计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(6):148−153 [doi: 10.15888/j.cnki.csa.007932]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

利用单分类 SVM 算法检测 Android 应用程序①

管　峻[1], 毛保磊[2], 刘慧英[1]

[1](西北工业大学 自动化学院, 西安 710072)
[2](郑州大学, 郑州 450001)
通讯作者: 毛保磊, E-mail: maobaolei524@gmail.com

摘　要: 目前, Android 应用市场大多数应用程序均采取加壳的方法保护自身被反编译, 使得恶意应用的检测特征只能基于权限等来源于 AndroidManifest.xml 配置文件. 基于权限等特征的机器学习分类算法因为恶意应用与良性应用差异性变小导致检测效果不理想. 如果将更加细粒度的应用程序调用接口 (Application Program Interface, API) 作为特征, 会因为应用程序加壳的原因造成正负样本数量的严重失衡. 针对上述问题, 本文将大量的恶意应用作为训练样本, 将良性应用样本作为新颖点, 采用单分类 SVM 算法建立恶意应用的检测模型. 相比于二分类监督学习, 该方法能有效地检测出恶意应用和良性应用, 具有现实意义.

关键词: 安卓; 单分类算法; 支持向量机; 恶意应用检测

引用格式: 管峻,毛保磊,刘慧英.利用单分类 SVM 算法检测 Android 应用程序.计算机系统应用,2021,30(6):148–153. http://www.c-s-a.org.cn/1003-3254/7932.html

Android Malware Detection Based on One Class SVM Algorithm

GUAN Jun[1], MAO Bao-Lei[2], LIU Hui-Ying[1]

[1](School of Automation, Northwestern Polytechnical University, Xi'an 710072, China)
[2](Zhengzhou University, Zhengzhou 450001, China)

Abstract: At present, most benign applications in the Android market adopt a shelling method to protect themselves from being decompiled so that the detection of malicious applications can only rely on the permissions from AndroidMnifest.xml. However, the machine-learning-based classification algorithm based on permission features has a poor detection effect because of a small difference between malicious applications and benign applications. If a more fine-grained Application Program Interface (API) is taken as a feature, a serious imbalance in the number of positive and negative samples will be caused due to application shelling. In response to the above problems, with a large number of malicious applications as training samples and some benign applications as the point of novelty, we use the one-class SVM algorithm to establish a detection model for malicious applications. Compared with two-class supervised learning, this method can effectively distinguish malicious applications from benign applications, which has practical significance.

Key words: Android; one class learning; Support Vector Machine (SVM); malware detection

随着 5G 时代的到来, 移动智能终端成为必不可少的重要载体. 因此, 针对移动终端的恶意攻击层出不穷, 尤其是市场占有率第一的 Android 系统. 360 安全大脑发布的《2019 年 Android 恶意软件专题报告》[1] 指出, 2019 年全年, 360 安全大脑共截获移动端新增恶意软件样本约 180.9 万个, 足以说明恶意应用时刻威胁着移动智能终端用户的切身利益、不同行业领域的正常发展, 甚至影响到国家的安全建设. 目前, 恶意应用的检

① 基金项目: 河南省高等学校重点科研项目 (21A520041)
Foundation item: Key Research Program of Higher Education, Henan Province (21A520041)
收稿时间: 2020-09-08; 修改时间: 2020-09-25, 2020-10-13, 2020-10-20; 采用时间: 2020-11-04; csa 在线出版时间: 2021-06-01

测技术主要分为静态检测和动态检测. 从效率上讲, 静态检测的效率高, 适用于大规模的恶意应用检测. 动态检测需要实际运行应用程序, 在运行期间捕获恶意行为, 效率相对较差, 但可以应对代码混淆等对抗静态分析的手段. 另外, 无论良性应用还是恶意应用出于对自身的保护, 越来越多地采取加壳的方式保护自身被反编译. 这也造成恶意应用检测特征只能来源于应用程序反编译后得到的 AndroidManifest.xml 全局配置文件, 该文件提供应用程序申请的权限、组件等信息.

