基于动态特征选择的 Android 应用隐私风险自动化检测

高龙良 杜素果 杨金萍
(上海交通大学安泰经济与管理学院 上海 200240)

摘 要 针对 Android 应用中可能存在的用户隐私泄露问题,提出一种基于机器学习方法的自动化检测模型。该模型选择使用 App 申请的权限作为特征,动态地选取特征集,并采用四种经典的机器学习算法进行独立的训练与预测,最终确定最适用于 Android 应用的隐私风险检测模型。实验结果表明,对于隐私风险应用,该模型能够实现平均 95% 以上的识别准确率。该模型能够从多层面更好地进行应用风险管理以及用户隐私保护,具有较高的社会效益与实际应用价值。

关键词 隐私风险 权限 机器学习 动态特征选择

中图分类号 TP399 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.06.045

AUTOMATED DETECTION OF PRIVACY RISKS IN ANDROID APPLICATIONS BASED ON DYNAMIC FEATURE SELECTION

Gao Longliang Du Suguo Yang Jinping
(*Antai College of Economics and Managements, Shanghai Jiao Tong University, Shanghai 200240, China*)

Abstract Aimed at the user privacy leakage problem that may exist in Android applications, an automated detection model based on machine learning methods is proposed. This model chose to use the permission items applied by the App as features, dynamically selected the feature set, and used four classical machine learning algorithms to independently train and predict. And the most suitable privacy risk detection model for Android applications was determined. Experimental results show that the model can achieve an average prediction accuracy of more than 95% for privacy risk applications. This model can better manage application risk and protect user privacy from multiple aspects, which has high social benefit and practical value.

Keywords Privacy risk Permission Machine learning Dynamic feature selection

0 引 言

随着移动互联网高速发展,智能手机以及 Android 应用的普及越来越迅速。Statista 研究显示,截至 2020 年,全球智能手机用户已达到 35 亿[1]。同时,在手机的操作系统中,Android 系统占据着绝对的主导地位,拥有着 80% 以上的市场份额。搭载 Android 系统的智能移动设备催生了全新的移动服务体系,在当前社会的诸多领域都引发了颠覆性的变革:在线支付取代了传统货币支付模式,在线医疗让人们足不出户便能够享受专家问诊等。这种由新技术带来的冲击极大地丰富了人类的生活,但与此同时也带来了诸多问题和隐患。Android 系统的开源性虽然有利于系统自身的发展以及应用产品的多元化,但也意味着每个人都能根据自己的意图去设计开发 App 并将其投放到应用商店。信息时代的到来赋予了个人信息愈来愈高的价值,而作为人类与外界交流互通的主要渠道之一,手机以及各式各样的移动应用中往往存储着大量用户的个人信息。把用户个人信息作为攻击目标的恶意应用已经屡见不鲜,在 2010 年第一款针对用户隐私信息的 Android 恶意应用被披露以来,每年被检测出来的此类恶意应用数量都呈爆炸式增长[2],严重威胁着用户隐私安全[3]。然而,大多数应用商店对恶意应用的检测

收稿日期:2021 - 02 - 06。国家自然科学基金项目(71671114)。**高龙良**,硕士生,主研领域:隐私风险分析度量与管理,机器学习。**杜素果**,副教授。**杨金萍**,硕士生。

甄别能力远远落后于后者发展和数量增长的步伐。此外,作为移动应用开发平台,Android 系统虽然内置了用户隐私保护机制,即权限请求机制,但该机制在实际应用中却具有较大的局限性,因此效果甚微。

随着社会中隐私关注度的提升以及隐私风险问题的热化,越来越多的学者也将研究目标聚焦至 Android 应用的隐私风险检测。现有的研究根据实现方式主要可分为两类:动态分析和静态分析[4-5]。其中动态分析侧重于分析应用的运行时行为,例如动态日志[6]、CPU 负载[7]、API 调用[8]和网络流量[9]等,通过监视和跟踪这些行为来判断 App 是否存在异常。与动态分析不同,静态分析通常侧重于对静态属性的分析,如源代码、数字签名等,通过大量样本的学习从中找出风险应用与安全应用的一般差异性。相对于动态分析,静态分析在实现上更加便捷易行,资源时间消耗也更少,因此更加适用于实际应用[10-11]。此外,静态分析在本质上属于先验分析,能够在安装或使用 App 之前检测其风险性,因此能够避开恶意软件的"反侦察"。静态分析的表现性能很大程度上取决于所选择的特征集,因此特征集的选取是至关重要的一环。常被选作静态分析特征的属性有权限、静态 API 调用、源代码等。如文献[12-14]从 Android APK 文件中提取权限作为特征,并使用机器学习方法来评估权限检测恶意应用的能力。文献[15-16]通过分析静态 API Calls 来区分恶意程序和良性应用程序,并挖掘了恶意程序中广泛使用的 API 调用模式。Cen 等[17]则通过反编译 Android 应用程序的源代码,采用基于正则逻辑回归的概率判别模型对待测试的应用程序进行分类,得到了较高的准确率。与 API 调用、源代码相比,权限条目具有更加明确、易被提取、客观性好的优点,而且权限机制实质上是 App 与用户之间信息流动的关卡,只有 App 申请且用户同意了某项权限,App 才能够获取相对应的个人信息。

