On the Lack of Consensus in Anti-Virus Decisions: Metrics and Insights on Building Ground Truths of Android Malware

Médéric Hurier, Kevin Allix, Tegawendé F. Bissyandé,
Jacques Klein, and Yves Le Traon

SnT - University of Luxembourg

Abstract. There is generally a lack of consensus in Antivirus (AV) engines' decisions on a given sample. This challenges the building of authoritative ground-truth datasets. Instead, researchers and practitioners may rely on unvalidated approaches to build their ground truth, e.g., by considering decisions from a selected set of Antivirus vendors or by setting up a threshold number of positive detections before classifying a sample. Both approaches are biased as they implicitly either decide on ranking AV products, or they consider that all AV decisions have equal weights. In this paper, we extensively investigate the lack of agreement among AV engines. To that end, we propose a set of metrics that quantitatively describe the different dimensions of this lack of consensus. We show how our metrics can bring important insights by using the detection results of 66 AV products on 2 million Android apps as a case study. Our analysis focuses not only on AV binary decision but also on the notoriously hard problem of *labels* that AVs associate with suspicious files, and allows to highlight biases hidden in the collection of a malware ground truth—a foundation stone of any malware detection approach.

1 Introduction

Malware is ubiquitous across popular software ecosystems. In the realm of mobile world, researchers and practitioners have revealed that Android devices are increasingly targeted by attackers. According to a 2015 Symantec Mobile Threat report [1], among 6.3 million Android apps analyzed, over 1 million have been flagged as malicious by Symantec in 2014 and classified in 277 Android malware families. To stop the proliferation of these malware, device owners and market maintainers can no longer rely on the manual inspection of security analysts. Indeed, analysts require to know beforehand all patterns of malicious behaviors so as to spot them in new apps. Instead, the research and practice of malware detection are now leaning towards machine learning techniques where algorithms can learn themselves to discriminate between malicious and benign apps after having observed features in an a-priori labelled set. It is thus obvious that the performance of the detector is tightly dependent on the quality of the training dataset. Previous works have even shown that the accuracy of such detectors can be degraded by orders of magnitude if the training data is faulty [2–6]. Following these findings, one can easily infer that it is also possible to artificially improve the performance of malware detectors by selecting a "ground truth" that splits around malware corner cases.

To build training datasets, Antivirus (AV) engines appear to be the most affordable means today. In particular, their use have become common thanks to online free services such as VirusTotal [7] that accepts the submission of any file for which it reports back the AV decisions from several vendors. Unfortunately, AV engines disagree regularly on samples. Their lack of consensus is actually observed in two dimensions: 1) their binary decisions on the maliciousness of a sample are often conflicting and 2) their labels are challenging to compare because of the lack of standard for naming malware samples.

To consolidate datasets as ground truth based on AV decisions, researchers often opt to use heuristics that they claim to be reasonable. For example, in the assessment of a state-of-the-art machine learning-based malware detection for Android [8], the authors have considered the reports from only 10 AV engines, selected based on their "popularity", dismissing all other reports. They further consider a sample to be malicious once two AV engines agree to say so. They claim that:

> "This procedure ensures that [their] data is (almost) correctly split into benign and malicious samples—even if one of the ten scanners falsely labels a benign application as malicious" [8, p. 7]

To gain some insights on the impact of such heuristics, we have built a dataset following these heuristics and another dataset following another common process in the literature [9], which considers all AV reports from VirusTotal and accepts a sample as malicious as long as any of the AV flags it as such. We compare the two datasets and find that the malware set in the first "ground truth" is reduced to only 6% of the malware set of the second "ground truth" dataset.

An in-depth study of different heuristics parameters can further reveal discrepancies in the construction of ground truth datasets, and thus further question any comparison of detectors performance. Similarly, the lack of consensus in label naming prevents a proper assessment of the performance of detectors across malware families.

In a recent work, Kantchellian et al. [10] have proposed weighting techniques towards deriving better, authoritative, ground truth based on AV labels. Our work is an in-depth investigation to further motivate this line of research by highlighting different facets of the problem. To that end, we propose metrics for quantifying various dimensions of comparison for AV decisions and labels. These metrics typically investigate to what extent decisions of a given AV are exclusive w.r.t other AVs, or the degree of genericity at which AV vendors assign malware labels.

Contributions: We make the following contributions:

- We extensively overview the lack of consensus in AV engines' decisions and labels. Our work is a call for new approaches to building authoritative ground truth datasets, in particular for the ever-growing field of machine learning-based malware detection.

– Building on a large dataset of thousands Android apps, we provide insights on the practice of building ground truth datasets based on VirusTotal AV decisions.
– We define metrics for quantifying the consensus (or lack thereof) among AV products following various dimensions. Based on the values of these metrics for extreme cases, they can be leveraged as good indicators for assessing a ground truth dataset. We further expect these metrics to be used as important information when describing experimental datasets for machine learning-based malware detection [1].

Findings: Among the findings of this study, we note that:

– AVs that flag many apps as malicious (i.e. AVs that seem to favor high Malware Recall) are more consensual than AVs that flag relatively few samples (i.e. AVs that seem to favor high Precision).
– Labels assigned to samples present a high level of genericity.
– Selecting a subset of AVs to build a ground truth dataset may lead to more disagreement in detection labels.

The remainder of this paper is presented as follows. Section 2 overviews related work which either inspired our work, or attempted to address the problem that we aim at quantifying. Section 3 presents the datasets that we have used for our study as well as the use cases we focus on. Section 4 presents our metrics and show-cases their importance. We discuss the interpretation of the metrics and their limitations in Section 5 before giving concluding remarks in Section 6.

2 Related Work

Our study relates to various work in the literature which have been interested in the collection of ground truth, in the automation of malware detection and those that have experimented with AV labels.

2.1 Security Assessment Datasets

Ground truth datasets are essential in the realm of security analysis. Indeed, on the one hand, analysts rely on them to manually draw patterns of malicious behaviors and devise techniques to prevent their damages. On the other hand, automated learning systems heavily rely on them to systematically learn features of malware. Unfortunately, these datasets are seldom fully qualified by the research community [11, 12]. This shortcoming is due to the rapid development of new malware [10] which forces the community to collect malware samples through generic techniques, which do not thoroughly validate the malicious behaviors [13].

A number of researchers have lately warned that flaws in security datasets are frequent [11] and can lead to false assumptions or erroneous results [3,10,14].

[1] We make available a full open source implementation under the name STASE at https://github.com/freaxmind/STASE

In their study, Rossow et al. [11] have analyzed the methodology of 36 papers related to malware research. Most notably, they observed that a majority of papers failed to provide sufficient descriptions of experimental setups and that 50% of experiments had training datasets with imbalanced family distributions. Related to this last point, Li et al. [14] raised a concern about such imbalances in clustering results. Using tools from a different domain (plagiarism detection), they were able to achieve results comparable to the state-of-the-art malware clustering algorithm at that time [15].

