BODMAS: An Open Dataset for Learning based Temporal Analysis of PE Malware

Deep Learning and Security 2021





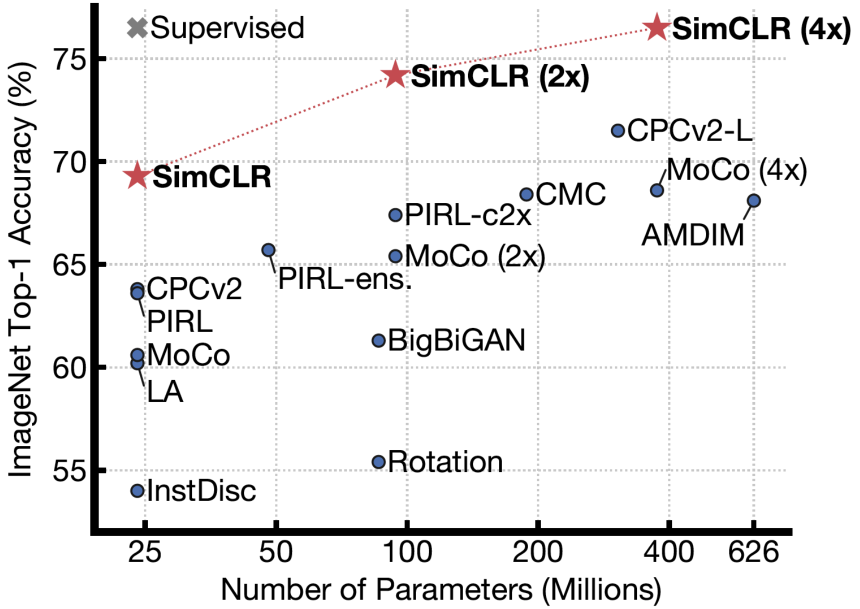
Limin Yang   Arridhana Ciptadi   Ihar Laziuk   Ali Ahmadzadeh   Gang Wang

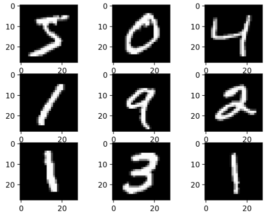
ILLINOIS
Computer Science

BLUEHEXAGON

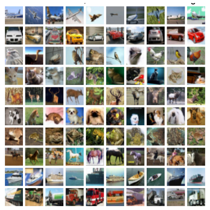
Why Open Malware Dataset?

- Facilitate new research to resolve open challenges
- Easily keep track of the state-of-the-art
- Security community lacks benchmark datasets





MNIST



CIFAR-10

ImageNet

1. A Simple Framework for Contrastive Learning of Visual Representations, ICML, 2020.
2. Gradient-based learning applied to document recognition, IEEE, 1998.

Why Releasing Malware Dataset is Hard?

Legal restrictions

- Benign binaries are often protected by copyright laws

Labeling costs and difficulties

- Time-consuming even for experts
- Anti-malware scanners' results may be proprietary

Security liability and precautions

- Risky to share malware to non-infosec audience

Constant need for new datasets

- Malware evolves and new malware family appears

Ember: An Open Dataset for Training Static PE Malware Machine Learning Models, arXiv, 2018.

What We Did

Legal restrictions

- Release feature vectors (malware + benign); and malware binaries

Labeling costs and difficulties

- In-house analysis + aggregate multiple antivirus vendors' labels
- ~1% labeled via manual analysis

Security liability and precautions

- We only share disarmed malware with researchers upon request

Constant need for new datasets

- We release a more recent dataset sampled from Blue Hexagon

Public PE Malware Datasets

Dataset	Malware Time	# Families	# Samples	# Benign	# Malware	Malware Binaries	Feature Vectors
Microsoft	N/A (Before 2015)	9	10,868	0	10,868	◑	○
UCSB-Packed	01/2017–03/2018	N/A	341,445	109,030	232,415	●	○
Ember*	01/2017–12/2018	◑	2,050,000	750,000	800,000	○	●
SOREL-20M	01/2017–04/2019	N/A	19,724,997	9,762,177	9,962,820	●	●
BODMAS	**08/2019–09/2020**	**581**	134,435	77,142	57,293	●	●

* Ember combines Ember2017 and Ember2018 and duplicates were removed.

Public PE Malware Datasets

Dataset	Malware Time	# Families	# Samples	# Benign	# Malware	Malware Binaries	Feature Vectors
Microsoft	N/A (Before 2015)	9	10,868	0	10,868	◑	○
UCSB-Packed	01/2017–03/2018				232,415	●	○
Ember*	01/2017–12/2018	◑	2,050,000	750,000	800,000	○	●
SOREL-20M	01/2017–04/2019	N/A	19,724,997	9,762,177	9,962,820	●	●
BODMAS	**08/2019–09/2020**	**581**	134,435	77,142	57,293	●	●

Existing datasets are slightly outdated

* Ember combines Ember2017 and Ember2018 and duplicates were removed.

Public PE Malware Datasets

Dataset	Malware Time	# Families	# Samples	# Benign	# Malware	Malware Binaries	Feature Vectors
Microsoft	N/A (Before 2015)	9	10,868	0	10,868	◑	○
UCSB-Packed	01/2017–03/2018	N/A	341			●	○
Ember*	01/2017–12/2018	◑	2,050,000	750,000	800,000	○	●
SOREL-20M	01/2017–04/2019	N/A	19,724,997	9,762,177	9,962,820	●	●
BODMAS	**08/2019–09/2020**	**581**	134,435	77,142	57,293	●	●

Most do not have curated families

* Ember combines Ember2017 and Ember2018 and duplicates were removed.

Public PE Malware Datasets

Dataset	Malware Time	# Families	# Samples	# Benign	# Malware	Malware Binaries	Feature Vectors
Microsoft	N/A (Before 2015)	9	10,868	0	10,868	◑	○
UCSB-Packed	01/2017–03/2018	N/A	341,445	109,030	232,415	●	○
Ember*	01/2017–12/2018	◑	2,050,000	[illegible]	[illegible]	[illegible]	●
SOREL-20M	01/2017–04/2019	N/A	19,724,997	[illegible]	[illegible]	●	●
BODMAS	**08/2019–09/2020**	**581**	134,435	77,142	57,293	●	●

Consistent format for longitudinal analysis

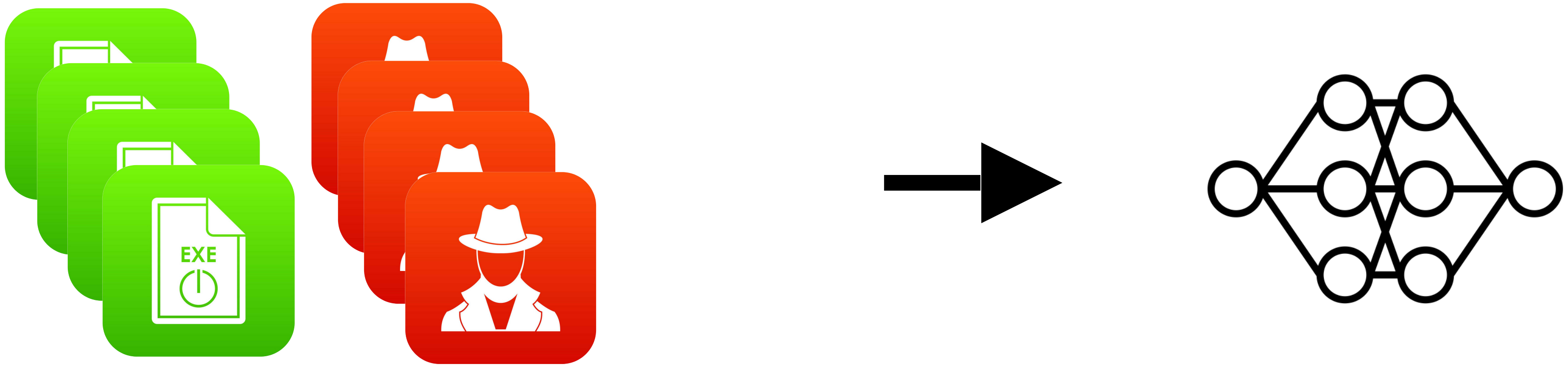
* Ember combines Ember2017 and Ember2018 and duplicates were removed.

Outline

- ~~Introduction~~
- Open problem: concept drift in binary classifiers across time
- Open problem: concept drift in malware family attribution

A Binary Malware Classification Model

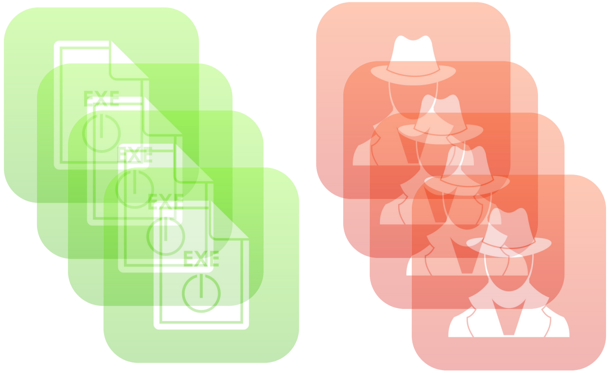
1. Train



Benign Malware

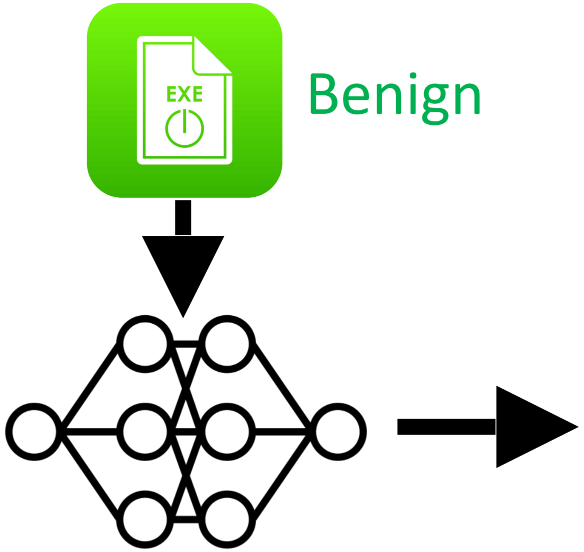
A Binary Malware Classification Model

1. Train



Benign Malware

2. Predict

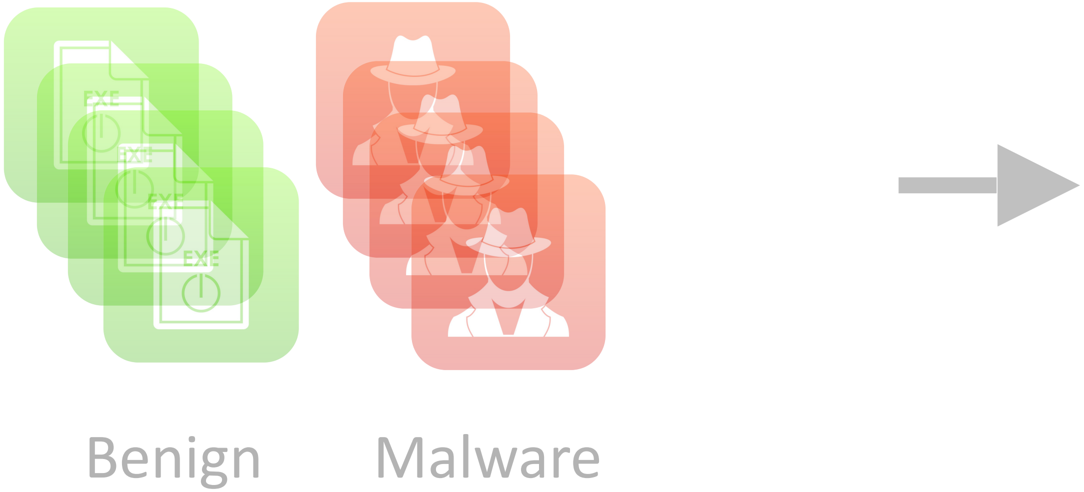


Benign

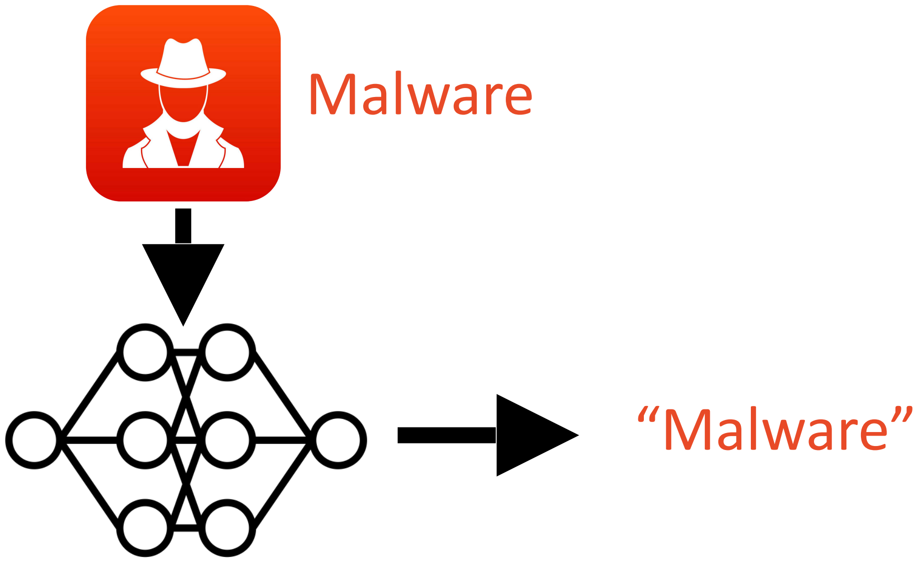
“Benign”

