DOI: 10.11992/tis.201609016
网络出版地址: http://kns.cnki.net/kcms/detail/23.1538.TP.20170317.1937.004.html

依特征频率的安卓恶意软件异常检测的研究

张玉玲, 尹传环
（北京交通大学 计算机与信息技术学院, 北京 100044）

摘　要: Android 系统由于开源性和可移植性等优点, 成为市场占有率最高的移动操作系统。针对 Android 的各种攻击也层出不穷, 面向 Android 的恶意软件检测已成为近些年移动安全领域非常重要的一个环节。面临的问题包括恶意软件收集困难, 异常样本和正常样本比例不平衡。为了有效应对上述问题, 提出了 Droid-Saf 框架, 框架中提出了一种挖掘数据隐含特征的数据处理方案; 把样本特征包含的隐藏信息当作新的特征; 建模时将样本特征融入算法当中, 建立动态的松弛变量。应用静态分析方法反编译 apk, 用改进的 svdd 单分类器分类, 克服了恶意软件检测系统中非正常软件收集困难的不足, 降低了异常检测的漏报率和误判率。实验结果验证了该算法的有效性和适用性。

关键词: 安卓系统; 恶意软件; 数据挖掘; 异常检测; svdd; 隐含特征; 单分类器; 特征频率
中图分类号: TP391　　**文献标志码:** A　　**文章编号:** 1673−4785(2018)02−0168−06

中文引用格式: 张玉玲, 尹传环. 依特征频率的安卓恶意软件异常检测的研究[J]. 智能系统学报, 2018, 13(2): 168–173.
英文引用格式: ZHANG Yuling, YIN Chuanhuan. Android malware outlier detection based on feature frequency[J]. CAAI transactions on intelligent systems, 2018, 13(2): 168–173.

Android malware outlier detection based on feature frequency

ZHANG Yuling, YIN Chuanhuan
(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: Due to the advantages of open source and portability, Android has become a mobile OS with the largest market share. Various attacks toward Android also emerge in endlessly, the Android-oriented detection for malwares has become a quite important link recently in the field of mobile safety. The problems to be faced include difficult collection of malicious software, imbalanced proportion of the abnormal samples and normal samples. In order to effectively overcome the above difficulties, Droid-Saf framework was proposed, a data processing scheme revealing the implicit characteristics of data was proposed in the framework; the hidden information contained in the sample was treated as a new feature; in modeling, the sample features were integrated into the algorithm and dynamic slack variables were established. Static analytic method was applied to decompile apk, the improved svdd single classifier was used for classification, the deficiency of difficult collection of abnormal software in the system for detecting malicious software was overcome, the rate of missing report and the misjudgment rate of abnormal detection were lowered. The Experimental results verified the effectiveness and applicability of the algorithm.

Keywords: Android system; malware; data mining; abnormal detection; svdd; implicit characteristics; single classifier; feature frequency

美国信息技术研究和顾问公司 Gartner 于 2016 年 2 月公布了 2015 年全球智能手机销售量[1]。2015 年全球智能手机销量达 14 亿部, 较 2014 年增长了 14.4%。Gartner 预计 2016 年全球手机出货量将达到 19.59 亿部, 高于 2015 年的 19.10 亿部, 而 2017 年将达到 19.83 亿部。德国网络安全公司 GDATA 最新公布的调查报告显示, 2015 年底, Android 恶意软件文件数量多达 230 万。Android 恶意软件数量众多, 已成为移动安全的重灾区[2]。移动

收稿日期: 2016−09−14.　**网络出版日期:** 2017−03−17.
基金项目: 国家自然科学基金项目 (61105056).
通信作者: 尹传环. E-mail: chhyin@bjtu.edu.cn.