Nowadays, research in malware detection is often relying on AV engines to build ground truth datasets. Unfortunately, AVs often disagree, and AVs may even change their decision over time [16]. With our work, we aim to provide metrics that describe the underlying properties of experimental settings, focusing on ground truth collection, to transparently highlight biases and improve reproducibility.

2.2 Studies on Anti-Virus Decisions and Labels

Canto et al. [3] support that clear interpretations of malware alerts should be provided due to inconsistencies between antivirus engines. In view of these concerns, Rossow et al. [11] have also proposed a set of recommendations to design prudent experiments on malware. Kantchelian et al. [10] referred to Li et al. [14] study to point out that malware datasets obtained from a single source (e.g. antivirus vendors) could implicitly remove the most difficult cases. They thus propose supervised models to weight AV labels.

Another work related to malware experiments is AV-Meter by Mohaisen & Alrawi [5]. In their paper, the authors have described four metrics to assess the performance of antivirus scanners on a reference set of malwares. To our knowledge, this is the first attempt to formalize the comparison of security datasets. Their study also revealed that multiple antivirus are necessary to obtain complete and correct detections of malwares. Yet, AV-meter can not fully qualify datasets used in most common experiments. First, the metrics proposed by Mohaisen & Alrawi [5] are only applicable on ground-truth datasets where applications are known to expose malicious behaviors. In reality, this constraint can not be met due to the rising number of new malware samples which are created each year [10]. For instance, GData [17] experts identified more than 575 000 new malware samples between July and September 2015. This is an increase of 50% compared to the same period in 2014. Consequently, their study relied on a small dataset of 12 000 samples in order to ensure the correctness of their labels. In comparison, Arp et al. [8] performed a recent experiment on more than 130 000 samples. Finally, only four metrics were proposed by the authors, which may not describe all the characteristics necessary to avoid potential biases as mentioned in [3, 11, 14].

2.3 Experiments in Android ML-based Malware detection

Android malware has attracted a lot of attention from the research community [18–22], and a number of machine learning based approaches have been proposed recently [8, 23, 24]. State-of-the-art work, such as DREBIN [8] have even shown promising results. However, we observe that machine learning approaches have not been widely implemented in the malware detection industry. Sommer & Paxson [25] have presented multiple reasons which distinguish the security domain from other Computer Science areas, such as image recognition or natural language translation, where machine learning has been applied successfully. In previous work, we have shown how experimental scenarios can artificially improve the performance of detectors *in the lab* and make them unreliable on real-world settings [26, 27].

Our work here is about providing metrics to help researchers characterize their datasets and highlight their potential biases, as was recommended by Rossow et al. [11] and Sommer & Paxson [25].

3 Preliminaries

3.1 Dataset of Android Apps and Antivirus

Our study leverages a large dataset of 2 117 825 Android applications and their analysis reports by 66 antivirus engines hosted by VirusTotal.

App dataset: Our application samples have been obtained by crawling well-known app stores, including Google Play (70.33% of the dataset), Anzhi (17.35%) and AppChina (8.44%), as well as via direct downloads (e.g., Genome - 0.06%) [28].

AV reports: The AV reports were collected from VirusTotal[2], an online platform that can test files against commercial antivirus engines[3]. For each app package file (APK) sent to VirusTotal, the platform returns, among other information, two pieces of information for each antivirus:

- A binary flag (`True` = positive detection, `False` = negative detection)
- A string label to identify the threat (e.g. `Trojan:AndroidOS/GingerMaster.A`)

Overall, we managed to obtain AV reports for 2 063 674 Android apps[4]. In this study we explore those reports and define metrics to quantify the characteristics of several *tentative ground truths*.

3.2 Variations in Experimental Ground Truth Settings

When experimenting with machine learning-based malware detector, as it is nowadays common among security researchers, one of the very first steps is to build a ground truth, for training and also assessing the detector. The question is then how to derive a ground truth based on AV reports of the millions of apps in existence. In particular, we focus on which samples are considered as malicious

[2] `https://www.virustotal.com`

[3] Since the goal of this study is not to evaluate the individual performance of antivirus engines, their names have been omitted and replaced by an unique number (ID)

[4] we could not obtain the results for 54 151 (2.56%) applications because of a file size limit by VirusTotal

and included in the malware set of the ground truth. Based on methods seen in the literature, we consider the following three settings for building a ground truth:

Baseline settings: In these settings, we consider a straightforward process often used [9, 26] where a sample is considered malicious as long as any AV reports it with a positive detection. Thus, our ground truth with the Baseline settings and based on our 2 million apps, contains 689 209 "malware" apps. These samples are reported by AVs with 119 156 distinct labels.

Genome settings: In a few papers of the literature, researchers use for ground truth smaller datasets constituted of manually compiled and "verified" malicious samples. We consider such a case and propose such settings where the malware set of the ground truth is the Genome [29] dataset containing 1 248 apps. AV reports on these apps have yielded 7 101 distinct labels.

Filtered settings: Finally we consider a refined process in the literature where authors attempt to produce a clean ground truth dataset using heuristics. We follow the process used in a recent state-of-the-art work [8]:

1. Use a set of 10 popular AV scanners[5].
2. Select apps detected by at least two AVs in this set.
3. Remove apps whose label from any AV include the keyword "adware".

With these settings the malware set of the ground truth include 44 615 apps associated with 20 308 distinct labels.

In the remainder of this paper, we use $\mathcal{D}_{genome}$, $\mathcal{D}_{base}$, and $\mathcal{D}_{filtered}$ to refer to the three ground truth datasets. We did not performed supplementary preprocessings besides the heuristics we mentioned in the previous paragraph to avoid potential biases in our study.

3.3 Notations and Definitions

Given a set of n AV engines $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$ and a set of m apps $\mathcal{P} = \{p_1, p_2, \cdots, p_m\}$, we collect the binary decisions and string labels in two $n \times m$ matrices denoted $\mathcal{B}$ and $\mathcal{L}$ respectively:

$$\mathcal{B} = \begin{array}{cc} & \begin{array}{cccc} a_1 & a_2 & \ldots & a_n \end{array} \\ \begin{array}{c} p_1 \\ p_2 \\ \vdots \\ p_m \end{array} & \begin{pmatrix} b_{1,1} & b_{1,2} & \ldots & b_{1,n} \\ b_{2,1} & b_{2,2} & \ldots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m,1} & b_{m,2} & \ldots & b_{m,n} \end{pmatrix} \end{array} \mathcal{L} = \begin{array}{cc} & \begin{array}{cccc} a_1 & a_2 & \ldots & a_n \end{array} \\ \begin{array}{c} p_1 \\ p_2 \\ \vdots \\ p_m \end{array} & \begin{pmatrix} l_{1,1} & l_{1,2} & \ldots & l_{1,n} \\ l_{2,1} & l_{2,2} & \ldots & l_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m,1} & l_{m,2} & \ldots & l_{m,n} \end{pmatrix} \end{array}$$

where entry $b_{i,j}$ corresponds to the binary flag assigned by AV a_j to application p_i and entry $l_{i,j}$ corresponds to the string label assigned by AV a_j to application p_i. String label $l_{i,j}$ is $\emptyset$ (null or empty string) if the app p_i is not flagged by AV a_j. For any settings under study, a ground truth $\mathcal{D}$ will be characterized by both $\mathcal{B}$ and $\mathcal{L}$.

