基于卡方检验的 Android 恶意应用检测方法

刘亚姝[1,2]， 王志海[1]， 李经纬[3]， 赵烜[3]， 文伟平[3]
(1. 北京交通大学 计算机与信息技术学院，北京 100044；2. 北京建筑大学 电气与信息工程学院，北京 100044；3. 北京大学 软件与微电子学院，北京 102600)

摘 要：移动终端爆发式增长造成了恶意应用的大量出现，给用户的隐私安全和财产安全带来了巨大的危害. 为提高 Android 应用恶意性检测的准确性，本文将卡方检验与基尼不纯度增量相结合获取更有价值的特征属性；并改进朴素贝叶斯算法提高 Android 应用恶意性判断的准确性. 实验结果表明：新的特征处理方法能够有效提高检测性能；同时，改进后的朴素贝叶斯算法相比原始算法而言准确率有较大的提升.

关键词：恶意软件；安卓；卡方检验；朴素贝叶斯
中图分类号：TP309.5 **文献标志码**：A **文章编号**：1001-0645(2019)03-0290-05
DOI：10.15918/j.tbit1001-0645.2019.03.011

An Android Malware Detection Method Based on Chi-Squared Test

LIU Ya-shu[1,2], WANG Zhi-hai[1], LI Jing-wei[3], ZHAO Xuan[3], WEN Wei-ping[3]
(1. School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China; 2. School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China; 3. School of Electronics Engineering and Computer Science, Peking University, Beijing 102600, China)

Abstract: The explosive growth of mobile terminals has produced endless malicious applications, bring on great harm to the security of users' privacy and property. To solve this problem, a method based on chi-squared test and Gini impurity increment was proposed for more valuable features extraction and the Naïve Bayes algorithm improvement, so as to improve the estimation accuracy of Android malevolence applications. Test shows that the new features processing method can improve the classification performance of algorithms. At the same time, the improved Naïve Bayes algorithm can achieve higher accuracy than before.

Key words: malware; Android; Chi-squared test; Naïve Bayes

随着互联网行业的深入发展，移动互联网切实融入了日常生活工作中，但移动操作系统的开放生态体系，使得终端安全和应用安全正遭受严峻考验. 特别对于 Android 系统，开放性和数据的大规模流动带来了大量的数据信息隐私安全等问题[1].

据统计，Android 平台上恶意程序的数量总体一直呈上升趋势. 诺基亚《2017 年度威胁情报报告》[2]中显示 2017 年智能手机占所有移动网络感染的 72%，其中 Android 手机感染率高达到 68%. 2017 年由互联网应急中心捕获的移动系统平台上的恶意程序样本数量为 253 万余个，同比增长 23.4%，其中，面向 Android 平台的约达 96.55%[3]. 这些数据说明，目前 Android 平台已成为最主要的攻击对象，因此有效的恶意应用检测技

收稿日期：2018-10-18
基金项目：国家重点研发计划资助项目(2018YFB0803604)；国家自然科学基金重点资助项目(U1736218)；国家自然科学基金面上资助项目(61672086)
作者简介：刘亚姝(1977—)，女，博士生，E-mail：liuyashu@bucea.edu.cn.
通信作者：王志海(1963—)，男，教授，E-mail：zhhwang@bjtu.edu.cn.

术是非常必要的.

杨欢等[4]提出了基于 Android 组件特征、函数调用特征以及系统调用类特征的 3 层混合算法,以此建立最优的分类器,实现恶意行为的判定;曾立鹃等[5]通过汇编代码获得组件信息和敏感 API 调用图,并分析其安全性;秦中元等[6]利用 API 签名、Method 签名、Class 签名、APK 签名生成多级签名,并通过签名检测 Android 应用是否含有恶意行为;马锐等[7]提出基于粒子群优化算法的 Android 应用检测方法,能够提高测试用例生成效率和自动化率;Venugopal 等[8]应用 DJB 哈希函数算法提炼字符串、形成特征码,通过特征码匹配检测应用的恶意性;Zheng 等[9]通过权限信息、方法和类的信息构成综合特征码检测恶意应用.