应用安全事件频发, 主要有以下问题: 简单的应用市场安全审查、粗粒度的权限系统、有限的系统级安全监测, 以及鼓励快速传播的应用分派模式[3]。传统分类方案虽然实现了恶意软件的检测, 但其安全性、实时性、性能等方面均存在不足, 特别是在移动端发展迅速的情况下, 其检测机制难以满足恶意软件检测需求, 不足以应对不断变化的未知恶意软件和变异问题。Mudflow[4]是目前少有的有关安卓恶意异常检测的研究, 利用单分类支持向量机(One-class SVM)[5]分类, 检测正常软件和恶意软件的准确率只有 83.8%, 该方法利用静态污点分析技术, 找出正常软件和恶意软件之间的数据流差别来标记可疑特征, 模型的建立并没有脱离恶意软件。

针对上述问题, 本文提出 Droid-Saf 框架, 利用轻量级的静态分析, 训练集仅包含正常软件的样本, 基于 svdd 算法建立模型, 并对该算法做了改进, 融入了隐含特征, 以便更好地应对软件的不断变化。检测恶意软件的误报率从 10% 降低到 0.6%, 并且新算法对参数不敏感, 更容易找到比较理想的结果。

1　静态分析与动态分析

检测恶意软件的方法, 大致可分为两种: 静态分析和动态分析。静态分析不运行程序只分析代码, 试图找到表明应用程序的行为的控制流, 也被用来分析正常软件的代码缺陷和故障, 甚至在不运行应用程序的情况下直接检测其恶意行为。静态检测优点是速度快, 缺点是需要构建并且动态维护恶意行为的特征库。动态分析不分析代码而在专有的虚拟环境中运行代码, 用自动化工具追踪应用程序的行为, 然后再分析它的日志文件或者系统调用序列等。其缺点是需要进行实时监控, 内存和电量的消耗高。

近年来, 众多研究者提出了很多方案。静态分析方面, 例如 PiggyDroid[6]使用 API 调用等静态特征来做检测; RiskRanker[7]检测软件是否利用 root 漏洞和是否发送后台信息等筛选出需要深入研究的软件, 但都是靠专家系统检测。DroidAPIMiner[8]把数据流相关的 API 作为特征, 然后用改进的 KNN 算法进行分类, 还有研究者提出了一些有效的方法与框架, 例如 Droid-Sec[9]、Shina[10]、Drebin[11]。动态分析方面, 例如 Crowdroid 框架[12]由客户端和服务器端组成, 客户端使用 Linux 系统的 Strace 机制监控 Android 系统调用, 然后把调用信息发送到服务器端处理; CopperDroid[13]不关注底层动作的调用, 而是捕捉 java 代码和本地执行代码发起的行为。DroidAnalyst[14]提出一个新的自动化应用, 审查和分析恶意软件的框架, 集成了静态和动态分析的协同作用来提高分析的精度和效率。

上述方法都是利用现有的正常软件和恶意软件进行训练和分类, 随着恶意软件的不断更新变化, 应对未知恶意行为又是一道难题, 应用异常检测是目前比较流行的方法。

2　异常检测

异常检测是指将正常的习惯行为特征存储在数据库中, 然后将当前的行为特征与特征数据库中的特征进行比较, 如果两者偏差足够大, 则说明发生了异常[15]。目前异常检测的研究主要基于无监督学习框架[16], 因此本文基于单分类算法进行研究。现有方法对恶意软件检测困难主要有两个原因: 1) 恶意软件经常出现在检测方法之前。传统的检测依赖已知样本进行训练, 不能自动更新规则库, 也无法检测新的恶意行为; 2) 恶意软件数据收集不完整。现实检测任务中异常数据普遍难以采集或者采集代价过高, 往往只有一类数据可供使用, 因此对于没有收集到的恶意软件无法辨别。针对恶意软件不断变化且数据收集不完整的问题, 异常检测是比较可行的方法, 本文基于 SVDD 单分类算法进行研究。

3　SVDD 的基本原理及改进方案

支持向量数据描述, 英文名 (support vector data description, SVDD)[17], 通俗来讲, 它是一种单分类模型, 假设正常数据服从球形分布, 利用核函数把样本空间映射到高维核空间, 在核空间找到一个能包含所有样本的超球体, 寻求一个最小包含球以包含正常样本, 该方法不需假设原点为异常点, 并且该方法以极小极大化方法求解最小包含球 (球心、半径)。当识别一个新的样本时, 如果样本在球体内部, 就认为是正常的, 否则就是异常的[18]。

