RPAL: Recovering Malware Classifiers from Data Poisoning using Active Learning

Shae McFadden, Zeliang Kan, Lorenzo Cavallaro, Fabio Pierazzi
King's College London & University College London

Recovering From Poisoning

We introduce the following metrics to measure model recovery from poisoning

Parameter: Tolerance Margin

The **margin on vanilla performance** which denotes recovered performance if the poisoned model is within it.

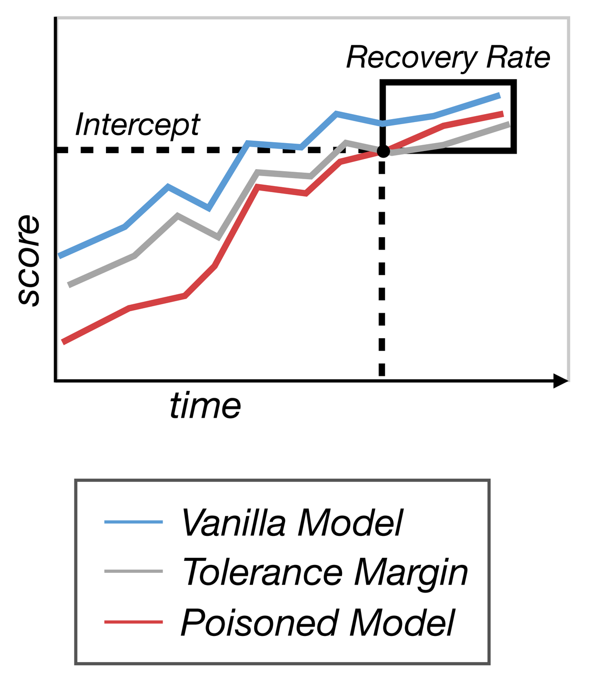
Metric: Intercept

The **first month** where the poisoned model's performance is within the *Tolerance Margin*.

Metric: Recovery Rate

The **percentage of months** that the poisoned model **maintains** within the *Tolerance Margin* after the *Intercept*.