A Binary Malware Classification Model

1. Train

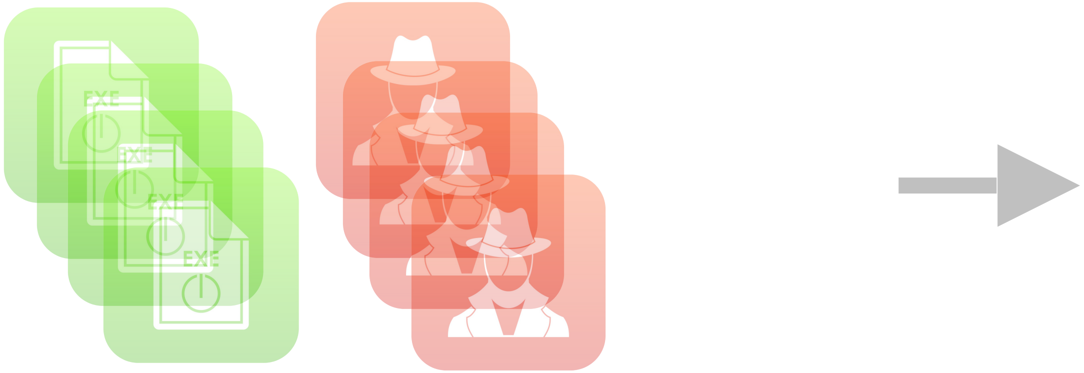


2. Predict



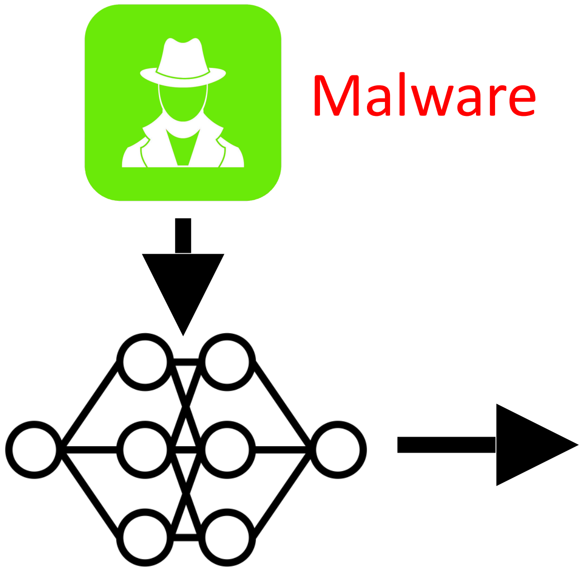
A Binary Malware Classification Model

1. Train



Benign Malware

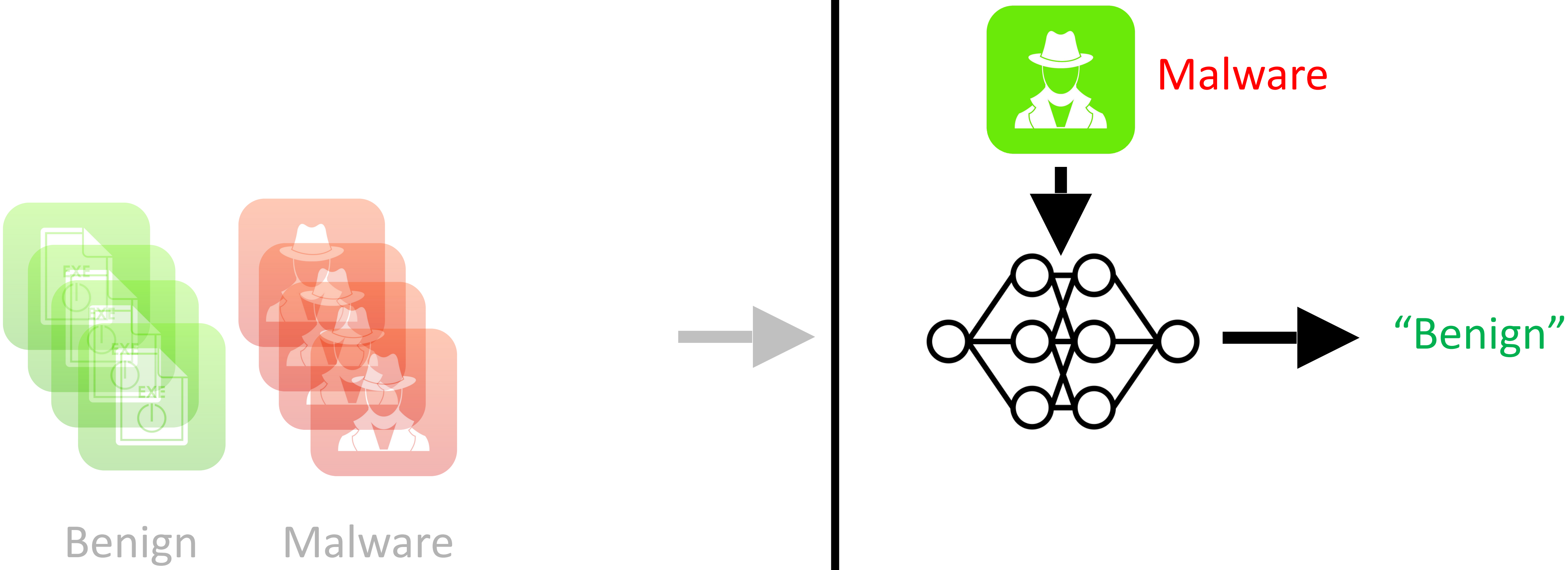
2. Predict



A Binary Malware Classification Model

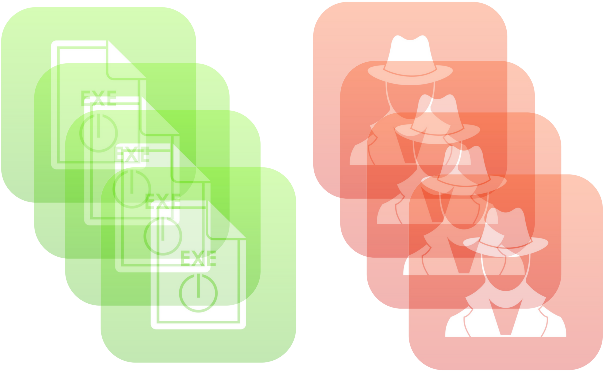
1. Train

2. Predict



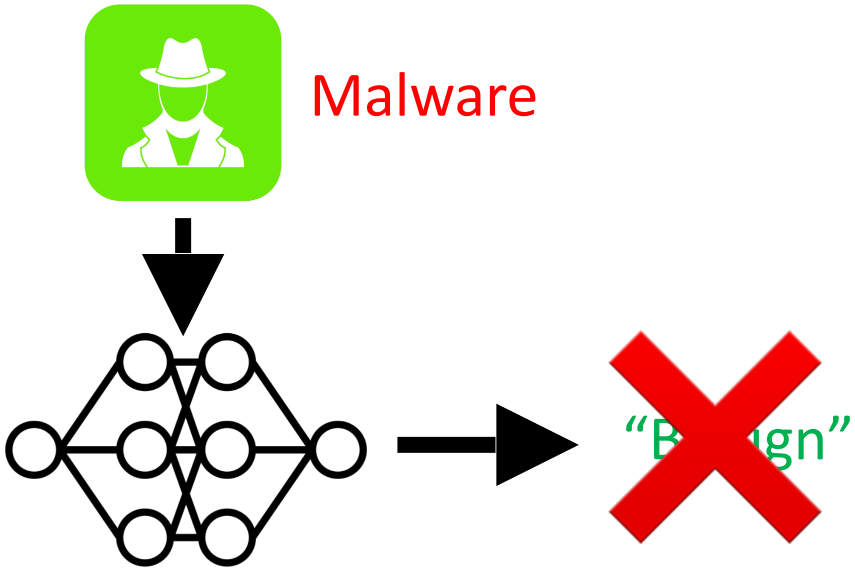
A Binary Malware Classification Model

1. Train



Benign Malware

2. Predict

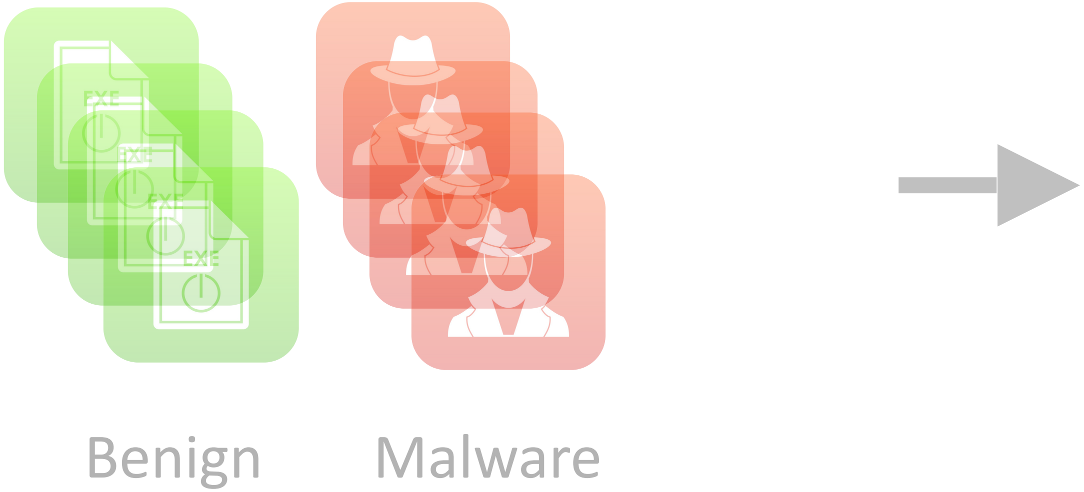


Concept Drift!

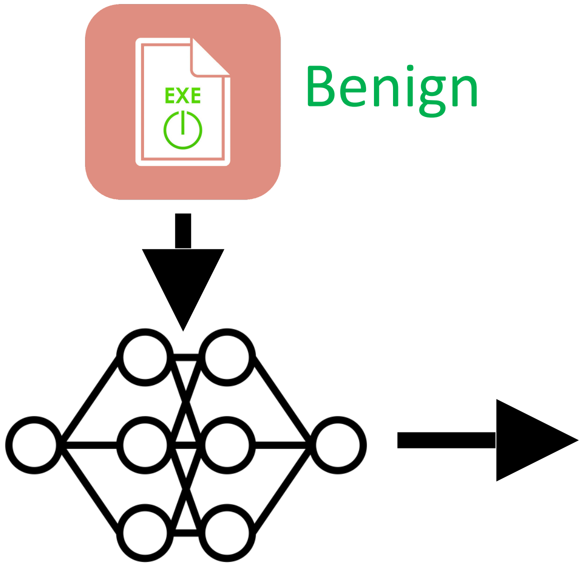
(In-class evolution)

