Poster: Barometer: Sizing Up Android Applications Through Statistical Evaluation

Santanu Kumar Dash, Guillermo Suarez-Tangil, Johannes Kinder and Lorenzo Cavallaro
Royal Holloway, University of London

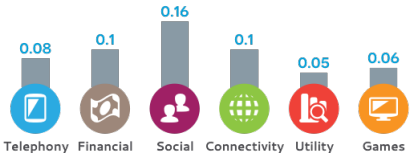
As the number of Android based devices continues to grow rapidly, malware is quickly permeating the Android ecosystem. Recent efforts at automatically detecting and classifying malware have focused on the static analysis of Android applications [1]. In principle, static analysis covers all paths in the code. However, in practice, it is thwarted by obfuscation, native and dynamically-loaded code, or modification of objects at runtime. In contrast, dynamic analysis involves running and observing the resulting behaviors and circumvents these shortcomings. It has been recently shown that dynamic analysis can be successfully used to classify Android malware [2]. Although this is useful for characterising family of threats, it is essential to design a framework that also identifies and precisely classifies zero-day Android malware—variants of existing samples or new, previously-unseen families.

The difficulty in dealing with zero-day samples is that it may not be possible to train on similar samples. Interestingly, however, this impediment can be turned into an advantage using statistical evaluation and used towards understanding whether the sample is a zero-day threat. In [3], a statistical assessment of a classification decision was used to understand how well a test sample fits into the training classes. Identification of zero-day threats is a related problem; it is necessary to establish whether the malware does not belong to any of the established classes that the classifier was trained on. In this work we show how statistical assessment of decisions made by a trained classifer can be used to identify previously unseen malware samples from unknown families. We present a framework which is based on statistical evaluation of Android samples to reliably classify zero-day malware. We describe the statistical overview of our approach in §I and discuss how it can be applied to detecting and classifying zero-day malware in §II.

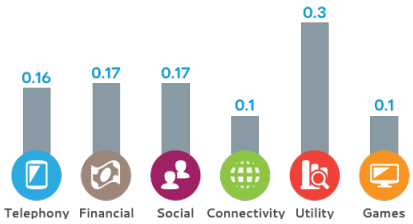
I. Statistical Assessment of Samples

In traditional classification, the algorithm often chooses a single category label per sample and ignores alternate choices available even though they may be viable. In areas such as malware detection, consequently, one is often forced to classify samples as either benign or malicious when they may actually not fit into either categories. In such cases, it is useful to assess how well a sample fits into a given set of categories before taking a classification decision.

To address these shortcomings, Conformal Evaluation (CE) has recently been suggested as a technique to statistically assess the quality of a machine learning algorithm and further understand how well a sample fits into all the available families [3]. Classification decisions are often made on the basis of distances between an object and a set of objects. CE converts these distances to *p-values*, statistical measures of (i) the algorithm under evaluation choices, and (ii) the distribution of data set across the label space. Unlike traditional techniques for classification that compute a probability of a sample belonging to a particular category, the advantage of using a statistical score is that these scores do not have to add up to one. Consequently, it is possible to point out when a sample does not belong to any class by checking if the p-values for all classes are low for the sample under consideration.



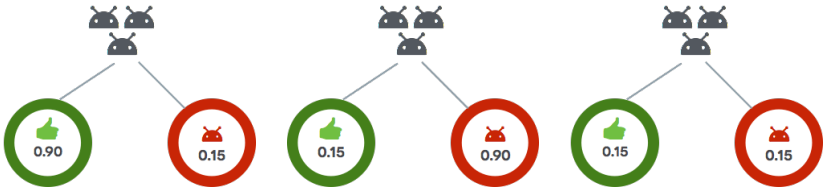
(a) P-values per class



(b) Probability of mapping to class

Fig. 1: A comparison of p-values and probability scores in assessment of how well a new sample belongs to a set of families. Probability scores need to add up to one so they may be skewed towards a class even if the sample does not belong to it.