[5] AVs considered in [8]: AntiVir, AVG, Bit- Defender, ClamAV, ESET, F-Secure, Kaspersky, McAfee, Panda, Sophos

Let note $R_i = \{m_{i,1}, m_{i,2}, \cdots, m_{i,n}\}$ the i^{th} row vector of a matrix M, and $C_j = \{m_{1,j}, m_{2,j}, \cdots, m_{m,j}\}$ the j^{th} column. The label matrix $\mathcal{L}$ can also be vectorized as a column vector $\mathcal{L}' = (l_1, l_2, \cdots, l_k)$ which includes all distinct labels from matrix $\mathcal{L}$, excluding null values ($\emptyset$).

We also define six specific functions that will be reused through this paper:

- Let *positives* be the function which returns the number of positive detections from matrix $\mathcal{B}$, or the number of not null labels from matrix $\mathcal{L}$.
- Let *exclusives* be the function which returns the number of samples detected by only one AV in matrix $\mathcal{B}$.
- Let *distincts* be the function which returns the number of distinct labels (excluding $\emptyset$) in matrix $\mathcal{L}$.
- Let *freqmax* be the function which returns the number of occurrences of the most frequent label (excluding $\emptyset$) from matrix $\mathcal{L}$.
- Let *clusters* be the function which returns the number of applications which received a given label l_o with $l_o \in L'$.
- Let *Ouroboros* be the function which returns the minimum proportion of groups including 50% elements of the dataset, normalized between 0 and 1 [30]. This function is used to quantify the uniformity of a list of frequencies, independently of the size of the list.

4 Definition of Metrics and Experiments

In this section we consider the two pieces of information, AV decision and AV label, and perform analyses that investigate various aspects of the inconsistencies that may be present among AV reports. We then propose metrics to quantify these aspects and allow for comparison between different ground truth datasets.

4.1 Analysis of AV Decisions

The primary role of an AV engine is to decide whether a given sample should be considered as malicious [13]. These decisions have important consequences in production environments since a positive detection will probably trigger an alert and an investigation to mitigate a potential threat. False positives would thus lead to a waste of resources, while False negatives can have dire consequences such as substantial losses. AV engines must then select an adequate trade-off between a deterring high number of false positives and a damaging high number of false negatives.

In this section, we analyze the characteristics of AV decisions and their discrepancies when different engines are compared against each other.

4.1.1 Equiponderance

The first concern in using a set of AV engines is to quantify their detection accuracies. If there are extreme differences, the collected “ground truth” may be polluted by decisions from a few engines. In the absence of a significant golden set to compute accuracies, one can estimate, to some extent, the differences among AVs by quantifying their detection rates (i.e., number of positive decisions).

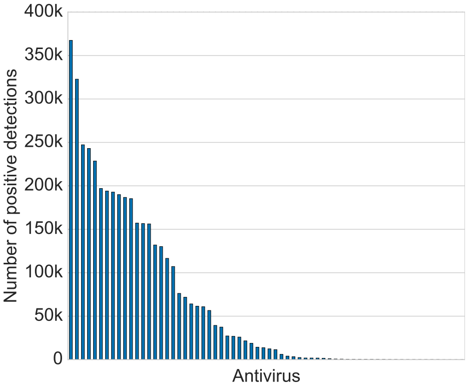

Fig. 1. AVs positive detections in $\mathcal{D}_{base}$

Fig. 1 highlights the uneven distribution of positive detections per AV in the $\mathcal{D}_{base}$ baseline ground truth. The number of detected apps indeed ranges from 0 to 367 435. This raises the question of the confidence in a "ground truth" when malicious samples can be contributed by AVs from the head and tail of the distribution. Indeed, although we cannot assume that AV engines with high (or low) detection rates have better performances, because of their potential false positives (or false negatives), it is important to consider the detection rates of AVs for a given dataset to allow comparisons on a common ground. A corollary concern is then to characterize the ground truth to allow comparisons. To generalize and quantify this characteristic of ground truth datasets, we consider the following research question:

RQ1: Given a set of AVs and the ground truth that they produce together, Is the resulting ground truth dominated by only a few AVs, or do all AVs contribute the same amount of information?

We answer this RQ with a single metric, *Equiponderance*, which measures how balanced—or how imbalanced—are the contributions of each AV. Considering our baseline settings with all AV engines, we infer that 9, i.e., 13.5%, AVs provided as many positive detections as all the other AVs combined. The *Equiponderance* aims to capture this percentage in its output. Because maximum value for this percentage is 50%[6], we weigh this percentage, by multiplying it by 2, to yield a metric between 0 and 1. We define the function *Ouroboros* [30] which computes this value and also returns the corresponding number of AVs, which we refer to as the Index of the *Equiponderance*.

$$Equiponderance(\mathcal{B}) = Ouroboros(X) \text{ with } X = \{positives(C_j) : C_j \in \mathcal{B}, 1 \leq j \leq n\}$$

- **Interpretation** – minimal proportion of antivirus that detected at least 50% applications in the dataset. The metric value is weighted.
- **Minimum**: 0 – when a single antivirus made all the positive detections
- **Maximum**: 1 – when the distribution of detection rates is perfectly even

When the *Equiponderance* is close to zero, the ground truth analyzed is dominated by the extreme cases: a large number of AV engines provide only a few positive detections, while only a few AVs engine provide most positive detections. In comparison with $\mathcal{D}_{base}$'s *Equiponderance* value of 0.27, $\mathcal{D}_{genome}$ and $\mathcal{D}_{filtered}$ present *Equiponderance* values of 0.48 and 0.59 respectively.

[6] If one set of AVs leads to a percentage x over 50%, then the other set relevant value is 100-x% < 50%.

4.1.2 Exclusivity

Even in the case where several AVs would have the same number of detections, it does not imply any agreement of AVs. It is thus important to also quantify to what extent each AV tends to detect samples that no other AV detects.

