Approaches to Adversarial Drift

Alex Kantchelian, Sadia Afroz, Ling Huang, Aylin Caliskan Islam, Brad Miller, Michael Carl Tschantz, Rachel Greenstadt, Anthony D. Joseph & J. D. Tygar

Elham Baqazi

CISC850
Cyber Analytics

Outline

- Challenges of applying ML systems for security applications
- Exploratory & Causative attack
- Families Isolation & Responsiveness
- Data Exploration

Adversarial Drift

- Designing changes to evade the classifier immediately or to make future evasion easier
- Handling the adversarial drift

Machine learning in Security Application

- One-Shot Approach
  - Training data
  - Building the model
  - Testing data

Problem Statement

- Security Apps data: Big & non-stationary data, drift over the time
- The typical ML approach fail

Proposed Solution

- Designing adaptive, adversarial-resistant ML systems
    - Ensemble of classifiers
    - Responsive classifier

Formalism

- Retraining the system to learn from new instances
  - Producing a series of models H_t
  - $H_t(x_i) = c(x_i)$ [correctly classifies]

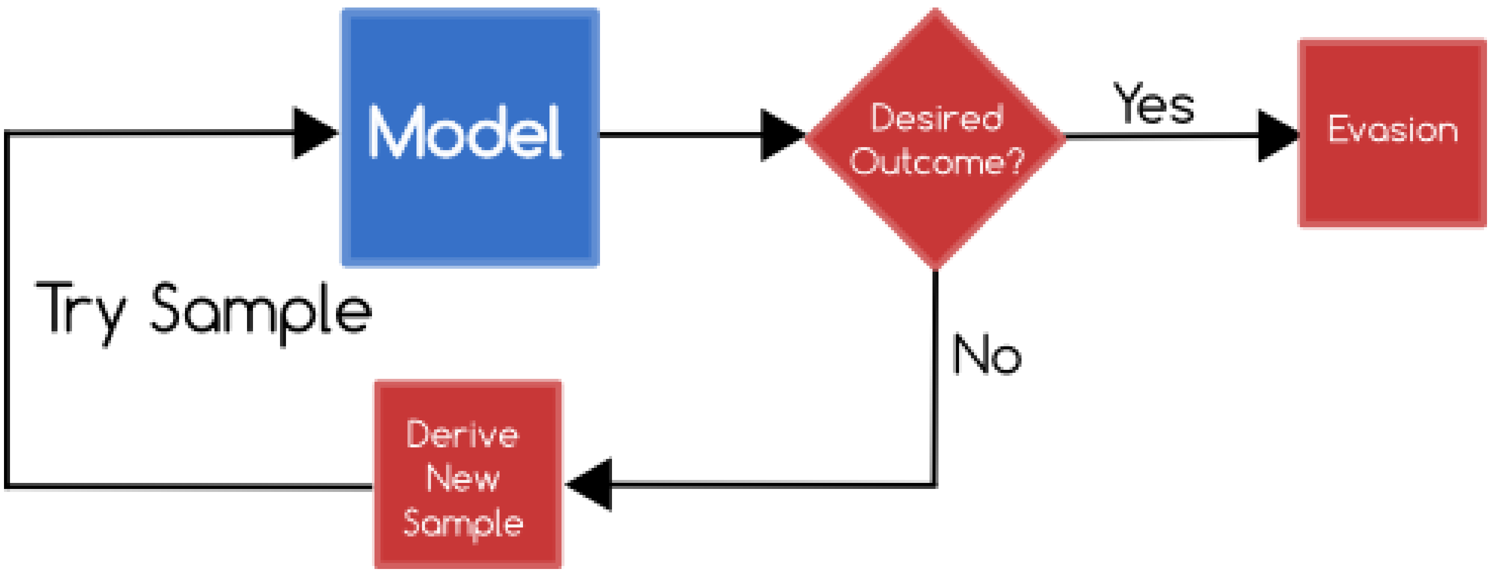
Population Drift

- X_t (x) is the probability of encountering instance “x” at time t
- Adversaries post new malware X_{t+1}
- Population Drift → $X_t \mathrel{!}= X_{t'}$

