Poster Misleading Metrics: On Evaluating ML for Malware with Confidence

Roberto Jordaney, Zhi Wang, Davide Papini, Ilia Nouretdinov, Kumar Sharad, and Lorenzo Cavallaro
Royal Holloway, University of London

Malware poses a serious and challenging threat and due to the sheer scale the need for automated learning-based approaches to deal with it has become rapidly clear. Swift analysis and prompt detection of these threats present one of the most pressing and important issues that plague the security of the Internet and its users. With more than 550,000 unique malware samples per day reported in Q4 2015[1], it is clear that manual analysis does not scale and therefore the shift is towards automatic and adaptive techniques that can identify unknown and previously-unseen threats. To this end, machine learning, with a particular emphasis on clustering and classification, has long been acknowledged as a promising technique to address such a fundamental need in a number of security-related domains, including botnet [4, 10], mobile [1], and traditional malware [2, 3, 8, 9, 11].

The advances in the area would seem to suggest that the problem is almost solved. However, assessing the actual results of a given algorithm is problematic. With few exceptions, e.g., [1, 10, 13], the lack of publicly-available datasets hinders the ability to reproduce and compare results. Furthermore, the usage of traditional metrics (e.g., accuracy, precision, recall) to assess the performance of a machine learning algorithm might produce misleading results: such metrics report statistics on correct and incorrect decisions, but do not capture their quality and are hence ill-suited to evaluate a given task. The problem is further exacerbated when machine learning algorithms are deployed in real-world settings, especially in a context which often sees new labels (malware families) and changes in the underlying data distribution (malware variants, new behaviors).

Li et al. consider this problem [6], empirically showing that traditional metrics with high accuracy do not necessarily imply that the underlying machine learning is good. They show how the dataset is often chosen to support the claim of the author. Their work focuses primarily on methods specifically built on the available datasets that suffered from data over-fitting issues. Conversely, in our work, we aim at tackling the problem on a broader scope, providing a way to assess the quality of a given algorithm in a scientific and rigorous manner.

Our work aims to provide quality metrics that can help in the development of machine learning algorithms that provide an insight into the process and furthermore help predict the performance of a deployed algorithm.

Another factor that influences the outcome of a machine learning algorithm is the process of feature selection. In fact, the algorithm in itself, although designed appropriately, might not work if the selected features do not correctly capture and separate the peculiarities of the samples. On the other hand, one might say that any algorithm works as long as features are properly selected, hence it is important to focus on the feature selection process mainly. As a matter of fact, there might be algorithms that perform very well even with badly separated features or algorithms that do not work even with very well separated features.

In this work, we show that, to have a comprehensive evaluation of a given approach, we need to focus on the combination of the algorithm and the feature selection process as a whole.

To address these problems, we propose *conformal evaluator*, an evaluation framework that uses statistical metrics to provide a quality evaluation of a given machine learning algorithm.

In particular, we aim to answer the following research questions.

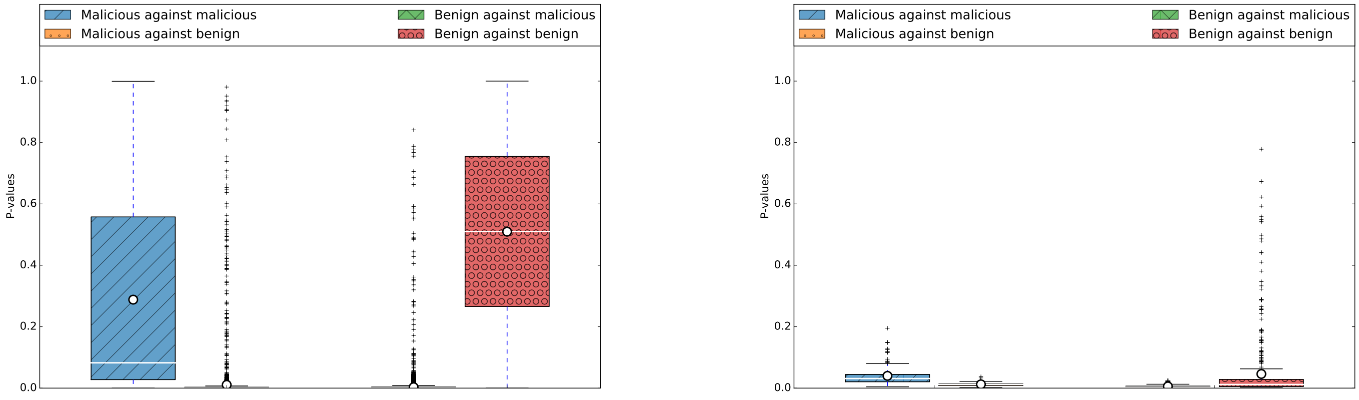
Q1: Can we define quality metrics to better understand and design a detection/classification algorithm?

