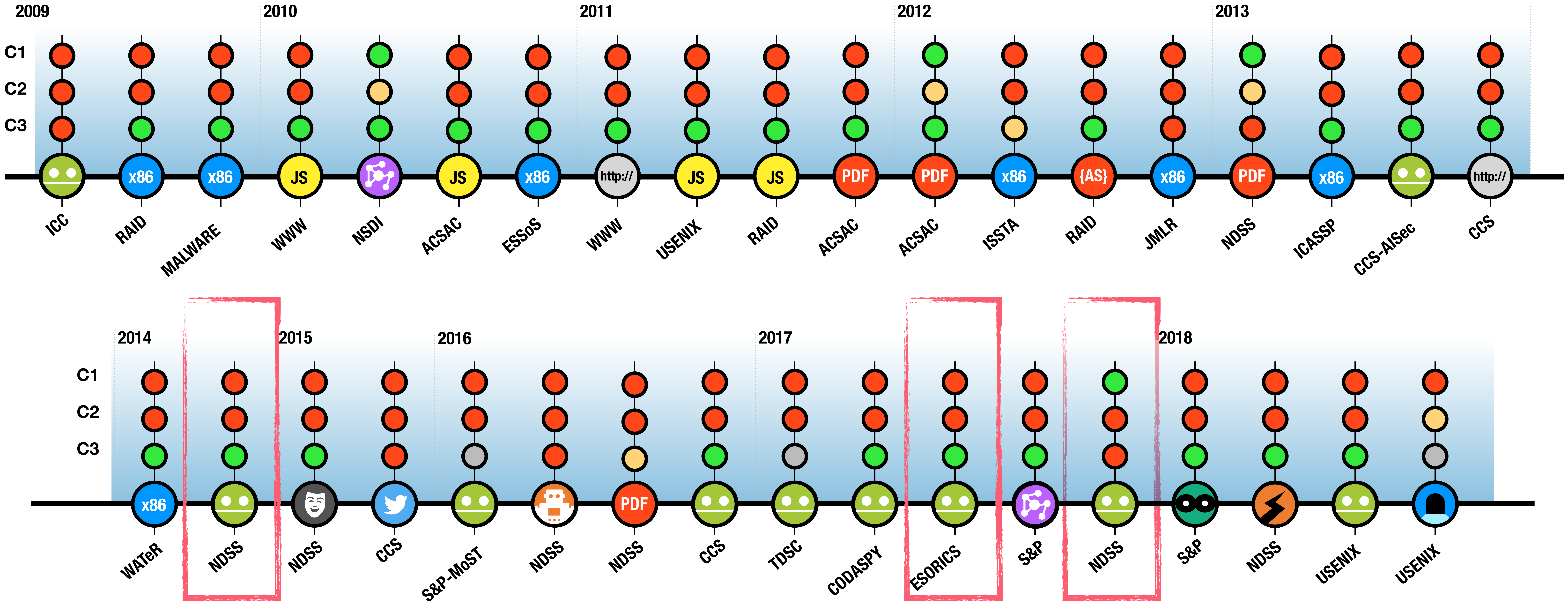
# Endemic Problem




**Details:** https://s2lab.kcl.ac.uk/projects/tesseract/poster-references.pdf

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

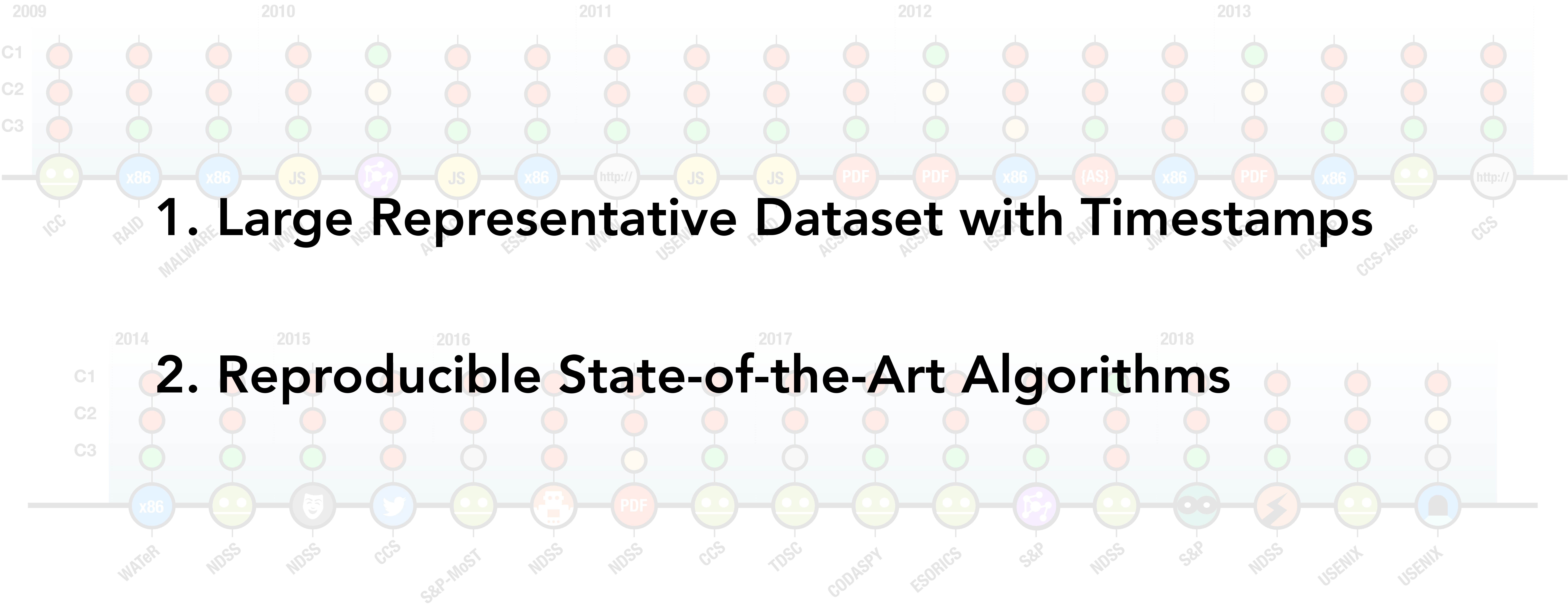
# Endemic Problem




**Details:** https://s2lab.kcl.ac.uk/projects/tesseract/poster-references.pdf

**[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

Details: https://s2lab.kcl.ac.uk/projects/tesseract/poster-references.pdf

**[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)

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

# 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

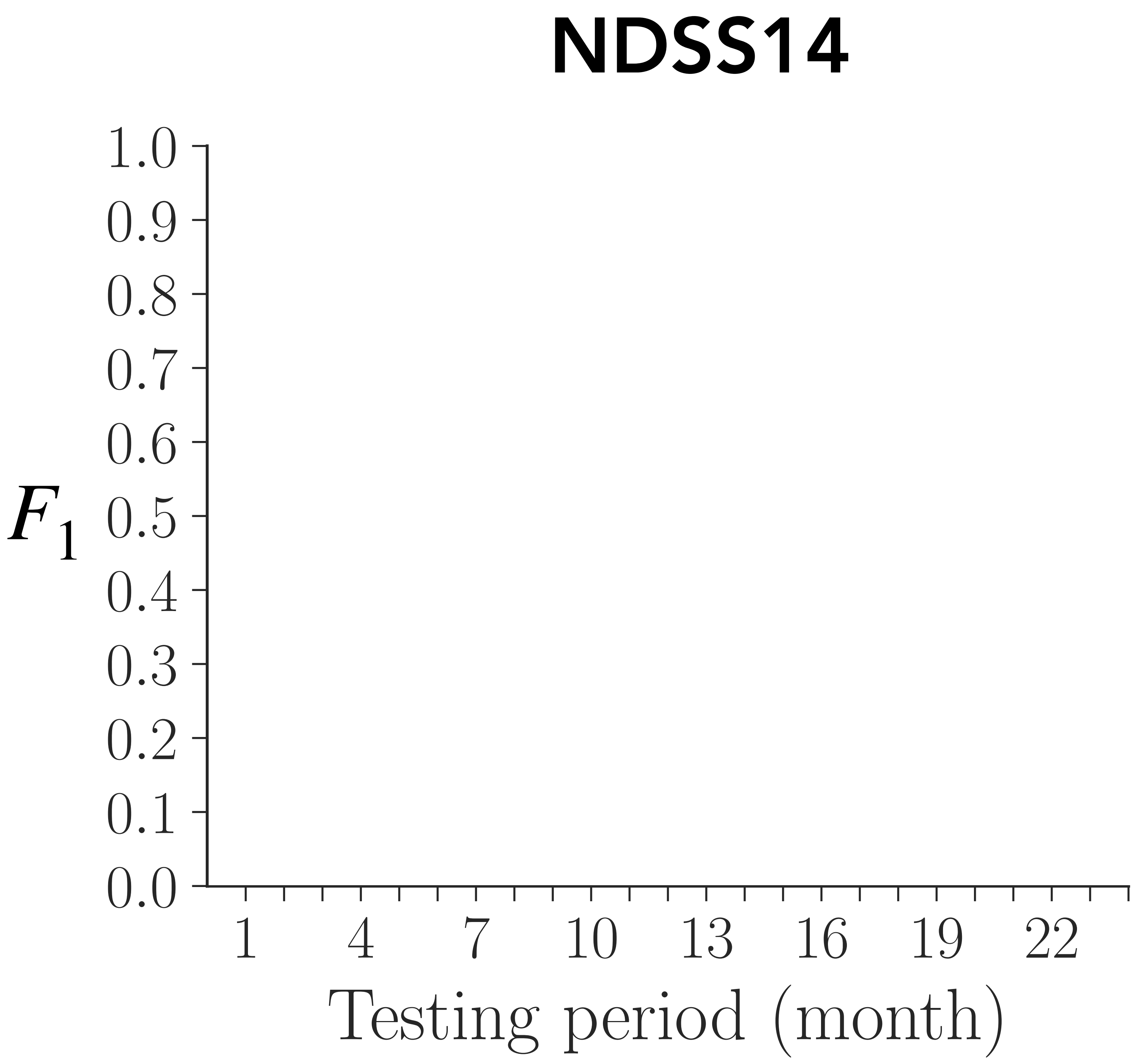
| | |
|---|---|
| C1 | Temporal training consistency |
| C2 | {good\|mal}ware temporal consistency |
| C3 | Realistic testing classes ratio |

**[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 |



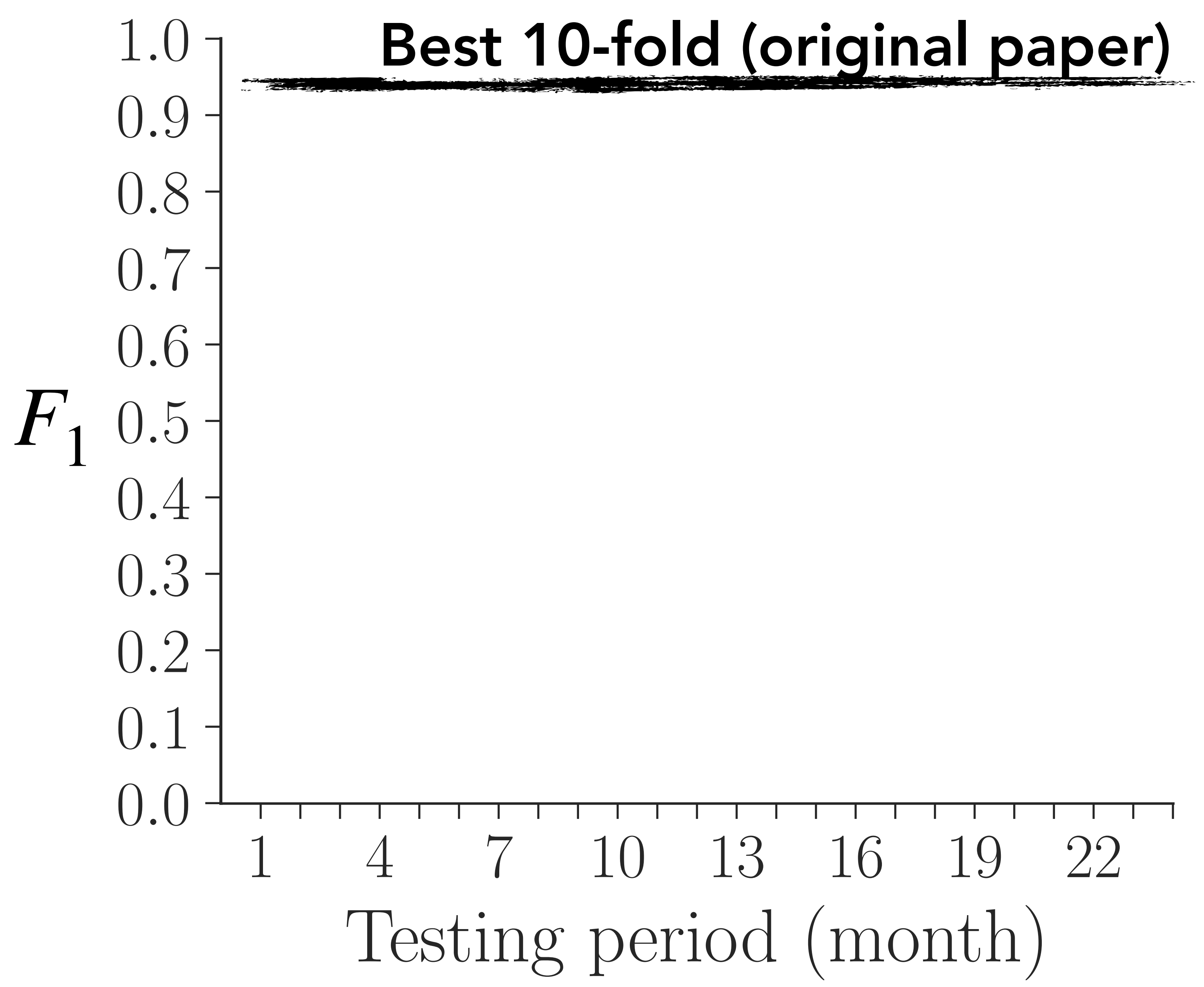

**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# TESSERACT Evaluations

**Experimental Constraints**

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

**NDSS14**

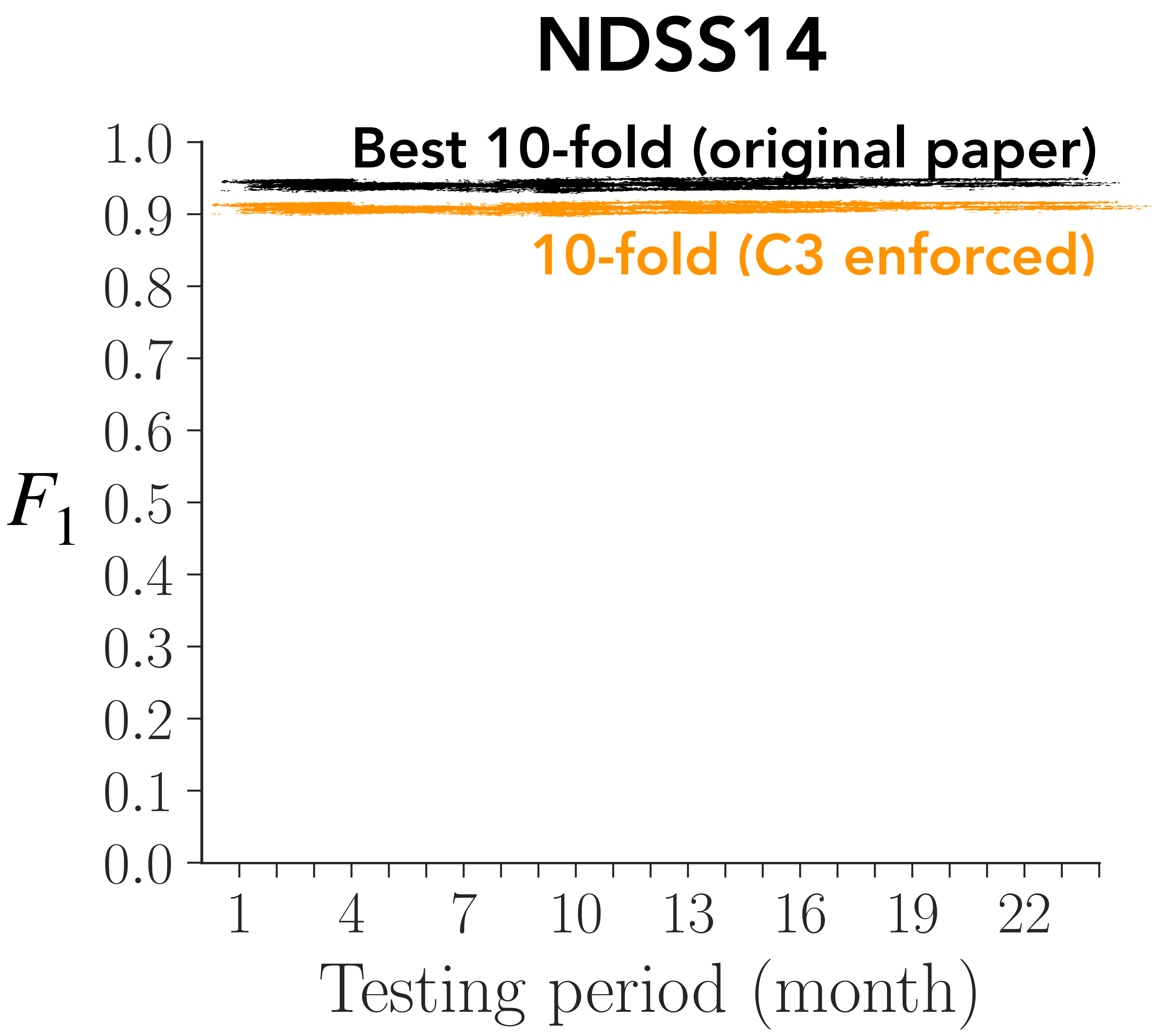



**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# TESSERACT Evaluations

**Experimental Constraints**

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

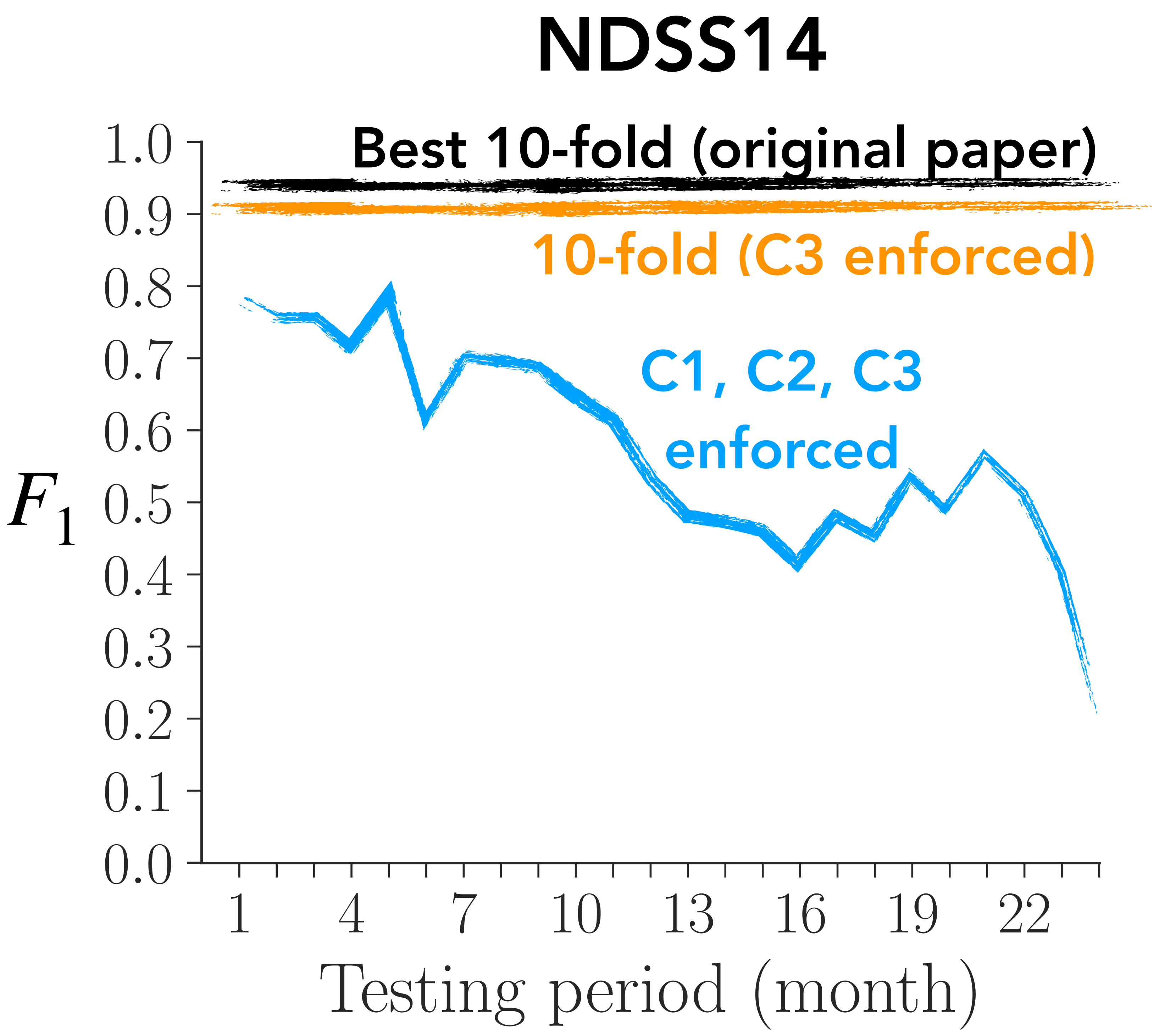



**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# TESSERACT Evaluations

**Experimental Constraints**

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



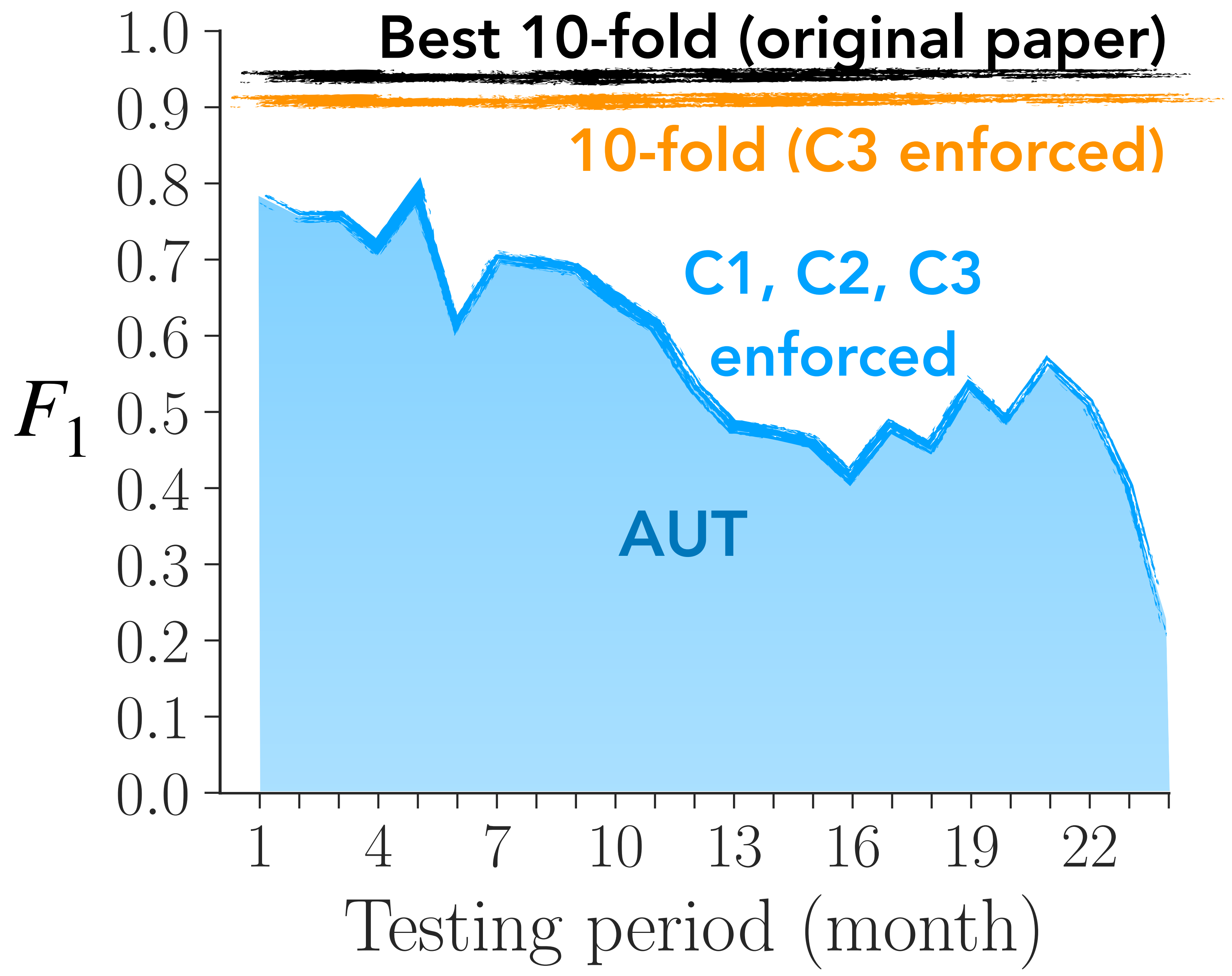

**[USENIX Sec 2019]** TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# TESSERACT Evaluations

**Experimental Constraints**

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

**NDSS14**




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

**Area Under Time**
**AUT(Metric, Period)**

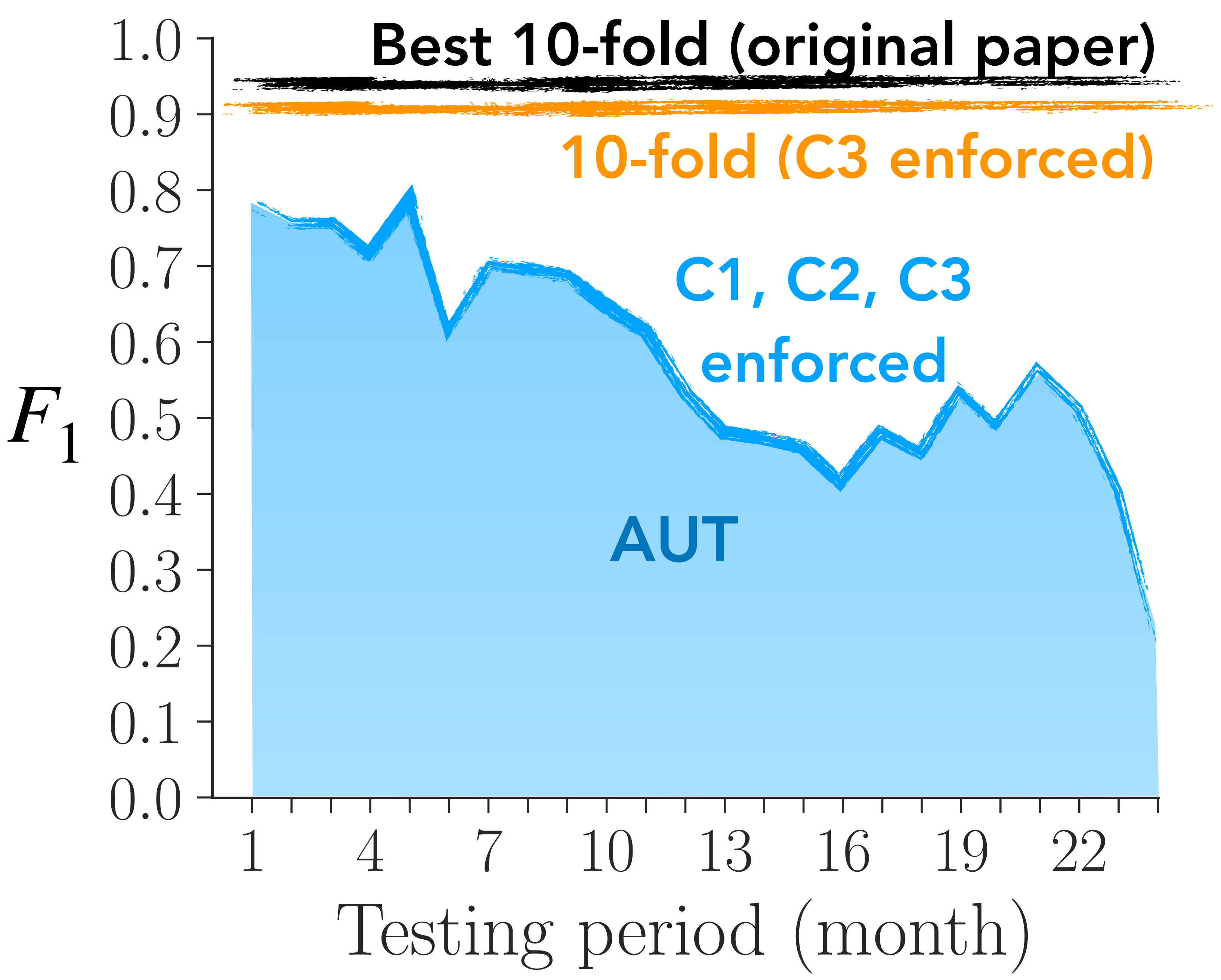
**[USENIX Sec 2019]** **TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time**
https://s2lab.cs.ucl.ac.uk/projects/tesseract

# TESSERACT Evaluations

Experimental Constraints

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

## NDSS14




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

**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

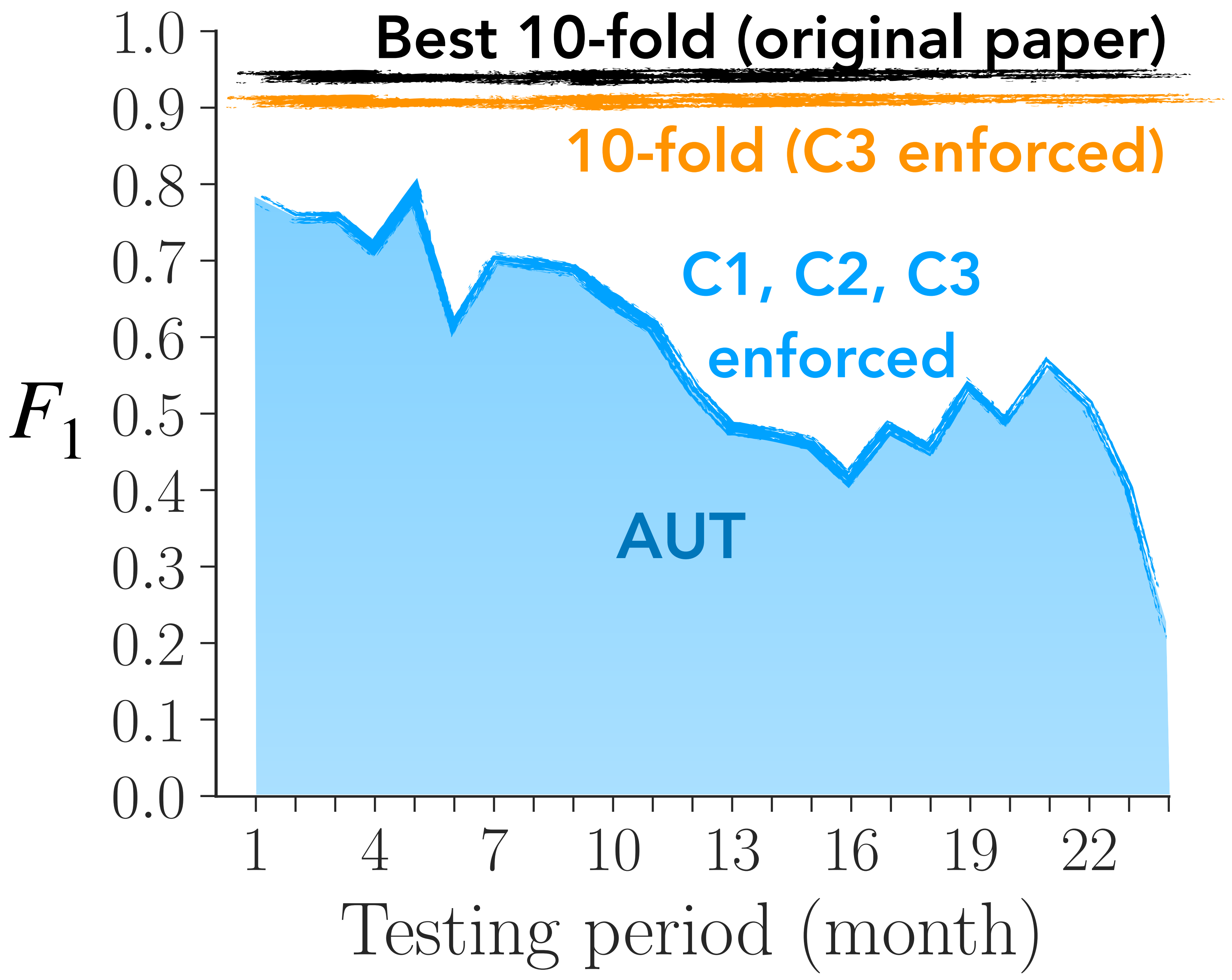
# TESSERACT Evaluations

**Experimental Constraints**

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

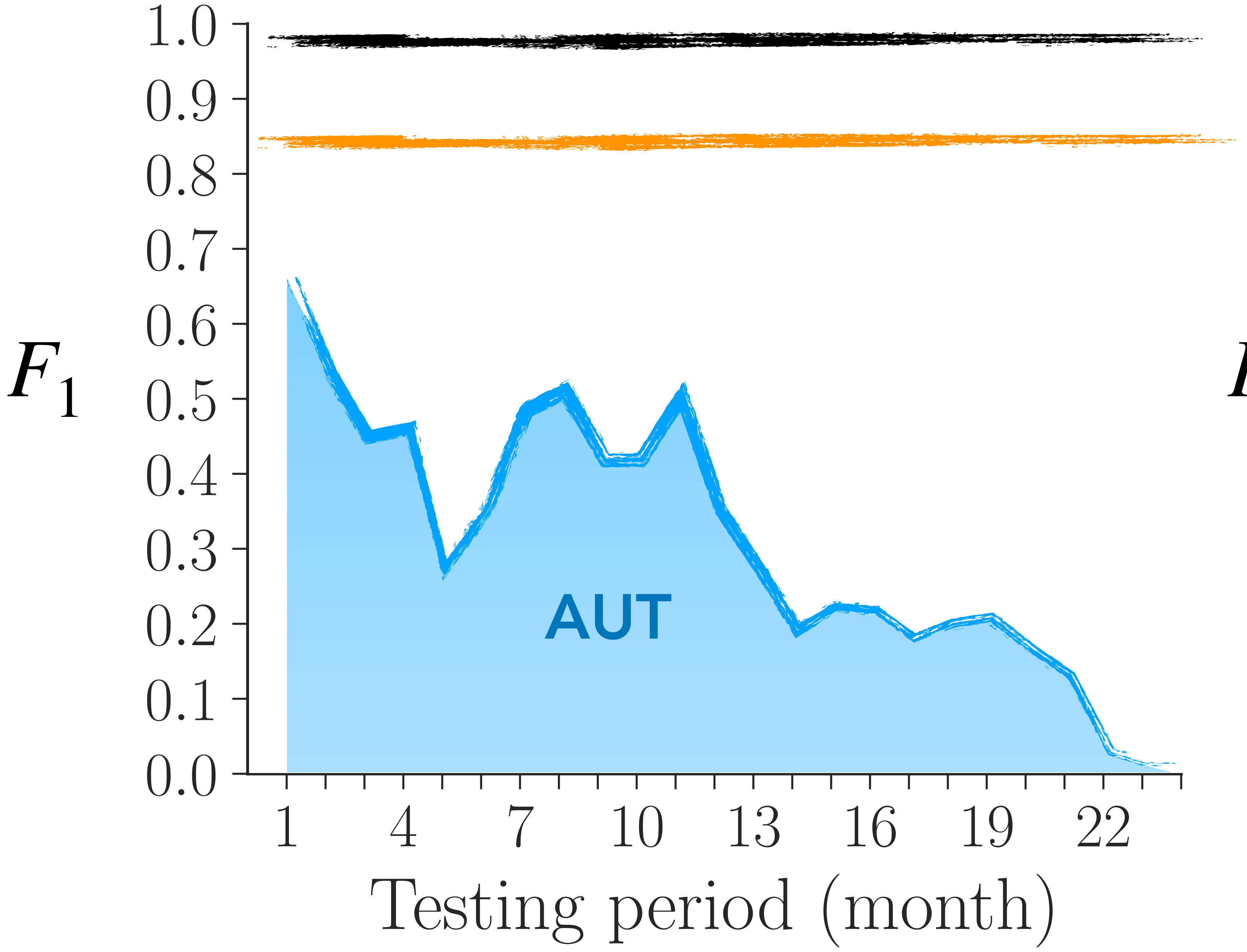
**Area Under Time**
**AUT(Metric, Period)**

