基于应用描述的 Android 应用异常行为检测研究

王然[1] **王浩宇**[2] **郭耀**[3] **徐国爱**[4]
（北京邮电大学网络技术研究院 北京市 100876）[1]
（北京邮电大学计算机学院 北京市 100876）[2]
（北京大学信息科学技术学院 北京市 100871）[3]
（北京邮电大学网络空间安全学院 北京市 100876）[4]

摘 要 移动应用的行为是否恶意，与用户对该应用的期望相关。例如，一个地图导航应用获取用户的位置信息是合理的，但一个声称只有计算器功能的应用获取位置信息会被很多用户拒绝。相关工作提出通过检测应用描述和应用行为的不一致性来检测恶意应用。然而，Android 应用的一个特点是广泛使用第三方库。研究表明，Android 应用中平均 60%的代码都是属于第三方库，第三方库的使用会对应用异常行为的检测造成影响。因此，基于自然语言处理、机器学习和第三方库识别技术，实现一个改进的基于描述的 Android 应用异常行为检测工具。首先，通过自然语言处理技术对应用描述进行分析，使用遗传算法对应用描述聚类并寻找最佳聚类数目，得到相似描述（功能）的应用集合。然后对应用静态分析，检测应用中的第三方库，并分析应用的敏感行为。最后，相似描述的应用集合中的具有异常行为的应用即被检测为可疑应用。工具对 Google Play 应用市场中 27.6 万余个应用进行分析，实验结果验证所提出方法的有效性且说明第三方库对 Android 应用的异常行为检测有较大影响。

关键词 Android，第三方库，应用描述分析，聚类，异常点检测

中图法分类号 TP309 **文献标识码** A DOI

Automated Detection of Abnormal Behaviors of Android Apps based on App Descriptions

WANG Ran[1] WANG Hao-yu[2] GUO Yao[3] XU Guo-ai[4]

(Institution of Network Technology, Beijing University of Posts and Telecommunications, 100876, China)[1]

(School of Computer Science, Beijing University of Posts and Telecommunications, 100876, China)[2]

(School of Electronics Engineering and Computer Science, Peking University, 100871,China)[3]

(School of Cyberspace Security, Beijing University of Posts and Telecommunications, 100876, China)[4]

Abstract Whether the sensitive behaviors of mobile apps should be granted is related to the users' expectation of the app. For example, a map navigation app is reasonable to obtain user location information, but an app that claims just to have calculator functionality gets location information that may be rejected by users. Related work proposed to detect malware by detecting the inconsistence between app description and app behavior. However, third-party libraries are widely used in Android apps. Recent study showed that more than 60% of the code belongs to third-party libraries in Android apps on average, which could greatly impact the accuracy of abnormal behavior detection. Therefore, based on Natural Language Processing, machine learning and third-party libraries detection techniques, we have implemented an improved abnormal behavior detection tool by analyzing app

到稿日期： 返修日期：
本文受国家自然科学基金（编号：61702045，61401038），国家电网公司科技项目（SGRIXTKJ[2017]265 号），国家高技术研究发展计划项目（编号：2015AA017202），广东省科学技术厅前沿与关键技术创新项目（2016B010110002），北京邮电大学青年科研创新计划专项（2017RC40），北京邮电大学智能通信软件与多媒体北京市重点实验室项目（ITSM200601）资助。
王然（1993-），男，研究生，主要研究方向为移动安全，E-mail:wangran2015@bupt.edu.cn
王浩宇（1991-），男，博士，讲师，CCF 会员，主要研究方向为移动安全，
E-mail:haoyuwang@bupt.edu.cn(通信作者)。
郭耀（1976-），男，博士，副教授，CCF 会员，主要研究方向为系统软件，Email:yaoguo@pku.edu.cn
徐国爱（1972-），男，博士，教授，CCF 会员，主要研究方向为软件安全，Email:xga@bupt.edu.cn

description. First, we analyze the app description using Natural Language Processing technique, and use genetic algorithm to cluster apps based on app descriptions and identify the optimal number of clusters, then we could obtain the app set of similar description (function). Then, the static analysis is applied to detect the third-party libraries and analyze the sensitive behaviors. Finally, in a collection of similar description android apps, the apps with abnormal behaviors are detected as outlier apps. We use the tool to analyze more than 276K apps in the Google Play app market, and experiment results show that the proposed method is effective to detect outlier apps, and the third-party libraries have a great impact on the abnormal behavior detection of Android apps.

