Enabling Fair ML Evaluations for Security

Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, Lorenzo Cavallaro

Royal Holloway, University of London & King's College London

ML Security evaluations are flawed

Best practices for ML evaluations, such as k-fold CV, fail when i.i.d. assumptions do not hold, e.g., concept drift, adversarial ML.

Temporal Experimental Bias

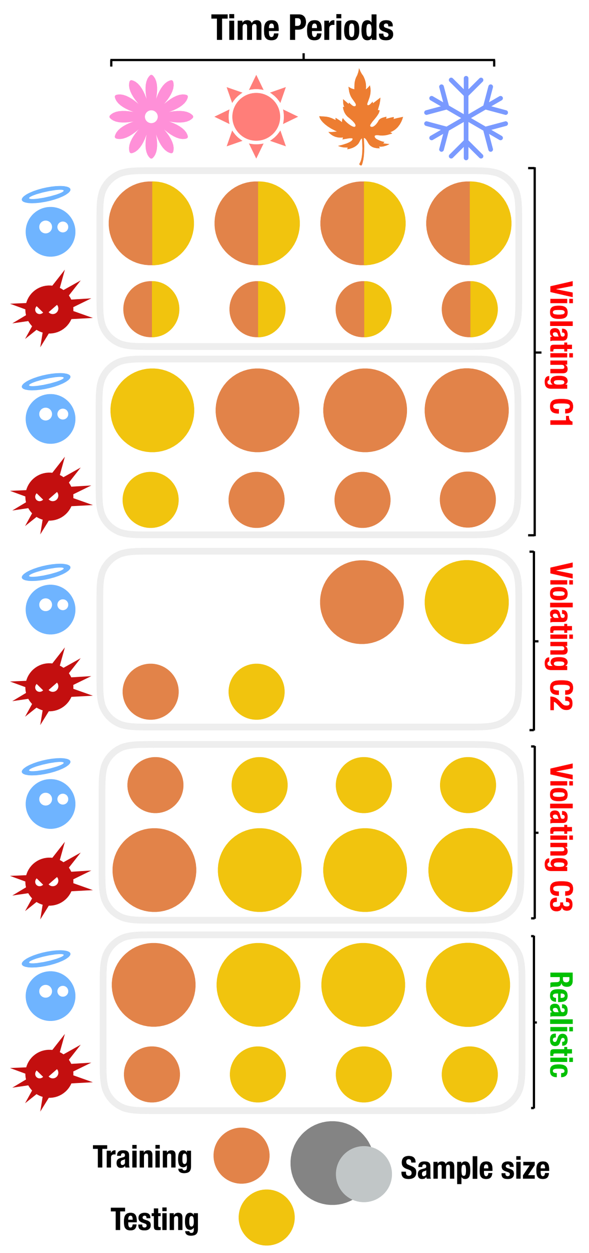
Caused by violating the temporal consistency of train and test sets.

Spatial Experimental Bias

Caused by using unrealistic class ratios in the test set.

TESSERACT

space-time bias-free evaluation framework

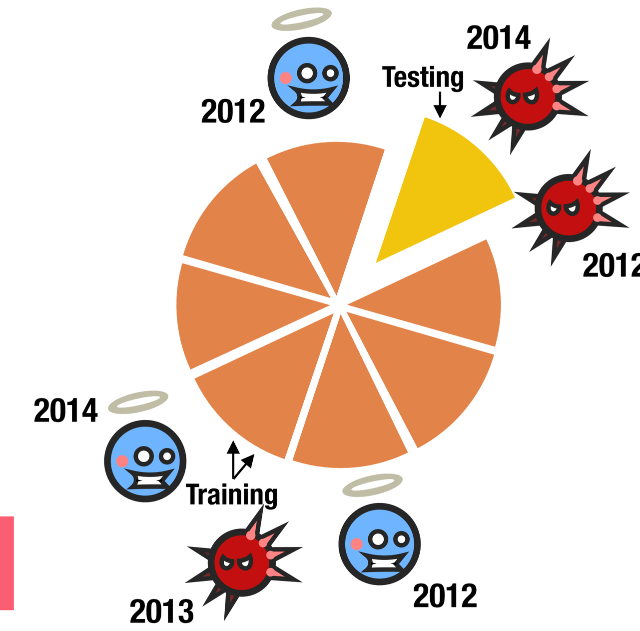


C1 Temporal training consistency

All the objects in the training must be strictly temporally precedent to those in the testing.

