一种基于协同训练的 Android 恶意代码检测方法

王全民,张帅帅,杨 晶
(北京工业大学 信息学部,北京 100124)

摘 要:对于传统的恶意程序检测方法,将机器学习算法应用在未知恶意程序的检测方法进行研究。使用单一特征的机器学习算法无法充分发挥其数据处理能力,检测效果一般。使用两视图协同训练,对于一个未知样本两个分类器预测结果相反时处理不佳。因此,在机器学习的基础上,采用一种三视图协同训练算法,三个分类器对未知样本预测有分歧时,基于“少数服从多数”的思想进行“投票”决定,具有比较理想的效果。该方法对 APK 软件进行逆向分析和特征提取,选取权限申请特征、API 调用序列特征和 OpCode 特征三个非重叠子视图,针对每个子视图甄选最优算法分别生成分类器。在此基础上,采用 Co-training 算法思想,对三个分类器协同训练,实现了在已知样本较少的情况下,三个单独分类器检测性能的同步提升。从安卓市场下载各类良性样本 4 600 个,从恶意软件样本分享网站 VirusShare 下载最新恶意样本 4 360 个,按照已标记样本数量从 30 到 120 个分为 10 组实验,对约 1 800 个样本进行分类测试,实验结果表明该检测方法具有更优的效果。

关键词:机器学习;Co-training;三视图;投票;分类器
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2019)01-0135-05
doi:10.3969/j.issn.1673-629X.2019.01.028

An Android Malicious Code Detection Method Based on Cooperative Training

WANG Quan-min, ZHANG Shuai-shuai, YANG Jing
(Department of Informatics, Beijing University of Technology, Beijing 100124, China)

Abstract: For the traditional detection method of malicious program, the machine learning algorithm is applied to the detection method of unknown malware. The machine learning algorithm with a single feature cannot give full play to its data processing ability, and the detection effect is general. The two view collaborative training is not well for two classifiers with unknown samples when the prediction results are opposite. Therefore, based on machine learning, we adopt a collaborative training algorithm based on three views. When three classifiers are divided into unknown samples, voting is decided based on the idea of “majority obeys the majority”. This method carries out reverse analysis and feature extraction for APK software. It selects three non-overlapping sub-views of permission application features, API calling sequence feature and OpCode feature, and generates classifiers for each sub view to select the best algorithm. Based on that, the Co-training algorithm is used to train three classifiers and achieve synchronous performance improvement of three individual classifiers under less known samples. We download more than 4 600 benign samples from the Android Market, and more than 4 360 latest malware samples from VirusShare, a malware samples sharing site. According to the number of labeled samples from 30 to 120, 10 groups of experiments are conducted and about 1 800 samples are classified. The experiment shows that the detection method has a better effect.
Key words: machine learning; Co-training; three-view; voting; classifier

0 引 言

随着科技的高速发展,智能手机已经基本普及,其中手机应用软件已渗透到人们生活的方方面面,为人们的生活提供了极大的便利。智能手机在方便人们生活的同时,也存在着诸多安全隐患。其中 Android 系统尤为突出。据相关媒体报道,2012 年,恶意软件在 Android 平台从几十万发展到几千万级别,并且种类多样,显示出移动恶意软件已进入高速稳定增长期。尽

收稿日期:2018-02-08 **修回日期**:2018-06-14 **网络出版时间**:2018-11-15
基金项目:国家自然科学基金(61272500)
作者简介:王全民(1963-),男,副教授,博士,硕导,CCF 高级会员(E200005398S),研究方向为网络与信息安全;张帅帅(1993-),男,硕士研究生,研究方向为分布式系统和网络与信息安全。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20181114.1556.028.html