虽然过去许多基于权限、组件信息等作为特征的机器学习算法研究取得了较好的检测结果, 但实验选取的恶意应用样本大多数是 2013 年左右的, 与同时期及现在下载的良性应用在权限的使用上存在明显差异. 本文对收集的 2013 年左右的恶意应用、VirusShare[2] 收集的 2016 年恶意应用和 2020 年随机从小米应用市场下载的良性应用申请的权限信息进行统计, 2013 年的 1200 个恶意应用平均申请权限 13.31 个, 2016 年 VirusShare 收集的恶意应用平均申请权限为 30.41 个, 从小米应用市场下载的 1500 个应用程序平均申请权限 32.22 个. 由此可以看出, 恶意应用和良性应用申请的权限从数量上来讲差异变小, 而且进一步分析发现在具体权限的使用上也十分接近.

因此, 仅依靠权限作为机器学习算法特征已经很难达到过去的检测效果, 需要更加细粒度的 API 作为特征参与检测, 多类的应用程序特征有利于提高检测的准确率. 然而, 应用程序的加壳使得研究者无法通过对应用程序反编译获取应用程序调用的 API, 尤其是来源于正规的第三方应用市场的应用程序. 广大研究者的恶意应用样本主要来自于公共的数据库, 其中大多数样本可以被反编译. 这样就会出现机器学习正负样本的严重失衡, 导致现有基于二分类的恶意应用检测方法出现欠拟合、过拟合的情况, 严重影响检测的效果.

针对正负样本数量严重失衡的情况, 本文采取单分类 SVM 算法检测应用程序, 将 11 900 个 VirusShare 提供的 2016 年收集的恶意应用分为训练集和测试集, 将前期收集的 187 个良性应用作为新颖点, 主要作出的贡献如下:

(1) 通过对 2013 年左右收集的恶意应用、2016 年恶意应用和良性应用进行反编译分析, 发现无论是权限还是 API 的调用, 2013 年左右的恶意应用与良性应用差异大, 2016 年的恶意应用则相反, 说明本文提出的检测方法具有现实意义.

(2) 为了提高机器学习检测的效果, 对 2016 年收集的恶意应用和良性应用进行反编译, 根据特征使用频率进行归一化处理, 提取具有差异性的权限、API 作为机器学习算法的特征, 提高了机器学习分类器的性能.

(3) 本文采取单分类 SVM 算法解决了在恶意应用、良性应用样本数量严重失衡情况下对 Android 恶意应用的检测, 解决了欠拟合、过拟合的问题, 相比于二分类算法明显提高了恶意应用检测的效果, 对恶意应用、良性应用的检测同时有效.

1 相关工作

随着恶意应用指数级的增长, 机器学习算法广泛应用于恶意应用的检测. Singh 等[3] 将 Android 应用程序使用的权限和调用的 API 作为机器学习的特征, 取得了较好的恶意应用分类准确率. Shang 等[4] 利用信息增益法提取权限特征, 采用改进后的朴素贝叶斯算法对恶意应用进行检测. 但包括上述研究在内的许多研究方法都是采用正负样本平衡的数据集, 对正负样本严重失衡的数据集检测效果不理想.

针对正负样本严重失衡的研究主要集中在样本采样上, 分为欠采样技术和过采样技术, 但无论采取哪种技术目的都是为了达到正负样本数量的平衡[5]. 文献 [6–8] 基于少数类的现有样本去构建同类样本, 增加少数类样本的数量. 文献 [9] 采取孤立森林算法 (Isolation Forest) 非监督学习算法, 基于多数类样本的分布通过轮盘旋转算法选取多数类样本, 通过 K-means 方法形成若干多数类样本聚类中心, 实现正负样本数量的均衡. 文献 [10] 针对正负数据不均衡的问题, 提出基于迭代提升欠采样的集成分类方法, 从多数类中欠采样, 构建弱分类器并通过加权组合方式构成一个强分类器, 提升在样本数据不平衡情况下的检测效果. Xu 等[11] 采用模糊合成少数类样本的方式增加 Android 恶意应用样本数目达到正负样本数量平衡的目的. 文献 [12] 通过过采样技术合成少数类样本, 并对随机森林算法进行改进, 减少数据不平衡对机器学习分类器的影响.