Fig. 2 plots, for every AV product, the proportion of exclusive detections (i.e., samples no other AV detects) over the total number of positive detection of this AV. Five AVs provide a majority of exclusive detections while a large part of other AVs (45) provides less than 10% such detections. For the 21 AVs that made the most positive detections, the proportion of exclusive detections remains below 16%, while the highest ratios of exclusive detections are associated with AVs that made a (relatively) small number of positive detections. Fig. 2 provides an important insight into Android malware detection by AVs: A very high absolute number of detections comes from adding more non-exclusive detections—not from detecting apps no other AV detects as could have been intuitively expected. The following research question aims at formally characterizing this bias in datasets:

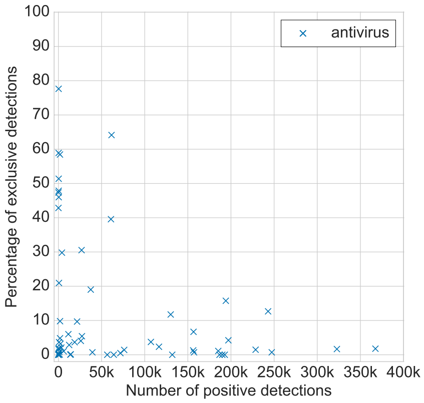

Fig. 2. Relation between positive and exclusive detections in $\mathcal{D}_{base}$

> *RQ2: Given a set of AVs and the ground truth that they produce together, what is the proportion of samples that were included only due to one AV engine?*

To answer this RQ, we propose the *Exclusivity* metric, which measures the proportion of a tentative ground truth that is specific to a single detector.

$$Exclusivity(\mathcal{B}) = \frac{exclusives(\mathcal{B})}{m}$$

- **Interpretation** – proportion of applications detected by only one antivirus
- **Minimum**: 0 – when every sample has been detected by more than one AV
- **Maximum**: 1 – when every sample has been detected by only one antivirus

In $\mathcal{D}_{base}$, 31% apps were detected exclusively by only one AV, leading to an *Exclusivity* value of 0.31. On the contrary, both $\mathcal{D}_{genome}$ and $\mathcal{D}_{filtered}$ do not include apps detected by only one AV and have an *Exclusivity* of 0.

4.1.3 Recognition

Because *Equiponderance* and *Exclusivity* alone are not sufficient to describe how experimental ground truth datasets are built, we investigate the impact of the threshold parameter that is often used in the literature of malware detection to consolidate the value of positive detections [8]. A threshold τ indicates that a sample is considered as a malware in the ground truth if and only if at least τ AV engines have reported positive detections on it. Unfortunately, to the best of our knowledge, there is no theory or golden rule behind the selection of τ. On one hand, it should be noted that samples rejected because of a threshold requirement may simply be either (a) new malware samples not yet recognized by all industry players, or (b) difficult cases of malware whose patterns are not easily spotted [10]. On the other hand, when a sample is detected by λ or γ AVs (where λ is close to τ and γ is much bigger than τ), the confidence of including the app in the malware set is not equivalent for both cases.

Fig. 3 explores the variations in the numbers of apps included in the ground truth dataset $\mathcal{D}_{base}$ as malware when the threshold value for detection rates (i.e., threshold number τ of AVs assigning a positive detection a sample) changes. The number of apps detected by more than τ AVs is also provided for the different values of τ.

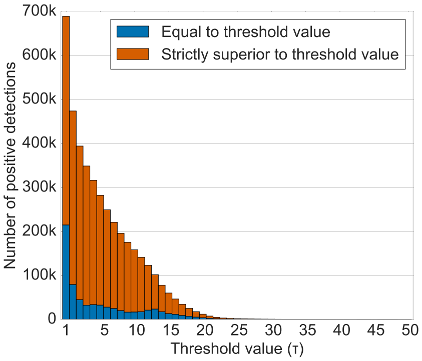

Fig. 3. Distribution of apps flagged by τ AVs in $\mathcal{D}_{base}$

Both bar plots appear to be right-skewed, with far more samples detected by a small number of antivirus than by the majority of them. Thus, any threshold value applied to this dataset would remove a large portion of the potential malware set (and, in some settings, shift them into the benign set). To quantify this property of ground truth datasets, we investigate the following research question:

> *RQ3: Given the result of antivirus scans on ground-truth dataset, have applications been marginally or widely recognized to be malicious ?*

We answer this RQ with a single metric, *Recognition*, which simply computes the average number of positive detections that are assigned to a sample. In other words, it estimates the number of AVs agreeing on a given app.

$$Recognition(\mathcal{B}) = \frac{\sum_{i=1}^{m} X_i}{n \times m} \text{ with } X = \{positives(R_i) : R_i \in \mathcal{B}, 1 \leq i \leq m\}$$

- **Interpretation** – proportion of antivirus which provided a positive detection to an application, averaging on the entire dataset
- **Minimum**: 0 – when no detections were provided at all
- **Maximum**: 1 – when each AV have agreement to flag all apps

When a threshold is applied on an experimental dataset, the desired objective is often to increase the confidence by ensuring that malware samples are widely recognized to be malicious by existing antivirus engines. Although researchers often report the effect on the dataset size, they do not measure the level of confidence that was reached. As an example, the *Recognition* of $\mathcal{D}_{base}$ is 0.09: on average, 6 (9%) AV engines provided positive detections per sample, suggesting a marginal recognition by AVs. The *Recognition* values for $\mathcal{D}_{filtered}$ and $\mathcal{D}_{genome}$ amounts to 0.36 and 0.48 respectively. These values characterize the datasets by estimating the extent to which AVs agree more to recognize samples from $\mathcal{D}_{filtered}$ as positive detections more widely than in $\mathcal{D}_{base}$. AVs recognize samples from $\mathcal{D}_{genome}$ even more widely.

4.1.4 Synchronicity

In complement to *Recognition* and *Exclusivity*, we investigate the scenarios where pairs of AV engines conflict in their detection decisions. Let us consider two AV engines U and V and the result of their detections on a fixed set of samples. For each sample, we can expect 4 cases:

	Detected by U	Not detected by U
Detected By V	(`True`, `True`)	(`True`, `False`)
Not detected by V	(`False`, `True`)	(`False`, `False`)

Even if the *Equiponderance* value of the dataset produced by AVs U and V amounts to 1, one cannot conclude on the distribution of those cases. The most extreme scenarios could be 50% (True, True) and 50% (False, False) or 50% (True, False) and 50% (False, True). For the first one, both AVs are in perfect synchrony while they are in perfect asynchrony in the second one.

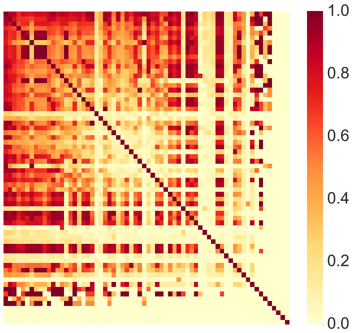

Fig. 4. Overlap between pairs of AVs in $\mathcal{D}_{base}$

Fig. 4 is a heatmap representation of the pairwise agreement among the 66 AV engines on our dataset. For simplicity, we have ordered the AV engines by their number of positive detections (the top row—left to right—and the left column—top to bottom—correspond to the same AVs). For each of the $\binom{66}{2}$ entries, we compute the *overlap* function [31]:

$$overlap(X, Y) = |X \cap Y| / min(|X|, |Y|)$$

This function normalizes the pairwise comparison with the case of the AV presenting the smallest number of positive detections. From the heatmap, we can observe two patterns: (a) The number of cells where a full similarity is achieved is relatively small w.r.t the number of entries. Only 12% of pairs of AVs achieved a pairwise similarity superior to 0.8, and only 1% of pairs presented a perfect similarity. (b) There is no continuity from the right to the left (nor from the top to the bottom) of the map. This indicates that AVs with comparable number of positive detections do not necessarily detect the same samples. We aim to quantify this level of agreement through the following research question:

RQ4: Given a dataset of samples and a set of AVs, what is the likelihood for any pair of distinct AV engines to agree on a given sample?