Keywords Android，Third-party Library, App description analysis，Clustering，Outlier detection

1 引言

在移动智能终端和多样的移动应用给用户带来便利的同时，移动平台上各种新的安全和隐私问题也日益凸显。移动平台的恶意应用增长迅速，这些恶意应用会在用户不知情的情况下，通过恶意扣费、系统破坏、隐私窃取等恶意行为，给用户带来经济损失和隐私泄露问题。

移动应用的行为是否恶意，与用户对该应用的期望相关。安卓系统在应用安装时就会向用户确认该应用正常使用的权限列表，等用户同意安装后该应用才会被授予所有被请求的权限。然而大部分恶意应用都会请求与应用功能无关的权限[1][2]。例如，一个恶意壁纸应用会使用 GPS 权限发送用户的地理位置信息。但是对于一个 GPS 追踪应用，把用户的地理位置信息通过网络端发送出去是正常行为，而并非隐私泄漏。因此，用户所期望的应用功能和应用真实行为之间的差异性应当是检测恶意应用的重要指标。

因此，一些相关工作提出分析用户所期望的应用功能和应用真实行为之间的差异性，并用此差异性来对应用进行风险评估。WHYPer[3]和 AutoCog[4]使用自然语言处理技术对应用描述进行分析，然后与应用申请的权限相比较，检查应用是否申请敏感权限但没有在描述中阐述请求特定权限的原因。CHABADA 提出[5]从应用描述中提取主题，并根据主题相关度对应用聚类，得到相似描述的应用集合。通过分析相似描述的应用集合中是否有应用存在异常行为，即可检测出潜在的恶意应用。

然而，Android 应用的一个特点是广泛使用了第三方库，例如广告库、社交网络库、工具库等。研究表明，Android 应用中平均 60%的代码都是属于第三方库[6]。第三方库的使用会对应用异常行为的检测造成影响，因为大部分应用不会将跟第三方库相关的功能写在描述中。例如，很多应用使用 Google Ads 广告库来展示广告。如果根据描述对应用聚类然后分析应用的异常行为，使用了 Google Ads 广告库的应用会比其他应用检测出多了一些敏感行为特征（例如获取位置信息等），因此很有可能被认为是异常应用。此外，之前研究表明，第三方库中存在着很多冗余功能，很多代码是不可达的。因此，基于应用描述的 Android 应用异常行为检测需要将第三方库分离开来考虑，否则会导致大量误报。

基于自然语言处理、机器学习和第三方库识别技术，本文实现了一个改进的基于描述的 Android 应用异常行为检测工具。首先，使用自然语言处理技术对应用描述进行分析，使用遗传算法和 K-means++算法结合的方法，对应用描述聚类并寻找最佳聚类数目，得到相似描述（功能）的应用集合。然后对应用静态分析，检测应用中第三方库，并分析应用的敏感行为。最后，相似描述的应用集合中的具有异常行为的应用即被检测为可疑应用。值得说明的是，本文的目标是检测出具有异常行为的应用，而不是专门检测恶意应用。异常行为的应用包括恶意应用、灰色应用以及其他非恶意但是行为较为特别的应用。

将工具应用于 Google Play 应用市场中超过

27.6 万余个应用，检测出 1836 个异常行为的应用。通过对每个类别前五个应用（共计 145 个检测具有异常行为的应用）进行人工验证，分析应用异常行为的原因。实验结果表明，42%检测出的异常应用包含恶意行为，46%的应用被 VirusTotal 检测为恶意应用，验证了所提出方法能够有效检测应用的异常行为，并且能够帮助检测恶意应用。同时，对去掉第三方库前后检测出包含异常行为的应用进行对比，说明第三方库对 Android 应用的异常行为检测有较大影响。

本文主要有以下贡献：

(1)研究了第三方库对基于描述的 Android 应用异常行为检测的影响。实验结果表明，在去除第三方库后，约 54%的应用不再被检测为异常行为的应用。去除第三方库后，异常应用的特征更为明显，能够检测到新型的异常应用。

(2)提出根据应用描述的主题向量，使用遗传算法和 K-means++算法结合的方法对应用描述聚类并寻找最佳聚类数目。使用该方法比简单的尝试寻找最佳聚类数目更为高效和自动化。

(3)将实现工具应用在 Google Play 应用市场中超过 27.6 万个应用，样本集比相关工作高了一个数量级，并通过人工确认和使用 VirusTotal 检测来验证工具的有效性。

