Poster: RPAL—Recovering Malware Classifiers from Data Poisoning using Active Learning

Shae McFadden
King's College London
London, United Kingdom
shae.mcfadden@kcl.ac.uk

Zeliang Kan
King's College London & University College London
London, United Kingdom
zeliang.kan@kcl.ac.uk

Lorenzo Cavallaro
University College London
London, United Kingdom
l.cavallaro@ucl.ac.uk

Fabio Pierazzi
King's College London
London, United Kingdom
fabio.pierazzi@kcl.ac.uk

ABSTRACT

Intuitively, poisoned machine learning (ML) models may forget their adversarial manipulation via retraining. However, can we quantify the time required for model recovery? From an adversarial perspective, is a small amount of poisoning sufficient to force the defender to retrain significantly more over time?

This poster paper proposes RPAL, a new framework to answer these questions in the context of malware detection. To quantify recovery, we propose two new metrics: *intercept*, i.e., the first time in which the poisoned model's and vanilla model's performance intercept; *recovery rate*, i.e., the percentage of time after intercept that the poisoned model's performance is within a *tolerance margin* which approximates the vanilla model's performance. We conduct experiments on an Android malware dataset (2014 – 2016), with two feature abstractions based on Drebin and MaMaDroid, with uncertainty-sampling active learning (retraining), and label flipping (poisoning). We utilize the introduced parameter and metrics to demonstrate (i) how the active learning and poisoning rates impact recovery and (ii) that feature representation impacts recovery.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**; **Systems security**.

KEYWORDS

supervised learning; malware detection; poisoning; active learning

ACM Reference Format:
Shae McFadden, Zeliang Kan, Lorenzo Cavallaro, and Fabio Pierazzi. 2023. Poster: RPAL—Recovering Malware Classifiers from Data Poisoning using Active Learning. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), November 26–30, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3576915.3624391

CCS '23, November 26–30, 2023, Copenhagen, Denmark

ACM ISBN 979-8-4007-0050-7/23/11.
https://doi.org/10.1145/3576915.3624391

1 INTRODUCTION

This paper investigates how retraining strategies help recover the model after training-time data poisoning attacks. Firstly, we propose a formalization of recovery from poisoning by introducing two new metrics: *intercept* and *recovery rate*. The intercept aims to estimate *when* the poisoned model's performance approximates the retrained vanilla model. Since model performance exhibits fluctuations in non-stationary context [2, 6], the recovery rate aims to quantify the percentage of time that the poisoned model has a performance equal to or higher than the vanilla retrained model. We introduce the concept of *tolerance margin*, which defines *how much* approximation is tolerated when assuming a model is recovered.

Finally, we propose RPAL as an extensible framework that can assess the recovery rate of an ML model for a given set of parameters, models, retraining and poisoning strategies. We evaluate RPAL in the context of Android malware detection to quantify recovery from label-flip poisoning via active learning retraining based on uncertainty sampling using the dataset from Pendlebury et al. [6] which contains the Drebin [1] and MaMaDroid [5] feature spaces.

Our results demonstrate that an ML model can recover from a poisoning attack subject to the strength of the poisoning attack and retraining mechanism. Different feature abstractions under the same classifier can greatly affect the speed of recovery. We show interesting trade-offs in the defender's diminished return in poison recovery when increasing active learning rates.

Novelty. Existing literature has indeed proposed some defenses for poisoning such as fine-pruning [4], however, to the best of our knowledge, we are the first to evaluate the recovery *over time* of a classification system from poisoning. Moreover, we are the first to consider it in the context of malware detection.

Contributions. In summary, we make the following contributions: (i) we propose new metrics and parameters for understanding recovery from poisoning over time, (ii) and we propose an evaluation framework, RPAL, through which (iii) we demonstrate trade-offs of poisoning-recovery in the Android malware context and confirm that feature abstractions impact recovery.

2 RPAL EVALUATION FRAMEWORK

Here we discuss the threat model, parameters, metrics and evaluation pipeline (Figure 1) of the RPAL framework.

Threat Model. In RPAL, the threat model requires defining *both* attackers and defenders capabilities, along with the attackers objective (e.g., increasing FNs or FPs). The attacker's capability is defined via a poisoning strategy [3, 4] and the percentage of training data affected. Instead, the defender capability is defined via a recovery strategy (e.g., active learning [7]) and the retraining budget. A recovery strategy is a method which enables the model to adapt over time, however, recovery strategies come with their own operational costs (such as labeling cost). When deciding on a recovery strategy it is important to take into account a realistic operational budget. When choosing a poisoning strategy for evaluation it is important that it fits within the threat model chosen and that the adversarial manipulations performed are realistic.

Parameters and Metrics. For recovery evaluation, poisoned model $\mathcal{M}_{\mathcal{P}}$'s and non-poisoned model $\mathcal{M}$'s only difference should be the poisoning at initial training time. We define tolerance margin, intercept and recovery rate as follows.