Types of Attacks

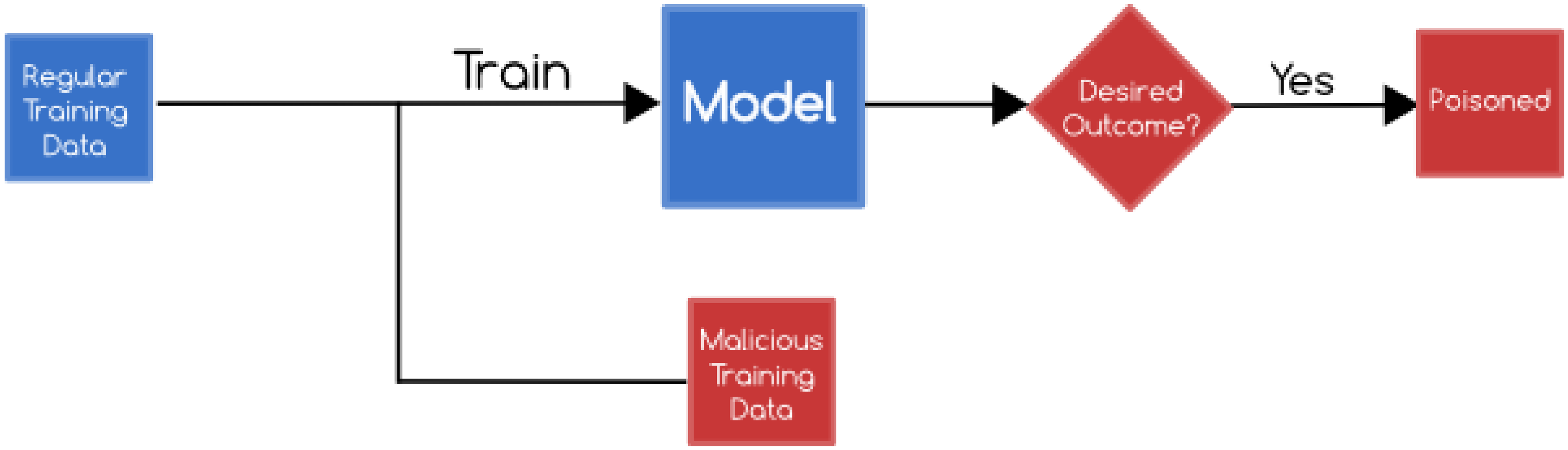
- Exploratory attacks
- Causative attacks

Exploratory Attacks



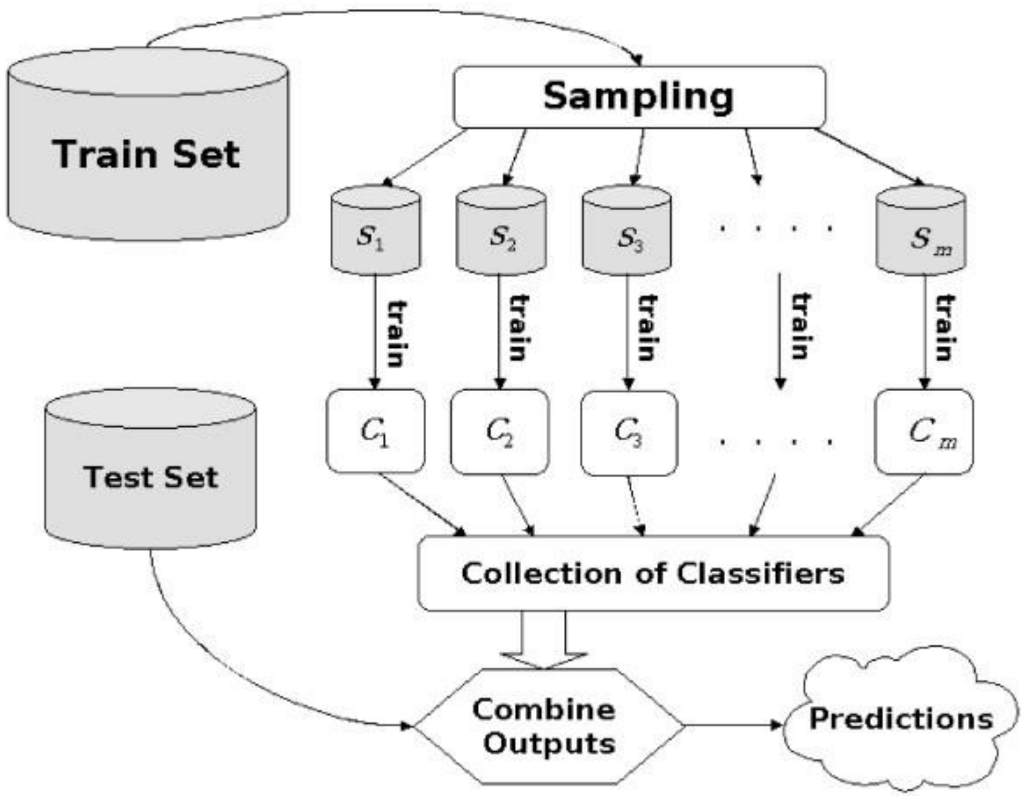
https://mascherari.press/introduction-to-adversarial-machine-learning/