Consider Figure 1 as an example. Here, the classifier has been trained on samples on six categories of applications and when a new sample comes along, both probability scores and p-values are computed for the six categories. We notice from the probability scores that the sample leans towards the *Utility* category. There are two possibilities: either the sample truly belongs to the *Utility* category or it artificially shows a propensity for the category because the probability scores for



(a) P-values pointing to benign (b) P-values pointing to malicious (c) Inconclusive p-values

Fig. 2: Assessment of category membership for samples using p-values.

other categories are poor and scores across all categories must sum up to one. It turns out that if p-values are used to assess the sample, we will notice that the p-value for all six categories is low meaning that the sample does not belong to any of the categories. Thus, p-values offer much better precision in assessing previously unseen categories of malware.

II. Assessment of Zero-Day malware

As a part of the framework to detect and classify zero-day malware based on statistical evaluation, we propose to train on known malware families and then test the trained model with new samples. Compared to a traditional train and test classification framework, we derive a geometric measure of the distance of the test sample from the families that we train on. This distance would then be used to derive p-values as summarised in section I.

Without loss of generality, we first describe how our approach can be applied to the 2-class problem where samples are either benign or malicious. In a typical deployment setting, we would derive p-values for a sample for both the benign and malicious classes as shown in Figure 2. We would classify the sample as belonging to a class X only if X is a clear winner, i.e., the sample has *high* p-value for class X. Note that the definition of *high* here is fully tunable. In practice, though, we intend to use the median p-value for all correct classifications into X during training. Ideally, we would expect *high* p-values for only one of the two classes. Depending on the quality of the training data and the novelty of the test sample, however, this may not be always true. If the p-values are high for both classes or low for both classes, we tag the classification as inconclusive and put the test sample into an *unknown* bucket. Once we reach a critical mass with *unknown* samples, our framework would cluster the uncategorised samples to see if there are families that we missed during training and/or redraw existing boundaries between families. Thus, the use of a statistical evaluation would lead to models of classification that evolve and get better over time.

An overview of how our framework would operate when deployed is shown in figure 3. In the first stage, we try to identify if a sample is malicious and in the second stage, we try to categorise the sample into a specific family if it is malicious. In both the steps, it is possible that the statistical evaluation yields an inconclusive result for the sample. If so, we put the sample into *unknown* buckets and cluster these buckets on a periodic basis. For the first stage of clustering, the number of

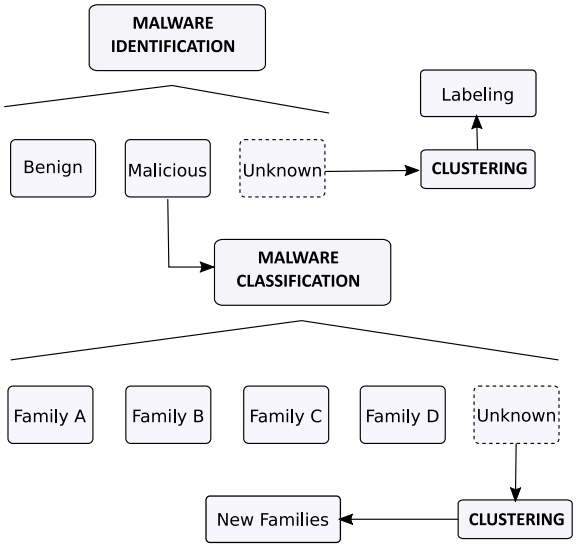


Fig. 3: A detection and family identification framework for Android malware based on statistical evaluation

classes in our framework is fixed i.e. benign and malicious. In this case, we have to run clustering on all samples to redraw the boundaries of benign and malicious samples and include the *unknown* samples into one of the two classes. In the second phase of the classification, the list of malware families is not exhaustive. In such a case, running clustering on the *unknown* bucket could potentially uncover new malware families.

Acknowledgments

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

References

[1] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, K. Rieck, and C. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket," *NDSS*, 2014.

[2] S. K. Dash, G. Suarez-Tangil, S. Khan, K. Tam, M. Ahmadi, J. Kinder, and L. Cavallaro, "Droidscribe: Classifying android malware based on runtime behavior," in *Mobile Security Technologies (MOST)*, 2016.

[3] R. Jordaney, Z. Wang, D. Papini, I. Nouretdinov, and L. Cavallaro, "Misleading metrics: On evaluating machine learning for malware with confidence," http://goo.gl/6YyrLK, Royal Holloway, University of London, Tech. Rep., 2016.