We answer this RQ with the *Synchronicity* metric which measures the tendency of a set of AVs to provide positive detections at the same time as other antivirus in the set:

$$Synchronicity(\mathcal{B}) = \frac{\sum_{j=1}^{n}\sum_{j'=1}^{n} PairwiseSimilarity(C_j, C_{j'})}{n(n-1)} \text{ with } j \neq j', C_j \in \mathcal{B}, C_{j'} \in \mathcal{B}$$

- **Interpretation** – average pairwise similarity between pairs of AVs
- **Minimum**: 0 – when no sample is detected at the same time by more than one AV
- **Maximum**: 1 – when each sample is detected by every AV
- **Parameters**
  - $PairwiseSimilarity$: a binary distance function [31]
    - Overlap: based on positive detections and normalized (default)
    - Jaccard: based on positive detections, but not normalized
    - Rand: based on positive and negative detections

High values of *Synchronicity* should be expected for datasets where no uncertainty remains to recognize applications as either malicious or not malicious. $\mathcal{D}_{base}$ presents a *Synchronicity* of 0.32, which is lower than values for $\mathcal{D}_{genome}$ (0.41), and $\mathcal{D}_{filtered}$ (0.75). The gap between values for $\mathcal{D}_{genome}$ and $\mathcal{D}_{filtered}$ suggests the impact that a selection of Antivirus can have on artificially increasing the *Synchronicity* of the dataset.

4.2 Analysis of Malware Labels

Besides binary decisions on detection of maliciousness in a sample, AV engines also provide, in case of positive detection, a string label which indicates the type/family/behavior of the malware or simply identifies the malicious trait. These labels are thus expected to specify appropriately the threat in a meaningful and consistent way. Nevertheless, previous work have found that the disagreement of multiple AVs on labelling a sample malware challenges their practical use [2–5].

In this section, we further investigate the inconsistencies of malware labels and quantify different dimensions of disagreements in "ground truth" settings.

4.2.1 Uniformity

Fig. 5 represents the distribution of the most frequently used labels on our $\mathcal{D}_{base}$ dataset. In total, the 689 209 samples detected by at least one AV were labeled with 119 156 distinct labels.

68% of positive detections were associated with the most infrequent labels, i.e., outside the top 20 labels (grouped together under the 'OTHERS' label). The most frequent label, `Android.Adware.Dowgin.I`, is associated with 9% of the positive detections. In a ground truth dataset, it is

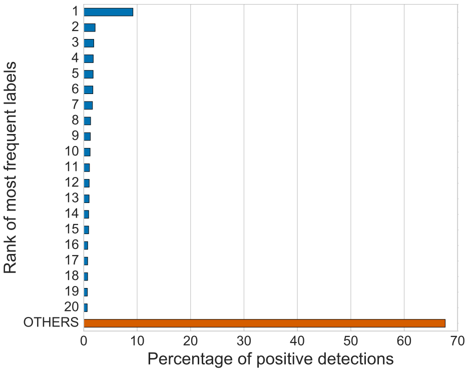

Fig. 5. Distribution of malware labels in $\mathcal{D}_{base}$

important to estimate the balance between different malicious traits, so as to ensure that the reported performance of an automated detector can generalize. We assess this property of ground truth by answering the following research question:

> *RQ5: Given a ground truth derived by leveraging a set of AVs, are the labels associated to samples evenly distributed?*

We answer this RQ with a single metric, *Uniformity*, which measures how balanced—or how imbalanced—are the clusters of samples associated to the different labels.

$$Uniformity(\mathcal{L}') = Ouroboros(X) \text{ with } X = \{clusters(l_k) : l_k \in \mathcal{L}', 1 \leq k \leq o\}$$

- **Interpretation** – minimal proportion of labels assigned to at least 50% of total number of detected samples. The metric value is weighted
- **Minimum**: 0 – when each sample is assigned a unique label by each AV
- **Maximum**: 1 – when the same label is assigned to every sample by all AVs

The *Uniformity* metric is important as it may hint on whether some malware families are undersampled w.r.t others in the ground truth. In can thus help, to some extent, to quantify potential biases due to malware family imbalance. $\mathcal{D}_{base}$ exhibits a *Uniformity* value close to 0 (12×10^{-4}) with an index of 75: 75 labels occur as often in the distribution than the rest of labels (119 081), leading to an uneven distribution. We also found extreme values for both Filtered and Genome settings with *Uniformity* of 0.01 and 0.04 respectively. These values raise the question of malware families imbalance in most ground truth datasets. However, it is possible that some labels, although distinct, because of the lack of naming standard, actually represent the same malware type. We thus propose to further examine labels on other dimensions.

4.2.2 Genericity

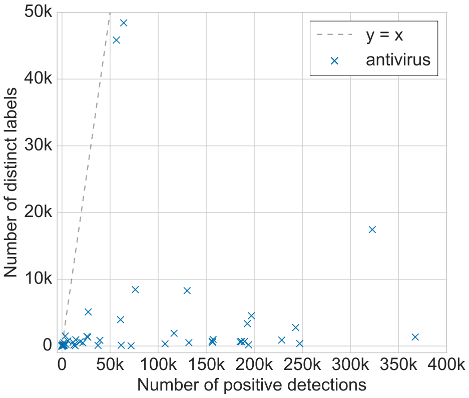

Fig. 6. Relation between distinct labels and positive detections per AV in $\mathcal{D}_{base}$

Once the distribution of labels has been extracted from the dataset, we can also measure how often labels are reused by antivirus. This property is an interesting behavior that Bureau & Harley highlighted [13]. If we consider the two extreme cases, AVs could either assign a different label to every sample (e.g. hash value), or a unique label to all samples. In both scenarios, labels would be of no value to group malware together [2].

In Figure 6, we plot the number of detections against the number of distinct labels for each AV. While two

AVs assign almost a different label for each detected sample (points close to the $y = x$ line), the majority of AVs have much fewer distinct labels than detected samples: they reuse labels amongst several samples. These two different behaviors might be explained by different levels of genericity of labels. For example, using very precise labels would make the sharing of labels among samples harder than in the case of very generic labels that could each be shared by several samples. To quantify this characteristic of labels produced by a set of AVs contributing to define a ground truth, we raise the following research question:

RQ6: Given a ground truth derived by leveraging a set of AVs, what is, on average for an AV, the degree of reuse of a label to characterize several samples?