## NDSS14



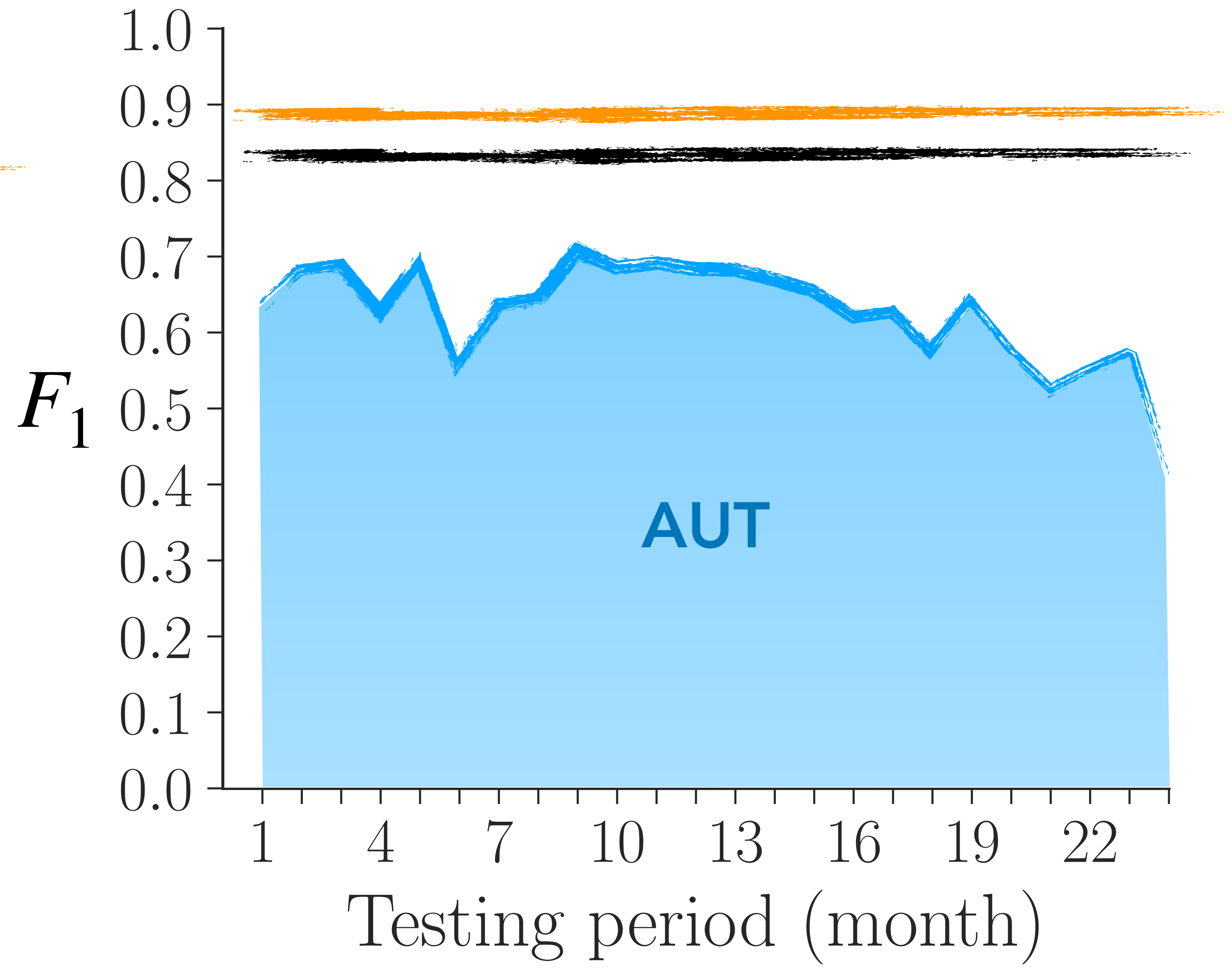

$$AUT(F_1, 24m) = 0.58$$

## 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)

**Rejection***  Incremental Retraining  Active Learning

# 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

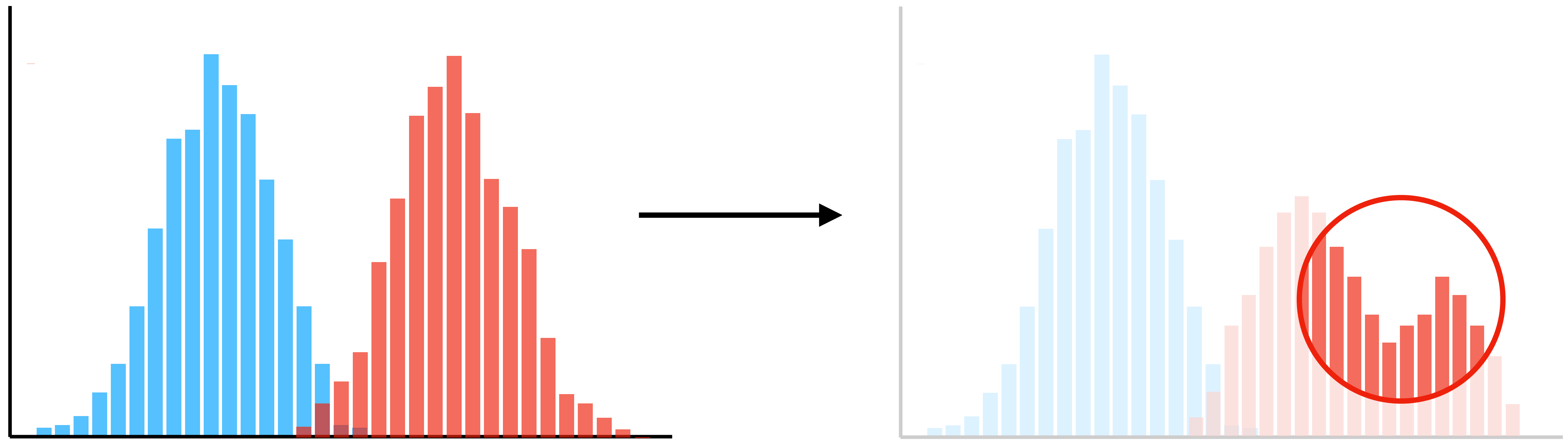
- **Detection Performance** (e.g., AUT $F_1$)
- **Labeling Cost** for retraining (e.g., manpower)
- **Quarantine Cost** for rejection (e.g., low-confidence decisions)

**Rejection***

Incremental Retraining

Active Learning

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

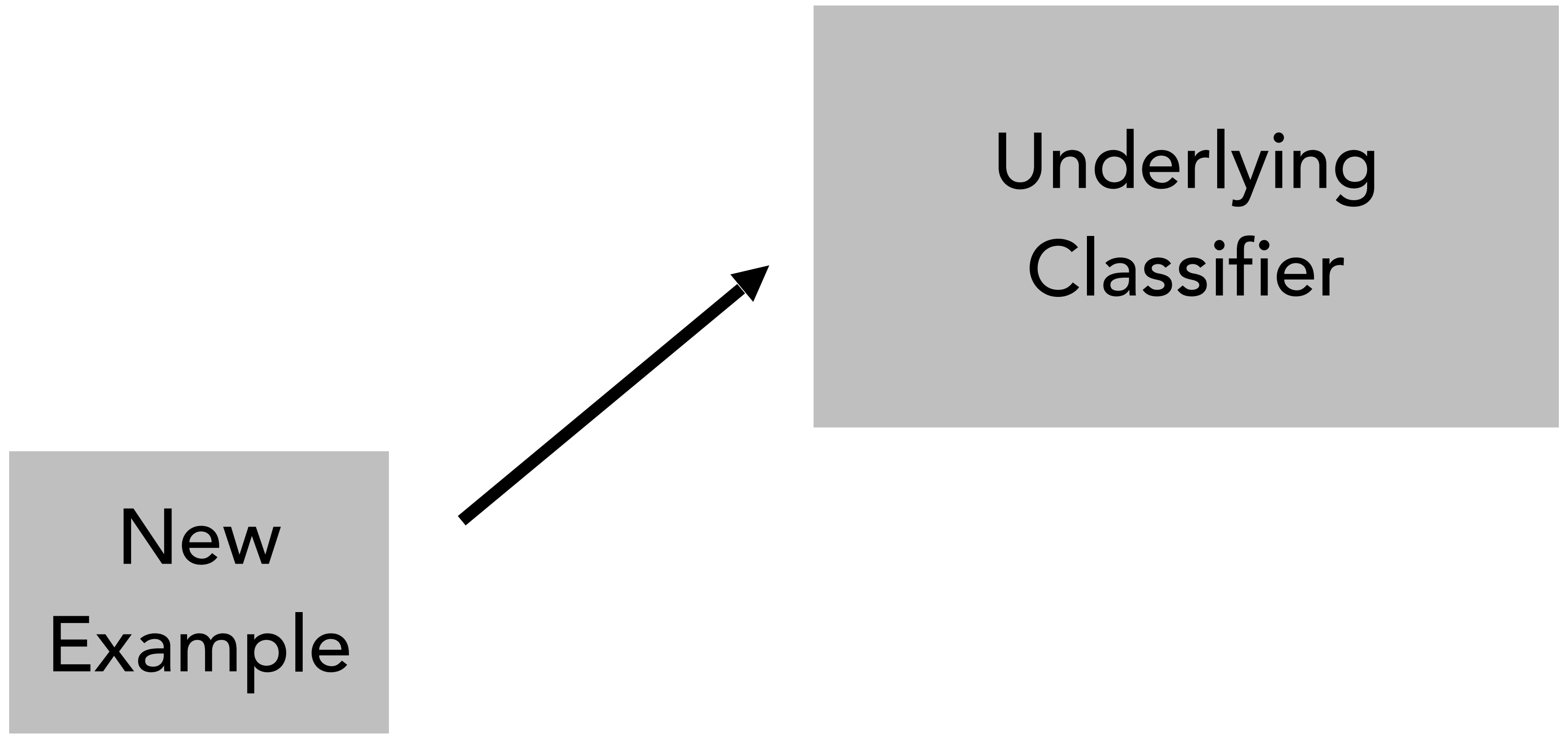
https://s2lab.cs.ucl.ac.uk/projects/transcend

# Revisiting Classification in the Presence of Concept Drift

# Revisiting Classification in the Presence of Concept Drift

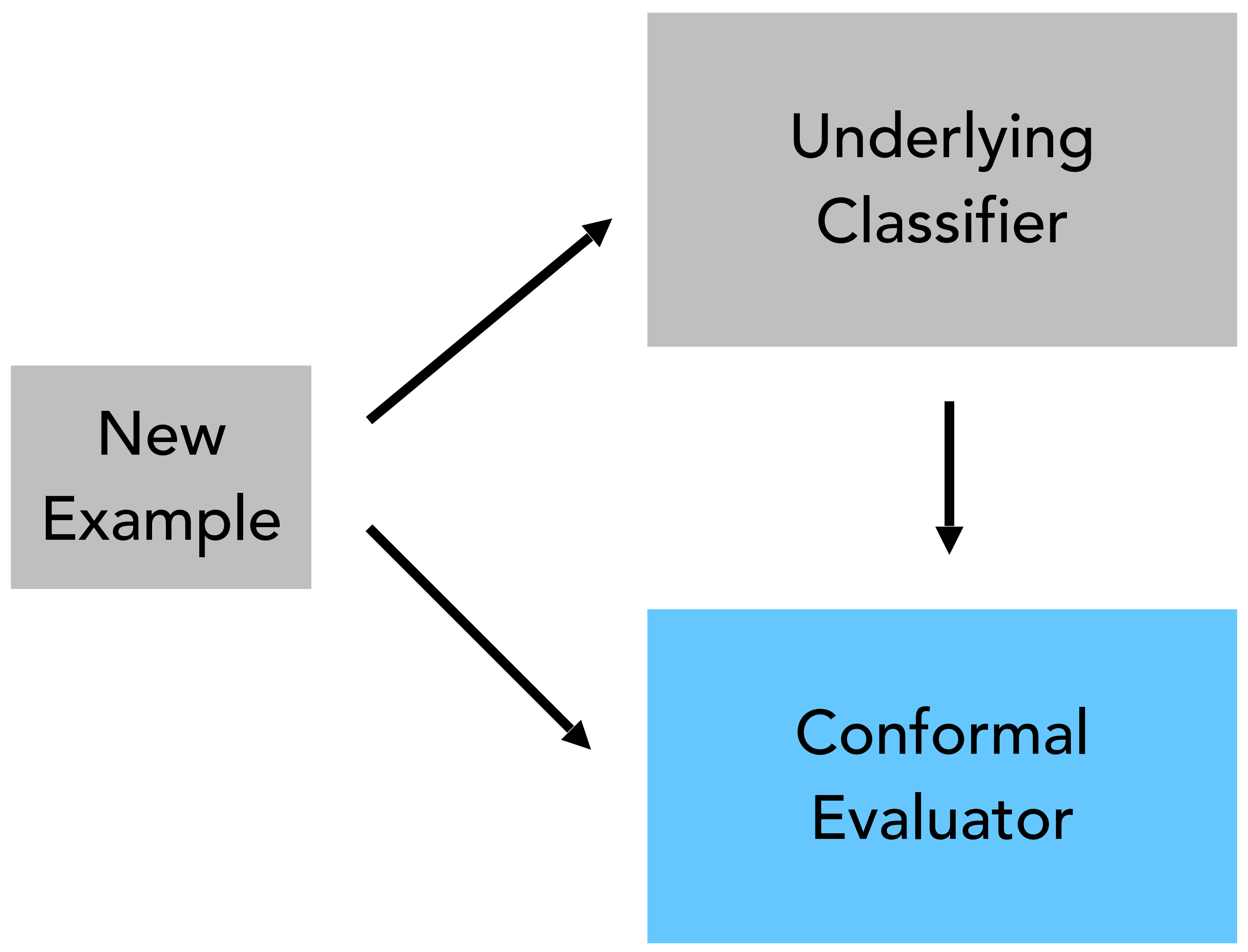
| | | |
|---|---|---|
| **Covariate Shift:** | Change in feature distribution | $P(x \in X)$ |
| **Prior-probability Shift:** | Change in class base rate | $P(y \in Y)$ |
| **Concept Drift:** | Change in ground truth definition | $P(y \in Y \mid x \in X)$ |

# Transcend at Test Time



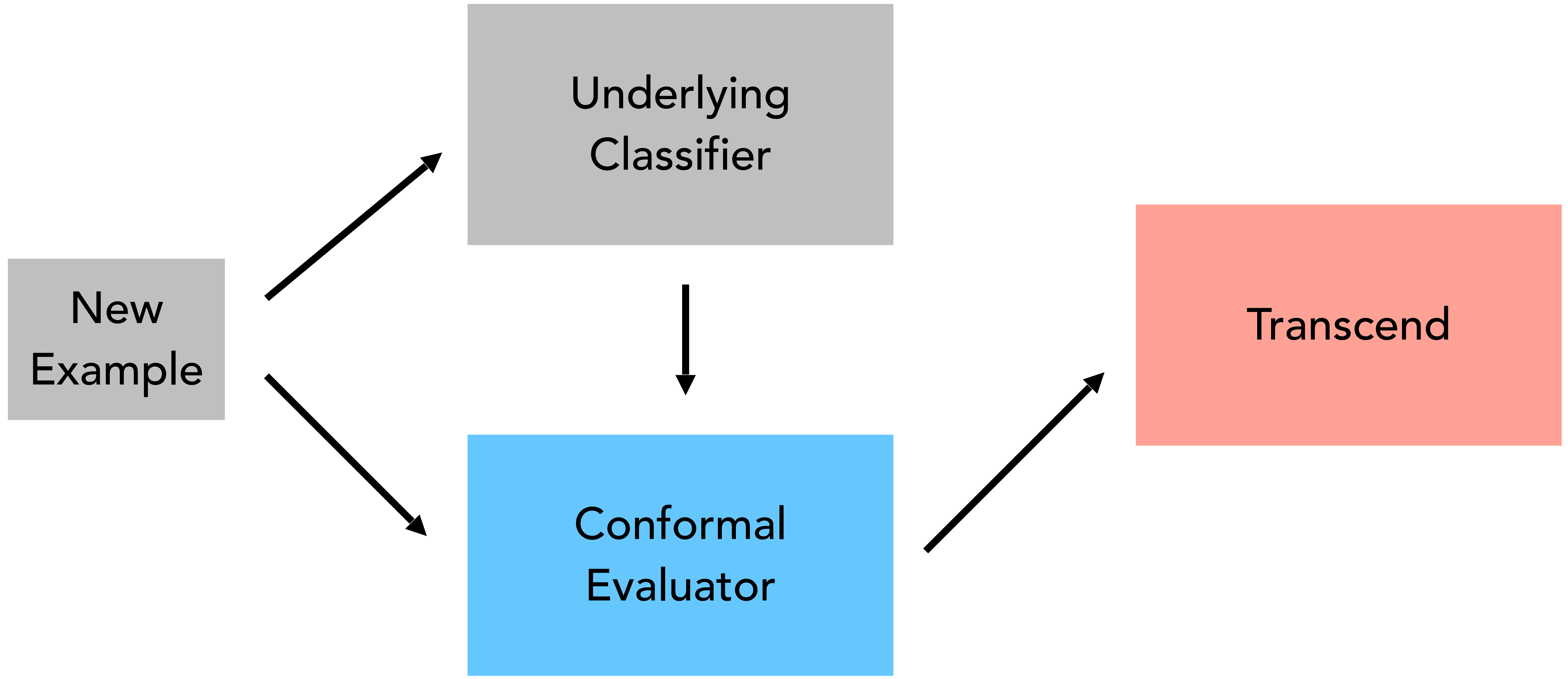

[USENIX Sec 2017] Transcend: Detecting Concept Drift in Malware Classification Models
https://s2lab.cs.ucl.ac.uk/projects/transcend/

# Transcend at Test Time



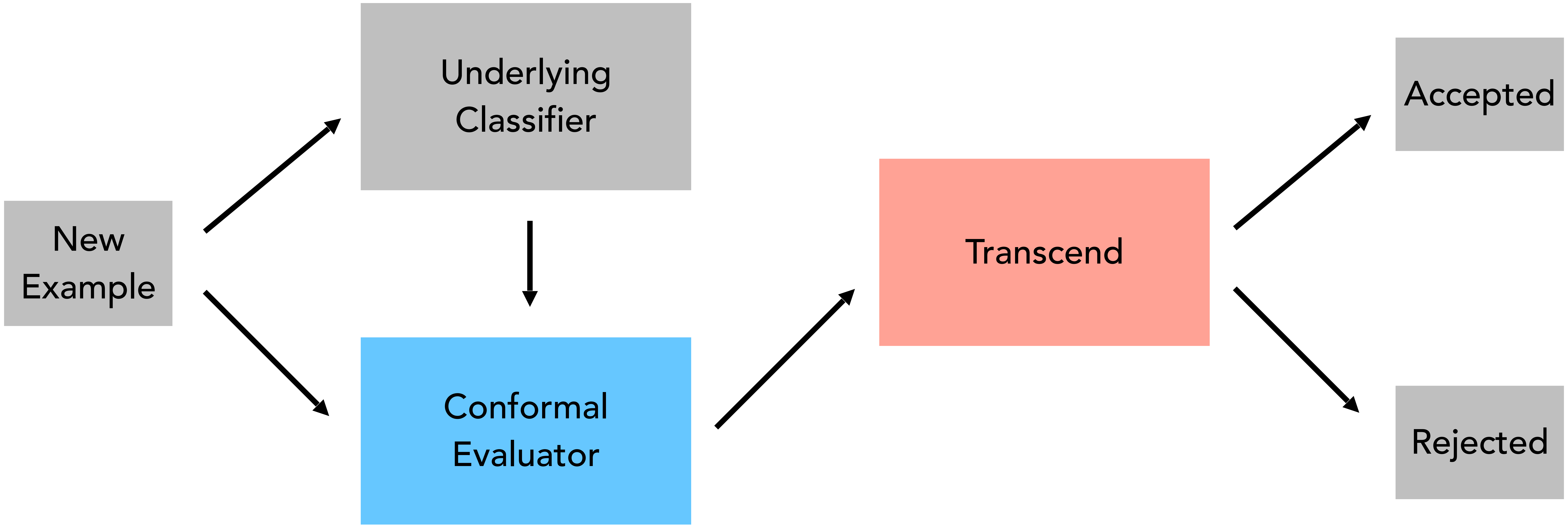

**[USENIX Sec 2017]** Transcend: Detecting Concept Drift in Malware Classification Models
https://s2lab.cs.ucl.ac.uk/projects/transcend/

# 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/

# Classification with Rejection


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

# Classification with Rejection

## Theoretical Understanding

- Provide missing link with Conformal Prediction Theory
- Motivate the effectiveness of Conformal Evaluation


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

# Classification with Rejection

**Theoretical Understanding**

- Provide missing link with Conformal Prediction Theory
- Motivate the effectiveness of Conformal Evaluation

**Computational Optimizations**

- New, sound and more flexible Conformal Evaluators
- Faster thresholding


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

# Classification with Rejection

**Theoretical Understanding**

- Provide missing link with Conformal Prediction Theory
- Motivate the effectiveness of Conformal Evaluation

**Computational Optimizations**

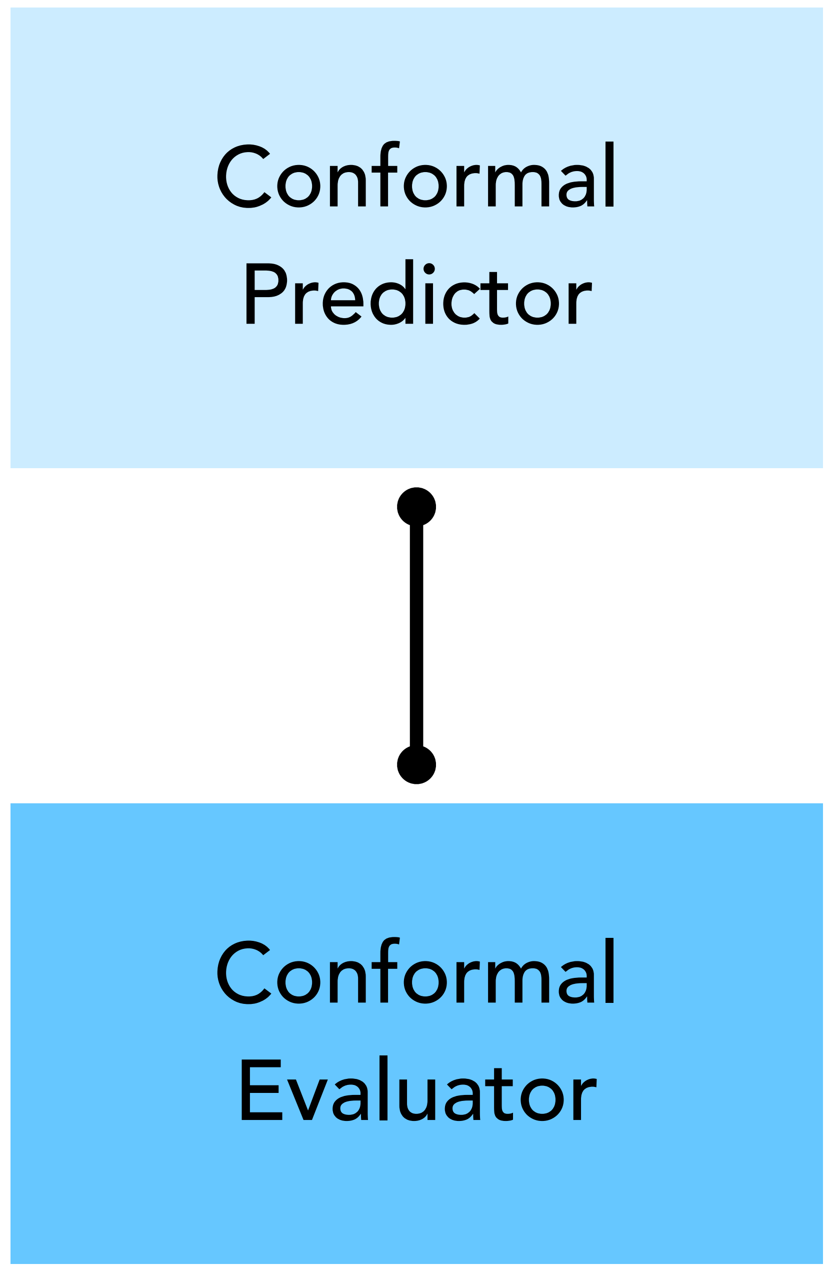
- New, sound and more flexible Conformal Evaluators
- Faster thresholding

**Extensive Evaluation**

- Android, Windows PE and PDF malware
- Different classifiers (SVM, RF, GBDT)


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

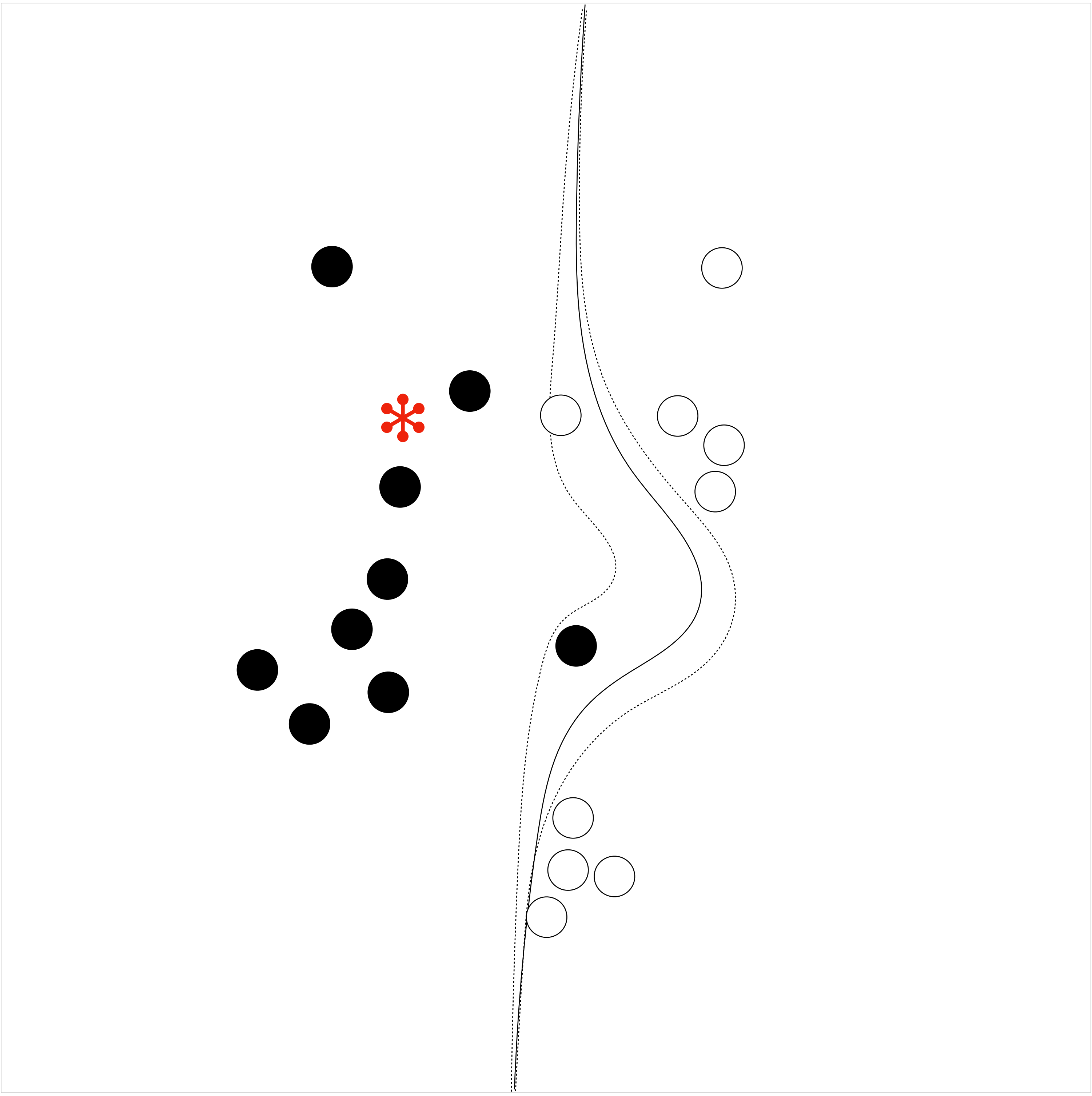
# Conformal Prediction and Evaluation




- 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/

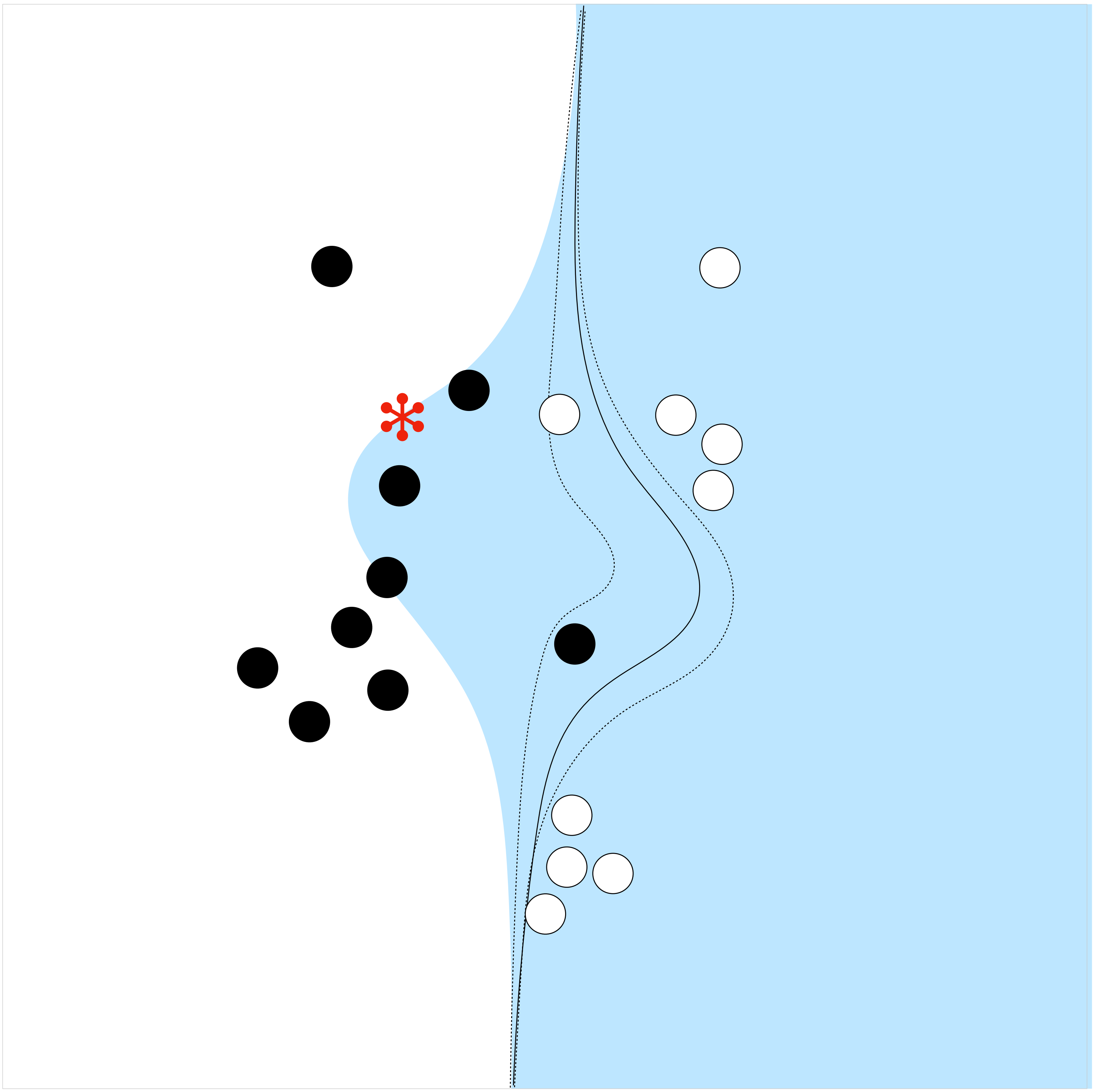
# Conformal Prediction and Non-Conformity Measure (NCM)



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/

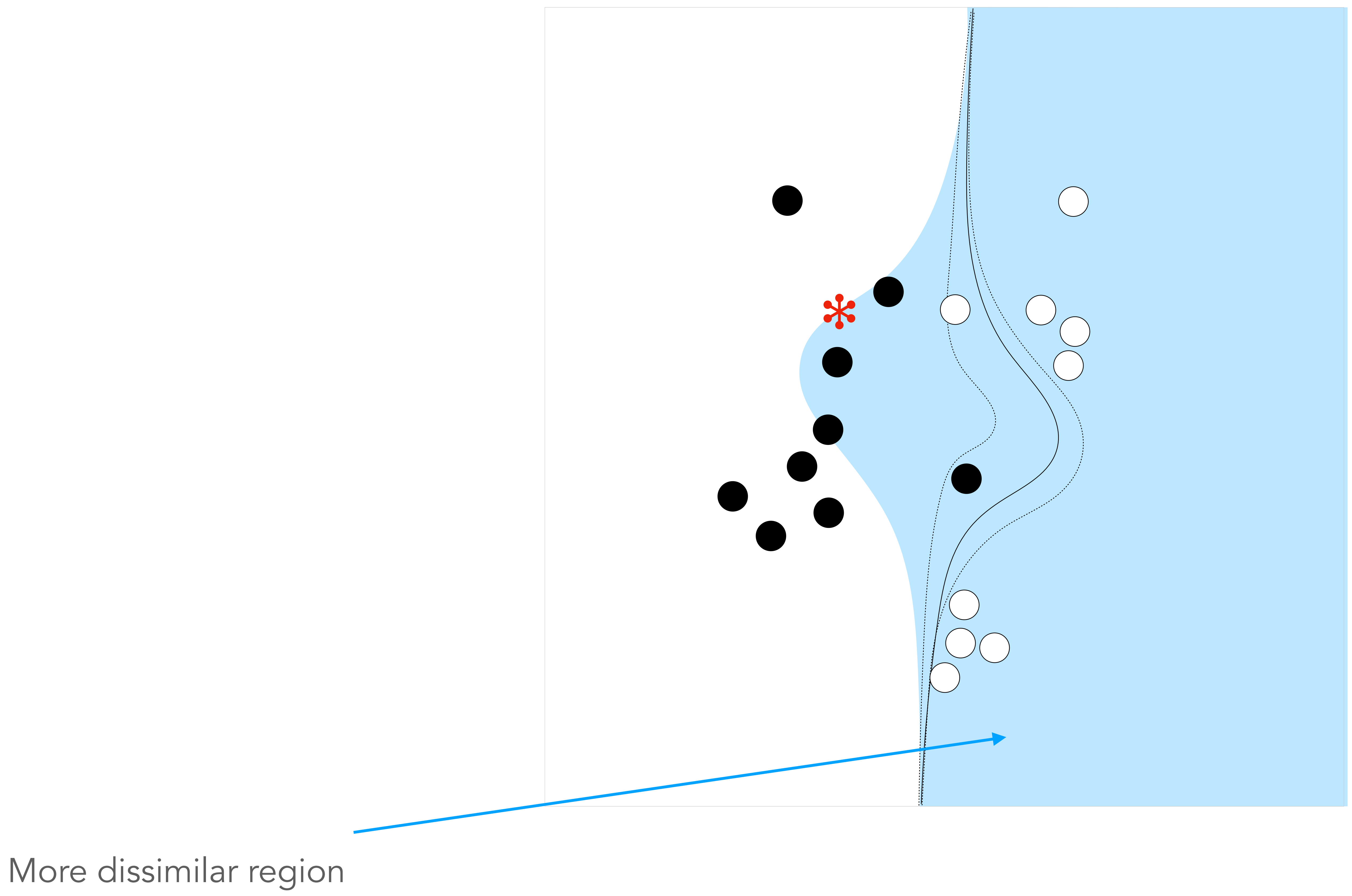
# Conformal Prediction and Non-Conformity Measure (NCM)



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/

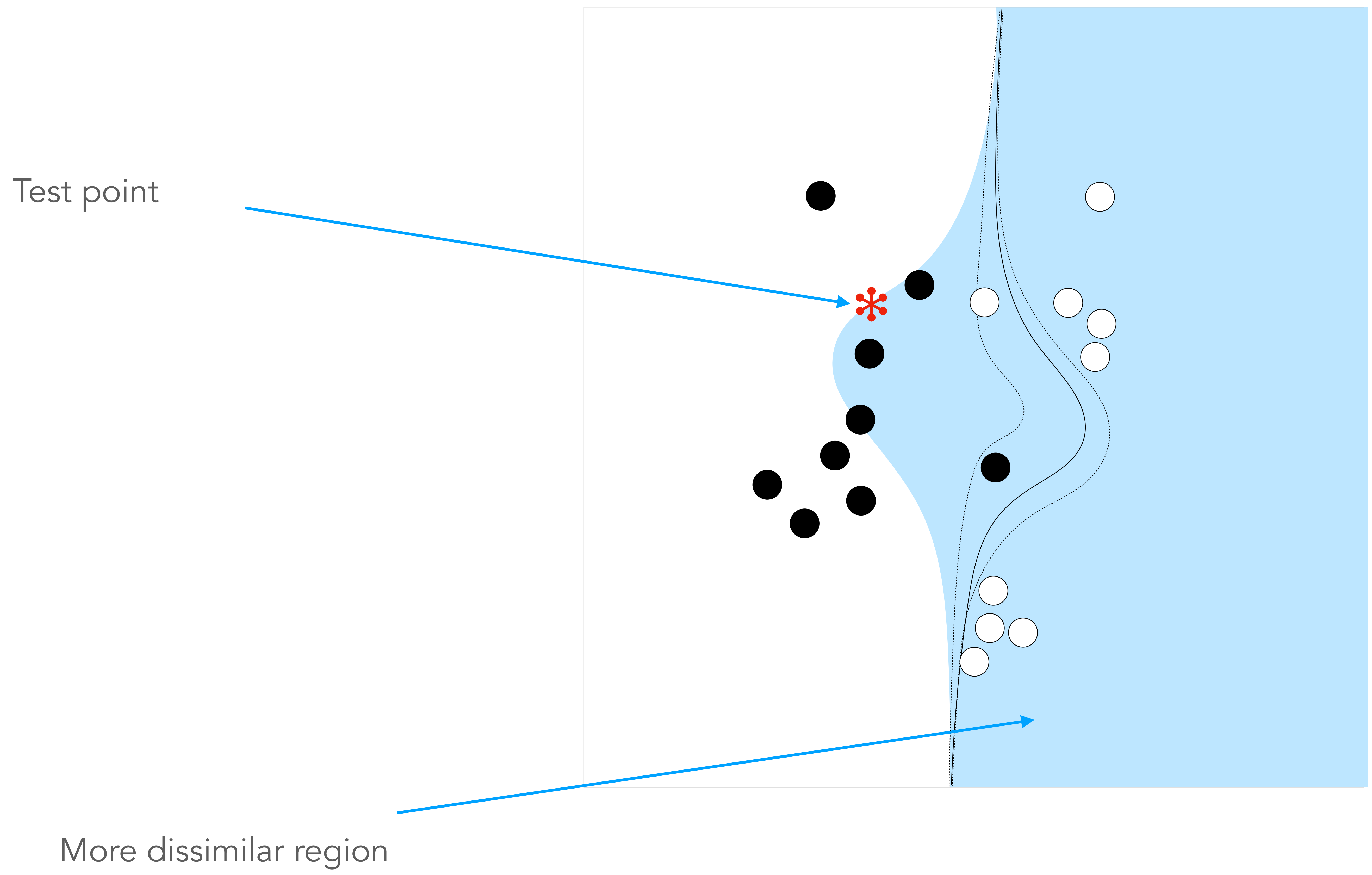
# Conformal Prediction and Non-Conformity Measure (NCM)




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/

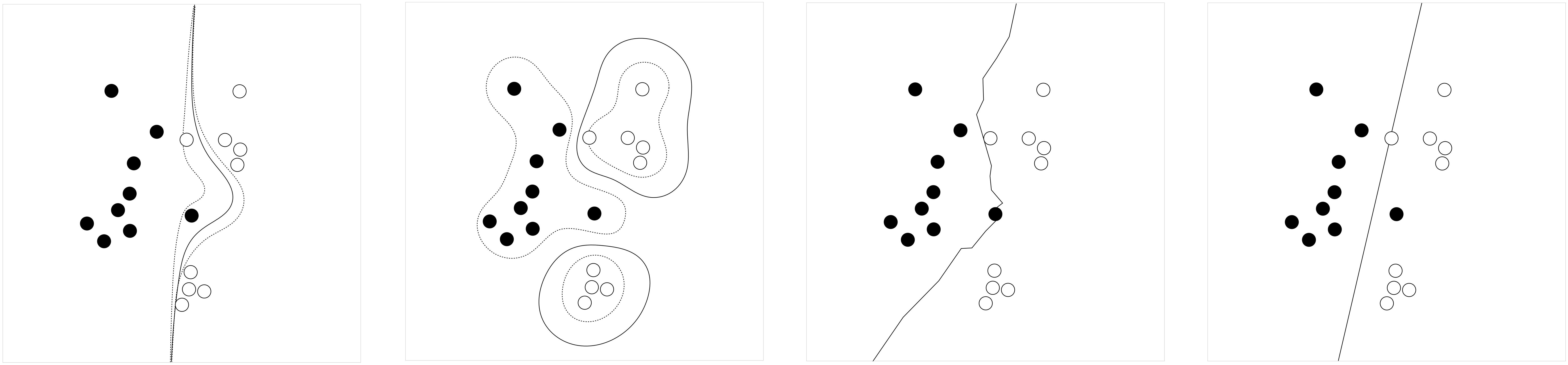
# Conformal Prediction and Non-Conformity Measure (NCM)