A Binary Malware Classification Model

1. Train

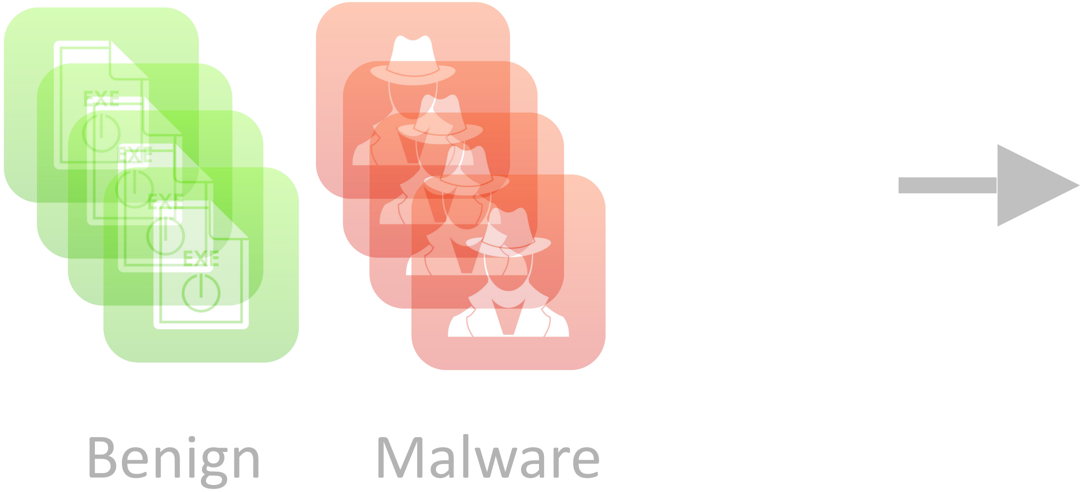


2. Predict

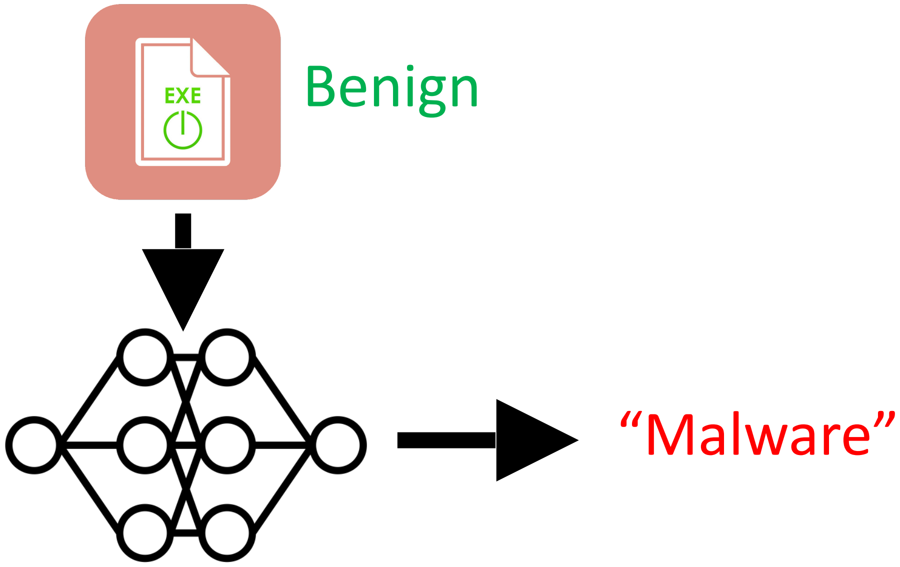


A Binary Malware Classification Model

1. Train

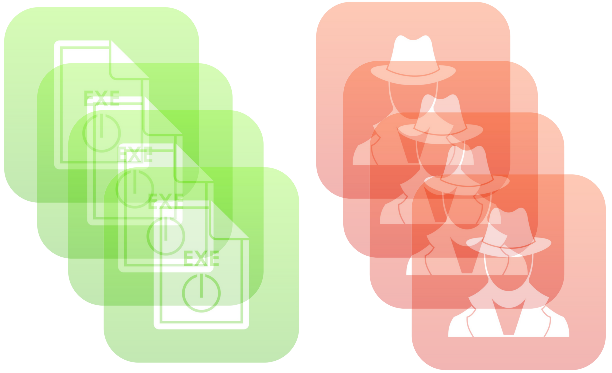


2. Predict



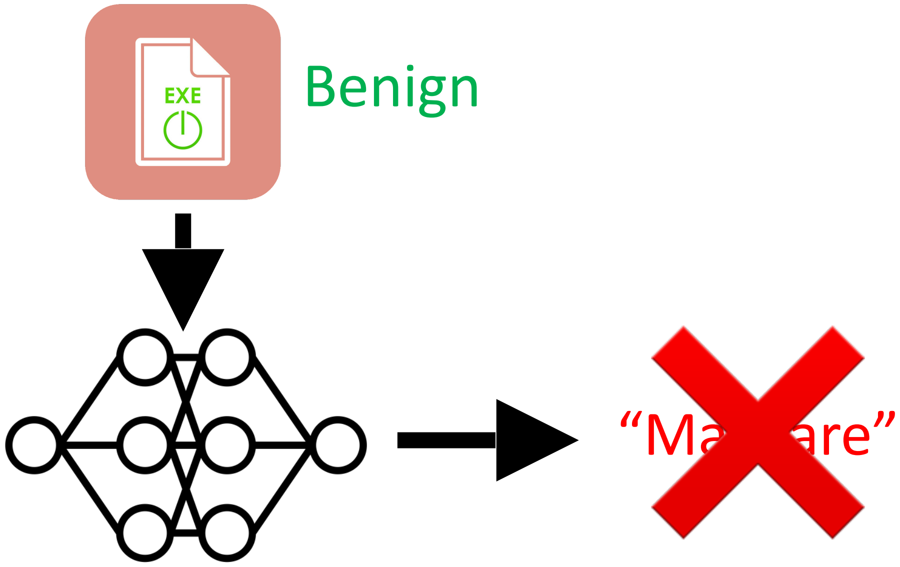
A Binary Malware Classification Model

1. Train



Benign Malware

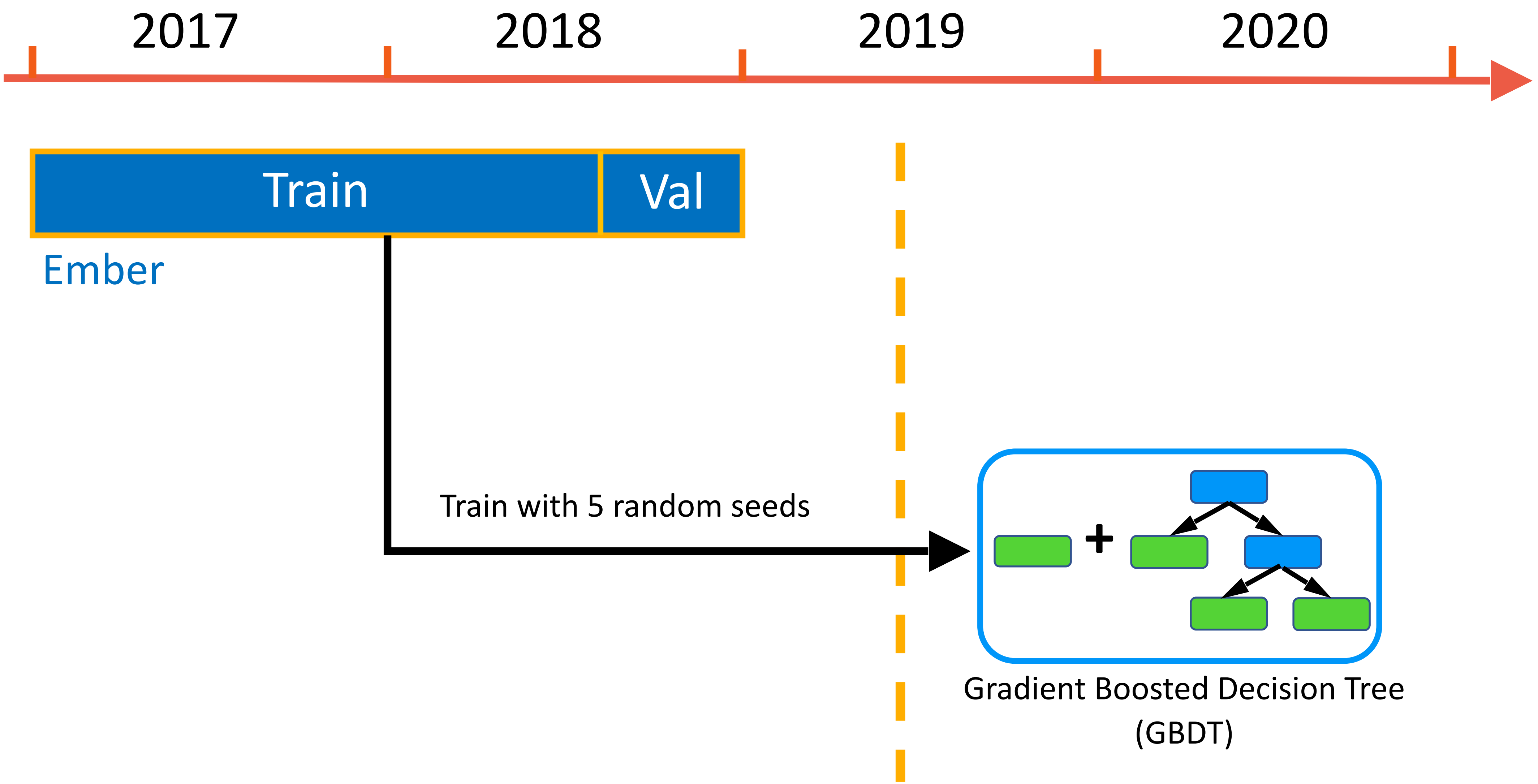
2. Predict



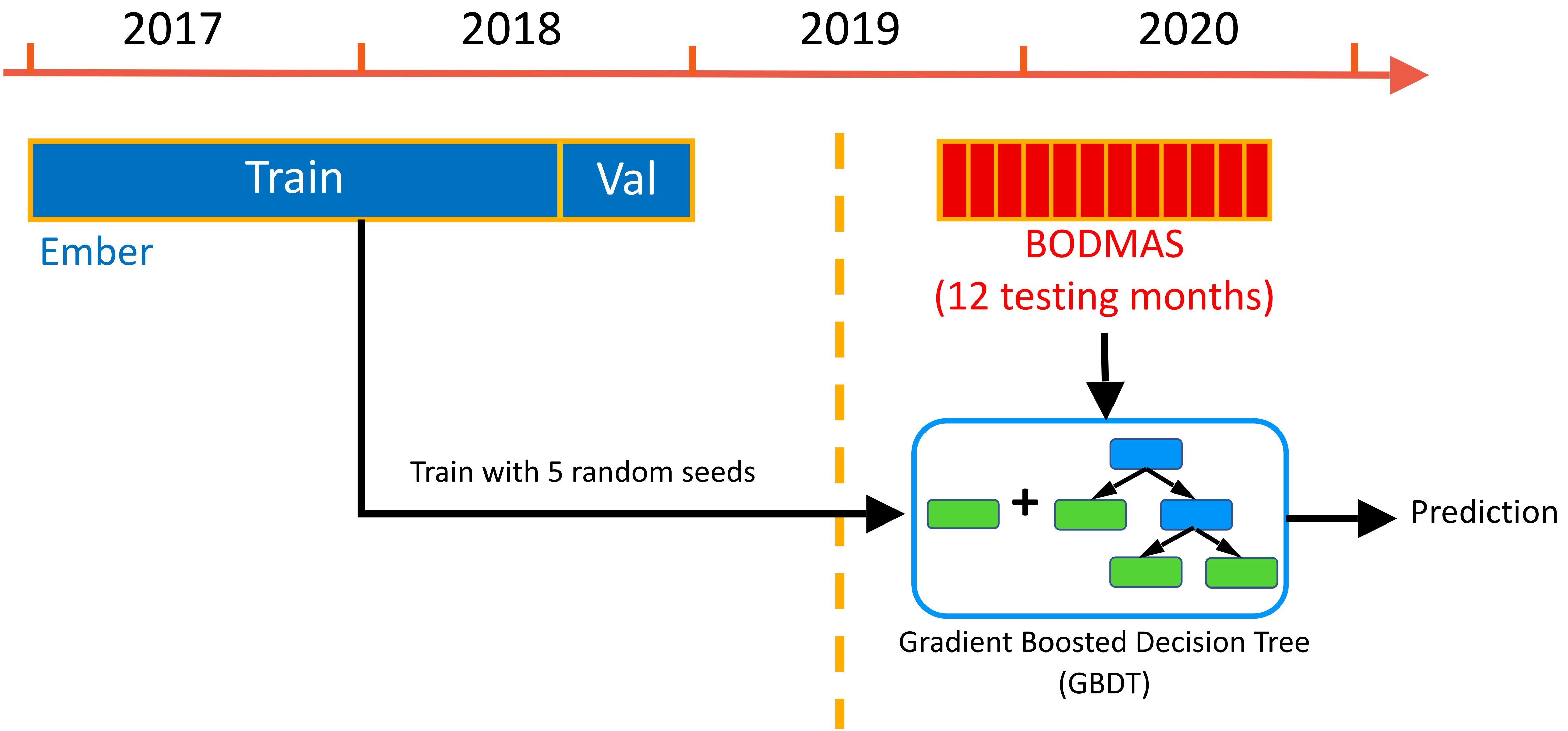
Concept Drift!

(In-class evolution)

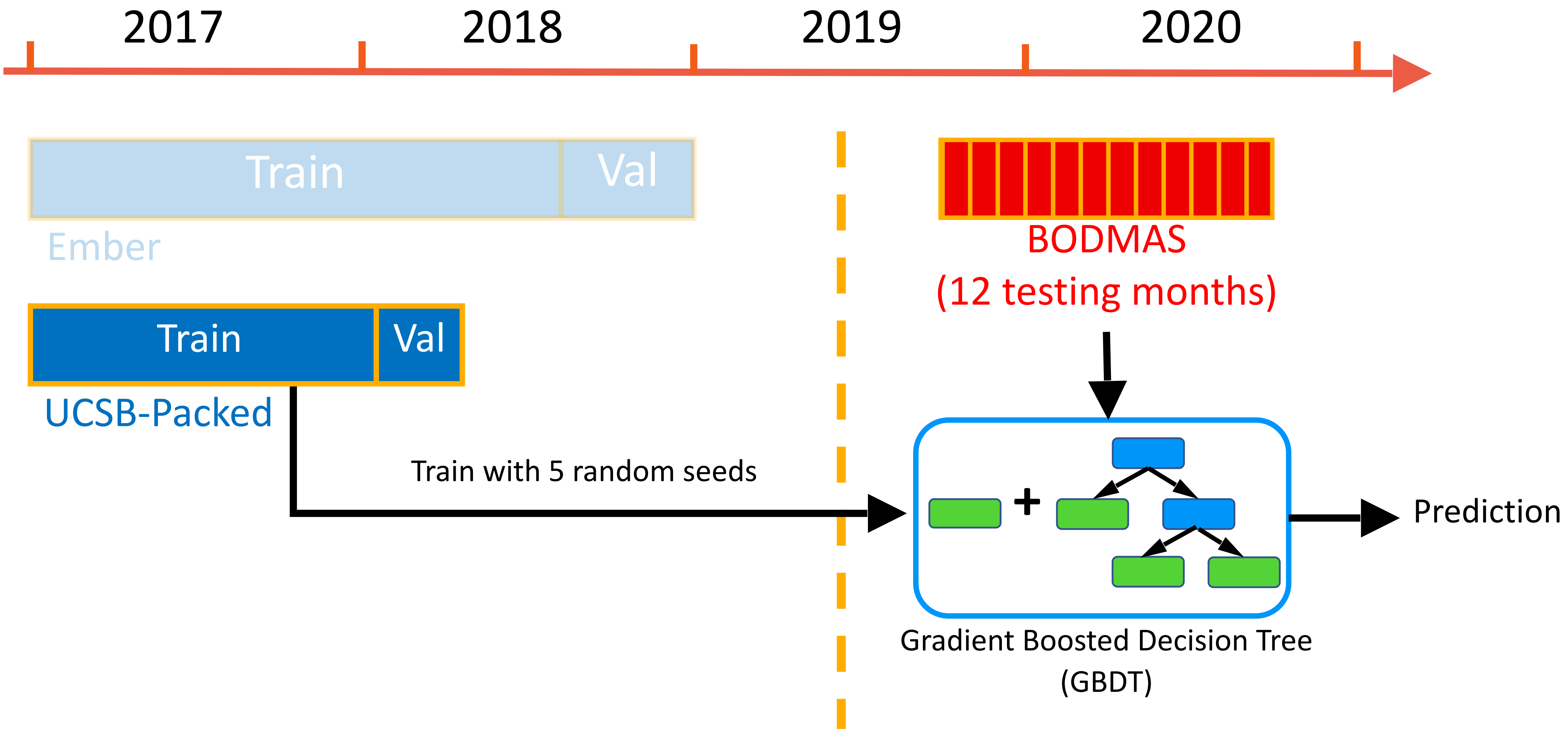
Experiment: Concept Drift Across Datasets



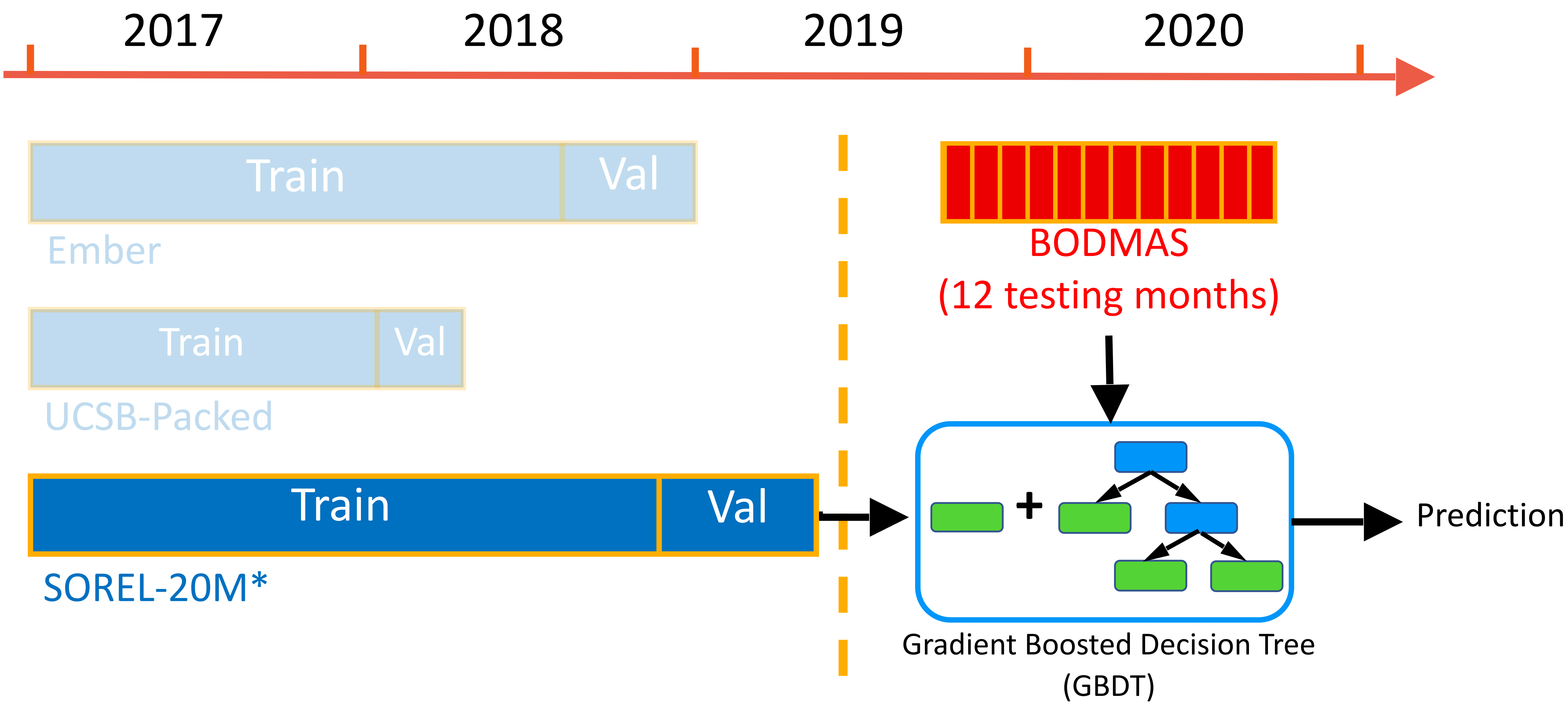
Experiment: Concept Drift Across Datasets



Experiment: Concept Drift Across Datasets



Experiment: Concept Drift Across Datasets



* For SOREL-20M, we use their pre-trained GBDT model and DNN model due to resource and time constraints.