2 单分类 SVM 算法的基础知识

机器学习方法分为监督学习和无监督学习[13]. 监督学习主要是通过对有类别标签的数据进行学习得到检测模型, 再利用这个模型对未知数据进行分类. 无监

督学习与监督学习最大的区别就是训练样本没有标签. 另外, 根据分类的类别个数可以将机器学习分类算法分为单分类、二分类和多分类, 其中最常见的是二分类算法. 但当二分类的正负样本数量严重失衡时, 检测效果较差. 例如, 10 000 个应用程序样本中有 9900 个良性应用, 100 个恶意应用, 那么当良性应用全部检测正确时, 即使恶意应用全部检测失败, 准确率也达到 99%, 但是这个检测模型却没有实际的意义. 针对上述情况, 本文采取单分类算法检测 Android 恶意应用.

单分类算法属于异常检测, 主要分为新颖点检测 (novelty detection) 和异常值检测 (outlier detection). 新颖点检测属于半监督学习的方法, 异常值检测属于无监督学习[14]. 本文采用新颖点检测方法, 以 Android 恶意应用检测来说明单分类算法, 如图 1 所示, 将多数类的恶意应用作为一类样本, 即在圆圈里面的样本; 而良性应用作为新颖点分布在圆圈的外面, 其中, 圆圈就是决策边界[15].

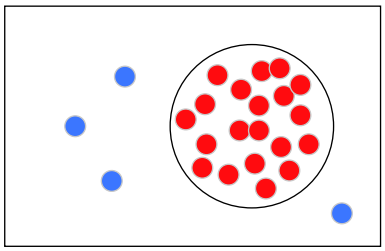

图 1 单分类算法示意图

目前, 主要的单分类算法有单分类 SVM 算法和单分类孤立森林算法. 由于本文采用新颖点检测的方法, 在训练集中不掺杂任何异常点, 所以选取单分类 SVM 算法更适用, 而孤立森林算法适用于训练集中包含异常点的检测.

单分类 SVM 算法也是基于 SVM 算法的, 将数据通过核函数映射到高维的特征空间, 利用超平面将多数数据与新颖数据隔开, 应用较多的是高斯核函数, 也称为 RBF (Radial Basis Function) 核[16]. 单分类 SVM 算法解决的同样是目标函数最优化的问题, 即式 (1). 式中ω为系数, $\|\omega\|$为ω的二阶范数, c为惩罚系数, ξ_i为松弛变量.

$$\min \frac{1}{2}\|\omega\|^2 + c\sum_{i=1}^{m}\xi_i \tag{1}$$

约束于:

$$\begin{cases} y^{(i)}\left(\omega^{\mathrm{T}}x^{(i)}+b\right) \geq 1-\xi_i \\ \xi_i \geq 0 \end{cases}$$

3 单分类 SVM 算法检测 Android 恶意应用的方法

本文收集的可用于反编译的恶意应用数量远超于良性应用, 所以选择恶意应用作为多数类样本进行训练建模, 良性应用作为新颖点进行检测. 首先通过 apktool 工具反编译恶意应用获取 AndroidManifest.xml 全局配置文件和应用程序 Smali 代码, 使用 Python 编写的程序分别从上述两个文件中通过搜索关键字的方法提取所需的特征, 基于 AndroidManifest.xml 文件搜索“uses-permission android:name="android.permission.”可以获取应用程序申请的权限、搜索“action android:name”可以获取组件的 Intent Filter 特征; 遍历反编译后得到的应用程序 Smali 文件夹, 搜索含有关键字“invoke”的语句, 获取应用程序调用的 API 特征, 将这些特征输入单分类 SVM 分类器进行训练构建检测模型, 需要注意的是恶意应用样本的 90% 作为训练集, 剩下的作为测试集; 再使用同样的方法提取良性应用的权限、Intent Filter 和 API 特征, 输入至恶意应用训练好的检测模型, 检测良性应用作为新颖点的分类结果. Android 应用程序使用单分类 SVM 检测方法如图 2 所示.