K-fold CV

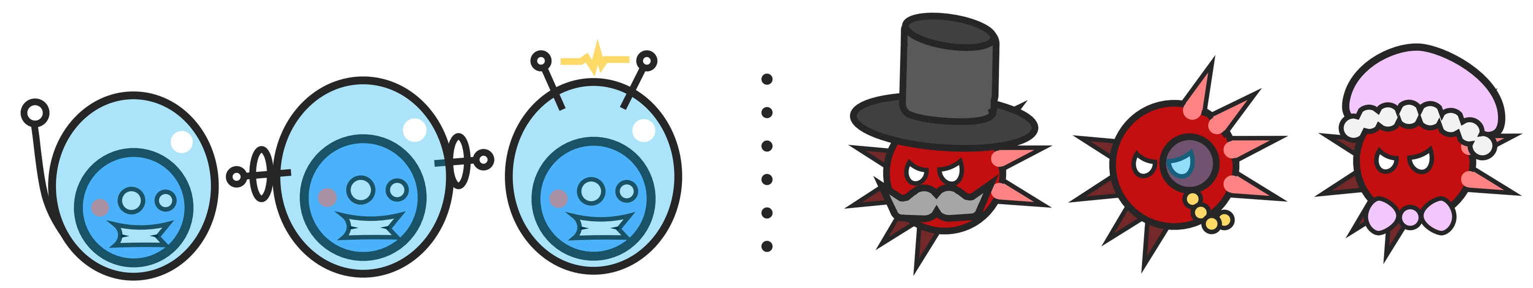
K-fold cross-validation randomly samples objects in a time-agnostic manner which fails to model a real-world deployment.



Violations use future knowledge in training.

C2 {good|mal}ware temporal consistency

In every testing period, all test objects must be from the same time window.



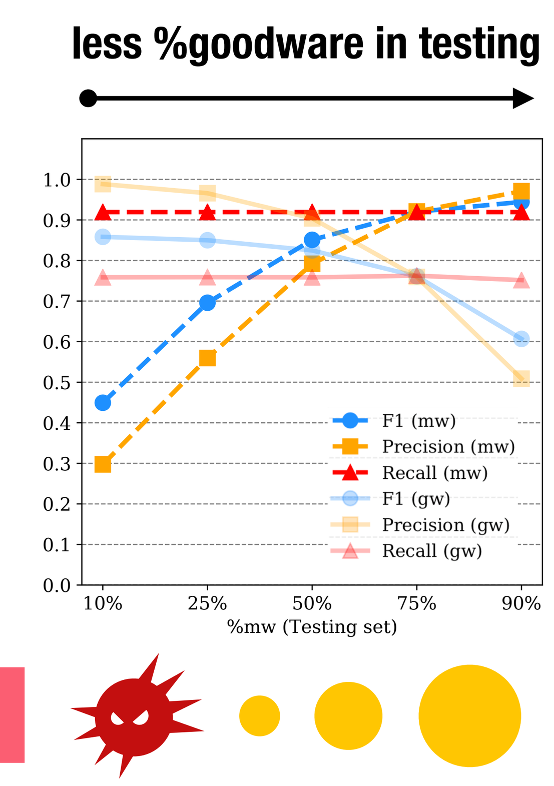
Violations may learn artifacts, such as old vs new APIs.

C3 Realistic testing classes ratio

The testing distribution must reflect real-world objects ratios, such as malware-to-goodware percentages in a given context.

