A New Method for Inferring Ground-Truth Labels and Malware Detector Effectiveness Metrics

John Charlton[1], Pang Du[2], and Shouhuai Xu[3](✉)

[1] Department of Computer Science, University of Texas at San Antonio, San Antonio, USA
John.E.Charlton@gmail.com
[2] Department of Statistics, Virginia Tech, Blacksburg, USA
[3] Department of Computer Science, University of Colorado Colorado Springs, Colorado Springs, USA
sxu@uccs.edu

Abstract. In the context of malware detection, ground-truth labels of files are often difficult or costly to obtain; as a consequence, malware detector effectiveness metrics (e.g., false-positive and false-negative rates) are hard to measure. The unavailability of ground-truth labels also hinder the training of machine learning based malware detectors. These issues are often encountered by researchers and practitioners and force them to use various heuristics without justification. Therefore, seeking principled methods has become an important open problem. In this paper, we present a principled method for tackling the problem.

Keywords: Malware detection · Security metrics · Security measurement · Inference.

1 Introduction

Cybersecurity metrics is one of the most notoriously difficult open problems, despite the numerous efforts that have been made by the research community (see, e.g., [2–6,8,14,17,18,21–23,27]). At a high level, security metrics research can be divided into two categories: (i) *defining metrics* to measure what needs to be measured, which is still largely open [6,18]; (ii) *designing procedures* to measure what needs to be measured, namely the measurement of well-defined security metrics. This paper falls into the latter category (ii), by considering malware detection and the measurement of *ground-truth* labels for files and malware detector effectiveness metrics, such as false-positive and false-negative rates. This is an important problem because ground-truth labels of files (e.g., malicious or benign) are often assumed to be given and then leveraged for measuring malware detector effectiveness or training malware detectors.

Unfortunately, ground-truth labels are often difficult or costly to obtain. For a small set of files, we may use human experts to provide labels. However, perfect labels cannot be guaranteed because humans are error-prone and can make mistakes. For a large set of files, it is unrealistic for human experts to label them. Given this, one may resort to some third-party services. However, third-party service providers often leave the problem to

W. Lu et al. (Eds.): SciSec 2021, LNCS 13005, pp. 77–92, 2021.
https://doi.org/10.1007/978-3-030-89137-4_6

the end users. A typical scenario that is often encountered in the real world is the following: Third-party service providers, such as VirusTotal, provide labels of files given by a number of malware detectors, but these labels are in conflict with each other (i.e., a file is labeled by some detectors as malicious, but benign by others) [1,7,12,15,16,19,28]. This phenomenon is widely exhibited in the cybersecurity domain, including blacklists of malicious websites [10,13,20,24–26].

Without support of principled solutions, the challenge has forced researchers and practitioners to use various heuristics. Two popular heuristics are: Given a set of malware detectors, a file is treated as malicious as long as a certain threshold *number* vs. *fraction* (e.g., majority) of them label it as malicious [15,16,19]. These heuristics are troublesome because each malware detector has different capabilities in detecting malware. This calls for principled solutions and has motivated a few efforts.

Kantchelian et al. [12] use a *Bayesian* method to infer the ground-truth labels of files and malware detector effectiveness metrics, by making four *assumptions*: (i) a detector has an equal chance in mislabeling any benign file as malicious and an equal chance in mislabeling any malicious file as benign, which may not hold because a detector can be better at recognizing one kind of malware than another; (ii) detectors label files in an independent fashion, which may not hold when they use some common techniques; (iii) the percentage of malicious files is about 50%; (iv) detectors incur low false-positives and high false-negatives. The preceding (iii) and (iv) are associated with the prior distributions that are inherent to the Bayesian approach. Du et al. [7] use a *frequentist* approach to design statistical estimators to measure malware detectors' effectiveness metrics and the fraction of malicious files in a given set of files (but *not* the ground-truth labels), while only making the preceding assumptions (i) and (ii).

Charlton et al. [1] propose a new approach to measuring detectors' *relative accuracy* in the absence of ground-truth labels. The relative accuracy is measured with an *ordinal scale* [18], meaning that it only tells which detector is more accurate than which other detector, but not how much more accurate (i.e., detectors' absolute accuracy or effectiveness metrics remains unaddressed, so do the ground-truth labels). While heuristic in nature, this approach is attractive because it does *not* need any of the preceding assumptions (i)–(iv). This motivates us to explore how to turn the heuristic into a principled method for inferring ground-truth labels and detector effectiveness metrics.

Our Contributions. Our core contribution is to show that the *relative accuracy* approach [1] can be turned into a principled *weighted* majority voting method for inferring ground-truth labels and detector effectiveness metrics. The key idea is to treat an *enhanced* version of relative accuracy as a detector's voting weight, where the enhancement comes from the introduction of a *bellwether* detector whose effectiveness is known because it labels any file uniformly at random and independent of anything else. In the case of binary classification, it labels each file as malicious (benign) with a 0.5 probability, independent of anything else; this means that the bellwether detector has an *expected* true-positive rate, true-negative rate, and accuracy of 0.5 when the number of files is large enough. Intuitively, the bellwether detector serves as a *reference point* in the ordinal scale of relative accuracy because we know both its relative accuracy and absolute accuracy (i.e., 0.5 for the latter). This reference point offers a better scaling of relative accuracy than the counterpart in its absence. This improved scaling brings detectors' relative accuracy closer to their absolute accuracy, respectively. The weighted majority voting method, which is actually an iterative process, leads to the *inferred* ground-truth labels, which allows us to infer malware detector effectiveness metrics.

In order to understand *why* the proposed method works, we give an *algebraic* interpretation of relative accuracy (with or without employing the bellwether detector), by making connections to the well-known Principal Component Analysis (PCA). We show that the *similarity matrix* for describing the similarity between the labels given by malware detectors plays a role similar to that of the *correlation matrix* in PCA. A similarity matrix can be decomposed to a set of eigenvectors and eigenvalues; the eigenvectors represent the unit vectors describing an n-dimensional space with n being the number of detectors, and the eigenvalues provide a measurement of magnitude, or importance, associated with these vectors respectively. The larger the eigenvalue, the higher the importance of the detectors corresponding to it; the detectors corresponding to the higher ordered eigenvectors have higher relative accuracy and can be deemed as more trusted and given larger weights. In other words, detectors corresponding to the higher ordered eigenvectors and eigenvalues provide a more pivotal decision mathematically than the lower ordered ones. Because of this algebraic interpretation, we deem the relative accuracy approach an *algebraic* one, which is in contrast to *statistical* approaches [7, 12].

We validate the principled method (Algorithm 4) via synthetic data with known ground-truth labels and detector effectiveness metrics. Experimental results show it can accurately infer ground-truth labels and detector effectiveness metrics, especially so when eliminating the "poor" detectors with bellwether relative accuracy smaller than that of the bellwether detector. Then, we apply the method to analyze a real-world dataset, for which neither ground-truth labels nor detector effectiveness is known. We find that among other things, both the strategy of trading low false-positive for low false-negative and its opposite are widely employed by real-world malware detectors.

Related Work. This paper falls into the field of security metrics research. For the state-of-the-art security metrics research in a broader context, we refer to [2–6, 8, 14, 17, 18, 21–23, 27]. This paper deals with the measurement of well-defined metrics in the particular context of malware detection, namely: Given a number of malware detectors that have labeled a number of files, how can we infer the ground-truth labels of the files (i.e., malicious vs. benign) and malware detector effectiveness metrics? To tackle the problem, there are a number of heuristics [15, 16, 19, 28], but few principled solutions [7, 12] which make a number of assumptions as mentioned above. This study is inspired by the heuristic relative accuracy approach by [1], by making a solid step in turning the heuristic into a principled weighted majority voting method.