目前研究人员也陆续开发出一些 Android 恶意应用检测工具. 例如,Comdroid[10]是一个用来发现 APP 漏洞的项目,但是它仅仅警告可能存在的漏洞,并不能够验证是否存在攻击;DroidChecker[11]特别之处在于可以发现 Adobe Photoshop 应用的漏洞;ProfileDroid[12]是个多层监测系统;RiskRanker[13]能在存在风险行为的应用还在应用商店时就可将其识别出来,从而避免恶意软件进入用户的手机;Taintdroid[14]是一款 Android 动态污点分析工具,但是目前仅能在 Dakvik 上使用.

面对日益严重的 Android 应用的安全问题,本文提出了一种基于卡方检验和基尼不纯度增量相结合的特征预处理方法,能够选择更有效的特征;提出了基于卡方值加权属性改进的朴素贝叶斯算法,设计并实现一种基于卡方检验的 Android 恶意应用检测工具,能够快速、有效地检测出恶意应用.

1　基本原理

1.1　特征筛选

卡方检验(Chi-squared test)是一种统计量的分布在零假设成立时近似服从卡方分布的假设检验. 其根本思想是比较两个样本率和两个分类变量的关联性. 因此,可以通过计算两个分类变量的卡方值判断分类变量之间的关联性. 卡方值越大,说明两个分类变量之间的联系越大,独立性越低;反之则说明两个分类变量间的联系越小,独立性越高.

样本的两个特征在不同类别下的统计次数构成卡方检验四格表,四格卡方公式如式(1)所示.

$$\chi^2 = \frac{(ad-bc)^2 N}{(a+b)(c+d)(a+c)(b+d)}. \quad (1)$$

式中:a、b、c、d 代表的是两个分类特征属性类别两两组合统计四格表中 4 个格子中样本的频数;N 为总次数,即 a、b、c、d 频数之和.

一个式子中独立变量的数目称作该式的"自由度",四格表的自由度为 1,利用式(1)计算类别权限的卡方值. 查找卡方临界值表,自由度为 1 时检验水准通常用 0.05 作为阈值,即两个特征属性之间相关的概率为 95%. 将卡方值与临界值进行比较,若卡方值大于临界值,则认为二者相关性大于 95%,说明两个特征之间的联系过高,可视为冗余特征. 为了去除关联性较高的冗余特征,本文采用基尼不纯度增量进行度量.

基尼指数是一种数据不纯度的度量方法,如式(2)所示.

$$\mathrm{Gini}(D) = 1 - \sum_{i=1}^{t} p_i^2. \quad (2)$$

式中:D 为数据集;t 为类别总数;p_i 表示类别为 i 的样本占总数的概率. $\mathrm{Gini}(D)$表示数据集 D 的基尼不纯度. 与信息增益类似,可以定义如式(3)所示的基尼不纯度的增量:

$$\Delta\mathrm{Gini}(A) = \mathrm{Gini}(D) - \mathrm{Gini}(D-A). \quad (3)$$

式中:$\mathrm{Gini}(D-A)$为数据集 D 确定 A 特征后的基尼不纯度;$\Delta\mathrm{Gini}(A)$为 A 特征对应的基尼不纯度的增量,该值越大表明 A 特征对结果影响越大.

因此,可以根据式(3)的结果筛选特征,去除基尼不纯度增量较小的特征,选择增量较大的特征,因为该特征对于结果的影响较大,将有助于分类检测.

1.2　改进的朴素贝叶斯算法

朴素贝叶斯模型是一个有监督学习的分类模型,其基本原理是根据贝叶斯公式计算样本类别的后验概率,选择后验概率大的类别作为该样本的类别. 贝叶斯分类算法非常适合用来过滤大量的样本,因为它一旦经过训练就可以相对较快地执行分类、计算开销较低.

假设有 m 个类别 $C(C_1, C_2, \cdots, C_m)$,n 个属性 $X(X_1, X_2, \cdots, X_n)$,计算训练样本集中每个属性在各个类别下的概率,即 $P(X_1|C_i)$,$P(X_2|C_i)$,$\cdots$,$P(X_n|C_i)$,简写为 $P(X|C_i)$.