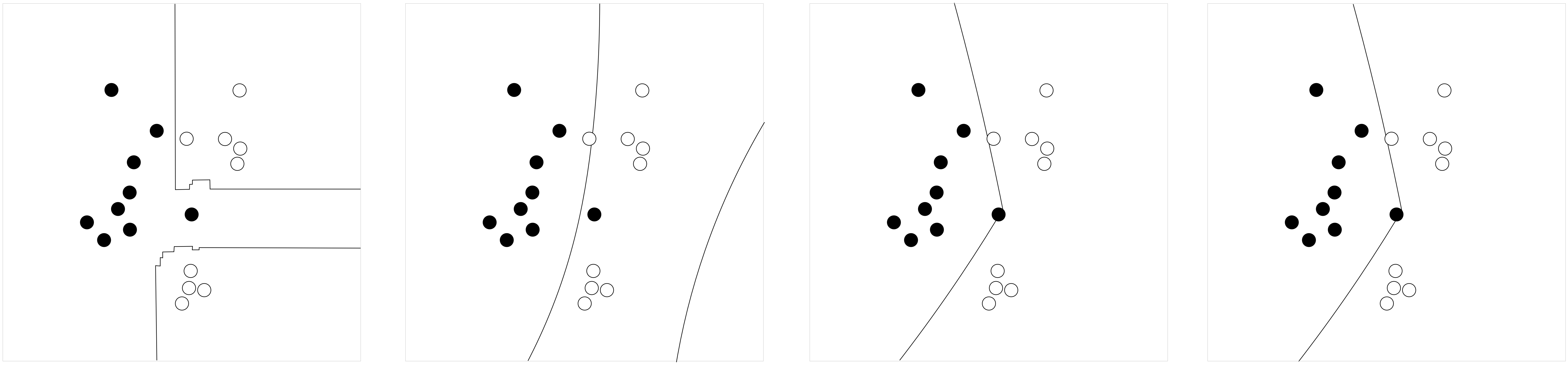

SVM Polynomial

[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)

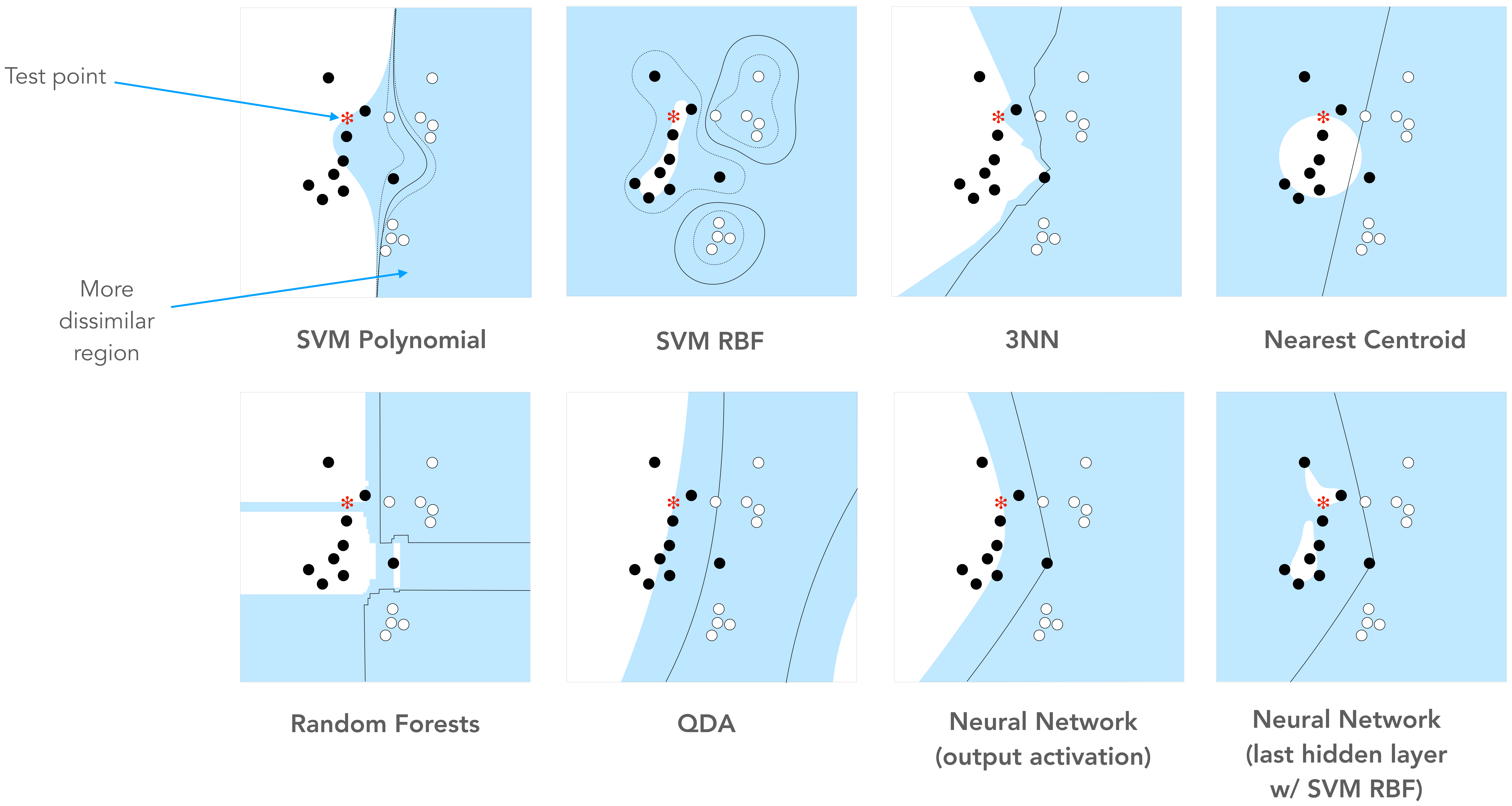


SVM Polynomial | SVM RBF | 3NN | Nearest Centroid

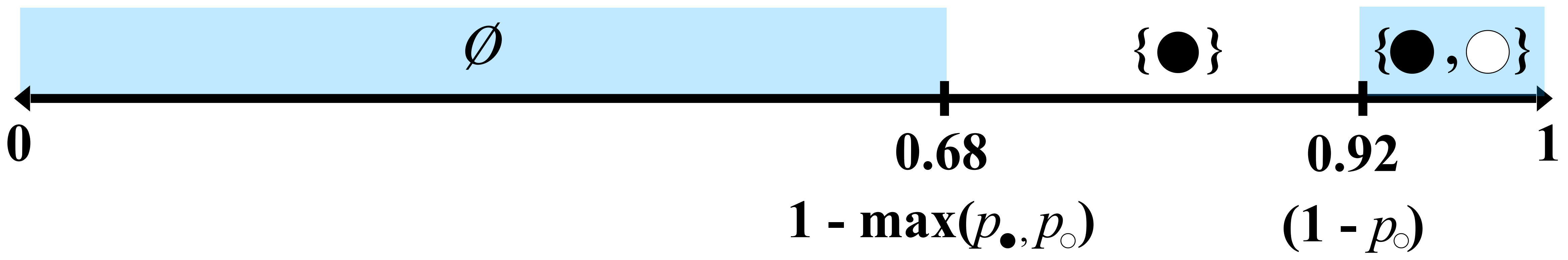


Random Forests | QDA | Neural Network (output activation) | Neural Network (last hidden layer w/ SVM RBF)

# Conformal Prediction and Non-Conformity Measure (NCM)



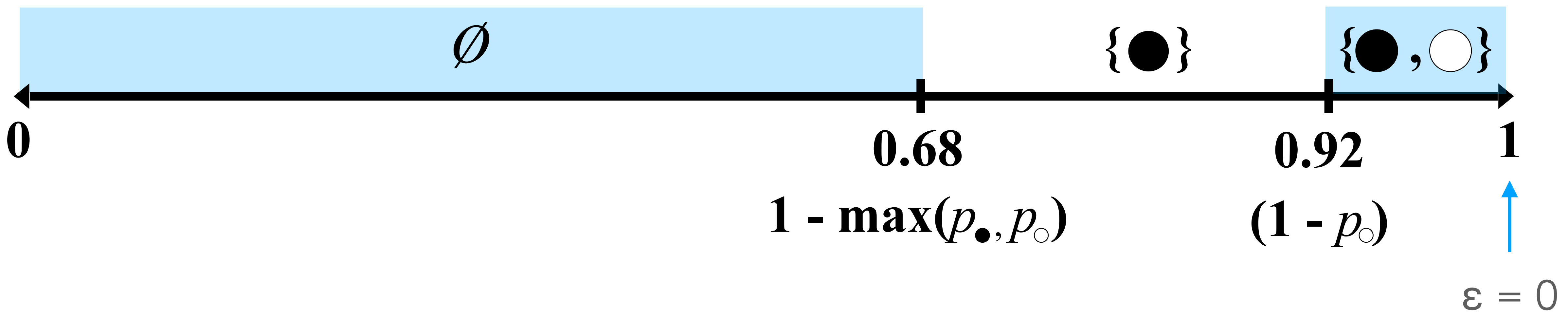
# Conformal Prediction vs Conformal Evaluation




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

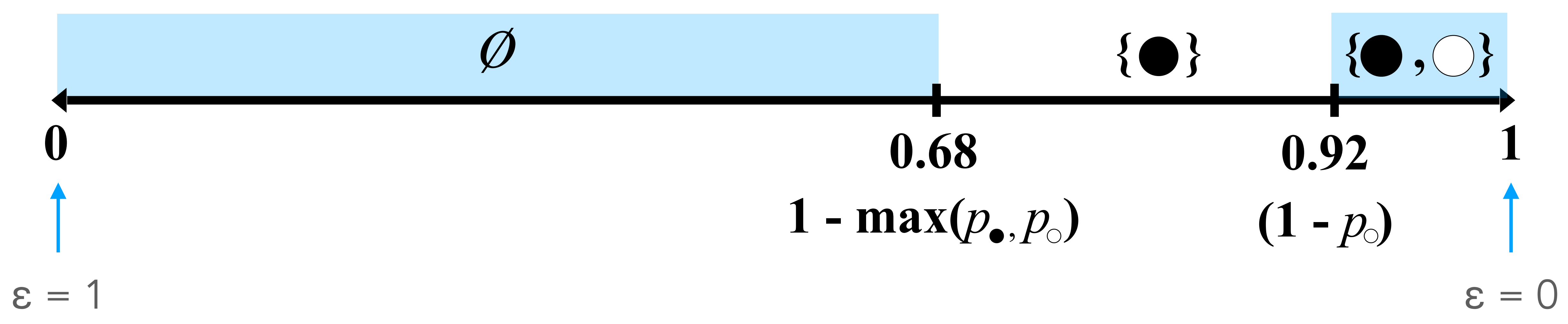
https://s2lab.cs.ucl.ac.uk/projects/transcend/

# Conformal Prediction vs Conformal Evaluation



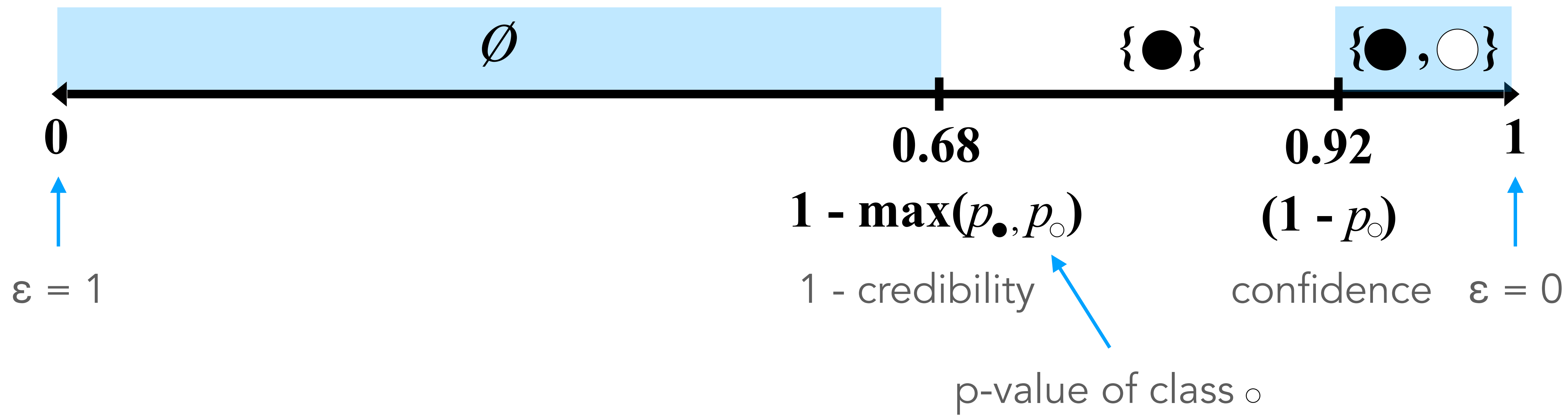

[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

# Conformal Prediction vs Conformal Evaluation




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

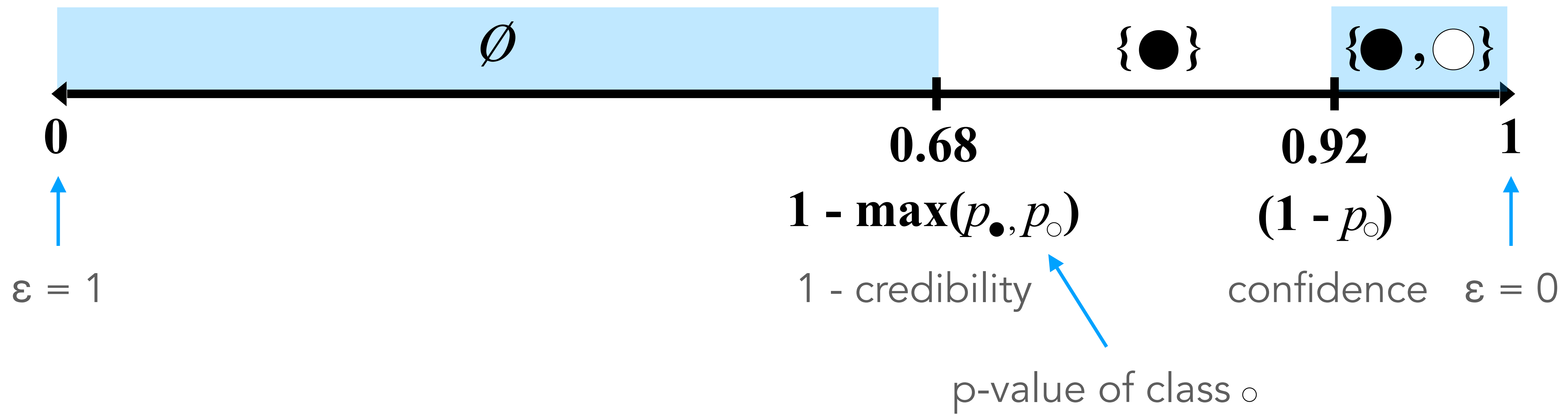
# Conformal Prediction vs Conformal Evaluation




[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 vs Conformal Evaluation

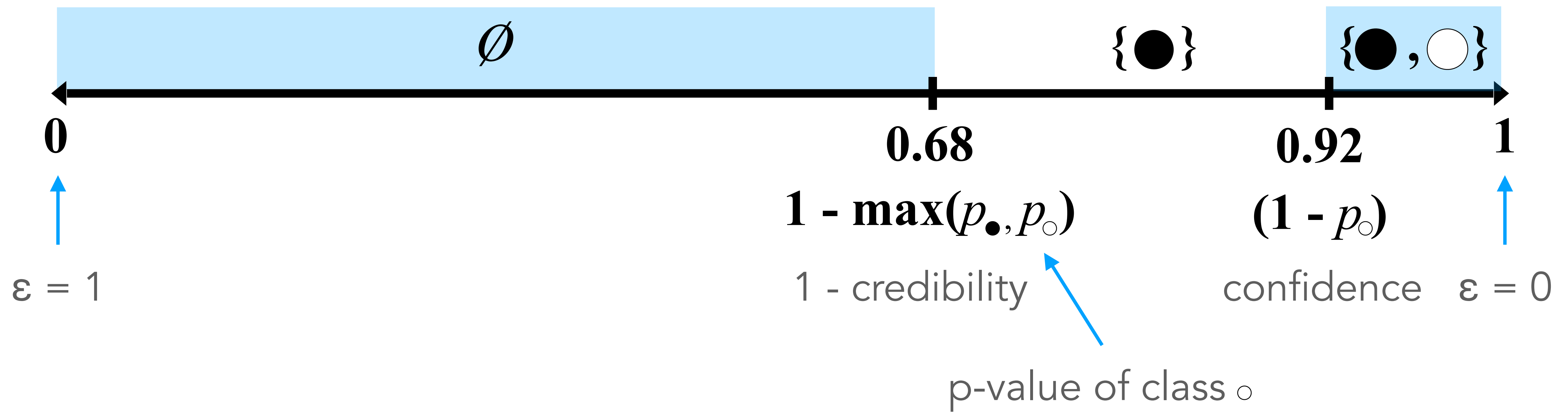
- Low credibility means high probability of an impossible result
- This means assumptions could have been violated — drift!




[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 vs Conformal Evaluation

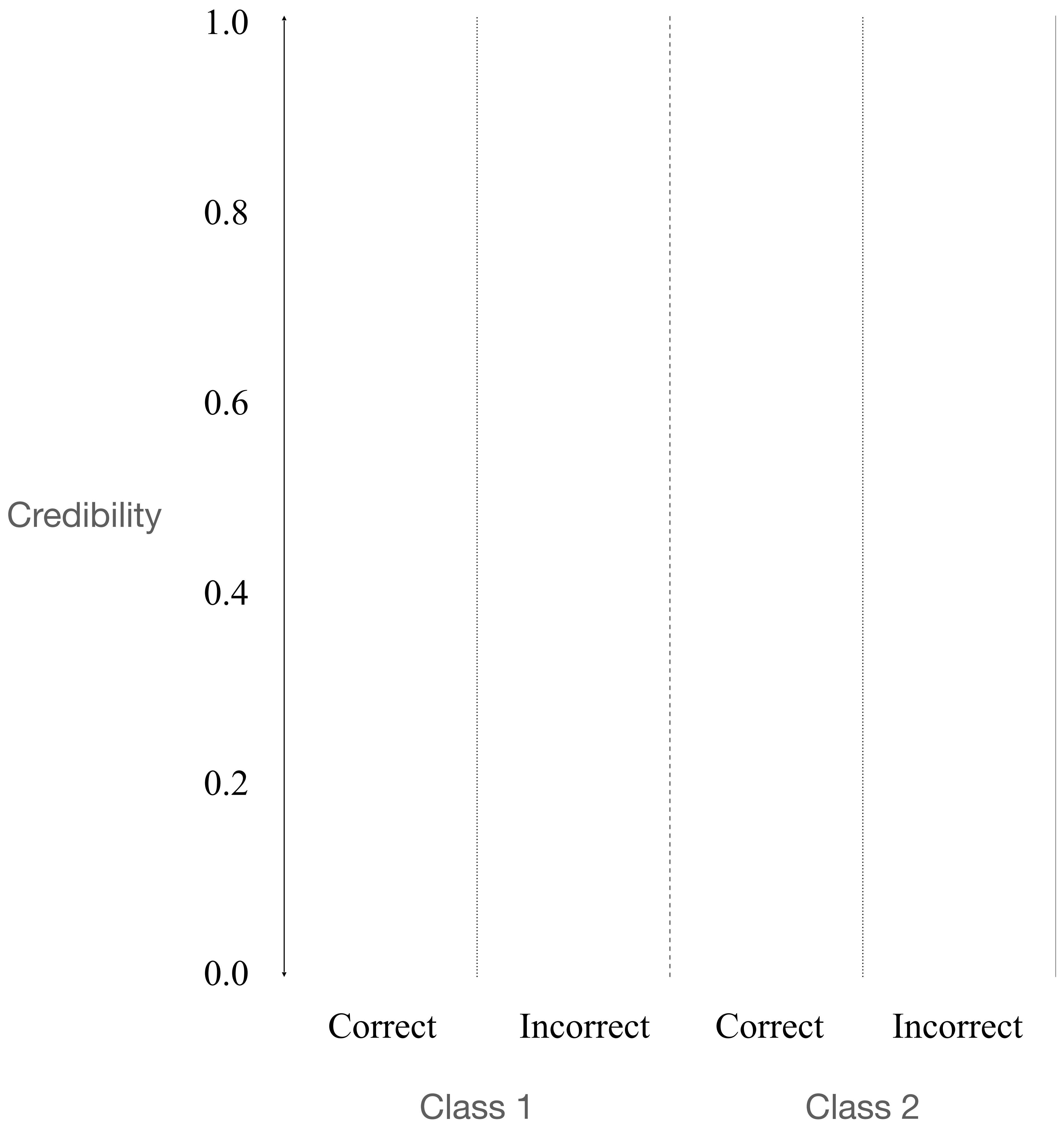
- Low credibility means high probability of an impossible result
- This means assumptions could have been violated — drift!




- Whereas CPs predict, CEs evaluate predictions using the same statistical tools as a signal for concept drift

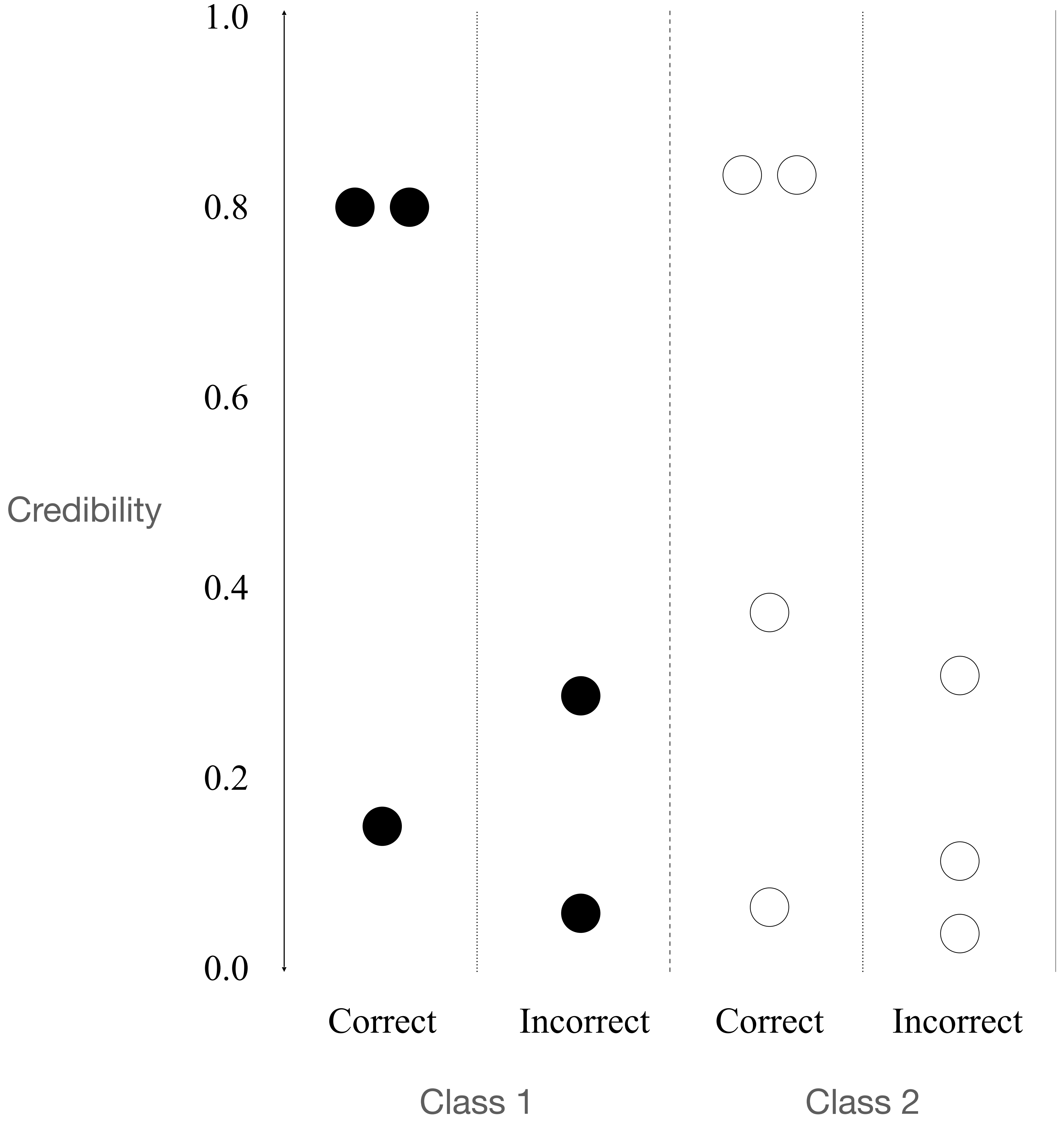
[IEEE S&P 2022] Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift
https://s2lab.cs.ucl.ac.uk/projects/transcend/

# Transcend Calibration



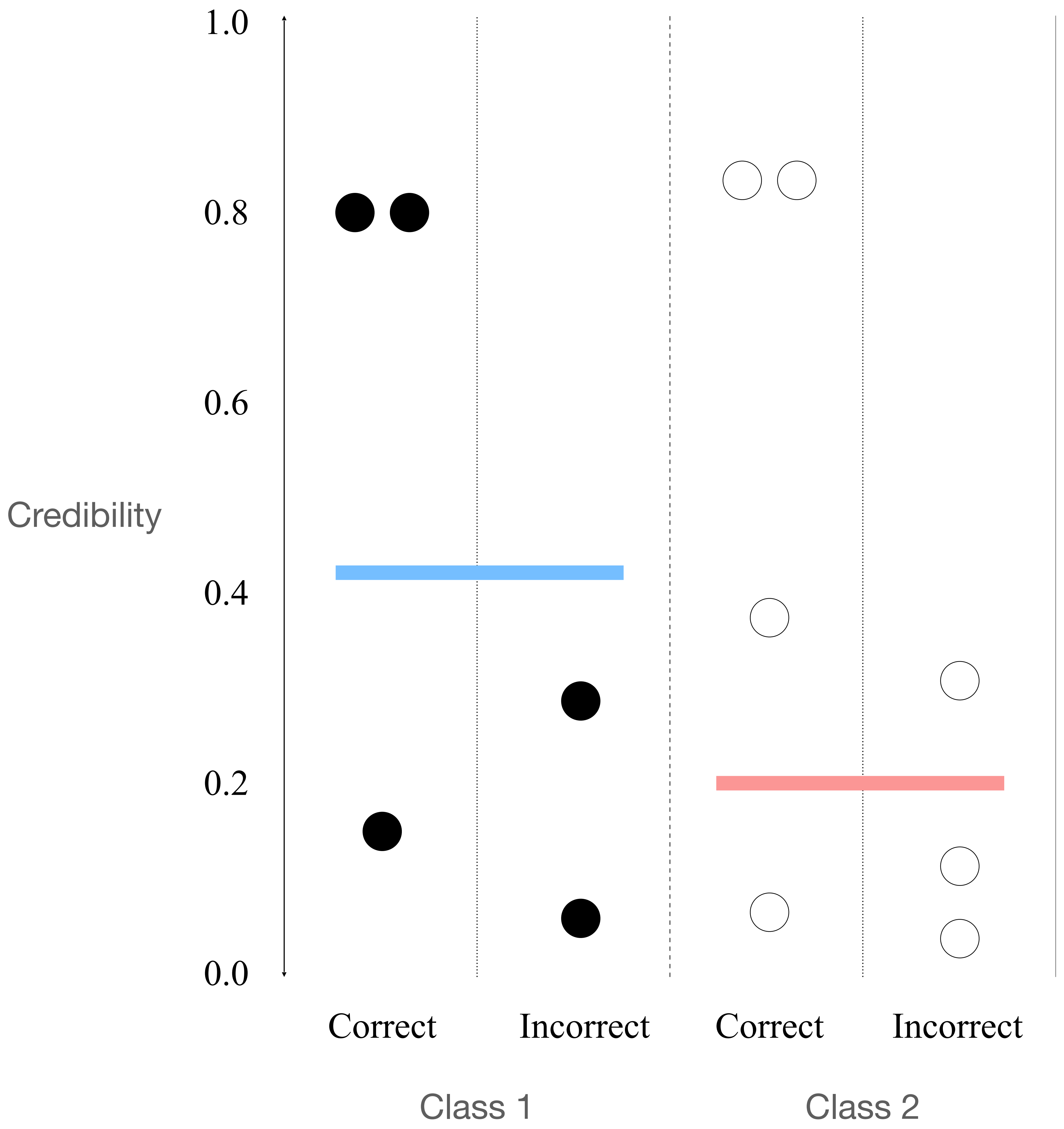

- **How much drift is too much?**
- Produce a threshold for each class
- Optimize cost vs performance on training and calibration sets
- Maximise separation between credibility of correct and incorrect decisions

# Transcend Calibration



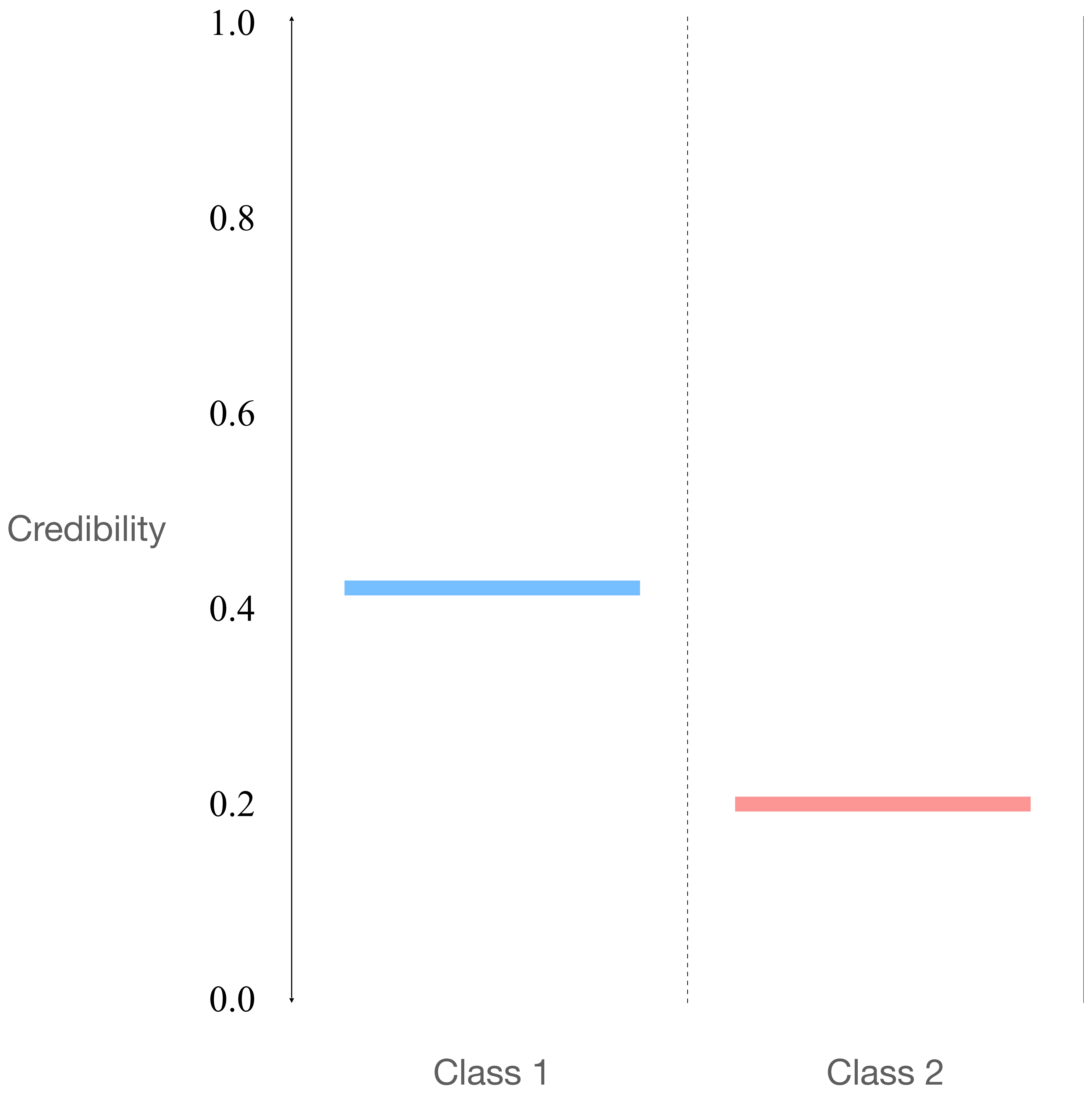

- **How much drift is too much?**
- Produce a threshold for each class
- Optimize cost vs performance on training and calibration sets
- Maximise separation between credibility of correct and incorrect decisions

# Transcend Calibration



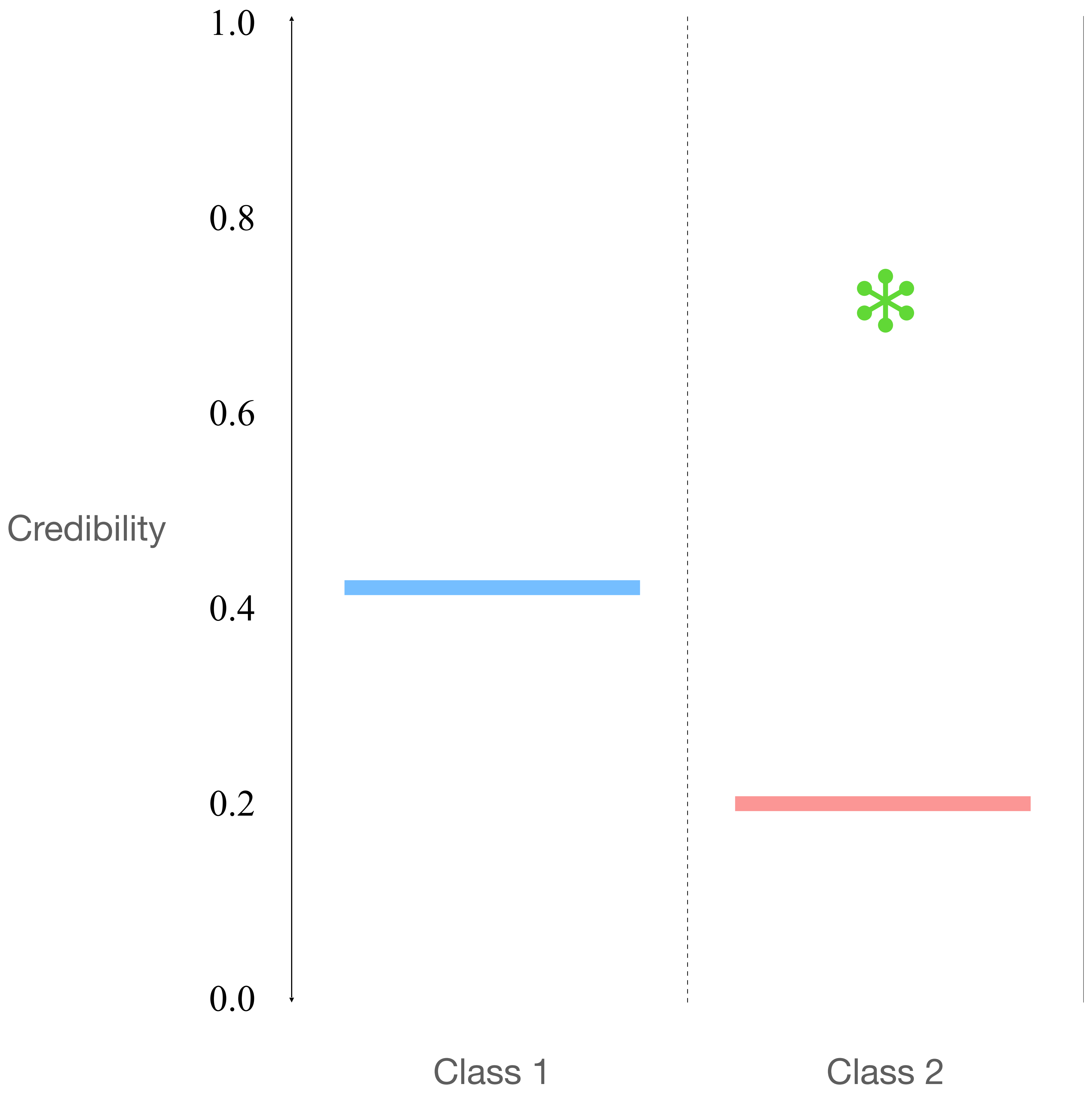

- **How much drift is too much?**
- Produce a threshold for each class
- Optimize cost vs performance on training and calibration sets
- Maximise separation between credibility of correct and incorrect decisions

# Transcend at Test Time



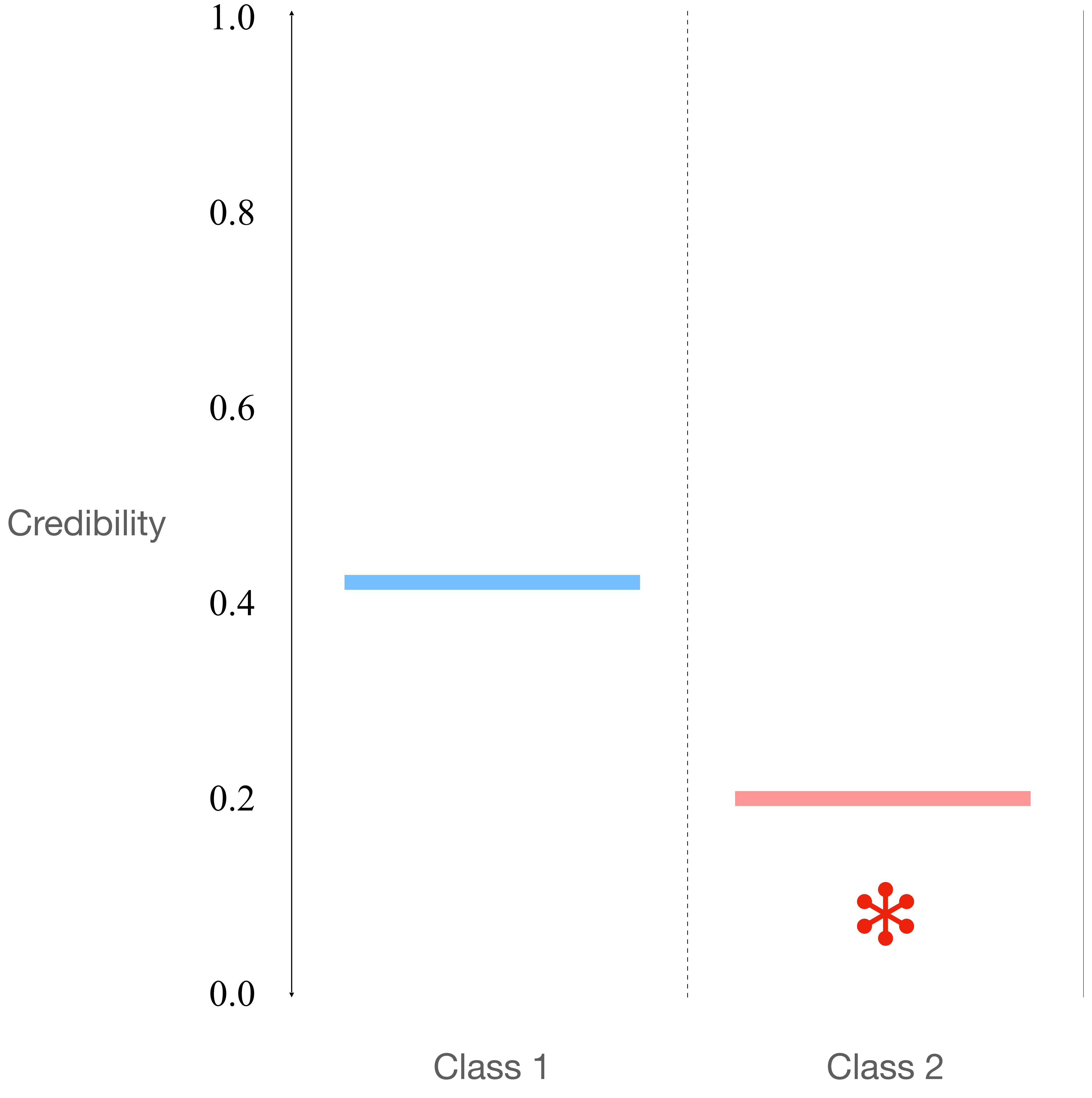

- Credibilities of new examples are compared against the threshold of their predicted class
- Above = keep the prediction
- Below = reject the prediction

