Android Malware Classification

How to deal with extremely sparse features?

Luiza Sayfullina

Aalto University

Table of contents

1. Problem formulation
2. Android application as an object for classification
3. Feature selection
4. Dimensionality reduction
5. Proposed method and evaluation

Problem formulation

Android Malware

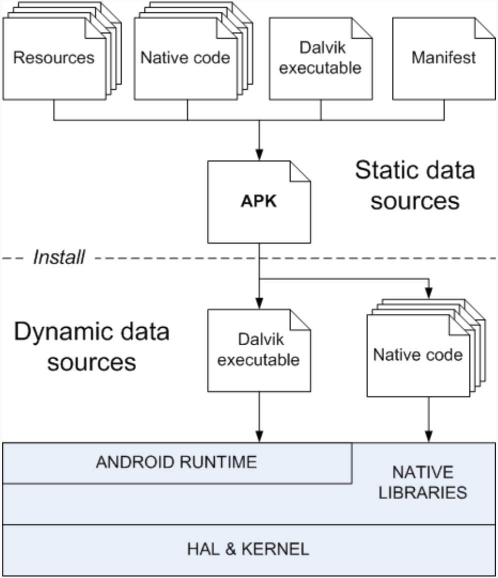
- More than 99% of all malware designed for mobile devices targets Android devices
- Extensive usage of mobile banking and internet browsing
- Harmful content via email attachments

Android application as an object for classification

Android app as an object for classification

- APK installation package distribution
- Application is described by a set of files, only 3 of which are fixed
- As a result files have only a small set of same kind of application descriptors

Dynamic and static approach in Android Malware Classification



Application JSON scheme

```
{
 "items": [{
  "name": "EICARAntiVirusTestFile.com",
  "hash": "3395856ce81f2b7382dee72602f798b642f14140",
  "path": "assets",
  "size": 68,
  "type": "file"
 },
 ...
 {
  "manifest": {
   "entries": [{
    "type": "activity",
    "name": ".EICARAntiVirusTestMainActivity",
    "filters": [{
     "actions": ["android.intent.action.MAIN"],
     "category": ["android.intent.category.LAUNCHER"]
    }]
   }],
   ...
  },
  "name": "AndroidManifest.xml",
  ...
  "type": "file"
 },
 ...
 {
  "name": "CERT.RSA",
  ...
  "type": "file"
 }],
 "name": "EICAR_Anti_virus_Test_1.0.apk",
 "hash": "3a5a91b58bb38454dedd90c2fc498a29a4def82c",
 "path": "",
 "size": 17922,
 "type": "file"
}
```

Extracted features

Feature type	Nmal=1K	Nclean=1K	Nmal=10K	Nclean=10K
Manifest strings (MSTR)	73,292	113,106	742,335	1,093,489
Permissions (PERM)	13,590	11,457	137,776	111,548
Dex strings (DEXS)	4,362,573	9,960,902	44,155,410	99,268,349
Hashes (HASH)	133,683	431,243	1,370,128	4,490,554

Table 1: The size of raw extracted features from different groups. With growth of dataset, the feature space reaches millions of features for DEXS and HASH features.

Feature selection

Log-odds feature selection

The chance of the feature i to contribute to a malicious file is defined as follows:

$$\theta_i = \frac{mal + k}{ben + k} \cdot \frac{B}{M}, \tag{1}$$

where B and M are the number of benign and malicious files respectively.

If we want to select the features being present only in the malicious class, we can achieve the following approximation, when k is small:

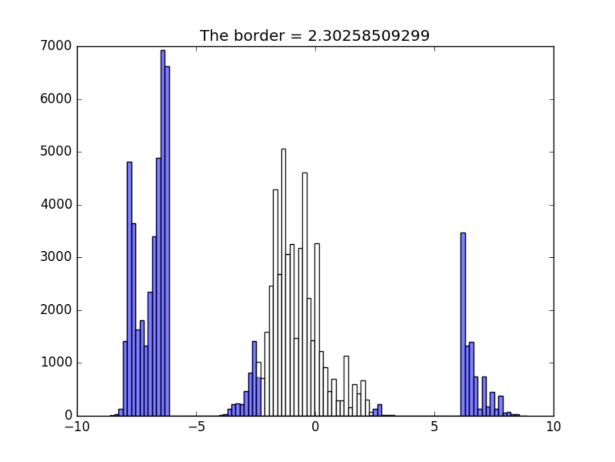
$$\log(\theta_i) = \log(mal + k) - \log(ben + k) \approx \log(mal) - \log(k) \tag{2}$$

Log-odds feature selection

- Hence the feature occurs one or more times in malicious file when $\log(\theta_i) \geq \log(1) - \log(k) \geq -\log(k)$.
- Accordingly to have the feature only being present in benign files the following inequality should hold: $\log(\theta_i) \leq \log(k)$.
- Both inequalities hold when feature occurs in either of the classes and $|\log(\theta_i)| \geq -\log(k)$.
- To allow the feature occur in both classes a constant c can be used: $|\log(\theta_i)| \geq -\log(ck)$.