基于以上背景,本文提出了一种基于机器学习方法的 Android 应用隐私风险静态分析模型。该模型选取 App 中的权限声明条目作为特征,同时创新性地采用动态方式来选取特征集。大多数相似研究往往选取 Android 系统定义的所有权限条目作为特征集,或者选取其中特定的某一部分(如危险级别)作为特征集。对于前者,执行所需要的资源和时间消耗会更大,并且可能会引发过拟合问题;而后者则忽略了不同级别权限之间的关联性,可能一些能显著影响结果的权限并没有被包含在权限集中。针对以上问题,本文提出并实现了一种动态特征集选取方式,首先基于在数据集中被声明的频率对权限进行排序,根据频率从高到低动态地将声明频率更低的权限加入特征集,对于每次选取的特征集,均在大小为 3 733 的 APK 样本集上进行基于不同算法的训练和预测。本文的贡献主要有以下三点:

1)本文提出并实现了一种将权限作为特征的 Android 应用隐私风险静态分析模型,通过使用机器学习方法进行训练和拟合,最终模型能够实现 95% 以上的预测准确率,对恶意软件检测和用户隐私风险管理均有着非常好的实用意义。

2)本文采用动态的方式来选取特征集,进而找到最佳的特征集组成,能够最大程度地消除特征冗余和特征遗漏带来的影响,在理论上实现了创新,且对于实际应用具有一定的启发意义。

3)结合提出的隐私风险检测模型,本文从用户、平台、政府三个方面探讨了如何在实际应用中更好地进行用户隐私保护,并给出了一些建议。

1　Android 权限机制

样本分析数据显示,具有隐私风险的恶意软件往往更有可能申请不必要的权限或者一些敏感权限。事实上,权限已经成为恶意软件对用户隐私的一个主要攻击途径,也正因此,在本研究中,Android 应用所声明的权限被选择作为机器学习模型的特征。在 Android 系统架构中,权限机制是保护用户隐私的主要途径之一,该机制要求开发者在开发 App 时必须在 Manifest.xml 文件声明所有的权限信息。对于可能涉及到用户敏感信息的操作,用户会在安装及使用该 App 时收到权限许可请求。只有用户同意了这些请求,App 才能够进行后续对应的操作。然而,Android 系统的这种基于请求-授权模式的权限机制在实际实现上具有很大的局限性,主要原因有以下三点:1)Android 权限模型的粗粒度[18]。用户只能在安装时被提示需要对哪些权限进行授权,但并不知道这些权限具体会被用在何处、会涉及到哪些个人信息。而大多数用户显然不具备 Android 权限相关的专业知识,因此也无法根据提示做出合理的权衡和判断。2)用户的惰性。大多数用户安装及使用某款 App 的目的只是为了享受对应的功能和服务,而对权限请求进行科学的权衡和选择是一件很花费时间和精力的工作。3)隐私关注度以及隐私防范意识不够。许多用户在使用 App 的过程中可能很少考虑过隐私泄露的问题,或者持一种侥幸的心理,认为自己的社会角色并不是特别重要,因而自己的个人信息不会成为恶意软件的目标[19]。

在 Android 系统中,目前已经定义了超过 500 种权

限条目,根据其与用户信息之间的敏感度,可分为以下四个级别:

1) 普通级别(Normal Level):此级别的权限通常对用户隐私几乎没有任何风险,因此只要开发者在 Manifest. xml 文件中声明了此类权限,Android 系统会默认自动授予 App 这些权限。

2) 危险级别(Dangerous Level):危险权限控制的行为包括访问用户、设备的敏感资源或私有数据。这类权限不会被默认授予,而是需要得到用户的同意。在低版本的 Android 系统(≤5.1.1)中,用户需要在安装 App 时一次性对应用声明的危险级别权限进行授权,而更高的版本则加入了运行时权限的概念。

3) 签名级别(Signature Level):此级别的权限由系统在安装时授予,前提是请求该权限的应用程序和声明该权限的应用程序具有相同的证书签名。

4) 特殊级别(Special Level):未被归类于危险级别,但与用户隐私之间敏感度较高。在 App 请求该权限时,系统会为用户弹出一个详细的管理界面帮助进行选择。

虽然 Android 系统的权限机制在实际应用上无法有效地保护用户隐私,但权限在实质上依然是 App 与用户之间信息流动的关卡,因为只有某个用户同意了权限许可请求,App 才能够获取相对应的个人信息。因此本研究中选择权限作为 Android 应用隐私风险静态分析模型的特征是非常合理的。