管在 2015 年移动端恶意代码在数量上和感染用户量上呈现下降趋势,但是全年内,仍有 18% 的设备被病毒感染,而且病毒攻击方式的专业性显著提高。

当前,检测恶意软件的主要方法是从签名和行为两个角度来进行[1-3]。由于不论是恶意软件的数量还是种类都在不断增加,而且恶意软件的技术又不断提升,所以传统的基于签名和行为分析的方法已经逐渐失去了一定的适用性。签名方法由于需要提前获取恶意软件的样本,然后将软件的签名发送给额外的程序去检查,所以该方法不能检测未知的恶意软件。基于行为的方法又可以分为动态检测方法和静态检测方法。动态检测是基于虚拟机上运行的程序,观察软件的行为来判断该软件是否为恶意软件。其优点在于它避开了静态分析中加密算法和代码混淆的问题,缺点是不能覆盖程序的所有有效路径,导致功能分析不够彻底。此外,动态方法需要在模拟器中运行未知应用,需要人工操作,效率低下。所以,众多研究人员将静态检测方法作为研究重点。静态检测方法主要使用反编译或者反汇编技术对 Android 应用软件进行解压,然后提取文件的静态特征并使用机器学习算法建立特征库,通过比较未知应用程序的特征来判断该软件是否为恶意软件。单纯使用权限或者函数调用作为特征会有较高的误报率。

结合传统恶意软件检测方法的缺点,在以权限和 API 函数调用两视图协同训练的做法上再加一个 native代码 OpCode 视图,构成三视图协同训练,对不同视图甄选出最合适的机器学习算法,对于未知样本预测,通过三个分类器投票决定。

1 研究现状

随着 Android 平台恶意软件持续不断的增长,且恶意软件不断使用新技术,使得越来越多的研究者投入研究。当前,国内和国外的研究方法中,主要是基于行为的检测方法和基于签名的检测方法。基于签名和特征码进行检测的方法是最早出现的检测方法[4],但是该方法的缺点也很明显,即无法检测未知软件。因此,众多研究者开始对基于动态行为特征的检测方法进行研究[5]。

基于恶意应用行为的检测方法是在软件运行过程中,提取软件的行为特征作为判断鉴别其是否为恶意软件的参考标准。Kim 等[6]基于电量消耗特征,提出了一种功率感知恶意软件检测方法,检测并分析应用软件电池电量消耗特征,由此判断该软件是否为恶意软件。路程等[3]开发了一种可以动态检测手机软件行为的检测工具,如通讯录、即时消息等一些隐私信息。Bläsing 等[7]提出了一种能够动态分析 Android 应用程序的系统调用方法,该方法可以检测已知恶意软件,也可以检测未知恶意软件。但是,动态检测方法需要人工的干预。

由于人工智能技术在近年来的高速发展,研究者们开始将机器学习和数据挖掘技术应用到检测恶意软件的方法中。早在 2008 年,Bose 等[8]就提出了应用机器学习算法来检测移动恶意软件的思想。他们指出,一个程序的真正意图常常可以由一系列行为的逻辑顺序随着时间的推移显示出来。他们研究了大量不同种类的病毒和木马样本,观察这些病毒样本运行时表现出的一系列行为,建立了行为特征数据库,然后使用支持向量机分类器对恶意程序和良性程序进行分类。Shabtai 等开发的 Andromaly[9-10]是一个基于主机的入侵检测系统(HIDS)。该系统可以持续监测到手机状态事件和手机日志信息,再通过机器学习算法对收集来的数据进行分类,然后判断该数据是良性还是恶意的。杨欢等[11]设计了基于多个特征的检测系统,并使用三层混合集成算法。关于对协同训练算法的研究,最早可追溯到 Blum and Mitchell[12]提出的 Co-Training 算法,该算法假设数据属性拥有两个充分的冗余视图[13],记为 $view_1$ 和 $view_2$。首先分别在已标记数据集 L 的 $view_1$ 视图上训练分类器 C_1,在 $view_2$ 视图上训练出分类器 C_2,然后从未标记数据集 U 上随机选取若干个(记为 u 个)示例放入到集合 U' 中;用 C_1 和 C_2 分别对 U' 中的所有元素进行标记,记为 E_1 和 E_2,再从 E_1 和 E_2 中分别选取置信度比较高的 p 个正标记和 n 个标记放入 L 中,最后从 U 中选取 $2p+2n$ 个数据加入 U' 中,重复上述过程直到满足截止条件。