Paper Outline. Section 2 presents the problem. Section 3 revisits the notion of relative accuracy. Section 4 describes the methods used to tackle the problem. Section 5 applies the method to a real-world dataset. Section 6 concludes the paper with future directions.

2 Problem Statement

Suppose we are given a matrix $\mathbf{V} = (V_{ij})_{1\le i\le n, 1\le j\le m}$, where m is the number of files, n is the number of malware detectors denoted by $D_1, \ldots, D_n$, and

$$V_{ij} = \begin{cases} 1 & \text{if detector } D_i \text{ labels file } j \text{ as malicious} \\ 0 & \text{if detector } D_i \text{ labels file } j \text{ as benign} \\ -1 & \text{if detector } D_i \text{ does not label file } j \text{ at all (which can happen in practice).} \end{cases}$$

The problem is to infer (i) the ground-truth labels of the files (i.e., malicious vs. benign) and (ii) the detectors' effectiveness metrics in terms of the widely-used true-positive rate (TPR), true-negative rate (TNR) and accuracy (ACC) [18]. Let TP_i, TN_i, FP_i and FN_i respectively denote the number of true-positives, true-negatives, false-positives, and false-negatives associated with detector D_i. Recall that $\mathsf{TPR}_i = \mathsf{TP}_i/(\mathsf{TP}_i + \mathsf{FN}_i)$, $\mathsf{TNR}_i = \mathsf{TN}_i/(\mathsf{TN}_i + \mathsf{FP}_i)$ and $\mathsf{ACC}_i = (\mathsf{TP}_i + \mathsf{TN}_i)/(\mathsf{TP}_i + \mathsf{TN}_i + \mathsf{FP}_i + \mathsf{FN}_i)$ [18].

3 Relative Accuracy Revisited

A *weaker* variant of the preceding problem is to infer if D_i is *more accurate* than D_j. This variant is weaker because a solution to this problem is not guaranteed to solve the preceding problem. Nevertheless, this weaker problem leads to the notion of *relative accuracy*, or RA, on an *ordinal* scale, which is a discrete ordered set that permits comparisons between two measurements [18]. Let RA_i denote inferred relative accuracy of D_i. It is empirically shown $RA_i > RA_k$ means D_i is *more accurate* than D_k and $RA_i \neq \mathsf{ACC}_i$ [1].

3.1 Review of Previous Approach [1] to Computing Relative Accuracy

The known approach to computing relative accuracy is based on the following concepts with respect to matrix $\mathbf{V}$ [1]. In order to deal with the fact that some files may not be labelled by every detector, a *count matrix*, denoted by $\mathbf{C} = (C_{ik})_{1\le i,k\le n}$, is defined to describe the number of files that are labelled by a pair of detectors D_i and D_k as follows:

$$C_{ik} = C_{ki} = \sum_{\ell=1}^{m} \begin{cases} 1 & \text{if } V_{i\ell} \neq -1 \wedge V_{k\ell} \neq -1, \\ 0 & \text{if } V_{i\ell} = -1 \vee V_{k\ell} = -1. \end{cases}$$

In order to measure the agreement between the labels given by a pair of detector, an *agreement matrix*, denoted by $\mathbf{A} = (A_{ik})_{1\le i,k\le n}$, is defined to describe the number of files that are given the same label by detectors D_i and D_k as follows:

$$A_{ik} = A_{ki} = \sum_{\ell=1}^{m} \begin{cases} 1 & \text{if } V_{i\ell} = V_{k\ell} \wedge V_{i\ell} \neq -1 \wedge V_{k\ell} \neq -1 \\ 0 & \text{if } V_{i\ell} \neq V_{k\ell} \vee V_{i\ell} = -1 \vee V_{k\ell} = -1. \end{cases}$$

In order to measure the similarity between two detectors, a *similarity matrix*, denoted by $\mathbf{S} = (S_{ik})_{1\le i,k\le n}$ where $S_{ik} = A_{ik}/C_{ik}$, is defined to describe the degree of agreement between detectors D_i and D_k. Note that $\mathbf{S}$ is a real, symmetric matrix. These lead to:

Algorithm 1. Known approach to computing relative accuracy [1]

```
Input: similarity matrix $\mathbf{S} = (S_{ik})_{1 \leq i,k \leq n}$; tolerable error threshold $\epsilon$
Output: relative accuracy vector $\mathbf{RA} = (RA_1, \ldots, RA_n)^\mathsf{T}$
1: $\delta \leftarrow 2\epsilon$ {It suffice to initialize $\delta$ as any value that is greater than $\epsilon$}
2: $\mathbf{RA} \leftarrow [1, \ldots, 1]^\mathsf{T}_{1 \times n}$ {Initiate relative accuracy vector $\mathbf{RA}$}
3: while $\delta > \epsilon$ do
4:     $\mathbf{NextRA} \leftarrow \mathbf{S} \times \mathbf{RA}$ {Multiply similarity matrix by current relative accuracy}
5:     $\mathbf{NextRA} \leftarrow \mathbf{NextRA} / \max(\mathbf{NextRA})$ {Normalize calculated relative accuracy
       with respect to the largest element in matrix $\mathbf{NextRA}$}
6:     $\delta \leftarrow \sum_{1 \leq i \leq n} |RA_i - \mathsf{NextRA}_i|$ {Calculate $\delta$ between the current and next $\mathbf{RA}$}
7:     $\mathbf{RA} \leftarrow \mathbf{NextRA}$
8: end while
9: Return $\mathbf{RA}$
```

Definition 1. (relative accuracy [1]). *The* relative accuracy *of detector D_i, denoted by RA_i where $1 \leq i \leq n$, is defined by a number in interval $[0, 1]$ such that $RA_i > RA_k$ means detector D_i is more accurate than detector D_k.*

Algorithm 1 [1] computes detectors' relative accuracy vector $\mathbf{RA} = (RA_1, \ldots, RA_n)$, by taking similarity matrix $\mathbf{S} = (S_{ik})_{1 \leq i,k \leq n}$ as input. The algorithm halts when $\delta < \epsilon$, where δ is the sum of the difference for all of the RA_i's between two consecutive iterations and ϵ is a given threshold. In each iteration of the algorithm, the following is conducted: multiply the current relative accuracy vector $\mathbf{RA}$ by $\mathbf{S}$ from the left-hand side (Step 4); scale the result by the largest value in the resulting matrix (Step 5); update $\mathbf{RA}$ for the next iteration (Steps 6–7).

3.2 New Approach to Computing Relative Accuracy and Deeper Analysis

Using synthetic data with known ground-truth information, Algorithm 1 is shown to assure $RA_i > RA_k$ when $\mathsf{ACC}_i > \mathsf{ACC}_k$ [1]. However, it is a heuristic because no explanation on *why it works* is given [1]. Our new approach starts with an observation. Let N_z be the normalization scalar at the z-th iteration in Algorithm 1 and $N_* = N_1 \times \ldots \times N_z$. By leveraging the associative property of scalar multiplication, we observe

$$\mathbf{RA} = \underbrace{\mathbf{S} \ldots (\mathbf{S}(\mathbf{S}}_{z \text{ times}}(\mathbf{1})/N_1)/N_2) \ldots /N_z = \underbrace{\mathbf{S} \ldots (\mathbf{S}(\mathbf{S}}_{z \text{ times}}(\mathbf{1})))/N_* = \mathbf{S}^z \mathbf{1}/N_*. \quad (1)$$