2 Android 应用隐私风险检测模型

针对具有隐私风险的 Android 应用,本文提出了一种基于权限特征的静态分析模型,如图 1 所示,该模型包含三个部分:第一部分为权限信息的提取过程,通过反编译工具样本中 APK 文件的权限信息以文本格式提取出来;第二部分为特征集的选取过程,与其他研究不同,本文开创性地采用了动态的特征集选取方式,从不同大小的特征集中找出最优的特征集组成,从而同时解决了困扰很多已有研究的过拟合问题和欠拟合问题;第三部分则是机器学习模型的拟合训练和预测,本研究将在该部分使用多种算法进行训练,消除单一算法可能导致的偶然性误差。通过该模型,我们能够通过随机抽样得到的样本快速找出隐私风险应用与无风险应用在权限模式上的差异。在统计学允许的误差下,只要得到某项未知 App 的权限信息,训练后的模型便能够识别该 App 对用户而言是否具有隐私风险。

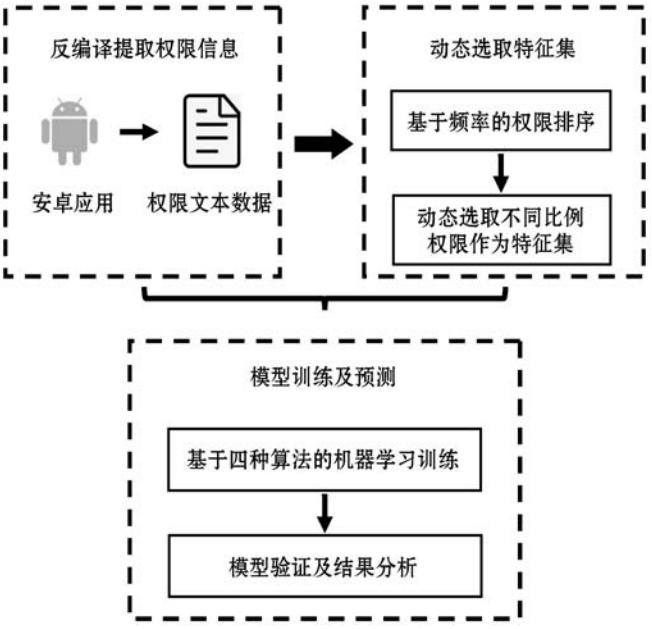


图 1 Android 应用隐私风险检测模型

2.1 数据收集

本文使用的原始数据为 Android 应用的 APK 文件,包括两部分:已标注具有隐私风险的 APK 文件(Abnormal)和无隐私风险的安全 APK 文件(Normal)。本文的 Abnormal 数据来自 Android Malware Dataset(AMD),该网站收集了大量经过充分研究后被鉴定为恶意软件的 APK 文件,并且对每个 APK 文件都注明了该应用的详细配置信息和恶意行为。通过将个人隐私泄露行为作为筛选条件,本文在该网站上收集到 2 000 个 APK 文件作为 Abnormal 数据集,表 1 中列出了 Abnormal 数据集的详细组成。同时,我们从中国市面上的一个主流 App 市场——应用宝上爬取了 1 733 个正常应用,组成 Normal 数据集。每个被爬取的应用都已记录在“315 可信应用白名单”[20]中,这意味着 Normal 数据集中的每个 APK 文件都已经通过相关安全部门的检测,不会对用户的隐私造成危害。Normal 数据集的详细组成和类别信息见表 2。

表 1 Abnormal 数据集构成

分类	家族	数量
Airpush	Adware	786
Bankbot	Trojan-Banker	276
DroidKungFu	Backdoor	127
Gumen	Trojan-SMS	145
Mecor	Trojan-SPY	183
Mseg	Trojan	133
RuMMS	Trojan-SMS	350

表 2　Normal 数据集构成

功能分类	数量	功能分类	数量
儿童	79	新闻	97
教育	113	办公	98
娱乐	78	摄影	98
理财	73	阅读	82
健康	91	安全	63
生活	110	手机美化	28
音乐	77	购物	105
社交	85	手机系统	81
通信	35	工具	101
交通出行	91	旅游	92
视频	58	—	—

2.2　反编译提取权限文本信息

在 2.1 中所获取到的原始数据均为 APK 压缩文件,而在 Android 系统架构中,一个 App 的所有权限信息都会在 Manifest. xml 文件进行声明。因此需要对原始数据进行处理,提取出所有 APK 样本的权限条目信息。从 APK 文件中提取权限信息的方式有多种,本文选择使用 Android 组件工具 aapt 对 APK 文件进行反编译以提取每个样本的权限条目信息,对于每个 APK 样本,提取其权限条目信息并写入到 txt 文件中,如图 2 所示。首行为该 APK 文件的包名信息,后续则是该应用所声明的所有权限,均为 Android 标准化的格式,因此可以很方便地使用文本处理方式进行分析。

```
package: com.appsministry.litres.book163913
uses-permission: name='android.permission.INTERNET'
uses-permission: name='android.permission.WAKE_LOCK'
uses-permission: name='android.permission.WRITE_SETTINGS'
uses-permission: name='android.permission.WRITE_EXTERNAL_STORAGE'
uses-permission: name='android.permission.ACCESS_NETWORK_STATE'
uses-permission: name='android.permission.CHANGE_CONFIGURATION'
uses-permission: name='android.permission.READ_PHONE_STATE'
uses-permission: name='android.permission.ACCESS_COARSE_LOACTION'
uses-permission: name='android.permission.ACCESS_FINE_LOCATION'
uses-permission: name='android.permission.ACCESS_WIFI_STATE'
```

