Hittite Journal of Science and Engineering, 2021, 8 (2) 103–121
ISSN NUMBER: 2148–4171
DOI: 10.17350/HJSE19030000221

Gaining New Insight into Machine-Learning Datasets via Multiple Binary-Feature Frequency Ranks with a Mobile Benign/Malware Apps Example

Gurol Canbek
ASELSAN, Ankara, Turkey

Article History:
Received: 2021/01/08
Accepted: 2021/03/08
Online: 2021/06/30

***Correspondence to**: Gurol Canbek,*
ASELSAN, 06200, Ankara, TURKEY
E-Mail: gurol@canbek.com,
gcanbek@aselsan.com.tr

ABSTRACT

Researchers compare their Machine Learning (ML) classification performances with other studies without examining and comparing the datasets they used in training, validating, and testing. One of the reasons is that there are not many convenient methods to give initial insights about datasets besides the descriptive statistics applied to individual continuous or quantitative features. After demonstrating initial manual analysis techniques, this study proposes a novel adaptation of the Kruskal-Wallis statistical test to compare a group of datasets over multiple prominent binary features that are very common in today's datasets. As an illustrative example, the new method was tested on six benign/malign mobile application datasets over the frequencies of prominent binary features to explore the dissimilarity of the datasets per class. The feature vector consists of over a hundred "application permission requests" that are binary flags for Android platforms' primary access control to provide privacy and secure data/information in mobile devices. Permissions are also the first leading transparent features for ML-based malware classification. The proposed data analytical methodology can be applied in any domain through their prominent features of interest. The results, which are also visualized in three new ways, have shown that the proposed method gives the dissimilarity degree among the datasets. Specifically, the conducted test shows that the frequencies in the aggregated dataset and some of the datasets are not substantially different from each other even they are in close agreement in positive-class datasets. It is expected that the proposed domain-independent method brings useful initial insight to researchers on comparing different datasets.

Keywords:
Machine learning; Binary classification; Dataset comparison; Malware analysis; Feature engineering; Quantitative analysis.

INTRODUCTION

The success and performance of Machine Learning (ML) algorithms closely depend on the datasets used, their sample and feature spaces, and sampling quality. Researchers who build a classifier that is trained and tested on a dataset publish their classification performances in terms of standard metrics such as accuracy, true positive rate, or F1 [1]. The classifiers are compared with other classifiers that are trained and tested on different datasets via the same performance metrics. The datasets are usually not compared or analyzed. On the other hand, researchers who wish to enrich their datasets usually merge new datasets they acquired from other sources without analyzing them. They could not be sure how these datasets are different from the existing ones.

Indeed, some statistical methods could be used to describe datasets. However, those statistical approaches summarize a dataset based on a single feature that is usually continuous. A box plot, for example, visualizes and compares the descriptive statistics such as mean, median, range, and outliers [2]. Likewise, the statistics related to the shape of the feature distribution, such as skewness, kurtosis, and the number of peaks, can be analyzed [3]. Dataset profiling based on other statistical properties such as timeliness (freshness of the samples), sample duplication, and feature density gives extra insight among the compared datasets [4]. Nevertheless, interpreting and comparing statistical figures alone are not convenient; besides, they are usually not suitable for discrete or qualitative features. To avoid such problems, new methods should be developed to give insights about one or comparatively more than one dataset. Better, the methods should be enhanced by visualization.

This study has proposed a method to compare datasets by adapting the Kruskal-Wallis test with a novel approach to compare the medians of a prominent feature's frequencies to determine if the samples come from the same population or equivalently having the same distribution. This study aims to provide a new method for the researchers to compare more than one dataset over the common binary features. The study also adopts three visualization techniques to assess the comparisons based on the proposed method's outputs. A developed API described in Appendix A to calculate and visualize the method is provided to conduct such comparisons conveniently.

The method was tested and evaluated on Android mobile benign applications and malware datasets in the literature. The mobile malware classification problem was chosen because it is a critical emerging cyber security field where ML-based classification approaches are highly studied and practiced in the literature and the industry to enhance the capacities related to the human factor [5]. The results of the proposed comparison method summarized in Section 6 are encouraging, and shed light on using datasets on malware classification. Note that the proposed method is not specific to malware analysis, and it is expected that it could be used in any other area for comparing datasets in binary and even multi-class classification problems.

The rest of the paper is organized as follows. Section 2 introduces the classification problem domain. Section 3 describes and demonstrates techniques for an initial manual analysis of the reviewed datasets, namely basic quantitative comparison of sample/feature spaces and binary-feature space graphical analysis. It summarizes the negative and positive-class datasets to be compared in this study. Two suggested graphics, one of which is provided online as an interactive chart, to support such analysis are also demonstrated. Section 4 presents the followed methodology and the activities for comparing the datasets from different perspectives, including how to aggregate datasets. Section 5 explains the proposed comparison method based on a novel adaptation of the Kruskal-Wallis test. Section 6 provides the dataset comparison results enhanced with the suggested visualization techniques. The last two sections present the discussion and summarize the advantages of the proposed comparison methods and outline this study's contributions. Appendix A lists online supplementary materials (open-source API, interactive chart, and datasets). Appendix B surveys the related chosen pieces of work about Android application permissions and highlights the Android permission mechanism's significant aspects related to static malware analysis.

THE CASE STUDY CLASSIFICATION PROBLEM DOMAIN

The following subheadings introduce the case study problem domain, the binary features to be used in comparisons, and dataset usage in the related literature.

Android Mobile-Malware Classification Problem

Android is a mobile platform that provides a large number and a wide range of mobile applications. Android applications are developed by anyone and released on third-party application markets besides the official market named Google Play. Despite this diversity, the platform could be the target of malicious people who develop or make injections into existing applications that exposes some risks against end-users. Malware authors develop and use different techniques in those applications appearing as legitimate to overcome the platform's security or exploit human factors. Therefore, mobile malware detection, which labeling a given application as 'benign' ('negative') or 'malign' ('positive', also known as 'malware'), is one of the urging areas to be studied by the security sector and academia. Experts examine the applications manually with the help of specialized tools (e.g., reverse engineering software) and decide whether they are benign or malign. This human-involved process is called malware analysis [6]. In addition to dynamic malware analysis that concentrates on applications' behaviors observed at run-time, static malware analysis examines binaries, files, and codes to classify Android malware from benign applications [7].

Mobile Application Permission Requests as Features

Manual analysis is impossible to conduct, considering the excessive number of applications. Solely in Google Play Store, on average, 3,700 new mobile applications are released every day [8]. To some degree, machine learning comes as a promising solution to classify malware among many mobile applications based on various features [9]. Android's permission mechanism limits the specific operations performed by applications or provides ad hoc access to particular data at the end-users discretion [10]. If an application is required to initiate a phone call without going through the standard dialer user interface for the user to confirm the call, for example, it must manifest or request CALL_PHONE permissions. Please, refer to Android API (Application Programming Interface) documentation for the list of the permissions and their descriptions [11]. For static analysis, application permissions requested are the first natural and noticeable (i.e., prominent) feature category to be examined among the wide range of possibilities. The dynamic analysis could also take application permissions into account [12]. Requested permissions could not provide conclusive evidence that an application conducts malicious activity. However, not requested permissions could generally absolve applications from possible abuses, and some of the requested permissions are notable in most malign applications. Android application permissions have been used as a prominent feature in many ML studies on static malware analysis, some of which are reviewed in Appendix B.

Some might argue that the change in Android 6.0 (API level 23) deferring permission check from install time to run time should affect the permission feature and related studies. This ostensible change will not affect the underlying mechanism shortly. Only the permission ranks will be reordered, but the features are still discriminative from an inter-class perspective. For further information, see the Appendix reviewing Android mobile malware detection literature, explicitly focusing on application permission request features.

Mobile Application Datasets

It is observed that the related literature compares classification performances with others via performance metrics, and the researchers do not consider the similarities or dissimilarities among the datasets they used. Moreover, the literature has not explicitly compared the datasets used in those studies. Whereas the performance of supervised machine learning algorithms closely depends on the datasets used, their sample sizes, sampling quality, and class ratios. Android mobile application datasets can hold many features that can be used for comparing different datasets such as the range or distribution of the application's creation date that maybe not definite or other metadata, even the exact hash of the application samples. Nevertheless, these features could be arbitrary or manipulative, comparing permission features that are still at the core of the Android security mechanism. Hence, application permissions were chosen as a prominent feature category to compare the datasets.

AN INITIAL MANUAL ANALYSIS OF THE DATASETS

Before describing the proposed method and providing the results obtained from the case study domain, namely Android mobile malware detection, a manual analysis and comparison approach is described. Such an approach is also valuable to show the difference between the manual and the proposed method. The proposed method is then verified by a demonstration that examines and compares negative (benign) datasets and positive (malign) used in various binary classification (malware classification) studies based on binary features (application permission requests) as summarized in Table 1.

The initial manual analysis conducted in this study comprises the following two techniques:

Table 1. The aspects of demonstrating dataset comparison for the case study classification domain.

Binary Classification	*Demonstration*
Classification problem (domain)	*Android mobile malware classification*
Examples (samples)	*Android mobile applications*
Negative class label	*"Benign" application*
Positive class label	*"Malign" application or "Malware"*
Prominent binary features	*Android application permission requests* *(shortly 'application permissions' or 'permissions')*
Example binary feature	*CALL_PHONE: It allows an application to initiate a phone call without going through the Dialer user interface for the user to confirm the call.*
Binary feature values	*0: No permission is given for the application (not allowed, default)* *1: The permission is given (allowed)*
Missing values	*Datasets might have a missing value (i.e. they do not have at least one sample (application) with the specific binary feature).* *Such features are taken as default 0 (not allowed) in dataset comparisons.*
Number of features	*Minimum: 69 and maximum: 118*
Compared datasets	*Five pairs (negative/positive class) of datasets (DS_0, DS_1, DS_2, DS_3, and DS_5) and one positive-only dataset (DS_4). An aggregated dataset (DS_A) per class is also generated, as described in Section 4. The details are provided in Table 2.*

Table 2. Summary of sample and feature spaces of the benign (negative) and malign (positive) dataset.