2 解决方案

廖彦文[14]利用多视图协同训练算法实现了基于权限视图和 API 调用序列视图的检测系统,但是该系统在预测未知样本时,如果两个分类器对该样本预测的结果相反时,作者以置信度较高的结果为准,但如果两个分类器对未知样本预测结果相反且置信度相等时,该系统暂未处理。针对此种情况,文中在保留权限特征和 API 特征之外,再增加一种 OpCode 特征,首先在少量已标记数据集的三个视图上分别训练出三个分类器,然后从未标记数据集 U 中随机选出部分未标记数据给三个分类器进行标记,从单个分类器标记结果中选取置信度比较高的前 n 个样本加入另外两个训练集中,重复上述过程直至达到结束条件。在对未知样本进行预测时,三个分类器进行投票决定,以票数多的结果为准。协同训练流程如图 1 所示。

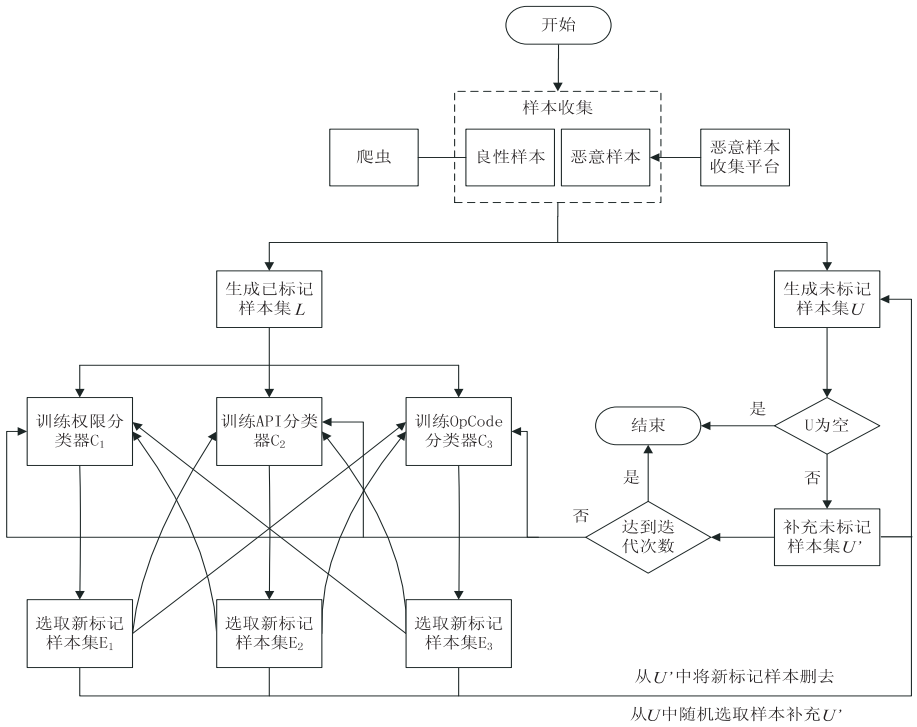

图 1　协同训练流程

2.1　样本收集

利用 Python 语言编写爬虫脚本,通过 re 模块的正则表达式来匹配应用下载链接。为了满足实验检测需求,从第三方应用市场安卓市场爬取良性样本 4 600 个,包含社交、教育、金融、健康、影音、拍照、购物、系统工具、电子阅读、日常应用,一共 10 类。从恶意样本分享网站 VirusShare 下载最新恶意样本 4 360 个。

2.2　特征提取

提取一个 APK 文件的权限特征、API 调用特征和 OpCode 特征时,因 APK 文件是一个压缩包,其中的 AndroidManifest. xml 文件和 classes. dex 文件是二进制文件,需要先使用反编译工具 apktool 对 APK 文件进行反编译处理,得到相应的可读文件 AndroidManifest. xml 和大量 smali 文件,将大量 smali 文件汇总到一个文件中后进行特征处理和提取。特征提取过程如图 2 所示。