本文收集的恶意应用样本库来源于两部分, 一部分是 2013 年左右通过 AndroMalShare[17] 和 Malgenomeproject[18] 收集的 1200 个恶意应用, 另一部分是向 VirusShare 申请后下载该网站收集的 2016 年的 11 900 个恶意应用, 其中绝大部分恶意应用可以反编译成功. 良性应用也来源于两部分, 一部分是 2020 年从正规应用商城[19] 随机下载的 1500 个无法成功被反编译但可以获取 AndroidManifest.xml 配置文件的应用程序. 另另一部分是 2016 年左右从小米应用市场下载的可以被反编译的良性应用 187 个, 即可以通过对应用程序反编译获取更加细粒度的 API 特征.

本节二分类算法均使用 WEKA 工具完成, 单分类 SVM 算法利用 Python Sklearn 完成. 分类算法的评价指标主要选取 *Accuracy*、*Precision*、*Recall*、*F-measure* 和 AUC, 其中 AUC 指标是指 ROC 曲线下的面积, 面积越大说明分类器的性能越好. 另外, 下列式 (2)–式 (4) 中的 True Positive (*TP*) 即真正类, False Negative (*FN*) 即假负类, False Positive (*FP*) 即假正类, True Negative (*TN*) 即真负类[16].

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F-measure = \frac{\left(\alpha^2+1\right)Precision \times Recall}{\alpha^2(Precision+Recall)} \quad (5)$$

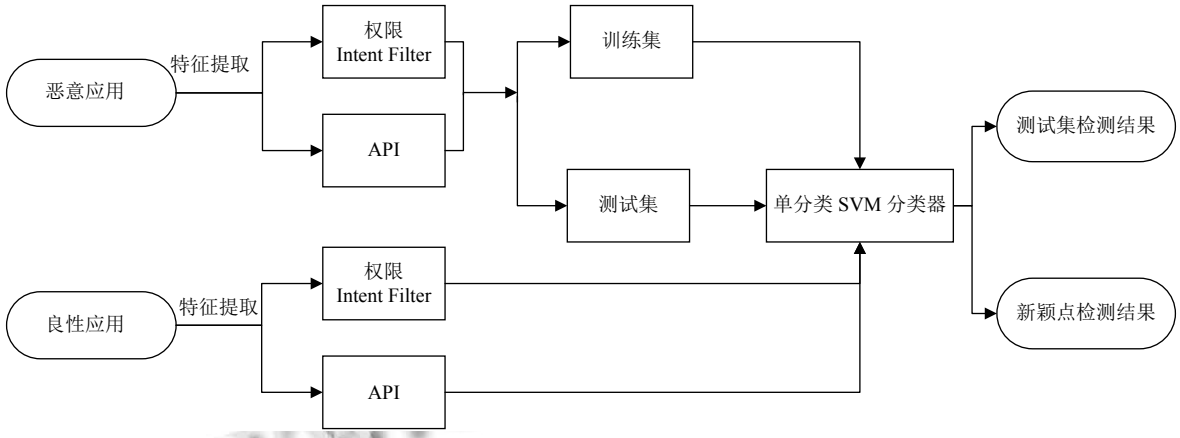

图 2 本文单分类 SVM 算法检测应用程序流程图

3.1 权限与 Intent Filter 为特征的二分类检测

无论应用程序是否进行加固, 通过反编译都是可以获取 AndroidManifest.xml 全局配置文件, 所以基于该文件提供的特征进行机器学习的研究很多[20–22]. 研究内容的重点大多是如何选择有效的特征, 减少特征维度, 提高检测性能.