2 框架介绍

本研究框架如图 1 所示，主要分为五个部分：应用描述及应用下载模块、应用描述处理模块、应用文件处理模块、第三方库处理模块、异常应用检测模块。

2.1 应用描述及应用下载

在应用描述及应用下载模块中，在应用描述及应用下载模块中，从 Google Play 下载了共近 40 万个应用的信息，包括应用的 APK 文件以及应用介绍页面的元信息（应用名称、描述、评分、下载量等数据）。

2.2 应用描述处理模块

在该模块中，首先将所有应用的描述进行自然语言处理，包含检测应用描述所使用的语言（本文只关注英语），去除截止词、单词转化为词根形式、删除冗余的文本信息（如数字和标点符号）。结合应用文件处理模块中成功提取出 API 的应用列表，本研究最终确定使用的应用样本数为 276，333 个。之后根据整理好的应用描述进行主题提取（topic modeling），为每个应用生成一个主题向量。最后，根据应用的主题向量进行聚类，并通过遗传算法和 K-means++结合的方式找出最佳聚类数目，本文最终确定聚类数目为 29 类。

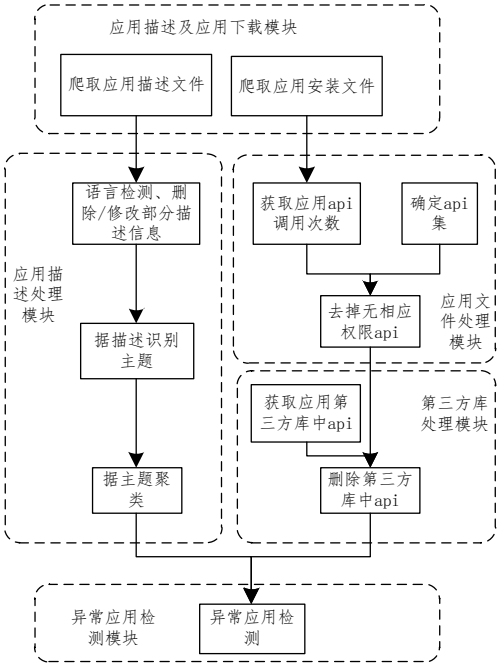

图 1 整体框架

2.3 应用文件处理模块

应用文件处理模块实现从应用安装文件（APK 文件）中提取出敏感 API 并统计敏感 API 被调用次数的功能。本研究中指定敏感 API 为与权限直接相关的 API。考虑到有部分 API 虽然代码中被其他方法调用，但调用该 API 的方法并不会被执行到的情况（存在死代码），本研究将没有声明相关权限的 API 调用次数置为零，来尽量避免统计到不可达的 API。

2.4 第三方库处理模块

第三方库处理模块中检测了所有应用的第三方库的使用情况，统计所有应用的第三方库中敏感 API 的被调用情况，并在应用文件处理模块的

敏感 API 数据基础上，减去在第三方库中敏感 API 被调用的次数，统计出在非第三方库中敏感 API 被调用次数。

2.5 异常应用检测模块

在本模块中，根据每一类簇的应用 API 调用情况进行异常点检测。每个应用均有其 API 调用向量，向量中每一元素代表相应的 API 被调用的次数，使用 Isolation Forest 算法对每一类簇的应用的 API 调用向量进行异常点检测。

3 应用描述处理

考虑到过短的描述并不能体现应用的功能，并且会影响接下来的语言检测结果，将描述少于 10 个单词的应用忽略。使用 language-detector[1]工具进行语言检测，只取描述为英文的应用。为了使识别出的主题更为准确，对应用描述中所有大写字母修改为相应的小写字母，剔除所有非英文字符（如数字，标点符号等），使用第三方工具 Mallet[2]去掉 stop words(例如 him，is，the 等)。考虑到单词的单复数亦会对识别主题工作有影响，最后使用第三方工具 snowball[3]将单词转化为词根形式。最终，结合提取 API 成功的应用名单，确定分析样本为 276,333 个。如图 2 所示，为随机抽取的经过以上所有自然语言处理后的一个应用的最终描述：

search share music easily feature beauti visual fast search album art display fast search share list simpl layout notif play music list update music play stop remov headset face crash addit feature contact san gmail

图 2：应用的描述内容示例

接下来，将 Mallet[4]的文本分析工具引入，将处理好的应用描述进行主题识别，对于识别出的主题数目，可以根据需要来设定。本研究最终设定识别的主题数目为 30，这个数字也为 Gorla[5]等人研究中所设定的主题数目。经过主题识别后，取出每个主题的前 20 个关键词列出如表格 1 所示：