Dataset	*Name*	*Authors and reference*	m_N	*PREV*	m_P	n_N	n_P
			Sample space			*Feature space*	
DS_0	*Touchstone Dataset*[1]	*Lindorfer et al., [13]*	*264,303*	*60%*	*399,353*	*84*	*90*
DS_1	*Contagio*	*Aswini and Vinod, [14]*	*254*	*52%*	*280*	*94*	*81*
DS_2		*Wang et al.[15]*[2]	*310,926*	*2%*	*4,868*	*83*	*69*
DS_3		*Yerima et al., [16]*[2]	*1,000*	*50%*	*1,000*	*99*	*75*
DS_4	*Android Malware Genome Project*	*Jiang and Zhou [17]*		*100%*	*1,260*		*83*
DS_5		*Peng et al., [18]*	*207,865*	*0.2%*	*378*	*118*	*73*
DS_A	*Aggregated Dataset*	$DS_1 - DS_5$	*520,045*	*1%*	*7,786*	*59*	*47*
-DS_6		*Hoffmann et al., [19]*	*136,603*		*6,187*		
-DS_7	*Contagio*	*Sarma et al., [20]*	*158,062*		*121*		
-DS_8		*Canfora et al., [21]*			*400*		
-DS_9		*Peiravian and Zhu, [22]*[2]	*1,250*		*1,260*		
-DS_{10}		*Felt et al., [23]*	*900*				

1. Original dataset name: ANDRUBIS
2. The positive-class datasets contain AMGP samples.

Basic Quantitative Comparison of Sample/Feature Spaces: The negative and positive class datasets are described based on sample space and feature space sizes. The distribution of positive/negative class ratios is another critical attribute for quantitative dataset comparisons.

Binary-Feature Space Graphical Analysis: The binary-feature space per dataset is analyzed and compared via the following attributes:

- The frequency distribution of the features that are common in all the datasets (a dataset might have a missing value, i.e. binary-feature)

- The change in top-ranked features (a bump-chart is recommended; an interactive version is also provided online).

After elaborating the manual analysis, the next sections describe the possible approaches to compare datasets (i.e. the types of the comparison activities), provides the definition and description of the proposed comparison method, and finally demonstrates the results when the method is applied to the reviewed datasets.

The Datasets

This study reviewed six academic studies providing Android mobile benign and malign datasets. These datasets are used to demonstrate some initial manual analysis techniques and the proposed comparison method. The following subsections describe each technique and present the results for the reviewed datasets.

Basic Quantitative Comparison of Sample/ Feature Spaces

Table 2 lists the basic quantitative information for the datasets and introduces the related studies that are also reviewed in Appendix A. The two dimensions, namely sample-space size (m) and feature-space size (n), are valid for any datasets, whereas prevalence (*PREV*; The proportion of total positive samples (m_P), e.g., having a malign characteristic, in total sample size [$m_P + m_N$]) is determined by comparing sample-space sizes of the positive and negative class datasets. In the related literature, it is observed that authors compare their malware classification performance with others, most of which are based on different benign and malign datasets. The method proposed in this study can help to compare those datasets. Highlighting once again, there has been no large-scale comparative study on comparing datasets used for mobile malware classification encountered in the literature. However, it was not possible to see to what extent the proposed aggregation and comparison methods can be valid. A more recent independent study is used for assessing validity. Lindorfer et al. [13] presented their findings based on a dataset collection called "ANDRUBIS" from a wide range of sources.

The DS_0 dataset listed in the first row in Table 2 has not only a higher number of samples but also the highest number of malware (positive-class examples) compared with other datasets. Thus, it was selected as a kind of correctness measure that is called 'touchstone' in this study, to support verifying the comparisons. In this study, the permission frequencies in DS_1 to DS_5 datasets per class were also aggregated into single combined values. The aggregated dataset, named DS_A is used to search for their consistencies among the datasets and to provide a baseline for further research. The aggregated frequencies are calculated by the weighted arithmetic mean of frequencies in individual datasets according to dataset sample sizes per class, as explained in Section 4 in detail. This is a natural calculation approach considering combining all the datasets into one dataset named DS_A (ignoring the duplicate samples due to the same samples existing in one or more datasets). Note that the aggregated dataset (DS_A) and the touchstone dataset (DS_0) are entirely different and independent.

Note that two published datasets were combined, one from 2011 and one from 2012 in [18] into one dataset (DS_5). The six datasets (DS_6 – DS_{11}) encountered in the literature were excluded from this study due to the following reasons. The DS_9 dataset [22] is the same as the original DS_4 dataset [17]. The datasets DS_8 [21], DS_{10} [23] have missed one class. Only the top ten permissions were published for DS_6 [19], and only the top 20 permissions were published for DS_7 [20], but the whole feature space could not be obtained for this study.

Binary-Feature Space Graphical Analysis

As seen in Table 2, dataset sample sizes, prevalence, and feature space sizes of the datasets are dispersed. Sample sizes and equal class sample sizes (i.e. near 50% prevalence) are critical for generalization and unbiased classification. The low number of samples and low prevalence rates also cause limited credibility in the literature. The feature-space sizes and elements (permissions existing in each dataset) are also different in Table 2. Moreover, frequencies and ranks of permission requests vary from dataset to dataset.

Binary-feature frequency distribution

Fig. 1 shows the frequency distribution of the prominent binary features in negative and positive-class datasets together in one graphic, including only the common features (i.e. the permissions existing in all the datasets per class). The lower left part shows the distribution for the positive-class, while the upper right part is for the negative-class in reverse order of binary-feature frequency. The permissions within five datasets (from DS_1 to DS_5) and aggregated dataset (DS_A) are sorted according to the touchstone dataset's (DS_0) permissions with descending frequency order of corresponding class. Fig. 1 also exhibits a discrepancy between the datasets per class when the permissions are ordered according to DS_0. The proposed method helps to assess the discrepancy, as explained in the next sections.

Nevertheless, interpreting Fig. 1, the following findings were deduced:

- Negative-class datasets, except for dataset DS_1 having very few samples, are more similar to the touchstone dataset than positive-class ones.

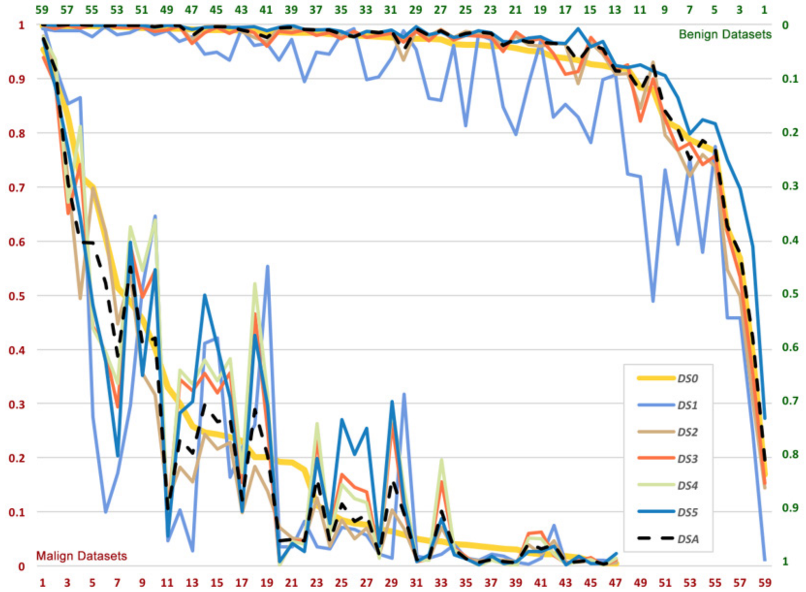


Figure 1. Binary-feature frequency distribution: Lower-Left Group: Frequency distribution of 47 common permissions in positive-class datasets and Upper-Right Group: Frequency distribution of 59 common permissions in negative-class datasets. Common permissions are the intersection of all datasets per class and sorted according to the corresponding touchstone dataset (DS_0, with thicker gold colored lines).

- The distribution of aggregated datasets (DS_A) seems closer to the touchstone dataset (DS_0) than individual datasets.

The first finding suggests that positive classes (generally abnormal entities like malware in provided applications or illness for a medical classification or diagnosis test) possess high variability (or entropy). The second one implies that the aggregation of different datasets reduces noise and enhances sampling. Concerning the first finding, this is especially valid for the example domain where malware propagating by repackaging benign applications are the most common ones that request one or more extra permission from benign ones [24]. For the second finding, as seen in the dataset DS_1 example, the low number of samples does not provide sufficient generalization; therefore, they should be used with caution in machine learning applications.

The top binary feature ranks

Fig.s 2 and 3 show the changes in the ranks of permissions between DS_0, DS_1, …, DS_5 for positive and negative-class datasets, respectively, for the top 15 permissions only (for the sake of simplicity). The readership is encouraged to visit http://tabsoft.co/32CQGIP for interacting with the online chart prepared for this study in full-intersected permission space coverage.