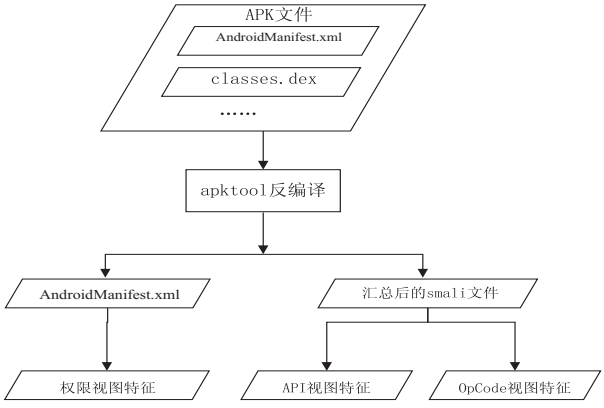

图 2　特征提取过程

2.2.1　权限特征提取

Android 系统中设置了一系列权限来控制应用程序访问敏感资源。应用程序必须在 AndroidManifest. xml 文件中用标签<uses-permission>进行声明定义后才能获取相应权限,应用程序申请的权限将在用户安装应用时提示用户。一个程序申请的权限表示它可以执行的操作,也就能间接表示一个安卓应用的行为[15]。因此,可以通过编写脚本使用正则表达式对反编译后的 AndroidManifest. xml 文件中的<uses-permission>标签进行匹配,提取出应用程序所申请的所有权限,再和标准系统权限集合作对比,选出实验所需权限。

2.2.2　API 调用特征提取

API 代表着程序实现的功能,API 和该程序的行为有着直接的联系。所以统计出程序源码中的 API 调用情况能够很好地反映出该应用程序是否有一些恶意行为。smali 代码是 dex 文件反编译后的代码,也就是说,smali 语言是 Dalvik 虚拟机的反汇编语言。对汇总后的 smali 文件进行匹配敏感 API,得到所有敏感 API 在源码中出现的次数。

2.2.3　OpCode 特征提取

Dalvik 虚拟机拥有专属的 dex 可执行文件格式和指令集代码。由图 2 可知,对 dex 文件反汇编后得到 smali 文件,通过 Python 编写脚本进行正则匹配来提取 smali 文件里的 OpCode 指令,对提取到的操作符进行 N-gram 建模,经过实验对比,对于 OpCode 特征选取 $n=3$,即建立 3-gram 模型,实验效果最佳。如提取

的 OpCode 特征向量为{input-object, invoke-direct, return-void, iget-object, invoke-static, return-void}，则 3-gram 模型为{input-object, invoke-direct, return-void}，{invoke-direct, return-void, iget-object}，{return-void, iget-object, invoke-static}，{iget-object, invoke-static, return-void}，同时统计每个子集出现的次数。