则根据贝叶斯定理,属性属于每个样本的后验概率,如式(4)所示.

$$P(C_i \mid X) = \frac{P(C_i)P(X \mid C_i)}{P(X)}. \quad (4)$$

式中，$P(X)$对于所有类别都是常数，故只需要考虑 $P(C_i)P(X|C_i)$为最大值即可.

朴素贝叶斯是贝叶斯分类模型中常用的一种分类方法. 朴素贝叶斯分类算法中假定属性对于分类的影响相互独立，则 $X_1, X_2, \cdots, X_n$ 对于类别的概率乘积等于属性对于类别的概率，故有

$$P(X \mid C_i) = \prod_{k=1}^{n} P(X_k \mid C_i). \quad (5)$$

因此将式(5)代入式(4)中，有

$$P(C_i \mid X) = \frac{P(C_i)\prod_{k=1}^{n} P(X_k \mid C_i)}{P(X)}. \quad (6)$$

但是朴素贝叶斯算法要求事件之间相互独立，这一条件限制使得在实际应用中分类的准确度受到影响.

为此，本文对朴素贝叶斯算法进行了改进. 经过卡方检验和基尼不纯度增量的处理，样本属性间的相关性已经大大减弱，在此基础上，相比原始的朴素贝叶斯算法，本文增加了卡方值加权.

由于特征属性对于恶意应用的影响各不相同，将 Android 应用是否含有某一特征(例如某一权限)，与是否为恶意应用软件作为特征，两个特征构成卡方检验四格表计算卡方值 χ^2，进行特征属性的加权筛选，以此考虑特征对于分类的权重影响，因为卡方值越大则说明特征属性与分类间关系越密切，最终得到属性加权朴素贝叶斯算法公式，如式(7)所示，这为后续设计实现 Android 应用恶意性检测工具提供了理论基础.

$$P(C_i \mid X) = \frac{P(C_i)\prod_{k=1}^{n} P(X_k \mid C_i)\chi_k^2}{\sum_{i=1}^{m}\chi_i^2 \over P(X)}. \quad (7)$$

2 BayesDroid 检测工具的实现

基于第 1 节中提出的特征选择方法与改进的朴素贝叶斯算法，本文设计并实现了基于卡方检验的 Android 恶意应用检测工具(BayesDroid).

在 BayesDroid 中使用了“权限信息”和“API 调用”作为检测特征. 为了获得这些特征，首先使用 androidguard 工具对 APK 文件进行反编译以及静态分析，提取特征得到所需的权限特征集和敏感 API 调用特征集. 然后利用卡方检验算法结合基尼不纯度增量筛选特征；最后把选择出来的、有价值的特征结合改进的朴素贝叶斯算法训练模型，模型能够对 Android 恶意应用进行检测. BayesDroid 检测工具的工作流程如图 1 所示.

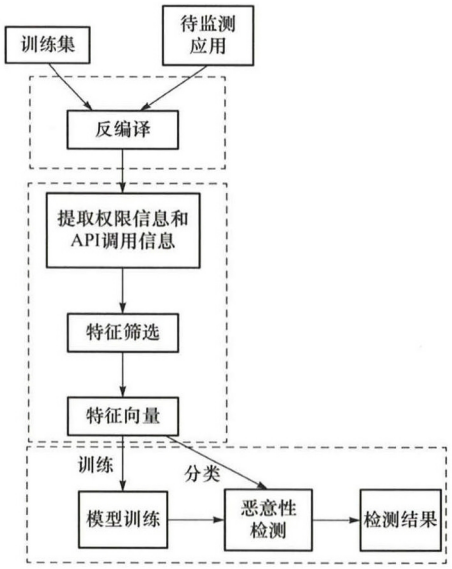

图 1 BayesDroid 检测流程图

Fig. 1 BayesDroid detection flow

BayesDroid 检测流程大体可以分为以下 3 个阶段.

① 反编译阶段：由于 Android 应用数据样本都为 APK 文件，需要借助 androidguard 静态分析工具逆向反编译，同时结合 python 脚本遍历应用数据样本所在的文件夹，最后得到所有需要的文件格式.

② 特征筛选阶段：本文主要选择应用的权限和敏感 API 调用作为分类特征. 首先通过 androidguard 中的 get_permissions 获得应用中的权限信息；get_apis 获得应用中的 API 调用信息，再利用字符串匹配算法形成权限特征向量和敏感 API 调用特征向量.