RPAL

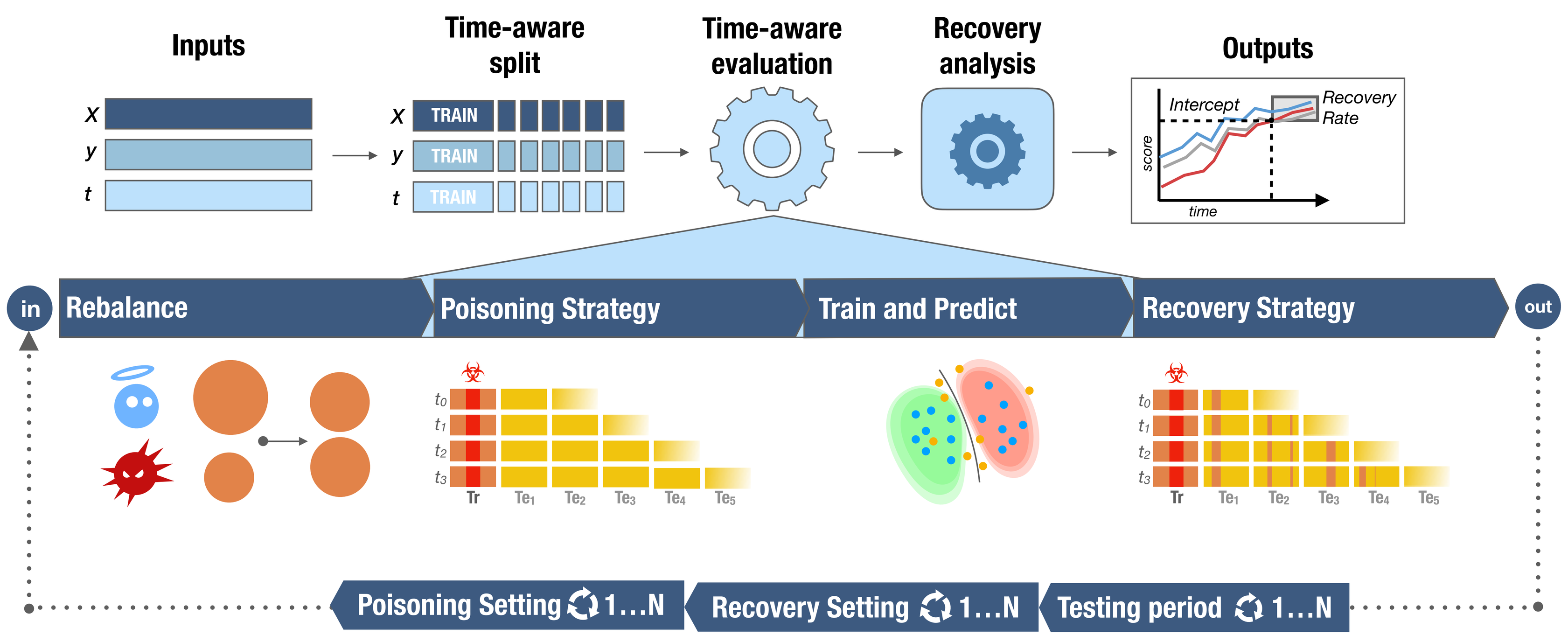


How to Compare Recovery Performance

A system has better recovery if it has a <u>sooner *Intercept* and a higher *Recovery Rate*</u>. If only one of these conditions is true then it is a mixed result.

RPAL Evaluation Framework

The RPAL framework utilizes *Tolerance Margin, Intercept,* and *Recovery Rate* to **evaluate the recovery of a system**.



Experimental Settings

Time-Stamped Data: The dataset consists of *129,728* applications, ranging from *2014–2016* with a *10%* malware distribution and is extracted to both Drebin[1] and MaMaDroid[2].

Time-Aware Evaluation: Tesseract[3] is used to perform the time-aware evaluations, using *2014* data for training and *2015–2016* data for testing.

Recovery Strategy: The recovery strategy is uncertainty sampling with *2%–16%* sampling rates and the *Tolerance Margin* is set to *0.02* for all experiments.

Poisoning Strategy: The poisoning strategy is label-flip poisoning with enforced maintenance of class distribution and *2%–16%* poisoning rates.

Recovery Results Table

Speed of Recovery

The *Interept* consistently increases when both rates are increased equally showing that poisoning has a stronger impact on intercept.