2.3 算法甄选

使用不同的分类算法实现的分类器针对不同视图特征的特性，分类效果是不同的。为了得到最佳的实验效果，为单一视图筛选出最合适的分类器，具体甄选过程如图 3 所示。

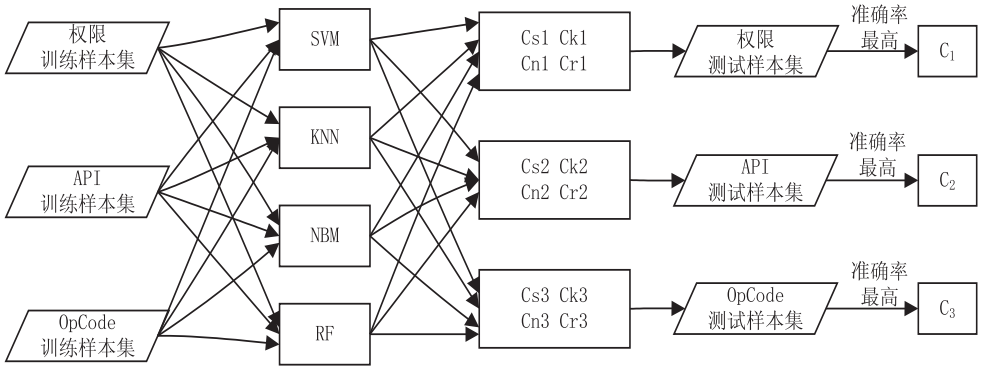

图 3 单一分类器算法甄选模块

从单一视图实验，在相同的训练样本集和测试集的情况下，比较 SVM、KNN、朴素贝叶斯、随机森林 4 种算法实现的分类器，选出分类器准确率最高的一个作为单一视图上的分类器。经单一视图实验结果验证，在 API 调用选择 KNN 模型，权限视图上选择 SVM 模型，在 OpCode 视图上选择随机森林模型。

3 实 验

3.1 实验环境

实验的代码编写采用 Python 语言实现。物理主机为 4 GB 内存，处理器为 Intel(R) Core(TM) Quad Q8200 2.33 GHz，硬盘大小为 500 G，系统开发环境为 64 位 Windows 7 旗舰版。

3.2 实验结果与分析

为了保证实验的可靠性，从安卓市场下载各类良性样本 4 600 个，从恶意软件样本分享网站 VirusShare 下载最新恶意样本 4 360 个。

在协同训练实验中，将已标记样本集最大设为 100 个，按数量规模从 30 到 120 分为 10 组进行实验。在未标记样本中随机选取 150 个样本作为辅助协同训练的未知样本集。每组实验的协同训练进行了 20 轮，每轮迭代在各视图上选取的最置信样本数量为本轮训练已标记样本总数的 15%。实验对总样本数的 20%(1 800 个)测试样本进行分类性能测试。

下面使用 ACCR(总检测准确率)、LDR(异常样本漏检率)、FPR(正常样本误判率)三个指标来评价实验结果。其中，ACCR 为主要评价指标，其值越高越好，LDR 和 FPR 为辅助评价指标，其值越低越好。假设 N_1 为测试样本集恶意样本总数，N_2 为测试样本集良性样本总数，TP 为被正确检测到异常样本的总数，TN 为被正确检测到良性样本的总数，ACCR = (TP+TN)/(N_1 + N_2)，LDR = (N_1 - TP)/N_1，FPR = (N_2 - TN)/N_2。

表 1 展示了在不同规模的训练样本基础上，SVM 分类器、KNN 分类器和随机森林分类器协同训练前后，总检测率 ACCR 的结果数据。

表 1 协同训练前后各视图分类准确率对比

已标记样本数量	权限(SVM)		API(KNN)		OpCode(RF)	
	协同训练前	协同训练后	协同训练前	协同训练后	协同训练前	协同训练后
30	75.565	85.195	72.695	87.235	70.56	79.47
40	73.17	84.395	73.316	87.029	71.749	81.095
50	76.92	86.171	77.13	88.643	74.27	82.586
60	78.153	85.723	79.64	89.18	75.891	82.651
70	78.23	85.185	80.67	88.48	76.03	81.35
80	81.06	86.717	80.19	89.148	76.892	83.008
90	79.26	85.097	79.36	87.517	78.613	83.65
100	82.6	88.056	81.39	89.863	79.574	85.33
110	82.96	87.939	81.82	89.131	80.39	85.662
120	83.43	87.8	82.64	90.21	81.561	87.181

由表 1 数据可得在不同已知样本基础上，三个视图在协同训练前后总检测准确率的提高幅度变化曲线，如图 4 所示。

由图 4 可知，协同训练前后，利用文中方法，在权限视图上的分析准确率有 4.37% ~ 11.23% 的提升，在 API 视图上有 7.3% ~ 14.54% 的提升，在 OpCode

视图上有 5.04% ~9.37% 的提升。权限、API 和 OpCode 三个视图上的分类准确率的提升幅度总体上随着已标记样本的增多而降低,该情形说明,文中提出的方案比较适用于已标记样本较少的情形,即在已知样本较少的情况下,检测效果作用更佳。