Definition 2.1 (Parameter: Tolerance Margin). The tolerance margin δ ($\delta \in \mathbb{R}, 0 \leq \delta \leq 1$) is an input hyper-parameter of RPAL that is subtracted to the monthly performance of a clean model $\mathcal{M}$, such that the monthly performance of the poisoned $\mathcal{M}_{\mathcal{P}}$ if compared with that of $\mathcal{M}$ minus δ.

Definition 2.2 (Metric: Intercept). Given a tolerance margin δ, the intercept I is a metric defined as the first month τ where $\mathcal{M}_{\mathcal{P}}$'s monthly performance is greater than or equal to $\mathcal{M}$'s monthly performance minus δ.

Definition 2.3 (Metric: Recovery Rate). The recovery rate R is defined as the percentage of months that $\mathcal{M}_{\mathcal{P}}$'s performance is greater than or equal to $\mathcal{M}$'s performance minus δ, after the intercept I.

To the best of our knowledge, the metrics defined in this paragraph are the first temporally-aware poisoning recovery metrics.

Recovery Evaluation. Figure 1 shows the pipeline of the RPAL evaluation framework. The initial input is a timestamped dataset, consisting of a feature matrix X, labels y, and time-stamps t. The dataset is temporally split into a training set and monthly testing sets via the TESSERACT framework [6]. In the time-aware evaluation, the training dataset is rebalanced to a realistic class distribution (e.g., approximately 10% in the case of Android malware [6]). The dataset is then poisoned based on the current iteration's poisoning rate (starting from 0%, for a clean model baseline), and is then used to train the model. The model predicts on the current test month data. After prediction, the recovery strategy (e.g., active learning [7]) selects samples to retrain the model on. The process of train, predict and sample is then repeated for all testing months. After all test months have been evaluated then the process is repeated for all poisoning and recovery strategy rates in the evaluation settings. Once all setting evaluations have been completed then, the intercept and recovery rate are extracted from the results.

3 EVALUATION

We consider two research questions: does the feature space affect the recovery rate? (RQ-FEATURE) and how do the active learning and poisoning rates impact the recovery time? (RQ-INTERCEPT).

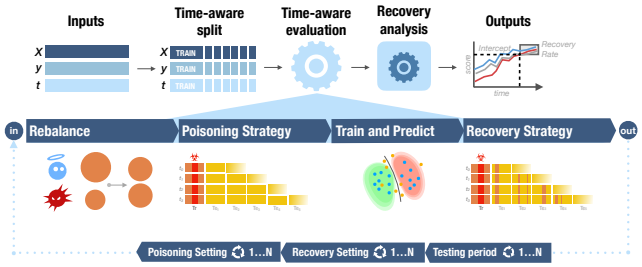
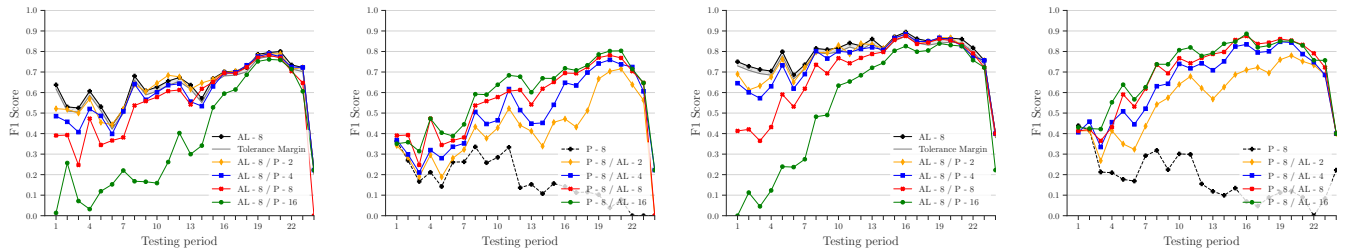


Figure 1: RPAL Evaluation Framework.

Experimental Settings. We use the TESSERACT dataset by Pendlebury et al. [6], which ranges from 2014 to 2016 and offers two feature abstractions: MAMADROID [5] and DREBIN [1]. We train on the 57,740 samples from 2014 and test on the 71,988 samples from 2015/2016 however, a lack of samples in month 24 of testing causes performance drop off. We assume the defender relies on active learning with uncertainty sampling [7] as the recovery strategy for our evaluation with strength from 2% to 16%. The tolerance margin δ for the experiments in this research was set to 0.02. We assume the attacker has the capability to fully access the training data and is utilizing label-flip poisoning [4] to compromise the general availability of the model with granularity 2% to 16%. We utilize the Random Forest classifier with the settings of 101 decision trees with a max depth of 64, the same as the original paper [5].