# Transcend at Test Time



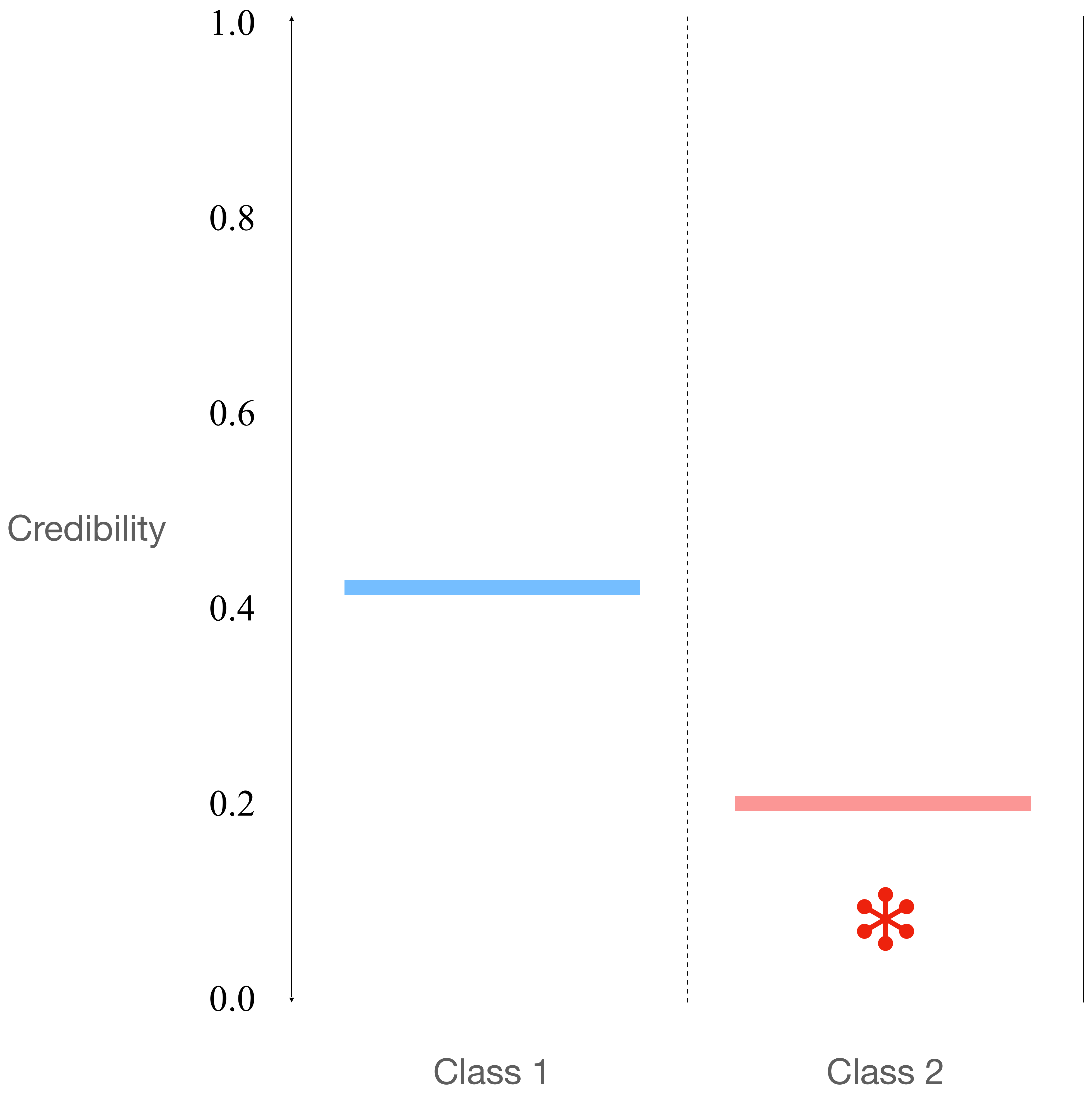

- Credibilities of new examples are compared against the threshold of their predicted class
- Above = keep the prediction
- Below = reject the prediction

# Transcend at Test Time

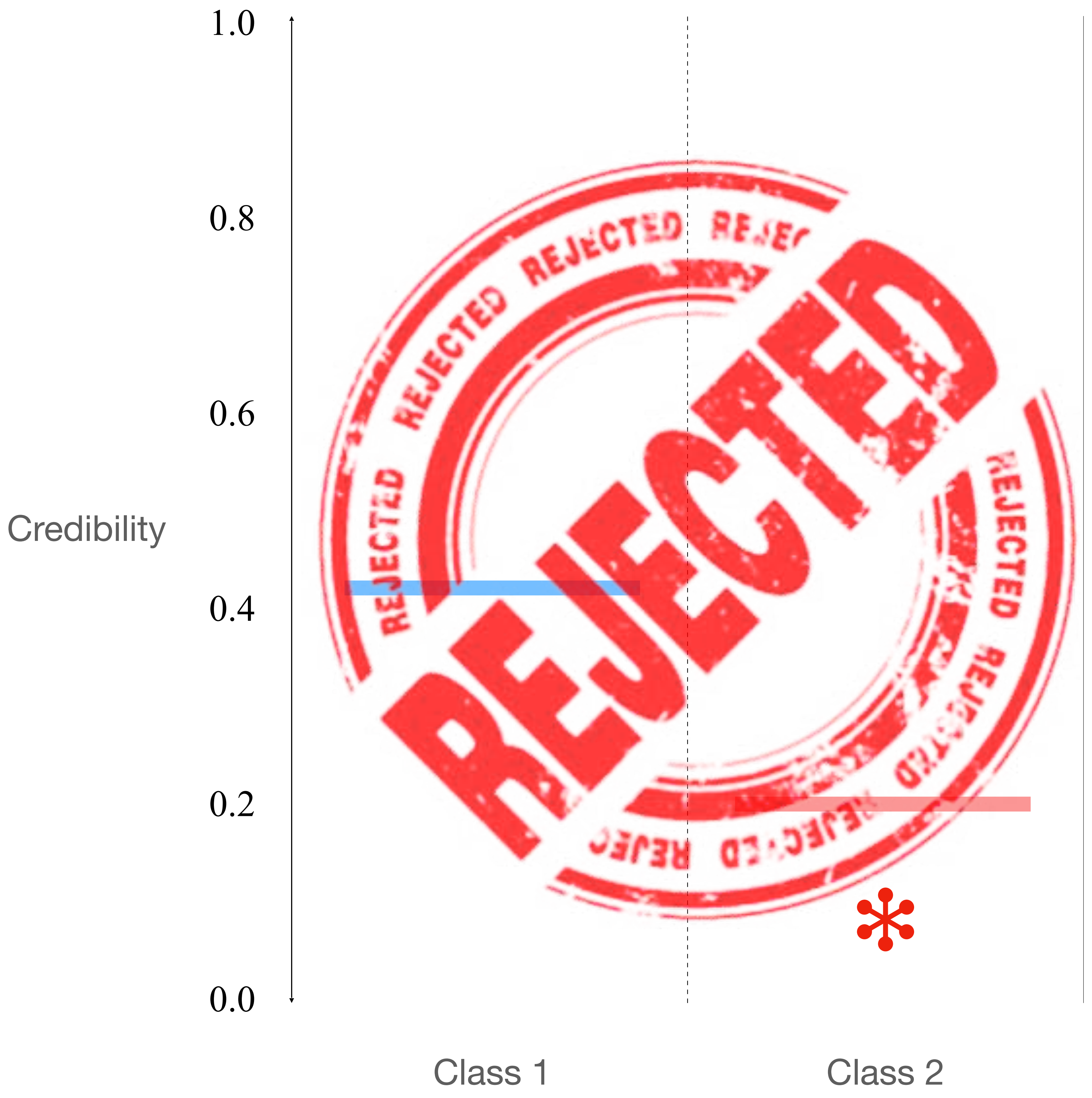



- Credibilities of new examples are compared against the threshold of their predicted class
- Above = keep the prediction
- Below = reject the prediction

# Rejection Cost



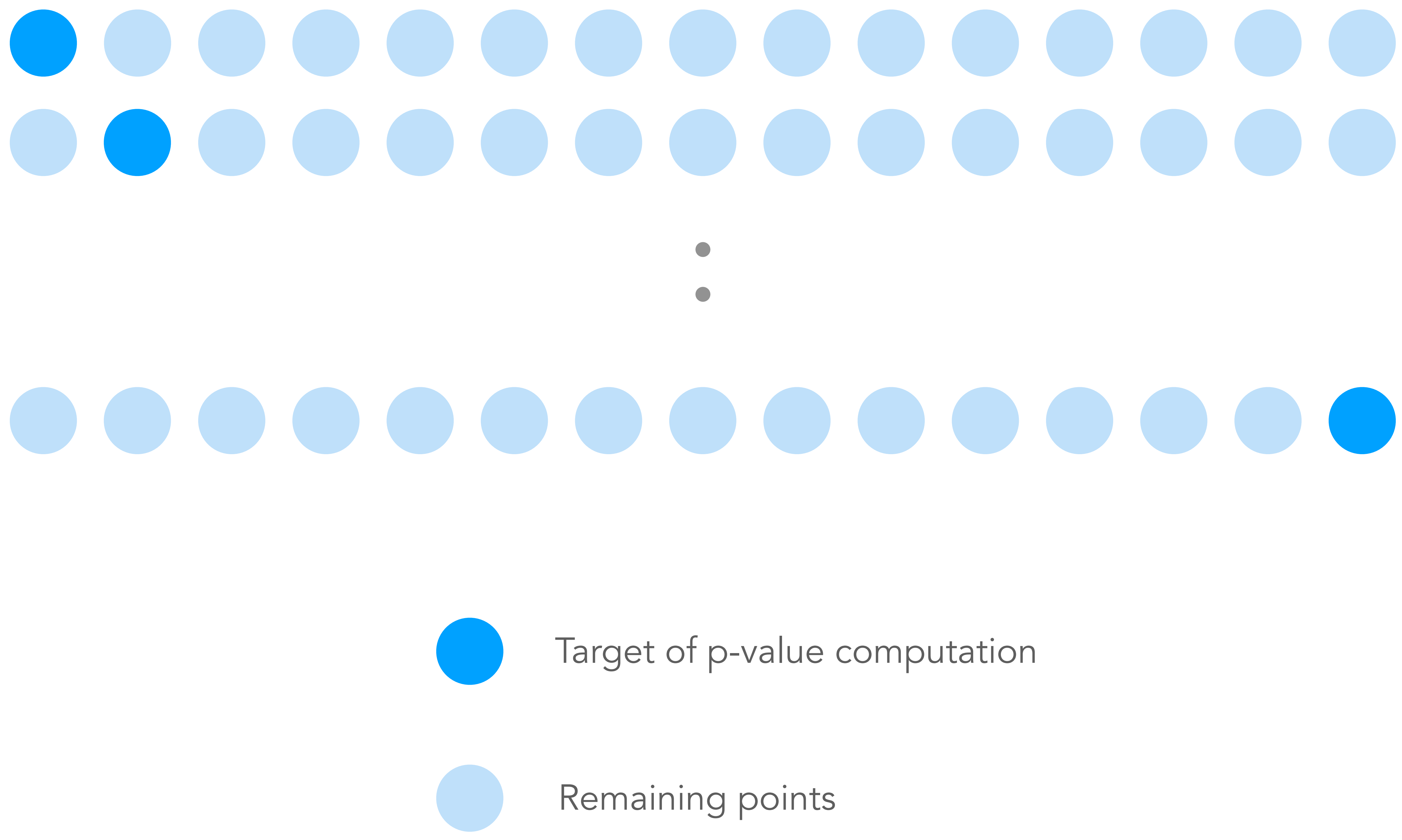
# Rejection Cost




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

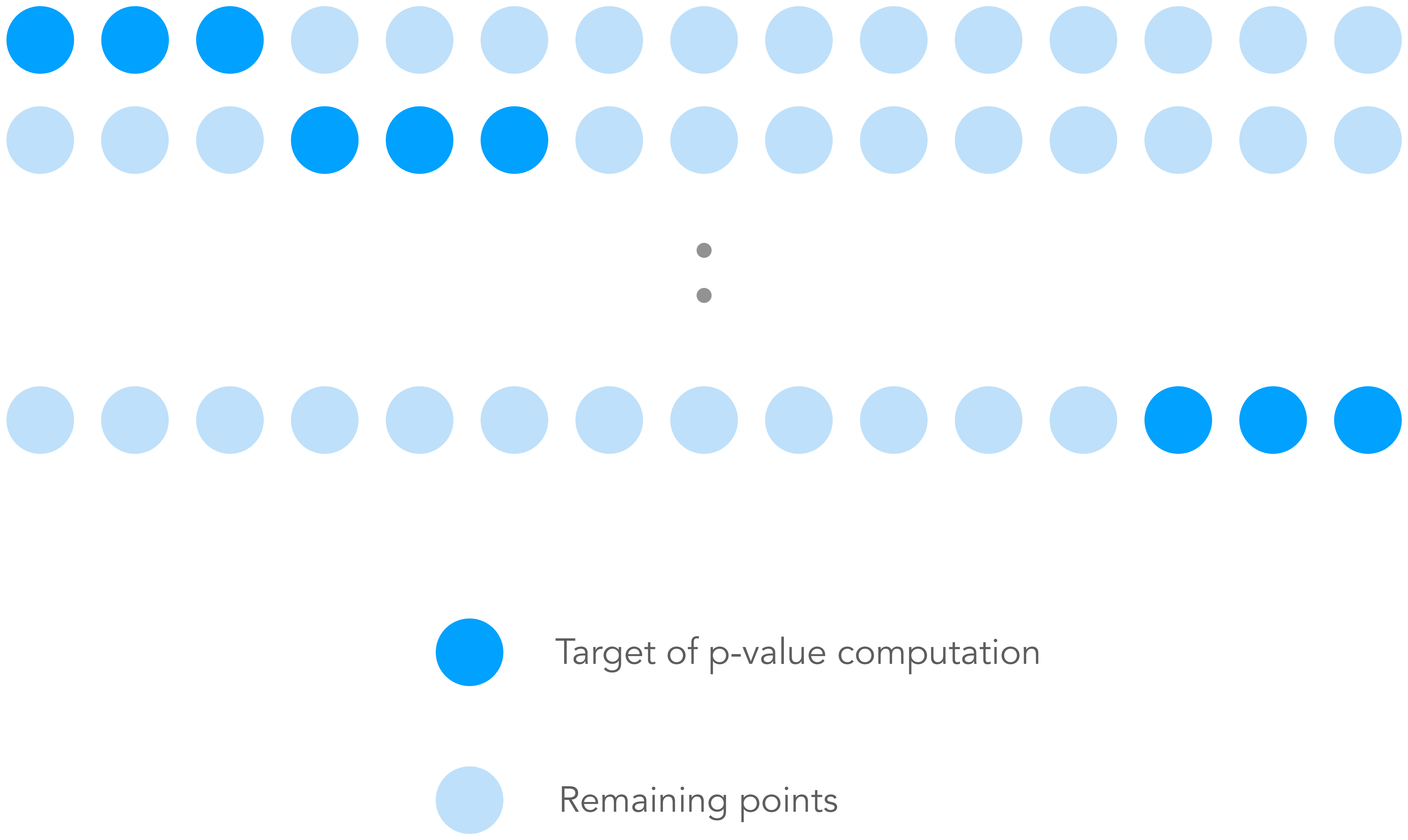
* [AISec 2021] Investigating Labelless Drift Adaptation for Malware Detection
* [AISec 2021] INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection

# The Cost of Transductive Conformal Evaluators



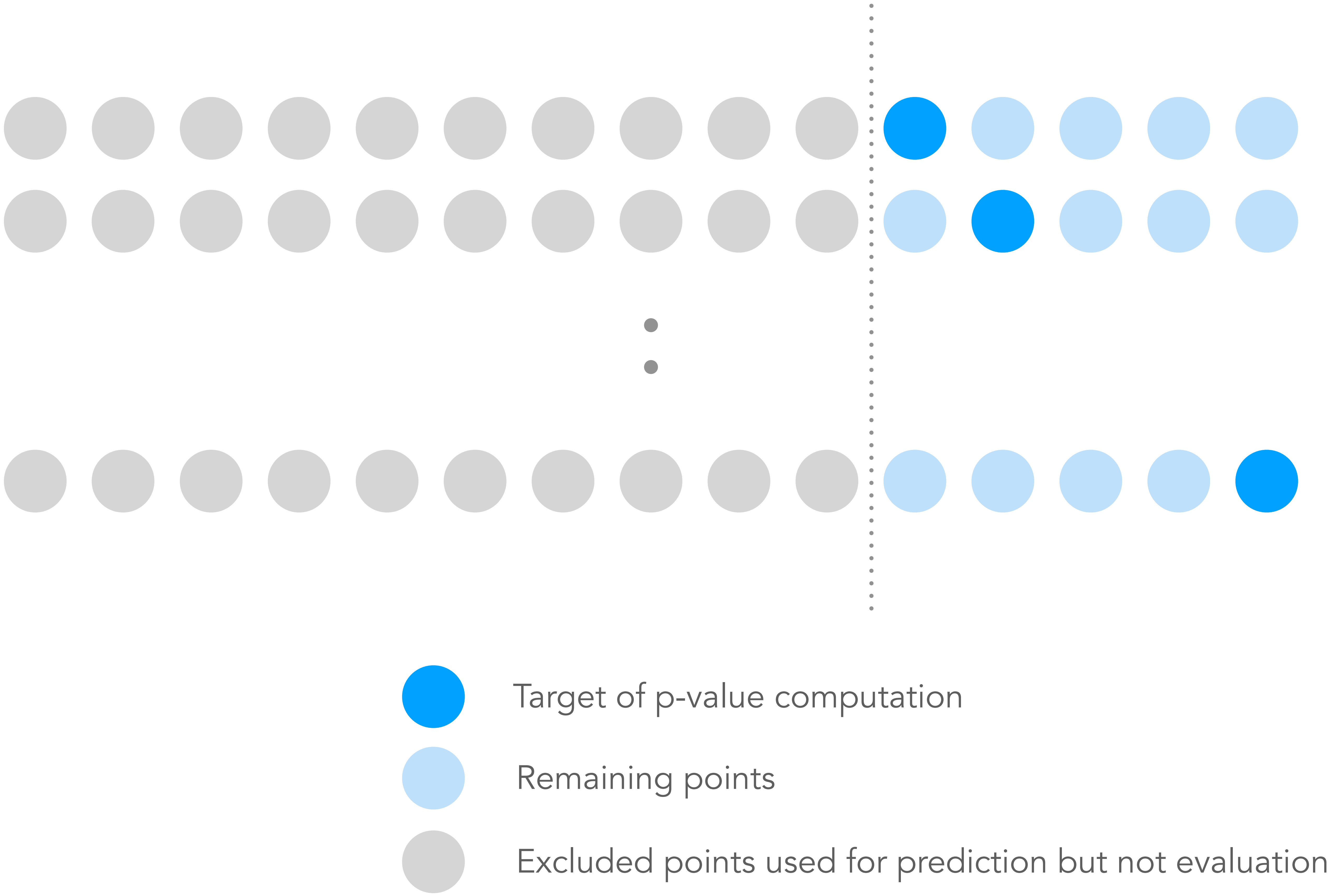

- Underlying classifier retrained for every training point
- Rooted in CP theory
- Often computationally infeasible

# Approximate TCE



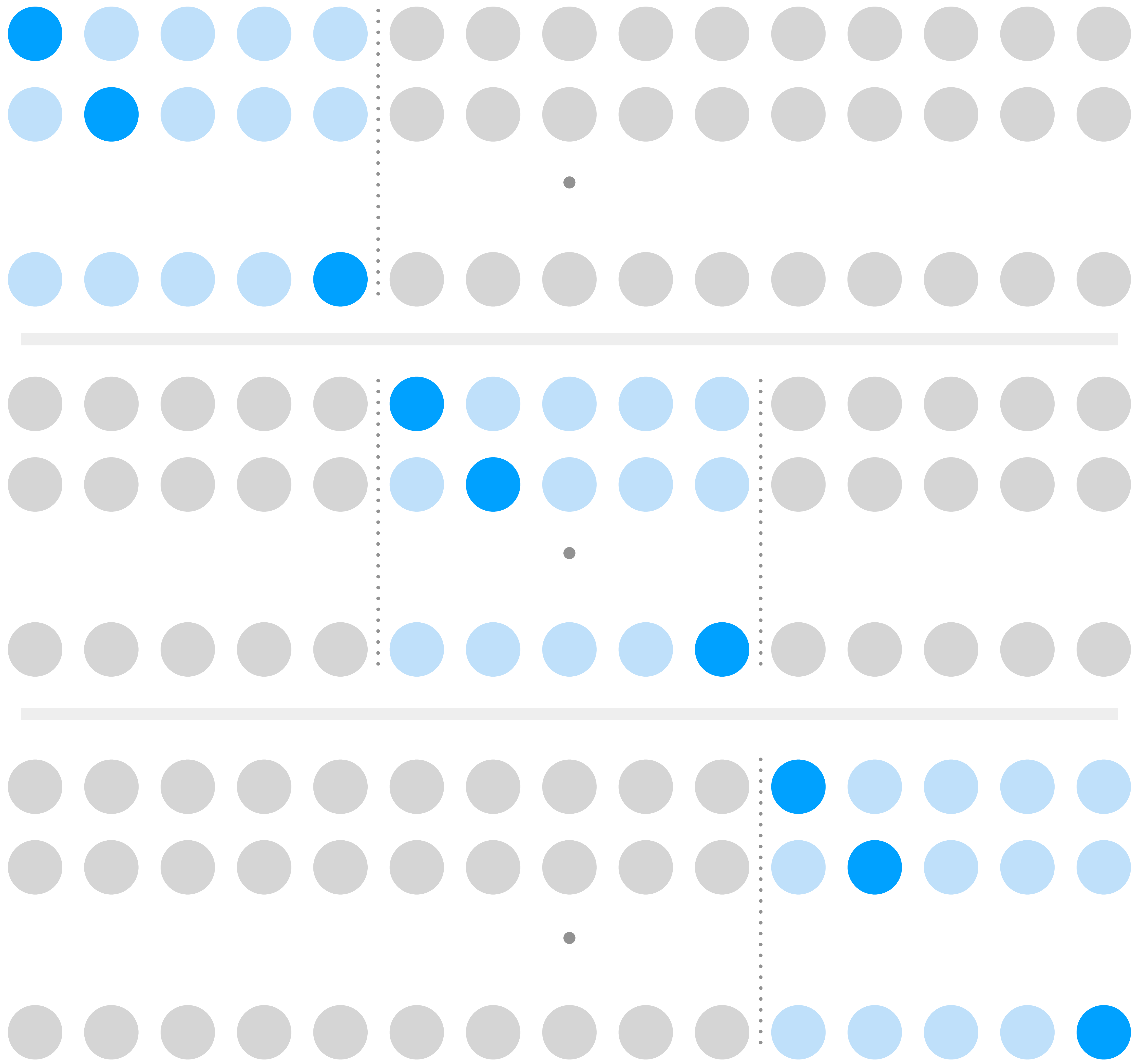

- First attempt to improve on the TCE
- P-values computed in batches
- Relies on unsound assumption

# 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

# Experimental Setup

# Experimental Setup

**Android**

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

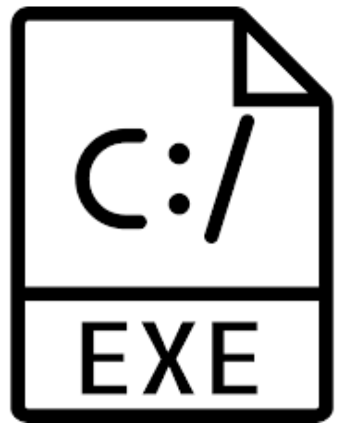
# Experimental Setup

**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

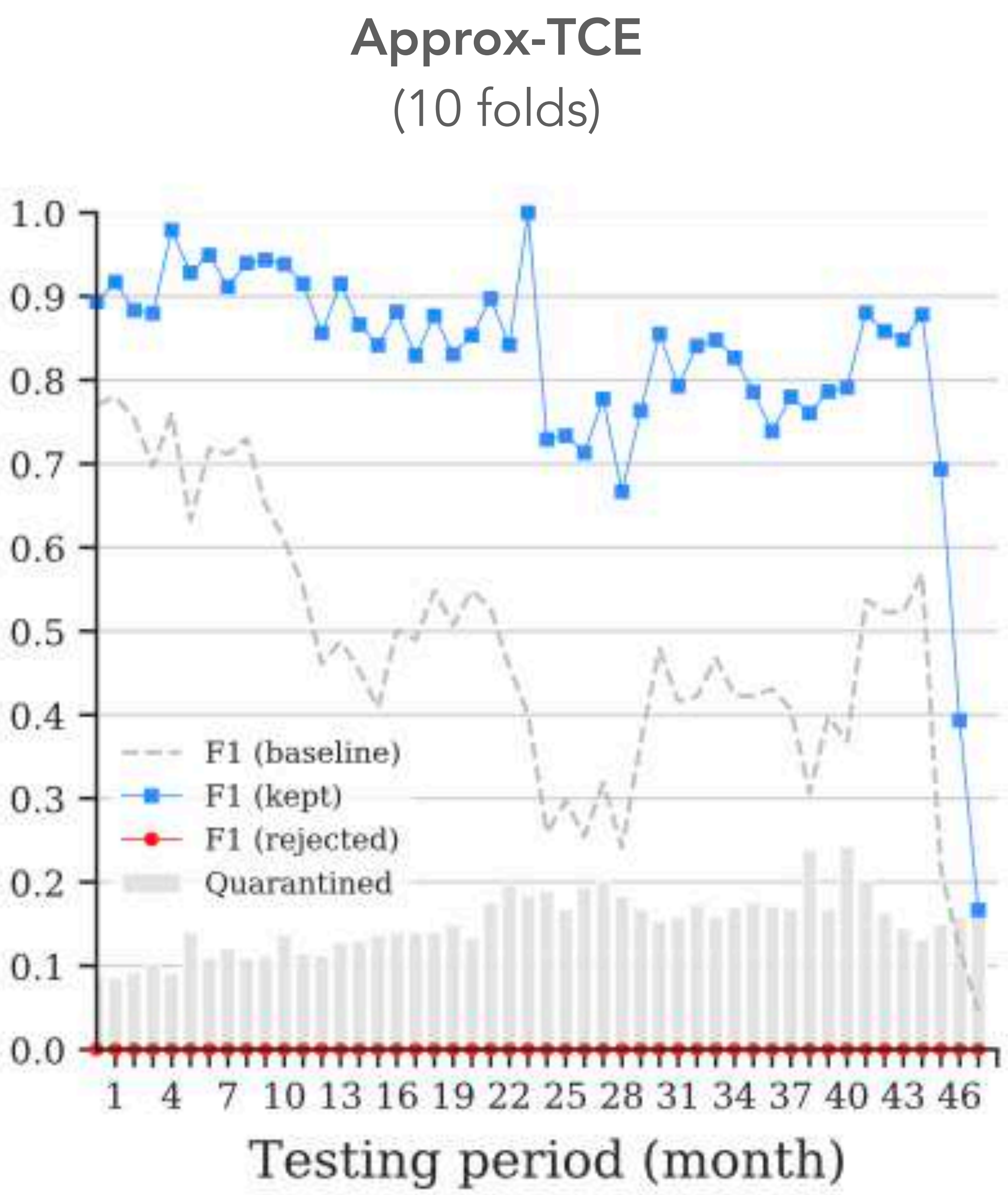
### Thresholding Optimization

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

# 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

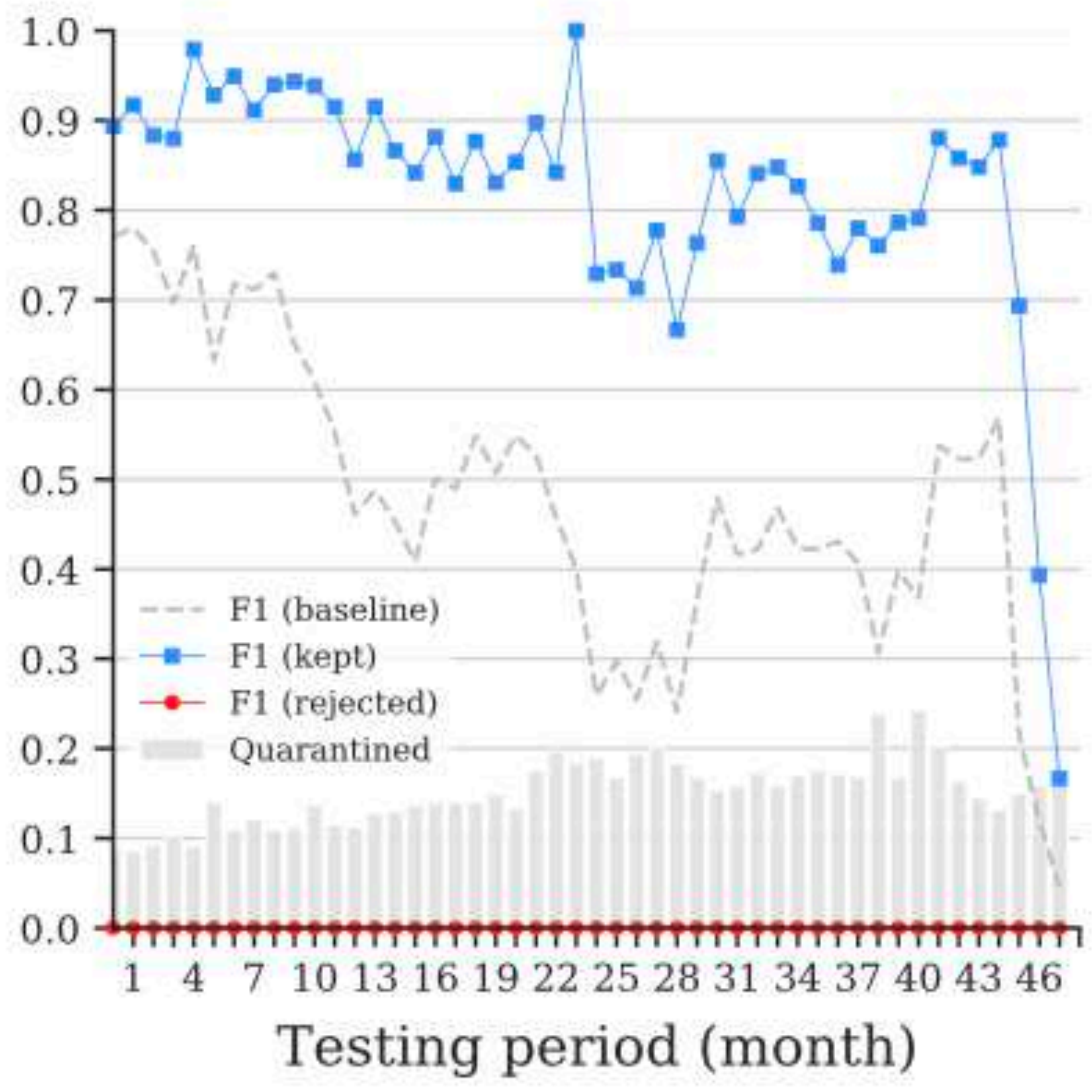



| | |
|---|---|
| AUT(F1, kept): | 0.82 |
| AUT(F1, rejected): | 0.00 |
| CPU hours: | 46.1 |

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

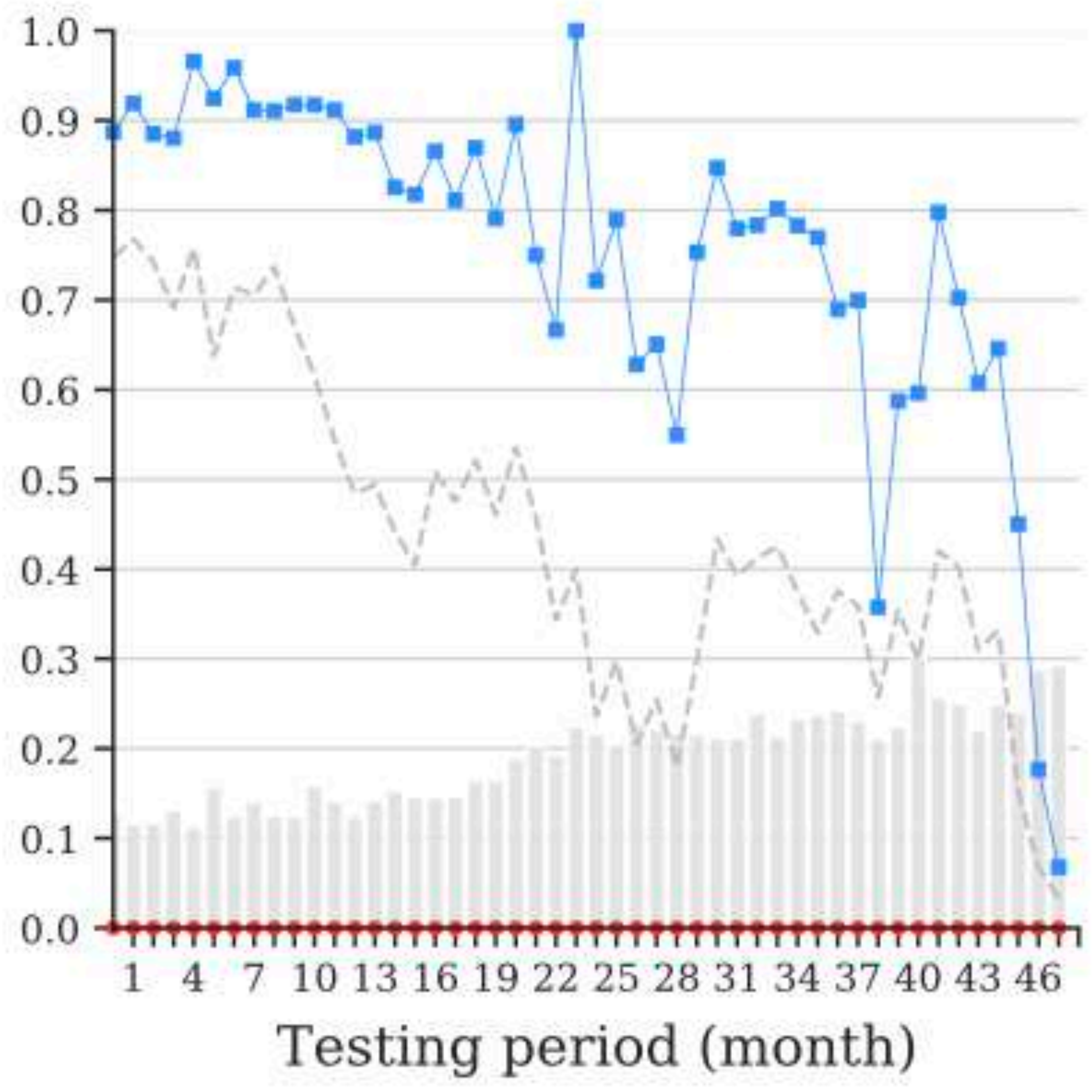
# Results: Rejection Performance

Approx-TCE
(10 folds)




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

ICE
(0.33 calibration split)

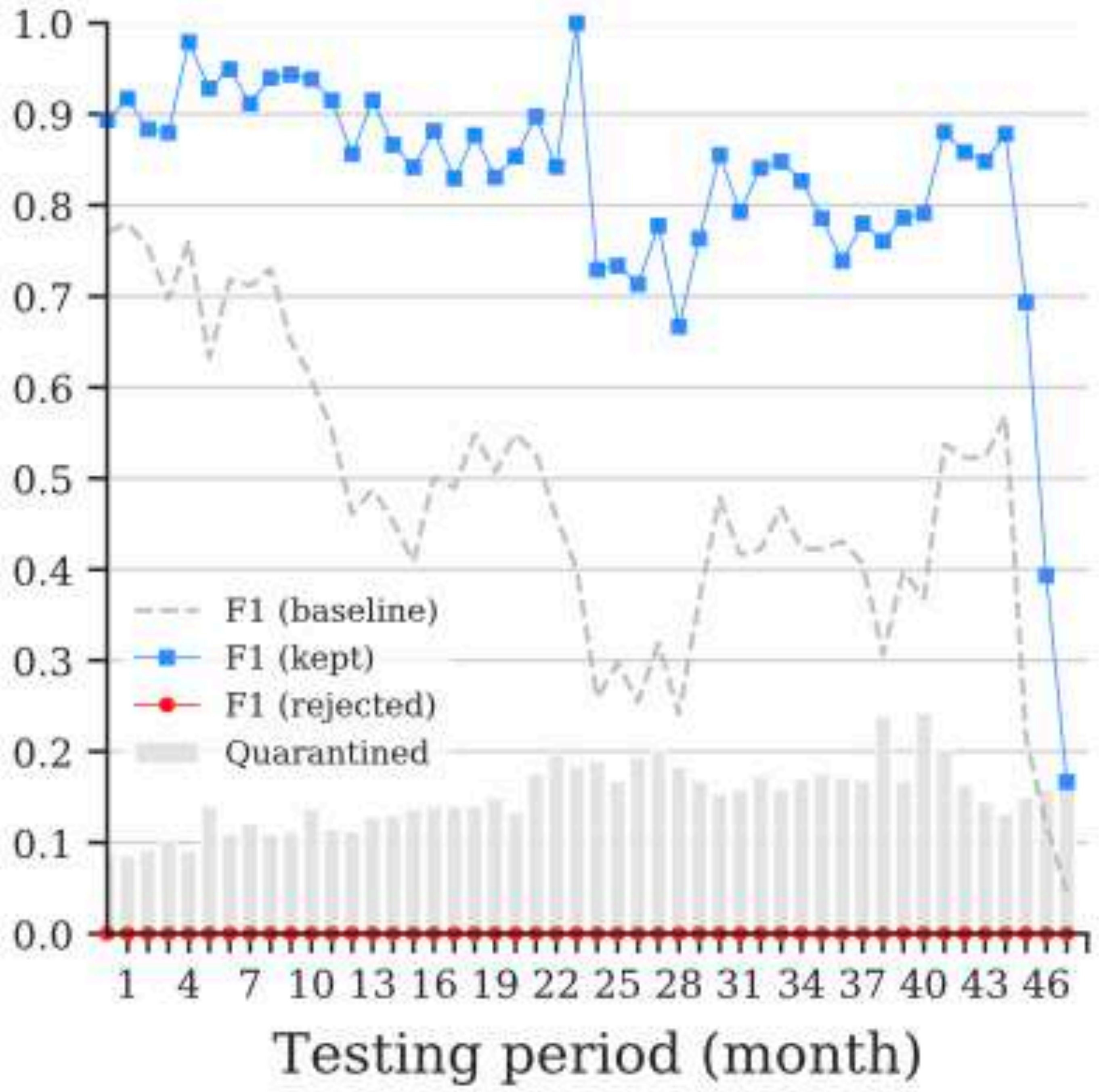



```
AUT(f1, kept):       0.76
AUT(f1, rejected):   0.00
CPU hours:           11.5
```