Since $\mathbf{S}$ is a real-valued symmetric matrix, it can undergo eigendecomposition, the eigenvalues are real, and the eigenvectors can be selected real and orthonormal [11]. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ and $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ be respectively the eigenvalues and eigenvectors of $\mathbf{S}$ [11]. The spectral decomposition of $\mathbf{S}$ is:

$$\mathbf{S} = \mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T} = \sum_{i=1}^{n} \lambda_i \mathbf{e}_i \mathbf{e}_i^\mathsf{T}, \quad (2)$$

where $\mathbf{U} = (\mathbf{e}_1, \ldots, \mathbf{e}_n)$ is an orthonormal matrix such that $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{U}\mathbf{U}^\mathsf{T} = \mathbf{I}$, $\mathbf{I}$ being the identity matrix, and $\mathbf{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$[11]. Denote by $\mathbf{D}^z =$

$\mathrm{diag}(\lambda_1^z, \ldots, \lambda_n^z)$. By combining (1) and (2), we have

$$N_* \mathbf{RA} = \mathbf{S}^z \mathbf{1} = (\mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T})^z \mathbf{1} = \mathbf{U}\mathbf{D}^z\mathbf{U}^\mathsf{T}\mathbf{1} = \left(\sum_{i=1}^n \lambda_i^z \mathbf{e}_i \mathbf{e}_i^\mathsf{T}\right)\mathbf{1} = \sum_{i=1}^n \lambda_i^z \mathbf{e}_i(\mathbf{e}_i^\mathsf{T}\mathbf{1}). \tag{3}$$

By replacing $\mathbf{U} \times \mathbf{D}^z \times \mathbf{U}^T$ for $\mathbf{S}^z$ in Eq. (1), we rewrite Algorithm 1 as Algorithm 2.

Algorithm 2. New approach to computing relative accuracy using eigenvalues

```
Input: similarity matrix $\mathbf{S} = (S_{ik})_{1 \le i,k \le n}$; tolerable error threshold $\epsilon$
Output: relative accuracy vector $\mathbf{RA} = (RA_1, \ldots, RA_n)^\mathsf{T}$
1: $\delta \leftarrow 2\epsilon$ {It suffices to initialize $\delta$ as any value that is greater than $\epsilon$}
2: $\mathsf{c} \leftarrow 1$ {Set initial step}
3: $(\mathbf{U}, \mathbf{D}) \leftarrow (eigenvector(\mathbf{S}), eigenvalue(\mathbf{S}))$ {Eigendecomposition}
4: $\mathbf{CurrentRA} \leftarrow \mathbf{U} \times \mathbf{D}^c \times \mathbf{U}^\mathsf{T} / \max(\mathbf{U} \times \mathbf{D}^c \times \mathbf{U}^\mathsf{T})$ {Calculate initial RA and normalize}
5: while $\delta > \epsilon$ do
6:     $\mathbf{NextRA} \leftarrow (\mathbf{U} \times \mathbf{D}^c \times \mathbf{U}^\mathsf{T}) / \max(\mathbf{U} \times \mathbf{D}^c \times \mathbf{U}^\mathsf{T})$ {Update RA and normalize}
7:     $\delta \leftarrow \sum_{1 \le i \le n} |\mathbf{NextRA}_i - \mathbf{CurrentRA}_i|$ {Calculate $\delta$ between two consecutive RA's.}
8:     $c \leftarrow c + 1$ {Iterate step}
9:     $\mathbf{CurrentRA} \leftarrow \mathbf{NextRA}$ {Assign new value}
10: end while
11: Return $\mathbf{CurrentRA}$ {$\mathbf{RA} \leftarrow \mathbf{CurrentRA}$}
```

Algorithm 2 has two advantages: (i) We can rigorously prove Algorithm 2, and therefore Algorithm 1, converges; this is assured by Theorem 1. (ii) Algorithm 2 permits an *algebraic* interpretation of relative accuracy. Details follow.

Theorem 1. *Algorithm 2 always converges, meaning error $\delta \le \epsilon$ eventually.*

Proof. The z-th iteration of the loop leads to vector $\mathbf{RA}_z \leftarrow \mathbf{U} \times \mathbf{D}^z \times \mathbf{U}^\mathsf{T} / \max(\mathbf{U} \times \mathbf{D}^z \times \mathbf{U}^\mathsf{T})$. By leveraging the aforementioned $\mathbf{U} \times \mathbf{U}^T = \mathbf{U} \times \mathbf{U}^{-1} = \mathbf{I}$, we obtain

$$\begin{aligned}
\mathbf{RA}_z &= \frac{\mathbf{U} \times \mathbf{D}^z \times \mathbf{U}^\mathsf{T}}{\max(\mathbf{U} \times \mathbf{D}^z \times \mathbf{U}^\mathsf{T})} = \frac{\mathbf{U} \times \mathbf{D}_1 \times \mathbf{D}_2 \times \cdots \times \mathbf{D}_z \times \mathbf{U}^\mathsf{T}}{\max(\mathbf{U} \times \mathbf{D}_1 \times \mathbf{D}_2 \times \cdots \times \mathbf{D}_z \times \mathbf{U}^\mathsf{T})} \\
&= \frac{\mathbf{U} \times \mathbf{D}_1 \times (\mathbf{U}^\mathsf{T} \times \mathbf{U}) \times \mathbf{D}_2 \times (\mathbf{U}^\mathsf{T} \times \mathbf{U}) \ldots \mathbf{D}_z \times \mathbf{U}^\mathsf{T}}{\max(\mathbf{U} \times \mathbf{D}_1 \times (\mathbf{U}^\mathsf{T} \times \mathbf{U}) \times \mathbf{D}_2 \times (\mathbf{U}^\mathsf{T} \times \mathbf{U}) \ldots \mathbf{D}_z \times \mathbf{U}^\mathsf{T}})\\
&= \frac{(\mathbf{U} \times \mathbf{D}_1 \times \mathbf{U}^\mathsf{T}) \times (\mathbf{U} \times \mathbf{D}_2 \times \mathbf{U}^\mathsf{T}) \ldots (\mathbf{U} \times \mathbf{D}_z \times \mathbf{U}^\mathsf{T})}{\max((\mathbf{U} \times \mathbf{D}_1 \times \mathbf{U}^\mathsf{T}) \times (\mathbf{U} \times \mathbf{D}_2 \times \mathbf{U}^\mathsf{T}) \ldots (\mathbf{U} \times \mathbf{D}_z \times \mathbf{U}^\mathsf{T})})\\
&= \frac{(\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})_1 \times (\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})_2 \ldots (\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})_z}{\max((\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})_1 \times (\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})_2 \ldots (\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})_z)}\\
&= \frac{(\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})^z}{\max((\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})^z)} = \frac{(\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})^z}{(\max(\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T}))^z}.
\end{aligned}$$

Let $\mathbf{Y} = \mathbf{U} \times \mathbf{D} \times \mathbf{U}^{\mathsf{T}}$, y be an arbitrary element of $\mathbf{Y}$, and $y^* = \max(\mathbf{Y})$ being the maximum element. Then, $\forall y \in \mathbf{Y}$, $y \in (0,1)$, $\max(\mathbf{Y}^x) = (\max(\mathbf{Y}))^x$, and $y \leq y^*$. Thus,

$$\lim_{z \to +\infty} \frac{y^z}{(y^*)^z} = \begin{cases} 1 & \text{if } y = \max(\mathbf{Y}) \\ 0 & \text{if } y < \max(\mathbf{Y}) \end{cases}$$

assures that the difference δ between two consecutive iterations monotonically decreases, namely $\delta \leq \epsilon$ eventually and the algorithm halts. □