图 2　权限条目文本信息

2.3　动态选取特征集

静态分析具有低成本的优势,而且可以在不安装或运行 App 的情况下检测应用的风险性。然而,静态分析的表现结果在很大程度上取决于所选择的特征集。选取所有权限作为静态分析的特征集,这种做法毫无疑问不会遗漏任何信息,但是其中显著性很小甚至完全是噪声的特征会给我们的模型带来误差,也容易导致训练过程的过拟合问题。此外,更大的特征集往往意味着需要消耗更多的训练时间、更多的 CPU 和内存。而在另一方面,只选择一部分权限也不一定是一个更好的选择,因为缺少任何一个显著影响预测结果的权限都可能会大大降低模型的性能。

上述两种特征集的选取方式均可以看作是静态的,实质上,对于任何一个未知的 App 数据集,采用静态的特征集选取方式都难以避免出现噪声过多导致的过拟合问题或者遗漏重要特征而导致的欠拟合问题,进而干扰最终结果。因此,本文认为应该使用一种动态的特征集选取方式,在某个维度下实现对所有待选特征的遍历,从中找到最优的特征集组成。在实际应用中,可通过随机取样先对小样本数据进行实验找到最优特征集,然后应用于整个数据集,这样可以在最大程度上避免出现之前所描述的问题。关于遍历维度,根据研究目的、方法等可以有多种选择,本文研究中选择了权限被声明的频率作为遍历维度,并假设如下:被声明频率越高的权限在 Android 应用隐私风险的检测中的重要性越大。值得说明的是,该假设严格意义上并非指被声明频率越高的权限在区分隐私风险 App 上总是比声明频率低的权限更加显著,而是指模型的特征集应该尽可能包含被声明频率更高的权限。因为被声明频率越高的权限往往包含相对更多的信息,且由于样本选择而导致的偶然性更低,这一点也说明了该假设的合理性。

基于该假设,本文提出的动态特征集选择方式主要思路为:首先分别对 Abnormal 数据集和 Normal 数据集中的权限条目按频率进行排序,然后根据频率从高到低逐渐增加被选取为特征集的权限比例,这样得出的特征集中总是能包含被声明频率最高的那些权限。通过动态地改变这个比例,便能够得到不同大小的特征集,进而找到最佳的特征集使得模型性能达到最优。动态选取特征集能够综合完善上面两种特征集选取方法的局限性,在全面考虑应用程序权限与隐私风险之间的潜在关联的前提下,达到准确性、泛化能力和简便性的统一。

在进行特征集选取之前,本文首先计算每个权限在两个样本集中分别被声明的频率,如式(1)所示(本文所有公式中的变量及其含义见表 3)。随后根据被声明的频率对所有出现的权限进行降序排序,通过排序结果,我们可以初步获取两个样本集权限声明的大致分布情况。

$$F_{\mathrm{Ab}}^{i}=\frac{n_{\mathrm{Ab}}^{i}}{N_{\mathrm{Ab}}}\quad F_{\mathrm{Nor}}^{i}=\frac{n_{\mathrm{Nor}}^{i}}{N_{\mathrm{Nor}}}\tag{1}$$

表 3 公式中的变量含义

变量	含义
F	权限在数据集中被声明的频率
n	数据集中声明某权限的 App 数量
N	数据集中的 App 总数量
Ab	表示 Abnormal 数据集
Nor	表示 Normal 数据集
i	表示某项权限 *i*
j	表示某款 App *j*
FS	表示被选取的特征集
t	选取为特征集的权限的比例($0 \leqslant t < 1$)
m	当前特征集的大小,即被选为特征的权限的数量
V	App 的权限声明信息向量
p	App 是否声明了某个权限:声明为 1,未声明为 0

图 3 和图 4 中分别根据被声明频率由高到低列举出了 Normal 样本集和 Abnormal 样本集中出现频率最高的前 20 种权限,可以看出,除了有数的几个权限之外,其他绝大多数权限在两个数据集中出现的频率具有较大差异,这一点也证明了权限在检测 Android 应用隐私风险中的显著性。与此同时,从图中也可以观察出一个有趣的现象:大多数权限在 Normal 数据集中出现的频率都比在 Abnormal 数据集中出现的频率高,这表明一个 App 申请权限的数量并非严格和其隐私风险成正比。此外,对于相当大一部分权限,例如“SET_ALARM”,无论是在 Abnormal 数据集还是 Normal 数据集中,该权限出现的概率都低于 1%。这种被声明概率很小的权限实质上可能对模型的贡献率非常低,甚至完全有可能以噪声存在。考虑到这一点,本文提出的假设以及基于该假设的动态选取特征集方法是必要且合理的,通过动态地调动比例去找到最优的特征集,从而剔除那些重要性微乎其微的权限。