本节通过信息增益法选择信息增益值靠前的 58 个权限和 Intent Filter 信息作为分类算法的特征, 采用常见的、有差异性的 SVM、NB (Naïve Beyesian)、J48 (决策树)、KNN (K-Nearest Neighbor) 作为分类器对样本进行训练、测试. 实验采用十折交叉法, 分别选取不同的实验样本和相同的特征, 删除特征提取后重复的样本, 检测结果如表 1、表 2 所示.

表 1 2013 年收集的恶意应用和良性应用分类结果

分类器	*Accuracy*	*Precision*	*Recall*	*F-measure*	AUC
SVM	0.973	0.974	0.973	0.971	0.975
NB	0.937	0.943	0.937	0.940	0.981
J48	0.943	0.943	0.943	0.947	0.942
KNN	0.967	0.968	0.968	0.968	0.983

表 2 2016 年收集的恶意应用和良性应用分类结果

分类器	*Accuracy*	*Precision*	*Recall*	*F-measure*	AUC
SVM	0.792	0.792	0.792	0.792	0.791
NB	0.759	0.765	0.759	0.759	0.809
J48	0.785	0.785	0.785	0.785	0.802
KNN	0.759	0.765	0.759	0.759	0.809

表 1 为 2013 收集的 1200 个恶意应用和从 1500 个良性应用中随机选取 1200 个作为良性样本的分类结果. 其中, SVM 分类器的准确率最高, 达到 0.973; NB 分类器性能指标相对较差, 准确率为 0.937. 表 2 为随机从 2016 年 VirusShare 恶意样本库中选取的 1500 个恶意应用和上述 1500 个良性应用作为样本的分类结果. 通过表 1、表 2 的对比, 可以看出表 2 中分类器的各项性能指标都远低于表 1, 反映了针对 2013 年收集的恶意应用检测效果较好的特征及其分类算法不适用于 2016 年收集的恶意应用检测, 这与恶意应用与良性应用在权限上差异性越来越小有直接的关系. 另外, 良性样本是 2020 年下载的, 与 2013 年良性应用申请的权限可能也存在不小的差异, 间接导致表 1 的检测结果要明显优于表 2.

3.2 权限、Intent Filter 和 API 为特征的二分类检测

本节从 2016 年 VirusShare 样本库中选取 11 900 个恶意应用和上述可反编译成功的 187 个良性应用作为样本进行研究. 实验中删除了提取特征后重复的样本避免过拟合和欠拟合的发生. 最终正负样本比为 42.7:1.

为了检测取得较好的效果, 根据 API 函数在恶意应用和良性应用中的使用频率选取有差异性的 289 个 API 函数和 3.1 节选取的权限、Intent Filter 一起作为机器学习分类器的特征, 采用十折交叉法进行检测. 实

验分为两部分, 第一部分是将全部应用程序都作为样本, 正负样本数量差异较大, 分类结果如表 3、表 4 所示. 表 3 是对应用程序的检测结果, 表 4 是针对少数类良性应用的检测结果. 实验中, SVM 分类算法完全偏向于样本集多数类 (恶意应用), 无法检测良性应用, 所以未采用该算法. NB 算法对应用程序与良性应用的分类性能指标相对接近且较好, 但检测效果仍欠佳.