An Algebraic Interpretations of Relative Accuracy. Algorithm 2 permits an algebraic interpretation of relative accuracy. Consider a data matrix $\mathbf{V} = (V_{ij})_{1 \leq i \leq n, 1 \leq j \leq m}$, where the i-th row represents the i-th variable and each column is a vector of observed values for the n variables. Let $\bar{V}_i = \sum_{j=1}^{m} V_{ij}/m$ and $\sigma_i^2 = \sum_{j=1}^{m} (V_{ij} - \bar{V}_i)^2/(m-1)$ be respectively the sample mean and sample variance of the i-th row, or variable V_i. The sample correlation between variables V_i and V_k is $R_{ik} = \{(n-1)\sigma_i\sigma_k\}^{-1} \sum_{j=1}^{m} (V_{ij} - \bar{V}_j)(V_{kj} - \bar{V}_j)$. In PCA [9], the sample correlation matrix is $\mathbf{R} = (R_{ik})_{1 \leq i,k \leq n}$. Let $\gamma_1 \geq \gamma_2 \geq \ldots, \geq \gamma_n \geq 0$ and $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n$ with $\mathbf{f}_i = (f_{i1}, \ldots, f_{im})^{\mathsf{T}}$ be respectively the eigenvalues and eigenvectors of the correlation matrix $\mathbf{R}$. The spectral decomposition of $\mathbf{R}$ is $\mathbf{R} = \sum_{i=1}^{n} \gamma_i \mathbf{f}_i \mathbf{f}_i^{\mathsf{T}}$, where $\mathbf{f}_i$ is the weight vector for the i-th *principal component* defined as $Z_i = \sum_{j=1}^{m} f_{ij} V_j$, which is a linear combination of the original (column) variables $V_1, \ldots, V_n$. Due to the orthogonality between the $\mathbf{f}_i$'s, the principal components, namely the Z_i's are uncorrelated with each other, meaning that their pairwise correlations are all zeros. Recall that $V_i = (V_{i1}, \ldots, V_{im})$ represents an observation. Imagine that we plot the n observations (i.e., $V_1, \ldots, V_n$) in the n-dimensional space such that each dimension representing one V_i for $1 \leq i \leq n$. Then, the first principal component weight vector $\mathbf{f}_1$ represents the direction where these n observations exhibit the most variation (i.e., having the widest spread); the second principal component weight vector $\mathbf{f}_2$ is orthogonal to $\mathbf{f}_1$ and represents the direction of the most variation among all the directions that are perpendicular to $\mathbf{f}_1$; the third principal component weight vector $\mathbf{f}_3$ is orthogonal to both $\mathbf{f}_1$ and $\mathbf{f}_2$ and represents the direction of the most variation among all the directions that are perpendicular to both $\mathbf{f}_1$ and $\mathbf{f}_2$.

The relative accuracy measure groups detectors according to the magnitudes of the corresponding entries in the vector $\mathbf{RA}$. Note that if $\lambda_i > \lambda_j$, then as z increases the gap between λ_i^z and λ_j^z will become increasingly larger. At some point, we will have $\lambda_i^z >> \lambda_j^z$. These λ values are the elements of the diagonal matrix $\mathbf{D}$ from Equation (3). Due to the uniqueness of eigenvalues, we observe strict ordering $\lambda_1 > \lambda_2 > \lambda_3 > \ldots \geq 0$. The iterations in Algorithm 2 increase z, meaning that at some point we would observe $\lambda_1^z >> \lambda_2^z >> \lambda_2^z >> \ldots \geq 0$. When Algorithm 2 converges, the highest value entries in relative accuracy vector $\mathbf{RA}$, according to Eq. (3), correspond to $\lambda_1^z \mathbf{e}_1(\mathbf{e}_1^{\mathsf{T}}\mathbf{1})$; that is, the highest value entries in the eigenvector $\mathbf{e}_1$ determine the directions of $\mathbf{e}_1$ in the n-dimensional space, with each dimension representing a detector. Similarly, the second group of detectors identified by the relative accuracy would match up with the highest entries in the second eigenvector $\mathbf{e}_2$ of the similarity matrix $\mathbf{S}$. Just like that an entry R_{jl} of the correlation matrix $\mathbf{R}$ in PCA represents how similar two variables V_j and V_l co-vary with each other, an entry S_{ik} of similarity matrix $\mathbf{S}$ represents how similar two detectors D_i and D_k label files. Therefore, while the ordered eigenvectors of $\mathbf{R}$ in PCA represent the ordered directions where the variables $\{V_1, \ldots, V_n\}$ have the most covariance, the ordered eigenvectors of $\mathbf{S}$ represent the ordered directions where detectors $\{D_1, \ldots, D_n\}$ make the most similar decisions. This means that the clustering yielded by the relative magnitudes of entries in vector $\mathbf{RA}$ is indeed meaningful.

Remark. The preceding algebraic interpretation is sound when "good" detectors are not overwhelmed by "poor" detectors, namely when the average ground-truth accuracy (ACC) of each detector involved is greater than 50%. Otherwise, the algebraic meaning behind the algorithm would be inverted, and we'd be calculating which detectors that make the most similar, but incorrect, decisions. This inspires us to propose the idea of leveraging a bellwether detector to recognize/filter the "poor" detectors. Details follow.

3.3 Enhancing Algorithm 2 with a Bellwether Reference Detector

Intuitively, the normalization in Algorithm 2 (Step 6) assures that the most accurate detector would have the highest relative accuracy 1, meaning that each detector has a relative accuracy falling into $[0, 1]$. We observe that if we know the absolute effectiveness of a reference detector, we can make a better use of the relative accuracy because we can see the distance between the reference detector's relative accuracy to its absolute accuracy. This motivates us to introduce a *bellwether* detector, so called in reference to herd animals, where the bellwether member of the flock acts as an indicator and predictor into the behavior of the other members. For our purposes, the bellwether detector is a new artificial detector with known absolute effectiveness which can be derived from the fact that it uniformly labels files at random, independent of anything else. In the case of binary classification, the bellwether detector labels each file as malicious (benign) with a 0.5 probability, independent of anything else. This means that the bellwether detector has an *expected* $\mathsf{TPR} = \mathsf{TNR} = \mathsf{ACC} = 0.5$ when the number of files is large enough.

Recall that $\mathbf{Y} = \mathbf{U} \times \mathbf{D} \times \mathbf{U}^{\mathsf{T}}$ and the relative accuracy in the z-th iteration is $\mathbf{Y}^z/(y^*)^z$ where $y^* = \max(\mathbf{Y})$ is the maximum element in $\mathbf{Y}$ (Theorem 1). It is evident that RA_i is proportional to both its initial value and y^*, making it a geometric series if the iterations were infinite. This means that it's possible to scale $\mathbf{RA}$ *without* iterations while maintaining the geometric relationship between the entries, by leveraging a single entry of known value (i.e., $\mathsf{ACC}_{bellwether} = 0.5$), which is the role of the bellwether detector. The preceding discussion leads to Algorithm 3, which produces an improved relative accuracy, denoted by *bellwether accuracy* or $\mathbf{BA_i}$, for detector i.

In Algorithm 3, we use the eigenvectors and eigenvalues to calculate an initial relative accuracy vector with the bellwether detector (Step 2). The relative accuracy of the bellwether detector, where $BA_{bellwether} = BA_{n+1}$, is subtracted from each entry of $\mathbf{BA}$ (Step 3). This has the effect of transforming the values of $\mathbf{BA}$ so that $BA_{bellwether}$ lies on point 0. The entries of $\mathbf{BA}$ are normalized to the range $(-0.5, 0.5)$ and then transformed to $(0, 1)$, which sets $BA_{bellwether} = 0.5$ and distributes the rest of the values according to their geometric relationship. Due to the distribution of this geometric series and the fixed location of the bellwether detector in the array, the results are the same for the first, second, or z-th iteration. This eliminates the need for iterations.