提取出主题后，每个应用都有自己的主题向量，主题向量的每个元素代表该应用对对应主题的符合程度。例如，某应用主题向量如下所述：

[0.4494,0.2351,0.0030,0.0030,0.0208,0.0030,……]

该主题向量表示该应用属于 0 号主题的概率为 0.4494，属于 1 号主题的概率为 0.2351，属于 2 号主题的概率为 0.0030,等等。

表 1 主题及其前 20 个关键词

主题编号	主题关键词
0	app share friend facebook twitter messag send social email love free featur easi chat photo network post creat sms peopl
1	photo pictur imag beauti make camera frame color design galleri girl dress effect share hair style app choos fashion creat
2	app bibl church god book india prayer indian read islam audio quran chapter christian vers holi lord hindi day listen
3	game car race free play slot drive win coin machin park real truck excit featur experi speed simul fun spin
4	calcul weight number app exercis fit workout time train bodi result measur track unit simpl base math system program convert
5	app servic shop offer book order search find locat deal featur custom store product price direct special inform mobil home
6	kid anim game children learn fun babi color child play app draw cat dog educ pet pictur free age sound
7	game play level jump run control shoot fli enemi fun tap score collect power featur world challeng avoid bird zombi
8	team footbal world sport player score club countri golf match leagu game app fan live cup play soccer flag includ
9	year citi state south art area includ counti nation north american san west york world histori local award countri cultur
10	theme instal keyboard icon launcher download font app appli set free select home screen press support android widget phone function
11	app video mobil free watch content youtub movi download connect updat stream copyright channel applic latest devic offici disclaim share
12	fish christma beauti water natur magic world tree enjoy sea room night flower beach time day light blue halloween hous
13	account mobil bank app check money bill pay credit view access onlin payment secur inform manag balanc transact transfer servic
14	recip food make cook eat easi drink step kitchen ice ingredi fruit cake app delici love healthi cream restaur chicken
15	busi servic product compani provid develop market manag design industri custom profession technolog job work qualiti project offer client experi
16	music radio song listen play app station audio record artist stream free featur favorit danc download player lyric live rock
17	life make time peopl day work thing good person start feel love give find tip mind learn don't fact back
18	phone call devic mobil android app connect applic number messag contact user sms network control

[1] https://github.com/optimaize/language-detector
[2]https://github.com/mimno/Mallet/blob/master/stoplists/en.txt
[3] http://snowballstem.org/
[4] https://github.com/mimno/Mallet

	secur wifi send data password
19	event app inform school schedul view student calendar access date mobil confer featur news class offici connect communiti find updat
20	applic list real properti inform estat agreement licens data provid term licensor termin servic sale user state right relat mls
21	sound app rington quot free phone applic friend funni alarm set android make fun joke sleep inspir notif download effect
22	word english languag learn question app translat test answer dictionari letter quiz chines search spanish text french featur phrase studi
23	screen set button time app widget mode tap click display version press select phone light batteri devic chang featur start
24	game play level puzzl score time fun challeng mode player match card free number move point bubbl ball block simpl
25	news app read latest updat magazin issu articl content stori inform access featur www free world download page subscript star
26	wallpap live set background imag free screen phone app applic galaxi devic android support samsung download tablet anim pictur home
27	app inform medic health mortgag doctor care complet treatment patient time emerg diseas profession applic featur access free easi help
28	map locat citi app travel guid inform hotel find place gps weather rout tour search restaur featur trip time navig
29	app file android version support devic applic data note featur manag search list creat user code card record googl save

最后，根据 27.6 万余应用的主题向量，使用 k-means++算法进行聚类。在已有主题的基础上进行聚类，而不是直接将属于同一主题的应用聚为一类的原因是：每个应用有其自己的主题向量，并不是完全吻合某一确定的主题，而直接将应用归为主题向量中最大数值对应的主题这种做法过于极端，即并不能确定某个应用一定属于某个主题，故直接使用主题作为类簇划分标准不能得到较好效果。

K-means++算法是 K-means 算法的改进版本，其与 K-means 算法区别在于初始化聚类中心点的选择上，K-means++算法不再随机选择初始化聚类中心点，而是按照初始的聚类中心之间的相互距离要尽可能的远的原则选择聚类中心点。K-means++算法能获得更好的聚类效果，而且其算法执行完毕所需时间也更少[7]。