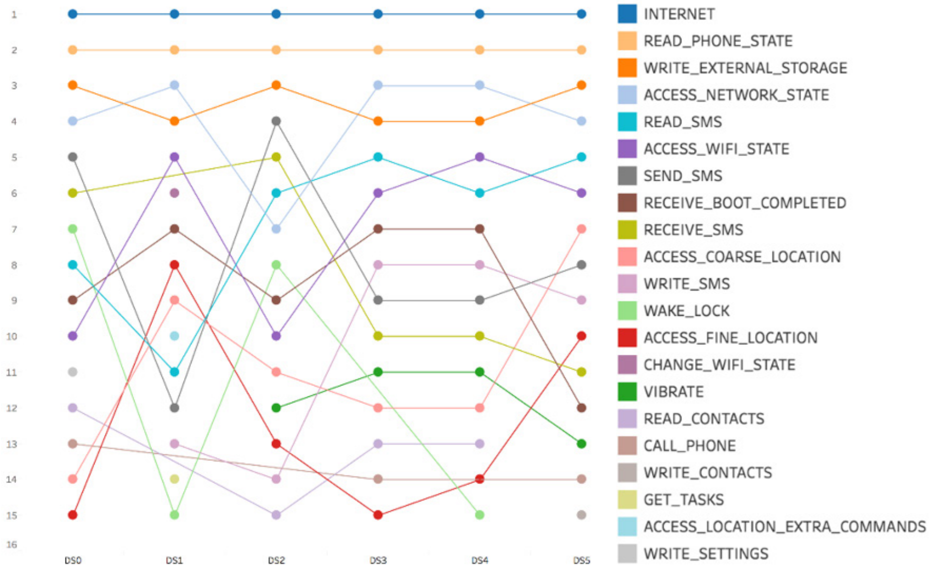


Figure 2. Ranked top 15 permissions for positive-class (malign) datasets (from DS_0 to DS_5). Visit http://tabsoft.co/32CQGIP for full data and an interactive chart.

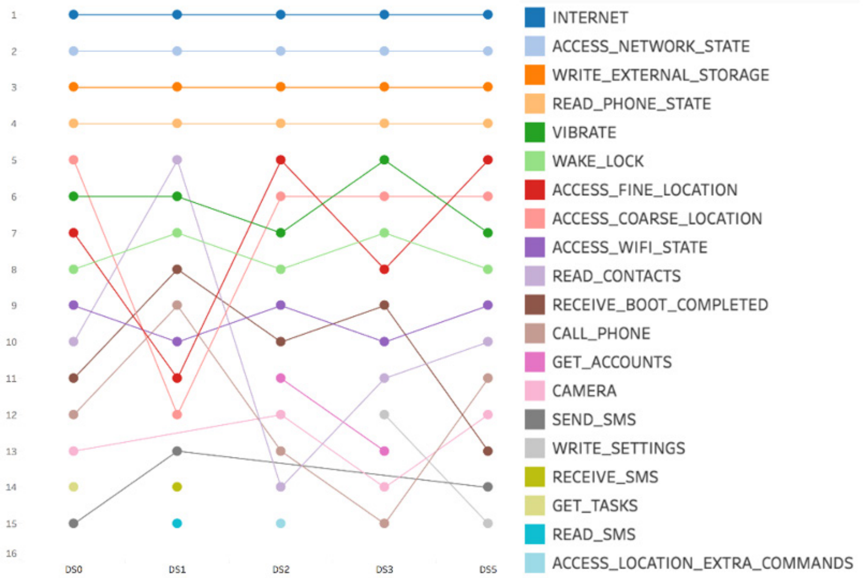


Figure 3. Ranked top 15 permissions for benign datasets (from DS_0 to DS_5)

Please, refer to Android API documentation for descriptions of the permissions (at developer.android.com). Considering the top 15 permissions in positive-class datasets, Fig. 2 shows DS_3 with DS_4, and DS_4 with DS_5 are relatively similar rankings for corresponding features (permissions). Using an interactive chart hovering on permissions (circles) in the DS_0 dataset's column, you can see that DS_0 with DS_5 are also similar rankings (although they are not adjacent).

Concerning negative-class datasets, Fig. 3 shows DS_2 with DS_5 and DS_0 with DS_5 are relatively similar rankings considering the top 15 permissions. If positive-class (Fig. 2) and benign-class (Fig. 3) feature ranks are compared, the top two permissions are the same in all the malign datasets while the top four ones in benign datasets. This supports the interpretation of high variability in malign datasets in Fig. 1 above. These two types of graphs help to analyze and compare datasets, but it is manual and may be subjective. Therefore, it is necessary to measure similarities that provide more accurate results.

METHODS

Fig. 4 describes the general methodology followed in this study. The permissions were collected directly from different negative and positive-class datasets of the related six studies. Some of the authors were contacted to receive their datasets covering all the permission requests (i.e. full feature space for a dataset). After pre-analyzing the permission request features, their frequencies (i.e. ratio of the number of samples requesting permission to total sample size) were calculated for each class, and binary features were ranked according to these frequencies per each dataset from the most frequent to the least frequent.

For a dataset with *c* binary class (positive (*P*) or negative (*N*)), the existing n_c binary features $\{x_1, x_2, \ldots, x_{n_c}\}$ are presented as *X* vector. $f_{X_c DS_i}$ denotes binary-feature frequencies vector for i. dataset. $F_{X_c DS_i}$ denotes ranked feature-frequencies vector and holds ranks within the same datasets instead of frequencies. The ranked feature-frequencies vector for the aggregated dataset (DS_A) per each class was calculated by applying a weighted average of feature frequencies in each dataset (from DS_1 to DS_5) and ranked from top to bottom as shown in Eq. (1) where m_{P_i} and m_{N_i} denote the total sample size of i. dataset per *c* class, and S_c is the number of datasets compared.

$$F_{X_{c=P,N} DS_A} = \text{rank}\left(\frac{\sum_{i=1}^{S_c} f_{x_c DS_i} \cdot m_{c_i}}{\sum_{i=1}^{S_c} m_{c_i}}\right) \qquad (1)$$

The ranked binary-feature frequencies per negative and positive classes are compared between:

- (Comparison-1) all the dataset including the touchstone dataset (DS_0) and the aggregated dataset (DS_A)
- (Comparison-2) pair of all the datasets (e.g., between DS_1 and DS_A or DS_1 and DS_0)

The results of the two comparisons on the reviewed datasets are given in Section 6.

NEW METHOD: COMPARISON VIA ADAPTED KRUSKAL-WALLIS TEST

The Kruskal-Wallis test is a nonparametric test to calculate the null hypothesis assuming that independent samples are from the same population. The test, which was developed by and named after Kruskal and Wallis [25], is an extension of the Wilcoxon Rank Sum Test on two groups. As a nonparametric test, the Kruskal-Wallis test does not assume that populations have normal distributions. The test is applicable for measurement variables as well as nominal variables classifying observation values into discrete categories (like binary features) among at least three or more samples.

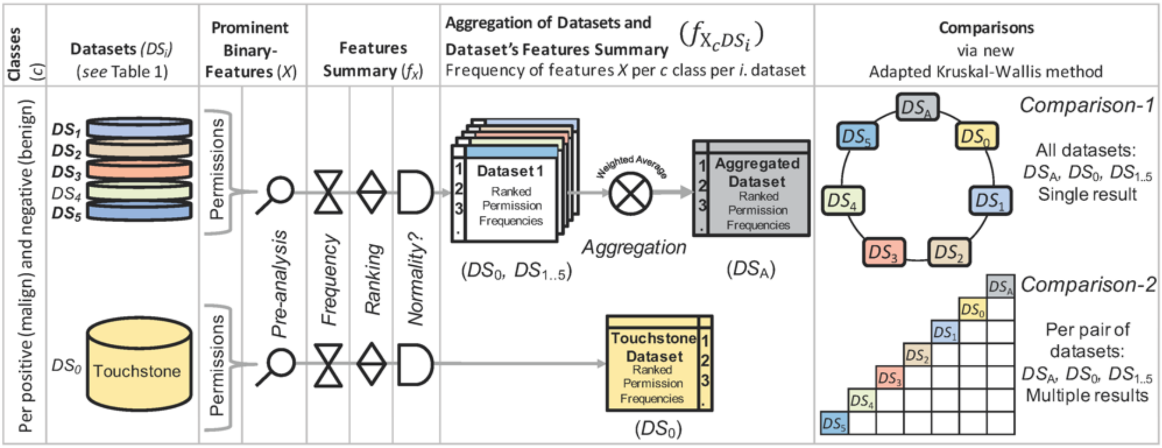
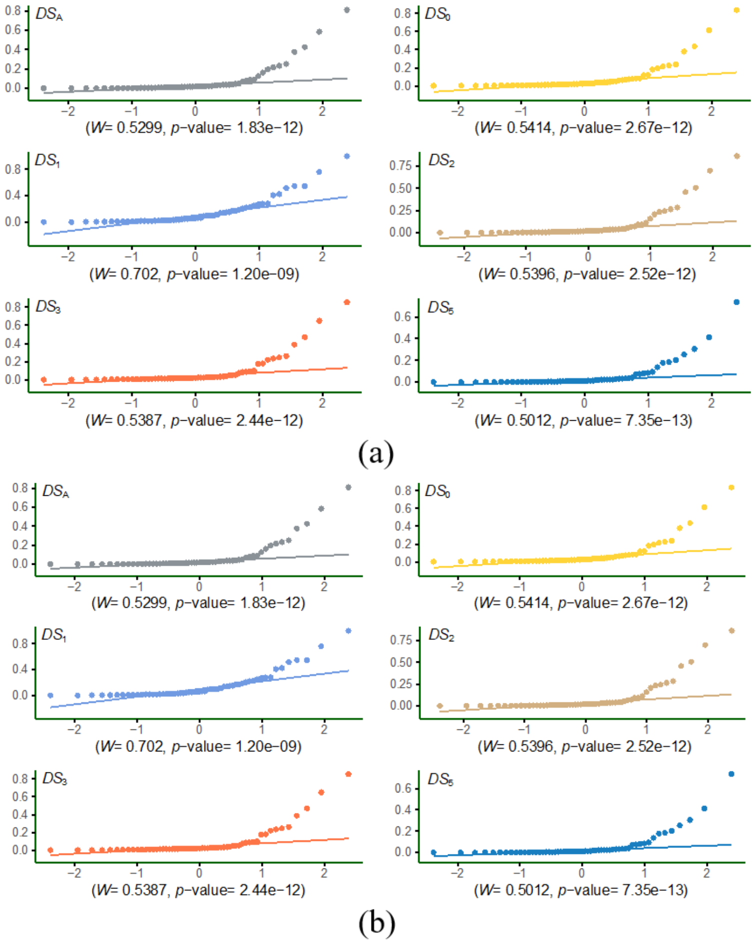