Another potential property of Algorithm 3 is: It may be universally true that $|BA_i - \mathsf{ACC}_i| \leq |RA_i - \mathsf{ACC}_i|$, meaning that BA_i is a better approximation of ACC_i than

Algorithm 3. Computing relative accuracy using eigenvalues and a bellwether detector

```
Input: similarity matrix $\mathbf{S} = (S_{ik})_{1\le i,k\le n+1}$ with $D_{n+1}$ being the bellwether detector;
Output: bellwether accuracy vector $\mathbf{BA_{n+1}} = (BA_1, \ldots, BA_n, 0.5)^\mathsf{T}$
1: $(\mathbf{U}, \mathbf{D}) \leftarrow (eigenvector(\mathbf{S}), eigenvalue(\mathbf{S}))$ {Eigendecomposition}
2: $\mathbf{BA} \leftarrow \mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T} / \max(\mathbf{U} \times \mathbf{D} \times \mathbf{U}^\mathsf{T})$ {Calculate $\mathbf{BA}$}
3: $\mathbf{BA} \leftarrow \mathbf{BA} - \mathbf{BA_{n+1}}$ {Making bellwether-based adjustment}
4: $\mathbf{BA} \leftarrow \mathbf{BA}/(2\times|\max(\mathbf{BA})|)+0.5$ {Normalizing $(-0.5, 0.5)$ and transforming to $(0, 1)$}
5: Return $\mathbf{BA}$
```

RA_i does. Although we cannot prove this rigorously at the time of writing, we do find empirical evidence using synthetic data with known ground-truth information in **Experiments 1–10**, as shown in Fig. 1.

4 Inferring Ground-Truth Labels and Effectiveness Metrics

Method. Consider $\mathbf{V} = (V_{ij})_{1\le i\le \eta, 1\le j\le m}$ and $\mathbf{BA} = (BA_1, \ldots, BA_\eta)$ as input, where η is the number of detectors that will participate in the process of inferring ground-truth labels and detector effectiveness metrics. In the recommended use case, we only use detectors with a bellwether relative accuracy higher than that of the bellwether detector's, namely $BA_i > BA_{bellwether} = 0.5$ for any $1 \le i \le \eta$; in this case we have $1 \le \eta \le n$. It is possible for $\eta = n+1$, with D_{n+1} being the bellwether detector; that is, all of the detectors (including the bellwether detector) participate in the process of inference. In practice it would not be a good idea to include $BA_{bellwether}$ as part of the input because of its random nature (i.e., it is a known source of noise and adds no useful information). Even if included, it should not be involved in any useful computation (e.g., voting, if applicable). Other variants are possible, for example using detectors with bellwether relative accuracy *significantly* higher than $BA_{bellwether} = 0.5$.

Given the input mentioned above, now we design a novel method, Algorithm 4, to infer the ground-truth labels, denoted by $(\mathsf{malign}_1, \ldots, \mathsf{malign}_m)$, and detector effectiveness metrics TPR'_i, TNR'_i and ACC'_i for $1 \le i \le \eta$. At the core of the algorithm is *weighted majority voting*, where weights are iteratively derived from $\mathbf{BA}$. Specifically, the initial $\mathbf{BA}$ is used to weigh individual detector's votes, providing a detector of a higher BA_i with a larger vote weight. The weighted voting leads to malicious or benign labels of files. The updated labels are used to update effectiveness metrics $\mathbf{TPR}'$, $\mathbf{TNR}'$ and $\mathbf{ACC}'$, which in turn are used to update the detectors' weights. The algorithm halts when the inferred $\mathbf{ACC}$ between two consecutive iterations is below a threshold ϵ. Its computational complexity depends on the number of iterations, for which are are unable to give an explicit estimation at this point; in each iteration, the computational complexity is $\mathcal{O}(m\eta)$. In our experiments presented later, we will empirically measure the actual computational complexity.

Designing Experiment with Synthetic Data to Validate the Method. In order to generate synthetic data with known ground-truth labels and detector effectiveness, we consider TPR and TNR, which are often interpreted as probabilities in practice. We gener-

Algorithm 4. Inferring ground-truth labels and effectiveness metrics

Input: Detector labelling results $\mathbf{V} = (V_{ij})_{1 \le i \le \eta, 1 \le j \le m}$; bellwether-incurred accuracy vector $\mathbf{BA} = (BA_1, \ldots, BA_\eta)^\mathsf{T}$; tolerable error threshold ϵ

Output: Inferred ground-truth labels $(\mathsf{Malign}_1, \ldots, \mathsf{Malign}_m)$ and detectors' effectiveness metrics $\mathbf{TPR}' = (\mathsf{TPR}'_1, \ldots, \mathsf{TPR}'_\eta)^\mathsf{T}$, $\mathbf{TNR}' = (\mathsf{TNR}'_1, \ldots, \mathsf{TNR}'_\eta)^\mathsf{T}$, $\mathbf{ACC}' = (\mathsf{ACC}'_1, \ldots, \mathsf{ACC}'_\eta)^\mathsf{T}$;

```
 1: $\delta \leftarrow 2\epsilon$ {It suffices to initialize $\delta$ as any value that is greater than $\epsilon$}
 2: $\mathbf{ACC}' \leftarrow \mathbf{BA}$ {Initiate accuracy vector as $\mathbf{BA}$}
 3: while $\delta > \epsilon$ do
 4:    $\mathbf{TN}, \mathbf{TP}, \mathbf{FN}, \mathbf{FP} \leftarrow (0, \ldots, 0)^\mathsf{T}_{1\times\eta}$ {Initialize $\mathsf{TN}_i = \mathsf{TP}_i = \mathsf{FN}_i = \mathsf{FP}_i = 0$ for
       $1 \le i \le \eta$}
 5:    for $j := 1$ to $m$ do
 6:       $\mathsf{MalignWeight} \leftarrow 0$;   $\mathsf{BenignWeight} \leftarrow 0$;   {Initialize weights as 0}
 7:       for $i := 1$ to $\eta$ do
 8:          if $V_{ij} = 1$ then
 9:             $\mathsf{MalignWeight} \leftarrow \mathsf{MalignWeight} + \mathsf{ACC}'_i$ {Count detector $i$'s vote for malicious}
10:          else
11:             $\mathsf{BenignWeight} \leftarrow \mathsf{BenignWeight} + \mathsf{ACC}'_i$ {Count detector $i$'s vote for benign}
12:          end if
13:       end for
14:       if $\mathsf{MalignWeight} > \frac{\mathsf{MalignWeight}+\mathsf{BenignWeight}}{2}$ then
15:          $\mathsf{Malign}_j \leftarrow 1$ {If majority of weighted votes are malicious, label file $j$ as malicious}
16:       else
17:          $\mathsf{Malign}_j \leftarrow 0$
18:       end if
19:       for $i := 1$ to $\eta$ do
20:          if $\mathsf{Malign}_j = 0$ then
21:             if $V_{ij} = 1$ then
22:                $\mathsf{FP}_i \leftarrow \mathsf{FP}_i + 1$ {file $j$ is a false-positive by detector $i$}
23:             else if $V_{ij} = 0$ then
24:                $\mathsf{TN}_i \leftarrow \mathsf{TN}_i + 1$ {file $j$ is a true-negative by detector $i$}
25:             end if
26:          else
27:             if $V_{ij} = 1$ then
28:                $\mathsf{TP}_i \leftarrow \mathsf{TP}_i + 1$ {file $j$ is a true-positive by detector $i$}
29:             else if $V_{ij} = 0$ then
30:                $\mathsf{FN}_i \leftarrow \mathsf{FN}_i + 1$ {file $j$ is a false-negative by detector $i$}
31:             end if
32:          end if
33:       end for
34:    end for
35:    $\delta \leftarrow \sum_{i=1}^{\eta} |\mathsf{ACC}'_i - \frac{\mathsf{TN}_i+\mathsf{TP}_i}{\mathsf{TN}_i+\mathsf{TP}_i+\mathsf{FN}_i+\mathsf{FP}_i}|$ {Changes in ACC between this and last iterations}
36:    for $i := 1$ to $\eta$ do
37:       $\mathsf{ACC}'_i \leftarrow \frac{\mathsf{TN}_i+\mathsf{TP}_i}{\mathsf{TN}_i+\mathsf{TP}_i+\mathsf{FN}_i+\mathsf{FP}_i}$, $\mathsf{TPR}'_i \leftarrow \frac{\mathsf{TP}_i}{\mathsf{TP}_i+\mathsf{FN}_i}$, $\mathsf{TNR}'_i \leftarrow \frac{\mathsf{TN}_i}{\mathsf{TN}_i+\mathsf{FP}_i}$
38:    end for
39:    $\mathbf{Malign} \leftarrow (\mathsf{Malign}_1, \ldots, \mathsf{Malign}_m)$; $\mathbf{ACC}' \leftarrow (\mathsf{ACC}'_1, \ldots, \mathsf{ACC}'_\eta)^\mathsf{T}$; $\mathbf{TPR}' \leftarrow$
       $(\mathsf{TPR}'_1, \ldots, \mathsf{TPR}'_\eta)^\mathsf{T}$; $\mathbf{TNR}' \leftarrow (\mathsf{TNR}'_1, \ldots, \mathsf{TNR}'_\eta)^\mathsf{T}$
40: end while
41: Return inferred ground-truth labels $\mathbf{Malign}$ and $\mathbf{TPR}'$, $\mathbf{TNR}'$ and $\mathbf{ACC}'$
```