Impact of Concept Drift

Phase	Ember-GBDT		UCSB-GBDT		SOREL-GBDT		SOREL-DNN	
	FPR	F_1	FPR	F_1	FPR	F_1	FPR	F_1
Val	0.10%	98.6%	0.10%	92.1%	0.10%	98.8%	0.10%	98.0%
10/19	0.00%	94.9%	0.03%	71.1%	0.09%	97.7%	0.31%	94.8%
11/19	0.00%	95.8%	0.02%	81.0%	0.05%	98.1%	0.40%	96.2%
12/19	0.01%	96.6%	0.06%	84.9%	0.24%	98.3%	0.45%	96.8%
01/20	0.18%	93.7%	0.12%	78.0%	2.14%	96.3%	2.27%	95.4%
02/20	0.07%	93.4%	0.33%	68.3%	4.82%	95.7%	6.68%	93.2%
03/20	0.01%	95.8%	0.01%	75.3%	0.13%	98.1%	0.35%	96.0%
04/20	0.00%	97.0%	0.02%	80.8%	0.14%	98.9%	0.26%	97.3%
05/20	0.00%	97.5%	0.05%	85.7%	0.13%	98.6%	0.29%	96.0%
06/20	0.01%	97.8%	0.04%	83.2%	0.22%	98.9%	0.43%	96.7%
07/20	0.01%	96.4%	0.03%	66.2%	0.07%	98.7%	0.33%	93.9%
08/20	0.01%	92.9%	0.02%	47.2%	0.06%	96.0%	0.10%	85.9%
09/20	0.02%	92.1%	0.03%	56.0%	0.08%	95.7%	0.13%	82.9%

Impact of Concept Drift

Phase	Ember-GBDT		UCSB-GBDT		SOREL-GBDT		SOREL-DNN	
	FPR	F_1	FPR	F_1	FPR	F_1	FPR	F_1
Val	0.10%	98.6%	0.10%	92.1%	0.10%	98.8%	0.10%	98.0%
10/19	0.00%	94.9%	0.03%	71.1%	0.09%	97.7%	0.31%	94.8%
11/19	0.00%	95.8%	0.02%	81.0%	0.05%	98.1%	0.40%	96.2%
12/19	0.01%	96.6%	0.06%	84.9%	0.24%	98.3%	0.45%	96.8%
01/20	0.18%	93.7%	0.12%	78.0%	2.14%	96.3%	2.27%	95.4%
02/20	0.07%	93.4%	0.33%	68.3%	4.82%	95.7%	6.68%	93.2%
03/20	0.01%	95.8%	0.01%	75.3%	0.13%	98.1%	0.35%	96.0%
04/20	0.00%	97.0%	0.02%	80.8%	0.14%	98.9%	0.26%	97.3%
05/20	0.00%	97.5%	0.05%	85.7%	0.13%	98.6%	0.29%	96.0%
06/20	0.01%	97.8%	0.04%	83.2%	0.22%	98.9%	0.43%	96.7%
07/20	0.01%	96.4%	0.03%	66.2%	0.07%	98.7%	0.33%	93.9%
08/20	0.01%	92.9%	0.02%	47.2%	0.06%	96.0%	0.10%	85.9%
09/20	0.02%	92.1%	0.03%	56.0%	0.08%	95.7%	0.13%	82.9%

Impact of Concept Drift

Phase	Ember-GBDT		UCSB-GBDT		SOREL-GBDT		SOREL-DNN	
	FPR	F_1	FPR	F_1	FPR	F_1	FPR	F_1
Val	0.10%	98.6%	0.10%	92.1%	0.10%	98.8%	0.10%	98.0%
10/19	0.00%	94.9%	0.03%	71.1%	0.09%	97.7%	0.31%	94.8%
11/19	0.00%	95.8%	0.02%	81.0%	0.05%	98.1%	0.40%	96.2%
12/19	0.01%	96.6%	0.06%	84.9%	0.24%	98.3%	0.45%	96.8%
01/20	0.18%	93.7%	0.12%	78.0%	2.14%	96.3%	2.27%	95.4%
02/20	0.07%	93.4%	0.33%	68.3%	4.82%	95.7%	6.68%	93.2%
03/20	0.01%	95.8%	0.01%	75.3%	0.13%	98.1%	0.35%	96.0%
04/20	0.00%	97.0%	0.02%	80.8%	0.14%	98.9%	0.26%	97.3%
05/20	0.00%	97.5%	0.05%	85.7%	0.13%	98.6%	0.29%	96.0%
06/20	0.01%	97.8%	0.04%	83.2%	0.22%	98.9%	0.43%	96.7%
07/20	0.01%	96.4%	0.03%	66.2%	0.07%	98.7%	0.33%	93.9%
08/20	0.01%	92.9%	0.02%	47.2%	0.06%	96.0%	0.10%	85.9%
09/20	0.02%	92.1%	0.03%	56.0%	0.08%	95.7%	0.13%	82.9%

Impact of Concept Drift

Phase	Ember-GBDT		UCSB-GBDT		SOREL-GBDT		SOREL-DNN	
	FPR	F_1	FPR	F_1	FPR	F_1	FPR	F_1
Val	0.10%	98.6%	0.10%	92.1%	0.10%	98.8%	0.10%	98.0%
10/19	0.00%	94.9%	0.03%	71.1%	0.09%	97.7%	0.31%	94.8%
11/19	0.00%	95.8%	0.02%	81.0%	0.05%	98.1%	0.40%	96.2%
12/19	0.01%	96.6%	0.06%	84.9%	0.24%	98.3%	0.45%	96.8%
01/20	0.18%	93.7%	0.12%	78.0%	2.14%	96.3%	2.27%	95.4%
02/20	0.07%	93.4%	0.33%	68.3%	4.82%	95.7%	6.68%	93.2%
03/20	0.01%	95.8%	0.01%	75.3%	0.13%	98.1%	0.35%	96.0%
04/20	0.00%	97.0%	0.02%	80.8%	0.14%	98.9%	0.26%	97.3%
05/20	0.00%	97.5%	0.05%	85.7%	0.13%	98.6%	0.29%	96.0%
06/20	0.01%	97.8%	0.04%	83.2%	0.22%	98.9%	0.43%	96.7%
07/20	0.01%	96.4%	0.03%	66.2%	0.07%	98.7%	0.33%	93.9%
08/20	0.01%	92.9%	0.02%	47.2%	0.06%	96.0%	0.10%	85.9%
09/20	0.02%	92.1%	0.03%	56.0%	0.08%	95.7%	0.13%	82.9%

Impact of Concept Drift

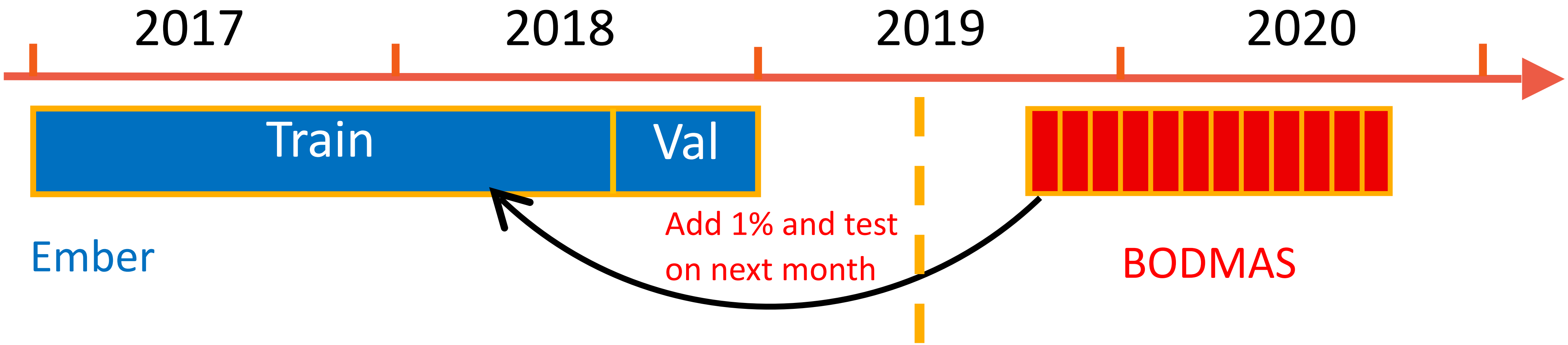
Phase	Ember-GBDT		UCSB-GBDT		SOREL-GBDT		SOREL-DNN	
	FPR	F_1	FPR	F_1	FPR	F_1	FPR	F_1
Val	0.10%	98.6%	0.10%	92.1%	0.10%	98.8%	0.10%	98.0%
10/19	0.00%	94.9%	0.03%	71.1%	0.09%	97.7%	0.31%	94.8%
11/19	0.00%	95.8%	0.02%	81.0%	0.05%	98.1%	0.40%	96.2%
12/19	0.01%	96.6%	0.06%	84.9%	0.24%	98.3%	0.45%	96.8%
01/20	0.18%	93.7%	0.12%	78.0%	2.14%	96.3%	2.27%	95.4%
02/20	0.07%	93.4%	0.33%	68.3%	4.82%	95.7%	6.68%	93.2%
03/20	0.01%	95.8%	0.01%	75.3%	0.13%	98.1%	0.35%	96.0%
04/20	0.00%	97.0%	0.02%	80.8%	0.14%	98.9%	0.26%	97.3%
05/20	0.00%	97.5%	0.05%	85.7%	0.13%	98.6%	0.29%	96.0%
06/20	0.01%	97.8%	0.04%	83.2%	0.22%	98.9%	0.43%	96.7%
07/20	0.01%	96.4%	0.03%	66.2%	0.07%	98.7%	0.33%	93.9%
08/20	0.01%	92.9%	0.02%	47.2%	0.06%	96.0%	0.10%	85.9%
09/20	0.02%	92.1%	0.03%	56.0%	0.08%	95.7%	0.13%	82.9%

Impact of Concept Drift

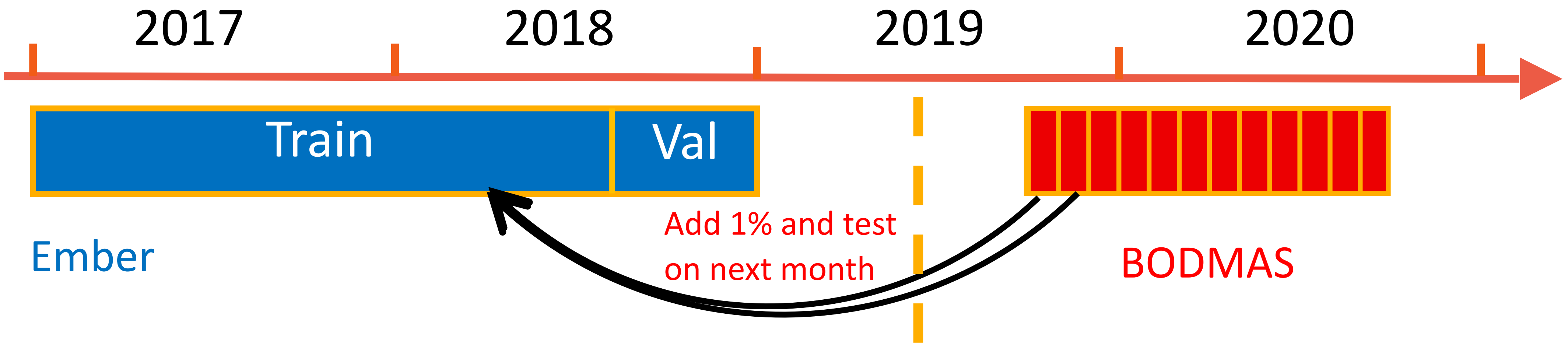
Phase	Ember-GBDT		UCSB-GBDT		SOREL-GBDT		SOREL-DNN	
	FPR	F_1	FPR	F_1	FPR	F_1	FPR	F_1
Val	0.10%	98.6%	0.10%	92.1%	0.10%	98.8%	0.10%	98.0%
10/19	0.00%	94.9%	0.03%	71.1%	0.09%	97.7%	0.31%	94.8%
11/19	0.00%	95.8%	0.02%	81.0%	0.05%	98.1%	0.40%	96.2%
12/19	0.01%	96.6%	0.06%	84.9%	0.24%	98.3%	0.45%	96.8%
01/20	0.18%	93.7%	0.12%	78.0%	2.14%	96.3%	2.27%	95.4%
02/20	0.07%	93.4%	0.33%	68.3%	4.82%	95.7%	6.68%	93.2%
03/20	0.01%	95.8%	0.01%	75.3%	0.13%	98.1%	0.35%	96.0%
04/20	0.00%	97.0%	0.02%	80.8%	0.14%	98.9%	0.26%	97.3%
05/20	0.00%	97.5%	0.05%	85.7%	0.13%	98.6%	0.29%	96.0%
06/20	0.01%	97.8%	0.04%	83.2%	0.22%	98.9%	0.43%	96.7%

1. Most classifiers got 98% F_1 on validation; but degraded (sometimes a lot) on BODMAS.
2. Concept drift could be discrete events instead of a monotonic trend over time.

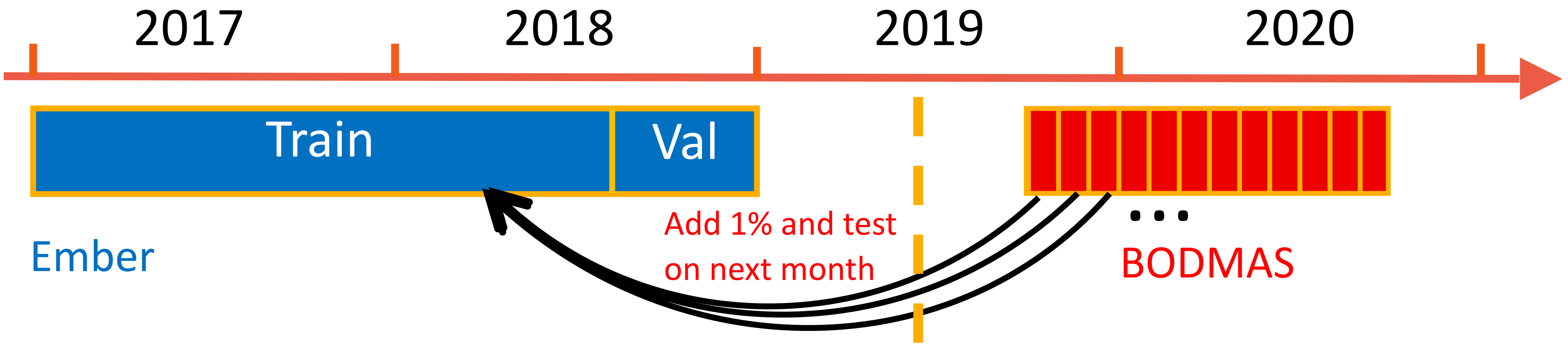
Mitigation Strategy 1: Incremental Retraining



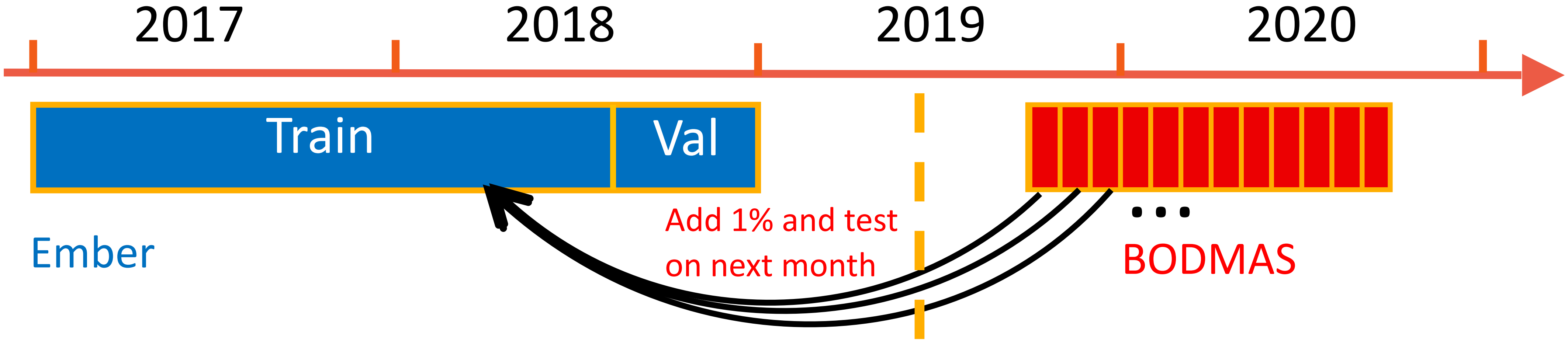
Mitigation Strategy 1: Incremental Retraining