随后采用卡方检验结合基尼不纯度增量对提取的特征向量进行筛选以获取更有价值的特征.

③ Android 应用检测阶段：将特征向量中的每一个特征与类别(是否为恶意应用软件)构成卡方检验四格表计算卡方值 χ^2，进行特征属性的加权处理，得到特征加权朴素贝叶斯算法的权重，并训练模型，进而检测“待检测应用”输出检测结果.

3 实验结果分析

本文实验使用的数据由 Android 恶意应用的样

本和非恶意应用的样本组成. 其中恶意应用来自于著名的恶意应用样本网站 https://virusshare.com,按照类别随机选取了 5 000 个恶意应用样本. 同时在国内的 Android 应用市场收集了 4 500 个非恶意应用,包含多种类型应用,例如工具、游戏、通信等应用程序,并且通过 Android 应用杀毒软件的过滤以保证它们的非恶意性.

本文通过正确率(accuracy, ACC)、误报率(false positive rate, FPR)以及运行时间(runtime)来衡量检测方法的性能.

3.1　特征筛选的有效性实验

实验 1 检测特征筛选方法的性能. 比较未进行特征筛选和已完成特征筛选之后的原始朴素贝叶斯算法性能,结果如表 1 所示.

表 1　特征筛选性能对比

Tab. 1　Performance comparison of feature selection

特征	ACC/%	FP/%	运行时间/s
原始	69.66	23.89	73.18
已筛选	75.95	18.63	72.83

实验 1 中,特征筛选采用了卡方检验和基尼不纯度增量. 从表 1 可以看出,"原始"列朴素贝叶斯算法检测准确率只有 69.66%,低于经过了卡方检验筛选的特征的准确率(75.95%). 该结果表明本文的特征筛选方法能够去除冗余特征,提高朴素贝叶斯算法的检测性能.

为了进一步验证特征筛选的有效性,本文采用随机森林(random forest, RF)和支持向量机(support vector machine, SVM)算法完成上述实验,结果如表 2 所示.

表 2　特征筛选的有效性对比

Tab. 2　Effectiveness comparison of feature selection

特征	RF			SVM		
	ACC/%	FP/%	运行时间/s	ACC/%	FP/%	运行时间/s
原始	72.96	18.23	84.69	71.66	18.09	76.18
已筛选	80.59	17.63	80.38	79.02	17.63	76.06

从表 2 中可以看到,采用"已筛选"的 RF 和 SVM 的分类结果高于使用"原始特征"的结果.

从表 1 和表 2 也可以看到,不论是朴素贝叶斯、RF 还是 SVM,使用未经过筛选的特征的检测准确率显著低于采用卡方检验筛选的特征的准确率. 这说明本文的特征处理方法与分类算法无关,可以普遍地使分类算法在检测性能上得到提高.

3.2　改进的朴素贝叶斯算法实验

实验 2 检测改进的朴素贝叶斯算法的性能. 如 1.2 节所述,本文采用卡方值对特征属性加权从而改进朴素贝叶斯算法,为了检测改进后的朴素贝叶斯算法的性能,针对原始特征和筛选后的特征分别进行实验,结果如表 3 所示.

表 3　改进的朴素贝叶斯算法评价表

Tab. 3　The evaluation table of improved Naïve Bayes

特征	ACC/%	FP/%	运行时间/s
原始	76.65	17.81	73.36
已筛选	82.06	18.06	73.65

对比表 1 和表 3 中"原始"列和"已筛选"列的结果可以看到,改进后的朴素贝叶斯算法相比原始朴素贝叶斯算法而言检测准确率有了明显的提高. 将经过筛选的特征用于改进的朴素贝叶斯算法分类,可以得到 82.06%的准确率,时间消耗也没有显著增加. 说明 BayesDroid 检测系统相比传统方法确实具有较好的性能.

4　结　论

Android 系统在智能终端系统中市场份额越来越大,利用恶意应用非法获取用户的信息和利益的恶意行为也越来越多. 本文采用了静态分析方法,将卡方检验结合基尼不纯度增量对特征进行预处理,并对朴素贝叶斯算法进行了基于卡方检验的加权改进,在此基础上实现了 BayesDroid 恶意应用检测工具,为 Android 平台下海量恶意应用的快速检测提供了可行的方法和工具. 本文方法虽然相比原始方法在有效性和准确性上都有了很大的提高,但是还有很大的提高空间,需要进一步研究以提升检测的准确率.