Classes (c)
Datasets (DSi) (see Table 1)
Prominent Binary-Features (X)
Features Summary (fX)
Aggregation of Datasets and Dataset's Features Summary (fXcDSi)
Frequency of features X per c class per i. dataset
Comparisons via new Adapted Kruskal-Wallis method
Per positive (malign) and negative (benign)
DS1
DS2
DS3
DS4
DS5
DS0
Touchstone
Permissions
Permissions
Pre-analysis
Frequency
Ranking
Normality?
Dataset 1 Ranked Permission Frequencies
1
2
3
Weighted Average
Aggregation
(DS0, DS1..5)
Aggregated Dataset Ranked Permission Frequencies
(DSA)
Touchstone Dataset Ranked Permission Frequencies
(DS0)
Comparison-1
DSA
DS0
DS1
DS2
DS3
DS4
DS5
All datasets: DSA, DS0, DS1..5 Single result
Comparison-2
Per pair of datasets: DSA, DS0, DS1..5 Multiple results

Figure 4. Activity flows for comparing datasets via feature frequency ranks.

This test is based on ranks instead of the original observation values (i.e. frequencies). This makes the test much insensitive to outliers that make it more suitable for this experimental study on the negative (benign) and positive (malign) mobile applications like other practical research studies such as clinical ones [26]. The ranks are calculated across all the samples by ordering the observation values from smallest (a rank of 1) to largest and could be fractional. The sum of the ranks per sample is also calculated.

Typical usage of the Kruskal-Wallis test in machine-learning is using as a filtering method for feature selection in high-dimensional datasets [27,28]. It is appropriate for not only binary classification but also multi-class classification problems [29]. The literature has successfully used the test on analyzing and comparing data with different characteristics, for example, censored data [30] and microarray gene expression data [31], but also addressed the limitations when applied in high dimensional low sample size data (shallow datasets) [32]. Another usage of the Kruskal-Wallis test, along with the one-way analysis of variance test, Friedman's test, in ML is in testing the statistical significance between the different individual classifiers (i.e. whether a classifier is significantly different from the others) [33]. The significance in algorithm factors or parameters such as the data-size effect or fitness values is also tested with the Kruskal-Wallis test [34,35]. From an information security perspective, the test was used for evaluating different alternatives, such as measuring differences in password behaviors and attitudes between research participants [36] or selecting more discriminative features in the forensic analysis [37]. It was encountered that only one study uses the Kruskal-Wallis test in malware analysis in the literature. Asmitha and Vinod [38] employ the test for selecting prominent features from benign and malign applications on the Linux desktop platform. According to their classification experiment, the Kruskal-Wallis test achieves slightly better than the other feature selection methods. The review reveals that the literature uses the test in comparing the dataset's features and classifier's performances. However, it is not used to compare datasets. This study explores and proposes such usage demonstrated in real-world datasets in a specific domain.



(a)

(b)

Figure 5. Normality check by Quantile-Quantile chart with Shapiro-Wilk test values and p-values. y-axis shows binary-feature frequencies for (a) positive-class datasets (b) negative-class datasets. Note that some of the frequency values (points) are outside the corresponding normal distribution indicated by a shaded area.

Normality Check

Before applying the proposed adapted Kruskal-Wallis test, we must ensure that the frequencies do not present a normal distribution [39]. If a normal distribution exists, the distribution can be entirely defined by using merely two parameters: mean and standard deviation, which may be used for dataset comparison statistically instead of this method.

Two supportive approaches are employed for checking normality:

- A formal method by using the Shapiro-Wilk Test

- A manual method by drawing Quantile-Quantile charts

Eq. (2) is the Shapiro-Wilk test explicitly written for binary-classification datasets where a_j normalized standard normal-order statistics and $\overline{f_{X_c DS_i}}$ is the mean value for an *i.* dataset:

$$W_{DS_i} = \frac{\left(\sum_{j=1}^{n_{cmn}} aj \cdot f_{x_c DS_{ij}}\right)^2}{\sum_{j=1}^{n_{cmn}} \left(\mathrm{f}_{\mathrm{X_c DS_{ij}}} - \overline{\mathrm{f}_{\mathrm{X_c DS}_i}}\right)^2} \tag{2}$$

W is between 0 and 1, and lower *W* values against the corresponding test table value indicate the rejection of the normality null hypothesis. Fig. 5 shows not only the quantile-quantile chart but also the Shapiro-Wilk test values with *P* probability values (*p*-values) for each dataset in x-axes.

Lower *W* values, or better specifically, lower corresponding *p*-values (less than 0.05 for 95% significance level), reject the normal distribution. Here we have *p*-values that are even very close to zero (more than 99% significance level). Note that the original Shapiro-Wilk test is suitable for less than 50 observations. In this study, Royston's [40] extension is used here to avoid such a limit. Benign and malign datasets have 59 and 47 common (intersected) feature-space sizes (n_{cmn}). Ensuring non-normality, the test can be employed as described in the following subsection.

Adapted Kruskal-Wallis Test

In the standard notation, given *C* samples with *N* number of total observations in all samples combined, with n_i observations yielding the sum of the ranks as R_i in the *i.* sample, the Kruskal-Wallis Test value (*H*) is calculated by the following equation:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{C} \frac{R_i^2}{n_i} - 3(N+1) \tag{3}$$

Eq. (4) has specifically annotated for the reviewed dataset comparisons where S_c is the total number of datasets in this study (seven for positive, six for negative, including aggregated dataset DS_A). *N* in Eq. (3) corresponds to the total of samples' common (intersected) feature-space size (n_{cmn}) (7x59 for negative-class, 6x47 for positive-class). R_i corresponds to $\mathrm{rank}(f_{X_C D_i})$, the sum of binary-feature frequencies ranks in the *i.* dataset. Rank orders are determined within all the datasets as if there is one dataset where the lowest value corresponds to the lowest rank. Fractional ordering is used for ties by averaging orders.

$$H = \frac{12}{S_c \cdot n_{cmn} \cdot (\mathrm{S_c} \cdot \mathrm{n_{cmn}} + 1)} \sum_{i=1}^{S_C} \frac{\mathrm{rank}(f_{p_C DS_i})^2}{n_{cmn}} - 3(\mathrm{S_C} \cdot \mathrm{n_{cmn}} + 1) \tag{4}$$

Low *H* values or low *p*-value as an approximate chi-square statistic (with S_c – 1 the number of degrees of freedom, DoF) in the range [0, 1] rejects the null hypothesis that independent samples are from the same population.

RESULTS

The following subsections provide comparison results for all the datasets together (Comparison-1) and per pair of all the datasets (Comparison-2).

Comparison-1 (All)

The proposed adapted Kruskal-Wallis test was conducted for all the permission frequencies in negative and positive-class datasets (touchstone, aggregated, and four negative-class or five positive-class datasets, respectively) listed in Table 2. The conducted test produced two different results per class. Table 3 displays the summary of the test. The *p*-values less than the significance level (α = 0.05) reject the null hypothesis that the samples in negative-class datasets are from the same population concerning ranks of the frequencies of the same permission features or "negative-class datasets are different from

Table 3. Dataset comparison summary based on adapted Kruskal-Wallis method.

Class (c)	n_{cmn}	S_c *(DoF)*	*H*	*p-value*	*Test Result**
Positive (Malign)	*47*	*Seven datasets (6)*	*2.45*	*0.8735*	*Failed to reject the null hypothesis*
Negative (Benign)	*59*	*Six datasets (5)*	*27.84*	*3.92e-05*	*Rejected*

* Significance level, α = 0.05

each other". In comparison, we could not conclude if the positive-class datasets are different, although the *p*-value is close to 1. The alternative hypothesis indicating "dissimilarity" assumes that at least one dataset comes from a different population than the others.

The *H* value obtained by Eq. (4) is merely for stating whether the group of datasets together differs in some way. This is important because one could not express this evidently by analyzing and comparing the samples as tried in Section 3.3 via different graphs. However, the dissimilarity of individual datasets should also be interpreted separately afterward.

Comparison-2 (Pairs) with Suggested Visualization Techniques

Comparison-2 shows the similarity test per pairs of the dataset. Instead of giving the results in a cross-tabular fashion, three visualization techniques are recommended:

1) Multiple comparisons of mean ranks

2) All-in-one binary-feature frequency descriptive statistics

3) Complete clustered pairwise comparison of *p*-values.

The suggested visualization techniques demonstrated in Fig.s 6 and 7 are straightforward, informative, and easy to interpret.

Visualization-1 (Multiple comparisons of mean ranks)

The first visualization technique depicts the pairwise comparison of the datasets based on rank means calculated by the Kruskal-Wallis test. The graph is developed by using MATLAB's multi compare functionality [41]. The interactive version of the graph shows the mean rank difference between a selected dataset and the others. The findings of the multiple comparisons of mean ranks to be highlighted are

- "No positive-class datasets have mean ranks significantly different from the aggregated positive-class dataset (DS_A)," as shown in Fig. 6 (b) (Kruskal-Wallis test can reject the null hypothesis even the means or medians are the same. Therefore, p-values are valid.).

- The same findings are not valid for negative-class datasets. However, four datasets, including the aggregated dataset (DS_A), have mean ranks significantly different from the benign (DS_1) dataset, as shown in Fig. 7 (b).