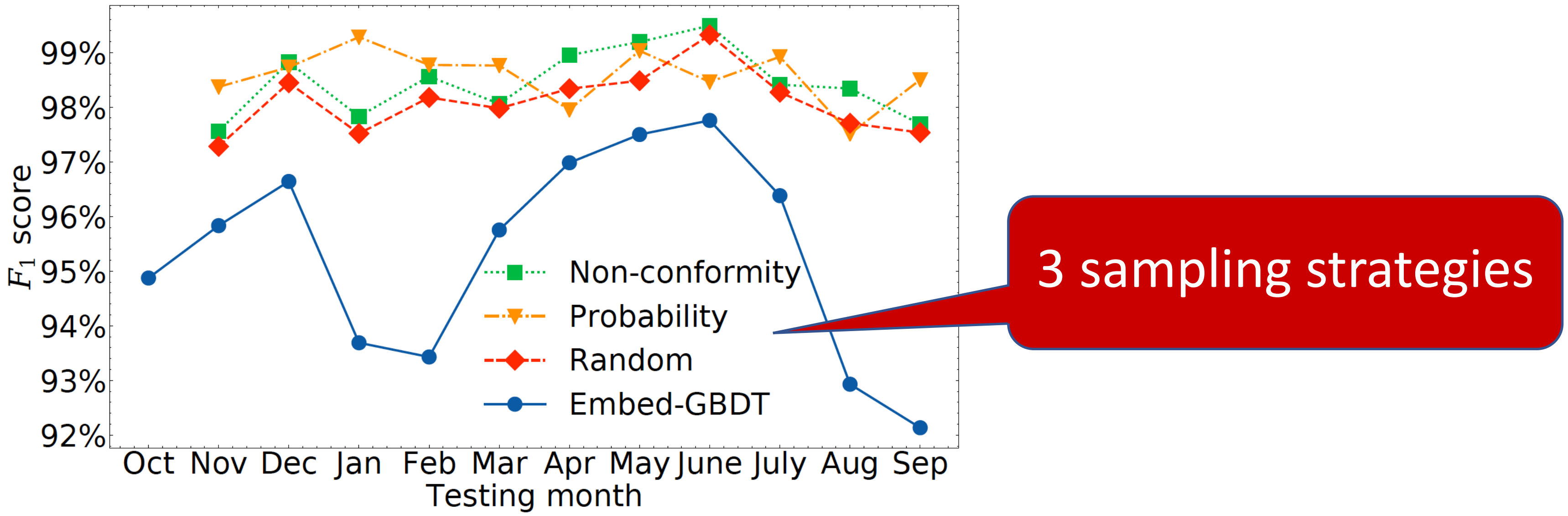


Mitigation Strategy 1: Incremental Retraining

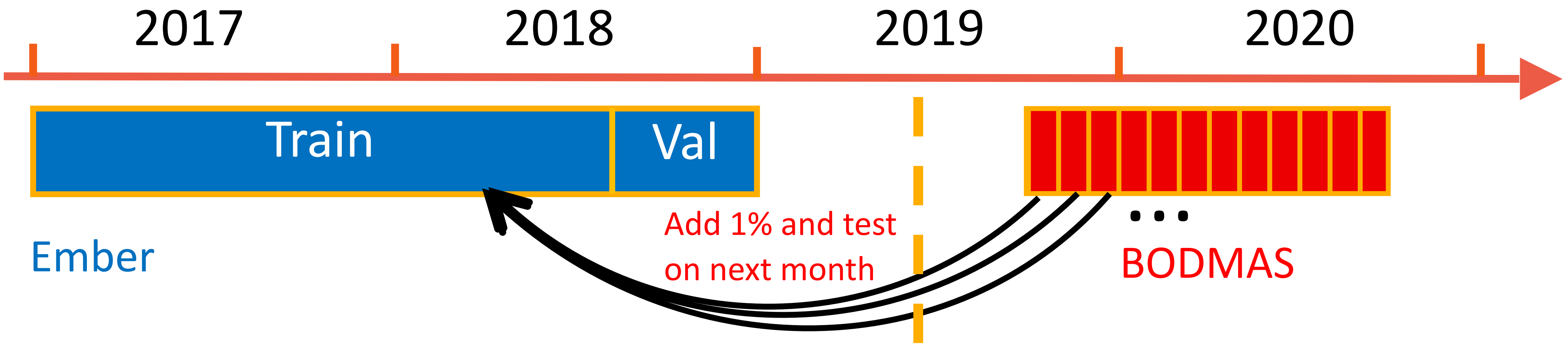


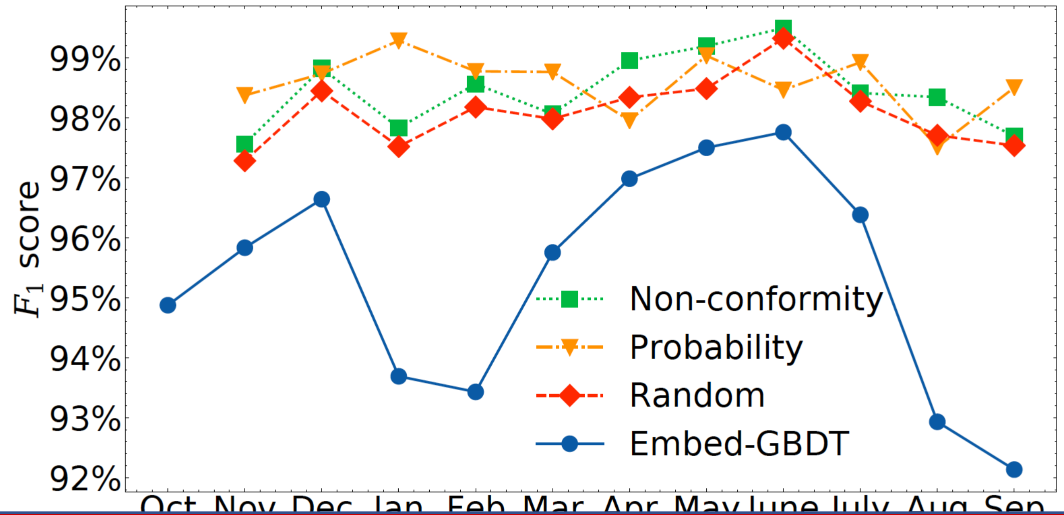
Mitigation Strategy 1: Incremental Retraining





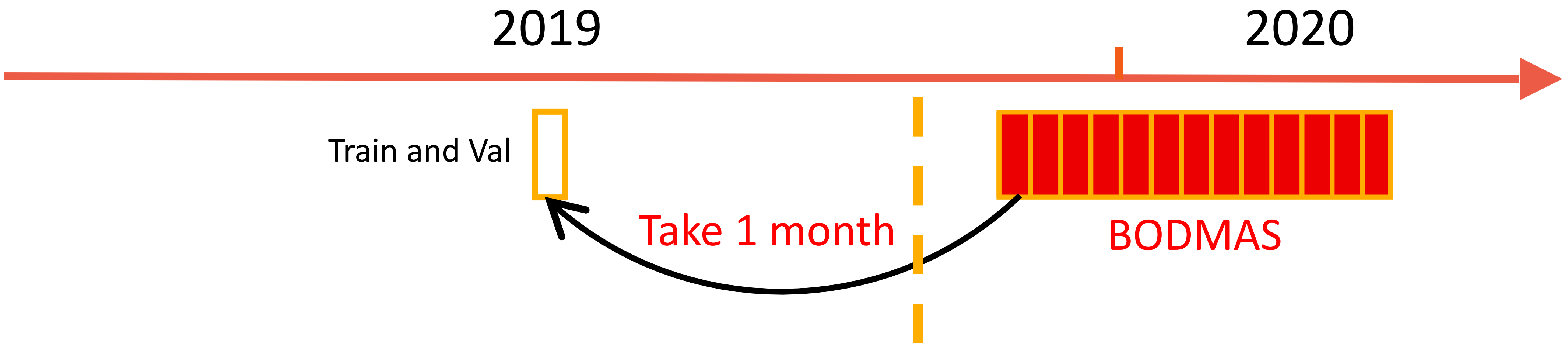
Mitigation Strategy 1: Incremental Retraining



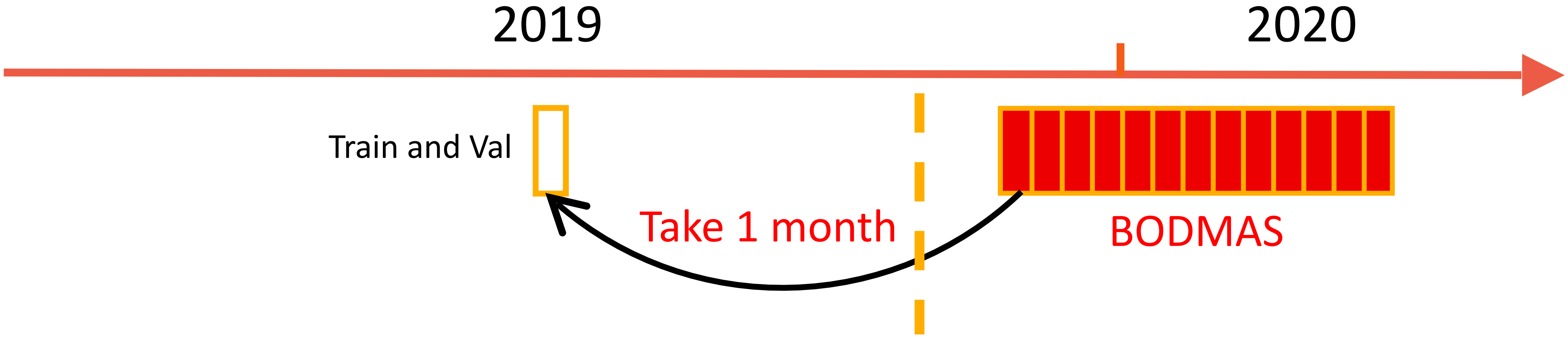


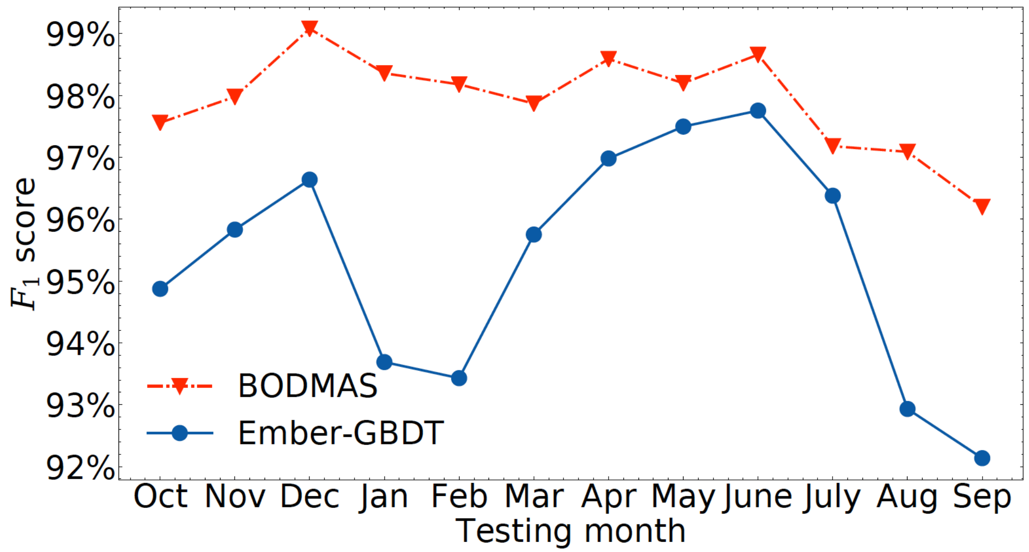
1. Labeling 1% samples per month, all the F_1 scores surpass 97%.
2. Different sampling methods have close performance.

Mitigation Strategy 2: Train with New Data

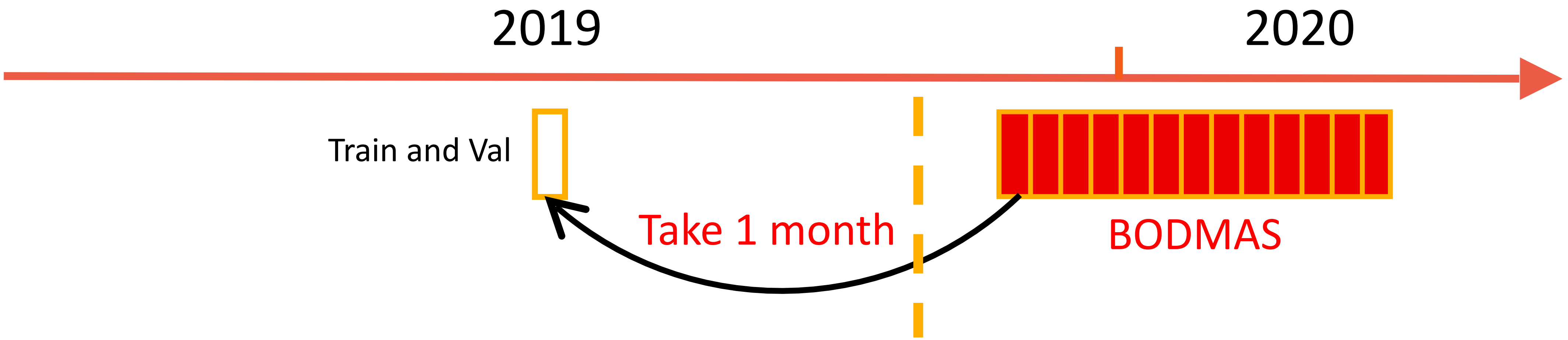


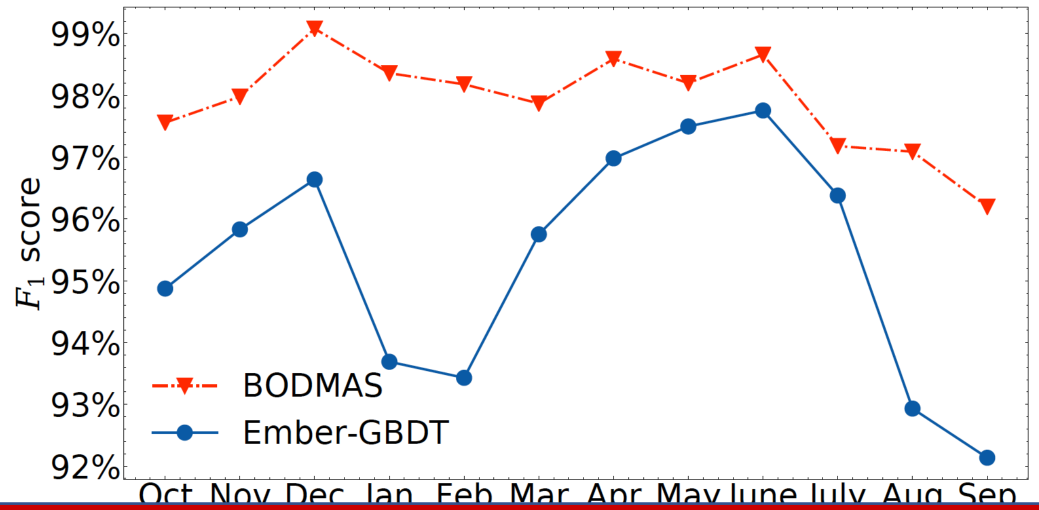
Mitigation Strategy 2: Train with New Data





Mitigation Strategy 2: Train with New Data





1. Labeling new data and train a new classifier indeed improve the F_1 score.
2. A slight downward trend still exists, indicating the impact of concept drift.

Breakdown of False Negatives

Testing month	FNR	Existing Family FNR	Unseen Family FNR
10/19	4.8%	3.4%	43.0%
11/19	4.0%	2.7%	35.4%
12/19	1.7%	1.4%	16.7%
01/20	3.0%	2.2%	27.0%
02/20	3.1%	2.4%	26.2%
03/20	4.2%	3.6%	20.0%
04/20	2.7%	2.5%	8.1%
05/20	3.5%	2.7%	9.4%
06/20	2.6%	2.3%	6.3%
07/20	5.5%	5.2%	6.8%
08/20	5.7%	4.8%	15.6%
09/20	7.2%	5.8%	16.4%

Breakdown of False Negatives

Testing month	FNR	Existing Family FNR	Unseen Family FNR
10/19	4.8%	3.4%	43.0%
11/19	4.0%	2.7%	35.4%
12/19	1.7%	1.4%	16.7%
01/20	3.0%	2.2%	27.0%
02/20	3.1%	2.4%	26.2%
03/20	4.2%	3.6%	20.0%
04/20	2.7%	2.5%	8.1%
05/20	3.5%	2.7%	9.4%
06/20	2.6%	2.3%	6.3%
07/20	5.5%	5.2%	6.8%
08/20	5.7%	4.8%	15.6%
09/20	7.2%	5.8%	16.4%

Breakdown of False Negatives

Testing month	FNR	Existing Family FNR	Unseen Family FNR
10/19	4.8%	3.4%	43.0%
11/19	4.0%	2.7%	35.4%
12/19	1.7%	1.4%	16.7%
01/20	3.0%	2.2%	27.0%
02/20	3.1%	2.4%	26.2%
03/20	4.2%	3.6%	20.0%
04/20	2.7%	2.5%	8.1%
05/20	3.5%	2.7%	9.4%
06/20	2.6%	2.3%	6.3%
07/20	5.5%	5.2%	6.8%
08/20	5.7%	4.8%	15.6%

1. Existing families indeed produce false negatives, e.g., malware variants.
2. Unseen families are more likely to be misclassified than existing families.