SVDD 首先通过核函数将输入空间映射到一个高维空间, 在这个高维空间构造一个包含所有训练样本点的球体; 在球面上的样本点即为 SVDD 所求得的支持向量。假设模型 $f(x; w)$ 表示一类紧密的有界数据集, SVDD 的优化目标就是求一个中心为 a 半径为 R 的最小球面, 而且使训练集 X 的所有样本都落在此球体内。它的原理和 SVM 很像, 类比于 SVM[19], 定义一个最小化问题:

$$F(R,\boldsymbol{a},\xi) = R^2 + C\sum_i \xi_i \tag{1}$$

使得这个球面满足:

$$\begin{aligned}&(\boldsymbol{x}_i-\boldsymbol{a})^2\leqslant R^2+\xi_i\\&\text{s.t.}\quad \xi_i\geqslant 0,\quad \forall i\end{aligned}\tag{2}$$

式中：这里的 ξ_i 是松弛变量；C 是调节松弛变量的影响大小。利用 Lagrange 函数求解上述约束下的最小优化问题，得到：

$$\begin{aligned}&L=\sum_i\alpha_i(\boldsymbol{x}_i\cdot\boldsymbol{x}_i)+\sum_{i,j}\alpha_i\alpha_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)\\&\text{s.t.}\quad \sum_i\alpha_i=1,\quad 0\leqslant\alpha_i\leqslant C\end{aligned}\tag{3}$$

对上述问题相对 $\boldsymbol{\alpha}$ 求最大，可以用标准的二次规划算法来解决。这样就可以求得 $\boldsymbol{\alpha}$ 的最优值，对于非 0 的 α_i，其对应的样本点是支持向量，位于球面上；而 α_i=0 则表示对应的样本点位于球体内。$\boldsymbol{\alpha}$ 和 R 可用含 $\boldsymbol{\alpha}$ 的表达式隐式地表示。判断一个数据点属于这个类，那么当满足：

$$(z-\boldsymbol{a})^{\mathrm{T}}(z-\boldsymbol{a})\leqslant R^2\tag{4}$$

即判 Z 属于正常类，否则为异常类。将超球面的中心用支持向量来表示，那么判定新数据是否属于这个类的判定条件就是：

$$(z\cdot z)-2\sum_i\alpha_i(z\cdot\boldsymbol{x}_i)+\sum_{i,j}\alpha_i\alpha_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)\leqslant R^2\tag{5}$$

式中：R 是任意一个支持向量 $\boldsymbol{X}_k$ 到球心 a 的距离。当输入空间的样本点不满足球状分布时，可以通过核技巧把输入空间先映射到高维空间，然后在映射后的高维空间内求解。如果使用核函数上述判定条件改写为

$$K(z\cdot z)-2\sum_i\alpha_iK(z\cdot\boldsymbol{x}_i)+\sum_{i,j}\alpha_i\alpha_jK(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)\leqslant R^2\tag{6}$$

常用的核函数有线性核函数、多项式核函数、RBF 核函数等，不同的核函数，对实验结果影响很大，用合适的核函数分类才能得到理想的实验结果。传统的静态分析关注样本申请的权限和 API 调用等特征，受系统调用序列关注系统调用频率[20]的启发，上述特征的总和可以体现出恶意软件的行为活跃程度。在正常样本中，申请次数多的样本代表它本身活跃，行为复杂。正常软件中频繁申请权限或者 API 的样本应该更加得到关注，所以可以从这方面入手体现不同软件的共有特性。将频率融入到 svdd 单分类算法当中，以寻找最适应于本方案的超球体，尽可能地将这些样本包含在 SVDD 的超球体内，达到静态分析特征进而降低单分类检测的误警率的目的。

将样本频率融入到算法当中：定义正常样本活动频率矩阵 $\boldsymbol{c}$=[c_1 c_2 c_3 … c_n]，n 为训练样本个数。

$$\begin{aligned}&c_i=\frac{\sum_{j=1}^{m}P_{ij}}{\max(\sum_{j=1}^{m}P_{ij})}\\&\text{s.t.}\quad i=1,2,\cdots,n\end{aligned}\tag{7}$$