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

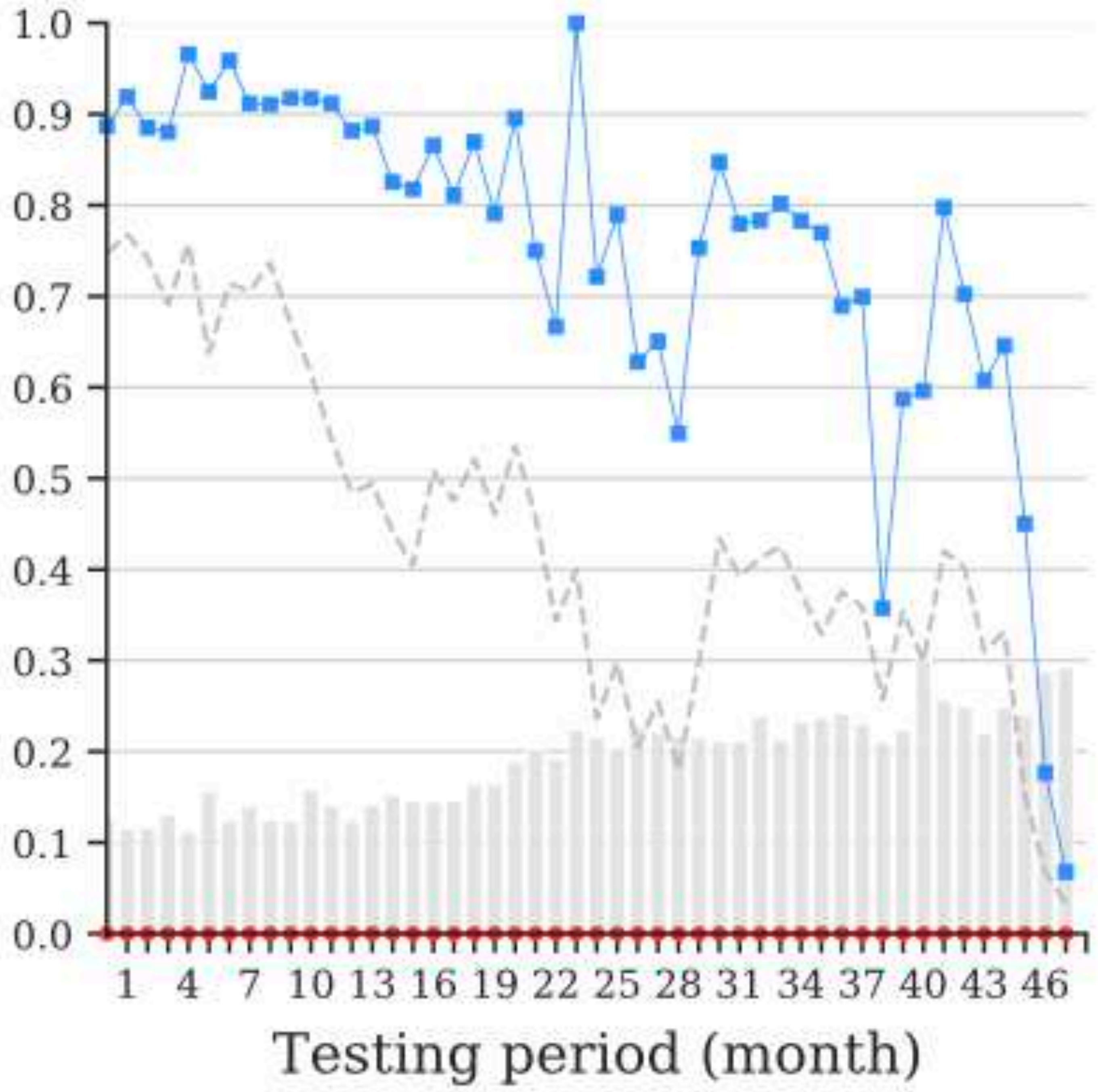
# Results: Rejection Performance

**Approx-TCE**
(10 folds)



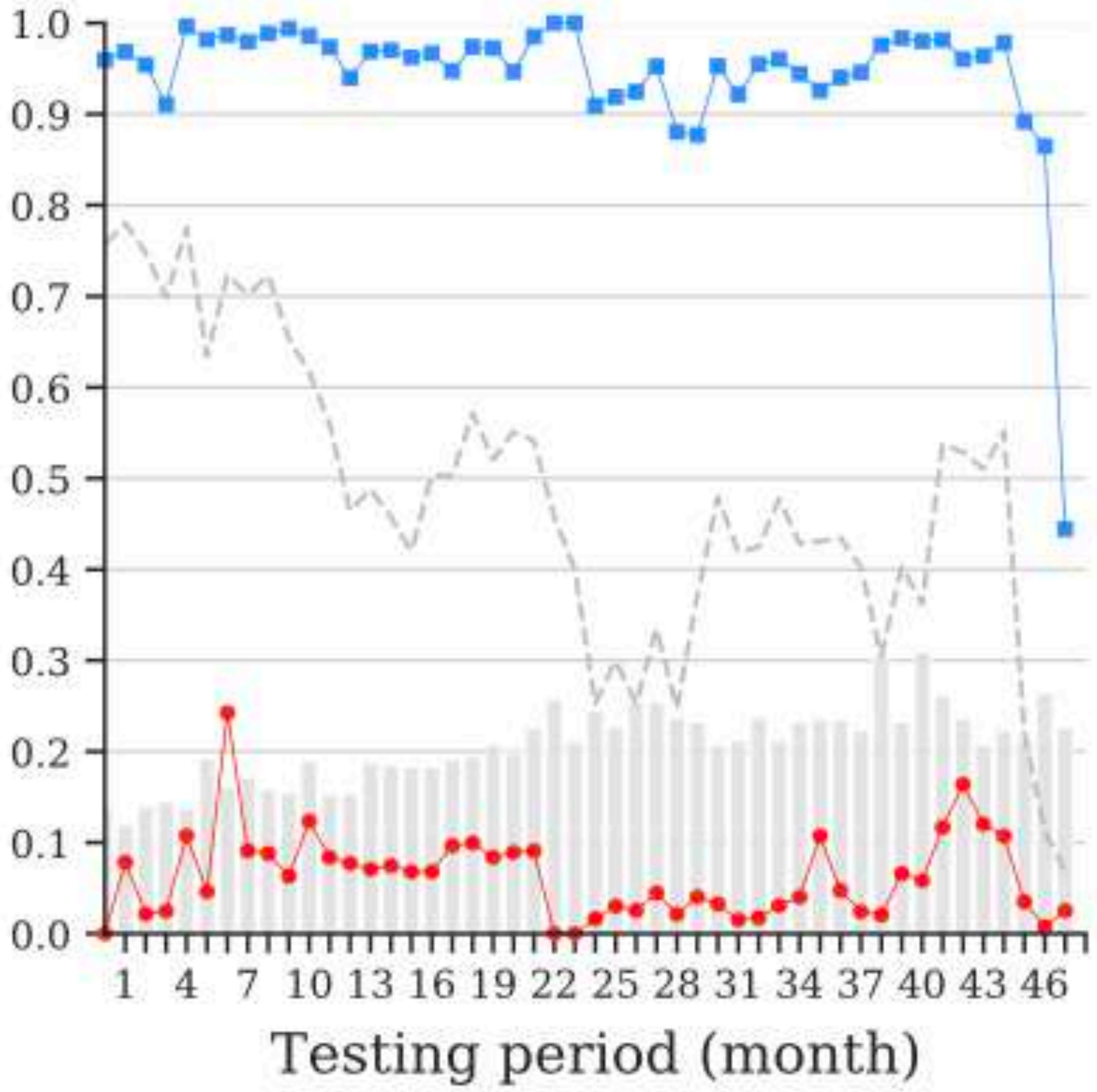

```
AUT(F1, kept):       0.82
AUT(F1, rejected):   0.00
CPU hours:           46.1
```

**ICE**
(0.33 calibration split)




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

**CCE**
(10 folds)

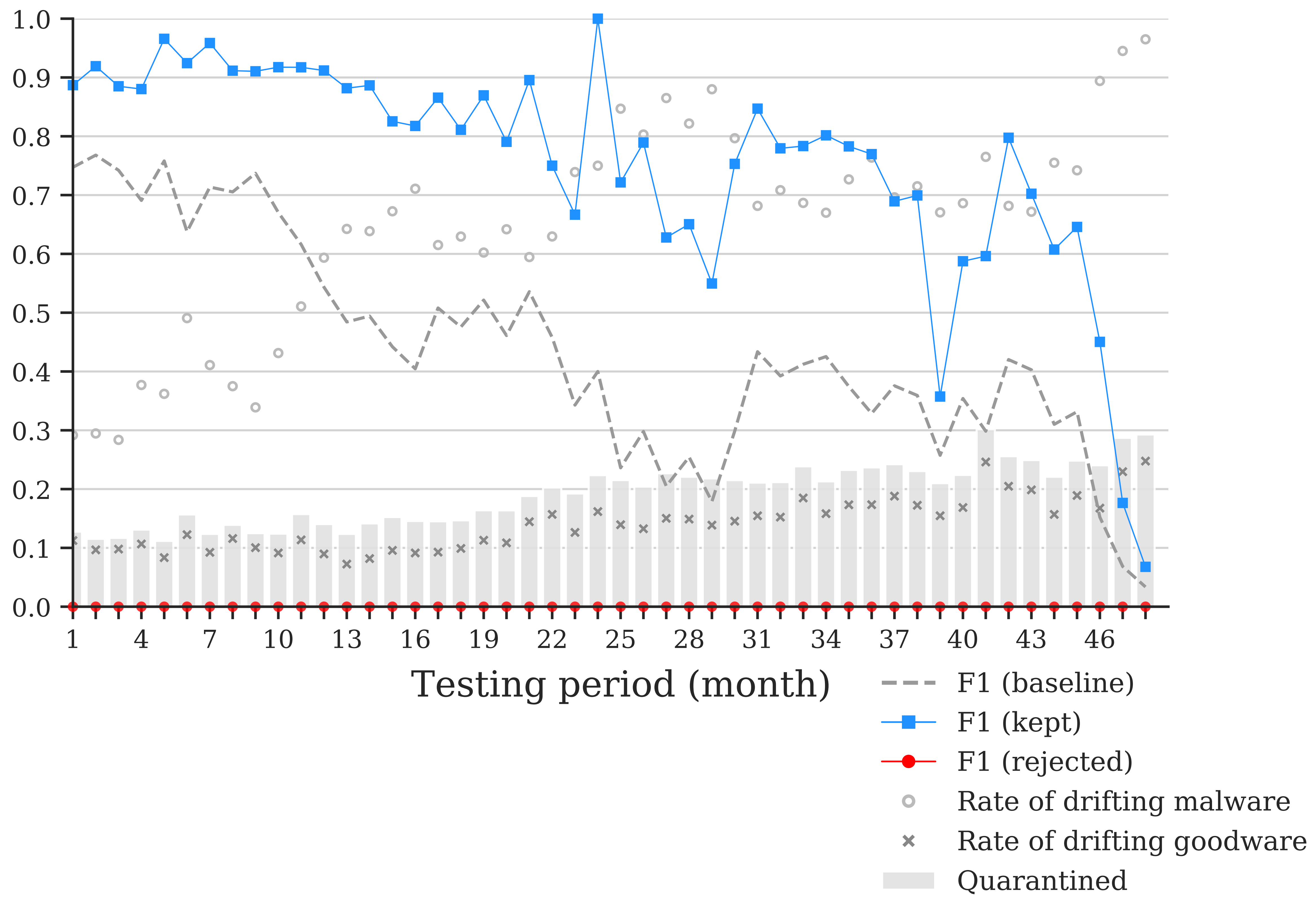



```
AUT(f1, kept):       0.94
AUT(f1, rejected):   0.06
CPU hours:           35.6
```

**[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 — Drift Rate

Android Malware (maximizing F1)

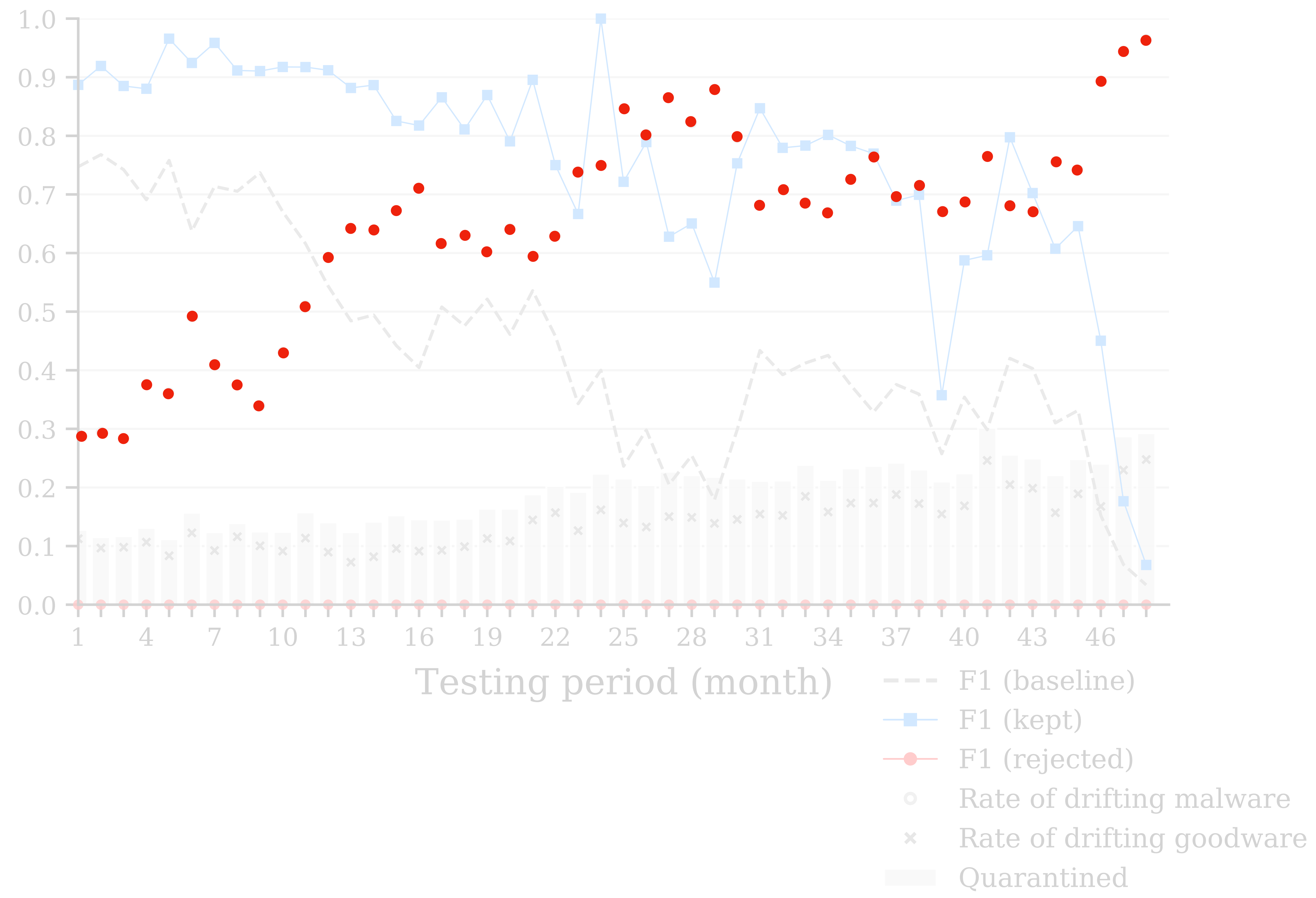



**[IEEE S&P 2022]** Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift

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

# Results: Rejection Performance — Drift Rate

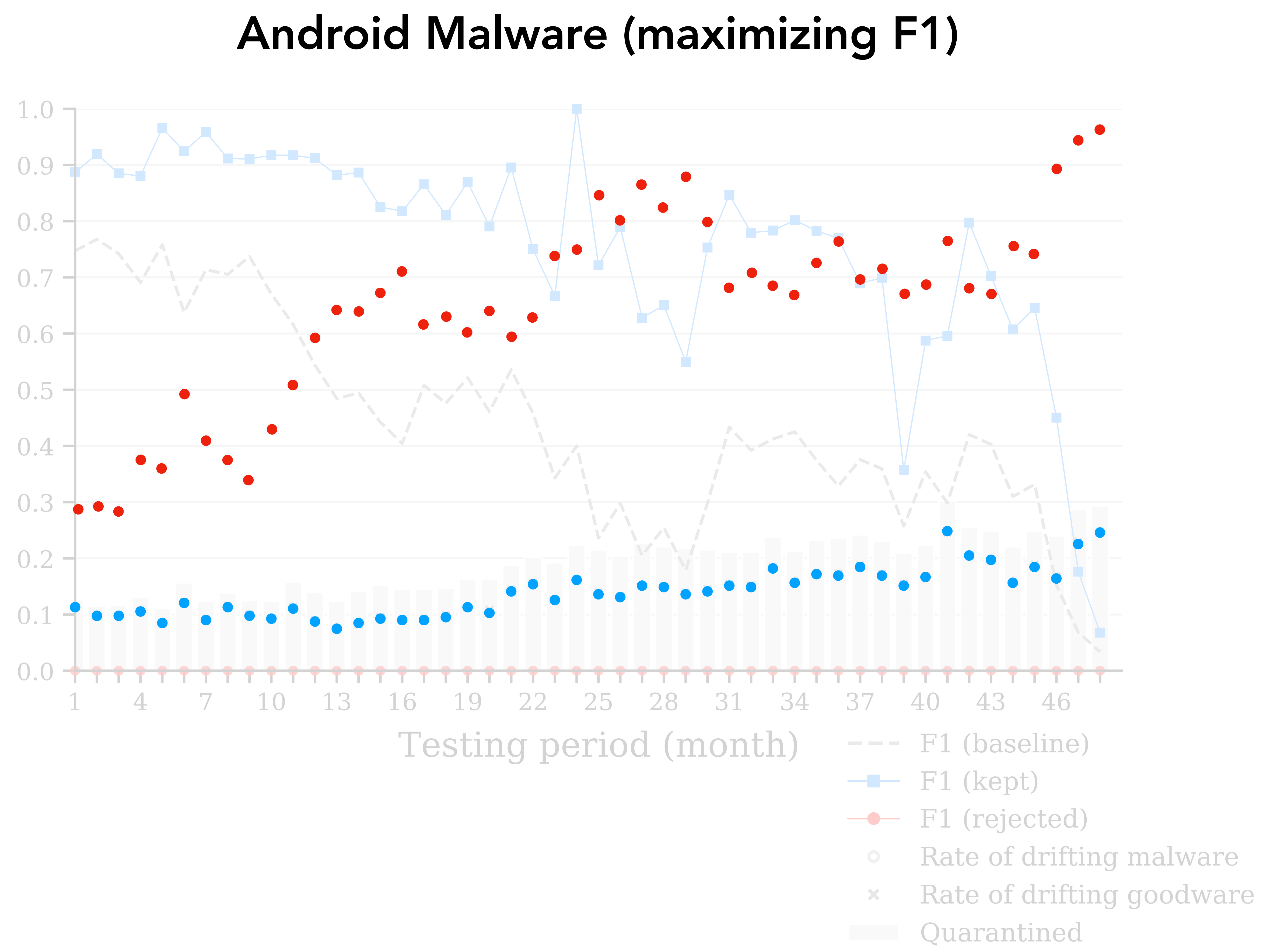
Android Malware (maximizing F1)




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

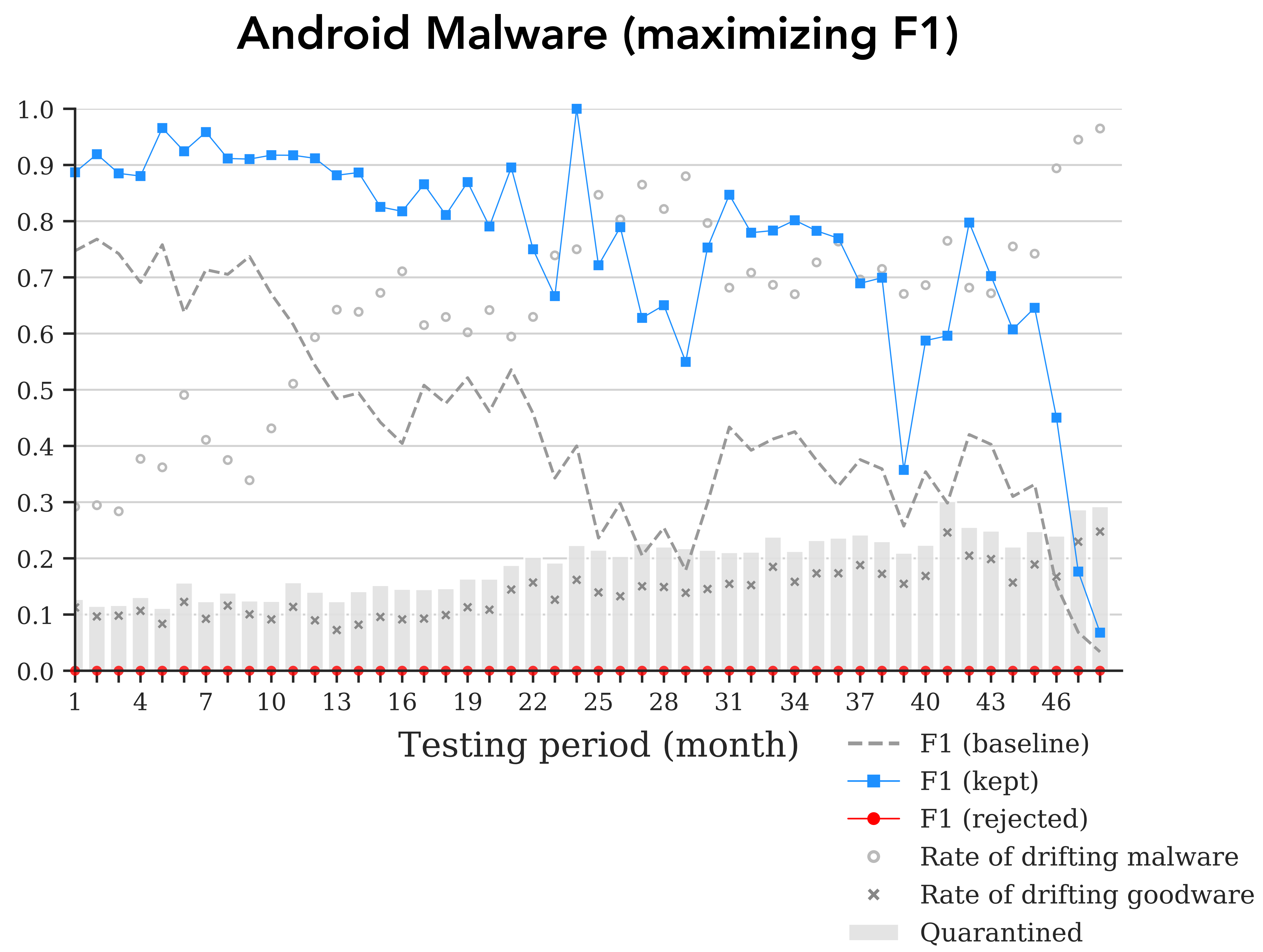
https://s2lab.cs.ucl.ac.uk/projects/transcend/

# 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/

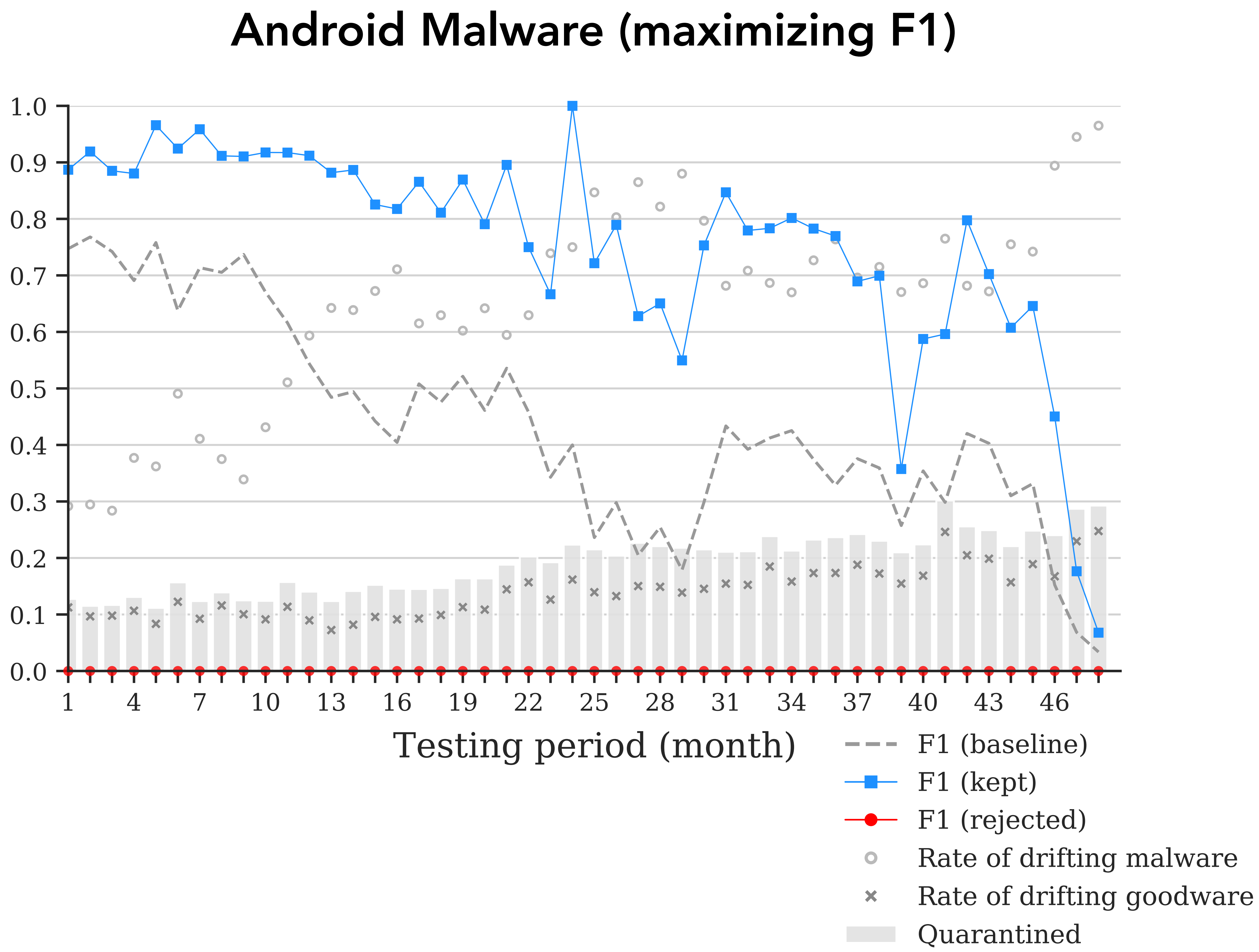
# Results: Rejection Performance — Drift Rate




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

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

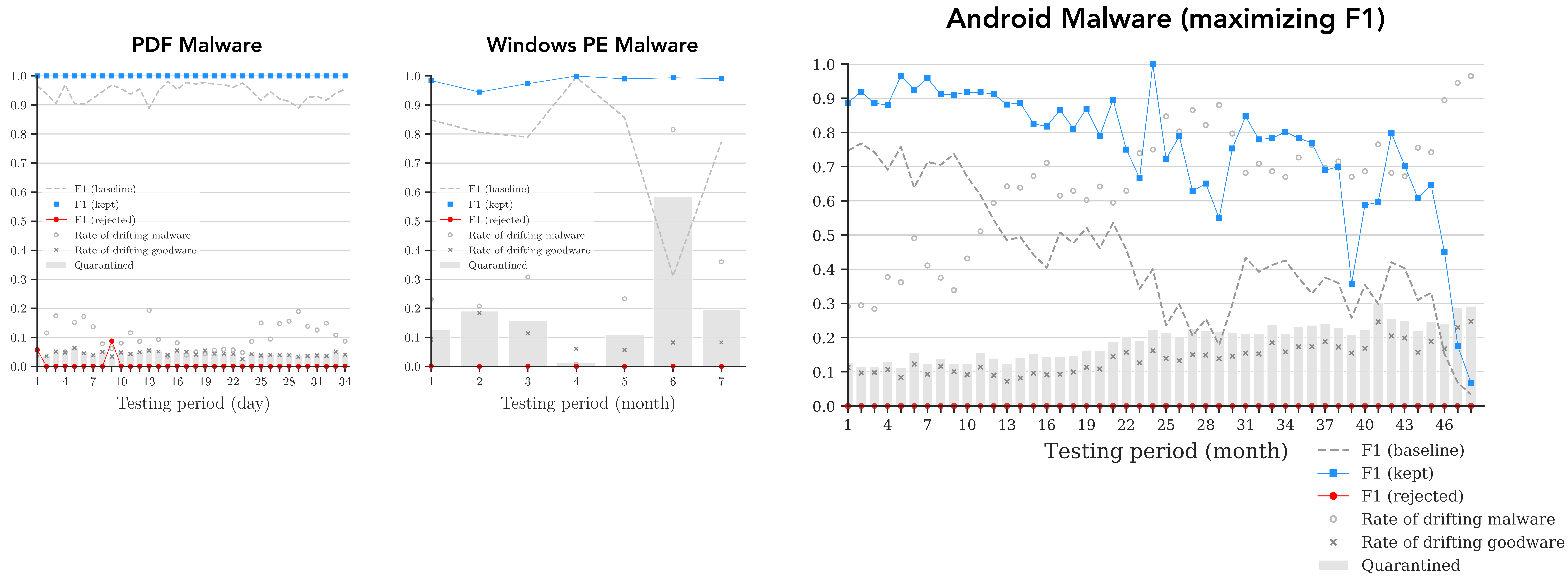
# Results: Rejection Performance — Drift Rate




**[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

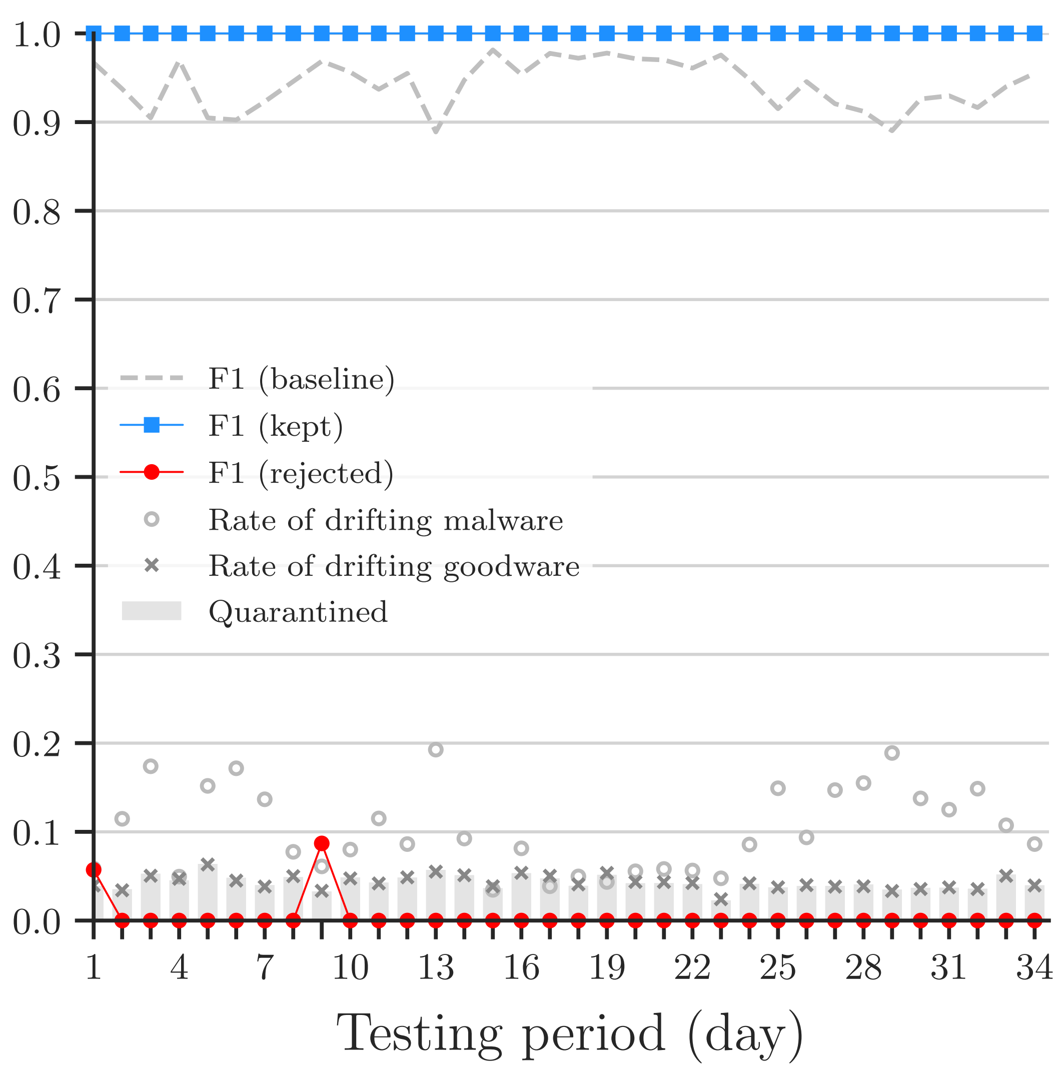



**[IEEE S&P 2022]** Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift

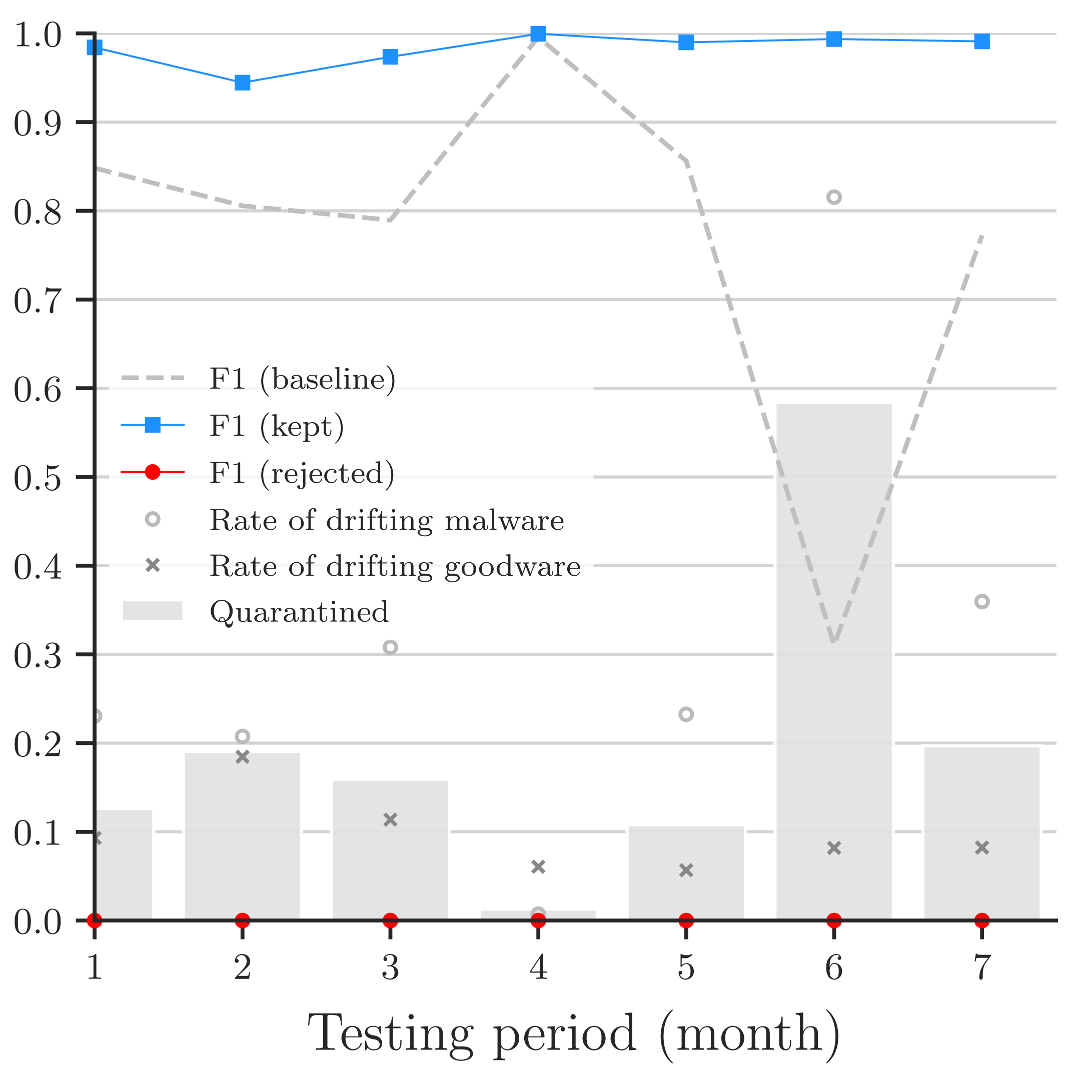
https://s2lab.cs.ucl.ac.uk/projects/transcend/