由于 K-means++算法的聚类数目（k 值）需要人工确认，而且使用相同的 k 值，多次聚类仍可能有不同的聚类结果，也就是说，单次运行 K-means++算法不能保证聚类结果是最佳的、最符合实际数据分布的。故需多次运行该算法，取聚类效果最好的一次聚类结果。本文使用遗传算法结合 K-means++来决定最佳聚类数目。使用遗传算法的好处是：遗传算法在兼顾对同一个 k 值多次聚类同时，还兼顾了以一定概率使用其他 k 值进行聚类寻找最优 k 值，这比简单的尝试多种 k 值更为高效和自动化。

具体遗传算法结合 K-means++的方法如下：使用 Pyevolve 框架[5]实现遗传算法，在编码方面，使用经典的“1D Binary String”，也就是说用二进制编码表示聚类数目（k 值），如“0001100”代表聚类数 12，设置二进制为 7 位，也就是说 k 值的范围为 0 到 127。设置遗传代数为 10 代，每代人口数为 20，为了能多测一些不同的 k 值，将变异率提高至 0.1。对于初代人口的 k 值，使用框架中默认的“G1DBinaryStringInitializator”函数，这个函数随机将随机产生二进制值。在遗传算法中的选择阶段，使用“GRankSelector”函数。另外，在算法的“交叉”阶段，使用“G1DBinaryStringXUniform”函数实现二进制编码之间的交叉生成子代。

设置遗传算法中的适应度函数为自己编写的函数，将遗传算法的适应值设为该次 K-means++算法聚类结果的轮廓系数。轮廓系数最早由 Kogan[8]提出，常用来评估聚类好坏程度[9]。轮廓系数的数值范围为-1 到 1，该值越靠近 1 代表聚类效果越好，越靠近-1 代表聚类效果越差。

使用工具 sklearn[6]实现 K-means++算法及其轮廓系数的计算。sklearn 是一个 python 语言的开源工具，其中包含了诸多聚类算法。是机器学习中常用的工具。

由于将 27.6 万余个应用全部用来求最佳聚类数目，耗时过长，故在所有应用中随机抽取一万个应用，亦即约 1/27 的数据量作为求解最佳

[5] http://pyevolve.sourceforge.net/
[6] http://scikit-learn.org/stable/

聚类数目的样本。最终求得在主题数为 30 的情况下，最佳聚类数目为 29 类。

4 应用文件处理

在本模块中，实现对应用的调用 API 的获取和处理工作。

本研究在本模块中提取指定的 5 万余 API 在每个应用中被调用次数。在 Gorla 等[5]的研究中，定义的敏感 API 集来源于 Felt 等[2]的研究，但其研究结果是 11 年发表的，而本研究获取到的 google play 商城的应用则是 14 年的，在这两年内，从 11 年发布的 Android 3.0 版本已经更新到 Android 4.4，因此敏感 API 集需要做更改。本研究仍然将与权限相关的 API 作为敏感 API，但 API 集来源于 PScout[7]。PScout 是一个统计了 Android 各个版本与权限相关的 API 的项目，鉴于截止到 14 年 7 月，使用最多 Android 版本是 4.1，因此本研究将 PScout 上发布的 Android 4.1 文档 API 调用映射文件中的 680 个 API 作为敏感 API 集的原始来源。本研究借鉴 Ma 等人[10]的想法：鉴于相同名称但参数不同的 API 实际功能基本一致，不考虑 API 的参数，只考虑其名字，即将参数不同但名字相同的 API 计为同一个 API 处理。在做完这项处理后，680 个 API 剩余 428 个，结合之前已提取统计的 5 万余 API 的调用情况，最后成功提取并统计的敏感 API 为 396 个。对每个应用生成 API 调用次数向量，该向量中的每一个元素代表相应 API 的调用次数。

为了尽量排除不会被调用的 API 的干扰，本研究提取每个应用声明的权限，然后检查每个应用是否声明其包含的敏感 API 需要的权限。对没有声明需要权限的 API，修改其调用次数为 0。

5 第三方库处理

为了减少和验证应用使用第三方库对异常点检测的影响，本研究使用工具 LibRadar [8][11]检测每一个作为样本的应用的第三方库使用情况。LibRadar 是一个开源的 python 语言编写的 Android 应用第三方库检测工具，使用该工具可方便的检测出应用包含的第三方库及所在包名。

获取应用的第三方库使用情况后，检查敏感 API 在第三方库中的使用情况，计算敏感 API 在非第三方库中的使用情况，如：某应用 A 中总计调用 android.telephony.SmsManager 类中的 sendTextMessage 这个敏感 API 4 次，而在其第三方库中调用该 API 3 次，则该 API 在非第三方库中调用 1 次。