式中：n 是样本个数，m 是特征维数，所有样本的特征用一个 $m\times n$ 的矩阵 $\boldsymbol{P}$ 表示；分子指的是第 i 个样本的频率 (即样本的非 0 特征个数，分母指的是全部样本的最大频率)。那么函数的最小化问题变为

$$\begin{aligned}&\min\quad F(R,\boldsymbol{a},\xi)=R^2+\sum_{i=1}^{n}c_i\xi_i\\&\text{s.t.}\quad \|\boldsymbol{x}_i-\boldsymbol{a}\|^2\leqslant R^2+\xi_i\\&\quad\xi_i\geqslant 0,i=1,2,\cdots,n\end{aligned}\tag{8}$$

构造新的 Lagrange 函数如式 (9)：

$$\begin{aligned}L(\boldsymbol{a},R,\xi,\alpha,\gamma)=R^2+\sum_{i=1}^{n}c_i\xi_i-\sum_{i=1}^{n}a_i[R^2+\xi_i-\\(\boldsymbol{x}_i\cdot\boldsymbol{x}_i)+2(\boldsymbol{a}\cdot\boldsymbol{x}_i)-(\boldsymbol{a}\cdot\boldsymbol{a})]-\sum_{i=1}^{n}\gamma_i\xi_i\end{aligned}\tag{9}$$

由此，通过上式调整对应的约束条件：

$$\begin{aligned}&\sum_i\alpha_i=1\\&0\leqslant\alpha_i\leqslant c_i\\&\text{s.t.}\quad i=1,2,\cdots,n\end{aligned}\tag{10}$$

本章主要讲述了将正常样本作为数据源，用轻量级静态分析方法提取特征，并在特征库的基础上提出了一种挖掘隐形特征的方法，将特征频率作为新的特征融入到 SVDD 算法当中，实现降低检测恶意软件的误警率的目标。

4　实验部分

4.1　实验流程

实验的流程框架如图 1 所示。

其中，数据处理用到了 FexDroid[21]的降维方法。

4.2　数据提取

把从谷歌商店、安智、应用宝等安卓软件市场收集的软件作为正常样本，一共有 1 976 个；从 drebin 中随机选取了 1 952 个恶意软件作为负样本；然后用 apktool 等工具反编译这些 APP，运行得到 manifest 和 smali 文件，再提取 permission、api_call、activity 等特征，共得到 8 874 个特征；再将每个样本的特征长度 (也就是样本的行为频率) 作为新的特征融入 svdd 算法当中。