- Interestingly, mean ranks are not significantly different for DS_1 and the touchstone dataset DS_0.

Visualization-2 (All-in-one binary-feature frequency descriptive statistics)

Violin with a box-plot comparison diagram in Fig.s 6 and 7 (b) show the following binary-feature frequency descriptive statistics for negative and positive-class datasets:

- ranges (min/max values shown in vertical line ends),

- quartiles (lower and upper shown in the bottom and top edges of boxes),

- medians (horizontal line in box),

- means (black dot),

- outliers (pink dot), and

- probability densities (violin shape).

The significant difference of negative-class DS_1 and no-significance difference among positive-class datasets can be observed in Visualization-2 graphs (see the shapes of the violins). Note that DS_1 has the smallest samples for both classes.

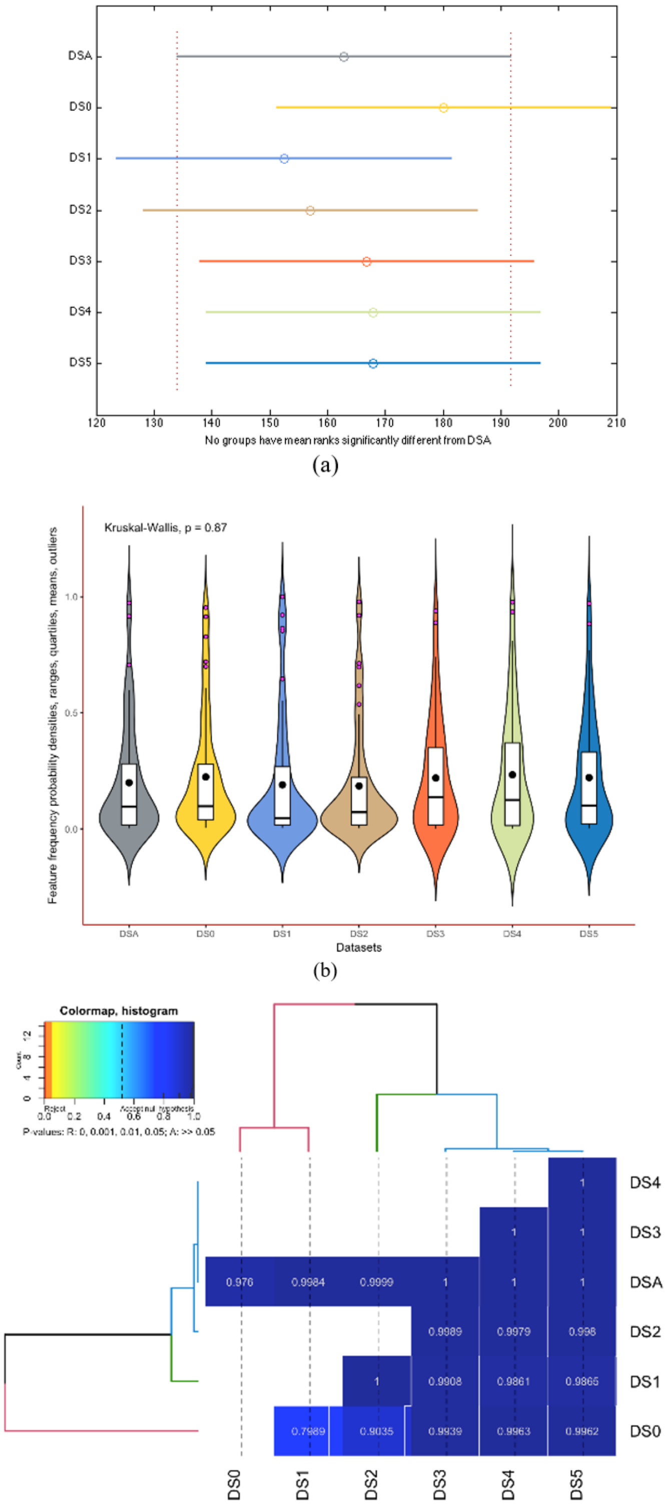
Visualization-3 (Complete clustered pairwise comparison of p-values)

The third visualization technique that is originally designed as an API in R by the author. The API displays the *p*-values for all the pairs of datasets. Pairwise dataset comparisons with heatmap diagrams in Fig.s 6 and 7 (c) present a complete set of comparison information. It shows colored *p*-values for the null hypothesis indicating similarity between the paired datasets. Datasets are also hierarchically clustered by Euclidean distances of *p*-values (i.e. their similarities). In other words, the datasets in row/columns are reordered according to row or column means and then hierarchically clustered using Euclidean distance. A similar group of datasets is shown as horizontal and vertical dendrograms.

The findings complying with the Comparison-1 shown in Table 3 are

- We could not reject the null hypothesis that each pair of the positive-class datasets are from the same population with ultimately high p-values. DS_0 and DS_1 have 0.7989 *p*-values at a minimum.

- The following dataset pairs are significantly different from each other: DS_A vs. DS_1 (with p-value: 8e-04), DS_A vs. DS_5 (with p-value: 0.00001), DS_1 vs. DS_2 (with p-value: 0.0021), and DS_1 vs. DS_3 (with p-value: 0.0361). For others, we could not reject the null hypothesis.

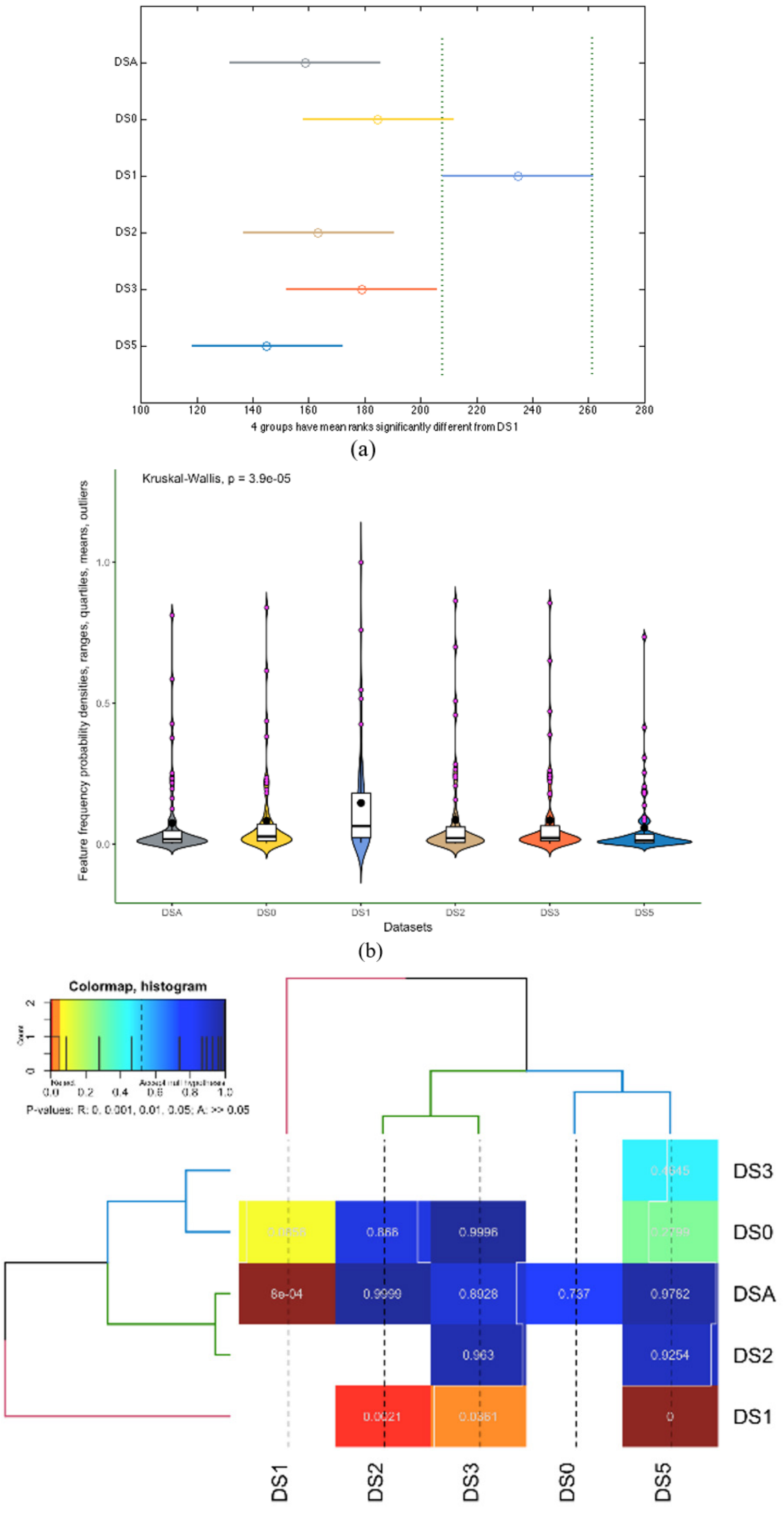


(a)

(b)

(c)

Figure 6. Comparison graphs for malign datasets: (a) multiple comparisons of mean ranks (graph shows DS_A comparison) (b) violin with a box-plot comparison diagram (c) Pairwise dataset comparisons with heatmap diagram



(a)

(b)

(c)

Figure 7. Comparison graphs for benign datasets: (a) multiple comparisons of mean ranks (graph shows DS_1 comparison) (b) violin with a box-plot comparison diagram (c) Pairwise dataset comparisons with heatmap diagram

Touchstone vs. Aggregated Datasets Comparison

Overall assessment of the test results suggest the following two highlighted findings of touchstone and aggregated datasets:

- For the comparison of the touchstone and aggregated datasets: There is no significant difference between the datasets DS_0 and DS_A. Rank means the difference between these datasets is 17.2 (162.9 and 180.1) for positive-class and 26.1 (158.6 and 184.7) for negative-class, as shown in Fig.s 6 and 7 (b).