6 异常应用检测

在删除第三方库中的敏感 API 后，即完成了据应用描述生成主题，据主题聚类，获取非第三方库敏感 API 调用次数的所有工作，接下来在最后的部分对每一类簇的应用分别根据敏感 API 的调用向量进行异常点检测。

本研究使用的异常点检测算法为 Isolation Forest[12]。该算法适用于大量正常数据中包含少量异常数据的异常点挖掘，该算法对内存要求较低，且因其时间复杂度为线性，故比较适合处理海量高维数据。该算法需要输入两个重要参数，一个是树的数目，一个是采样数，本研究均采用 skearn 默认值，即树为 100 棵、采样数为 256。

将 27.6 万应用的敏感 API 的调用向量作为 Isolation Forest 算法的输入，输出各个 APK 的“得分值”，该值越小说明对应的应用相对于其他应用越异常。

7 数据处理和分析

7.1 第三方库对应用异常行为检测的影响

考虑到相对于以 API 有无作为敏感 API 向量元素，以 API 被调用的次数为向量元素，包含更多的信息，能在代码层次上体现出调用相同 API 的不同应用的调用 API 频次差别，本研究以 API 被调用的次数作为向量元素。

为研究第三方库对检测异常应用的影响，本研究在去掉第三方库前使用 Isolation Forest 算法做一次异常点检测，在去掉第三方库后再次用 Isolation Forest 算法做一次异常点检测，

[7] http://pscout.csl.toronto.edu/downloads.php
[8] https://github.com/pkumza/LibRadar

结果发现在去掉第三方库前检测出的异常应用中只有 46%的应用在去掉第三方库后，仍然被判定为异常应用。为了研究这些异常应用不再判定为异常应用的原因，以及去掉第三方库后检测到新的异常应用的原因，人工对这些应用进行分析。

结果发现，大量的去掉第三方库之前被判定为异常的应用都是因使用了第三方库导致被判定为异常的，而这些第三方库有很多都并没有展现出“恶意”行为或者本身不是“恶意”的，例如，一个名为 Wooden Blocks（应用包名：com.redearstudio.woodenblocks）的实际正常的应用，在其第三方库中调用了敏感 API“ openConnection ”以及敏感 API“getActiveNetworkInfo”分别为 33 次和 34 次，但在第三方库外都只调用了 7 次，也就是说该应用第三方库中包含了频繁的网络操作，导致了该应用成为异常应用，去掉第三方库后使该应用重新判定为正常应用。

另外一种常见的情况是：应用包含了第三方库，但实际运行中却只使用了第三方库中极少的API，这就导致了未去掉第三方库前被判定为异常应用，去掉第三方库后判定为正常应用。例如，一个名为 Puzzle Code（应用包名：com.yiqusoft.puzzle）的应用，其包含了domob 和 adsmogo 这两个第三方广告库，该库代码中包含了多种敏感 API 的调用，例如调用了获取地理位置的 API “getLastKnownLocation”11 次，但实际使用过程中，该应用并未调用该API。当把第三方库去掉后，获取地理位置这一特征不再显著，应用也就不再判定为异常应用。

在未去掉第三方库被判定为异常的应用中，每类随机取 5 个应用，共计取 145 个应用，检查其是否在去掉第三方库后被判定为异常应用的名单里，并人工检查不再被判定为异常的原因，发现在不再被判定为异常的应用中有约 47%的应用不再被判定为异常的原因是使用了第三方库而第三方库没有展现恶意行为，有约 27%的应用不再被判定为异常的原因是包含的第三方库中只有很少的 API 被真正调用。

很多应用在去掉第三方库后才被判定为异常的原因是在去除第三方库后，其敏感 API 调用向量中仍有部分 API 具有较高的调用次数。例如，一个名为 friuts break 的应用（应用包名：com.appgame7.friutsbreak）在去掉第三方库后，仍然有较多的获取地理位置和手机号码的 API 的调用，而对于其他应用来说，获取地理位置和手机号码的 API 的调用大多集中在第三方库中，在除去第三方库后，获取手机号和地理位置的 API 的调用急剧减少，这样 friuts break 这个应用就会被检测为异常应用。

在已去掉第三方库被判定为异常的应用中，每类随机取 5 个应用，共计取 145 个应用，检查是否在未去掉第三方库被判定为异常的应用列表中，并人工确定去掉第三方库新被判定为异常应用的原因，发现在新被判定为异常的应用中有 43%的应用新被判定为异常的原因是在去除第三方库后，其敏感 API 调用向量中仍有部分 API 具有较高的调用次数。