We propose the *genericity* metric to quantify this information:

$$Genericity(\mathcal{L}) = 1 - \frac{o-1}{positives(\mathcal{L})-1} \text{ with } o \leftarrow \text{number of distinct labels}$$

- **Interpretation** – ratio between the number of distinct labels and the number of positive detections
- **Minimum**: 0 – when every assigned label is unique
- **Maximum**: 1 – when all labels are identical

Genericity assesses whether AVs assign precise labels or generic ones to samples. Although detectors with low *Genericity* would appear to be more precise in their naming, Bureau & Harley [13] support that such engines may not be the most appropriate w.r.t the exponential growth of malware variants. The *Genericity* $\mathcal{D}_{base}$ is 0.97, inline with our visual observation that there is far less distinct labels than positive detections. The *Genericity* values of $\mathcal{D}_{genome}$ and $\mathcal{D}_{filtered}$ are equal to 0.82 and 0.87 respectively.

4.2.3 Divergence

While *Uniformity* and *Genericity* can evaluate the overall distribution of labels that were assigned by AVs, they do not consider the question of agreement of AVs on each sample. Ideally, AVs should be consistent and provide labels similar to that of their peers. Even if this ideal case can not be achieved, the number of distinct labels per application should remain limited w.r.t the number of AVs agreeing to detect it.

For $\mathcal{D}_{base}$, Fig. 7 plots the relation between the number of positive detections of a sample and the average number of distinct labels associated to it. As

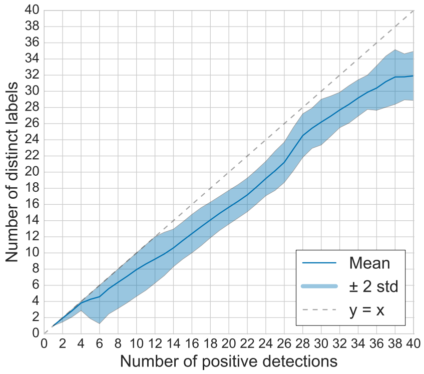

Fig. 7. Relation between distinct labels and positive detections per app in $\mathcal{D}_{base}$

a confidence margin, we also draw an area of two standard deviations centered on the mean. We note that the mean value for number of labels grows steadily with the number of detection, close to the maximum possible values represented by the dotted line. The Pearson correlation coefficient ρ between these variables evaluates to 0.98, indicating a strong correlation. Overall, the results suggest not only that there is a high number of different labels per application on our dataset, but also that this behavior is true for both small and high values of positive detections. The following research question investigates this characteristic of ground truth datasets:

> *RQ7: Given a set of AVs and the ground truth that they produce, to what extent do AVs provide for each sample a label that is inconsistent w.r.t. other AVs labels.*

We can quantify this factor with the following metric that measures the capacity of a set of antivirus to assign a high number of different labels per application.

$$Divergence(\mathcal{L}) = \frac{(\sum_{i=1}^{m} X_i) - n}{positives(\mathcal{L}) - n} \text{ with } X = \{distincts(R_i) : R_i \in \mathcal{L}, 1 \leq i \leq m\}$$

- **Interpretation**: – average proportion of distinct labels per application w.r.t the number of AVs providing positive detection flags
- **Minimum**: 0 – when AVs assign a single label to each application
- **Maximum**: 1 – when each AV assigns its own label to each application

Two conditions must be met in a ground truth dataset to reach a low *Divergence*: AVs must apply the same syntax consistently for each label, and they should refer to a common semantics when mapping labels with malicious behaviors/types. If label syntax is not consistent within the dataset, then the semantics cannot be assessed via the *Divergence* metric. It is, however, often possible to normalize labels through a basic preprocessing step.

The *Divergence* values of $\mathcal{D}_{base}$, $\mathcal{D}_{filtered}$ and $\mathcal{D}_{genome}$ are 0.77, 0.87 and 0.95 respectively. These results are counter-intuitive, since they suggest that more constrained settings create more disagreement among AVs in terms of labeling.

4.2.4 Consensuality

To complement the property highlighted by *Divergence*, we can look at the most frequent label assigned per application. Indeed, while the previous metric describes the number of distinct labels assigned per application, it does not measure the weight of each label, notably that of the most used label. Yet, to some extent, this label could be used to infer the family and the version of a malware, e.g., if it used by a significant portion of AVs to characterize a sample.

To visualize this information, still for $\mathcal{D}_{base}$, we create in Fig. 8 a plot similar to that of Fig. 7, looking now at the average number of occurrence of the Most Frequent Label (MFL) against the number of positive detections per application.

The correlation coefficient ρ between the two variables is 0.76, indicative of a correlation. Nevertheless, the relation is close to the potential minimum (x-axis). This is in line with our previous observations on $\mathcal{D}_{base}$ that the number of distinct labels per application was high. The plot further highlights that the most frequent label for an application is assigned simultaneously by one to six AVs (out of 66) on average. This finding suggests that, at least in $\mathcal{D}_{base}$, using the most frequent label to characterize the malicious sample is not a sound approximation. The following research question generalize the dimension of disagreement that we investigate:

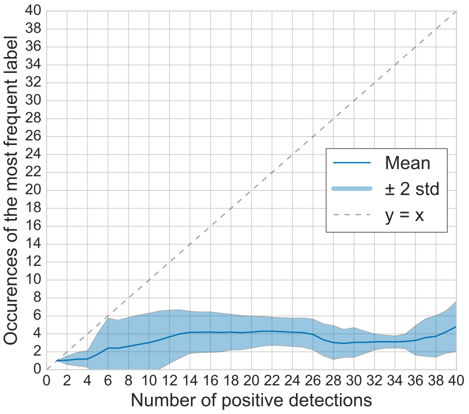

Fig. 8. Relation between MFL/τ and positive detections per app in $\mathcal{D}_{base}$

> *RQ8: Given a set AVs and the ground truth that they produce, to what extent can we rely on the most frequently assigned label for each detected sample as an authoritative label?*

We answer this RQ with the *Consensuality* metric:

$$Consensuality(\mathcal{L}) = \frac{(\sum_{i=1}^{m} X_i) - n}{positives(\mathcal{L}) - n} \text{ with } X = \{freqmax(R_i) : R_i \in \mathcal{L}, 1 \leq i \leq m\}$$

- **Interpretation** – average proportion of AVs that agree to assign the most frequent label. The frequency is computed per sample.
- **Minimum**: 0 – when each AV assigns to each detected sample its own label (i.e., unused by others on this sample)
- **Maximum**: 1 - when all AVs assign the same label to each sample. Different samples can have different labels however

A high *Consensuality* value highlights that the used AVs agree on most applications to assign a most frequent label. This metric is important for validating, to some extent, the opportunity to summarize multiple labels into a single one. In the $\mathcal{D}_{base}$ set, 79% detection reports by AVs do not come with a label that, for each sample, corresponds to the most frequent label on the sample. The *Consensuality* value of the set evaluates to 0.21. In comparison, the *Consensuality* values for $\mathcal{D}_{filtered}$ and $\mathcal{D}_{genome}$ are 0.05 and 0.06 respectively.