Log-odds feature selection

- The distribution of $\log(\theta_i)$ values
- Only features with $|\log(\theta_i)| > -\log(0.1) \approx 2.30$ are selected



From *Efficient Detection of Zero-day Android Malware Using*

Choosing Laplace smoothing parameter

K	FPR	TPR
1	0.08	43.16
0.1	0.08	62.78
0.01	0.18	75.0
0.001	0.32	80.04
0.0001	0.42	83.14

Table 2: The effect of parameter k on the classification performance in the test set with Naive Bayes Classifier using BOW model.

Dimensionality reduction

Random Projections

For a given matrix $A \in \mathbb{R}^{N \times D}$ a random projection [1] defines a transformation to a lower dimensional space by multiplying by a randomly generated matrix $R \in \mathbb{R}^{D \times K}$, namely:

$$B = A \cdot R\,. \tag{3}$$

Initialization of random matrix can be expressed as follows, s « $\sqrt{D}$:

$$R = \sqrt{s} \begin{cases} 1 & \text{with probability } \frac{1}{2s} \\ 0 & \text{with probability } 1 - \frac{1}{s} \\ -1 & \text{with probability } \frac{1}{2s} \end{cases}. \tag{4}$$

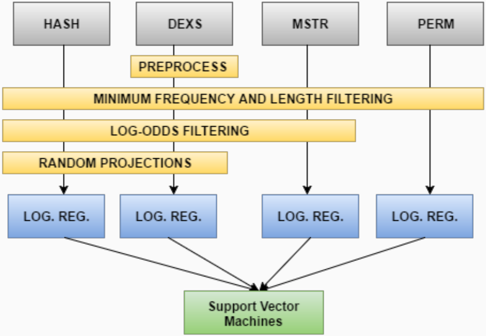
Validation of random projection matrix

K	ROC AUC
2,000	0.9913 ± 0.00052
4,000	0.9935 ± 0.00032
10,000	0.9944 ± 0.00027

Table 3: The effect of random projection initialization on DEX strings on the AUC. In other words, one does not need to try different RP initializations and find the best one; with sufficiently enough chosen features, the classification performance varies much less.

Proposed method and evaluation

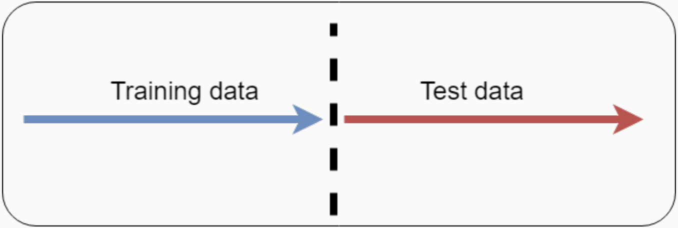
Feature ensemble



From *Android Malware Detection: Building Useful Representations by Sayfullina et al. [3]*

On evaluation

- Fixed datasets and unreliable estimates
- Importance of having chronologically separated training and test sets
- Aim at low false positive rate
- Continuous model retraining



From *Pragmatic Android Malware Classification" by Palumbo et al. [2]*

Experimental results

Method	Feat.	AUC	TP	FP
Proposed	DEXS, MSTR PERM, HASH	**0.9991**	**88.24**%	**0.04**%
NB	DEXS, MSTR PERM, HASH	0.9726	67.72%	0.04%
Proposed	DEXS, MSTR PERM	0.9986	86.92%	0.04%
NB	DEXS, MSTR PERM	0.9726	67.28%	0.04%

Table 4: The comparison between the performances of the proposed approach against the Naive Bayes implementation, split by different sets of features.

References i

D. Achlioptas.
Database-friendly random projections: Johnson-lindenstrauss with binary coins.
J. Comput. Syst. Sci., 66(4):671–687, June 2003.

P. Palumbo, L. Sayfullina, D. Komashinskiy, E. Eirola, and J. Karhunen.
A pragmatic android malware detection procedure.
Computers & Security, 70:689 – 701, 2017.

L. Sayfullina, E. Eirola, D. Komashinsky, P. Palumbo, and J. Karhunen.
Android malware detection: Building useful representations.
In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 201–206, 2016.

L. Sayfullina, E. Eirola, D. Komashinsky, P. Palumbo, Y. Miche, A. Lendasse, and J. Karhunen.
Efficient detection of zero-day android malware using normalized bernoulli naive bayes.
In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 198–205, 2015.