ate three synthetic datasets, denoted by **D1**, **D2** and **D3**. Each dataset contains one million example files, but different ratios of malicious vs. benign entries: (i) **D1** contains 300,000 malicious files and 700,000 benign files; (ii) **D2** contains 500,000 malicious files and 500,000 benign files; and (iii) **D3** contains 700,000 malicious files and 300,000 benign files. This is to show that the algorithms are equally applicable to various ratios. Note that no actual files are generated because we only need their ground-truth labels and their labels given by the detectors. For a detector, we generate its label on a file according to the detector's TPR and TNR, as follows. If the ground-truth label of the file is malicious (i.e., malign), then the detector labels it as 1 (malicious) with probability TPR and 0 (benign) with probability $1 - \mathsf{TPR}$; if the ground truth label of the file is benign, then the detector labels it as 0 with probability TNR and labels it as 1 with probability $1 - \mathsf{TNR}$. Given the labels, the accuracy ACC of a detector can be computed as described in Sect. 2. We apply Algorithm 3 to the synthetic datasets **D1**-**D3** as well as the detectors' labels on these files to derive the detectors' relative accuracy (i.e., relative accuracy with the bellwether detector $\mathbf{BA}$), and then apply Algorithm 4 with the $\mathbf{BA}$ to derive their inferred TPR', TNR' and ACC'. Finally, we compare these inferred metrics to their ground-truth counterparts TPR, TNR and ACC to validate the method.

We conduct 10 experiments, denoted by **Experiments 1–10**. For purposes of robustness, the experiments consider detectors with varying ACC's. The 10 experiments are: **(1)** 50 detectors with varying ground-truth ACC: 10 detectors with $\mathsf{ACC} \in [0.85, 0.95]$, 10 with $\mathsf{ACC} \in [0.75, 0.85]$, 10 with $\mathsf{ACC} \in [0.7, 0.8]$, and 20 with $\mathsf{ACC} \in [0.65, 0.75]$. **(2)** 50 detectors: 10 detectors with $\mathsf{ACC} \in [0.9, 1]$; 10 with $\mathsf{ACC} \in [0.85, 0.95]$; 10 with $\mathsf{ACC} \in [0.8, 0.9]$; 20 with $\mathsf{ACC} \in [0.75, 0.85]$. **(3)** 50 detectors with $\mathsf{ACC} \in [0.9, 1]$. **(4)** 90 detectors: 50 with $\mathsf{ACC} \in [0.9, 1]$, 10 with $\mathsf{ACC} \in [0.85, 0.95]$, 10 with $\mathsf{ACC} \in [0.8, 0.9]$, 10 with $\mathsf{ACC} \in [0.75, 0.85]$, and 10 with $\mathsf{ACC} \in [0.35, 0.45]$. **(5)** 50 detectors: 10 detectors with $\mathsf{ACC} \in [0.9, 1]$, 10 with $\mathsf{ACC} \in [0.85, 0.95]$, 10 with $\mathsf{ACC} \in [0.8, 0.9]$, 10 with $\mathsf{ACC} \in [0.75, 0.85]$, and 10 with $\mathsf{ACC} \in [0.35, 0.45]$. **(6)** 50 detectors: 40 detectors with $\mathsf{ACC} \in [0.9, 1]$ and 10 with $\mathsf{ACC} \in [0.35, 0.45]$. **(7)** 50 detectors: 30 detectors with $\mathsf{ACC} \in [0.9, 1]$ and 20 with $\mathsf{ACC} \in [0.35, 0.45]$. **(8)** 50 detectors with 25 detectors with $\mathsf{ACC} \in [0.9, 1]$ and 25 detectors with $\mathsf{ACC} \in [0.35, 0.45]$. **(9)** 50 detectors with 20 detectors with $\mathsf{ACC} \in [0.9, 1]$ and 30 detectors with $\mathsf{ACC} \in [0.35, 0.45]$. **(10)** 50 detectors with 10 detectors with $\mathsf{ACC} \in [0.9, 1]$ and 40 detectors with $\mathsf{ACC} \in [0.35, 0.45]$.

Experimental Results. The computational complexity of Algorithm 4 is $O(\eta m)$ for each iteration when $\delta > \epsilon$, where $\eta << m$. Unfortunately, we cannot estimate how many iterations it will take before reaching $\delta < \epsilon$ at the time of writing.

For each experiment and each detector D_i, there are four sets of values: the ground-truth effectiveness metrics TPR_i, TNR_i, and ACC_i, where TPR_i and TNR_i are the input for generating synthetic data for detector D_i and ACC_i is derived from them as mentioned above; the relative accuracy RA_i (Algorithm 1); the bellwether relative accuracy BA_i (Algorithm 3); and the inferred effectiveness metrics TPR'_i, TNR'_i, and ACC'_i (Algorithm 4). Owing to space limit, we only present the experimental results on ACC'_i, while noting that the results on TPR'_i and TNR'_i exhibit similar characteristics. Moreover, we only present the experimental results with **D1**, because the results with **D2** and **D3** are almost the same as that of **D1**.

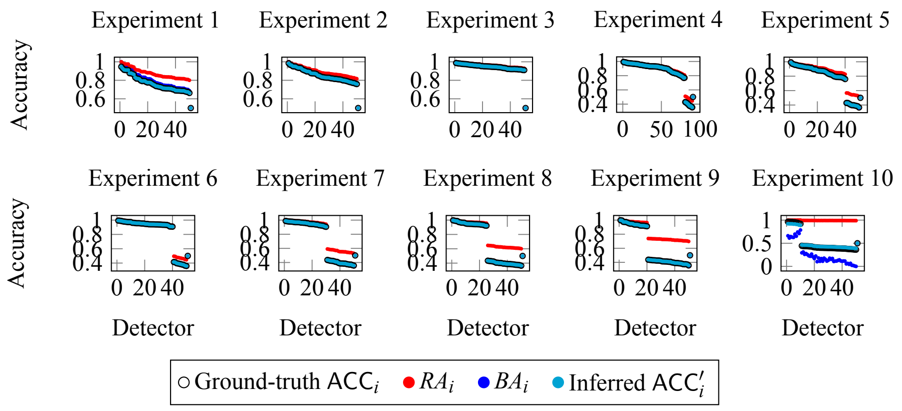

Fig. 1. Results of **Experiments 1–10** with synthetic dataset **D1**: ground-truth ACC_i (which is, when invisible, hidden behind the inferred ACC'_i curve), relative accuracy (RA_i, which is, when invisible, hidden behind the inferred ACC'_i curve), bellwether relative accuracy (BA_i, which is, when invisible, hidden behind the inferred ACC'_i curve), and inferred accuracy ACC'_i (y-axis) of detector i (x-axis), in the descending order of ACC.