Causative Attacks



https://mascherari.press/introduction-to-adversarial-machine-learning/

Families and Isolation



https://www.researchgate.net/figure/5850993_fig7_Architecture-of-the-ensemble-of-Support-Vector-Machine-classifiers-A-collection-of-m-SVM

Families and Isolation

- Training classifiers
  - *One-vs-all* method
  - *One-vs-good* method
  - Isolation
- Combining classification

Responsiveness

- Why it being overlooked?
  - Zero training error , poor generalization
  - Unreliable training data.
- Wrapped ML algorithm
  - Blacklist & Whitelist

Evaluation

- Executable malware dataset with chronological appearance for each instance.
- Demonstrating the importance of temporal drift in a very adversarial environment.
- Improving the robustness of ML algorithms.

Data Exploration - Dataset

- Sampled from two stratums :
  - TimeStamp, Label , Feature vector

	Old: Apr '07-Mar '13	New: Apr '13- Jul '13
Benign	85549	8803
Malware	40861	82984
Total	**126410**	**91787**

Top 10 Families

Family	# of instances	Duration
worm:win32/vobfus	14203	10/2008 - 06/2013
trojandownloader:win32/beebone	11125	03/2012 - 06/2013
pws:win32/zbot	5691	01/2008 - 06/2013
adware:win32/hotbar	3913	09/2010 - 07/2013
virus:win32/ramnit	2387	11/2010 - 06/2013
trojan:win32/ramnit	2078	12/2010 - 06/2013
rogue:win32/winwebsec	2022	05/2009 - 06/2013
trojan:win32/killav	1917	11/2007 - 06/2013
trojan:win32/vundo	1601	11/2007 - 06/2013
worm:win32/allaple	1567	05/2007 - 06/2013