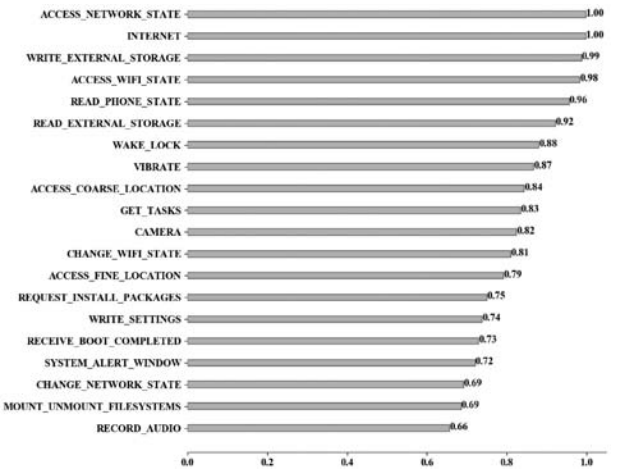


图 3 Normal 样本集中申请频率 Top20 的权限

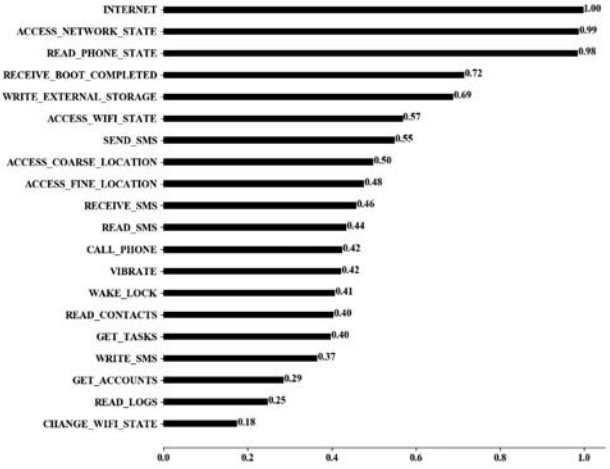


图 4 Abnormal 样本集中申请频率 Top20 的权限

在之前排序的基础上,本研究中通过引入比例数值 t 来进行动态地选择特征集:分别从 Abnormal 样本集和 Normal 样本集选取声明频率大于等于比例 t 的所有权限,将两者的并集作为特征集,如式(2)所示。通过设置比例数值,我们可以得到不同大小和粒度的特征集,如当 t 的值为 0 时,特征集包含样本集中被声明的所有权限。对不同的特征集分别进行模型训练,并对最终模型的预测结果进行比较分析,我们可以得到最优的特征集如式(2)所示。

$$FS = \{ i \mid F^{i}_{\mathrm{Ab}} \geqslant t \} \cup \{ i \mid F^{i}_{\mathrm{Nor}} \geqslant t \} \tag{2}$$

2.4 模型训练及预测

基于不同的特征集,本文将每个 App 的权限信息转化为一个长度与当前特征集大小相同的 0 - 1 向量,具体实现过程如式(3) - 式(4)所示。

$$V^{j} = (p^{j}_{1}, p^{j}_{2}, \cdots, p^{j}_{m}) \tag{3}$$

$$p^{j}_{i} = \begin{cases} 1 & i \text{ 被 } j \text{ 所请求} \\ 0 & \text{其他} \end{cases} \tag{4}$$

在实验中,本研究将比例数值 t 的步长设置为 0.1,意味着共将得到 10 个不同的特征集,这些特征集的组成如表 4 所示,对于每个特征集都进行一次独立的机器学习训练和验证过程。实际应用中可以通过设置更加细粒化的步长以实现更精准的特征集掌控。

表 4 根据不同的 *t* 值动态选取的特征集大小

t	特征集大小	*t*	特征集大小
0	512	0.5	24
0.1	67	0.6	20
0.2	47	0.7	17
0.3	38	0.8	12
0.4	29	0.9	6