表 3 应用程序样本分类结果

分类器	*Accuracy*	*Precision*	*Recall*	*F-measure*	AUC
NB	0.874	0.972	0.874	0.915	0.816
J48	0.978	0.973	0.978	0.975	0.641
KNN	0.979	0.979	0.979	0.979	0.800

表 4 良性应用样本分类结果

分类器	*Accuracy*	*Precision*	*Recall*	*F-measure*	AUC
NB	0.693	0.118	0.693	0.201	0.848
J48	0.268	0.557	0.268	0.362	0.641
KNN	0.535	0.548	0.535	0.542	0.800

第二部分实验是随机选取与良性应用等数量的恶意应用作为样本, 分类结果如表 5、表 6 所示. 表 5 中 J48 分类器的检测准确率最高, 为 0.854. 但所有分类器的性能指标相比于表 3 要差, 主要是因为表 3 的检测结果与分类算法完全偏向多数类 (恶意应用) 有关. 同时, 表 5、表 6 检测结果也说明了恶意应用和良性应用在 API 调用上的差异也在缩小.

表 5 应用程序样本分类结果

分类器	*Accuracy*	*Precision*	*Recall*	*F-measure*	AUC
SVM	0.823	0.832	0.823	0.822	0.823
NB	0.783	0.793	0.783	0.782	0.830
J48	0.854	0.854	0.854	0.854	0.833
KNN	0.823	0.824	0.823	0.823	0.862

表 6 良性应用样本分类结果

分类器	*Accuracy*	*Precision*	*Recall*	*F-measure*	AUC
SVM	0.740	0.887	0.740	0.807	0.823
NB	0.693	0.846	0.693	0.762	0.849
J48	0.850	0.857	0.850	0.854	0.833
KNN	0.795	0.842	0.795	0.818	0.862

3.3 单分类 SVM 算法检测应用程序

本节采取单分类 SVM 算法对应用程序进行检测, 将多数类的恶意应用作为正常值 (即+1), 将数量较少的良性应用作为新颖点 (−1). 将恶意应用分为训练集和测试集两部分, 当恶意应用被判断为新颖点 (−1) 时, 即为恶意应用分类错误; 同理, 当良性应用被判断为恶意应用 (+1) 时, 即为良性应用作为新颖点分类错误.

实验选取与 3.2 节相同的特征, 采用表 3 中选取的样本集, 将恶意应用的 90% 作为训练集, 剩下的 10% 作为测试集, 将全部良性应用作为新颖点. 单分类 SVM 算法的主要参数设置为 nu=0.16, kernel="rbf", gamma="auto". 检测结果如表 7 所示.

表 7 单分类 SVM 算法分类结果

数据集	*Accuracy*
恶意训练集	0.840
恶意测试集	0.856
良性应用	0.836

通过表 4、表 6 和表 7 的对比可以发现, 采用单分类 SVM 算法在正负样本严重失衡的情况下, 可以有效地检测出 83.5% 的良性应用, 而表 6 中良性应用分类指标最好的 J48 分类器的准确率为 85%, 需要注意的是单分类 SVM 算法对恶意应用训练集和测试集的检测准确率与均衡样本接近. 综上所述, 单分类 SVM 算法适用于当正负样本严重失衡情况下的应用程序检测.

4 总结

本文研究的重点是当 Android 恶意应用与良性应用因为加固等原因造成正负样本数量失衡时如何有效地进行应用程序分类, 研究内容具有现实意义. 从研究的结果可以看出, 随着恶意应用的不断升级, 与良性应用的差异越来越小, 对检测方法提出了更高的要求. 传统的基于权限等特征的二分类监督学习的检测手段无法应对这一变化. 而本文提出的使用单分类 SVM 分类算法检测 Android 应用程序的方法很好地解决了上述的问题.

在后续的研究中, 还需要重点研究 Android 应用程序特征的选取, 尽量选取恶意应用与良性应用存在差异的特征, 进一步提高应用程序分类的准确率. 另外单分类 SVM 算法的参数配置也是后期研究的重点内容.

参考文献

1 2019 年度 Android 恶意软件专题报告. http://pub-shbt.s3.360.cn/cert-public-file/2019 年度 Android 恶意软件专题报告.pdf. [2020-02-26].