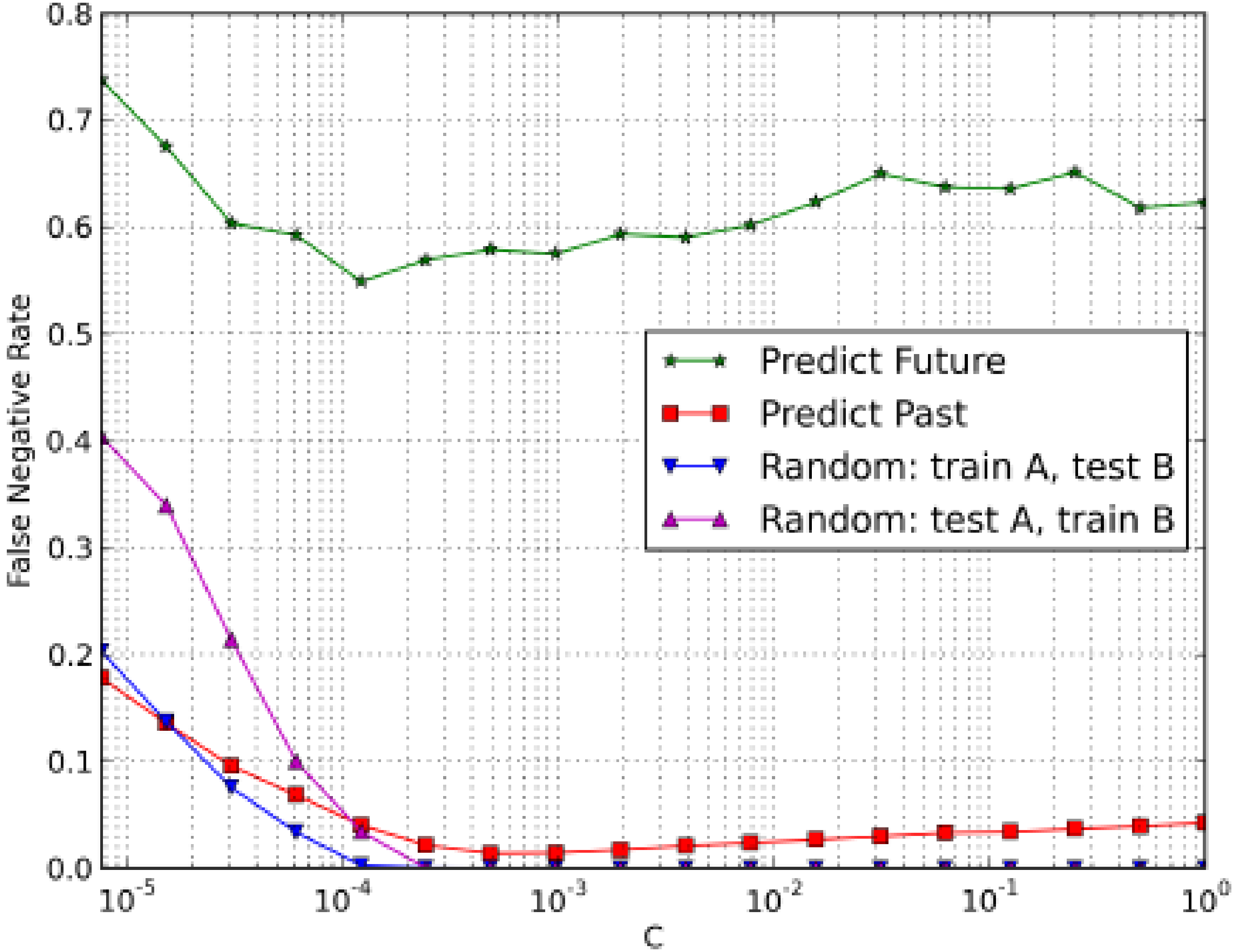
Experiments – Approach

- An empirical loss minimization approach

$$\mathbf{w} \mapsto \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \sum_{(\mathbf{x},y)\in\mathcal{D}} \max(0, 1 - y\mathbf{x}^T\mathbf{w})^2$$

Data Exploration – Experiments 1

- Splitting the dataset into two epochs [mid-April], 60,000 malware in each period
- Train two-class SVM models
  - Regularization factor: $10^{-5} < C < 1$
  - False Positive Rate (FPR) < 1%
- Calculating the Performance by two ways