We propose two novel evaluation metrics to capture the quality of an algorithm's results: algorithm credibility and algorithm confidence. The algorithm credibility captures how conformant the algorithm's prediction is for a given data point. The algorithm confidence measures how distinguished a sample's prediction is compared to others. Leveraging these metrics, we propose two novel analyses to produce qualitative and quantitative metrics to evaluate the correctness of a machine learning algorithm. In practical settings, our technique can be plugged on top of any machine learning algorithm that uses a real number score to make predictions.

Q2: What are the insights provided by the proposed quality metrics?

Our evaluation metrics provide a quantifiable per-choice level of assurance and reliability which can be exploited to evaluate its output. The intuition is that, if the choices made by the algorithm are supported by statistical evidence, it it is more likely that the decision is indeed correct. Moreover, we show how feature-only based methods, tailored to showing family interference, without taking the process carried on by the algorithm (e.g., t-SNE [12] and PCA [5]) into consideration do not produce sound results.

[1] http://www.mcafee.com/us/security-awareness/articles/mcafee-labs-threats-report-mar-2016.aspx



(a) P-values computed during the design phase of the algorithm.

(b) P-values computed on a deployed scenario.

Fig. 1: The figure shows how the distribution of p-values of malicious Android malware trained on a dataset collected between 2010 and 2012 decays radically when computed against a dataset colleced in 2014. In this context, the quality degradation suggests a retraining of the model.

Q3: How can the quality metrics facilitate the design of better machine learning models?

The credibility and confidence of a machine learning algorithm capture the uncertainty of a given prediction introduced due to various synthetic and organic changes in the ecosystem. The algorithm could be improved by only considering predictions above a certain credibility and confidence threshold. This does rule out predictions related to samples falling below the threshold but it keeps the error rate to expected levels as well as highlights the challenges in prediction.

Q4: How can we detect the changes in data distribution (e.g., new families, new variants) in a deployed scenario?

We show how to identify a decay in the quality of the results without having access to true labels (available only during the design phase). This provides an indicator that may suggest to retrain our models, focus on problematic cases, or automatically identify a new malware families. Figure 1 depicts such a scenario, where the distribution of p-values of malicious Android malware samples trained on a dataset collected between 2010 and 2012 (Figure 1a, Drebin dataset [1]) decays considerably when computed against a dataset collected in 2014 (Figure 1b, Marvin dataset [7])—the lack of labels in deployment settings would have not allowed to identify the process decay, but this seems quite straightforward with the analysis provided by CE.

Acknowledgments

This research has been partially supported by the UK EPSRC grants EP/K033344/1, EP/K006266/1, and EP/L022710/1.

References

[1] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck. DREBIN: effective and explainable detection of android malware in your pocket. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*, 2014.

[2] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Krügel, and E. Kirda. Scalable, behavior-based malware clustering. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2009, San Diego, California, USA, 8th February - 11th February 2009*, 2009.

[3] M. Christodorescu, S. Jha, and C. Kruegel. Mining specifications of malicious behavior. In *Proceeding of the 1st Annual India Software Engineering Conference, ISEC 2008, Hyderabad, India, February 19-22, 2008*, pages 5–14, 2008.

[4] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA*, pages 139–154, 2008.

[5] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[6] P. Li, L. Liu, D. Gao, and M. K. Reiter. On challenges in evaluating malware clustering. In *Recent Advances in Intrusion Detection, 13th International Symposium, RAID 2010, Ottawa, Ontario, Canada, September 15-17, 2010. Proceedings*, pages 238–255, 2010.

[7] M. Lindorfer, M. Neugschwandtner, and C. Platzer. Marvin: Efficient and comprehensive mobile app classification through static and dynamic analysis. In *Proceedings of the 39th Annual International Computers, Software and Applications Conference (COMPSAC)*, 7 2015.

[8] R. Perdisci, D. Ariu, and G. Giacinto. Scalable fine-grained behavioral clustering of http-based malware. *Computer Networks*, 57(2):487–500, 2013.

[9] R. Perdisci, W. Lee, and N. Feamster. Behavioral clustering of http-based malware and signature generation using malicious network traces. In *Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2010, April 28-30, 2010, San Jose, CA, USA*, pages 391–404, 2010.

[10] M. Z. Rafique and J. Caballero. FIRMA: malware clustering and network signature generation with mixed network behaviors. In *Research in Attacks, Intrusions, and Defenses - 16th International Symposium, RAID 2013, Rodney Bay, St. Lucia, October 23-25, 2013. Proceedings*, pages 144–163, 2013.

[11] K. Rieck, T. Holz, C. Willems, P. Düssel, and P. Laskov. Learning and classification of malware behavior. In *Detection of Intrusions and Malware, and Vulnerability Assessment, 5th International Conference, DIMVA 2008, Paris, France, July 10-11, 2008. Proceedings*, pages 108–125, 2008.

[12] L. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[13] Y. Zhou and X. Jiang. Dissecting android malware: Characterization and evolution. In *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA*, pages 95–109, 2012.