4.2.5 Resemblance

Divergence and *Consensuality* values on $\mathcal{D}_{base}$ suggest that labels assigned to samples cannot be used directly to represent malware families. Indeed, the number of distinct labels per application is high (high *Divergence*), and the most frequent label per application does not occur often (low *Consensuality*). We further investigate these disagreements in labels to verify whether the differences between label strings are small or large across AVs. Indeed, in previous comparison, given the lack of standard naming, we have chosen to compute exact matching. Thus, minor variations in label strings may have widely influenced our metric values. We thus compute the similarity between label strings for each application and present the summary in Fig.9. For each detected sample, we computed the Jaro-Winkler [32] similarity between pairwise combinations of labels provided by AVs. This distance metric builds on the same intuition as the edit-disance (i.e., Levenshtein distance), but is directly normalized betwen 0 and 1. A similarity value of 1 implies the identicality of strings while a value of 0 is indicative of high difference. We consider the minimum, mean and maximum of these similarity values and represent their distributions across all apps. The median of mean similarity values is around 0.6: on average labels only slightly resemble each other. The following research question highlights the consensus that we attempt to measure:

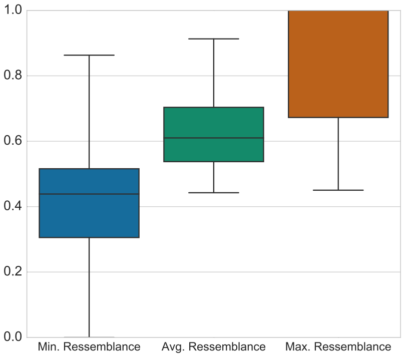

Fig. 9. String similarity between labels per app in $\mathcal{D}_{base}$

> *RQ9: Given a set AVs and the ground truth that they produce, how resembling are the labels assigned by AVs for each detected sample?*

We answer this metric with the *Resemblance* metric which measures the average similarity between labels assigned by set of AVs to a given detected sample.

$$Ressemblance(\mathcal{L}) = \frac{1}{m}\sum_{i=1}^{m}\frac{\sum_{j=1}^{n'_i}\sum_{j'=1}^{n'_i} Jaro-Winkler(l_{i,j}, l_{i,j'})}{n'_i(n'_i-1)}$$

with $j \neq j', l_{i,j} \neq \emptyset, l_{i,j'} \neq \emptyset, l_{i,j} \in \mathcal{B}, l_{i,j'} \in \mathcal{B}$ and $n'_i = positives(R_i), 2 \leq n'_i \leq n$

- **Interpretation** estimation of the global resemblance between labels for each app
- **Minimum** 0 when there is no similitude between labels of an application
- **Maximum** 1 when labels are identical per application

Resemblance assesses how labels assigned to a given application would be actually similar across the considered AVs. This metric, which is necessary when *Divergence* is high and *Consensuality* is low, can evaluate if the differences between label strings per application are small or large. $\mathcal{D}_{base}$, $\mathcal{D}_{filtered}$ and $\mathcal{D}_{genome}$ present *Resemblance* values of 0.63, 0.57 and 0.60 respectively. Combined with the *Divergence* metric values, we note that reducing the set of AVs has not yielded datasets where AVs agree more on the labels.

5 Discussions

5.1 Comparison of Ground-Truth Approaches

Table 1 summarizes the metric values for the three settings described in Section 3.3 that researchers may use to build ground truth datasets.

Table 1. Summary of Metrics for three common settings of Ground Truth constructions

	Equiponderance	Exclusivity	Recognition	Synchronicity	Uniformity	Genericity	Divergence	Consensuality	Resemblance
$\mathcal{D}_{base}$	0.27	0.31	0.09	0.32	0.001	0.97	0.77	0.21	0.63
$\mathcal{D}_{filtered}$	0.59	0	0.36	0.75	0.01	0.87	0.95	0.05	0.57
$\mathcal{D}_{genome}$	0.48	0	0.48	0.41	0.04	0.82	0.87	0.06	0.60

The higher values of *Recognition* and *Synchronicity* for $\mathcal{D}_{genome}$ and $\mathcal{D}_{filtered}$ in comparison with $\mathcal{D}_{base}$ suggest that these datasets were built with samples that are well known to be malicious in the industry. If we consider that higher *Recognition* and *Synchronicity* values provide guarantees for more reliable ground truth, then $\mathcal{D}_{genome}$ and $\mathcal{D}_{filtered}$ are better ground truth candidates than $\mathcal{D}_{base}$. Their lower value of *Genericity* also suggests that AV labels provided are more precise than that in $\mathcal{D}_{base}$. At the same time, higher values of *Equiponderance* and *Uniformity* imply that both AV detections and labels are more balanced across AVs.

Divergence and *Consensuality* values however suggest that the general agreement on AV labels has diminished in $\mathcal{D}_{genome}$ and $\mathcal{D}_{filtered}$ in comparison with $\mathcal{D}_{base}$. The *Exclusivity* value of 0 for $\mathcal{D}_{genome}$ and $\mathcal{D}_{filtered}$ further highlights that the constraints put on building those datasets may have eliminated corner cases of malware that only a few, if not 1, AV could have been able to spot.

We also note that $\mathcal{D}_{filtered}$ has a higher *Synchronicity* value than $\mathcal{D}_{genome}$, indicating that its settings lead to a selection of AVs which were more in agreement on their decision. In contrast, the *Divergence* values indicate that the proportion of distinct labels for each sample was higher in $\mathcal{D}_{filtered}$ than in $\mathcal{D}_{genome}$, suggesting that decisions in $\mathcal{D}_{genome}$ are easier to interpret for each sample. Nevertheless, the classification of samples in malware families would be more difficult because of the higher proportion of distinct labels to take into consideration.

5.2 Limitations and Future work

The collection of metrics proposed in this paper is focused on the quantification of nine characteristics that we considered relevant based on our experience and the literature related to malware experiments [3,10,11,13]. Hence, we do not attempt to cover the full range of information that could be quantified from the output of AV scans. In addition, our analysis of antivirus reports has exposed a global lack of consensus that has been previously highlighted by other authors for other computing platforms [2,4,13,33]. Our work cannot be used to solve the challenge of naming inconsistencies directly. Instead, the metrics we presented can be used to evaluate ground truth datasets prior and posterior to their transformation by techniques proposed by other authors [6, 10, 34].

As future work, we will focus on surveying parameter values to yield ground truths that are suitable to practionners' constraints for consensus and reliability in accordance to their use cases.

6 Conclusion

We have investigated the lack of consensus in AV decisions and labels using the case study of Android samples. Based on different metrics, we assessed the discrepancies between three ground truth datasets, independently of their size, and question their reliability for evaluating the performance of a malware detector. The objective of our work was twofold: (1) to further motivate research on aggregating AV decisions results and improving the selection of AV labels; (2) to provide means to researchers to qualify their ground truth datasets, w.r.t AVs and their heuristics, so as to increase confidence in performance assessment, and take a step further to improve reproducibility of experimental settings, given the limited sharing of of security data such as samples.

