Identifying Android Malware Using Network-Based Approaches

Emily Alfs,[1] **Doina Caragea,**[1] **Nathan Albin,**[2] **Pietro Poggi-Corradini**[2]
[1]Department of Computer Science, Kansas State University
[2]Department of Mathematics, Kansas State University
{emilyalfs, dcaragea, albin, pietro}@ksu.edu

Abstract

The proliferation of Android apps has resulted in many malicious apps entering the market and causing significant damage. Robust techniques that determine if an app is malicious are greatly needed. We propose the use of a network-based approach to effectively separate malicious from benign apps, based on a small labeled dataset. The apps in our dataset come from the Google Play Store and have been scanned for malicious behavior using Virus Total to produce a ground truth dataset with labels *malicous* or *benign*. The apps in the resulting dataset have been represented using binary feature vectors (where the features represent permissions, intent actions, discriminative APIs, obfuscation signatures, and native code signatures). We have used the feature vectors corresponding to apps to build a weighted network that captures the "closeness" between apps. We propagate labels from the labeled apps to unlabeled apps, and evaluate the effectiveness of the proposed approach using the F1-measure. We have conducted experiments to compare three variants of the label propagation approaches on datasets that include increasingly larger amounts of labeled data. The results have shown that a variant proposed in this study gives the best results overall.

Introduction

Android apps are produced and published at a very high rate in the Google Play Store, and many malicious apps can potentially enter the market. Malicious apps can have many different purposes, including obtaining the user's personal information, such as passwords, credit card information, and much more. With the threat of such information getting into the wrong hands, it is necessary to find efficient and effective solutions to this problem. Thus, the purpose of this study is to identify malicious apps, or most specifically to classify apps from the Google Play Store as malicious or benign. Malicious apps are referred to as malware, a shorthand for malicious software. While many malicious apps exist, the ground truth labeled data is generally limited. To account for the lack of labeled data, we use transductive labeled propagation approaches, and study the variation of performance with the amount of labeled data, for both balanced datasets and more realistic imbalanced datasets.

Approach

Our approach to the malware detection problem is to use the notion of "closeness" of apps to create a network of labeled and unlabeled apps, and subsequently, propagate labels in the network. We determine closeness using the similarity metric defined by the Gaussian kernel. A binary representation of the apps, derived in previous research (Roy et al. 2015), was used to calculate app-to-app similarity.

Label propagation is a standard technique that can be applied in various ways. There are three main variants of label propagation that we compare: hard-clamping, soft-clamping, and our proposed variant. The hard-clamping variant aligns with the `scikit-learn`'s implementation of Label Propagation (Pedregosa et al. 2011), which we used in our hard-clamping experiments. The algorithm for hard-clamping (Zhu and Ghahramani 2002) is as follows: Let ℓ and u be the number of labeled and unlabeled data points, respectively, c be the number of classes (in our case $c = 2$, for benign and malware). Let $X = \{x_1, ..., x_l, x_{\ell+1}, ...x_{\ell+u}\}$ be the data matrix. Let W and T be $(\ell + u) \times (\ell + u)$ matrices, where W_{ij} represents the similarity between apps x_i and x_j, and $T_{ij} = \frac{w_{ij}}{\sum_{k=1}^{\ell+u} w_{kj}}$ represents the probability that we move from app x_j to x_i. The algorithm propagates the labels using $Y \leftarrow TY$, where Y is initialized with the original labels for labeled data and random values for unlabeled data. Then it row normalizes Y, and finally it clamps the labeled data, and repeats the above steps until convergence. By clamping the labeled data, the algorithm ensures that the original labels, which are believe to be true, are maintained.

The soft-clamping variant aligns with the `scikit-learn`'s implementation of Label Spreading (Pedregosa et al. 2011), which we used in the soft-clamping experiments. The algorithm for soft-clamping (Zhou et al. 2004) works as follows: assuming the variables defined in the hard-clamping algorithm, let F be a $(\ell + u) \times c$ matrix where each row, i, corresponds to the labeling of node x_i. Construct $S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where D is the diagonal degree matrix. Then, iterate $F \leftarrow \alpha SF + (1 - \alpha)Y$ until convergence, where $0 < \alpha < 1$. The variable α, acts as a weight that determines if we keep the original labels, when $\alpha = 0$, or disregard the original labels, when $\alpha = 1$.

Our proposed variant is a hybrid of hard-clamping and soft-clamping. Assuming the variables defined in hard/soft

Table 1: Comparison of label propagation approaches on balanced datasets (Top) and datasets with imbalance ratio 1:100 (Bottom). The following approaches are compared: hard-clamping, soft-clamping (with $\alpha = 0.1$), and our variant (with labeled apps having $\alpha_{lab} = 0$ and unlabeled apps having $\alpha_{unlab} = 0.1$). The performance is reported in terms of F1-measure.