在预测模型的训练阶段,本文选用了四种机器学习算法,分别为:Logistic 回归(Logistic Regression)、支持

向量机(SVM)、随机森林(Random Forest)和 XGBoost。

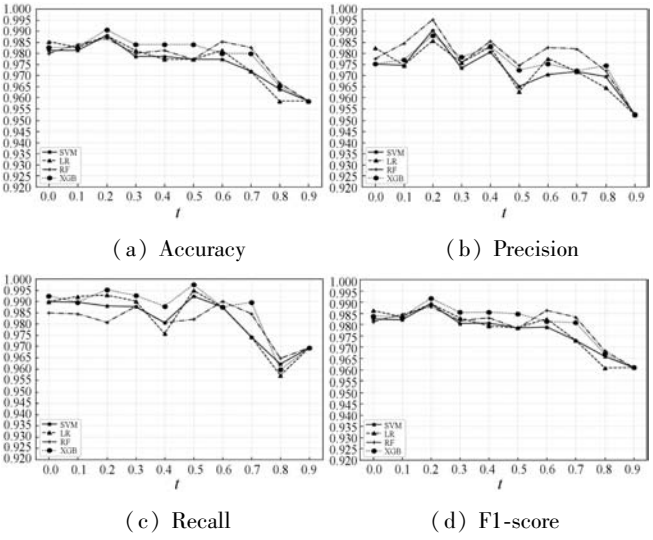
逻辑回归是广义线性模型的一种。与以连续值为因变量的线性回归不同,Logistic 回归的目的是预测样本属于每个类别的概率。预测模型的目的是将应用程序判别为是否具有隐私风险,因此引入 Logistic 回归作为分类算法是非常合适的。SVM 算法于 1963 年首次被提出,它是一种广义线性分类器,以监督学习的方式工作。支持向量机的决策边界为求解两类样本的最大边缘超平面。此外,通过设置软边缘或者引入复杂的核函数,支持向量机也可以很好地处理非线性问题。随着特征集的变化,预测模型面临的分类问题并不一定都是线性的,考虑到这种情况时,使用支持向量机算法就显得十分必要。前两种分类算法都属于弱分类器算法,而后两种算法则属于决策树的集成分类器算法。随机森林继承了经典的 Bagging 思想,对于每棵树,随机森林通过 bootstrap 抽样得到与原始数据集同等大小的数据集,并利用未选择的样本进行预测,评估误差。此外对于单树中的每个节点,随机森林还会对特征进行随机选择,并从中选择最优的特征。XGBoost 算法是近年来兴起并流行起来的一种强大分类算法。作为梯度增强决策树(gradient boosting decision tree,GBDT)的改进,在每一轮训练中,XGBoost 都根据上一轮的残差进行预测,并采用牛顿法将损失函数泰勒展开到二阶。并且,XGBoost 还在损失函数中引入了正则化项,从而能够一定程度上减小模型的方差,缓解过拟合。

对于由不同比例数值 t 确定的每个特征集,我们都进行一次独立的机器学习训练和验证。对于不同特征集和不同算法训练得到的预测模型,本文选取以下四种指标来进行评价:Accuracy、Precision、Recall、F1-Score。Accuracy 反映了模型在整个样本数据集上的表现性能,即在整个样本集上正确分类的概率。Precision 主要反映模型所检测出来的隐私风险应用中,实际上有多少 App 是真正具有隐私风险的。Precision 值越高,模型的误判率越低。Recall 则反映了模型检测出的隐私风险 App 占样本中所有隐私风险 App 的比例。F1-score 是 Precision 和 Recall 的结合,是对机器学习模型比较全面的评价指标。F1-Score 得分越高,意味着分类模型越稳健、综合表现越好。

本文的训练过程均采用了五折交叉验证。使用交叉验证能够一定程度上消除由数据集切分带来的偶然性误差,增加模型的正确性和泛化能力。

2.5　实验结果分析

图 5 给出了在不同比例数值 t 下基于不同算法的四种分类器的各项指标表现情况。同时,表 5 列出了不同比值 t 下的四种测量值的平均值。



(a) Accuracy　　(b) Precision

(c) Recall　　(d) F1-score

图 5　基于不同算法的分类器的预测结果

表 5　不同 t 值下各评价指标的平均值(%)

分类器	Accuracy	Precision	Recall	F1
SVM	97.27	96.94	97.98	97.45
LR	97.17	96.78	97.96	97.37
RF	97.44	97.43	97.77	97.60
XGB	97.52	97.27	98.10	97.67

表 5 中的结果显示,基于四种不同算法独立训练得到的分类器在测试集上均有着非常好的预测表现,平均预测准确率甚至均达到 97%,这很好地验证了将权限作为 Android 应用隐私风险检测特征的正确性,证明了权限信息确实是检测 Android 应用是否具有隐私风险的正确且关键性的特征。此外,从图 5 中可以很直观地看出,对于不同的比例数值 t,各个分类器均能保证一个相对良好的预测表现,预测准确率均能保持在 94% 以上,这表明本文中的模型还具有良好的鲁棒性,这一点为将该模型拓展至实际应用增加了理论保障。

本文通过改变比例数值 t 来进行动态地选取特征集,当 t 取值为 0 时,代表当前选取的特征集中包含了所有种类的权限,随着 t 值增大,特征集中的权限种类也在减少。特征维度的减少一般将会不可避免地导致信息的部分遗失,进而影响最终模型的表现精度。从图 5 中可以看出,当比例数值 t 越来越大,这意味着较少的权限将被选择为特征,分类器的表现性能总体上呈下降趋势,但却并非是严格下降,甚至在某些阶段表现反而上升。例如在图 5 中,当比率 t 从 0.1 增加为 0.2 以及从 0.5 变化到 0.6 的过程中,大多数分类器在所有四种指标评价下却都具有更好的表现。在训练