- For the comparison between each dataset and the aggregated dataset (DS_A): Fig.s 6 and 7 (b) show that the malign datasets are more similar to the aggregated dataset than the touchstone dataset. Considering the touchstone dataset (DS_0), the DS_4, DS_5, and DS_3 malign datasets and DS_3 and DS_2 are the most similar datasets to the touchstone dataset so that their sampling approaches are quite successful.

DISCUSSION

Two aspects addressed in this study are discussed in this section: first the issues and findings specific to the case study domain and the prominent feature category, second, the issues related to the proposed comparison method.

Firstly, permission requests are leading clues to anticipate the purpose of Android applications not only for regular users but also for malware analysts who use them as a prominent feature category to classify malware. A fundamental problem with much of the literature on mobile malware classification on the Android platform is that they use different datasets and focus on the results of their classification. However, the comparison of the datasets has not been dealt with in-depth.

Comparison of performances of malware classification attempts with various ML algorithms cannot be consistent without knowing the difference of the used datasets. To study this gap, this study has compared the permissions ranked by request frequencies of different datasets of the seven reviewed academic works. The ANDRUBIS dataset, as it is called the “touchstone” dataset in this study, was used as a verification dataset for comparing the similarity of binary-feature (permissions) frequencies of individual datasets.

This study has conducted a focused review of the literature and highlighted the different issues around permissions to classify Android mobile malware. In summary, it is concluded that;

- The Android permissions and frequency of permission requests do continue to hold its invaluable contribution to statically classify Android applications as long as they are selected comparatively and continuously updated;

- Satisfactory results were obtained showing that frequently requested permissions extracted benign/malign applications, as well as the permissions dominantly requested by malign applications, should be the first statistical features to examine for static malware analysis and dynamic analysis further;

- Comparing the performance of malware classification, the published research should consider the comparison of their datasets and others;

- Authors could use the proposed dataset comparison method and initial manual analysis approaches to compare their datasets with others easily. The permission-requests feature distribution could also be used as an indicator to examine datasets;

- Reducing the number of top permissions that are considered may provide more accurate comparison statistics; and

- The characteristics of the feature used for comparison, especially the factors affecting its frequency, should be scrutinized (as discussed in Appendix A). Eliminating this kind of external effect makes comparisons more accurate.

The followings are the summary of the overall findings in the conducted test on the case study domain:

- Further evidence has been provided on the effect of good sampling of negative-class (benign applications) and positive-class (malign or malware) datasets in static malware analysis research in the literature, which pointed towards the idea that even a small number of well-selected datasets could present a sufficient level of representation comparing the touchstone dataset.

- There is still a need for continuously updating samples to adapt to the existing trends in benign and malign applications.

Secondly, concerning the proposed comparison method, the Kruskal-Wallis test was conducted with a completely different approach. The test is typically applied through a single ordinal variable (apart from categorical or interval variables), for example, “levels of blood cholesterol” with different observations in more than two samples. For the proposed approach, the frequencies of the specific number of the same binary features, namely Android mobile application permission requests, are used as the observations in each dataset. In this manner, it is possible to create a kind of 'imitated' ordinal variable per dataset that could be expressed as 'the frequency of any binary feature of a specific number of requested permissions in the compared dataset.' The datasets were compared by using this variable. The comparison via binary-feature frequencies by this method has the following advantages:

- It provides a single metric (a test value (H) with easy to interpret p-value indicator) for similarity among datasets.

- This test also shows the similarity positions for all datasets without pairwise comparisons, which could be time-consuming and hard to analyze.

The method does not require any preference for the choice of parameter settings (except default significance level); therefore, it can be used as-is. The comparison does not need the feature-space details of all the samples in the dataset; the frequencies of the prominent binary features are sufficient. This is practical considering the difficulties or obstacles in sharing the datasets. The provided API facilitates the comparison process providing results and generating graphs for the recommended visualization techniques. The results that were reported from the complete perspective in this study are promising. The subject matter experts can find the methodology convenient and insightful. At least, the method addresses the dissimilarity among the datasets allowing the researchers and experts to focus. Nevertheless, theoretical validation cannot be found; therefore, more simulations should be conducted. The future work will be validating the method in synthetic datasets.

This study also includes comparing the individual datasets with the aggregation of the datasets. Aggregating compared datasets spots the missing frequent and rare patterns in samples. Thus, adding different samples having those missing patterns could improve the overall sampling quality of a dataset in hand.

Regarding the novel adaptation of the Kruskal-Wallis test, there could be some controversy surrounding the imitation of the ordinal variable. Instead of using values of a single variable from different observations for each sample (e.g., INTERNET permission request frequencies observed per dataset), using the values of a group of variables from different observations may seem unconventional. However, it becomes more understandable and valid for the test when the variable is stated as "the binary-feature frequency values of a specific group of observations". Upon suggesting this approach, other studies in different domains could try the usability of the methods.

Limitations comparison of the datasets over common features seems to discard the real differences among datasets. In this case, the missing values (i.e. nonexistent features) should also be reported in the comparisons. Nevertheless, as the datasets become large, having at least one sample per feature, the comparison over common features becomes more representative.

The comparison approaches and the proposed method has been demonstrated in real-world datasets. The manual analysis generally supports the results. Furthermore, the fact that the malign DS_2 and DS_3 datasets have the same samples as the malign DS_4 (Android Malware Genome Project) dataset is also validated via the clustered complete pairwise comparison of p-values in Comparison-2 (DS_3 and DS_4 in one dendrogram, which is then in the upper dendrogram with DS_2, as shown in vertical dendrograms in Fig. 6).

CONCLUSION

The researchers mostly focus on selecting and optimizing ML classification algorithms and improving the achieved performance expressed in terms of conventional performance metrics such as accuracy and F1. Selecting and maintaining a dataset is a secondary concern for not only classification problems but also clustering problems. Both in practice and the literature, performance metrics are the only criteria to claim success or improvement in a specific classification problem domain whereas the datasets are not taken into account in comparison of different studies.

The initial manual analysis of datasets demonstrated in Section 3 provides little insight and requires efforts for preparing summary data and related graphics. Basic quantitative comparison of sample and feature spaces presents the preliminary perspective in compared datasets whereas binary-feature space graphical analysis provides more detail. Especially, feature ranks are more understandable to readers; however, the approximation used on calculating the ranks according to the frequencies decreases. The precision, related calculations, and analyses are simplified.

To help to avoid such inefficiencies in the manual analysis of datasets, this study proposed a novel adaptation of the Kruskal-Wallis test. In the proposed method, instead of providing a single ordinal variable, a kind of variable was created, indicating the frequencies of the binary features. The features are selected from the intersection of existing features in all of the compared datasets. Each of those frequencies is provided as if they are the observations per dataset. Then the tests are conducted based on these variables in the case study domain. It is observed that the results of manual analysis for the case study domain and the proposed method are coherent. Although the method and approaches provided in this study were applied to the mobile malware domain, they could be used in other domains having a binary-feature space vector.

The demonstration in the case study domain has shown that the method gives clear and measurable initial insights to see the differences among available datasets. The researchers can publish the dataset comparison test results among their dataset and the other datasets along with the classification performance metrics. The method can also be particularly useful for the practitioners and researchers to compare different open ML datasets provided in different platforms such as Kaggle. It can be used in data mining, data quality, and data profiling activities. The provided API given in Appendix A supports the possible future uses of the method. Finally, it is expected that the proposed comparison method and findings potentially lead to practical improvements in dataset collection, sampling, profiling, and mobile malware analysis and provide a measurable indicator for comparing the used and related datasets.

CONFLICT OF INTEREST

Author approve that to the best of their knowledge, there is not any conflict of interest or common interest with an institution/organization or a person that may affect the review process of the paper.

