

## Research Article

# A Graph-Based Feature Generation Approach in Android Malware Detection with Machine Learning Techniques

Xiaojian Liu<sup>1</sup>, Qian Lei, and Kehong Liu

*School of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an, Shaanxi 710054, China*

Correspondence should be addressed to Xiaojian Liu; 780209965@qq.com

DWM[hW \$" A UfaTVd \$" #+-DVM[dW %/? SdLZ \$" \$" -3UWmfW) 3bd[^\$ " \$" -BgT[^ZW \$) ? Sk \$" \$"

Academic Editor: Eric Florentin

Copyright © 2020 Xiaojian Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An explosive spread of Android malware causes a serious concern for Android application security. One of the solutions to detecting malicious payloads sneaking in an application is to treat the detection as a *binary classification* problem, which can be effectively tackled with traditional machine learning techniques. The key factors in detecting Android malware with machine learning techniques are *feature selection* and *generation*. Most of the existing approaches select and generate features without fully examining the structures of programs, and thus the important semantic information associated with these features is lost, consequently resulting in a low accuracy rate in detection. To address this issue, we propose a new feature generation approach for Android applications, which takes components and program structures into consideration and extracts features in a graph-based and semantics-rich style. This approach highlights two major distinguishing aspects: the *context-based feature selection* and *graph-based feature generation*. We abstract an Android application as a collection of reduced iCFGs (interprocedural control flow graphs) and extract original features from these graphs. Combining the original features and their contexts together, we generate new features which hold richer semantic information than the original ones. By embedding the features into a feature vector space, we can use machine learning techniques to train a malware detector. The experiment results show that this approach achieves an accuracy rate of 95.4% and a recall rate of 96.5%, which prove the effectiveness and advantages of our approach.

## 1. Introduction

Android system, as one of the most popular mobile platforms, faces various serious security challenges due to its open-source characteristics, imperfect permission mechanisms, and the absence of full certification of applications at their publications. Malware or malicious payload exploits the vulnerabilities in Android system to implement a variety of attacks, such as privilege escalation, remote control, illegal financial charges, and personal information stealing, resulting in privacy leakage and even serious financial losses.

Therefore, there is an imperative need to detect malicious payload and analyze its potential impact when installing and using an Android application.

Machine learning techniques have been widely used in malware detection [1–8]. This kind of approach treats malware detection as a binary classification problem (i.e., differentiate an application as malicious or benign), which

can be tackled with the traditional techniques in pattern recognition or machine learning disciplines. Since this approach does not fully investigate the semantics and all details of programs, it achieves a better performance over traditional *dynamic approaches* [1, 2, 9] and *static approaches* [4–6, 10–15] in terms of scalability and time consumption.

The key factors affecting the performance of the machine learning-based approaches are feature selection and generation. APIs and permissions [1, 3] are commonly selected as the features in that they hold rich security-related information about what critical resources can be accessed by which operations. However, in most of the existing works, these features are extracted in a level of whole-application granularity, and their associated contexts are neglected, resulting in a high false-positive ratio in detection.

To tackle this problem, we propose a static detection method for Android malware, which improves the existing works by additionally considering the contexts of features.





