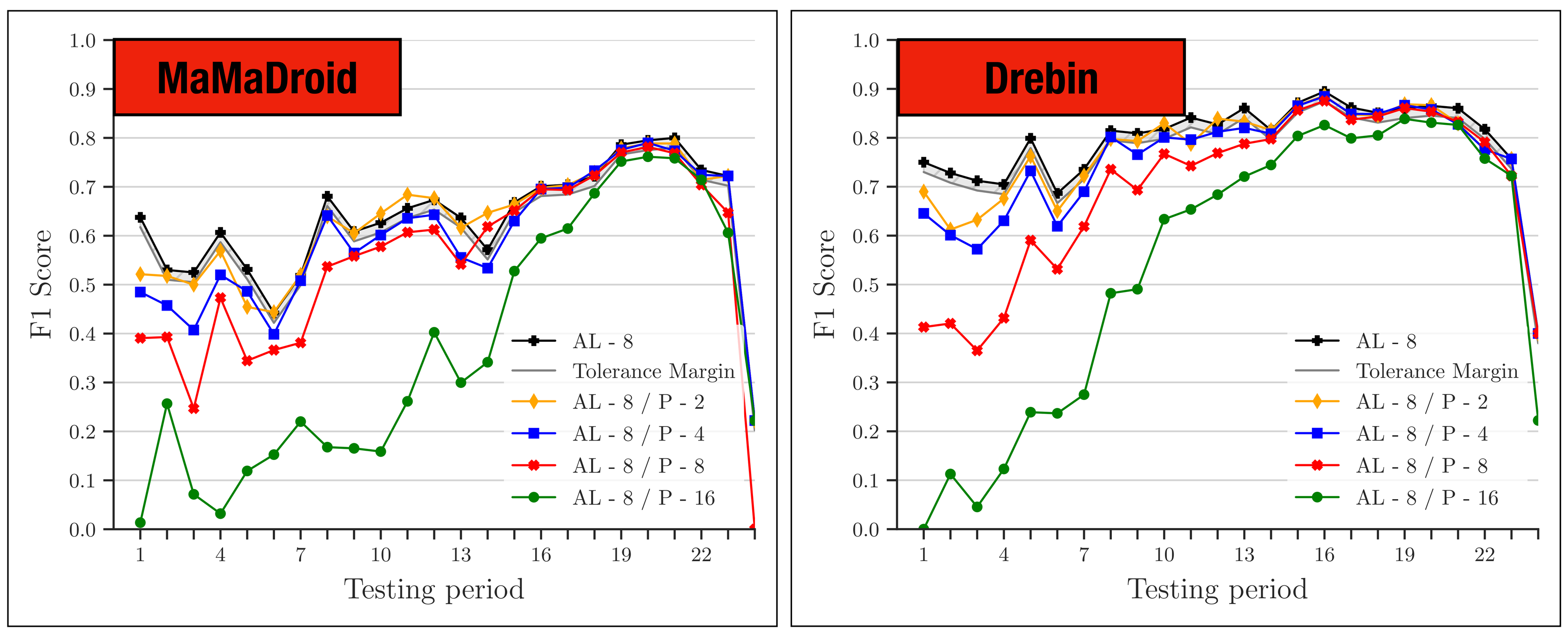
Results			Tolerance Margin = 0.02			
Feature Extraction	Active Learning Rate	Recovery Metric	Poisoning Rate			
			2%	4%	8%	16%
MaMaDroid	2%	Intercept (Month)	9	11	19	21
		Recovery Rate (%)	75%	64%	83%	75%
	4%	Intercept (Month)	9	11	11	22
		Recovery Rate (%)	88%	64%	71%	67%
	8%	Intercept (Month)	2	7	14	24
		Recovery Rate (%)	74%	50%	64%	100%
	16%	Intercept (Month)	3	12	16	21
		Recovery Rate (%)	73%	85%	89%	75%
Drebin	2%	Intercept (Month)	9	16	21	>24
		Recovery Rate (%)	62%	44%	50%	0%
	4%	Intercept (Month)	8	12	19	>24
		Recovery Rate (%)	82%	62%	33%	0%
	8%	Intercept (Month)	7	8	14	>24
		Recovery Rate (%)	78%	71%	64%	0%
	16%	Intercept (Month)	4	10	14	19
		Recovery Rate (%)	86%	80%	82%	67%

The table above displays the *Intercept* and *Recovery Rate* for all active learning sampling and poisoning rates.

Poisoning Rates

Poisoned Performance Convergence Over Time

Across the four poisoning settings the same trend of converging on the vanilla performance over time can be observed.