Outline

- ~~Introduction~~
- ~~Open problem: concept drift in binary classifiers across time~~
- Open problem: concept drift in malware family attribution

A Multi-class Malware Classification Model

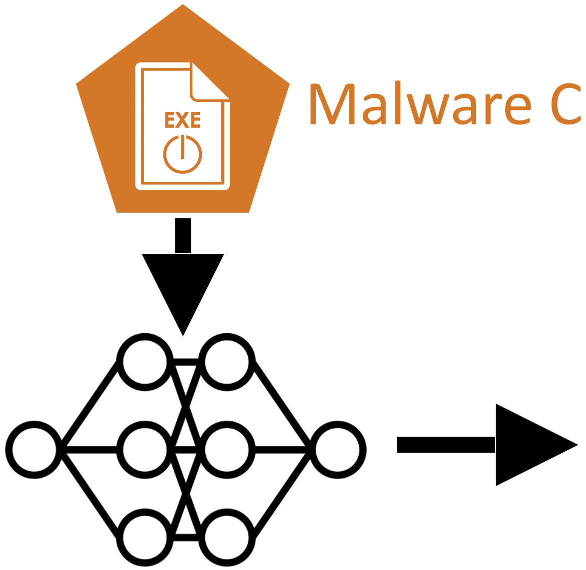
1. Train

Benign

Malware A

Malware B

2. Predict



A Multi-class Malware Classification Model

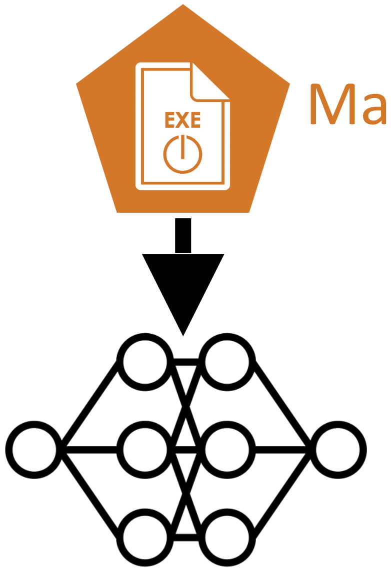
1. Train

Benign

Malware A

Malware B

2. Predict

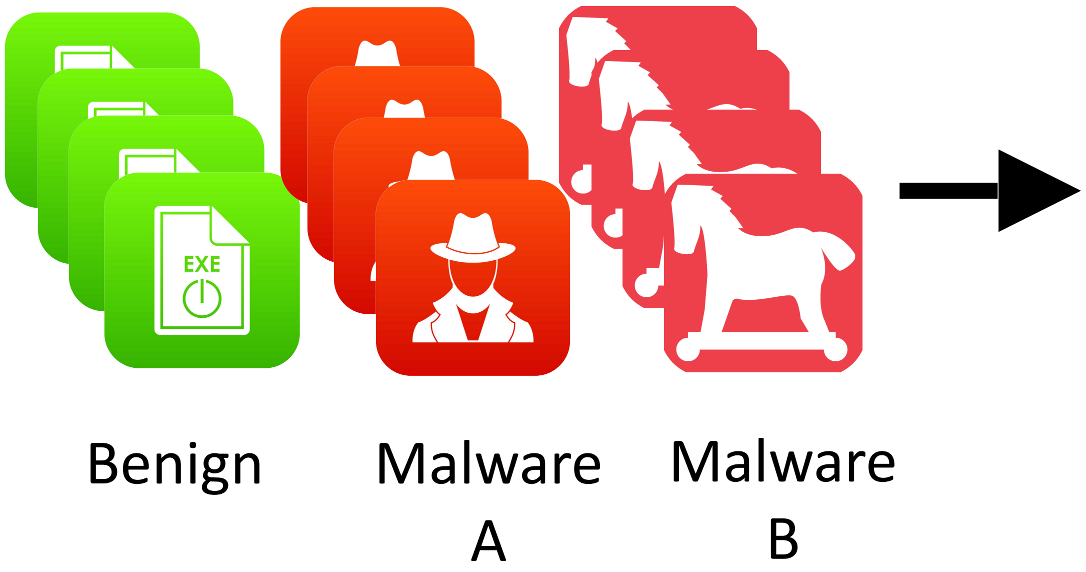


Malware C

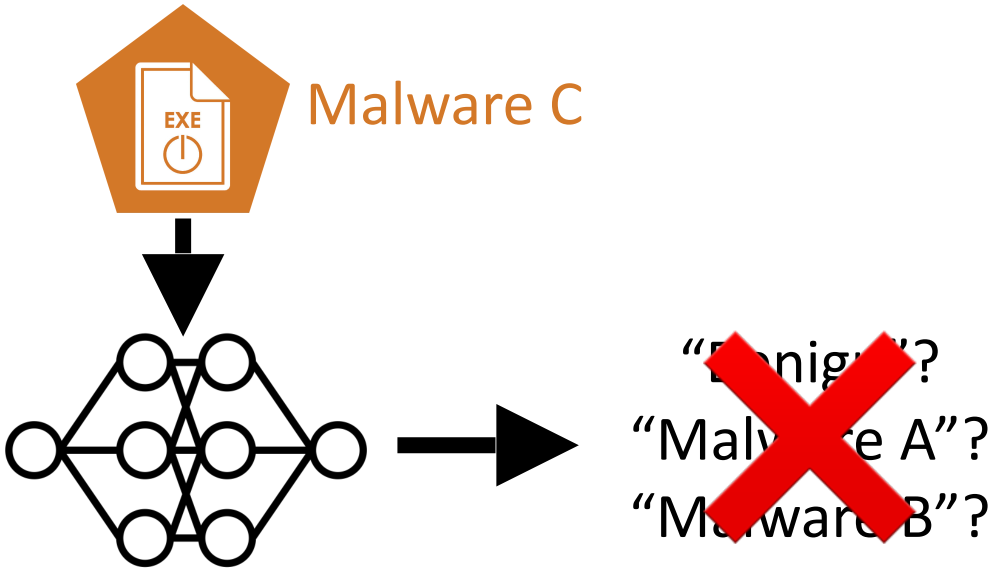
“Benign”?
“Malware A”?
“Malware B”?

A Multi-class Malware Classification Model

1. Train



2. Predict



Concept Drift!
(Unseen family)

Close-world VS. Open-world

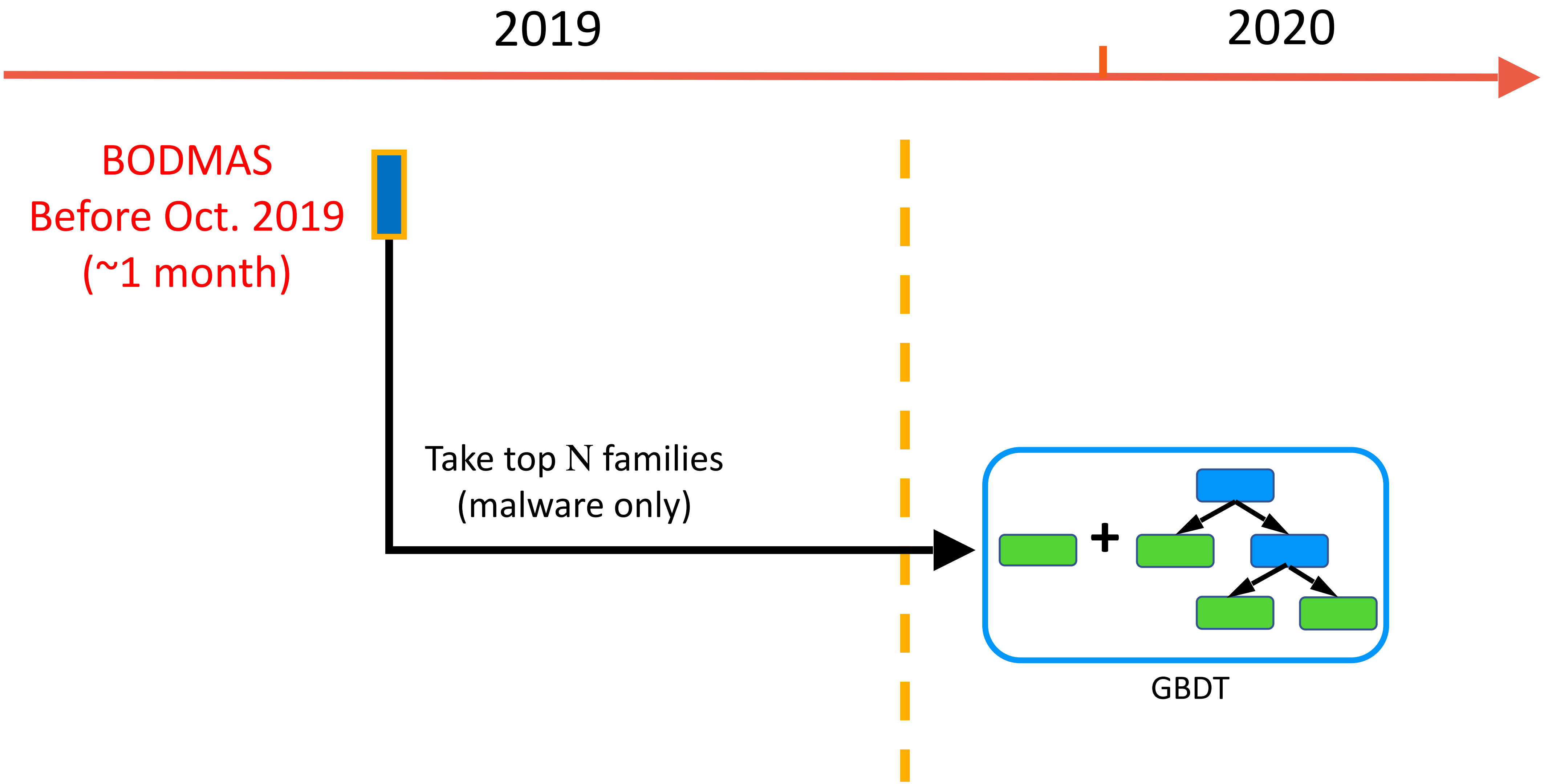
- Close-world
  - Both training and testing sets contain N families
- Open-world
  - N is large and increases over time
  - Malware from previously unseen families

FireEye Annual Report 2020

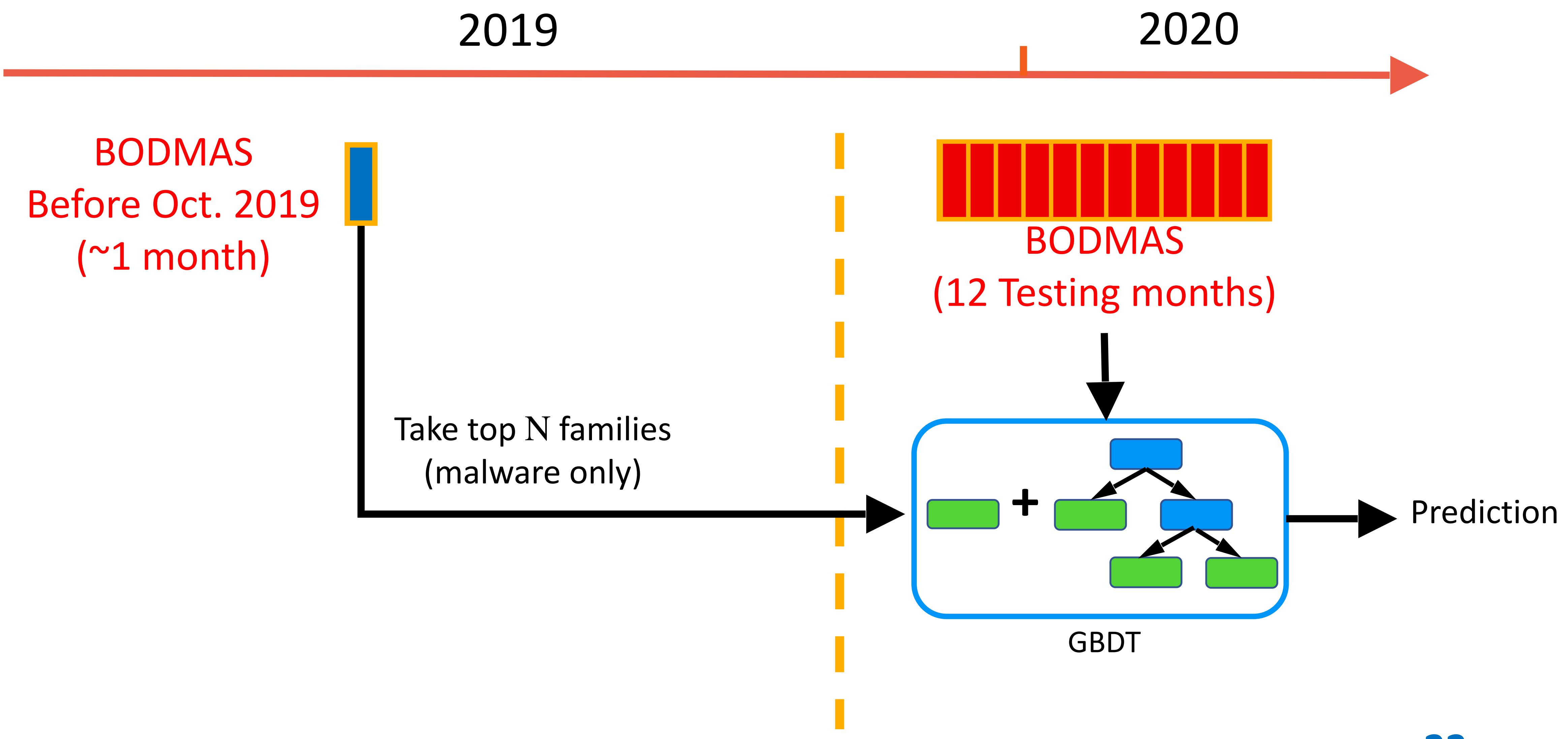
"1.1 million malware samples per day"

"41% malware families never seen before"