# Results: Rejection Performance — Drift Rate

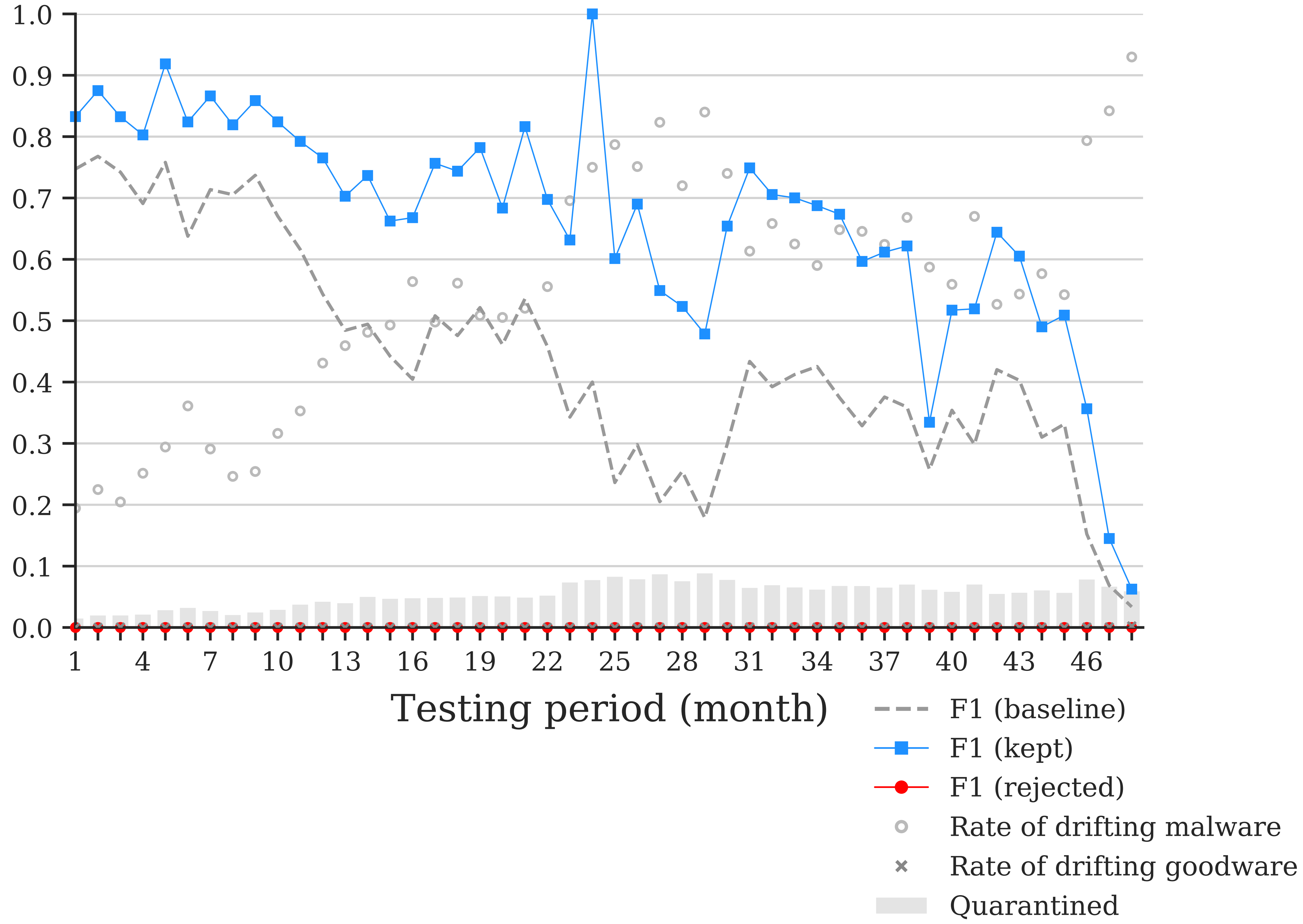
PDF Malware




Windows PE Malware



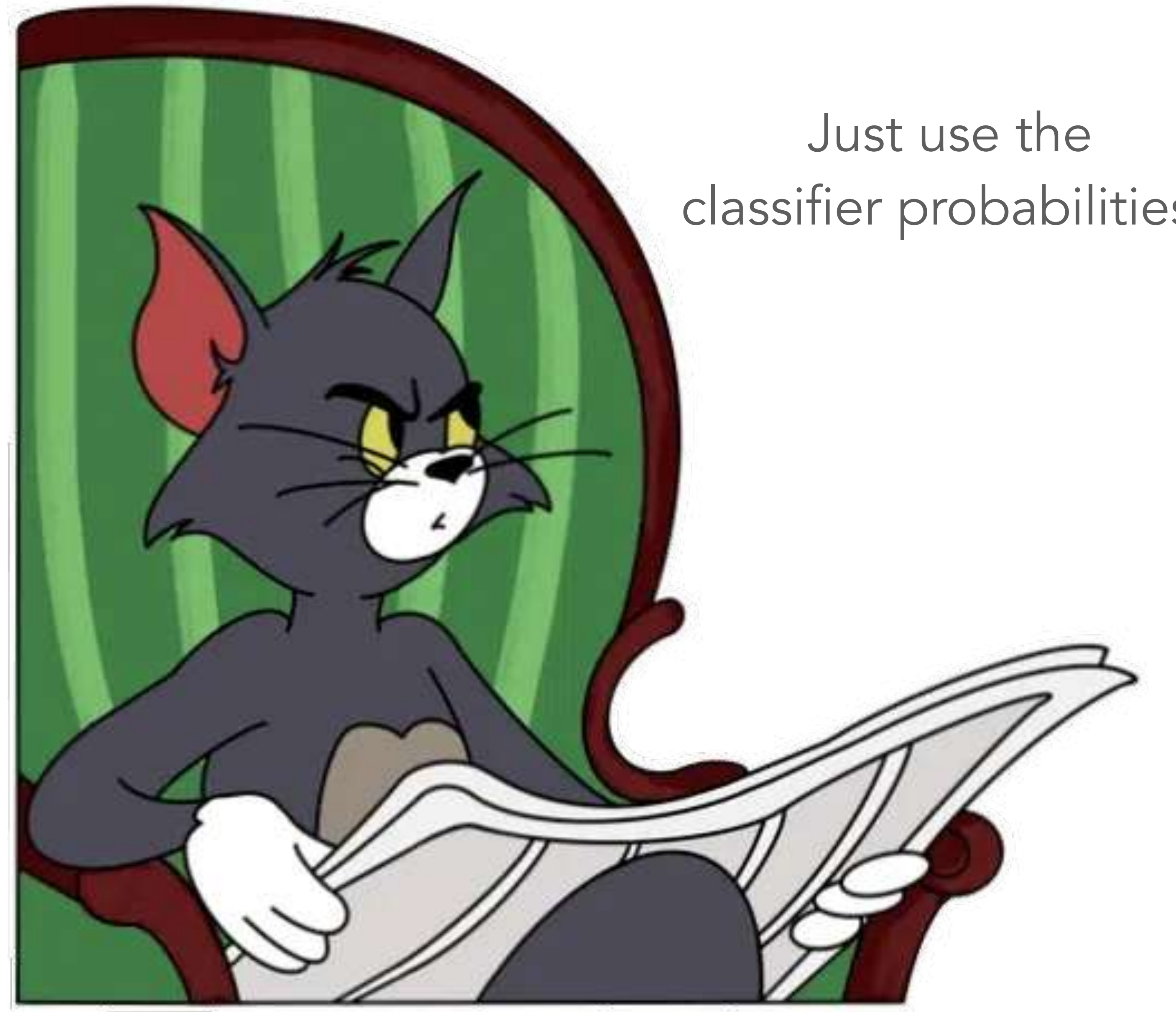

Android Malware (minimizing rejections)




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

# Results: Quality Measures

# Results: Quality Measures

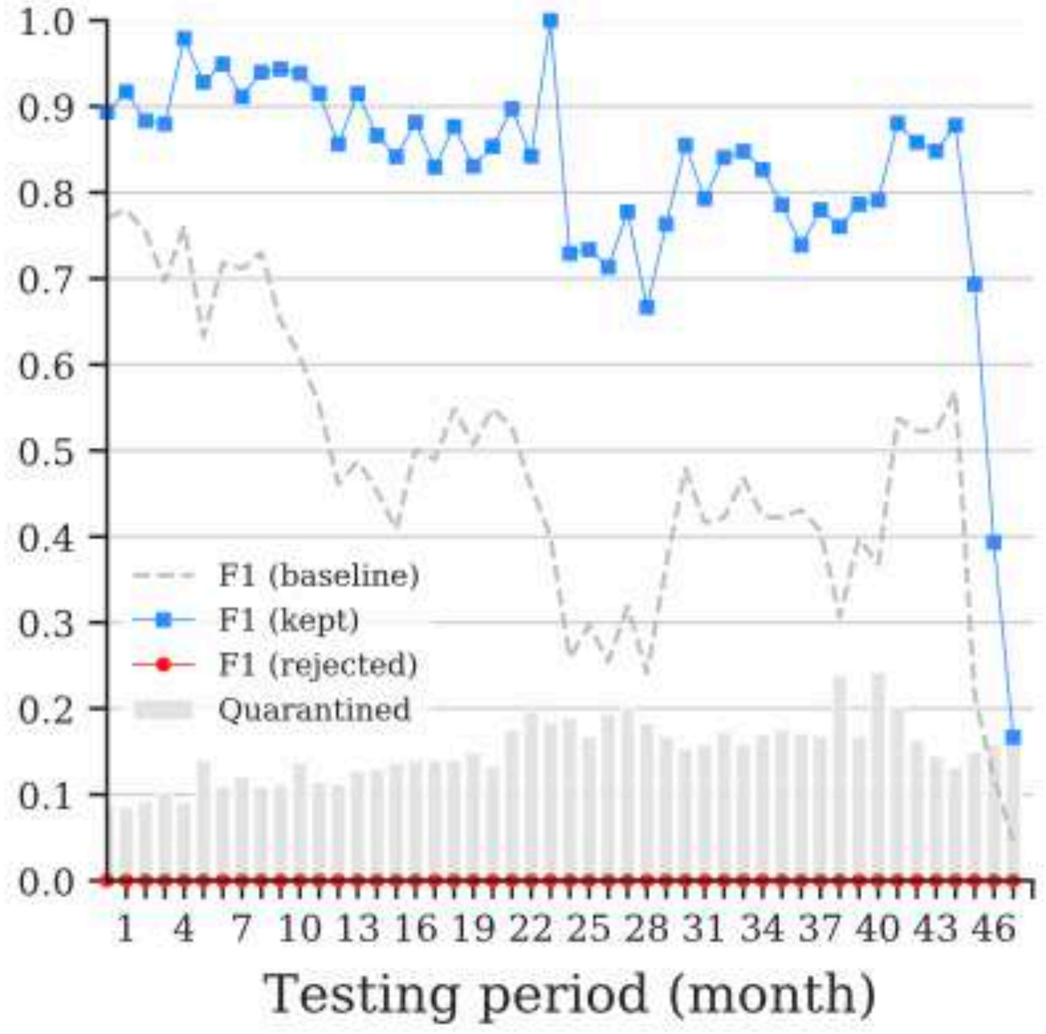


Just use the
classifier probabilities?

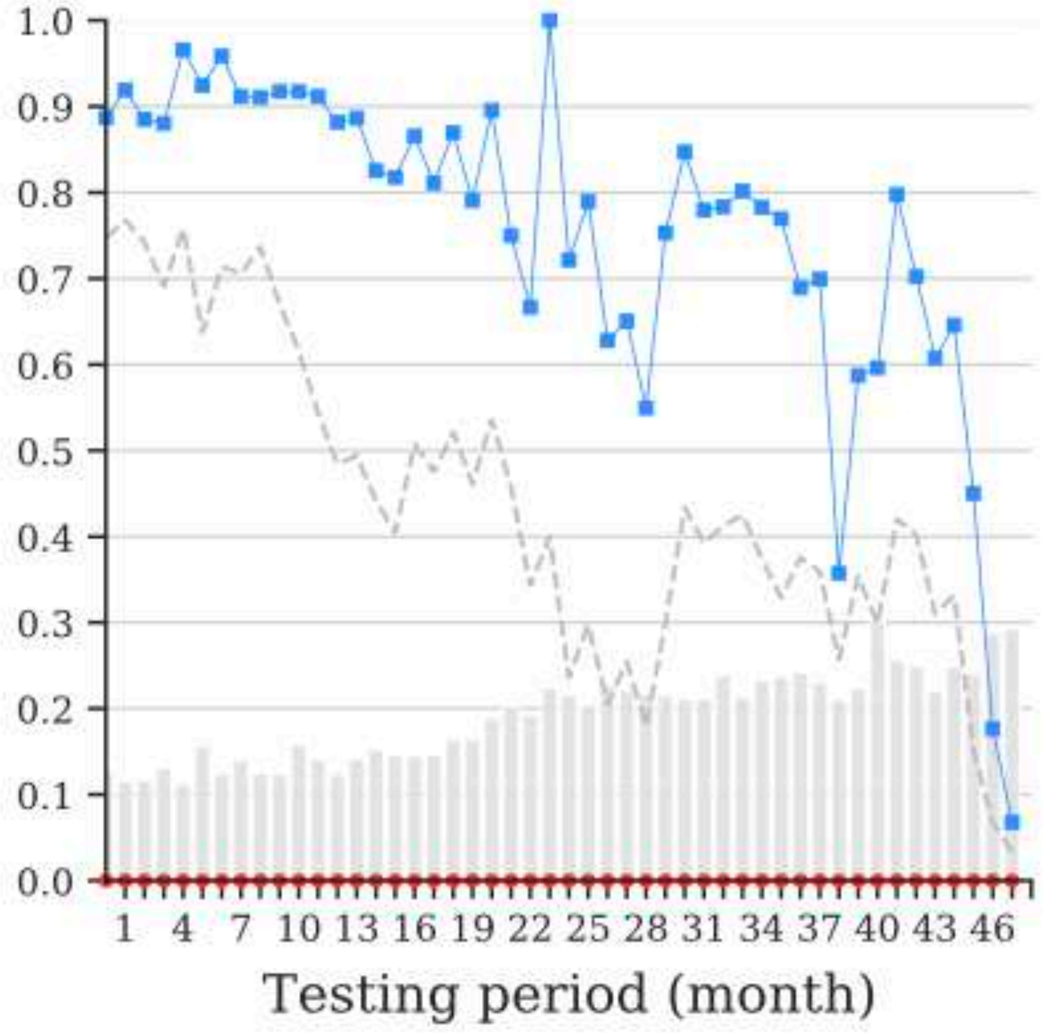
# Results: Quality Measures

**Conformal Evaluation**

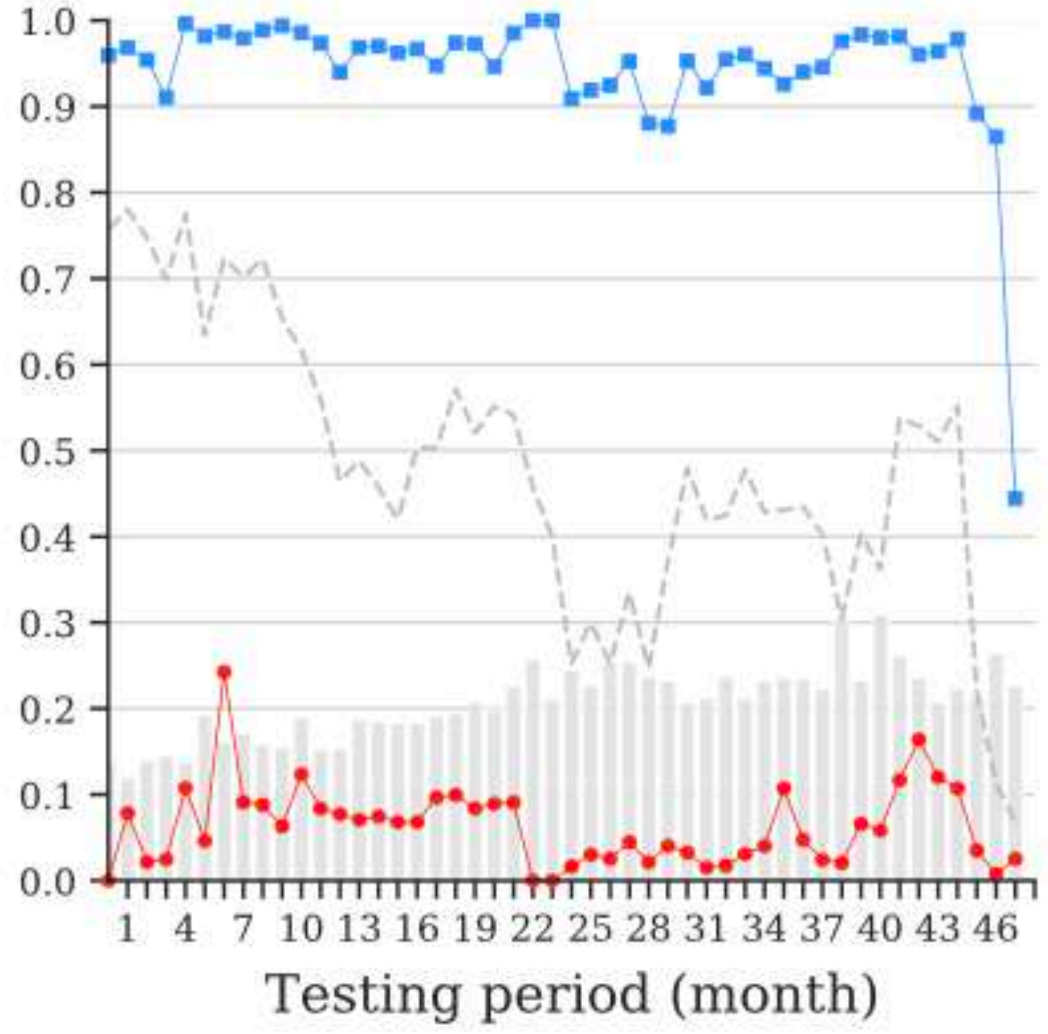
**Approx-TCE** (10 folds)




**ICE** (0.33 calibration split)




**CCE** (10 folds)

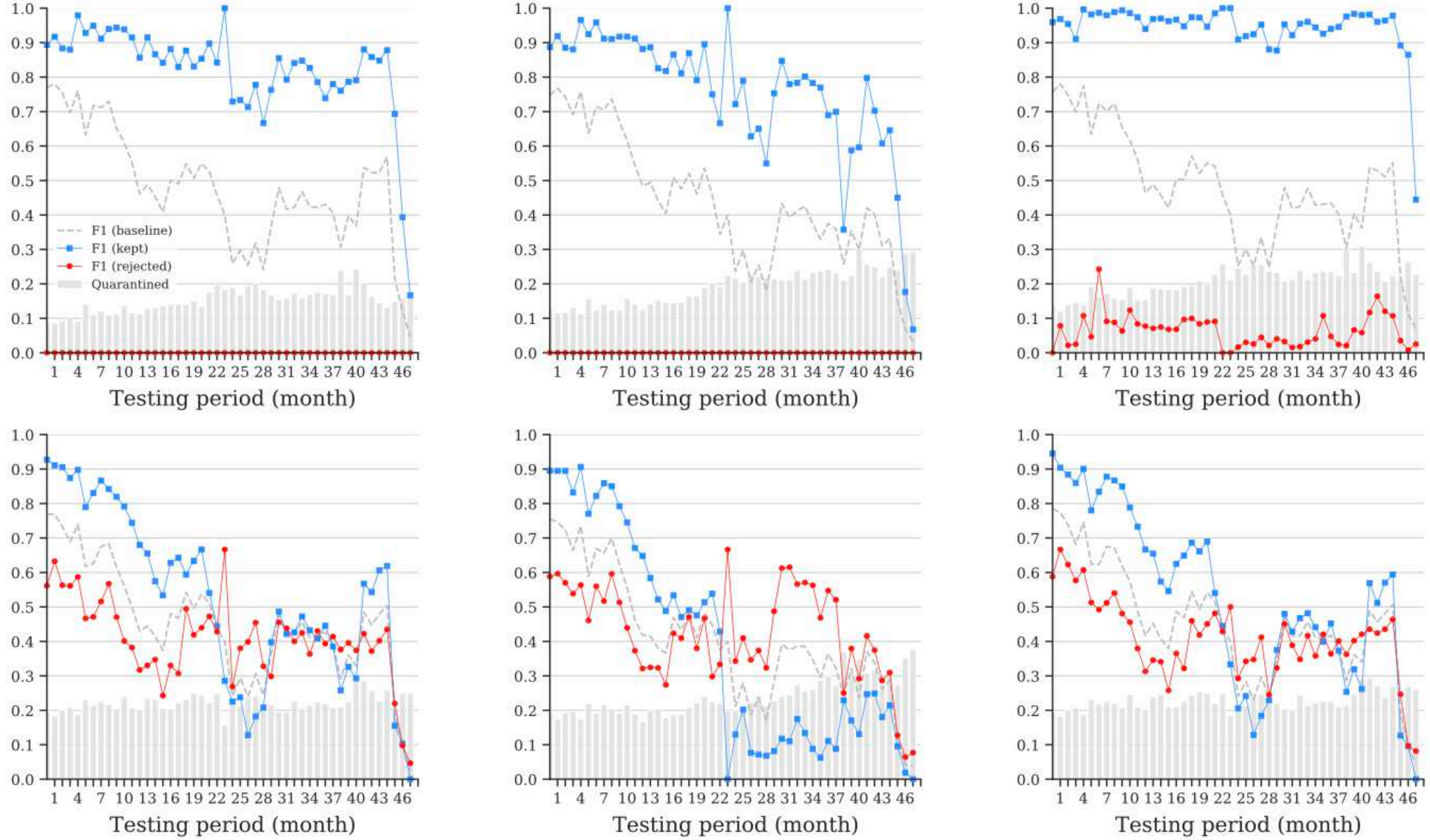


# Results: Quality Measures

| | Approx-TCE (10 folds) | ICE (0.33 calibration split) | CCE (10 folds) |
|---|---|---|---|

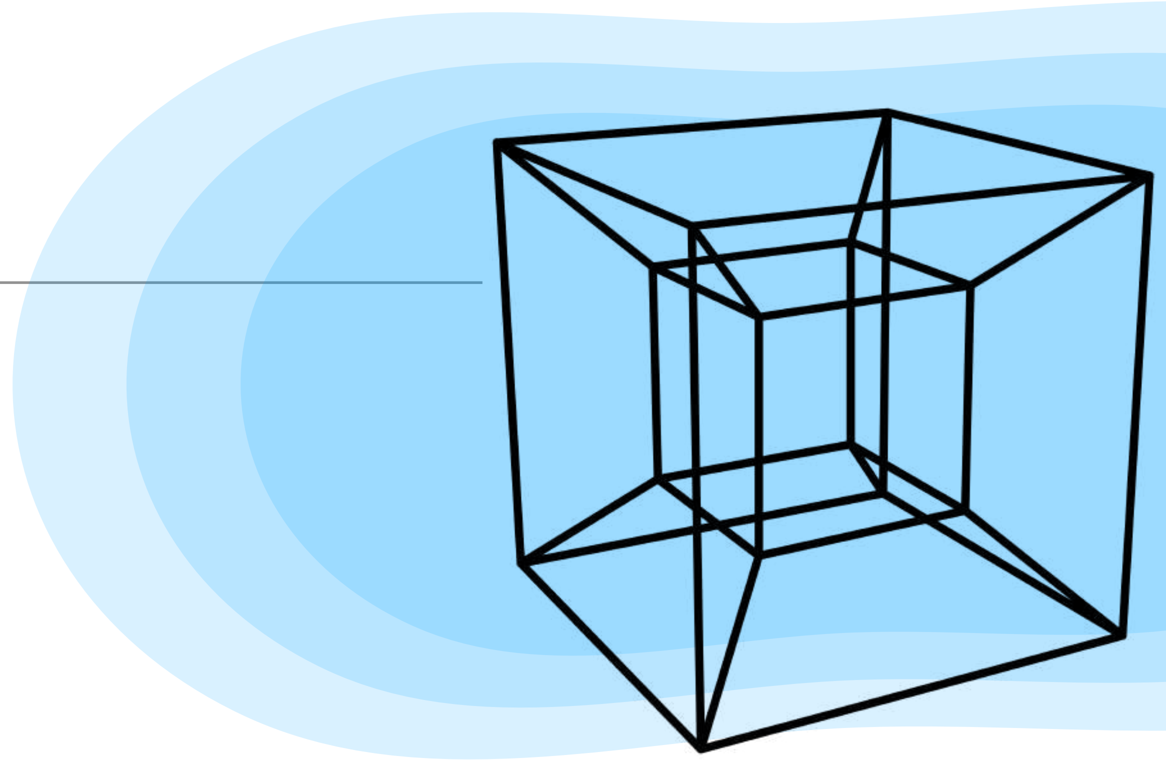
Conformal Evaluation

Probability



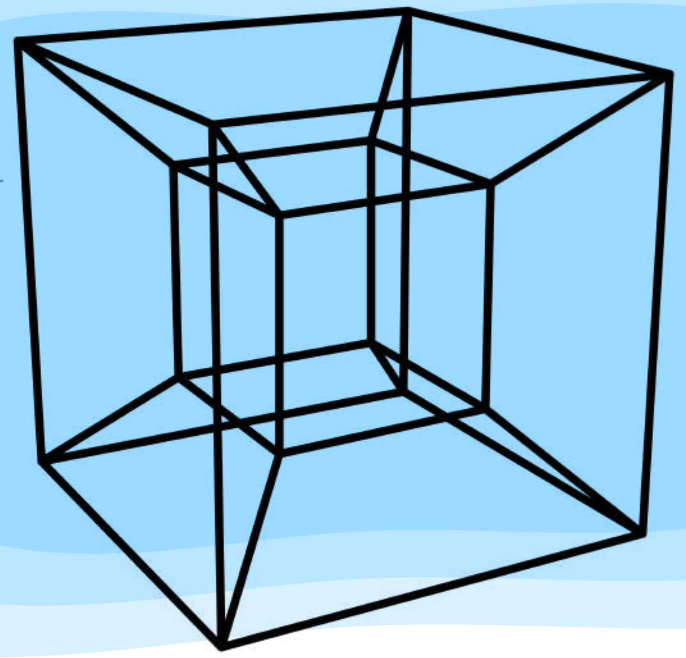
# Take Aways

# Take Aways



[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

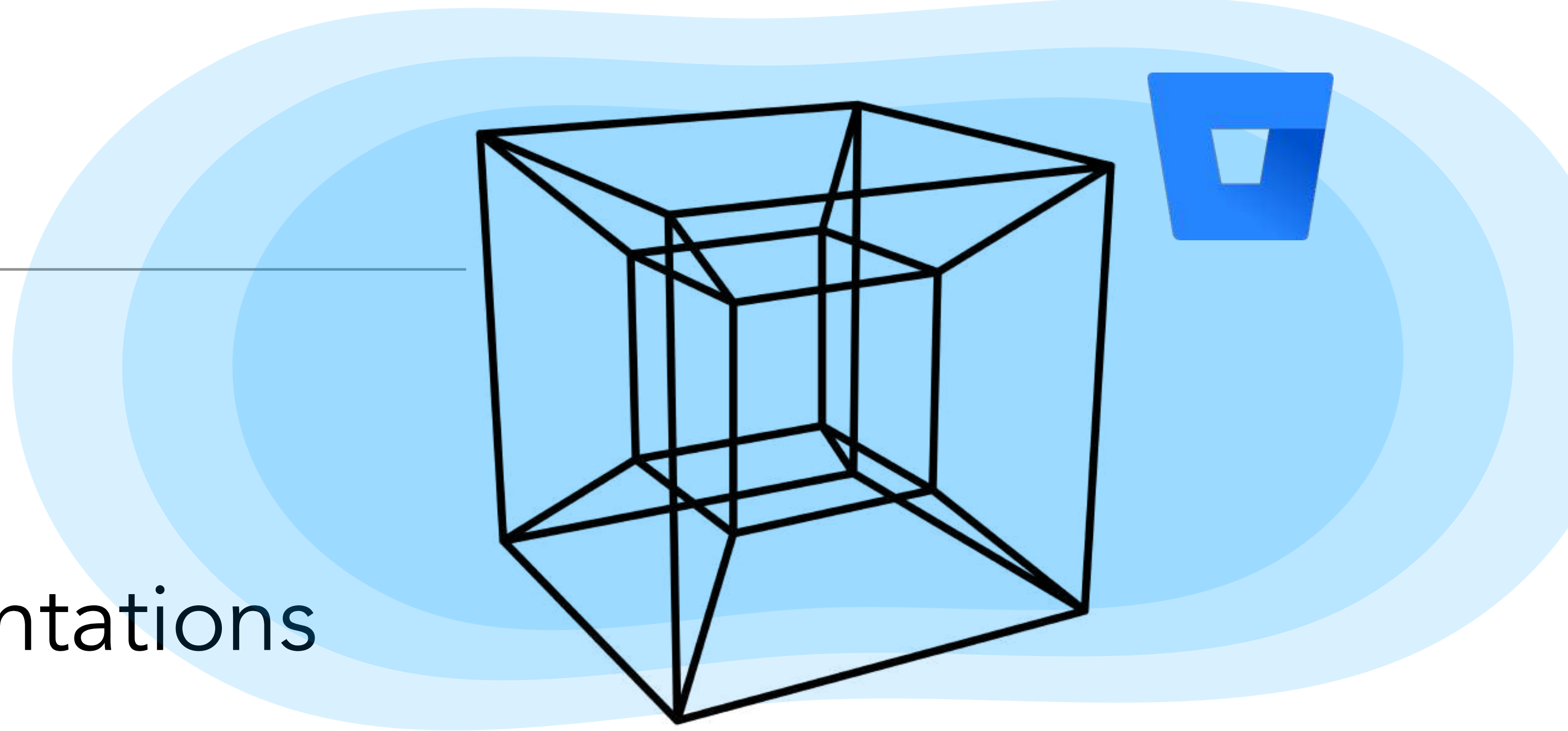
# Take Aways



- 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, and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning**, **online learning**


[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

# Take Aways

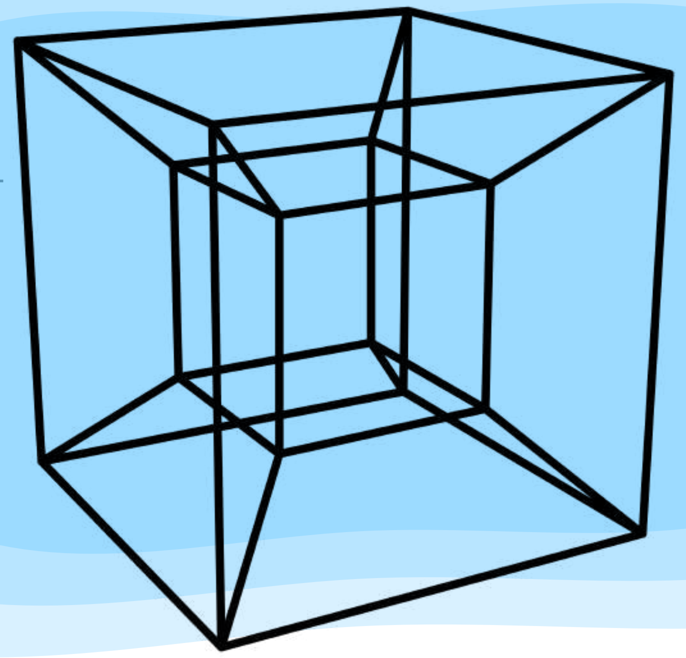


- Computer Security is highly **non-stationary** [1] and often **class-imbalanced**
  - Arms-race between attackers and defenders; role of abstractions/representations
  - Perform **time-aware** evaluations, and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active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

- Reason about problem space **(relizable) adversarial attacks and defenses** [2]


[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

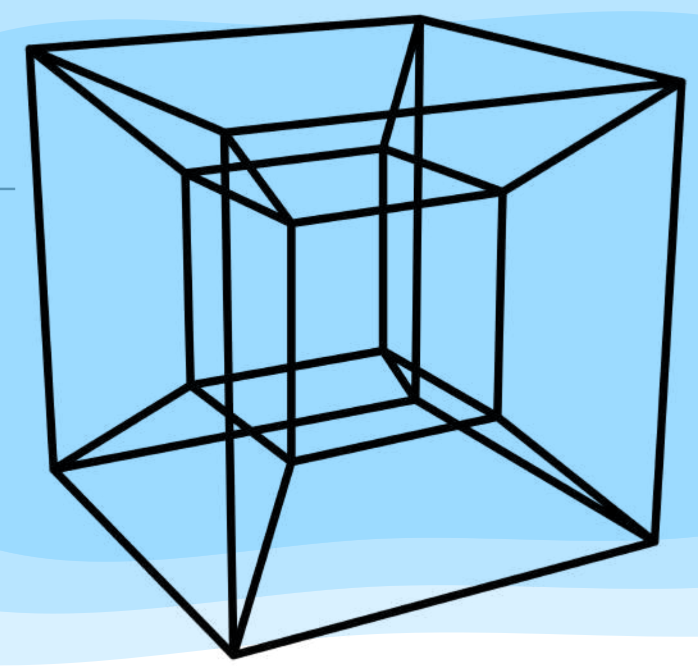
# Take Aways



- 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, and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning**, **online learning**
- Reason about problem space **(relizable) adversarial attacks and defenses** [2]
- 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

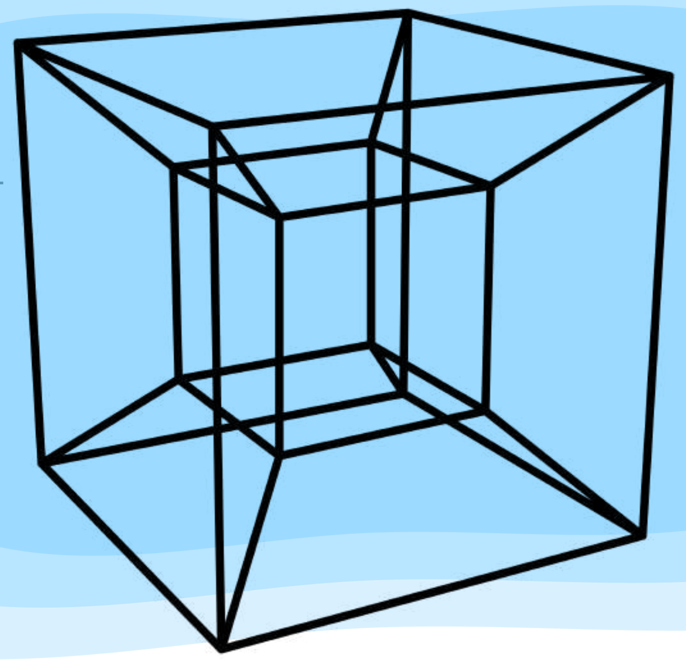
# Take Aways



- 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, and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning**, **online learning**
- Reason about problem space **(relizable) adversarial attacks and defenses** [2]
- 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

# 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, and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning**, **online learning**

- Reason about problem space **(relizable) adversarial attacks and defenses** [2]

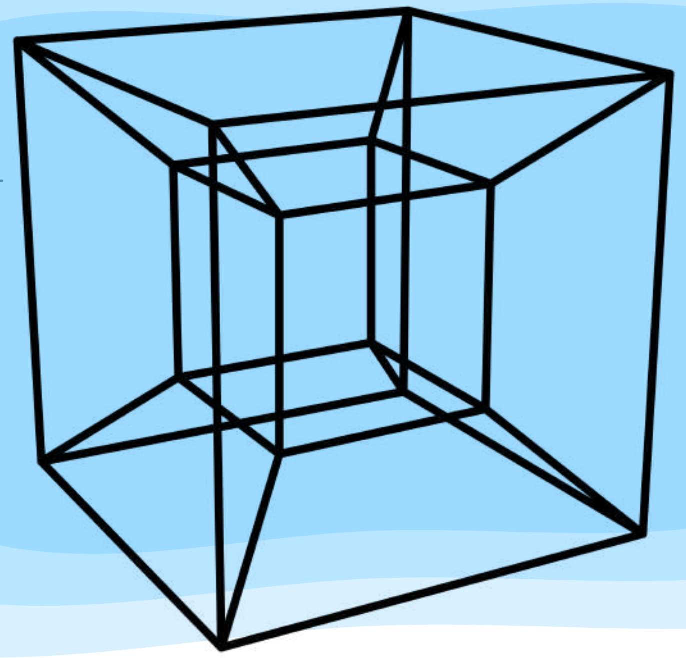
- Reason about the **relationship** between **adversarial ML** and **dataset shifts**

- 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

# 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, and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning**, **online learning**
- Reason about problem space **(relizable) adversarial attacks and defenses** [2]
- 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

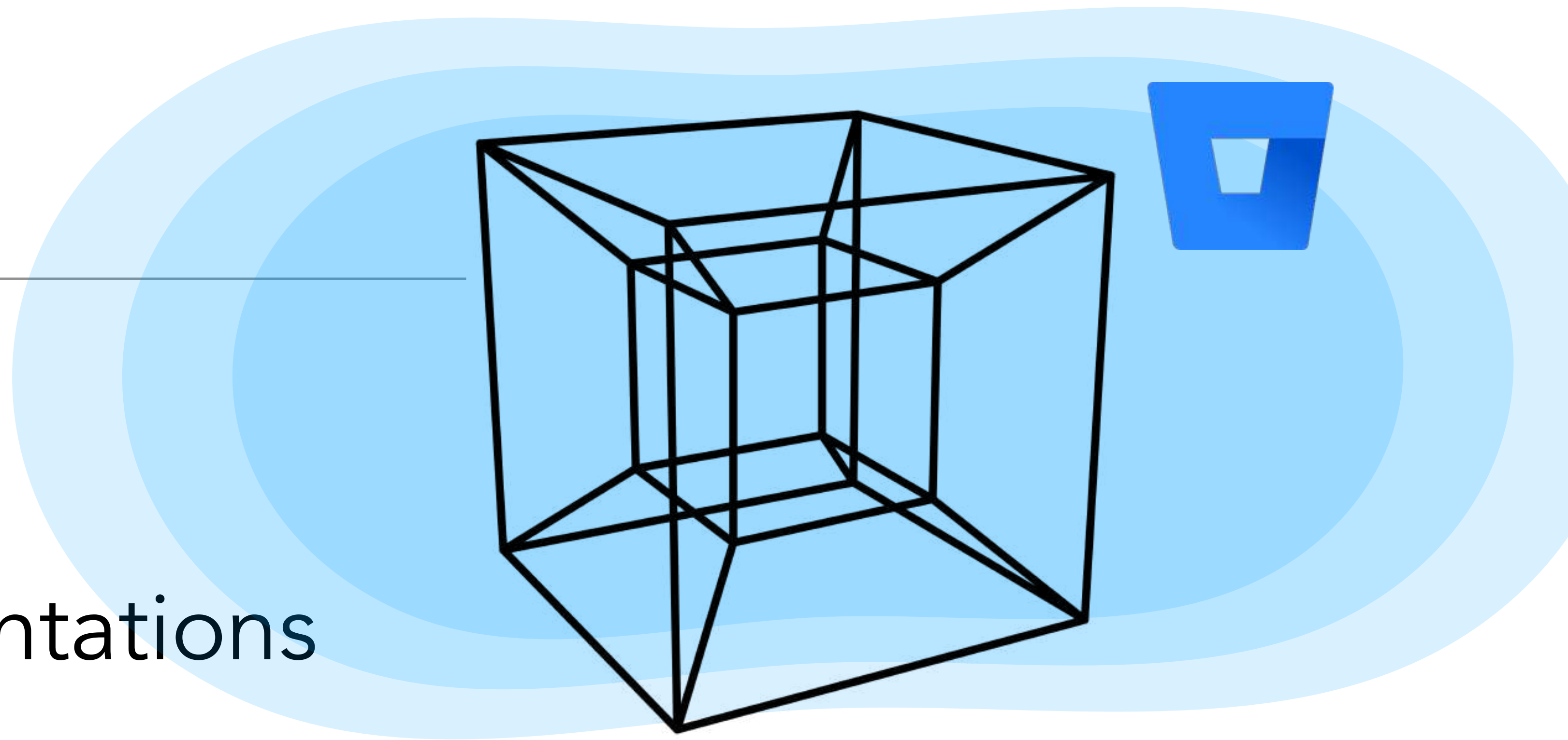
**https://twitter.com/joshua_saxe/status/1550545466072264704**


[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 Security 2022

# 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, and avoid pitfalls [5]
  - Assume things go wrong: explore **rejection options** [3,4], **active learning**, **online learning**
- Reason about problem space **(relizable) adversarial attacks and defenses** [2]
- 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

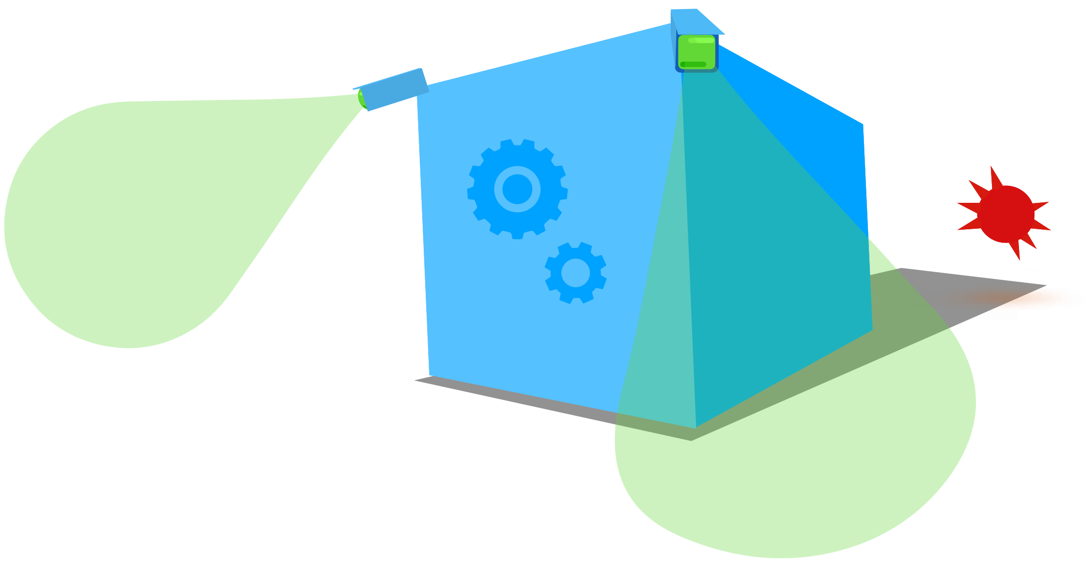
**https://twitter.com/joshua_saxe/status/1550545466072264704**


[1] Pendlebury et al., **TESSERACT: Eliminating experimental bias in malv**
[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 Security 2022

# Outline

Focus

## Adversarial ML evasion attacks against malware classifiers

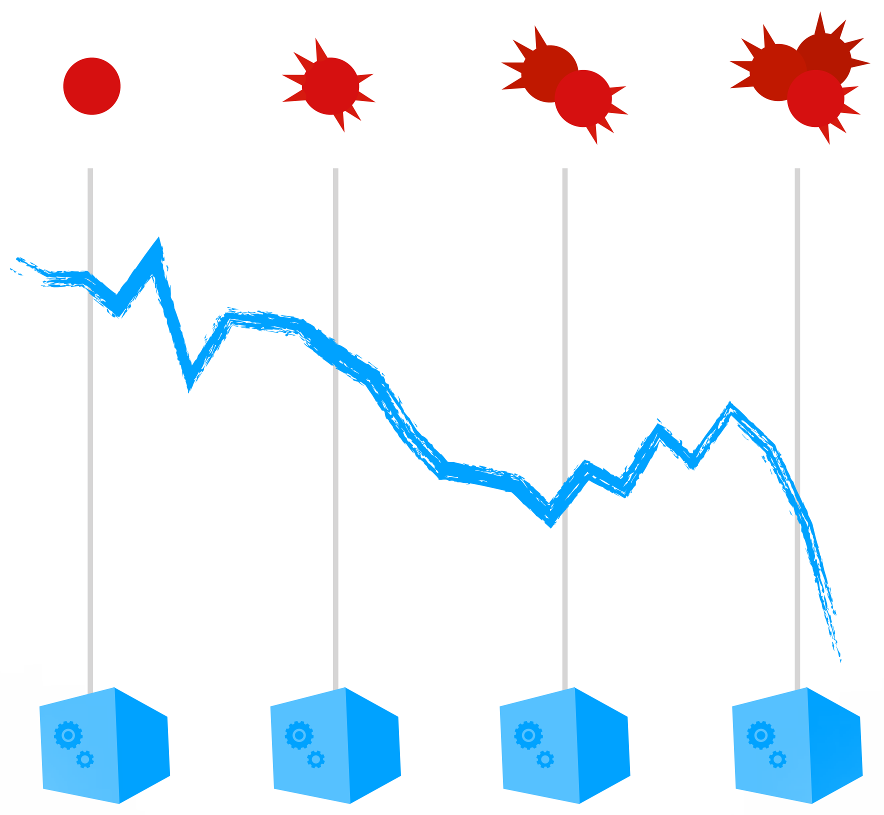


- Classic formulation of evasion attacks is ill-suited for reasoning about realizable evasive malware
- By reformulating, we can propose stronger attacks and easily compare against alternatives
- Practical end-to-end automatic adversarial malware as a service — how about defenses?