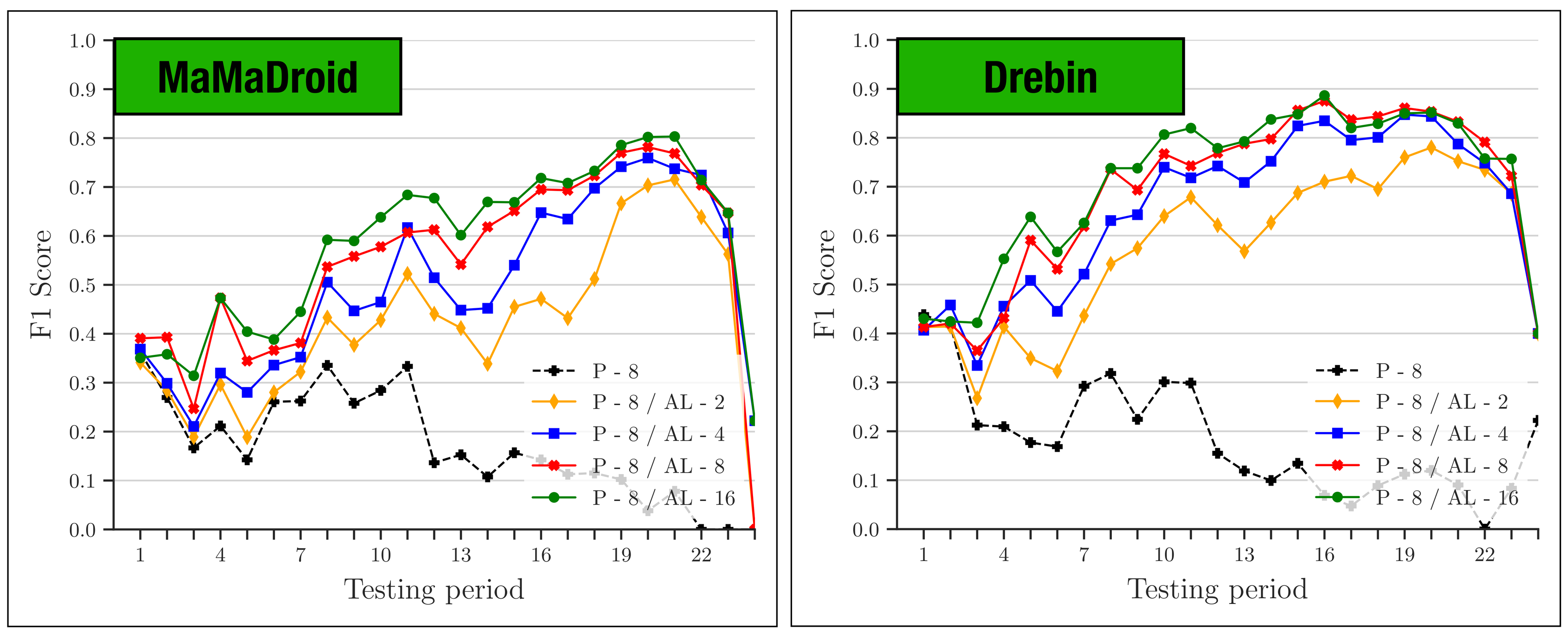


The plots show the impact of increasing poisoning rates against a fixed active learning rate.

Active Learning Rates

Diminishing Returns of Active Learning

Across the four active learning settings the diminishing returns of increased active learning rates can be observed against a fixed poisoning rate.



The plots illustrate the impact of varying active learning rates on a fixed poisoning rate.

Discussion

Drebin's Superior Performance: Across the plots, MaMaDroid has better performance *2.5%* and, <u>Drebin has better performance *96.5%* of the time</u> with the remaining *1%* being tied.

MaMaDroid's Superior Recovery Performance: Across, the table, out of the *sixteen* settings, <u>MaMaDroid is better in *eight* setting</u> with the <u>remaining *eight* being mixed results</u>.

Key Result: Feature

The **feature abstraction** has a significant **impact on recovery**, and a better-performing system does not equate to a better-recovering system.

Key Result: Intercept

Higher poisoning rates of the training dataset result in a **delayed intercept**, this corresponds to a diminished return for increasing active learning rates and not in poisoning rates.

Conclusion

Novelty of this Research: To the best of our knowledge, we are the first to evaluate the recovery *over time* of a classification system from poisoning.

Key Takeaway: Drift mitigation strategies *can* indeed facilitate recovery of the model, however, the speed of recovery *heavily* depends on the components of the system and data considered.

Shae McFadden, Zeliang Kan, Lorenzo Cavallaro, Fabio Pierazzi. 2023. ***POSTER: RPAL—Recovering Malware Classifiers from Data Poisoning using Active Learning***. In *CCS*.

This research has been partially supported by a research service agreement with the Alan Turing Institute's AI for Cyber Defense (AICD) Research Centre, by the King's-China Scholarship Council Ph.D. Scholarship programme (K-CSC), by a Google ASPIRE research award, and a by EPSRC Grant EP/X015971/1.

References

[1] Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, Konrad Rieck, and CERT Siemens. 2014. Drebin: Effective and Explainable Detection of Android Malware in your Pockert. In *Ndss*, Vol. 14. 23-26.

[2] Lucky Onwuzurike, Enrico Mariconti, Panagiotis Andriotis, Emiliano De Cristofaro, Gordon Ross, and Gianluca Stringhini. 2019. MaMaDroid: Detecting Android Malware by Building Markov Chains of Behabioral Models (Extended Version). *ACM Transactions on Privacy and Security (TOPS) 22*, 2 (2019), 1-34.

[3] Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, and Lorenzo Cavallaro. 2019. TESSERACT: Eliminating Experimental Bias in Malware Classification across Space. In *28th USENIX Security Symposium (USENIX Security 19)*. 729-746.