Figure 1 plots the experimental results with dataset **D1**. We make the following observations. First, we observe $RA_i \neq \mathsf{ACC}_i$ and $BA_i \neq \mathsf{ACC}_i$, which is expected and highlights the necessity of Algorithm 4. Nevertheless, we observe that $|\mathsf{ACC}_i - BA_i| \leq |\mathsf{ACC}_i - RA_i|$ holds for every i in the experiments, meaning that bellwether relative accuracy improves upon the relative accuracy. Second, we observe the inferred $\mathsf{ACC}'_i \approx \mathsf{ACC}_i$ in most cases. In **Experiments 1–9**, we observe $\max_i(|\mathsf{ACC}_i - \mathsf{ACC}'_i|) = 2 \times 10^{-6}$; in **Experiment 10**, we observe a higher error, with $\max_i(|\mathsf{ACC}_i - \mathsf{ACC}'_i|) \approx 0.04$ and average error 0.012 (among all detectors), due to the fact that there are only 10 good detectors with $\mathsf{ACC} \in [0.9, 1]$ but 40 poor detectors with $\mathsf{ACC} \in [0.35, 0.45]$. This confirms the usefulness of Algorithm 4. Third, we select detectors with $BA_i > 0.5$ and pass them to Algorithm 4 (which corresponds to the recommended use case mentioned above), also in **Experiments 1–10**. We contrast this result with what is plotted in Fig. 1, which uses all of the detectors (rather than a selection of them). Owing to space limit, we only report that **Experiments 1–3** do not differ in these two settings because no detectors have $BA_i \leq 0.5$; ACC'_i in **Experiments 4–9** are improved and match ACC_i exactly; ACC'_i in **Experiment 10** is most improved, with the largest error being $\max_i(|\mathsf{ACC}'_i - \mathsf{ACC}_i|) \approx 0.04$. In each of the 10 experiments, we highlight that at most 2 (out of 1 million) files are mislabelled, leading to negligible TNR'_i and TPR'_i. This highlights the usefulness of using the bellwether detector to filter out detectors with $BA_i \leq 0.5$. Fourth, the order-preserving property that $\mathsf{ACC}_i > \mathsf{ACC}_j$ implies $\mathsf{ACC}'_i > \mathsf{ACC}'_j$ is preserved when detectors with $BA_i \leq 0.5$ are eliminated but not before they are eliminated. This further highlights the importance of filtering out detectors with $BA_i \leq 0.5$.

In order to show that Algorithm 4 outperforms the heuristic of unweighted majority voting, we also conduct **Experiments 1–10** with the latter. By contrasting the results, we observe that unweighted majority voting, while performing moderately well in

Experiments 1–7 with a maximum number of mislabeled files being 17, performs poorly in **Experiments 8–10** where poor detectors begin to outnumber good detectors, with 4,350, 17,450, and 441,500 files mislabeled, respectively. This reiterates the advantage of the recommended use case of the principled Algorithm 4 (i.e., filtering out detectors with $BA_i \leq 0.5$) over the heuristic of unweighted majority voting.

Insight 1. *Algorithm 4 is a principled method for inferring ground-truth labels and detector effectiveness, especially when eliminating the detectors whose $BA_i \leq 0.5$. It also substantially outperforms the unweighted majority voting heuristic.*

5 Applying the Method to Real-World Dataset

Now we apply the method to a real-world dataset collected from VirusTotal, with $n = 62$ detectors and $m = 10,738,585$ files. Each file is labeled as malicious (1) or benign (0), but not every file is labeled by every detector. The dataset is succinctly represented by matrix $\mathbf{V} = (V_{ij})_{n\times m}$, which leads to similarity matrix $\mathbf{S}$, and the bellwether relative accuracy vector $\mathbf{BA}$ according to Algorithm 3. We conduct two experiments with Algorithm 4: the first experiment corresponds to the recommended use case of only using detectors with $BA_i > 0.5$; the second experiment uses all of the 62 detectors together with the bellwether detector which is always given voting weight 0 (i.e., $\eta = 63$), while aiming to draw more insights. We observe that in the aforementioned experiments, with an input of $m = 10,738,585$ files and $\eta \leq 63$ detectors, Algorithm 4 took on the order of tens of minutes to complete each iteration, with a maximum number of 8 iterations.

In the first experiment, we observe 56 (out of the 62) detectors have $BA_i > 0.5$. Algorithm 4 outputs $7,408,449$ files as malicious and the TPR'_i, TNR'_i and ACC'_i's for the 56 detectors. We observe 22 (out of the 56) detectors have $\mathsf{TNR}'_i > \mathsf{TPR}'_i$ (i.e., they prefer low false-positives to low false-negatives) and the other 34 have $\mathsf{TNR}'_i < \mathsf{TPR}'_i$ (i.e., these detectors prefer the opposite). This suggests that a majority of the good detectors prefer low false-negatives to low false-positives; this is consistent with a finding reported in [7]. Treating the inferred labels as ground-truth (as validated in Sect. 4), we can test the heuristic unweighted majority voting, which has to use all of the detectors because it cannot distinguish which detectors are more accurate than others. We find that the unweighted majority voting method only labels $6,021,073$ (out of the $7,408,449$ malicious) files as malicious, meaning high false-negatives.

For the second experiment, Fig. 2 plots the bellwether relative accuracy BA_i and the inferred ACC'_i, TPR'_i and TNR'_i for detectors $1 \leq i \leq 62$, in descending order respective of ACC'_i, with $BA_{bellwether} = 0.5$ as the reference line; note that $\mathsf{TPR}'_{bellwether} = 0.506$, $\mathsf{TNR}'_{bellwether} = 0.504$, $\mathsf{ACC}'_{bellwether} = 0.505$ are not plotted. We make the following observations. First, four detectors $i \in \{59, 60, 61, 62\}$ have $BA_i \approx 0$, $\mathsf{TPR}'_i \approx 0$, $\mathsf{TNR}'_i \approx 0$, and $\mathsf{ACC}'_i \approx 0$, meaning they are not reliable. Looking into the data we find that they only labeled 52, 51, 37, 1 files, respectively. Second, detectors D_5 and D_{37} have $BA_5 = 0.256$ but $\mathsf{ACC}'_5 = 0.92$ and $BA_{37} = 0.515$ but $\mathsf{ACC}'_{37} = 0.83$. This does not contradict with anything discussed above, because it can be explained by the fact that D_5 labels about 33% of the files and D_{37} labels

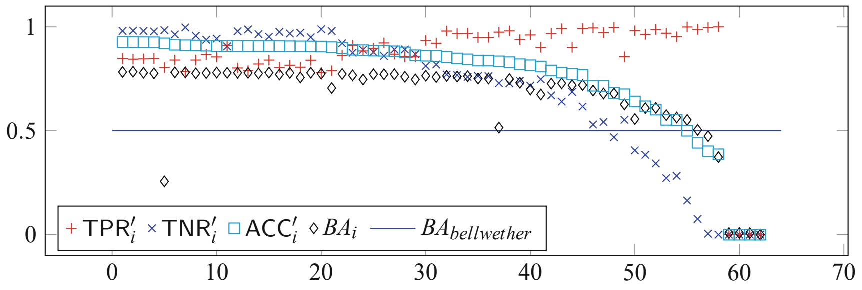

Fig. 2. Plots of bellwether relative accuracy BA_i, and inferred TPR'_i, TNR'_i and ACC'_i (the y-axis) for detectors $1 \leq i \leq 62$ (the x-axis), with $BA_{bellwether} = 0.5$ being the reference line.