**[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space**

Bigger Picture

## Drifting scenarios caused by threats evolving over time



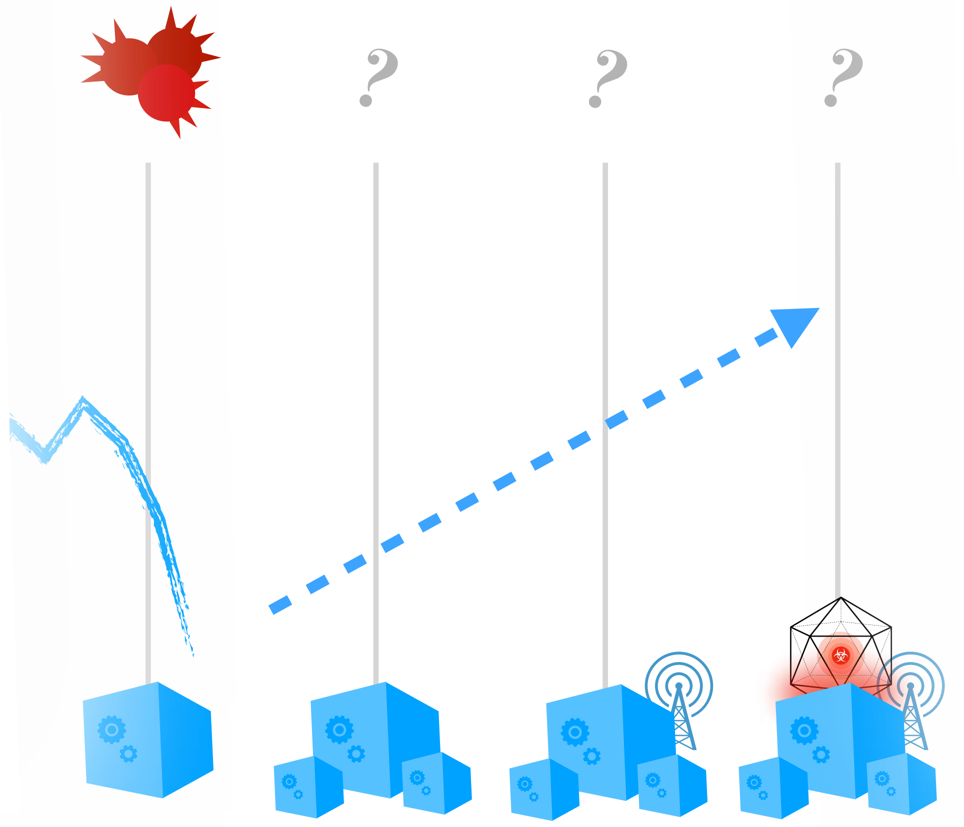
- How dataset shift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics
- Detecting shifts with abstaining classifiers and classification with rejection

**[USENIX Sec 2017 & IEEE S&P 2022] Transcend: Detecting Con**
**Transcending Transcend: Revisiting Malware Classification in t**

**[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias**

Looking Ahead

## Quo vadis?




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

**[USENIX Sec 2022] Dos and Don’ts of Machine Learning in Computer Security**

# 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
- Practical end-to-end automatic adversarial malware as a service — how about defenses?

**[IEEE S&P 2020] Intriguing Properties of Adversarial ML Attacks in the Problem Space**



Bigger Picture

## Drifting scenarios caused by threats evolving over time

- How dataset shift affects machine learning-based detectors in security settings
- The need for time-aware evaluations and metrics
- Detecting shifts with abstaining classifiers and classification with rejection

**[USENIX Sec 2017 & IEEE S&P 2022] Transcend: Detecting Con**
**Transcending Transcend: Revisiting Malware Classification in t**

**[USENIX Sec 2019] TESSERACT: Eliminating Experimental Bias**

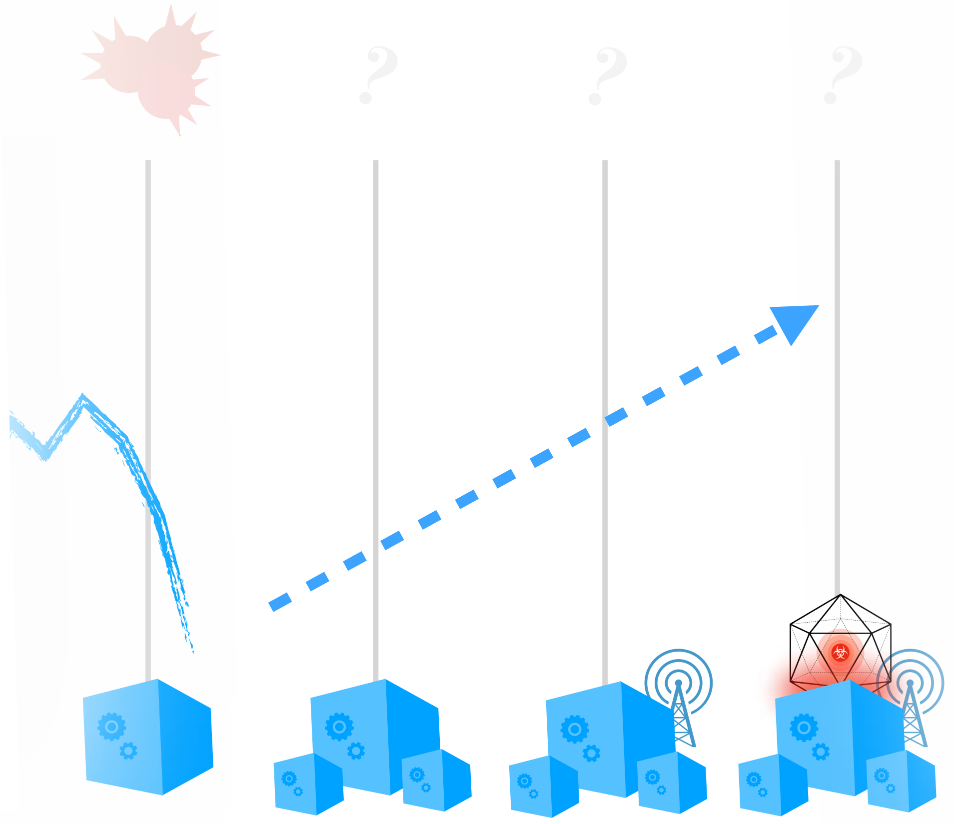


Looking Ahead

## Quo vadis?

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

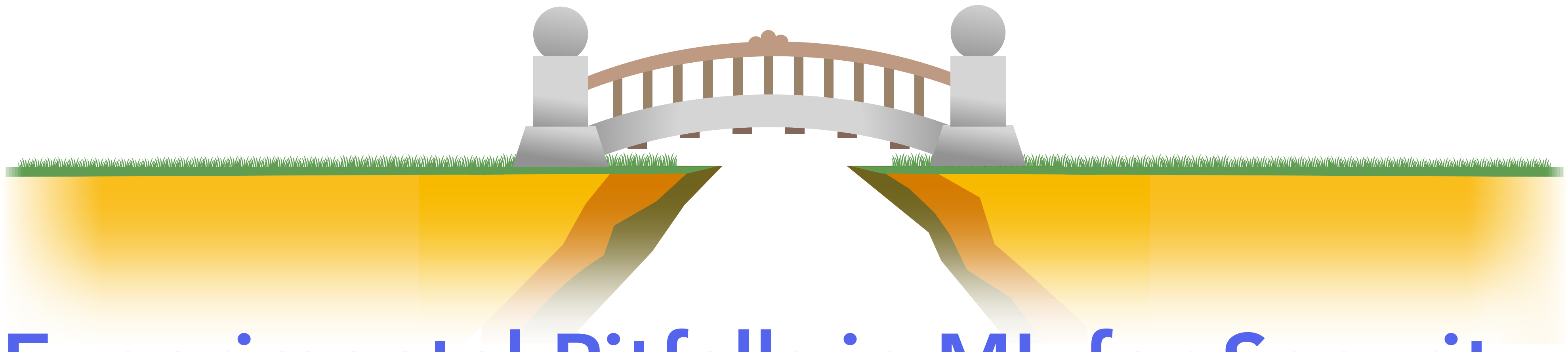
**[USENIX Sec 2022] Dos and Don’ts of Machine Learning in Computer Security**



# What Trustworthy ML for Systems Security May Look Like

Trustworthy ML for Systems Security

# What Trustworthy ML for Systems Security May Look Like



## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

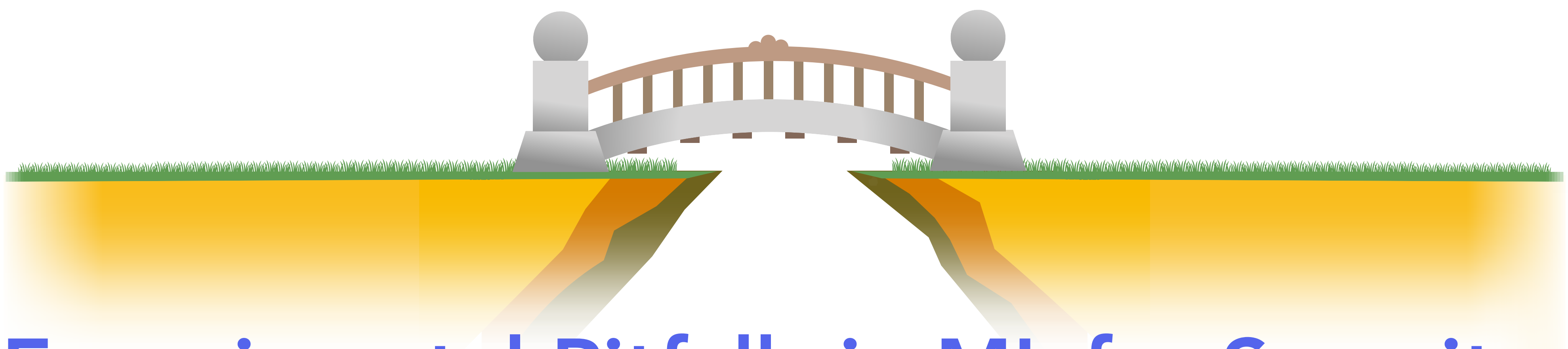
(collab. with TU Braunschweig)

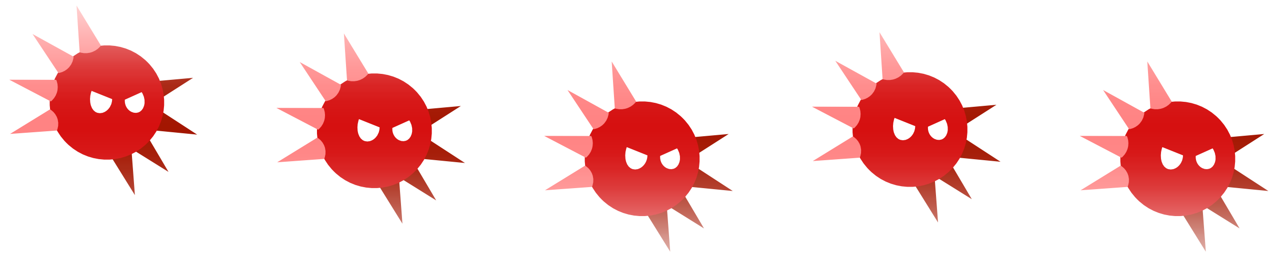
*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***

**Trustworthy ML for Systems Security**

# What Trustworthy ML for Systems Security May Look Like





**Experimental Pitfalls in ML for Security**

**To provide a solid foundation for empirical research**

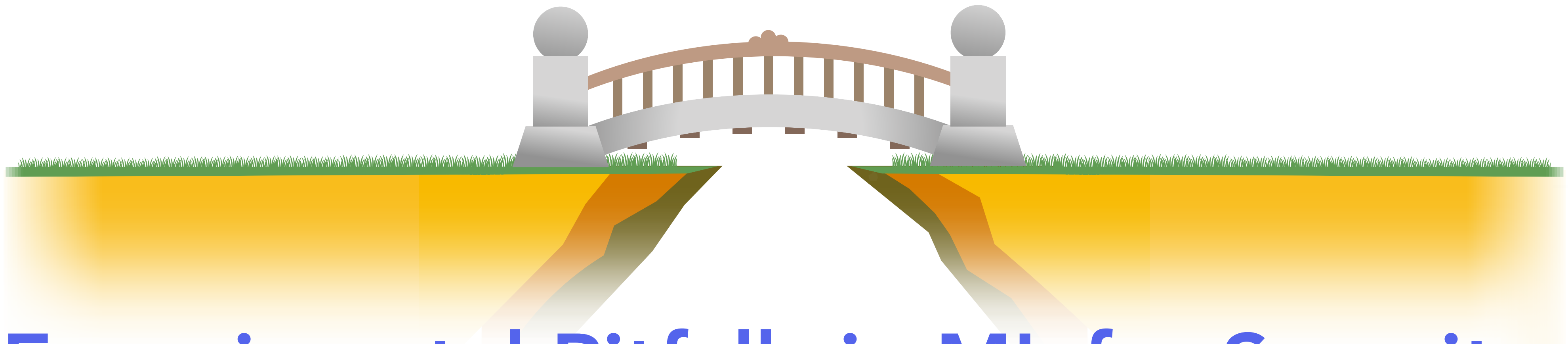
(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***

**Trustworthy ML for Systems Security**

# What Trustworthy ML for Systems Security May Look Like



## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***

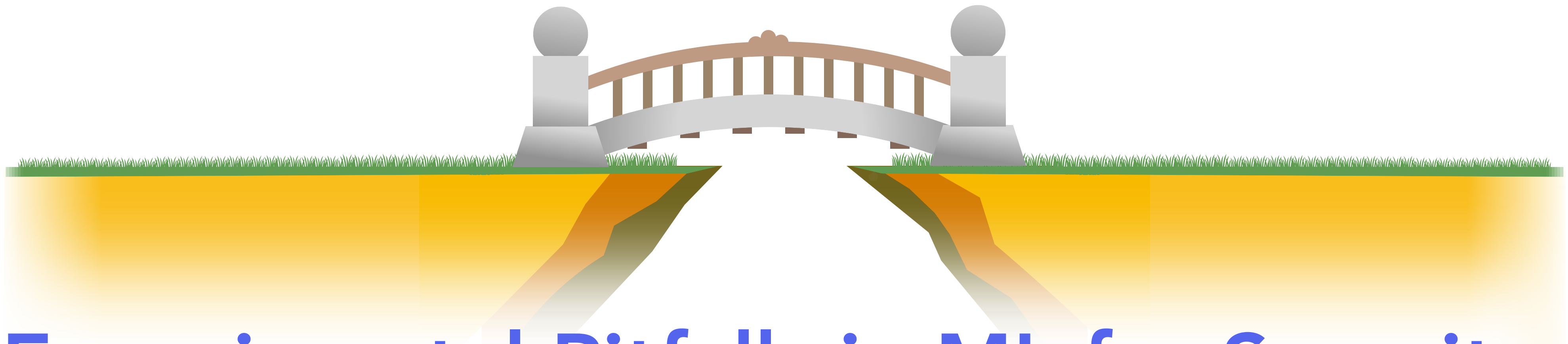
## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)

**Trustworthy ML for Systems Security**

# What Trustworthy ML for Systems Security May Look Like



## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***


## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

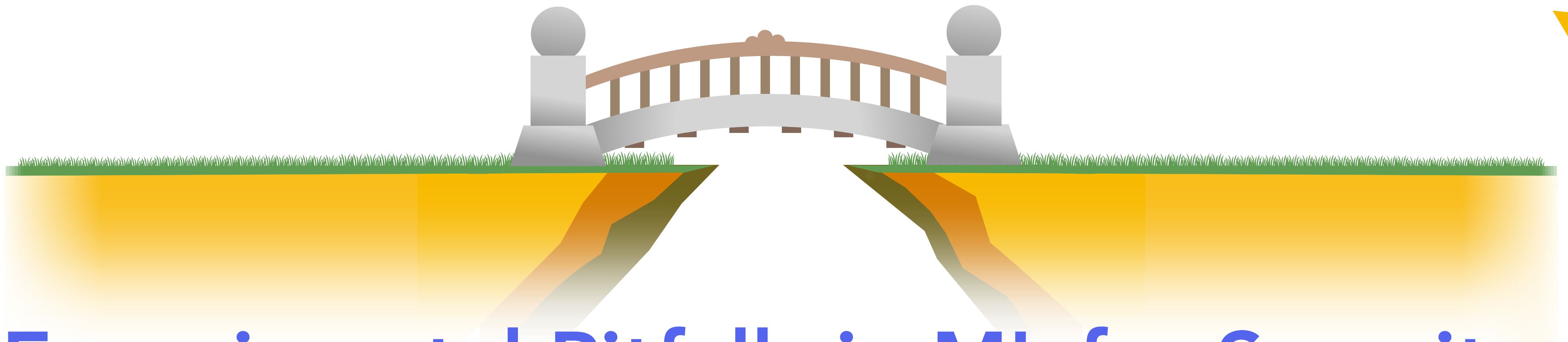
(collab. with Imperial College London and UniBw)

# What Trustworthy ML for Systems Security May Look Like

## Problem-Space Backdoors

**To explore realistic poisoning backdoors**

(collab. with University of Illinois at Urbana-Champaign)



## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***


## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

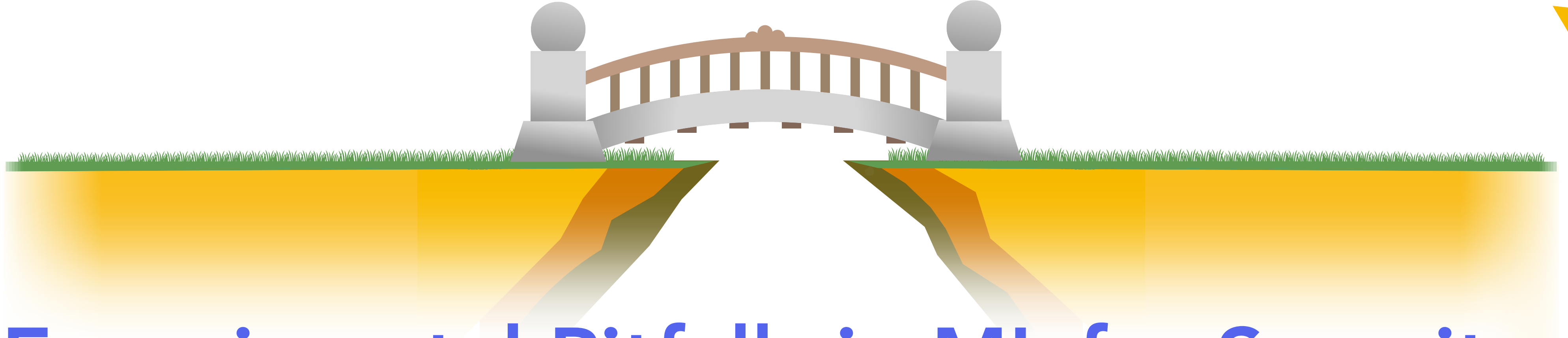
(collab. with Imperial College London and UniBw)

# What Trustworthy ML for Systems Security May Look Like

## Problem-Space Backdoors

**To explore realistic poisoning backdoors**

(collab. with University of Illinois at Urbana-Champaign)



## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***

## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

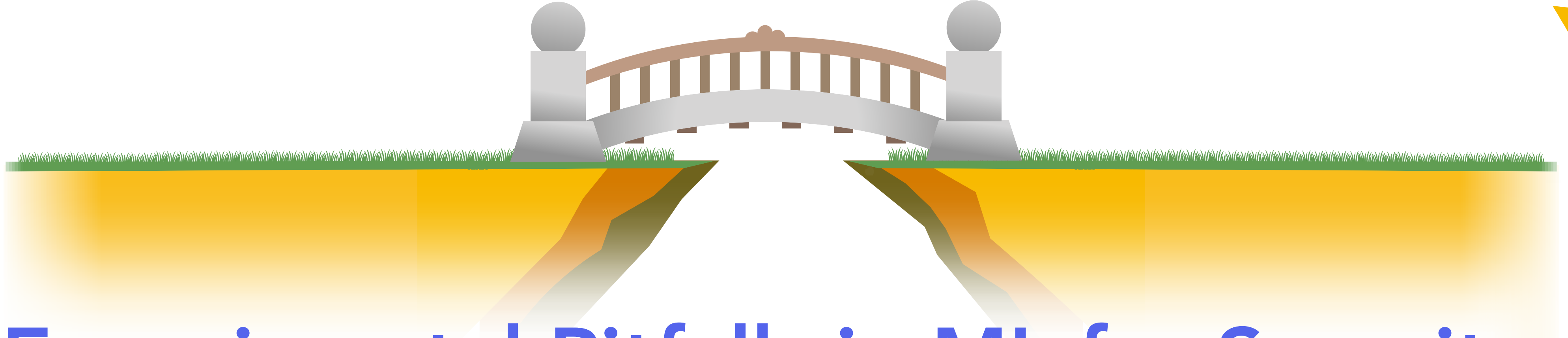
(collab. with Imperial College London and UniBw)

# What Trustworthy ML for Systems Security May Look Like

## Problem-Space Backdoors

**To explore realistic poisoning backdoors**

(collab. with University of Illinois at Urbana-Champaign)



## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***

## Robust Features

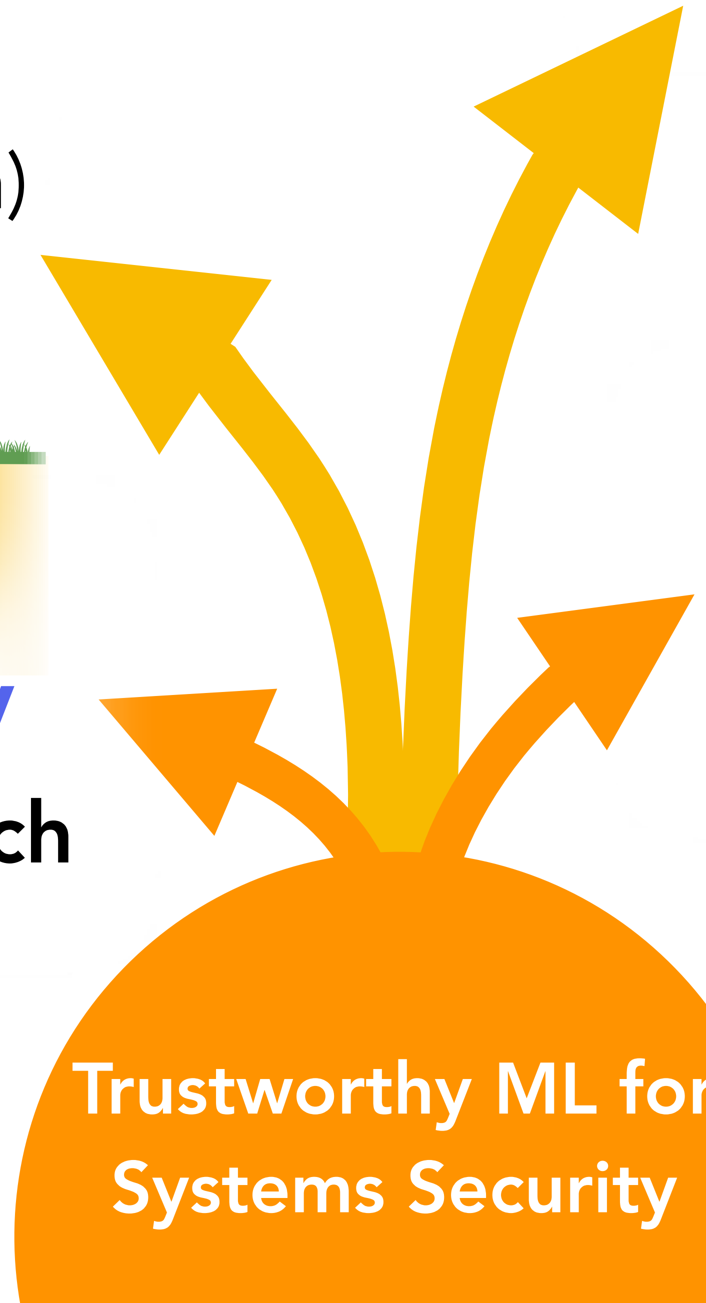
**To limit effectiveness of adversarial manipulation**

(collab. with Avast)

## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)

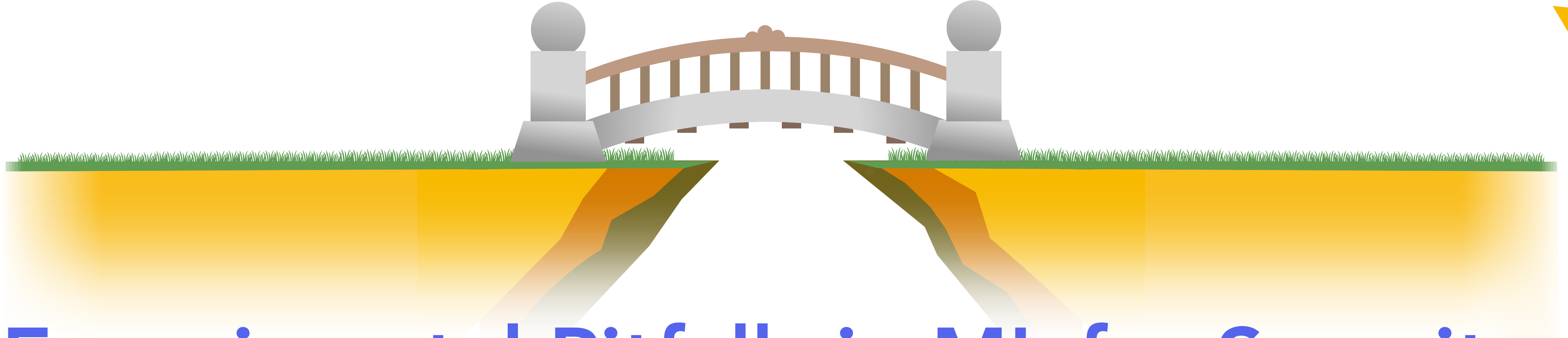


# What Trustworthy ML for Systems Security May Look Like

**Problem-Space Backdoors**

**To identify classifier poisoning vulnerabilities**

(collab. with University of Illinois at Urbana-Champaign)



**Experimental Pitfalls in ML for Security**

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***

**Robust Features**

**To limit effectiveness of adversarial manipulation**

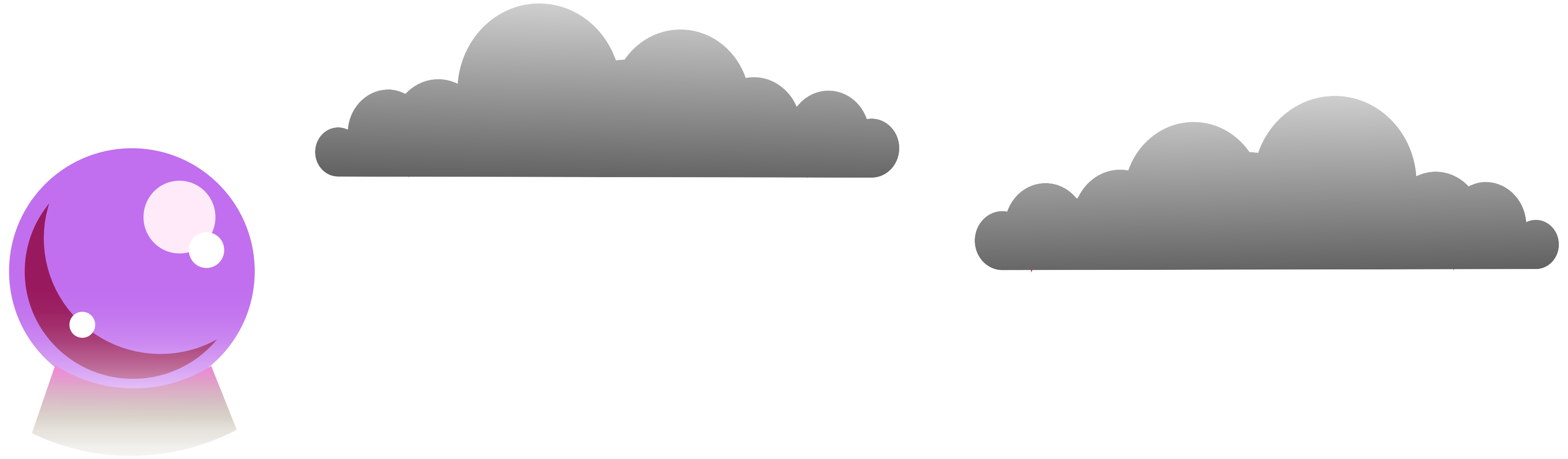
(collab. with Avast)

**Universal Adversarial Perturbations**

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)

# What Trustworthy ML for Systems Security May Look Like



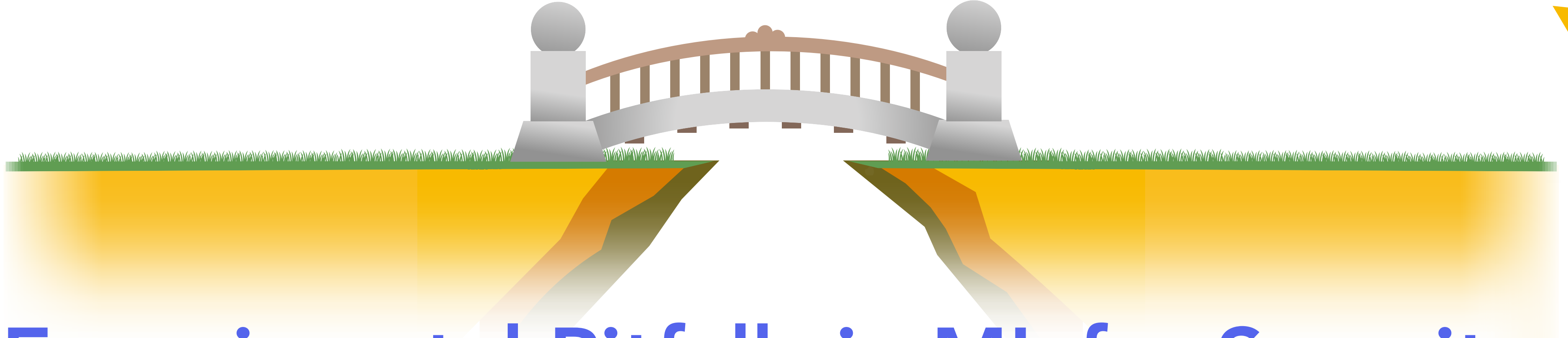
### Forecasting Future Drift

**To proactively anticipate and adapt to concept drift**

### Problem-Space Backdoors

**To identify classifier poisoning vulnerabilities**

(collab. with University of Illinois at Urbana-Champaign)



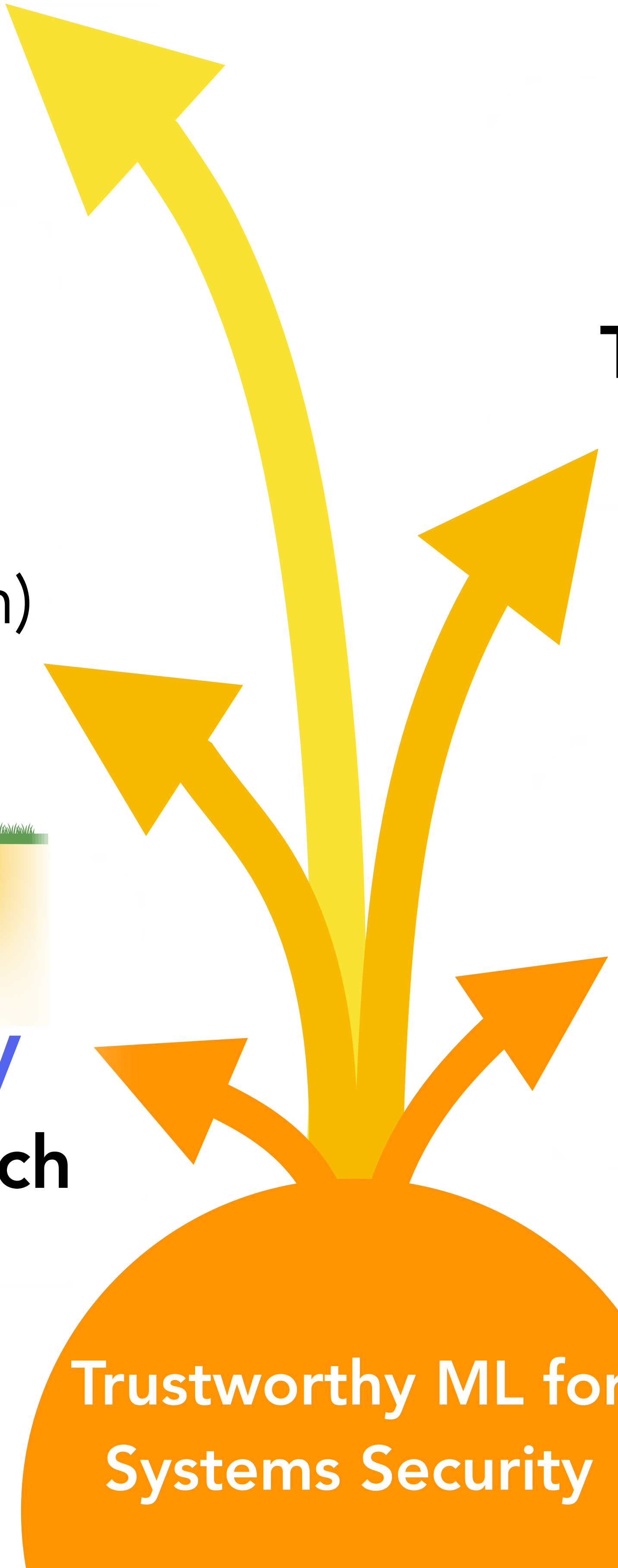
### Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***




### Robust Features

**To limit effectiveness of adversarial manipulation**

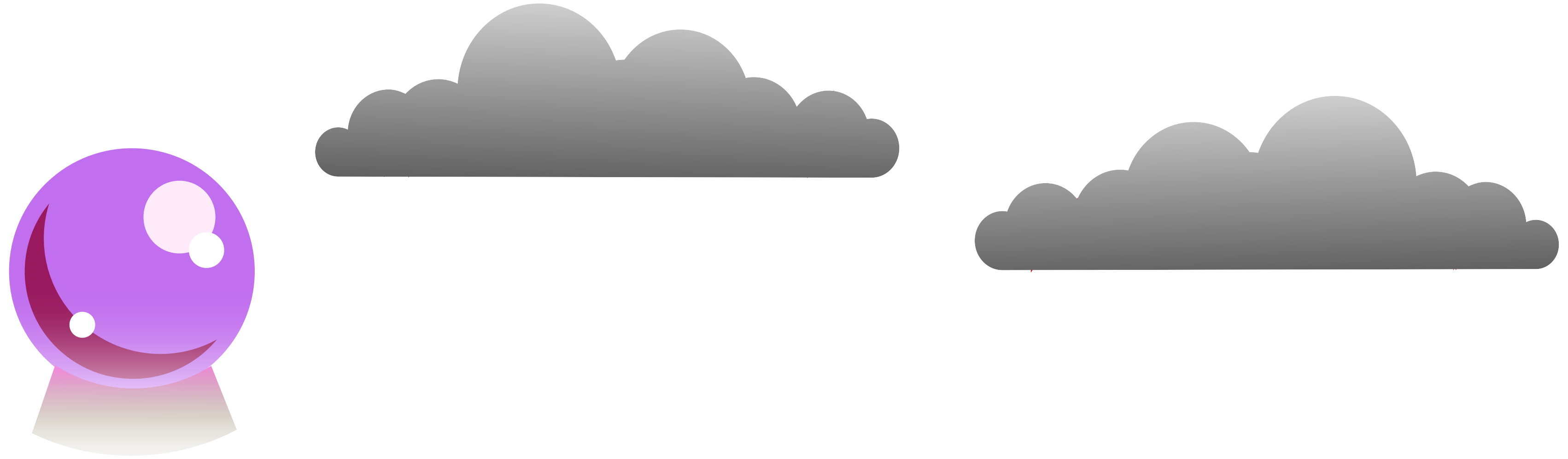
(collab. with Avast)

### Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)

# What Trustworthy ML for Systems Security May Look Like



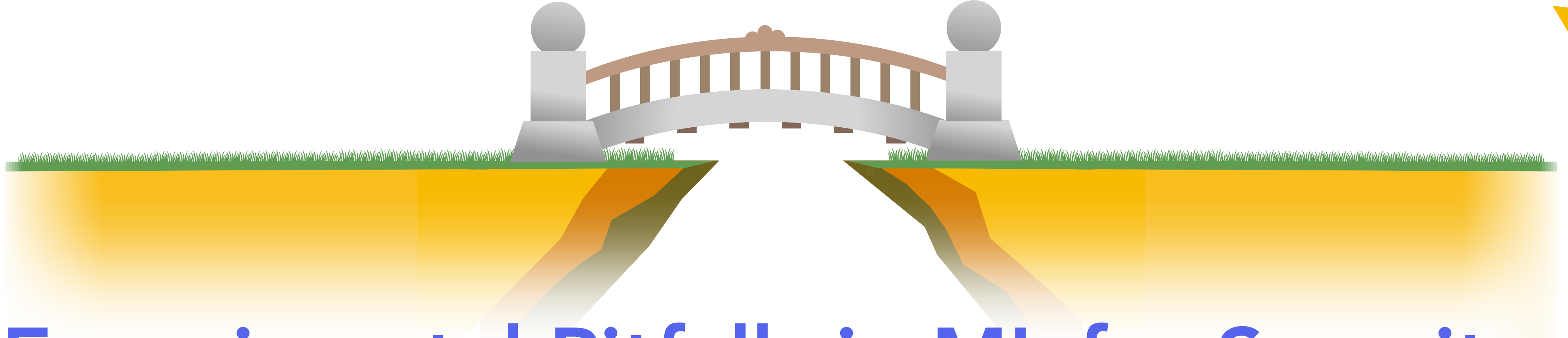
## Forecasting Future Drift

**To proactively anticipate and adapt to concept drift**

## Problem-Space Backdoors

**To identify classifier poisoning vulnerabilities**

(collab. with University of Illinois at Urbana-Champaign)



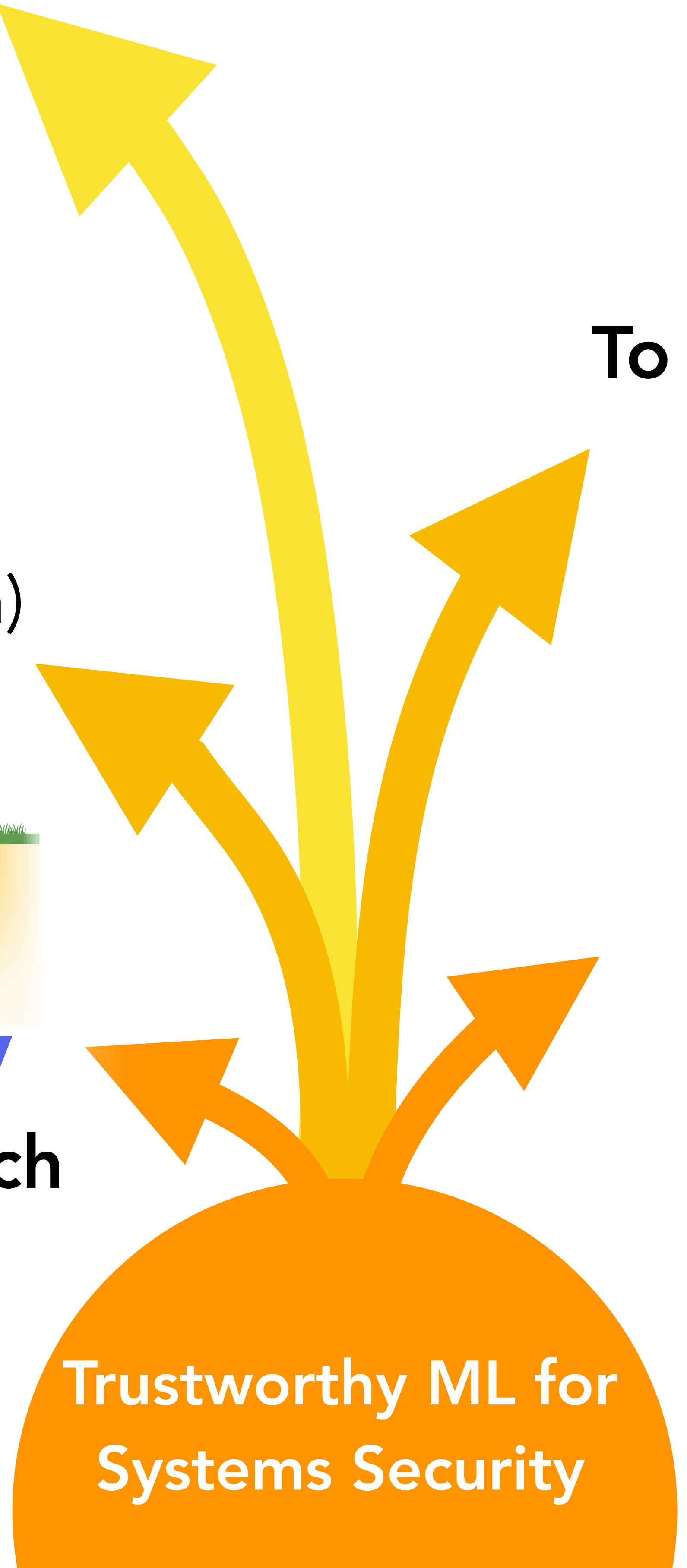
## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***




## Robust Features

**To limit effectiveness of adversarial manipulation**

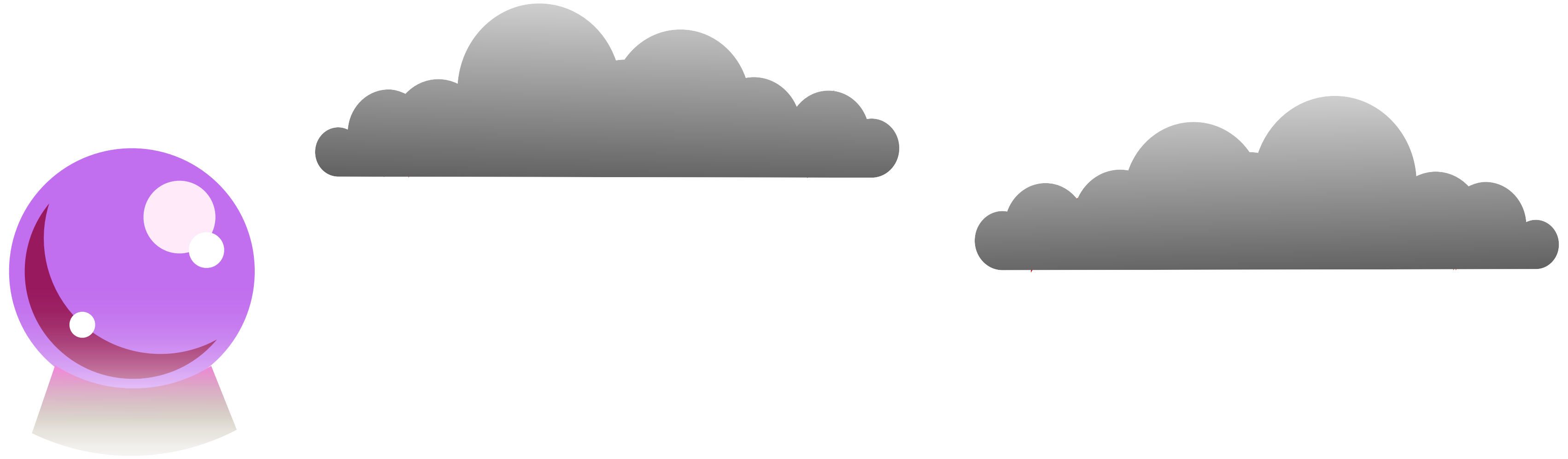
(collab. with Avast)

## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)

# What Trustworthy ML for Systems Security May Look Like



## Semantics Multitask Learning

**Towards a Platonic ideal of binary abstraction**

(collab. with Columbia University)

## Forecasting Future Drift

**To proactively anticipate and adapt to concept drift**

## Robust Features

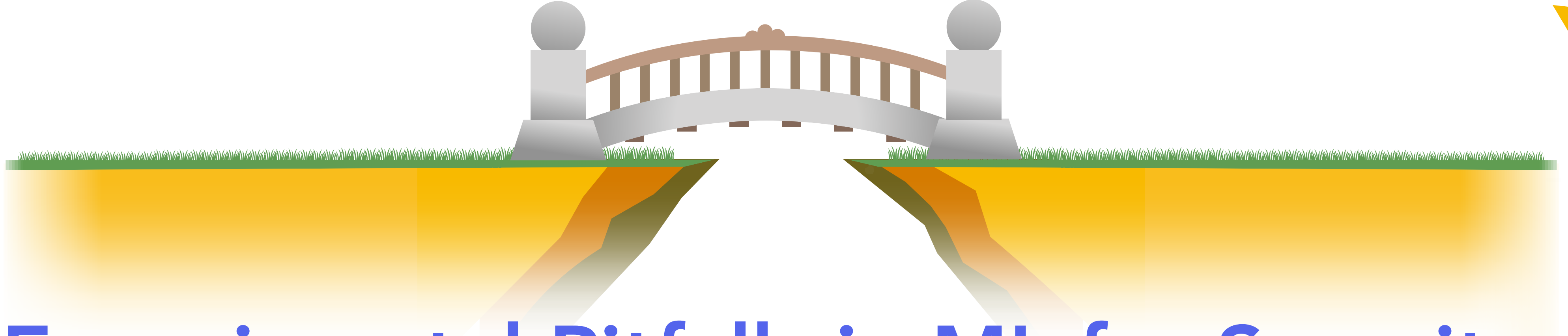
**To limit effectiveness of adversarial manipulation**

(collab. with Avast)

## Problem-Space Backdoors

**To identify classifier poisoning vulnerabilities**

(collab. with University of Illinois at Urbana-Champaign)



## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)

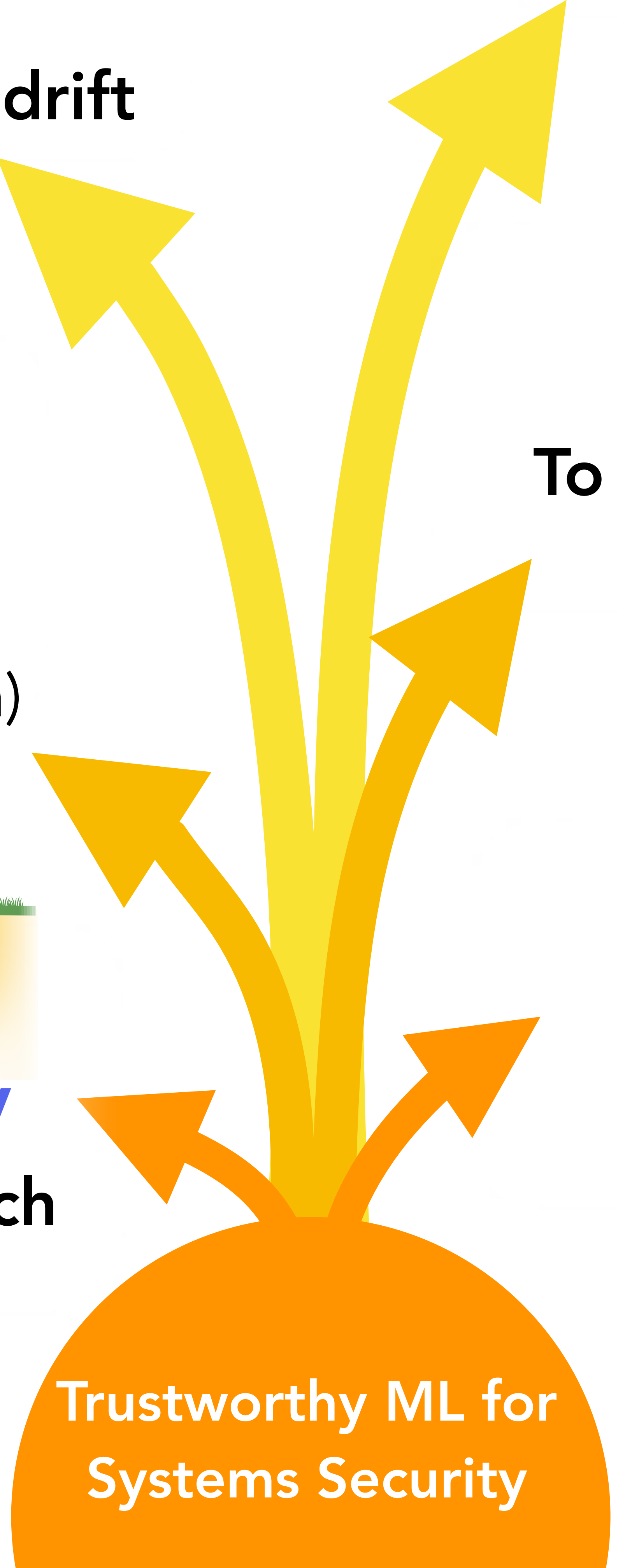
## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

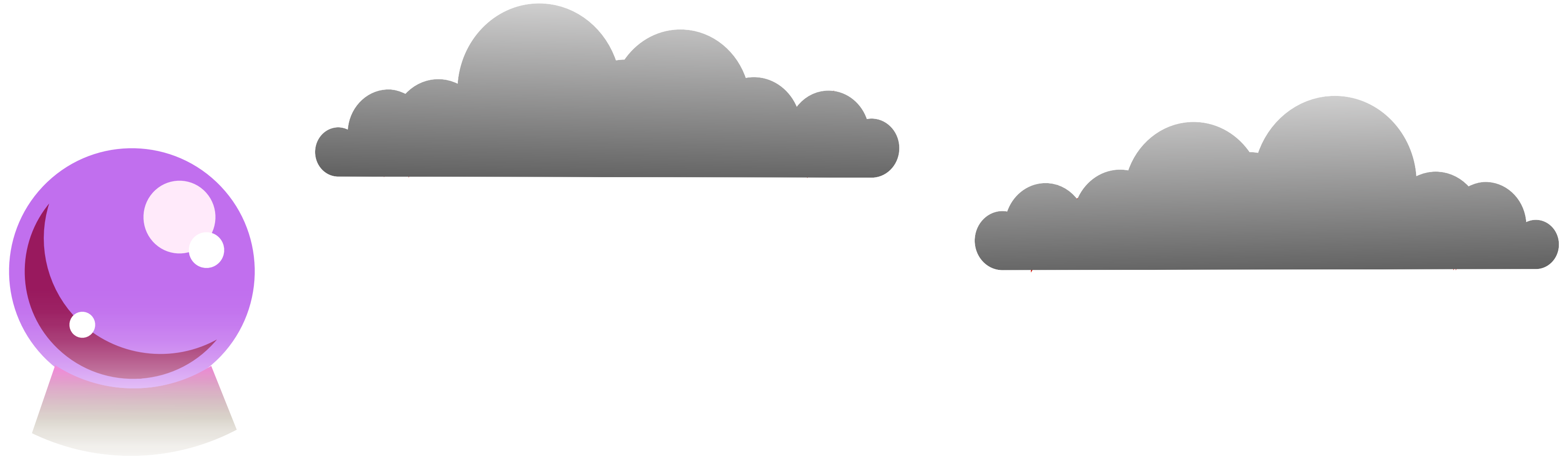
(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

*[USENIX Security 2022]*



# What Trustworthy ML for Systems Security May Look Like



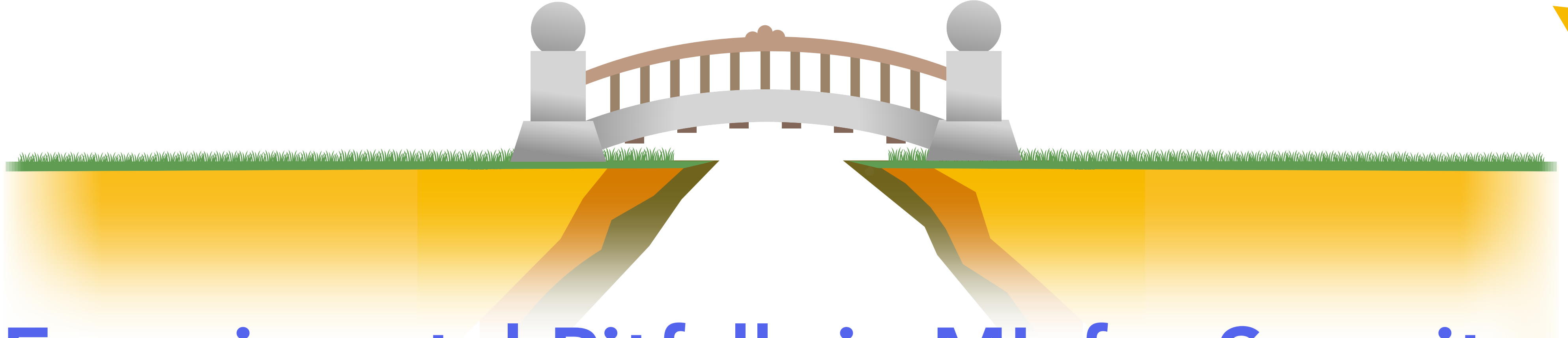
## Forecasting Future Drift

**To proactively anticipate and adapt to concept drift**

## Problem-Space Backdoors

**To identify classifier poisoning vulnerabilities**

(collab. with University of Illinois at Urbana-Champaign)



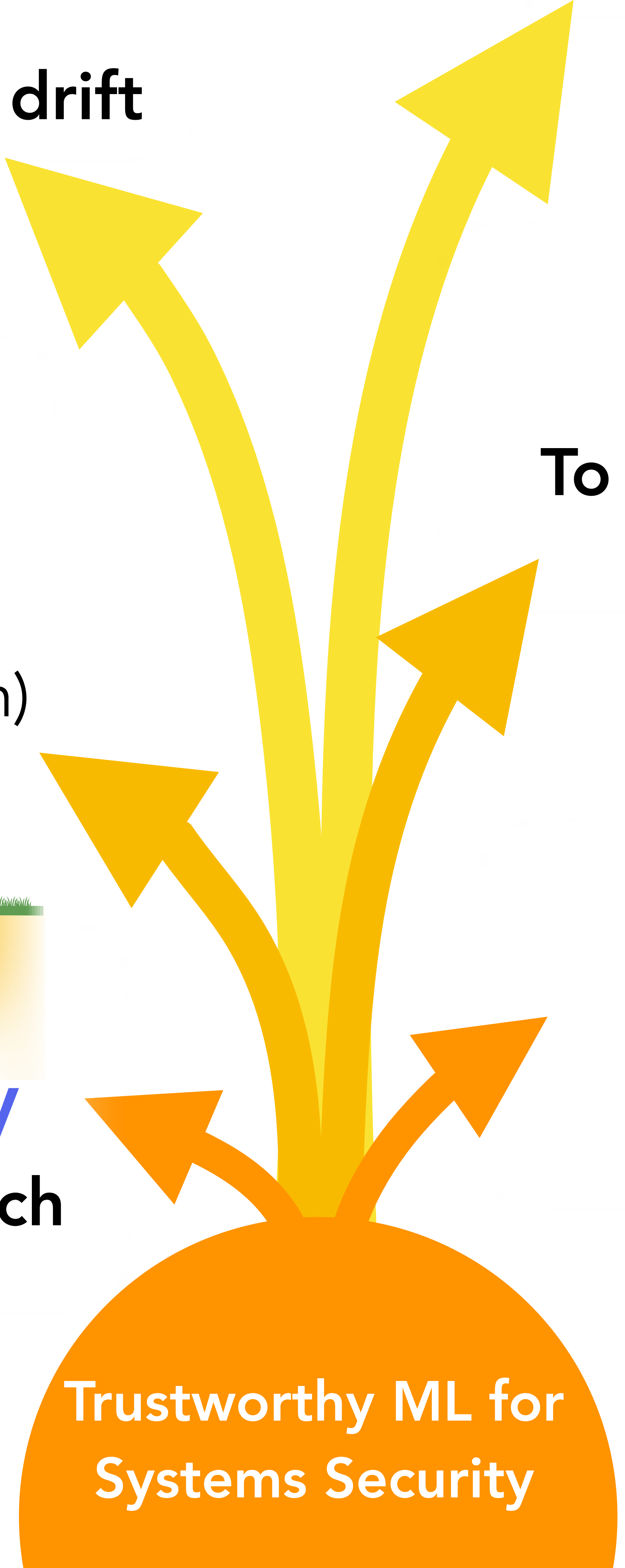
## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***




## Semantics Multitask Learning

**Towards a Platonic ideal of binary abstraction**

(collab. with Columbia University)

## Robust Features

**To limit effectiveness of adversarial manipulation**

(collab. with Avast Software s.r.o.)

## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)

# What Trustworthy ML for Systems Security May Look Like

## Semantics Multitask Learning

**Towards a Platonic ideal of binary abstraction**

(collab. with Columbia University)

## Forecasting Future Drift

**To proactively anticipate and adapt to concept drift**

## Robust Features

**To limit effectiveness of adversarial manipulation**

(collab. with Avast Software s.r.o.)

## Problem-Space Backdoors

**To identify classifier poisoning vulnerabilities**

(collab. with University of Illinois at Urbana-Champaign)

## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)

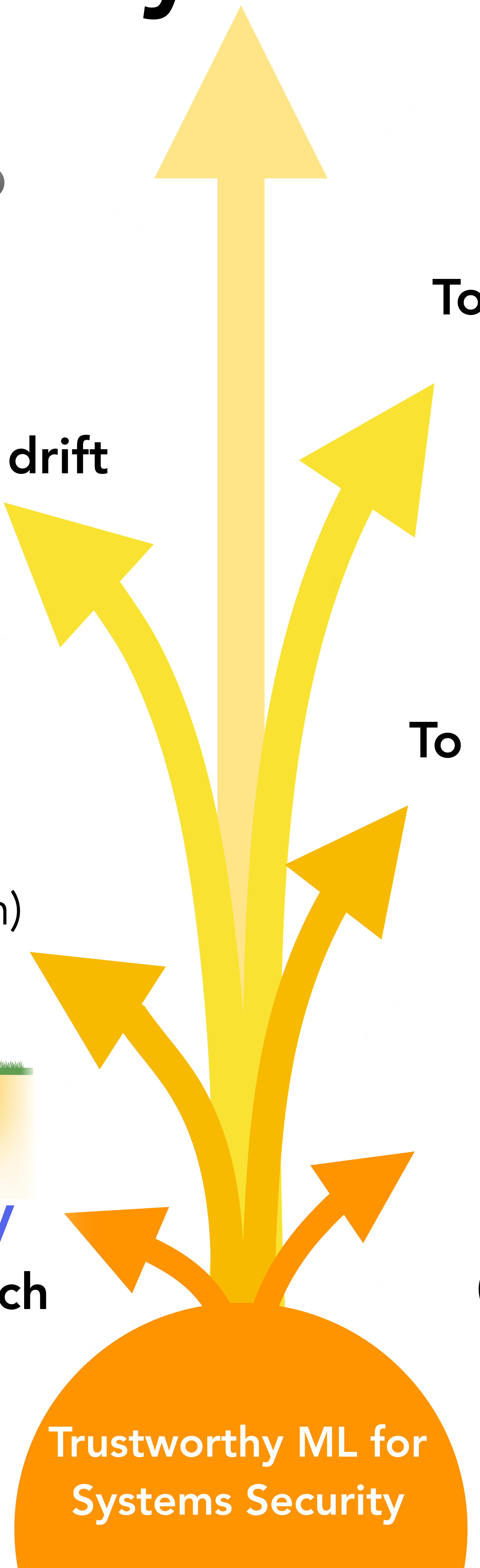
## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***



# Core Team

### Ph.D. Students

Mohamed

Jacopo

Mark

### Ph.D. Alumni

Feargus

### Team-ups

Fabio Pierazzi

KING'S College LONDON

### Current Research Collaborators

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Imperial College London

ZHEJIANG UNIVERSITY 1897

avast

TOSHIBA Leading Innovation

UNIVERSITÀ DEGLI STUDI DI MILANO

Universität der Bundeswehr München

UNIVERSITY OF CAMBRIDGE

Technische Universität Braunschweig

# Core Team

### Ph.D. Students

Mohamed

Jacopo

Mark

### Ph.D. Alumni

Feargus

### Team-ups

Fabio Pierazzi

KING'S College LONDON

### Current Research Collaborators

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Imperial College London

ZHEJIANG UNIVERSITY 1897

avast

TOSHIBA Leading Innovation

UNIVERSITÀ DEGLI STUDI DI MILANO

Universität der Bundeswehr München

UNIVERSITY OF CAMBRIDGE

Technische Universität Braunschweig

# Core Team

### Ph.D. Students

Mohamed

Jacopo

Mark

### Ph.D. Alumni

Feargus

### Team-ups

Fabio Pierazzi

Current Research Collaborators

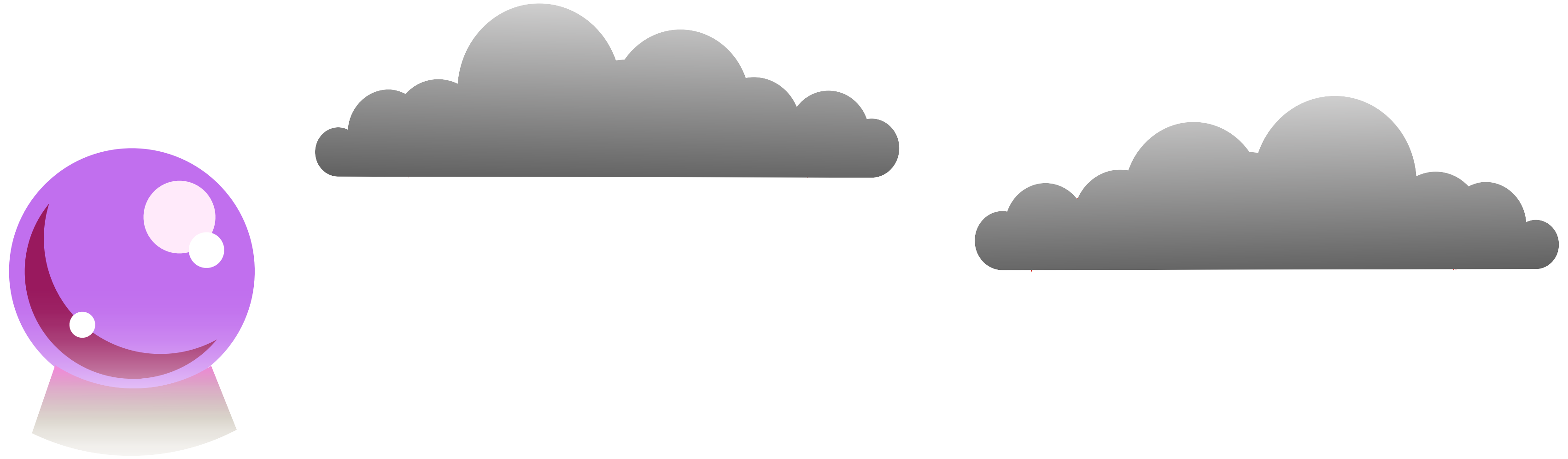
**I am hiring at UCL! :-)**

PhD1

PhD2

PostDoc

# What Trustworthy ML for Systems Security May Look Like



## Semantics Multitask Learning

**Towards a Platonic ideal of binary abstraction**

## Forecasting Future Drift

**To proactively anticipate and adapt to concept drift**

## Robust Features

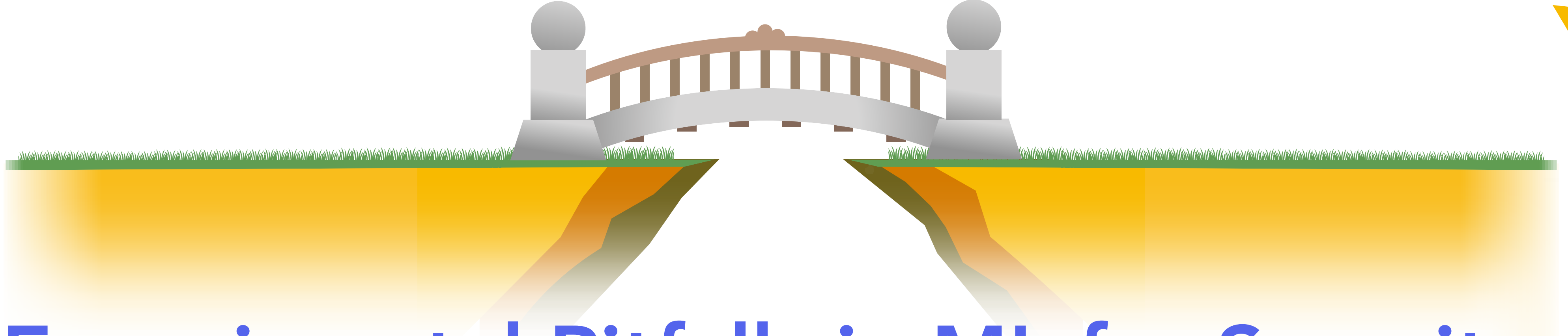
**To limit effectiveness of adversarial manipulation**

(collab. with Avast Software s.r.o.)

## Problem-Space Backdoors

**To identify classifier poisoning vulnerabilities**

(collab. with University of Illinois at Urbana-Champaign)



## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)

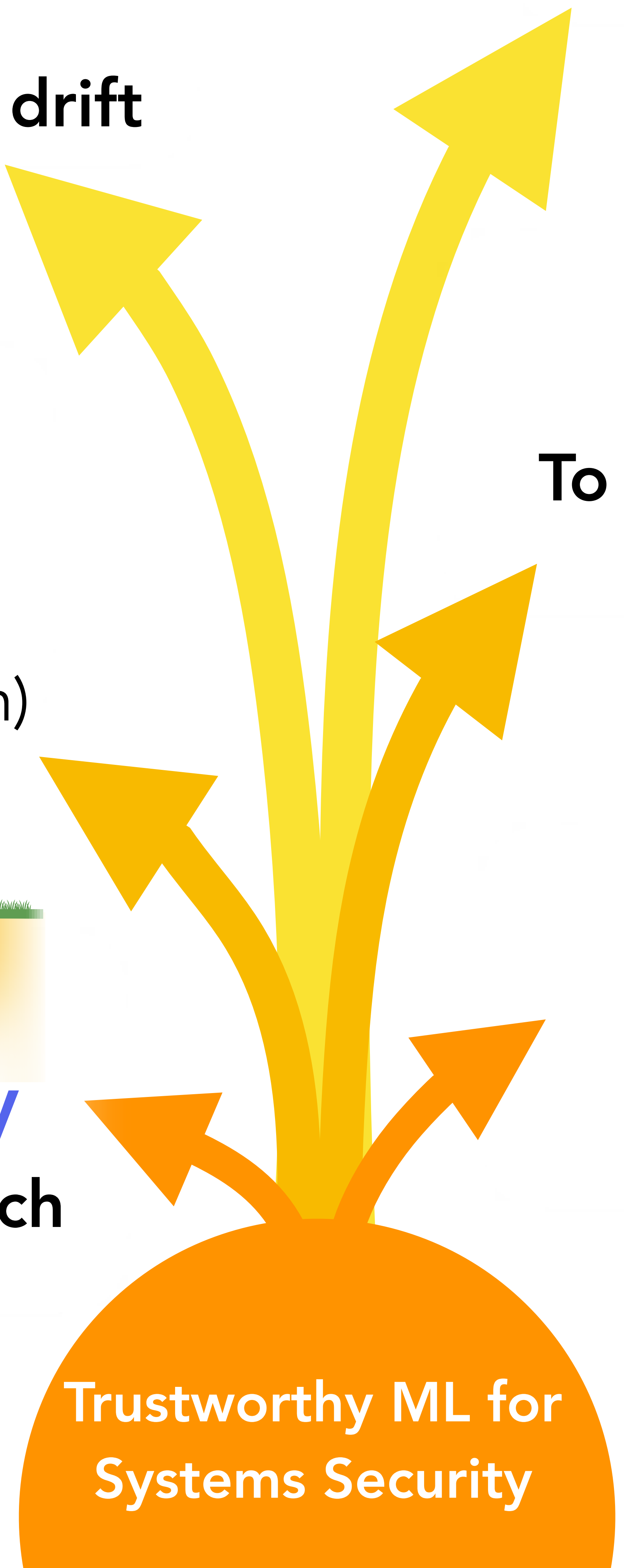
## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

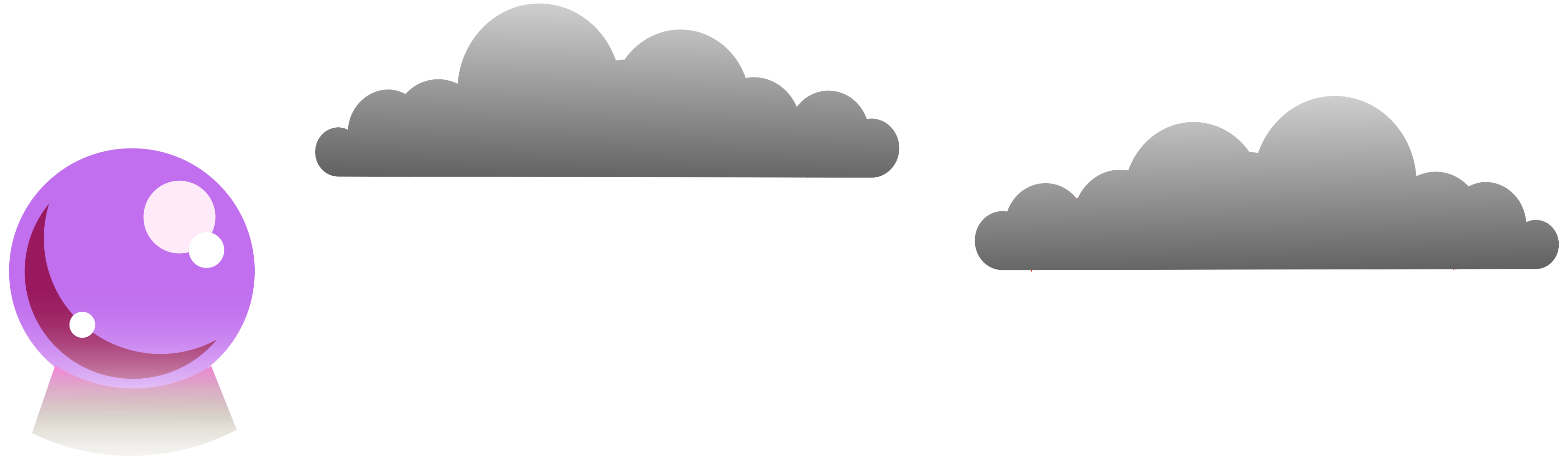
(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***



# What Trustworthy ML for Systems Security May Look Like



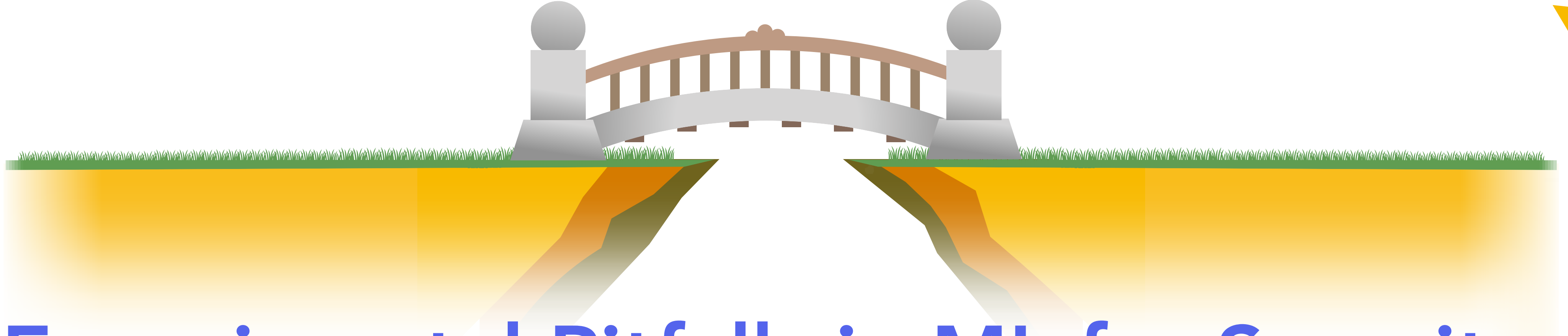
## Forecasting Future Drift

**To proactively anticipate and adapt to concept drift**

## Problem-Space Backdoors

**To identify classifier poisoning vulnerabilities**

(collab. with University of Illinois at Urbana-Champaign)



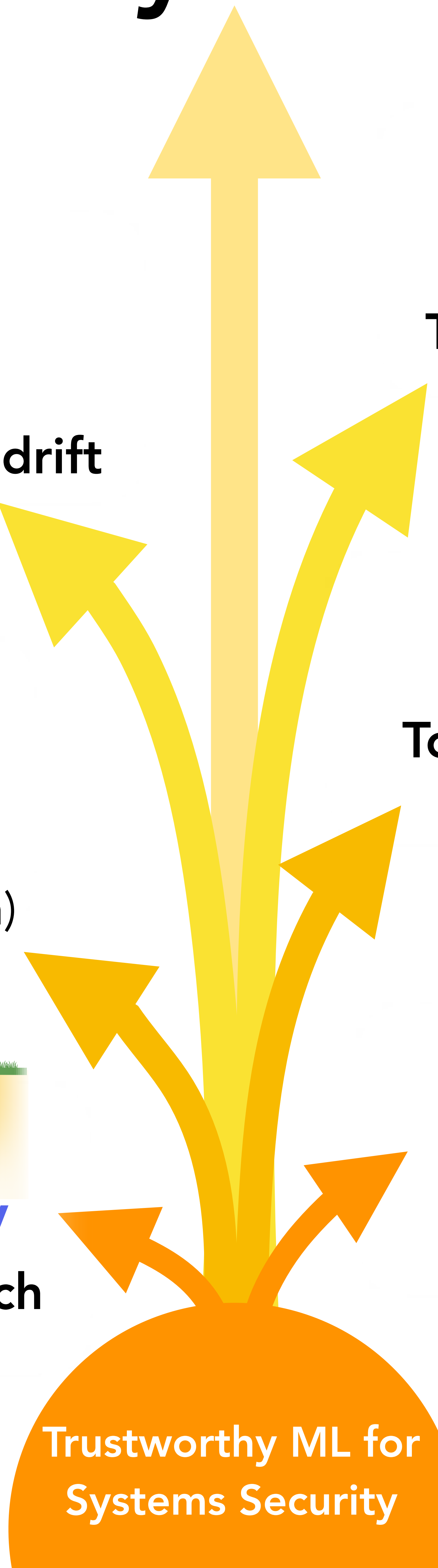
## Experimental Pitfalls in ML for Security

**To provide a solid foundation for empirical research**

(collab. with TU Braunschweig)

*Dos and Don'ts of Machine Learning in Computer Security*

***[USENIX Security 2022]***




## Semantics Multitask Learning

**Towards a Platonic ideal of binary abstraction**

## Robust Features

**To limit effectiveness of adversarial manipulation**

(collab. with Avast Software s.r.o.)

## Universal Adversarial Perturbations

**To identify classifier evasion vulnerabilities**

(collab. with Imperial College London and UniBw)