4.3　评价指标

评价指标是漏报率和误报率，为准确描述这两个概念，避免不必要的误会，给出以下计算公式。

我们引入 4 个参数 3 个评价标准，如表 1 所示，首先如下假设：

FP：将样本判定为正常样本，实际为恶意样本的数量。

TN：将样本判定为恶意样本，实际为恶意样本的数量。

TP：将样本判定为正常样本，实际为正常样本的数量。

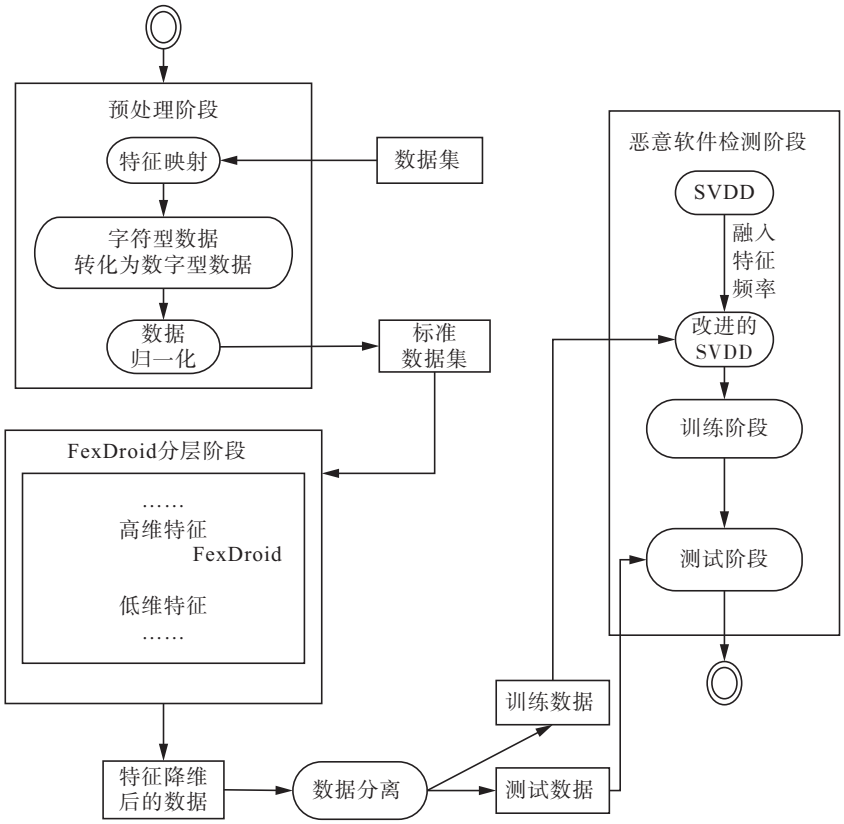

图 1　异常检测流程框架

Fig. 1　The framework of anomaly detection process

表 1　实验评价指标

Table 1　The evaluating indicator of experiment

评价指标	定义	含义
准确率	$\frac{TN}{TN+FP}$	恶意软件被正确分类
漏报率	$\frac{FP}{TN+FP}$	恶意软件被错误分类
误报率	$\frac{FN}{FN+TP}$	正常软件被检测为恶意软件

FN: 将样本判定为恶意样本, 实际为正常样本的数量。

4.4　实验步骤

随机取 1 000 个正样本作为训练集, 其他的正样本和负样本作为测试集, 频率是每个样本包含特征的个数, 把它归一化然后做新特征。分别用最常用的多项式核函数和 RBF 核函数模型。其中, c 是 svdd 的惩罚参数, b 是核函数参数; 改进算法中频率特征会取代参数 c 的作用。为方便描述, 模型改进之前的实验称为 Driod-svd, 改进之后的实验称为 Droid-Saf。两者对比表明改进算法在 RBF 核函数上取得了明显效果, 如图 2 所示。

如果样本的特征数非常多, 使用 RBF 核将样本映射到高维空间, 结果较差, 使用线性核比较合适高维数据。因此下面的实验使用常见的多项式核函数。

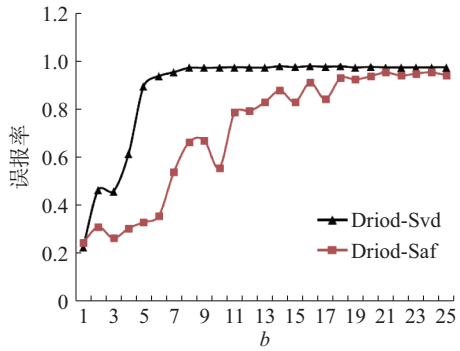

图 2　RBF 核函数的实验对比

Fig. 2　The experimental comparison of RBF kernel

Droid-Saf 对恶意软件检测的正确率基本达到 100%, 参数 c 对结果并没有影响, 因此对参数不敏感; 但是 Droid-Svd 的部分漏报率大于 0, c 越大, 误判率越高, 漏报率也越高。为了详细说明细节, 表 2 列出了 Droid-Svd 的实验情况。

基于多项式核函数的实验结果分别取了 Droid-Svd 和 Droid-Saf 在多项式核函数下最好的实验结果, 如图 3 所示。

当 c=0.05 时, 参数 b 对基于多项式核函数的 Droid-Saf 并没有作用, 漏报率基本为 0, 误报率明显低很多。

表 2　基于多项式核函数的 Droid-Svd 实验的平均值

Table 2　The experimental FPR comparison of Polynomial kernel function

c	平均误报率	平均漏报率
0.05	0.110	0.008
0.15	0.265	0.036
0.10	0.181	0.016
0.20	0.298	0.025
0.30	0.389	0.045
0.40	0.457	0.072
0.50	0.550	0.200
0.60	0.676	0.419
0.70	0.665	0.365
0.80	0.738	0.534
0.90	0.898	0.831