7.2 异常行为应用的检测标准和方法

本研究对 Gorla 等人对“恶意应用”的判断标准进行了挖掘和拓展，制定了新的“恶意应用”评判标准，符合以下条件至少一条的应用被判定为“恶意应用”：

(1) 被手机系统报为病毒的应用。

(2) 在应用描述中没有告知获取用户个人信息（如手机号和地理位置），而实际使用中会获取个人隐私并上传服务器，并且在获取隐私数据前并未通过弹窗或用户协议等方式告知用户的。

(3) 应用实际行为与其描述有巨大出入的。

根据以上评判标准，人工手动对每一类簇前 5 个异常应用进行检测，检测方法简述如下：将应用安装在测试机上，在手动使用应用过程中，利用抓包软件对应用的网络数据进行抓包，使用过程中尽量遍历到应用的每个功能，最后评判应用实际具有的功能和应用描述的差异，并对抓取的应用数据包进行分析。

7.3 异常行为应用分析

如表 2 所示，为对去掉第三方库后程序自动检测出的每簇前 5 个异常应用人工分析结果，可见包含恶意行为的应用比率为约 42%，对某些类簇，如 16 号和 28 号，所有前 5 个异常应用均有恶意行为，而对某些类簇则有较低的恶意行为比率，如 1 号和 15 号，前 5 个被程序判断为异常的应用经人工分析后发现均无恶意行为。

去掉第三方库后检测出的包含恶意行为的异常行为应用，其行为主要有以下几类：

(1)对于获取用户隐私数据而在应用描述及应用运行过程中均不通知用户的异常应用，大多数均是在应用启动后即通过调用“getlastknownlocation”、“TelephonyManager.getLine1Number”等 API 获取用户的隐私数据后直接上传服务器，例如一个名为 CB Jungle Run（应用包名：com.junglerunnazara.com.junglerun.nazara）的单机跑酷游戏类应用，在用户不知情的情况下获取了地理位置和手机号并上传至服务器。

(2)对于应用实际行为与应用描述差别较大的应用举例如下：一个名为 POPCLOCK（应用包名：com.appblast.popclock）的简单的系统闹钟的应用，却获取了用户手机号码上传到服务器。名为 lockfingerscanner（应用包名：com.yoursite.lockfingerscanner）的声称使用手指解锁上锁的应用，实际该应用无任何声称的功能，只是带有其他应用的下载链接。

(3)此外，有部分应用在安装后，手机系统自带安全软件（本研究使用的测试机是联想 ZUK 手机）会报毒，例如：在安装一个名为 TouchNPaint（应用包名：game.child.paint）的应用后，测试机自带的“安全中心”会提示该应用为病毒应用，包含病毒“a.gray.mfpad”，并弹框提示用户立即卸载该应用。

表 2-1 每类簇前 5 个异常应用人工分析结果

类簇号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	合计
去掉第三方库后检出恶意应用数	1	0	2	2	1	5	1	5	1	2	2	3	1	2	0	28

表 2-2 每类簇前 5 个异常应用人工分析结果

类簇号	15	16	17	18	19	20	21	22	23	24	25	26	27	28	合计
去掉第三方库后检出恶意应用数	0	5	1	0	4	2	2	2	5	2	0	0	5	5	33

表 3 VirusTotal 检测结果

报毒引擎数量范围	0	[1,5]	[6,10]	[11,15]	[16,20]	[21,30]	[31, +∞)
APK 数量	78	23	5	12	12	15	0

为验证检测出来的异常应用中恶意应用的比例，将对去掉第三方库后检测的每类簇前 5 个异常应用，总计 145 个应用利用 VirusTotal 检测，结果如表 3 所示，有 67 个应用被 VirusTotal 引擎报毒，其中报毒数超过 10 个的有 39 个应用。

8 相关工作

基于用户能直观看到的应用描述、应用界面（UI）等信息，近期的研究工作尝试从用户角度出发，分析并解决用户期望与应用行为的差异。

基于应用描述的研究：Pandita 等[3]检测应用描述与其声明的权限是否一致，在此基础上 AutoCog[4]提出结合机器学习和自然语言处理的方法来处理应用描述和行为之间的关系。Gorla 等[5]通过检查应用实际功能与应用描述的差别，进行检测异常应用。Ma 等[10]在 Gorla 等人的基础上，将已知的恶意应用加入进来作为训练集的一部分，实现半监督式检测异常应用。Zhang 等[13]提出了一种自动化工具 DESCRIBEME，为应用自动生成与隐私泄露有关的描述。