过程中,特征维度的减少会导致部分信息的丢失,但一定程度上也会缓解过拟合效应。而上述两个阶段中,缓解了过拟合效应给模型带来的增益显然要高于特征维度减少带来的损失。通过动态地遍历选取不同大小的特征集,我们可以实现过拟合和欠拟合之间的平衡并找到最优点,这一点证明了动态选择特征集的必要性和正确性。在实际应用中,动态选取特征集的方式也是相当必要的。特征维度的增加带来的问题一方面是训练过程中容易过拟合,另一方面也会因为特征过多而导致训练任务复杂,所需时间和内存开销更大。而通过在小样本集上进行动态选取特征集的方式去确定最佳的特征集组成能够避免不必要的资源浪费,实现性能好、成本低的统一。此外,从表 5 和图 5 都可以看出,基于集成算法的两种分类器的性能都优于其余两种分类器。相比弱分类器算法,集成算法一定程度上能够克服前者更容易出现的过拟合及欠拟合问题。因此,如果在实际中需要稳定准确地检测大批量 Android 应用的隐私风险,集成算法将会是更好的选择。

3 研究成果与应用讨论

3.1 研究成果

针对恶意软件导致的隐私风险问题,本文首先介绍了当前隐私风险检测的主要技术手段,并对其特点和不足进行了细致的分析。在此基础上,本文提出并实现了一个基于权限的 Android 应用隐私风险检测模型,采用机器学习方法,将 App 所声明的权限信息作为特征,并开创性地提出了一种新的动态特征集选取方法,为实现更准确、更稳定、更便捷的 Android 应用隐私风险检测提供了一个方案。通过使用本文中的模型,实际应用中检测风险/恶意 App 将能够达到性能好、成本低、执行方便的统一。

3.2 本文模型实际应用的讨论

信息化时代的今天,个人信息被赋予了前所未有的价值,隐私泄露对用户造成的损失也将会十分巨大,甚至可能会进而引发一系列社会问题。在验证了本文所提模型对 Android 应用隐私风险检测的有效性之后,本段将基于该结论进一步从用户、应用商店(平台)、政府三个层面探讨如何能够通过本文中的模型更加有效地保护用户隐私。

用户层面:作为隐私泄露的受害方,用户实质上完全可以通过自身来大幅降低个人隐私被泄露的概率。除了提升隐私防范意识、增加隐私关注程度等主观行为外,本文中的模型也能够帮助用户进行自我隐私保护。将本文中的模型实现为一个对用户开放的应用检测平台或 App,对于每个将要下载使用的 App,用户都能够直接通过该平台或 App 来检测其是否有隐私风险,从而提前进行规避。

应用商店层面:应用商店,如 Google Play 等为用户体验 App 最主要的下载源,大多数 App 在开发完毕后也会被上线到各应用商店中。作为 App 与用户的中间平台,在应用商店中进行集中式的应用隐私风险检测,使用本文中所提出的隐私风险检测模型,并对检测出含有潜在风险的 App 进行相关剔除处理或者警示标记,毫无疑问将会对用户隐私保护起到十分显著的作用。此外,应用商店作为中间平台,也应该严格做到每个 App 相关信息的公开展示,如 App 申请的权限条目、App 版本信息、开发者信息、相关隐私政策等,实现 App 与用户间的信息对称化。

政府层面:随着隐私泄露相关事件发生得愈发频繁,涉及的损失金额也越来越大,这一问题显然不再仅仅关乎个人利益,更值得整个社会保持警惕。在 App、应用商店、用户三者之间,政府相关部分可以以一个外部角色进行调控,比如使用本文中的模型建立一个对社会公开的检测平台,对各应用商店在线的 App 进行不定期的抽查检测,从而起到对应用商店的监督作用。此外,该平台也可对检测后得到的典型的恶意软件行为、特征等进行公示公开,帮助公民用户提升对此类应用的甄别能力。

面对隐私泄露问题,我们的社会缺少的并不仅仅是技术上的解决手段,更需要用户、平台(应用商店等)、政府三方合力来有效地利用技术手段,从各源头杜绝这类问题的发生,合力营造一个清洁美丽的网络世界。

4 结 语

面对层出不穷的恶意软件试图窃取用户的隐私信息,本文构建并实现了一个基于权限声明的 Android 应用隐私风险检测框架。该框架在本文的数据上实现了对隐私风险应用的较高的识别率。在该框架模型的基础上,本文进一步对如何有效进行用户隐私保护进行了探讨,并给出了相关意见。考虑到用户间的差异性,下一步的工作,我们将继续深入优化该框架系统,进一步挖掘更为细粒度的动态选取特征集方法,并尝试加入一些用户相关的元素,更深入地去分析如何帮助用户个性化地保护其个人隐私信息。

参 考 文 献

[1] Number of smartphone users worldwide from 2016 to 2021,

with forecasts from 2023 to 2028[EB/OL]. [2021 - 02 - 06]. https://www. statista. com/statistics/330695/number-of-smartphone-users-worldwide/.