Acknowledgment

This work was supported by the Fonds National de la Recherche (FNR), Luxembourg, under the project AndroMap C13/IS/5921289.

References

1. Symantec: Symantec. istr 20 - internet security threat report (Apr. 2015) http://know.symantec.com/LP=1123.
2. Bailey, M., Oberheide, J., Andersen, J., Mao, Z.M., Jahanian, F., Nazario, J.: Automated classification and analysis of internet malware. Recent Advances in Intrusion Detection **4637/2007** (2007) 178–197
3. Canto, J., Sistemas, H., Dacier, M., Kirda, E., Leita, C.: Large scale malware collection: lessons learned. 27th International Symposium on Reliable Distributed Systems. **52**(1) (2008) 35–44
4. Maggi, F., Bellini, A., Salvaneschi, G., Zanero, S.: Finding non-trivial malware naming inconsistencies. Proceedings of the 7th ICISS (2011) 144–159
5. Mohaisen, A., Alrawi, O.: Av-meter: An evaluation of antivirus scans and labels. DIMVA'14 **8550 LNCS** (2014) 112–131
6. Perdisci, R., U, M.: Vamo: towards a fully automated malware clustering validity analysis. Annual Computer Security Applications Conference (2012) 329
7. VirusTotal: VirusTotal about page. `https://www.virustotal.com/en/about/`
8. Arp, D., Spreitzenbarth, M., Malte, H., Gascon, H., Rieck, K.: Drebin: Effective and explainable detection of android malware in your pocket. Symposium on Network and Distributed System Security (NDSS) (2014) 23–26
9. Yang, C., Xu, Z., Gu, G., Yegneswaran, V., Porras, P.: DroidMiner: Automated Mining and Characterization of Fine-grained Malicious Behaviors in Android Applications. In: ESORICS 2014. (2014) 163–182
10. Kantchelian, A., Tschantz, M.C., Afroz, S., Miller, B., Shankar, V., Bachwani, R., Joseph, A.D., Tygar, J.D.: Better malware ground truth: Techniques for weighting anti-virus vendor labels. AISec '15, ACM (2015) 45–56
11. Rossow, C., Dietrich, C.J., Grier, C., Kreibich, C., Paxson, V., Pohlmann, N., Bos, H., Van Steen, M.: Prudent practices for designing malware experiments: Status quo and outlook. Proceedings of S&P (2012) 65–79
12. Allix, K., Jérome, Q., Bissyandé, T.F., Klein, J., State, R., Le Traon, Y.: A forensic analysis of android malware–how is malware written and how it could be detected? In: COMPSAC'14, IEEE (2014) 384–393
13. Bureau, P.M., Harley, D.: A dose by any other name. In: Virus Bulletin Conference, VB. Volume 8. (2008) 224–231
14. Li, P., Liu, L., Gao, D., Reiter, M.K.: On Challenges in Evaluating Malware Clustering. In: Recent Advances in Intrusion Detection: 13th International Symposium, RAID 2010. Proceedings. Springer Berlin Heidelberg (2010) 238–255

15. Bayer, U., Comparetti, P.M., Hlauschek, C., Kruegel, C., Kirda, E.: Scalable, behavior-based malware clustering. In: Proceedings of the 16th Annual Network and Distributed System Security Symposium (NDSS 2009). (1 2009)
16. Gashi, I., Sobesto, B., Mason, S., Stankovic, V., Cukier, M.: A study of the relationship between antivirus regressions and label changes. In: ISSRE. (Nov 2013)
17. GData: Mobile malware report (Q3 2015) https://secure.gd/dl-en-mmwr201503.
18. Enck, W., Ongtang, M., McDaniel, P.: On lightweight mobile phone application certification. Proceedings of the 16th ACM conference on Computer and communications security - CCS '09 (2009) 235–245
19. Enck, W., Octeau, D., McDaniel, P., Chaudhuri, S.: A study of android application security. In: Proceedings of the 20th USENIX Security. (2011) 21
20. Felt, A.P., Chin, E., Hanna, S., Song, D., Wagner, D.: Android permissions demystified. In: Proceedings of the 18th ACM conference on Computer and communications security. CCS '11, New York, NY, USA, ACM (2011) 627–638
21. Yan, L., Yin, H.: Droidscope: seamlessly reconstructing the os and dalvik semantic views for dynamic android malware analysis. Proceedings of the 21st USENIX Security Symposium (2012) 29
22. Zhou, Y., Wang, Z., Zhou, W., Jiang, X.: Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets. Proceedings of the 19th Annual Network and Distributed System Security Symposium (2) (2012) 5–8
23. Barrera, D., Kayacik, H.G.ü.b., van Oorschot, P.C., Somayaji, A.: A methodology for empirical analysis of permission-based security models and its application to android. Proceedings of the 17th ACM CCS (1) (2010) 73–84
24. Peng, H., Gates, C., Sarma, B., Li, N., Qi, Y., Potharaju, R., Nita-Rotaru, C., Molloy, I.: Using probabilistic generative models for ranking risks of android apps. In: Proceedings of the 2012 ACM CCS, ACM (2012) 241–252
25. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: Proceedings of the 2010 IEEE S&P. 305–316
26. Allix, K., Bissyandé, T.F., Jérome, Q., Klein, J., State, R., Le Traon, Y.: Empirical assessment of machine learning-based malware detectors for android. Empirical Software Engineering (2014)
27. Allix, K., Bissyandé, T., Klein, J., Le Traon, Y.: Are your training datasets yet relevant? an investigation into the importance of timeline in machine learning-based malware detection. In: ESSOS'15. Volume 8978 of LNCS. (2015) 51–67
28. Allix, K., Bissyandé, T.F., Klein, J., Le Traon, Y.: Androzoo: Collecting millions of android apps for the research community. In: MSR'16. (2016)
29. Zhou, Y., Jiang, X.: Dissecting android malware: Characterization and evolution. In: Proceedings of the 2012 IEEE S&P, IEEE Computer Society (2012) 95–109
30. Hurier, M.: Definition of ouroboros https://github.com/freaxmind/ouroboros.
31. Pfitzner, D., Leibbrandt, R., Powers, D.: Characterization and evaluation of similarity measures for pairs of clusterings. Knowledge and Information Systems **19**(3) (2009) 361–394
32. Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A comparison of string metrics for matching names and records. KDD Workshop on Data Cleaning and Object Consolidation **3** (2003)
33. Harley, D.: The game of the name malware naming, shape shifters and sympathetic magic. In: CEET 3rd Intl. Conf. on Cybercrime Forensics Education & Training, San Diego, CA. (2009)
34. Wang, T., Meng, S., Gao, W., Hu, X.: Rebuilding the tower of babel: Towards cross-system malware information sharing. In: Proceedings of the 23rd ACM CIKM. (2014) 1239–1248