基于应用 UI 的研究：PERUIM[14]和 AsDroid[15]使用程序分析技术识别和应用界面元素相关的权限，对 UI 组件中文本信息和 UI 权限的不一致性进行了分析，从而检测潜在的恶意应用。AppIntent[16]基于应用敏感行为相关的 GUI 操作序列，分析隐私信息的泄露是否为用户触发。

本研究基于 Gorla 等人的研究，实现了基于应用描述的 Android 应用异常行为检测工具，但本研究与 Gorla 等人的研究存在很大区别。本研究加入第三方库检测和处理，结果表明第三方库对应用的异常行为检测有较大影响。此外，本研究使用遗传算法和 K-means++算法结合的方法求解最佳聚类数目和最佳聚类模型，增加样本数目至 10 余倍。考虑到能用来检测未知模式的异常应用，本研究没有加入已知恶意应用作为训练。

9 总结

本文利用自然语言处理、机器学习和第三方库识别技术，通过先根据应用描述进行聚类，后对每一类簇应用根据 API 向量进行异常检测，并加入第三方库的分析，实现了一个改进的基于描述的 Android 应用异常行为检测工具。将工具应用于 Google Play 应用市场中超过 27.6 万余个应用，检测出 1836 个异常行为的应用。通过人工验证分析应用异常行为的原因，结果表明该工具在没有恶意应用先验知识的基础上检测出异常应用，并且有 42%检测出的异常应用包含恶意行为，46%的应用被 VirusTotal 检测为恶意应用。实验结果验证了所提出方法能够有效检测应用的异常行为，该工具拥有检测恶意应用的能力，能够帮助检测新型恶意应用。

参考文献

[1] Zhou Y, Wang Z, Zhou W, et al. Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets[J]. Proceedings of Annual Network & Distributed System Security Symposium, 2014.

[2] Felt A P, Finifter M, Chin E, et al. A survey of mobile malware in the wild[C]// ACM Workshop on Security and Privacy in Smartphones and Mobile Devices. ACM, 2011:3-14.

[3] Pandita R, Xiao X, Yang W, et al. WHYPER: towards automating risk assessment of mobile applications[C]// Usenix Conference on Security. USENIX Association, 2013:527-542.

[4] Qu Z, Rastogi V, Zhang X, et al. AutoCog: Measuring the Description-to-permission Fidelity in Android Applica-tions[C]// ACM Sigsac Conference on Computer and Communications Security. ACM, 2014:1354-1365.

[5] A. Gorla, I. Tavecchia, F. Gross, el al. Checking app be-havior against app descriptions. ICSE, 2014: 1025–1035.

[6] Wang H, Guo Y, Ma Z, et al. WuKong: a scalable and accurate two-phase approach to Android app clone detection[C]// International Symposium on Software Testing and Analysis. ACM, 2015:71-82.

[7] Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding[C]// Eighteenth Acm-Siam Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2007:1027-1035.

[8] Kogan J. Introduction to Clustering Large and High-Dimensional Data[M]. Cambridge University Press, 2007.

[9] Staab S, Hotho A. Ontology-based Text Document Clus-tering.[J]. Advances in Soft Computing, 2002, 4(6):48-54.

[10] Ma S, Wang S, Lo D, et al. Active Semi-supervised Ap-proach for Checking App Behavior against Its Descrip-tion[C]// IEEE, Computer Software and Applications Conference. IEEE Computer Society, 2015:179-184.

[11] Ma Z, Wang H, Guo Y, et al. LibRadar: fast and accurate detection of third-party libraries in Android apps[J]. 2016:653-656.

[12] Liu F T, Kai M T, Zhou Z H. Isolation-Based Anomaly Detection[M]. ACM, 2012.

[13] Zhang M, Yin H. Automatic Generation of Security-Centric Descriptions for Android Apps[M]// Android Application Security. Springer International Publishing, 2016.

[14] Li Y, Guo Y, Chen X. PERUIM: understanding mobile application privacy with permission-UI mapping[C]// ACM UbiComp, 2016:682-693.

[15] Huang J, Zhang X, Tan L, et al. AsDroid: detecting stealthy behaviors in Android applications by user interface and program behavior contradiction[C]// International Conference on Software Engineering. ACM, 2014:1036-1046.

[16] Yang Z, Yang M, Zhang Y, et al. AppIntent:analyzing sensitive data transmission in android for privacy leakage detection[C]// ACM Sigsac Conference on Computer & Communications Security. ACM, 2013:1043-1054.

作者联系电话：18612186396

作者邮箱：haoyuwang@bupt.edu.cn