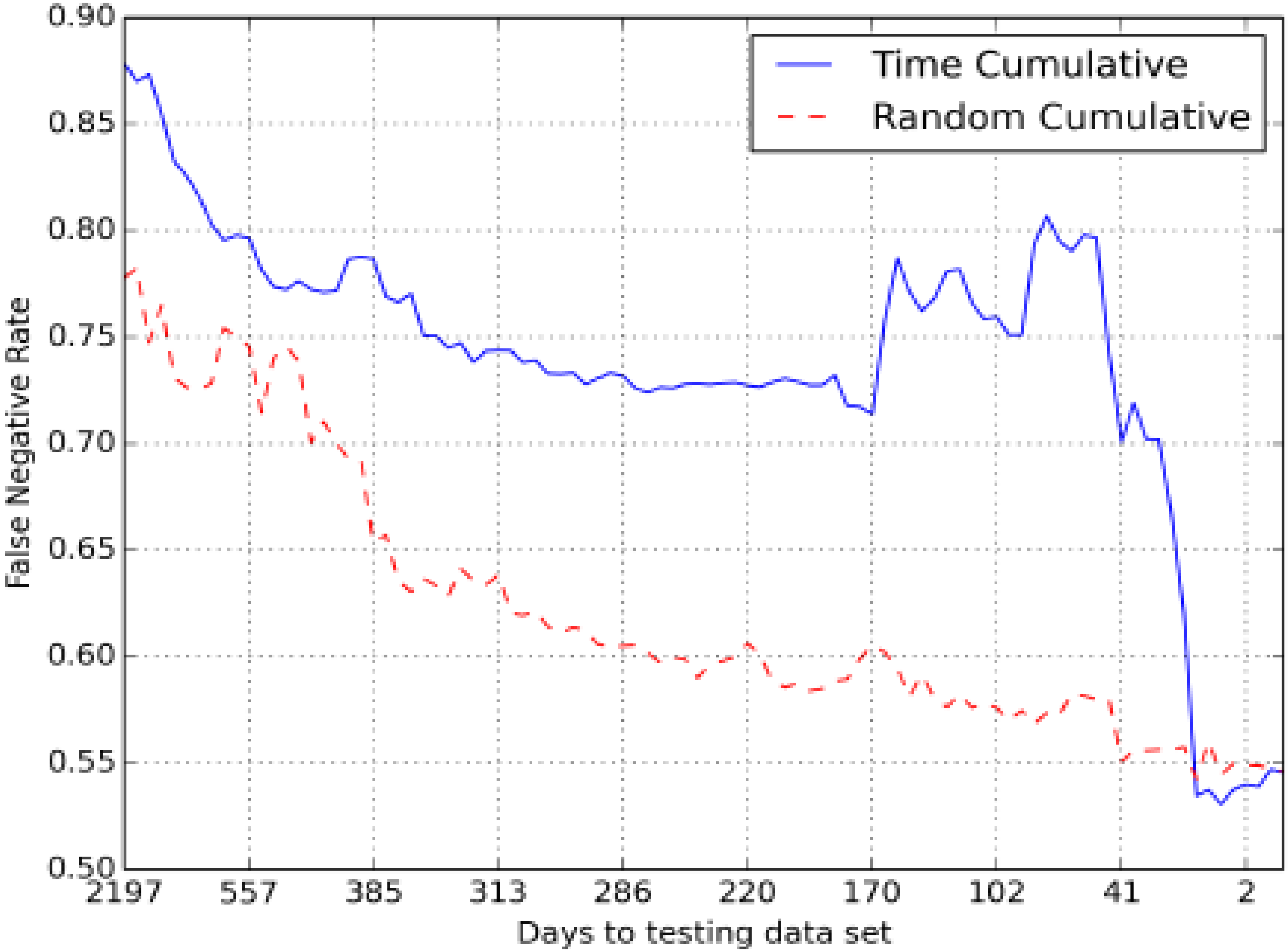

False Negative Rate
C
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
Predict Future
Predict Past
Random: train A, test B
Random: test A, train B

Result 1 _ conclusion

- The evaluation of ML based on security system should
  - Temporal nature of the instances
  - Avoid Random-cross-validation

Data Exploration – Experiments 2

- Fixed the testing set [most recent instances]
- Train SVM models
- Constant $C = 10^{-4}$
- Constant FPR < 1%
- Ignore the temporal order



Time Cumulative
Random Cumulative
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
False Negative Rate
2197
557
385
313
286
220
170
102
41
2
Days to testing data set

Conclusion

- Drift must be organized to limit the impact of campaigns
- Zero training error of high-impact instance means correctly classification
- Drift and temporal order must be respected in term of detector accuracy

Thank you

Questions?