2 VirusShare. https://virusShare.com. [2020-01-20]

3 Singh AK, Jaidhar CD, Kumara MAA. Experimental analysis of Android malware detection based on

combinations of permissions and API-calls. Journal of Computer Virology and Hacking Techniques, 2019, 15(3): 209–218. [doi: 10.1007/s11416-019-00332-z]

4 Shang FJ, Li YL, Deng XL, *et al*. Android malware detection method based on naive Bayes and permission correlation algorithm. Cluster Computing, 2018, 21(1): 955–966. [doi: 10.1007/s10586-017-0981-6]

5 王忠震, 黄勃, 方志军, 等. 改进 SMOTE 的不平衡数据集成分类算法. 计算机应用, 2019, 39(9): 2591–2596. [doi: 10.11772/j.issn.1001-9081.2019030531]

6 Huang X, Zhang CZ, Yuan J. Predicting extreme financial risks on imbalanced dataset: A combined kernel FCM and kernel SMOTE based SVM classifier. Computational Economics, 2020, 56(1): 187–216. [doi: 10.1007/s10614-020-09975-3]

7 Kumari C, Abulaish M, Subbarao N. Using SMOTE to deal with class-imbalance problem in bioactivity data to predict mTOR inhibitors. SN Computer Science, 2020, 1(3): 150. [doi: 10.1007/s42979-020-00156-5]

8 Zhang HP, Huang LL, Wu CQ, *et al*. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. Computer Networks, 2020, 177: 107315. [doi: 10.1016/j.comnet.2020.107315]

9 李国和, 张腾, 吴卫江, 等. 面向机器学习的训练数据集均衡化方法. 计算机工程与设计, 2019, 40(3): 812–818.

10 陈旭, 刘鹏鹤, 孙毓忠, 等. 面向不均衡医学数据集的疾病预测模型研究. 计算机学报, 2019, 42(3): 596–609.

11 Xu YP, Wu CH, Zheng KF, *et al*. Fuzzy-synthetic minority oversampling technique: Oversampling based on fuzzy set theory for android malware detection in imbalanced datasets. International Journal of Distributed Sensor Networks, 2017, 13(4): 155014771770311.

12 张家伟, 郭林明, 杨晓梅. 针对不平衡数据的过采样和随机森林改进算法. 计算机工程与应用, 2020, 56(11): 39–45. [doi: 10.3778/j.issn.1002-8331.1908-0338]

13 刘颖. 基于机器学习的遥感影像分类方法研究. 北京: 清华大学出版社, 2014.

14 李昊奇, 应娜, 郭春生, 等. 基于深度信念网络和线性单分类 SVM 的高维异常检测. 电信科学, 2018, 34(1): 2018006.

15 One Class SVM. https://scikit-learn. org/stable/modules/generated/sklearn.svm.OneClassSVM.html. [2019-12-21].

16 王洪波. 单分类支持向量机的学习方法研究 [博士学位论文]. 杭州: 浙江大学, 2012.

17 AndroMalshare. http://andromalshare.org: 8080/#overview. [2018-12-21].

18 Zhou Y, Jiang X. Dissecting Android malware: Characterization and evolution. 2012 IEEE Symposium on Security and Privacy. San Francisco, CA, USA. 2012. 95–109. [doi: 10.1109/SP.2012.16]

19 小米应用商店. http://app.mi.com/subject/167924. [2020-02-02].

20 Qiao MY, Sung AH, Liu QZ. Merging permission and API features for android malware detection. Proceedings of 2016 5th IIAI International Congress on Advanced Applied Informatics. Kumamoto, Japan. 2016. 566–571. [doi: 10.1109/IIAI-AAI.2016.237]

21 朱洪军, 陈耀光, 华保健, 等. 一种 Android 应用加固方案. 计算机应用与软件, 2016, 33(11): 297–300, 320. [doi: 10.3969/j.issn.1000-386x.2016.11.067]

22 彭守镇. Android APP 加固方案的研究. 软件工程, 2019, 22(6): 8–12. [doi: 10.19644/j.cnki.issn2096-1472.2019.06.003]