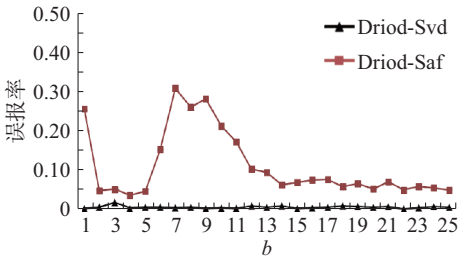

图 3　多项式核函数的实验对比

Fig. 3　The experimental comparison of Polynomial kernel function

4.5　实验总结

为了直观对比实验的改进效果，表 3 给出了 Droid-Svd 和 Droid-Saf 在核函数分别为多项式和 RBF 的最好结果。多项式核函数的实验结果在图 2 中，误报率在 c=0.05、b(1-25) 不同时总体上差别不大，b 对应 25 个实验数据，对这 25 个实验数据先求和再求平均数；RBF 核函数的实验结果在图 3 中，结果在 b(1-25) 不同时波动较大，因此取了两条曲线上结果最好的数据。

表 3　实验总结的平均结果

Table 3　The summarized average results of the experiments

核函数类型	实验方案	漏报率/%	误报率/%
RBF 核函数	Droid-Svd	5	0
	Droid-Saf	2.3	0
多项式核函数	Droid-Svd	11	0.8
	Droid-Saf	0.5	0

表 3 证明了改进方法的有效性和适用性，改进后的算法舍弃了对实验结果影响敏感的参数 c，调参简单。

5　结束语

传统分类方案虽然实现了恶意软件的检测，但是其安全性、实时性、性能等方面均存在不足，特别是在移动端发展迅速的情况下，其检测机制难以满足恶意软件检测需求。本文所提出的框架对以上问题给出 解决方案，具有应对未知恶意软件和恶意软件变异的能力。针对静态分析中的特征包含的隐藏信息，提出一种挖掘数据隐含特征的数据处理方案，并且将它融入到单分类算法中，进而改进了单分类算法模型，降低了恶意软件检测的误报率。其中，多项式核函数的漏报率从 11% 降低到了 0.5%；RBF 核函数的实验也有相应的提高。但是因为静态分析固有的特性，应对代码混淆需要更好的反编译技术；另外正常样本往往会不可避免地混入恶意行为而导致模型偏离现象；这些问题也是以后要研究的方向。

参考文献：

[1] 微头条. Gartner: 2016 全球手机出货预计 19.59 亿部[EB/OL]. http://www.wtoutiao.com/p/19cnOtt.html.

[2] 中文业界资讯站. 2015 年 Android 恶意软件样本数量超 230 万[EB/OL]. [2017-05-13]. http://www.cnbeta.com/articles/478843.html.

[3] 杨威, 肖旭生, 李邓锋, 等. 移动应用安全解析学: 成果与挑战[J]. 信息安全学报, 2016, 1(2): 1–14.
YANG Wei, XIAO Xusheng, LI Dengfeng, et al. Security analytics for mobile apps: achievements and challenges[J]. Journal of cyber security, 2016, 1(2): 1–14.

[4] AVDIIENKO V, KUZNETSOV K, GORLA A, et al. Mining apps for abnormal usage of sensitive data[C]//Proceedings of 37th IEEE International Conference on Software Engineering. Florence, Italy, 2015: 426–436.

[5] JUSZCZAK P. Learning to recognise: a study on one-class classification and active learning[D]. TU Delft, the Netherlands: Delft University of Technology, 2006.

[6] ZHOU W, ZHOU Y, GRACE M, et al. Fast, scalable detection of piggybacked mobile applications[C]//Proceedings of the third ACM conference on Data and application security and privacy. [s.l.], ACM, 2013: 185–196.