Labeled Malware	Labeled Benign	Unlabeled Malware	Unlabeled Benign	Hard-clamping F1-measure	Soft-clamping F1-measure	Our Variant F1-measure
50	50	5000	5000	0.727	0.728	0.887
100	100	5000	5000	0.855	0.856	0.950
500	500	5000	5000	0.942	0.856	0.961
1000	1000	5000	5000	0.955	0.955	0.967
5000	5000	5000	5000	0.866	0.973	0.979
10	1000	50	5000	0.456	0.567	0.562
20	2000	50	5000	0.588	0.632	0.630
25	2500	50	5000	0.604	0.625	0.634
50	5000	50	5000	0.712	0.739	0.752

clamping, we iterate using $F \leftarrow \Lambda SF + (1 - \Lambda)Y$, where Λ is a $(\ell + u) \times (\ell + u)$ diagonal matrix with $\lambda_{ii} = \alpha_i$. This allows for each app x_i to have a specific $0 \leq \alpha_i \leq 1$ value. Thus, for the ground truth apps, we can adhere more strongly to the original labels as in the hard-clamping method (while potentially allowing the labels to change with a small probability to account for noisy ground truth). As opposed to that, the unlabeled apps can change their assigned labels freely.

Experiments and Results

Dataset Description. Our total repository consists of more than one million apps, which have been labeled using the VirusTotal software. However, a relatively small number of apps (specifically, 18,000) have been manually verified as malicous (Wei et al. 2017). In our study, we sampled 10,000 malicious apps from the manually-verified set, and 10,000 benign apps from the total set. The selected malware and benign apps were used to create labeled/unlabeled datasets of various sizes and various benign-to-malware ratios. Each app, benign or malware, was represented with a feature vector with 471 binary entries. The features were extracted in previous research (Roy et al. 2015), and fall into the following categories: permissions, intent actions, discriminative APIs, obfuscation signatures, and native code signatures.

Experimental Setting. To study the variation of performance with the number of labeled apps, we constructed labeled datasets of increasing size, by starting with a small labeled dataset and incrementally increasing its size in subsequent experiments. In all experiments, the evaluation was performed on the same test dataset. We performed experiments with both balanced datasets and imbalanced datasets. The performance was measured using the F1-measure.

Results. The results of the experiments with balanced datasets are shown in Table 1 (Top), while the results of the experiments with imbalanced datasets are shown in Table 1 (Bottom). The results shown for soft-clamping use $\alpha = 0.1$, while our variant uses $\alpha_{lab} = 0$ and $\alpha_{unlab} = 0.1$ for labeled and unlabeled apps, respectively. We compared the F1-measure values obtained with the tested approaches. As can be seen, our variant gives the best results overall, regardless of the size of the labeled data and the imbalance ratio. As expected, the performance generally improves with the size of the datasets for all variants.

Conclusion and Future Work

We have experimented with network-based label propagation on the problem of identifying Android malware. The results suggest that the proposed variant, which uses different clamping values for labeled versus unlabeled data, works better than the hard-clamping and soft-clamping baselines. While the proposed variant gives the best results for both balanced and imbalanced datasets, as part of future work, we plan to fine-tune the parameters of the model (e.g., the α values), and specifically incorporate mechanisms for controlling the effect of the data imbalance in the approach (Zhu and Ghahramani 2002). Furthermore, we plan to expand our research by studying how noisy data impacts the results.

Acknowledgments

This project is partially supported by the National Science Foundation under Grants No. 1717871 and No. 1515810.

References

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Roy, S.; DeLoach, J.; Li, Y.; Herndon, N.; Caragea, D.; Ou, X.; Ranganath, V. P.; Li, H.; and Guevara, N. 2015. Experimental study with real-world data for android app security analysis using machine learning. In *Proc. of the 31st Annual Computer Security Applications Conf.*, ACSAC 2015, 81–90. New York, NY: ACM.

Wei, F.; Li, Y.; Roy, S.; Ou, X.; and Zhou, W. 2017. Deep ground truth analysis of current android malware. In Polychronakis, M., and Meier, M., eds., *Detection of Intrusions and Malware, and Vulnerability Assessment*, 252–276. Springer International Publishing.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 321–328. MIT Press.

Zhu, X., and Ghahramani, Z. 2002. Learning from labeled and unlabeled data with label propagation. Technical report, School of Computer Science Carnegie Mellon University.