References

1. Canbek G, Sagiroglu S, Taskaya Temizel T, Baykal N., Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights, in: 2017 International Conference on Computer Science and Engineering (UBMK), IEEE, Antalya, Turkey, 2017: pp. 821–826. doi:10.1109/UBMK.2017.8093539.
2. Ostertagová E, Ostertag O, Kováč J., Methodology and Application of the Kruskal-Wallis Test, Applied Mechanics and Materials. 611 (2014) 115–120. doi:10.4028/www.scientific.net/AMM.611.115.
3. Piringer H, Berger W, Hauser H., Quantifying and comparing features in high-dimensional datasets, in: Proceedings of the International Conference on Information Visualisation, IEEE, London, 2008: pp. 240–245. doi:10.1109/IV.2008.17.
4. Canbek G, Sagiroglu S, Taskaya Temizel T., New techniques in profiling big datasets for machine learning with a concise review of Android mobile malware datasets, 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT). (2018) 117–121. doi:10.1109/ibigdelft.2018.8625275.
5. Andrade RO, Yoo SG., Cognitive security: A comprehensive study of cognitive science in cybersecurity, Journal of Information Security and Applications. 48 (2019) 1–13. doi:10.1016/j.jisa.2019.06.008.
6. Canbek G, Sagiroglu S, Baykal N., New comprehensive taxonomies on mobile security and malware analysis, International Journal of Information Security Science (IJISS). 5 (2016) 106–138. http://www.ijiss.org/ijiss/index.php/ijiss/article/view/227.
7. Surendran R, Thomas T, Emmanuel S., A TAN based hybrid model for android malware detection, Journal of Information Security and Applications. 54 (2020) 1–11. doi:10.1016/j.jisa.2020.102483.
8. Clement J., Average number of new Android app releases via Google Play per month as of May 2020, New York, 2020. https://www.statista.com/statistics/276703/android-app-releases-worldwide.
9. Suarez-Tangil G, Tapiador JE, Peris-Lopez P, Ribagorda A., Evolution, detection and analysis of malware for smart devices, IEEE Communications Surveys & Tutorials. 16 (2014) 961–987. doi:10.1109/SURV.2013.101613.00077.
10. Deypir M, Horri A., Instance based security risk value estimation for Android applications, Journal of Information Security and Applications. 40 (2018) 20–30. doi:10.1016/j.jisa.2018.02.002.
11. Android, Manifest.permission, Android Developers. (2020). https://developer.android.com/reference/android/Manifest.permission.html (accessed September 2, 2020).
12. Cen L, Gates C, Si L, Li N., A probabilistic discriminative model for Android malware detection with decompiled source code, IEEE Transactions on Dependable and Secure Computing. 12 (2015) 400–412. doi:10.1109/TDSC.2014.2355839.
13. Lindorfer M, Neugschwandtner M, Weichselbaum L, Fratantonio Y, Van Der Veen V, Platzer C., ANDRUBIS - 1,000,000 apps later: a view on current Android malware behaviors, in: 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), Wroclaw, Poland, 2014: pp. 3–17.
14. Aswini AM, Vinod P., Droid permission miner: Mining prominent permissions for Android malware analysis, in: The 5th International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), IEEE, Bangalore, India, 2014: pp. 81–86. doi:10.1109/ICADIWT.2014.6814679.
15. Wang W, Wang X, Feng D, Liu J, Han Z, Zhang X., Exploring permission-induced risk in Android applications for malicious application detection, IEEE Transactions on Information Forensics and Security. 9 (2014) 1828–1842. doi:10.1109/TIFS.2014.2353996.
16. Yerima SY, Sezer S, McWilliams G., Analysis of Bayesian classification-based approaches for Android malware detection, IET Information Security. 8 (2014) 25–36. doi:10.1049/iet-ifs.2013.0095.
17. Jiang X, Zhou Y., Android Malware, Springer, Raleigh, NC, USA, 2013.
18. 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: 19th Conference on Computer and Communications Security (CCS), ACM, New York, New York, USA, 2012: pp. 241–252. doi:10.1145/2382196.2382224.
19. Hoffmann J, Ussath M, Holz T, Spreitzenbarth M., Slicing droids: Program slicing for smali code, in: SAC ’13 Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 2013: pp. 1844–1851. http://dl.acm.org/citation.cfm?id=2480706 (accessed October 22, 2013).
20. Sarma B, Li N, Gates C, Potharaju R, Nita-Rotaru C, Molloy I., Android permissions: A perspective combining risks and benefits, in: 17th Symposium on Access Control Models and Technologies (SACMAT), ACM, New York, New York, USA, 2012: pp. 13–22. doi:10.1145/2295136.2295141.
21. Canfora G, Mercaldo F, Visaggio CA., A classifier of malicious Android applications, in: The 8th International Conference on Availability, Reliability and Security (ARES), IEEE, Regensburg, 2013: pp. 607–614. doi:10.1109/ARES.2013.80.
22. Peiravian N, Zhu X., Machine learning for Android malware detection using permission and API calls, in: IEEE 25th International Conference on Tools with Artificial Intelligence

(ICTAI), IEEE, Herndon, VA, 2013: pp. 300–305. doi:10.1109/ICTAI.2013.53.

23. Felt AP, Chin E, Hanna S, Song D, Wagner D., Android permissions demystified, in: Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS), ACM Press, New York, New York, USA, 2011: p. 627. doi:10.1145/2046707.2046779.
24. Canbek G, Baykal N, Sagiroglu S., Clustering and visualization of mobile application permissions for end users and malware analysts, in: The 5th International Symposium on Digital Forensic and Security (ISDFS), IEEE, Tirgu Mures, 2017: pp. 1–10. doi:10.1109/ISDFS.2017.7916512.
25. Kruskal WH, Wallis WA., Use of Ranks in One-Criterion Variance Analysis, Journal of the American Statistical Association. 47 (1952) 583–621. http://www.jstor.org/stable/pdf/2280779.pdf?_=1463988119080.
26. Theodorsson-Norheim E., Kruskal-Wallis test: BASIC computer program to perform nonparametric one-way analysis of variance and multiple comparisons on ranks of several independent samples, Computer Methods and Programs in Biomedicine. 23 (1986) 57–62. doi:10.1016/0169-2607(86)90081-7.
27. Bommert A, Sun X, Bischl B, Rahnenführer J, Lang M., Benchmark for filter methods for feature selection in high-dimensional classification data, Computational Statistics and Data Analysis. 143 (2020) 1–19. doi:10.1016/j.csda.2019.106839.
28. Vora S, Yang H., A Comprehensive Study of Eleven Feature Selection Algorithms and their Impact on Text Classification, in: Computing Conference, London, United Kingdom, 2017: pp. 440–449. doi:10.1109/SAI.2017.8252136.
29. Boulesteix AL, Tutz G., Identification of interaction patterns and classification with applications to microarray data, Computational Statistics and Data Analysis. 50 (2006) 783–802. doi:10.1016/j.csda.2004.10.004.
30. Chen Y, Datta S., Adjustments of multi-sample U-statistics to right censored data and confounding covariates, Computational Statistics and Data Analysis. 135 (2019) 1–14. doi:10.1016/j.csda.2019.01.012.
31. Yu C, Zelterman D., A parametric model to estimate the proportion from true null using a distribution for p-values, Computational Statistics and Data Analysis. 114 (2017) 105–118. doi:10.1016/j.csda.2017.04.008.
32. Von Borries G, Wang H., Partition clustering of high dimensional low sample size data based on p-values, Computational Statistics and Data Analysis. 53 (2009) 3987–3998. doi:10.1016/j.csda.2009.06.012.
33. Semwal VB, Singha J, Sharma PK, Chauhan A, Behera B., An optimized feature selection technique based on incremental feature analysis for bio-metric gait data classification, Multimedia Tools and Applications. 76 (2017) 24457–24475. doi:10.1007/s11042-016-4110-y.
34. Yang C, Ji J, Liu J, Liu J, Yin B., Structural learning of Bayesian networks by bacterial foraging optimization, International Journal of Approximate Reasoning. 69 (2016) 147–167. doi:10.1016/j.ijar.2015.11.003.
35. Rueda R, Ruiz LGB, Cuéllar MP, Pegalajar MC., An Ant Colony Optimization approach for symbolic regression using Straight Line Programs . Application to energy consumption modelling, International Journal of Approximate Reasoning. 121 (2020) 23–38. doi:10.1016/j.ijar.2020.03.005.
36. Alomari R, Thorpe J., On password behaviours and attitudes in different populations, Journal of Information Security and Applications. 45 (2019) 79–89. doi:10.1016/j.jisa.2018.12.008.
37. Zhang D, Li Q, Yang G, Li L, Sun X., Detection of image seam carving by using weber local descriptor and local binary patterns, Journal of Information Security and Applications. 36 (2017) 135–144. doi:10.1016/j.jisa.2017.09.003.
38. Asmitha KA, Vinod P., Linux Malware Detection using non-Parametric Statistical methods, in: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, New Delhi, 2014: pp. 319–332.
39. Zorn C., Shapiro-Wilk Test, Encyclopedia of Social Science Research Methods. (2004) 1305.
40. Royston JP., Algorithm AS 181: The W Test for Normality, Applied Statistics. 31 (1982) 176–180.
41. MathWorks, Multiple Comparison Test - MATLAB multcompare, (2020). http://www.mathworks.com/access/helpdesk/help/toolbox/stats/multcompare.html (accessed September 2, 2020).
42. Enck W, Ongtang M, McDaniel P., On lightweight mobile phone application certification, in: 16th Conference on Computer and Communications Security (CCS), ACM, New York, New York, USA, 2009: pp. 235–245. http://www.patrickmcdaniel.org/pubs/ccs09a.pdf.
43. Pearce P, Felt AP, Nunez G, Wagner D., AdDroid: Privilege Separation for Applications and Advertisers in Android, in: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security - ASIACCS '12, ACM Press, Seoul, Korea, 2012: p. 71. doi:10.1145/2414456.2414498.
44. Sanz B, Santos I, Laorden C, Ugarte-Pedrero X, Bringas PG, Alvarez G., PUMA: Permission usage to detect malware in Android, in: International Joint Conference CISIS-ICEUTE-SOCO Special Sessions, Springer Berlin Heidelberg, Ostrava, Czech Republic, 2013: pp. 289–298.
45. Canbek G., "Prominent Binary-Feature (Permissions) Frequencies for Android Mobile Benign Apps and Malware Datasets", Mendeley Data, V1, https://doi.org/10.17632/ptd9fnsrtr.1

APPENDIX A. SUPPLEMENTARY MATERIAL

A.1. DsFeatFreqComp – Dataset Feature-Frequency Comparison R Package

The developed open-source API provides two categories of important functionality for dataset manipulation and visualization conducted and recommended in this study.

Address: https://github.com/gurol/dsfeatfreqcomp

Visualization functions (as appeared in Fig.s 5 – 7):

- plotDsFreqDistributionViolin
- plotQQ
- plotPairwiseDsPValuesHeatMap

Dataset manipulation functions:

- loadDsFeatFreqsFromCsv2
- meltDataFrame

- loadPairwiseDsComparisonOfMeanRanks
- getPairwiseDsPValueMatrix

More information is provided in the developed package. The installation is also described in the GitHub address above.

A.2. Online Interactive Bump Chart for Permission Ranks for Intersection of Malign/ Benign Datasets

The online interactive dataset comparison chart appeared in Fig.s 2 and 3. The number of top binary features can be changed per class. Tooltips provide extra information.

Address: https://tabsoft.co/32CQGIP

A.3. Prominent Binary-Feature (Permissions) Frequencies for Android Mobile Benign Apps and Malware Datasets

The datasets compared in this study are provided online at Mendeley Data.