Compare Different Feature Abstractions. The MAMADROID and DREBIN experiments both utilize the same underlying dataset, which allows the evaluation of the impact feature space has on recovery thereby addressing research question RQ-FEATURE. With a constant active learning rate, the plots Figure 2a and Figure 2c show a similar trend between the different rates of poisoning, however, when comparing the monthly F_1 performance of the same poisoning rates between the two feature abstractions with the same fixed active learning rate as in the plots, MAMADROID has better performance 1% of the time and DREBIN has better performance 98% of the time across the four poisoning settings. With a constant poisoning rate, the plots Figure 2b and Figure 2d show a similar starting performance, however, when comparing the monthly F_1 performance of the same active learning rates between the two feature abstractions with the same fixed rate as in the plots, MAMADROID has better performance 4% of the time, DREBIN has better performance 95% of the time with the remaining 1% being tied across the four active learning settings. When comparing the two feature extractions via the tables Table 1and Table 2. we can see that although MAMADROID's performance is worse than DREBIN, it outperforms DREBIN in recovery. When comparing the recovery across the 16 settings, MAMADROID is better in 8, DREBIN is better in 0, and 8 are deemed mixed results.

RQ-FEATURE. The feature abstraction has a significant impact on recovery, and a better-performing system does not equate to a better-recovering system. Although MAMADROID's performance is worse than DREBIN, MAMADROID achieves better recovery statistics on D1416.



(a) Fixed Active Learning Rate MaMaDroid **(b) Fixed Poisoning Rate MaMaDroid** **(c) Fixed Active Learning Rate Drebin** **(d) Fixed Poisoning Rate Drebin**

Figure 2: Recovery plots showing the impact of single variable change by varying either poisoning or active learning rate.

Table 1: MaMaDroid feature space results.

MaMaDroid Results		Tolerance Margin = 0.02			
Active Learning Rate		Poisoning Rate 2%	4%	8%	16%
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%

Table 2: Drebin feature space results.

Drebin Results		Tolerance Margin = 0.02			
Active Learning Rate		Poisoning Rate 2%	4%	8%	16%
2%	Intercept (Month)	9	16	21	X
	Recovery Rate (%)	62%	44%	50%	0%
4%	Intercept (Month)	8	12	19	X
	Recovery Rate (%)	82%	62%	33%	0%
8%	Intercept (Month)	7	8	14	X
	Recovery Rate (%)	78%	71%	64%	0%
16%	Intercept (Month)	4	10	14	19
	Recovery Rate (%)	86%	80%	82%	67%

Speed of Recovery. Through comparisons between the tables (Table 1 and Table 2), we can evaluate the rate in which the models recover thereby addressing research question RQ-Intercept. When comparing the intercepts of experiments with equal active learning and poisoning rates, we can see that the intercept consistently increases as the rates increase. This shows that an increase in poisoning rate has a larger impact on the intercept than the active learning rate. Poisoning could force a substantially higher active learning rate, which is the result of two factors: firstly, the known diminishing returns of active learning [6], which can be observed in the converging active learning rates in the fixed active learning plots in Figure 2; secondly, the ever increasing impact of poisoning, which can be observed by the growing separation in the poisoning rates in the fixed active learning plots in Figure 2.

RQ-Intercept. Higher poisoning rates of the training dataset result in a delayed intercept, even if the active learning rate is much higher than the poisoning rate (Table 1 and Table 2). This also corresponds to a diminished return in recovery speed for increasing active learning rates.

4 CONCLUSION

We demonstrated that drift mitigation strategies can indeed facilitate recovery of the model whereby the model forgets the poisoning effect over time. However, the speed of recovery heavily depends on the components of the system and data considered. Our initial investigation paves the way for further research on this topic which in addition to exploring more scenarios, could also investigate drift mitigation strategies that take poisoning into account for optimal recoveries, as well as evaluating impact of existing poisoning defenses to time-aware recovery.

ACKNOWLEDGMENTS

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

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 Pocket.. In *Ndss*, Vol. 14. 23–26.
[2] Federico Barbero, Feargus Pendlebury, Fabio Pierazzi, and Lorenzo Cavallaro. 2022. Transcending Transcend: Revisiting Malware Classification in the Presence of Concept Drift. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 805–823.
[3] Battista Biggio and Fabio Roli. 2018. Wild Patterns: Ten years After the Rise of Adversarial Machine Learning. (2018).
[4] Antonio Emanuele Cinà, Kathrin Grosse, Ambra Demontis, Sebastiano Vascon, Werner Zellinger, Bernhard A Moser, Alina Oprea, Battista Biggio, Marcello Pelillo, and Fabio Roli. 2022. Wild Patterns Reloaded: A Survey of Machine Learning Security against Training Data Poisoning. *Comput. Surveys* (2022).
[5] Lucky Onwuzurike, Enrico Mariconti, Panagiotis Andriotis, Emiliano De Cristofaro, Gordon Ross, and Gianluca Stringhini. 2019. MaMaDroid: Detecting Android Malware by Building Markov Chains of Behavioral Models (Extended Version). *ACM Transactions on Privacy and Security (TOPS)* 22, 2 (2019), 1–34.
[6] Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, and Lorenzo Cavallaro. 2019. TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time. In *28th USENIX Security Symposium (USENIX Security 19)*. 729–746.
[7] Burr Settles. 2009. Active learning literature survey. (2009).