参考文献:

[1] 付玉辉. 2016 年中国信息传播产业发展概述[J]. 移动通信,2017,41(1):7-12.
Fu Yuhui. Review of China's information and communication industries in 2016[J]. Mobile Communications, 2017,41(1):7-12. (in Chinese)

[2] Daniel G. Nokia releases annual threat intelligence report for 2017[R/OL]. [2017-11-02]. https://www.android-headlines.com/2017/11/nokia-releases-annual-threat-intelli-

gence-report-2017.html.

[3] 国家计算机网络应急中心.中国移动互联网发展状况及其安全报告(2017)[R/OL].[2017-11-02]. http://www.cert.org.cn/publish/main/46/index.html.
National Computer Network Emergency Center. China mobile internet development and securityreport (2017)[R/OL]. [2017-11-02]. http://www.cert.org.cn/publish/main/46/index.html. (in Chinese)

[4] 杨欢,张玉清,胡予濮,等.基于多类特征的Android应用恶意行为检测系统[J].计算机学报,2014,37(1):15-27.
Yang Huan, Zhang Yuqin, Hu Yupu, et al. A malware behavior detection system of Android applications based on multi-class features[J]. Chinese Journal of Computers,2014,37(1):15-27. (in Chinese)

[5] 曾立鹃,唐泉彬,牛斗. Android系统应用程序组件安全性分析[J].软件,2014,35(3):147-151.
Zeng Likun, Tang Quanbin, Niu Dou. Analysis the security of components in android application[J]. Computer Engineering & Software,2014,35(3):147-151. (in Chinese)

[6] 秦中元,王志远,吴伏宝,等.基于多级签名匹配算法的Android恶意应用检测[J].计算机应用研究,2016,33(3):891-894.
Qin Zhongyuan, Wang Zhiyuan, Wu Fubao, et al. Android malware detection based on multi-level signature matching[J]. Application Research of Computers,2016,33(3):891-894. (in Chinese)

[7] 马锐,任帅敏,马科,等.基于粒子群优化算法的Android应用自动化测试方法[J].北京理工大学学报:自然科学版,2017,37(12):1265-1270.
Ma rui, Ren Shuaimin, Ma Ke, et al. Test automation for Android applications based on particle swarm optimization algorithm[J]. Transactions of Beijing Institute of Technology,2017,37(12):1265-1270. (in Chinese)

[8] Venugopal D. An efficient signature representation and matching method for mobile devices[C]// Proceedings of the 2nd Annual International Workshop on Wireless Internet Article. Boston, USA: ACM Press, 2006: 1-9.

[9] Zheng M, Sun M, Liu J C S. Droid analytics: a signature based analytic system to collect, extract, analyze and associate android malware[C]// Proceedings of IEEE International Conference on Trust. Washington, USA: IEEE Press, 2013: 163-171.

[10] Chin E, Felta P, Greenwood K, et al. Analyzing inter-application communication in Android[C]// Proceedings of the 9th ACM International Conference on Mobile Systems, Applications and Services. New York: ACM Press, 2011: 239-252.

[11] Chan P P F, Hui L C K, Yiu S M. DroidChecker: analyzing android applications for capability leak[C]// Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks. New York: ACM Press, 2012: 125-136.

[12] Wei X T, Gomfz L, Neamtiu I, et al. ProfileDroid: multi-layer profiling of android applications[C]// Proceedings of the 18th Annual International Conference on Mobile Computing and Networking. New York: IEEE Press, 2012: 137-148.

[13] Grace M, Zhou Y, Zhang Q, et al. Risk ranker: scalable and accurate zero-day android malware detection[C]// Proceedings of the 10th ACM International Conference on Mobile Systems, Applications and Service. New York: ACM Press, 2012: 281-294.

[14] Enck W, Gilbert P, Chun B G, et al. TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones[J]. ACM Transactions on Computer Systems, 2012, 32 (2): 1-29.

(责任编辑:刘芳)