Experiment: Concept Drift in Family Attribution

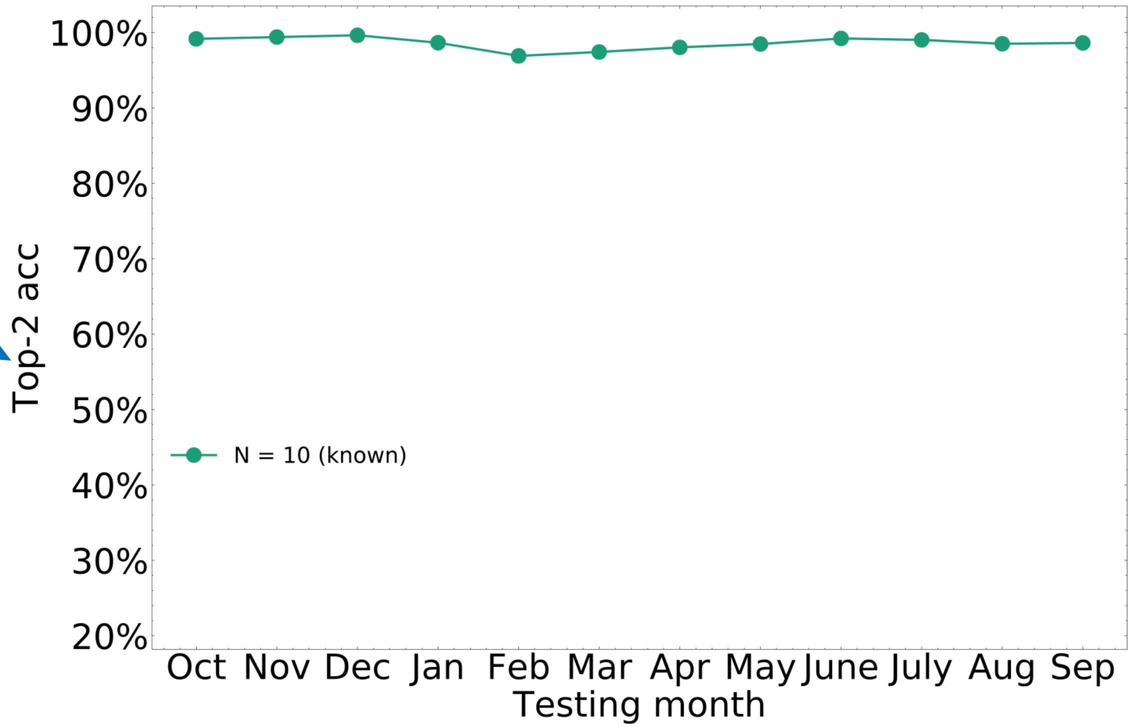


Experiment: Concept Drift in Family Attribution



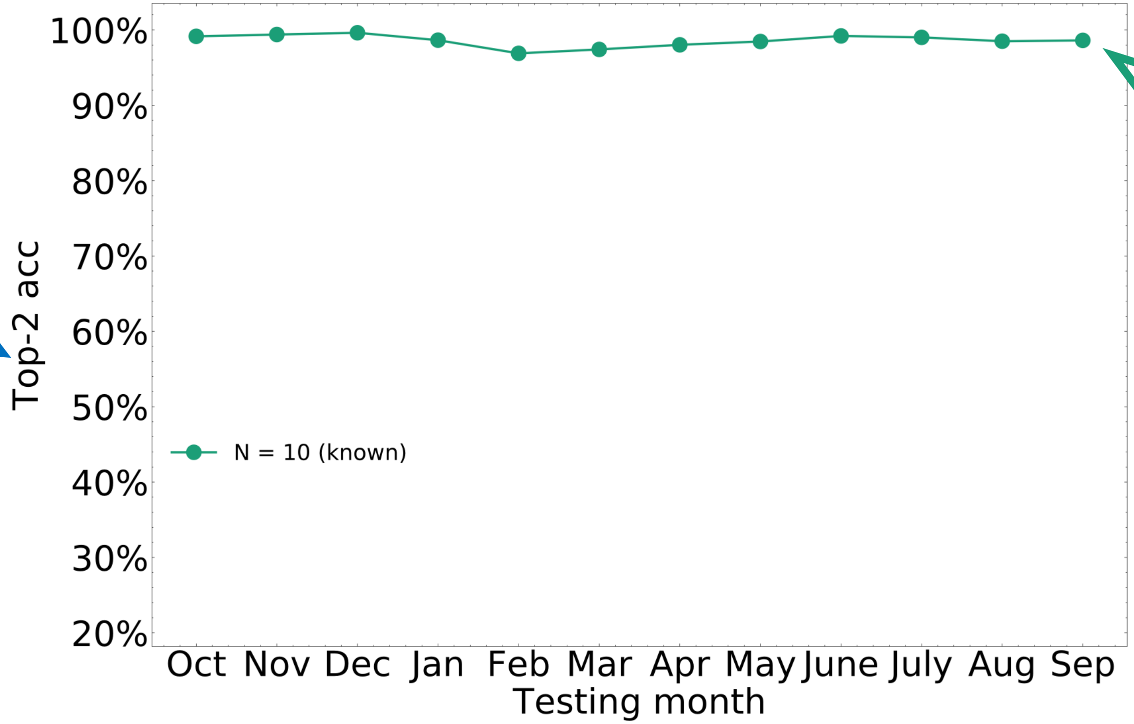
Impact of Concept Drift

Top-2 acc:
Likelihood that top-2 predicted families contain a sample's true family



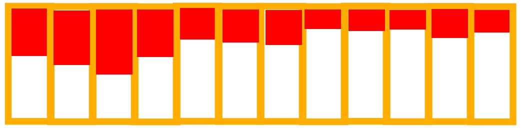
Impact of Concept Drift

Top-2 acc:
Likelihood that top-2 predicted families contain a sample's true family

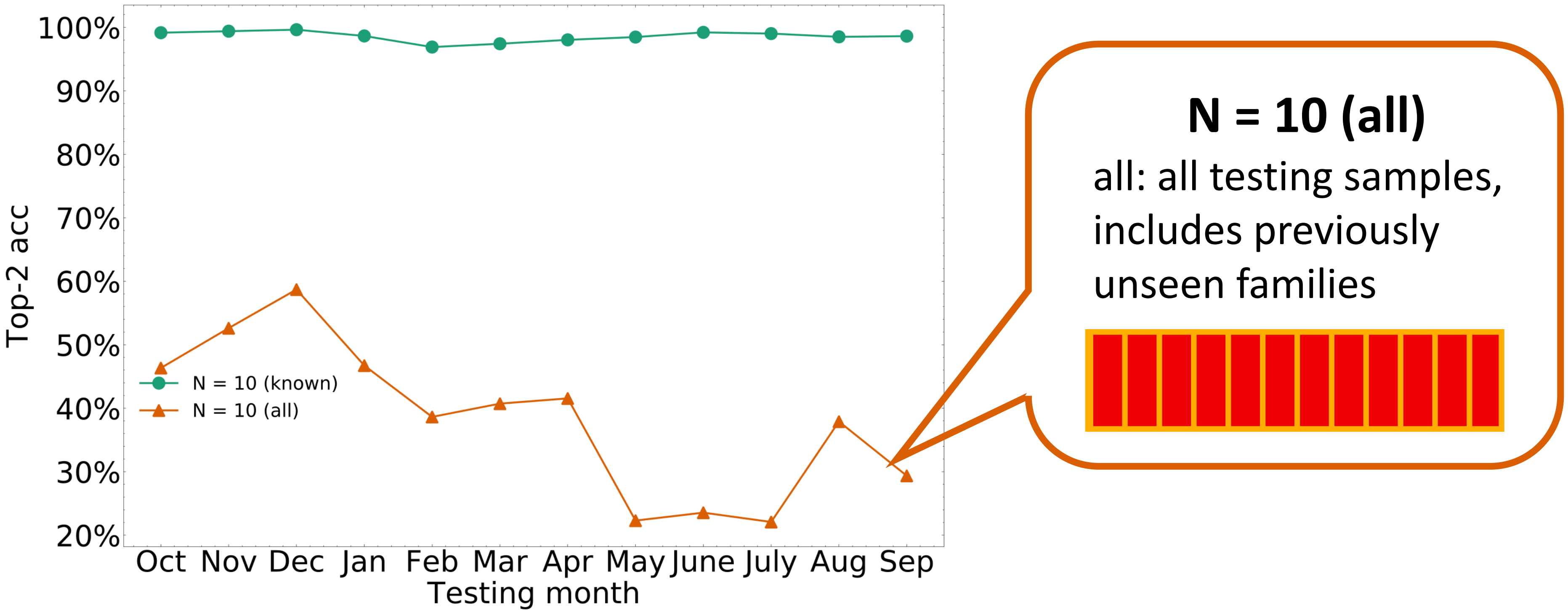


N = 10 (known)

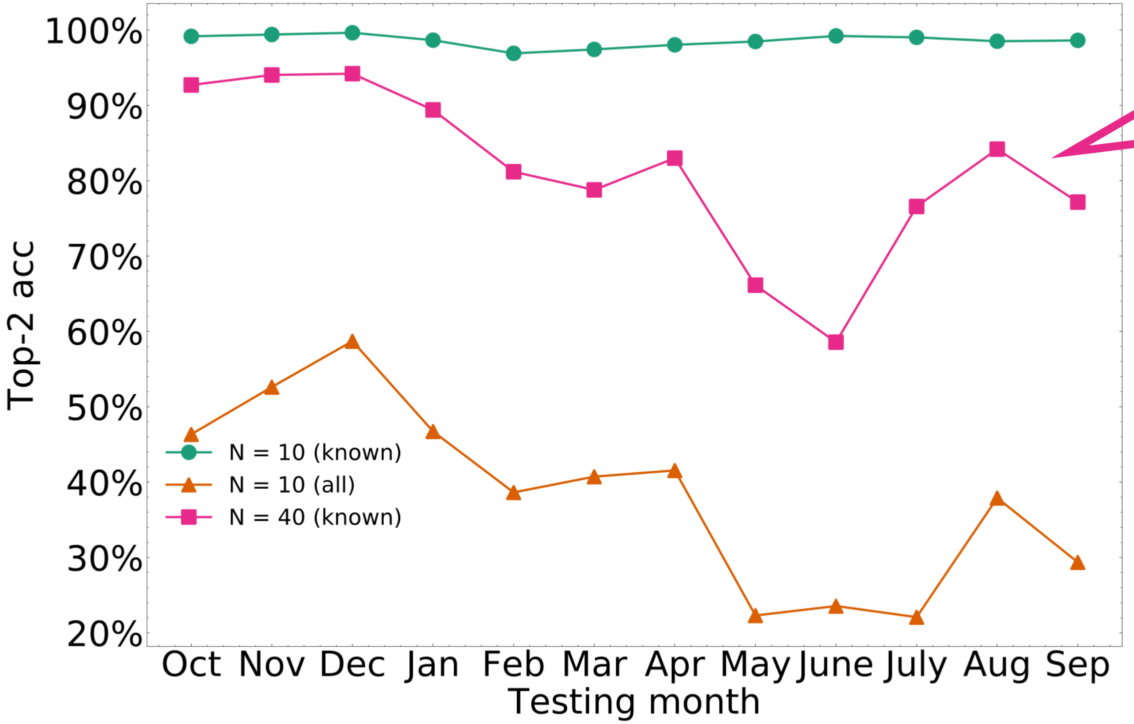
Known: testing set only includes samples from 10 known training families



Impact of Concept Drift



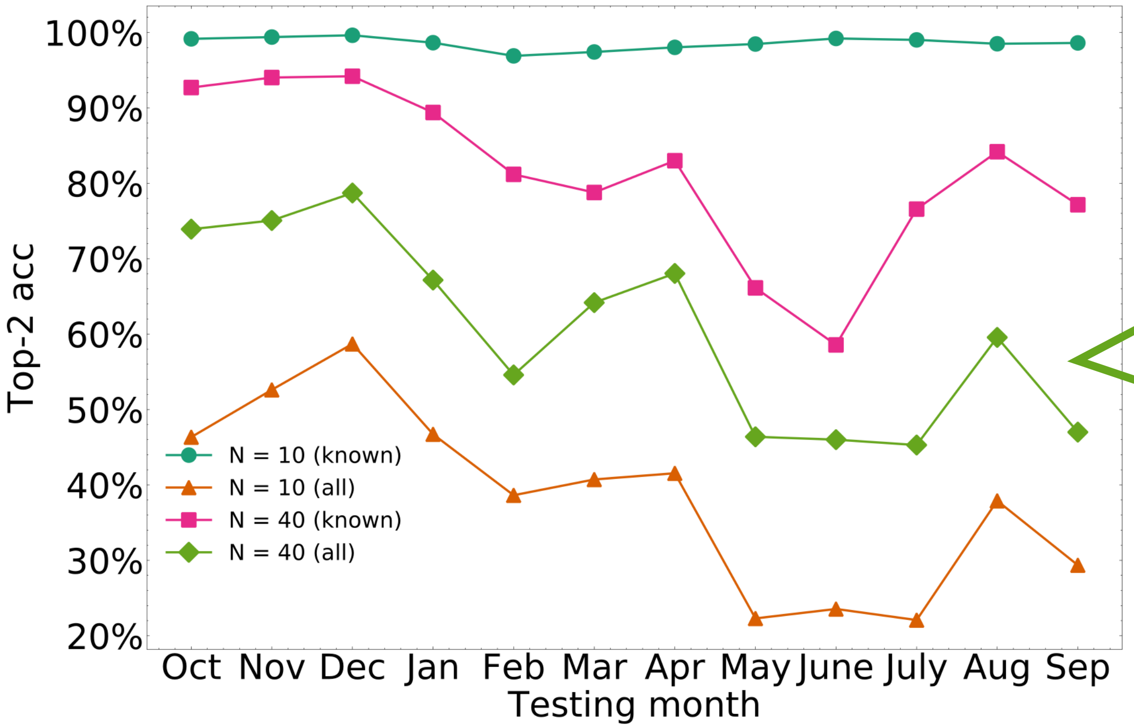
Impact of Concept Drift



N = 40 (known)

Larger number of families is harder to train

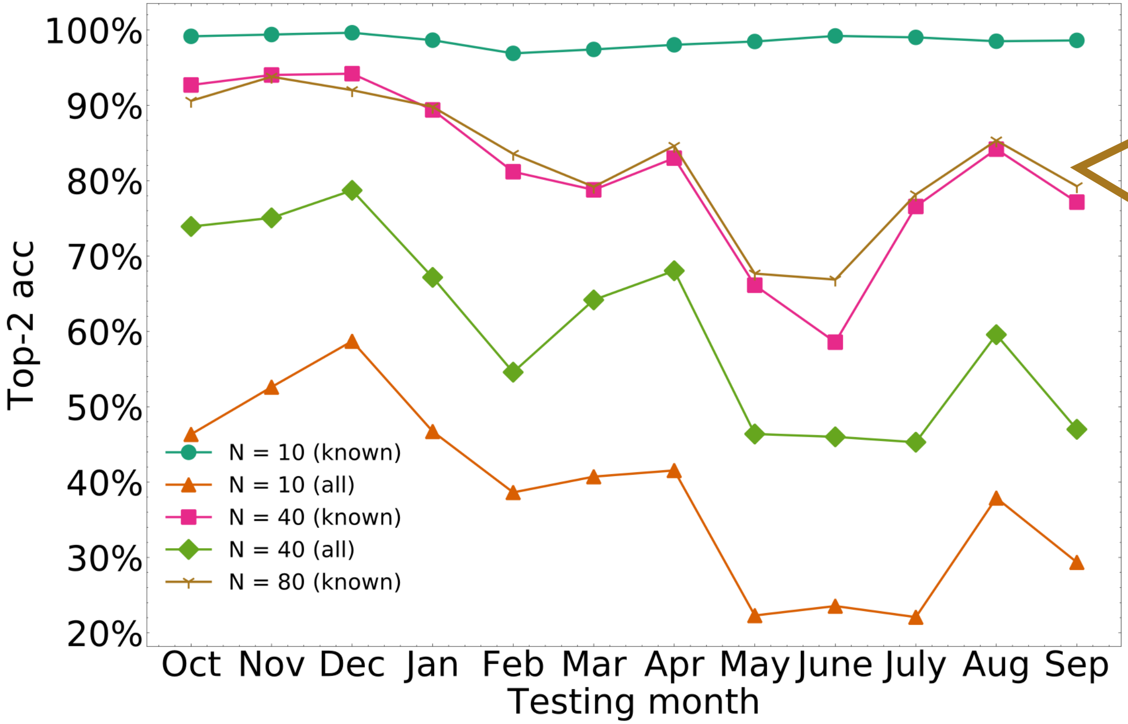
Impact of Concept Drift



N = 40 (all)