Address: http://dx.doi.org/10.17632/ptd9fnsrtr.1

APPENDIX B. RELATED EXAMPLE-DOMAIN WORKS

Since 2009 starting from the first version of Android, some studies have published frequent permission requests on benign/malign samples as a part of their static malware analysis. The following paragraphs outline the studies' review by only examining some of their highlights on permissions to explain the different aspects of permissions. As one of the earlier studies, Enck et al. [42] examined permission requests of 311 malicious applications and heuristically defined eight combinations of 13 permissions as the rules to signal malware. Table B.1 shows the rules decomposed in this study. Expressing their research solely based on a narrow set of permission combinations as a "certification" or "risk mitigation" process may cause misunderstanding. It is suggested that naming such an approach as 'suspiciousness indicator' for binary decisions or 'suspiciousness score' for rating the decision.

A composed rule is stated as "an application must not receive phone state, record audio, and access the Internet." In contrast, the actual threat is not requesting the permissions but allowing an application to record audio upon getting phone state (upon incoming or outgoing call), which is possible only by examining the code or catching the behavior

Table B.1. Decomposition of Rule-Based Classification in [42].

		RULES (Combination of permissions)							
PERMISSIONS	*Number of rules*	*R1*	*R6*	*R7*	*R8*	*R2*	*R3*	*R4*	*R5*
SEND_SMS	*1*			*X*					
RECEIVE_SMS	*1*		*X*						
READ_PHONE_STATE	*1*					*X*			
INSTALL_SHORTCUT	*1*				*X*				
UNINSTALL_SHORTCUT	*1*				*X*				
PROCESS_OUTGOING_CALLS	*1*						*X*		
ACCESS_FINE_LOCATION	*1*							*X*	
ACCESS_COARSE_LOCATION	*1*								*X*
SET_DEBUG_APP	*1*	*X*							
RECEIVE_BOOT_COMPLETED	*2*							*X*	*X*
WRITE_SMS	*2*		*X*	*X*					
RECORD_AUDIO	*2*					*X*	*X*		
INTERNET	*4*					*X*	*X*	*X*	*X*
Number of permissions involved in the combination		*1*	*2*	*2*	*2*	*3*	*3*	*3*	*3*

at run-time on dynamic analysis. One question that needs to be asked is how the permissions chosen in the rules are sufficient to indicate the suspiciousness, which is not elaborated in [42].

One of Android's permission mechanism's nontrivial aspects is the distinction between the request and the actual use of permissions. Android application developers can declare a permission request, but there is no related action in the existing code that needs the existence of that permission granted. In a related study, Felt et al. [23] examined Android applications' permission requests. They evaluated whether the applications need the requested permissions based on their generated API (Application Programming Interface) permission map. They implemented a tool to scan the API calls to determine the required permissions and compare them with those requested. The generated result shows that about one-third of the examined 940 sample applications are over-privileged, violating the least privilege principle in information security. The study is based on API-level 8 (2010) with 85% coverage and 134 permissions and finds that 6.5% of all API calls depend on permission checks. The authors address the following reasons for developers to request unnecessary permissions:

- Being misled by permission names (e.g., MOUNT_UNMOUNT_FILESYSTEMS, ACCESS_NETWORK_STATE, and ACCESS_WIFI_STATE)

- Making unnecessary permission requests for the intents of deputy applications even though the deputy application already requested them (e.g., asking INSTALL_PACKAGES for Google Play deputy application, CAMERA for default camera, INTERNET for opening a URL (Uniform Resource Locator) in a browser, and CALL_PHONE for default Phone Dialer)

- Requesting permissions for unprotected methods such as 'getters' (e.g., no need to ask WRITE_SETTINGS for only calling getters [not setters] for Settings Content Provider)

- Pasting code snippets found on the Internet having inaccurate permission requests

- Requesting deprecated permissions (e.g., ACCESS_GPS or ACCESS_LOCATION has been deprecated since 2008)

- Forgetting the permission requested for tests (e.g., ACCESS_MOCK_LOCATION) and trials

- Requesting invalid 'Signature' or 'SignatureOrSystem' permissions that are silently refused since they are valid for the applications signed by the device manufacturers

- Requesting permissions intentionally in advance for future versions.

These reasons do certainly cause discrepancies in permission request frequencies, which should be considered as a significant noise in mostly benign datasets. However, it could be hypothesized that malware authors tend to develop malware requesting the minimal set of necessary permissions to avoid falling under suspicion.

Another attribute is advertisement libraries that are immensely used in Android applications. However, they cause over privilege in applications and consecutively mislead the analysis of permission requests for malware classification. Pearce et al. [43] examined 964 sample applications and found that some of the permissions requested by applications do not need for their functionalities but requested on behalf of advertisement libraries. Fig. B.1 shows the prepared depiction of the top permissions causing 'over privilege by advertisement' as they called it. The application category is another attribute that characterizes the permission requirements of applications. For instance, an application in the 'Games' category tends to request certain permissions than those in other categories such as the 'Shopping' category. Sarma et al. [20] present an approach that evaluates an application's permissions with those requested from other applications in the same category. They proposed a warning mechanism as 'the first line of defense' to inform the given application's permissions frequency compared with its category's permissions frequencies. Permissions rarely used by the category trigger a warning. Peng et al. [18] suggested using probabilistic generative models instead of frequency analysis to formulate a suspiciousness score for applications. The preferred scoring approach is based on the application's permission requests besides its category.

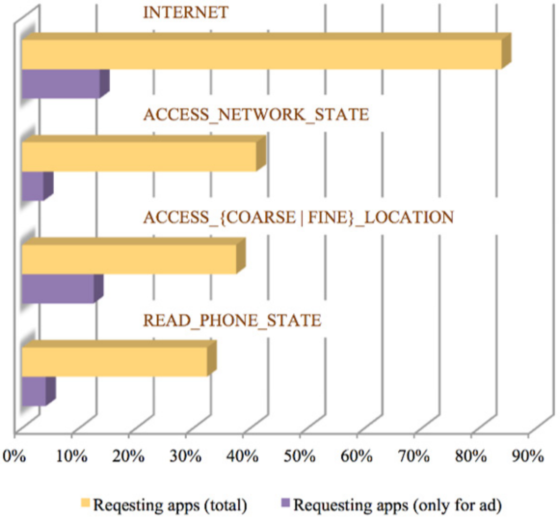


Figure B.1. The top permission requests cause over privilege due to the advertisement libraries, derived from [43].

Permissions in Android have other related attributes such as API-level, at which the permission is introduced, permission types (i.e. standard or custom), protection level, permission group, and used hardware or software features. The number of studies examining these additional attributes is very few. Sanz et al. [44] add informational used features as declared by 'uses-permissions' tags in an Android manifest file beside the permission requests. These features could be used for clustering permissions.

Besides the declarative static features extracted from an Android manifest file and Google Play data, as seen in the studies summarized above, some studies combine permissions with actual code structures, especially with Android API calls. Peiravian and Zhu [22] focus on the combination of permissions and API calls on potentially benign/malicious application classifications. Another observable attribute about permissions is comparing the number of requested permissions between benign and malicious applications. The executors of the "Android Malware Genome Project" Jiang and Zhou [17] conducted a very comprehensive analysis of Android malware, malware families' characteristics, propagation methods, triggering conditions, and payloads and permission usage. Analyzing 1,260 malware and 1,260 benign applications, they found that the malware usually requested more permissions than the benign applications, which are consistent with the other observations in the literature [13,15,18,19,22,44].

The study by Hoffman et al. [19] is noteworthy for expressing the possibility of data leakage threat by the existence of a specific pair of permissions. One permission is for accessing the critical or sensitive data (e.g., device information, contacts, location), and the other is for delivering them to the attacker (INTERNET permission with the overwhelming majority). Searching for critical permission combinations is not limited to permission pairs as in the indication of data leakage; more than two prerequisite permissions could also foresee other threats or abuse of privileges. Going beyond [21,42], Hoffman et al. [19] suggested, without giving sufficient explanation, a small number of suspicious permission patterns that are heuristically combined by logical connectives comprising not only ANDs but also ORs.

Yerima et al. [16] looked for the answers to 'which cardinality of the feature sets does yield a better result?' and 'which type or types of feature category generates more accurate classification result?' Comparing the top 30 permissions ranked by mutual information (MI), they concluded that a small number of features are sufficient, namely application permission requests and code attributes such as command calls, intent filters, embedded binaries, and API calls. The features could be used for statically classifying benign and malign applications at an underestimated performance. Wang et al. [15] examined the Android permission mechanism from different aspects and pointed to a different discriminative pattern in permission requests. Instead of reviewing a single or combination of a few permissions, the distribution of all permissions requested by applications could be used to classify applications as malign or benign. Aswini and Vinod [14] categorized the permissions according to their occurrence in two classes and assessed their contribution to classification. The common permissions occur at the intersection of two classes. Common and discriminant permissions are applied in different machine learning algorithms. They suggested that the common permissions have more influence on accurate classification. The ones having high inter-class variance are categorized as common prominent features. Another finding of the study in feature selection, contrary to assumptions, was the bottom BNS (Bi-Normal Separation) permissions exhibit better accuracy than the top BNS permissions because the distribution of top ones was nearly the same for both classes.

In summary, the review of related works highlights that several issues are related to permissions and the usage of permissions as a prominent feature for classifying malware such as

- effect of the combination of individual permissions, application category, and advertisement libraries,
- noise in permission request frequencies caused by over-privileged applications, and
- selection of the prominent permissions for achieving more successful classification.

However, as described in the examples of 'over privilege by advertisement' [43] and 'rule-based classification' [42] above, discriminative malign permissions' frequencies still provide a valuable indicator for practical malware classification.