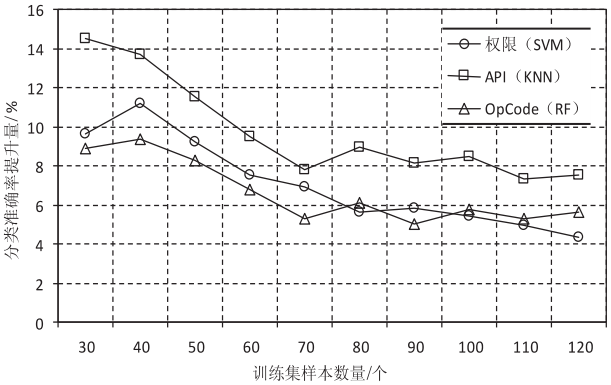

图 4　协同训练前后三个视图 ACCR 提升幅度对比曲线

4　结束语

在提出的 Android 恶意代码检测的方案中,在 APK 软件中,提取了其权限特征、API 调用序列特征和 OpCode 特征,建立三个视图,从不同角度来描述一个 Android 应用软件。在单一视图上甄选出能够使该模型分类准确率最高的算法。在单视图分类最优的前提下,利用协同训练思想,使得三个分类器检测准确率都进一步提升。在最后的实验里,在权限视图、API 视图和 OpCode 视图方面均有不同程度的提升。此外,利用三个视图通过对样本进行“投票”能很好地解决在两视图下由于两个分类器对一个未知样本预测不一致且置信度相同的情形,再一次验证了三个视图协同检测比单一视图和两视图分类更加有效。

参考文献:

[1] 张玉清,王　凯,杨　欢,等. Android 安全综述[J]. 计算机研究与发展,2014,51(7):1385-1396.

[2] 王菲飞. 基于 Android 平台的手机恶意代码检测与防护技术研究[D]. 北京:北京交通大学,2012.

[3] 路　程. Android 平台恶意软件检测系统的设计与实现[D]. 北京:北京邮电大学,2012.

[4] CHRISTODORESCU M,JHA S. Static analysis of executables to detect malicious patterns[R]. Wisconsin:University of Madison,2006.

[5] 徐曾春,卢　洲,叶　坤. 一种检测可疑软件的 Android 沙箱系统的研究与设计[J]. 南京邮电大学学报:自然科学版,2015,35(4):104-109.

[6] KIM H,SMITH J,SHIN K G. Detecting energy-greedy anomalies and mobile malware variants[C]//Proceedings of the 6th international conference on mobile systems,applications,and services. Breckenridge,CO,USA:ACM,2008:239-252.

[7] BLÄSING T,BATYUK L,SCHMIDT A D,et al. An Android application sandbox system for suspicious software detection[C]//5th international conference on malicious and unwanted software. Nancy,Lorraine,France:IEEE,2010:55-62.

[8] BOSE A,HU Xin,SHIN K G,et al. Behavioral detection of malware on mobile handsets[C]//Proceedings of the 6th international conference on Mobile systems,applications,and services. Breckenridge,CO,USA:ACM,2008:225-238.

[9] SHABTAI A,ELOVICI Y. Applying behavioral detection on android-based devices[M]//Mobile wireless middleware,operating systems,and applicaitons. Berlin:Springer,2010:235-249.

[10] SHABTAI A,KANONOV U,ELOVICI Y,et al. “Andromaly”:a behavioral malware detection framework for android devices[J]. Journal of Intelligent Information Systems,2012,38(1):161-190.

[11] 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,2015,37(1):15-27.

[12] BLUM A,MITCHELL T. Combining labeled and unlabeled data with co-training[C]//Proceedings of the 11th annual conference on computational learning theory. Berlin:Springer,1998:92-100.

[13] 周志华. 半监督学习中的协同训练风范[M]//机器学习及其应用. 北京:清华大学出版社,2007:259-275.

[14] 廖彦文. 基于多视图协同分类的安卓恶意软件检测方法研究[D]. 北京:北京工业大学,2016.

[15] HUANG C Y,TSAI Y T,HSU C H. Performance evaluation on permission-based detection for android malware[M]//Advances in intelligent systems and applications. Berlin:Springer,2013:111-120.