Larger N means we have fewer unseen families during testing, thus better than N = 10 (all)

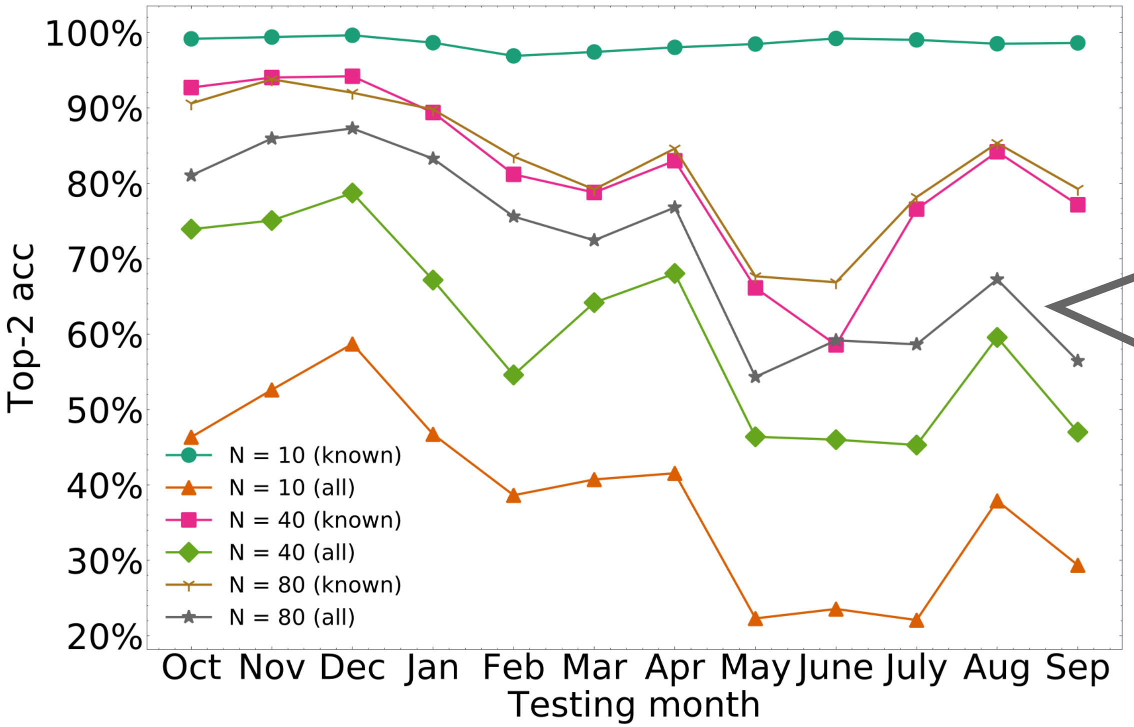
Impact of Concept Drift



N = 80 (known)

Further increasing N does not give a worse performance because later families do not have many samples

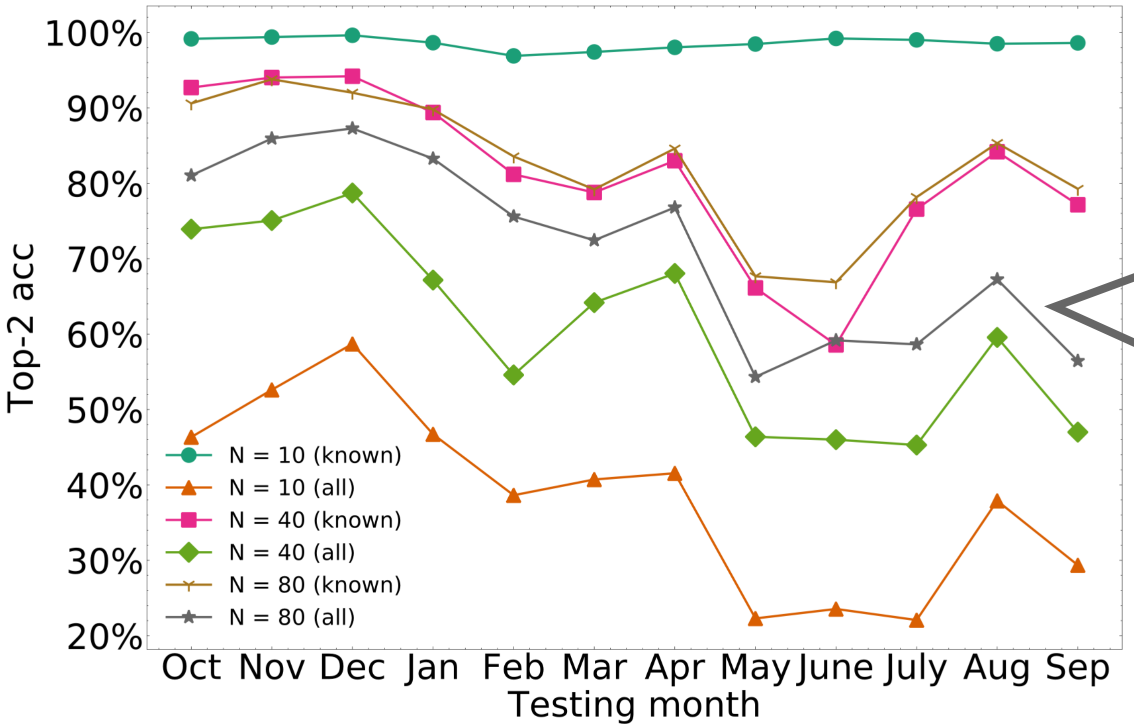
Impact of Concept Drift



N = 80 (all)

Fewer unseen families during testing making it better than N = 40 (all)

Impact of Concept Drift



N = 80 (all)

Fewer unseen families during testing making it better than N = 40 (all)

1. Unseen families significantly degrade the performance of a close-world classifier.
2. It becomes harder to train a decent classifier when N increases.

Open Problems and Challenges

- Out-of-distribution detection against malware evolution and unseen family
- Scale to large number of malware families and relationships among families
- Combat real-world adversarial samples of malware binaries

Conclusion

- We release a new PE malware dataset with timestamp and malware families
- Concept drift poses challenges for both malware detection and attribution
- Unseen families are more likely to be misclassified than known families

Thank you!

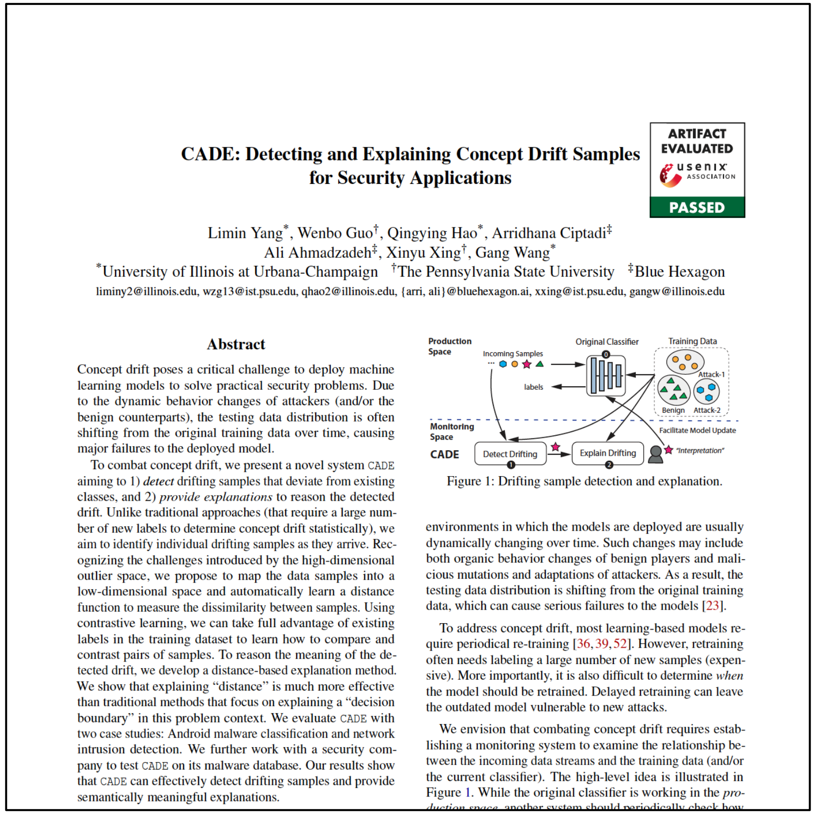
Homepage

https://liminyang.web.Illinois.edu

Features and metadata open to public

Malware binaries available upon request

https://whyisyoung.github.io/BODMAS/



CADE: Detecting and Explaining Concept Drift Samples for Security Applications

ARTIFACT EVALUATED usenix ASSOCIATION PASSED

Limin Yang*, Wenbo Guo†, Qingying Hao*, Arridhana Ciptadi‡
Ali Ahmadzadeh‡, Xinyu Xing†, Gang Wang*
*University of Illinois at Urbana-Champaign †The Pennsylvania State University ‡Blue Hexagon
liminy2@illinois.edu, wzg13@ist.psu.edu, qhao2@illinois.edu, {arri, ali}@bluehexagon.ai, xxing@ist.psu.edu, gangw@illinois.edu

Abstract

Concept drift poses a critical challenge to deploy machine learning models to solve practical security problems. Due to the dynamic behavior changes of attackers (and/or the benign counterparts), the testing data distribution is often shifting from the original training data over time, causing major failures to the deployed model.

To combat concept drift, we present a novel system CADE aiming to 1) *detect* drifting samples that deviate from existing classes, and 2) *provide explanations* to reason the detected drift. Unlike traditional approaches (that require a large number of new labels to determine concept drift statistically), we aim to identify individual drifting samples as they arrive. Recognizing the challenges introduced by the high-dimensional outlier space, we propose to map the data samples into a low-dimensional space and automatically learn a distance function to measure the dissimilarity between samples. Using contrastive learning, we can take full advantage of existing labels in the training dataset to learn how to compare and contrast pairs of samples. To reason the meaning of the detected drift, we develop a distance-based explanation method. We show that explaining "distance" is much more effective than traditional methods that focus on explaining a "decision boundary" in this problem context. We evaluate CADE with two case studies: Android malware classification and network intrusion detection. We further work with a security company to test CADE on its malware database. Our results show that CADE can effectively detect drifting samples and provide semantically meaningful explanations.

Figure 1: Drifting sample detection and explanation.

environments in which the models are deployed are usually dynamically changing over time. Such changes may include both organic behavior changes of benign players and malicious mutations and adaptations of attackers. As a result, the testing data distribution is shifting from the original training data, which can cause serious failures to the models [23].

To address concept drift, most learning-based models require periodical re-training [36, 39, 52]. However, retraining often needs labeling a large number of new samples (expensive). More importantly, it is also difficult to determine *when* the model should be retrained. Delayed retraining can leave the outdated model vulnerable to new attacks.

We envision that combating concept drift requires establishing a monitoring system to examine the relationship between the incoming data streams and the training data (and/or the current classifier). The high-level idea is illustrated in Figure 1. While the original classifier is working in the *production space*, another system should periodically check how

Check out our upcoming USENIX Sec'21 paper

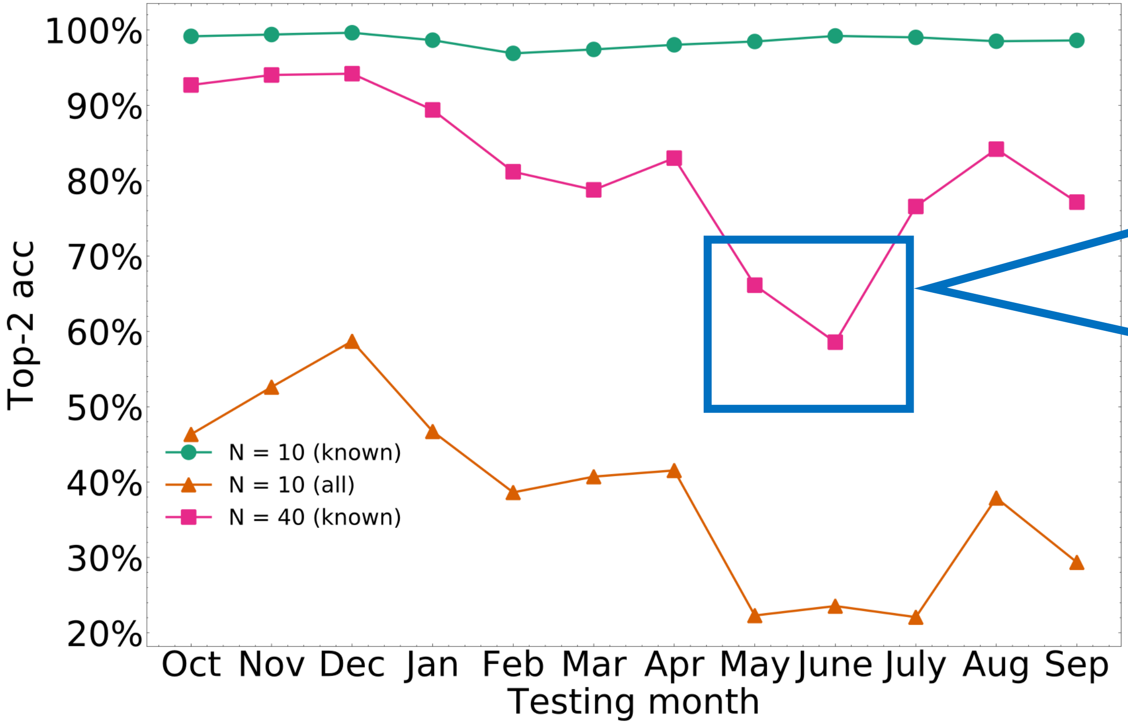
- CADE: Detecting and Explaining Concept Drift Samples for Security Applications

References

- **[arXiv’18]** Ember: An Open Dataset for Training Static PE Malware Machine Learning Models
  Hyrum S. Anderson and Phil Roth
  arXiv preprint arXiv:1804.04637, 2018
- **[arXiv’18]** Microsoft Malware Classification Challenge
  Royi Ronen, Marian Radu, Corina Feuerstein, Elad Yom-Tov, and Mansour Ahmadi
  arXiv preprint arXiv:1802.10135, 2018.
- **[NDSS’20]** When Malware is Packin’ Heat; Limits of Machine Learning Classifiers Based on Static Analysis Features
  Hojjat Aghakhani, Fabio Gritti, Francesco Mecca, Martina Lindorfer, Stefano Ortolani, Davide Balzarotti, Giovanni Vigna, and Christopher Kruegel. Proceedings of Network and Distributed Systems Security (NDSS) Symposium, February 2020.
- **[arXiv’20]** SOREL-20M: A Large Scale Benchmark Dataset for Malicious PE Detection
  Richard Huang and Ethan M. Rudd
  arXiv preprint arXiv:2012.07634, 2020.
- **[USENIX Sec’17]** Transcend: Detecting Concept Drift in Malware Classification Models
  Roberto Jordaney, Kumar Sharad, Santanu K. Dash, Zhi Wang, Davide Papini, Ilia Nouretdinov, and Lorenzo Cavallaro.
  Proceedings of The 26th USENIX Security Symposium (USENIX Security), August 2017.
- **[USENIX Sec’21]** CADE: Detecting and Explaining Concept Drift Samples for Security Applications
  Limin Yang, Wenbo Guo, Qingying Hao, Arridhana Ciptadi, Ali Ahmadzadeh, Xinyu Xing, and Gang Wang
  Proceedings of The 30th USENIX Security Symposium (USENIX Security), August 2021.

Backup Slides

Reasons of the Drop



A family called "sfone" is under-trained, only 52 samples in training. However, we saw a burst arrival of "sfone" in May and June (2,491 samples)