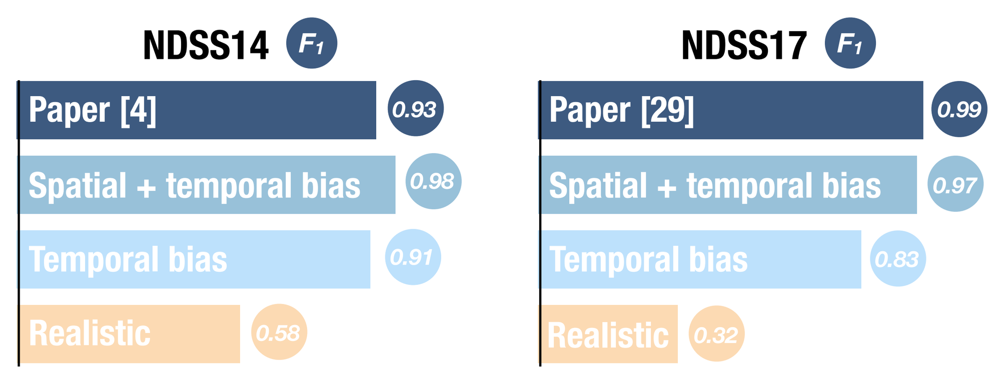
$$P^*_{mw} = \frac{TP}{TP + FP} \qquad R^*_{mw} = \frac{TP}{TP + FN}$$

*Undersampling goodware keeps R_{mw} steady and increases P_{mw}



Violations produce unrealistic results.

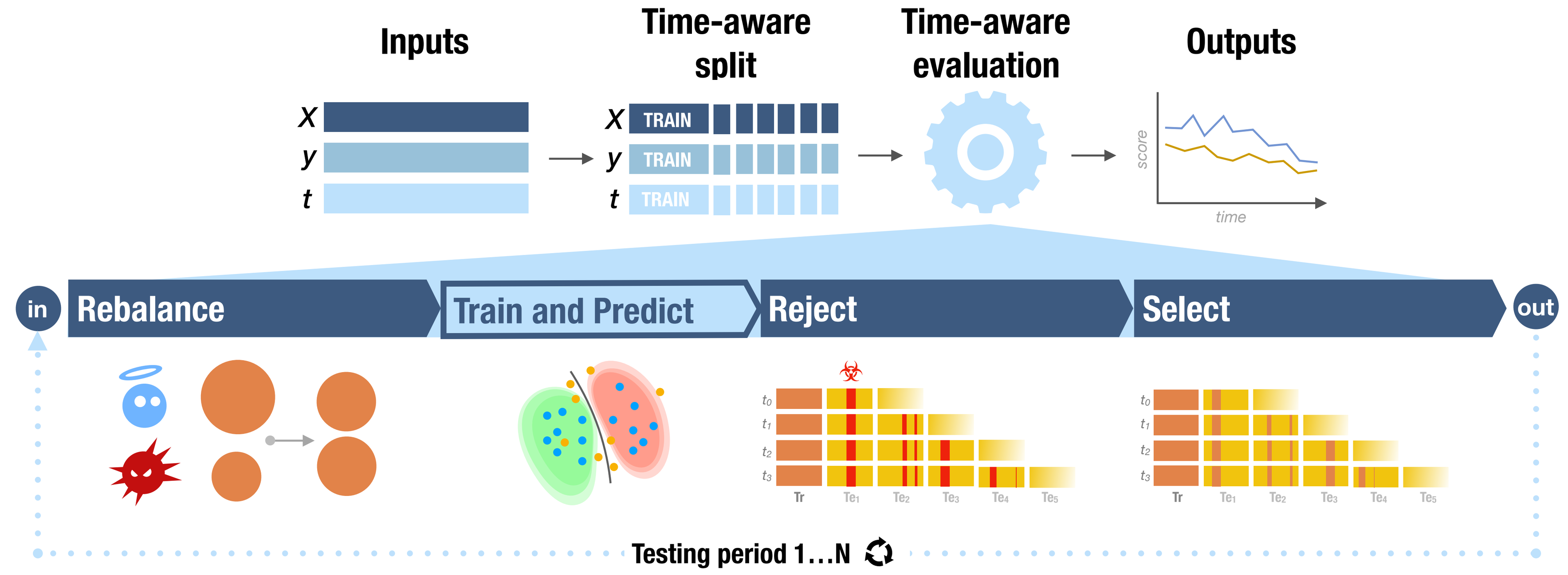
Obscuring real performance



[NDSS14] bit vector features (APIs, metadata, strings, etc.), linear SVM, 66-34% holdout evaluation.

[NDSS17] Markov Chain-derived features (caller-callee APIs), RF, k-fold CV and (biased) timeline evaluation.

TESSERACT: for when time matters!