about 69%. Scaling BA_5 and BA_{37} proportional to the number of files they label would rectify this phenomenon and place the values where expected in $\mathbf{BA}$, namely $BA_5 = 0.256/0.33 = 0.775$ and $BA_{37} = 0.515/0.69 = 0.746$. Nevertheless, this does highlight that $BA_i > BA_k$ does not necessarily imply $\mathsf{ACC}'_i > \mathsf{ACC}'_k$ when D_i and D_k label significantly different sets of files, which is inherent to the definition of similarity because C_{ik} applies to the *union* of the two sets of files that are labeled by D_i and D_k, whereas A_{ik} applies to the *intersection* of the two sets. In the full version of the paper we will investigate whether requiring that all detectors label the *same* set of files would make our method even more robust. Third, treating as the ground-truth the outcome in the first experiment (with detectors of $BA_i > 0.5$), the TPR'_i, TNR'_i and ACC'_i's of the detectors that are common to these two experiments are largely consistent, indicating that individual detector metrics are consistent during the recovery process. Fourth, 22 detectors have $\mathsf{TNR}'_i > \mathsf{TPR}'_i$ (i.e., they prefer low false-positives to low false-negatives) and the remainder show the opposite. This is consistent with what is observed in the first experiment mentioned above.

Insight 2. *The principled* weighted *majority voting method is significant more accurate than the unweighted majority voting heuristic. Most detectors prefer low false-negatives to low false-positives.*

6 Conclusion

We have presented a principled weighted majority voting method for inferring ground-truth labels of files and malware detector effectiveness metrics, taking as input the (conflicting) labels given by a set of malware detectors on a set of files. The proposed method is supported by an algebraic interpretation of the notion of relative accuracy introduced in [1]. Another key idea is to introduce the notion of bellwether detector for serving as a reference to eliminate the detectors that perform worse than random labeling of files. We empirically validate the method by using synthetic data with known ground-truth information. We apply the method to a real-world dataset collected from VirusTotal.

The present paper makes a solid step towards characterizing the proposed method, but there are some outstanding open problems: Can "$BA_i > BA_j$ implying $\mathsf{ACC}_i >$

ACC_j" be rigorously proven? Can the error bounds on $|\mathsf{ACC}_i - \mathsf{ACC}'_i|$ be rigorously characterized? Can the idea of eliminating detectors with $BA_i \leq 0.5$ be rigorously justified? If any of the above is not universally true, what is the necessary and sufficient condition under which it is true?

Acknowledgement. We thank the reviewers for their useful comments. This work was supported in part by NSF Grant #2122631 (#1814825) and by a Grant from the State of Colorado.

References

1. Charlton, J., Du, P., Cho, J.H., Xu, S.: Measuring relative accuracy of malware detectors in the absence of ground truth. In: Proceedings IEEE MILCOM, pp. 450–455 (2018)
2. Chen, H., Cho, J., Xu, S.: Quantifying the security effectiveness of firewalls and dmzs. In: Proceedings HoTSoS 2018, pp. 9:1–9:11 (2018)
3. Chen, H., Cho, J., Xu, S.: Quantifying the security effectiveness of network diversity. In: Proceedings HoTSoS 2018, p. 24:1 (2018)
4. Cheng, Y., Deng, J., Li, J., DeLoach, S., Singhal, A., Ou, X.: Metrics of security. In: Cyber Defense and Situational Awareness, pp. 263–295 (2014)
5. Cho, J., Hurley, P., Xu, S.: Metrics and measurement of trustworthy systems. In: IEEE Military Communication Conference (MILCOM 2016) (2016)
6. Cho, J., Xu, S., Hurley, P., Mackay, M., Benjamin, T., Beaumont, M.: Stram: measuring the trustworthiness of computer-based systems. ACM Comput. Surv. **51**(6), 128:1–128:47 (2019)
7. Du, P., Sun, Z., Chen, H., Cho, J.H., Xu, S.: Statistical estimation of malware detection metrics in the absence of ground truth. IEEE T-IFS **13**(12), 2965–2980 (2018)
8. Homer, J., et al.: Aggregating vulnerability metrics in enterprise networks using attack graphs. J. Comput. Secur. **21**(4), 561–597 (2013)
9. Hotelling, H.: Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. **24**(6), 417 (1933)
10. Invernizzi, L., Benvenuti, S., Cova, M., Comparetti, P.M., Kruegel, C., Vigna, G.: Evilseed: a guided approach to finding malicious web pages. In: IEEE Symposium on Security and Privacy, pp. 428–442 (2012)
11. Johnson, C.R., Horn, R.A.: Matrix Analysis. Cambridge University Press, Cambridge (1985)
12. Kantchelian, A., et al.: Better malware ground truth: techniques for weighting anti-virus vendor labels. In: Proceedings 2015 ACM Workshop on Artificial Intelligence and Security, pp. 45–56 (2015)
13. Kührer, M., Rossow, C., Holz, T.: Paint it black: evaluating the effectiveness of malware blacklists. In: Proceedings Research in Attacks, Intrusions and Defenses (RAID 2014), pp. 1–21 (2014)
14. Mireles, J., Ficke, E., Cho, J., Hurley, P., Xu, S.: Metrics towards measuring cyber agility. IEEE T-IFS **14**(12), 3217–3232 (2019)
15. Mohaisen, A., Alrawi, O.: Av-meter: an evaluation of antivirus scans and labels. In: Proceedings DIMVA, pp. 112–131 (2014)
16. Morales, J., Xu, S., Sandhu, R.: Analyzing malware detection efficiency with multiple anti-malware programs. In: Proceedings CyberSecurity (2012)
17. Noel, S., Jajodia, S.: A suite of metrics for network attack graph analytics. In: Network Security Metrics, pp. 141–176. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66505-4_7

18. Pendleton, M., Garcia-Lebron, R., Cho, J., Xu, S.: A survey on systems security metrics. ACM Comput. Surv. **49**(4), 62:1–62:35 (2016)
19. Perdisci, R., ManChon, U.: Vamo: Towards a fully automated malware clustering validity analysis. In: Proceedings. ACSAC, pp. 329–338 (2012)
20. Pritom, M., Schweitzer, K., Bateman, R., Xu, M., Xu, S.: Data-driven characterization and detection of COVID-19 themed malicious websites. In: IEEE ISI 2020 (2020)
21. Ramos, A., Lazar, M., Filho, R.H., Rodrigues, J.J.P.C.: Model-based quantitative network security metrics: a survey. IEEE Commun. Surv. Tutorials **19**(4), 2704–2734 (2017)
22. Wang, L., Jajodia, S., Singhal, A.: Network Security Metrics. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66505-4
23. Wang, L., Jajodia, S., Singhal, A., Cheng, P., Noel, S.: K-zero day safety: a network security metric for measuring the risk of unknown vulnerabilities. IEEE TDSC **11**(1), 30–44 (2014)
24. Xu, L., Zhan, Z., Xu, S., Ye, K.: Cross-layer detection of malicious websites. In: ACM CODASPY, pp. 141–152 (2013)
25. Xu, L., Zhan, Z., Xu, S., Ye, K.: An evasion and counter-evasion study in malicious websites detection. In: IEEE CNS, pp. 265–273 (2014)
26. Zhang, J., Durumeric, Z., Bailey, M., Liu, M., Karir, M.: On the mismanagement and maliciousness of networks. In: Proceedings NDSS 2014 (2014)
27. Zhang, M., Wang, L., Jajodia, S., Singhal, A., Albanese, M.: Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks. IEEE Trans. Inf. Forensics Secur. **11**(5), 1071–1086 (2016)
28. Zhu, S., et al.: Measuring and modeling the label dynamics of online anti-malware engines. In: 29th USENIX Security Symposium, USENIX Security 2020, 12–14, August 2020, pp. 2361–2378 (2020)