[2] 2018 Malware Forecast: The onward march of Android malware[EB/OL]. [2021 - 02 - 06]. https://news. sophos. com/en-us/2017/11/07/2018-malware-forecast-the-onward-march-of-android-malware/.

[3] 吴中超,许国祥,王瑜. 安卓应用中的信息泄露问题探究[J]. 网络安全技术与应,2017(12):94 - 96.

[4] 范铭,刘烃,刘均,等. 安卓恶意软件检测方法综述[J]. 中国科学:信息科学,2020,50(8):1148 - 1177.

[5] 刘玮,李蜀瑜. Android 移动应用检测研究[J]. 计算机应用与软件,2019,36(6):322 - 326.

[6] Enck W, Gilbert P, Han S, et al. TaintDroid: An information -flow tracking system for real time privacy monitoring on smartphones[J]. ACM Transactions on Computer Systems, 2014,32(2):1 - 5.

[7] Das S, Liu Y, Zhang W, et al. Semantics-based online malware detection: Towards efficient real-time protection against malware[J]. IEEE Transactions on Information Forensics & Security,2016,11(2):289 - 302.

[8] Afonso V M, Amorim M F, Gregio A R, et al. Identifying Android malware using dynamically obtained features[J]. Journal of Computer Virology and Hacking Techniques, 2015,11(1):9 - 17.

[9] Wang S, Yan Q B, Chen Z X, et al. Detecting android malware leveraging text semantics of network flows[J]. IEEE Transactions on Information Forensics and Security,2017,13(5):1096 - 1109.

[10] 王亚洲,王斌. 基于深度学习的安卓恶意应用检测[J]. 计算机工程与设计,2020,41(10):2752 - 2757.

[11] 杨鸣坤,罗锦光,欧跃发,等. 基于 API 和 Permission 的 Android 恶意软件静态检测方法研究[J]. 计算机应用与软件,2020,37(4):53 - 58,104.

[12] 徐永盛,纪跃波. 基于权限特征的一种 Android 应用风险评估策略[J]. 计算机应用与软件,2020,37(4):69 - 74,100.

[13] Sanz B, Santos I, Laorden C, et al. Puma: Permission usage to detect malware in android[M]//Advances in Intelligent Systems and Computing. Berlin: Springer,2013:289 - 298.

[14] 王家琰,徐开勇,戴乐育. 一种基于权限特征的 Android 恶意应用检测方法[J]. 计算机应用与软件,2018,35(3):316 - 320,326.

[15] 崔艳鹏,颜波,胡建伟. 基于抽象 API 调用序列的 Android 恶意软件检测方法[J]. 计算机应用与软件,2019,36(9):321 - 326.

[16] Tao G H, Zheng Z B, Guo Z Y, et al. MalPat: Mining patterns of malicious and benign Android APPs via permission-related APIs[J]. IEEE Transactions on Reliability,2017,67(1):355 - 369.

[17] Cen L, Gates C S, Si L, et al. A probabilistic discriminative model for android malware detection with decompiled source code[J]. IEEE Transactions on Dependable and Secure Computing,2014,12(4):400 - 412.

[18] 谢佳筠,伏晓,骆斌. Android 防护技术研究进展[J]. 计算机工程,2018,44(2):163 - 170,176.

[19] Felt A P, Ha E, Egelman S, et al. Android permissions: User attention, comprehension, and behavior[C]//8th Symposium on Usable Privacy and Security,2012:1 - 14.

[20] 315 可信应用白名单[EB/OL]. [2021 - 02 - 06]. https://baijiahao. baidu. com/s? id = 1627966159909264962&wfr = spider&for = pc.

(上接第 304 页)

[14] Zhou G B, Wu J X, Zhang C L, et al. Minimal gated unit for recurrent neural networks[J]. International Journal of Automation,2016,13(3):226 - 234.

[15] Collins J, Sohl-Dickstein J, Sussillo D. Capacity and trainability in recurrent neural networks[EB]. arXiv:161109913, 2016.

[16] Ororbia A, Elsaid A, Desell T. Investigating recurrent neural network memory structures using neuro-evolution[C]// Genetic and Evolutionary Computation Conference,2019:446 - 455.

[17] Arpit D, Kanuparthi B, Kerg G, et al. H-detach: Modifying the LSTM gradient towards better optimization[EB]. arXiv: 181003023,2018.

[18] Majumdar A, Gupta M. Recurrent transform learning[J]. Neural Networks,2019,118:271 - 279.

[19] Cho K, Nboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[EB]. arXiv:14061078,2014.

[20] Weiss G, Goldberg Y, Yahav E. On the practical computational power of finite precision RNNs for language recognition [EB]. arXiv:180504908,2018.

[21] Britz D, Goldie A, Luong M T, et al. Massive exploration of neural machine translation architectures[EB]. arXiv:17030 3906,2017.

[22] Zhou C T, Sun C L, Liu Z Y, et al. A C-LSTM neural network for text classification[EB]. arXiv:151108630,2015.

[23] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research,2014,15(1): 1929 - 1958.

[24] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[EB]. arXiv:12070580,2012.