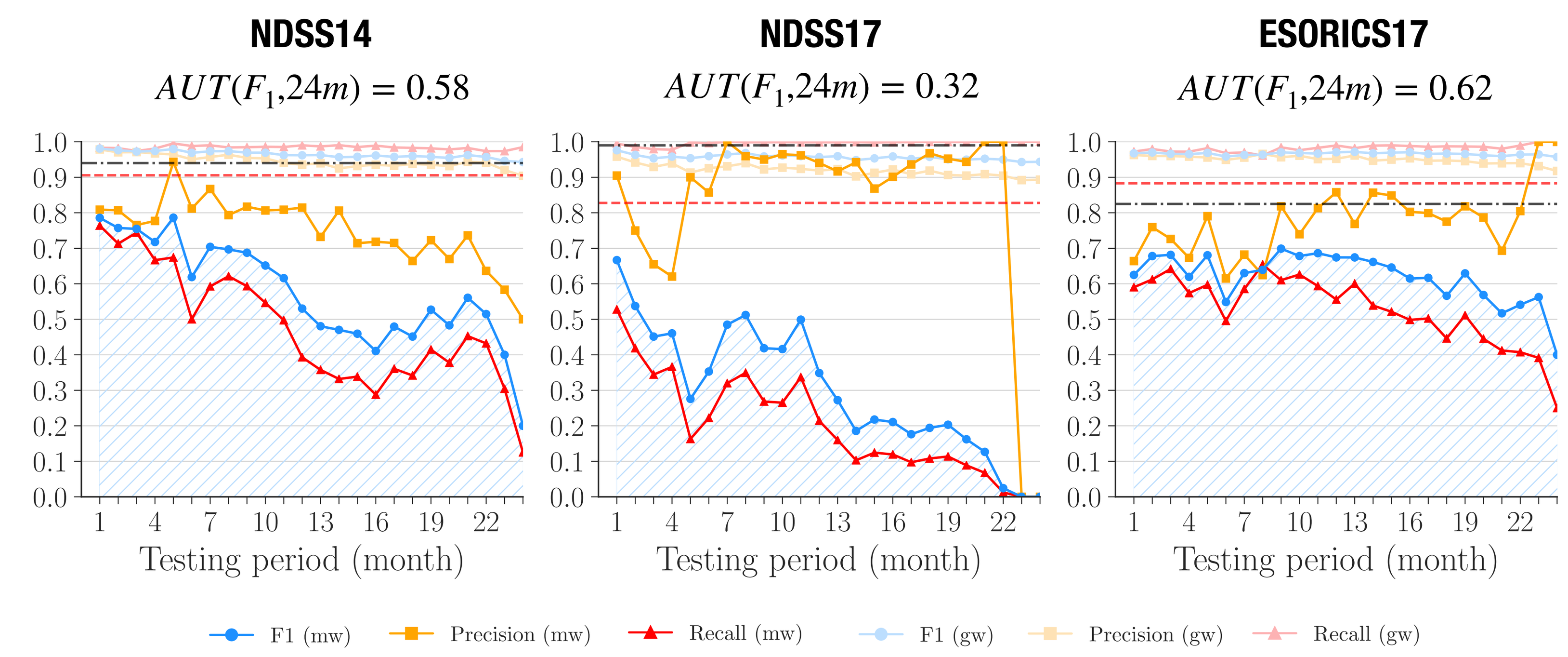
- Python 3 prototype
- Compatible with scikit-learn and Keras
- Support for different time partitions
- Time-aware plots and metrics
- Active learning and rejection strategies

Area Under Time (AUT)

$$AUT(f, \Delta) = \frac{1}{N-1} \sum_{k=1}^{N-1} \frac{[f(x_{k+1}) + f(x_k)]}{2N}$$

▼ **Available at: s2lab.kcl.ac.uk/projects/tesseract/**

Revealing real performance



F. Pendlebury*, F. Pierazzi*, R. Jordaney, J. Kinder, L. Cavallaro

- **When the Magic Wears Off: Flaws in ML for Security Evaluations (and What to Do about It)**—USENIX ENIGMA 2019
- **POSTER: Enabling Fair ML Evaluations for Security**—ACM CCS 2018
- **TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time**—arXiv 2018

Research funded by grants EP/L022710/1 and EP/P009301/1