[7] TAX D M J, DUIN R P W. Support vector data description [J]. Machine learning, 2004, 54(1): 45–66.

[8] ZHOU Wu, ZHOU Yajin, GRACE M, et al. Fast, scalable detection of “piggybacked” mobile applications[C]//Proceedings of the Third ACM Conference on Data and Application Security and Privacy. San Antonio, Texas, USA, 2013: 185–196.

[9] GRACE M, ZHOU Yajin, ZHANG Qiang, et al. Riskranker: scalable and accurate zero-day Android malware detection[C]//Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MO-

BISYS). Lake District, UK, 2012: 281–294.

[10] WU Songyang, WANG Pan, LI Xun, et al. Effective detection of android malware based on the usage of data flow APIs and machine learning[J]. Information and software technology, 2016, 75: 17–25.

[11] YUAN Zhenlong, LU Yongqiang, WANG Zhaoguo, et al. Droid-Sec: deep learning in android malware detection[C]// Proceedings of the 2014 ACM Conference on SIGCOMM. Chicago, Illinois, USA, 2014: 371–372.

[12] SHEEN S, ANITHA R, NATARAJAN V. Android based malware detection using a multifeature collaborative decision fusion approach[J]. Neurocomputing, 2015, 151: 905–912.

[13] TAM K, KHAN S J, FATTORI A, et al. CopperDroid: automatic reconstruction of android malware behaviors [OL/EB]/. [2016-03-24]. https://www.researchgate.net/publication/300925104.

[14] BURGUERA L, ZURUTUZA U, NADJM-TEHRANI S. Crowdroid: behavior-based malware detection system for android[C]//Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices. Chicago, Illinois, USA, 2011: 15–26.

[15] TAM K, KHAN S J, FATTORI A, et al. CopperDroid: Automatic Reconstruction of Android Malware Behaviors[C]// Proceedings of Annual Network and Distributed System Security (NDSS). San Diego, United States, 2015.

[16] FARUKI P, BHANDARI S, LAXMI V, et al. DroidAnalyst: synergic app framework for static and dynamic app analysis[M]//ABIELMONA R, FALCON R, ZINCIR-HEYWOOD N, et al. Recent Advances in Computational Intelligence in Defense and Security. Cham: Springer, 2016: 519–552.

[17] TAX M J D, DUIN ROBERT P W. Support vector domain description[J]. Pattern recognition letters, 1999, 20(11/12/13): 1191–1199.

[18] HASTIE T, TIBSHIRANI R, FRIEDMAN J. Unsupervised learning[M]//HASTIE T, TIBSHIRANI R, FRIEDMAN J. The Elements of Statistical Learning. New York, USA: Springer, 2009: 485–585.

[19] CRISTIANINI N, SHAWE-TAYLOR J. 支持向量机导论[M]. 李国正,译. 北京: 电子工业出版社, 2004: 57–61.
CRISTIANINI N, SHAWE-TAYLOR J. An introduction to support vector machines and other kernel-based learning methods[M]. LI Guozheng, Trans. Beijing: Publishing House of Electronics Industry, 2004: 57–61.

[20] 罗隽, 丁力, 潘志松, 等. 异常检测中频率敏感的单分类算法研究[J]. 计算机研究与发展, 2007, 44(Z2): 235–239.
LUO Jun, DING Li, PAN Zhisong, et al. Research on sequence-call-frequency-based one-class algorithm in abnormal detection[J]. Journal of computer research and development, 2007, 44(Z2): 235–239.

[21] 张玉玲, 尹传环. 基于 SVM 的安卓恶意软件检测[J]. 山东大学学报: 工学版, 2017, 47(1): 42–47.
ZHANG Yuling, YIN Chuanhuan. Android malware detection based on SVM[J]. Journal of Shandong university: engineering science, 2017, 47(1): 42–47.

作者简介:

张玉玲, 女, 1990 年生, 硕士研究生, 主要研究方向为机器学习。

尹传环, 男, 1976 年生, 副教授, 主要研究方向为网络安全 (入侵检测)、数据挖掘、机器学习 (支持向量机)。