基于深度学习的安卓恶意应用检测

王亚洲，王 斌

（中国航天科工集团第二研究院 北京计算机技术及应用研究所，北京 100854）

摘 要：针对传统的基于特征码的恶意应用检测技术，在应对新的恶意应用产生情况下处理速度上的不足，提出一种基于深度学习的安卓恶意应用检测方法。通过对包含应用静态信息的文件进行反编译处理，提取可表征应用是否为恶意应用的信息，经过数据预处理后生成特征信息输入矩阵，采用多层卷积神经网络进行训练，优化得到较优的参数。实验结果表明，所提方法能有效检测出恶意应用。

关键词：安卓恶意应用；静态检测；深度学习；卷积神经网络；反编译

中图法分类号：TP309.5 **文献标识号**：A **文章编号**：1000-7024（2020）10-2752-06

doi：10.16208/j.issn1000-7024.2020.10.010

Android malicious application detection based on deep learning

WANG Ya-zhou, WANG Bin

(Beijing Institute of Computer Technology and Applications, Second Academy of China Aerospace Science and Industry Corporation, Beijing 100854, China)

Abstract: Aiming at the traditional signature application-based malicious application detection technology and dealing with the lack of processing speed in the case of new malicious applications, a deep learning-based Android malicious application detection method was proposed. Through decompilation processing of files containing application static information, information that could be used to represent whether the application was a malicious application was extracted from it. After data preprocessing, the feature information input matrix was generated, and the multi-layer convolutional neural network was used for training and optimized to get better parameters. Experimental verification shows that the proposed method can effectively detect malicious applications.

Key words: Android malicious application; static detection; deep learning; convolutional neural network; decompilation

0 引 言

安卓诞生的数年来凭借其平台的开放性获得了巨大的市场渗透率，同时针对安卓的各类恶意应用数量急剧增加。从卡巴斯基首次发现短信特洛伊木马程序以来，安卓恶意软件不断发展，通过增加代码混淆、隐蔽的命令和控制通信通道等方式来加强躲避检测的能力[1]。而无论是广泛使用的签名和特征码技术、对于应用内容进行分析的静态检测技术，还是沙箱动态检测技术都有其弊端。

最早出现基于签名和特征码技术，优点是检测速度快、误报率低，仍被当前安全厂商采用，但是无法检测短时间爆发的未知恶意应用，并且随着恶意软件数量的不断增长，很难维持庞大的特征库[2]。并且传统的基于静态检测方法检测准确率不高，而动态检测方法需要占用大量的检测时间，越来越难以适应恶意应用数量快速增长的情况。从以上得出恶意应用的检测需要更精准的信息提取能力和更好的分类算法。

本文在分析了典型恶意应用检测方法的基础上，针对以上方法的问题，提出一种基于深度学习的恶意应用检测方法。首先构建了一个包含较大样本量的数据集，将 Android 应用处理后提取出关键信息，采用适用于文本分类的卷积神经网络，应用深度学习算法相比传统机器学习算法有更强的学习特征的能力，提高了检测效果。

1 恶意应用分析

对于安卓恶意应用检测，最直接接触到的原始分析素

收稿日期：2020-03-26；**修订日期**：2020-06-01

作者简介：王亚洲（1994-），男，河南周口人，硕士研究生，研究方向为信息安全、人工智能；王斌（1981-），男，山西运城人，博士，研究员，硕士生导师，研究方向为可信计算、网络安全。

E-mail: franklinwyz@163.com

材便是 Android 应用程序包（Android application package，APK），而对安装包中文件的静态分析对本文提出的方法尤其重要。本节将针对安卓应用进行解析，研究典型恶意应用检测方法。

1.1　安卓应用安装包解析

安卓应用安装采用的是后缀名为 APK 的安装文件，是一个包含各种数据的压缩包。

如图 1 所示，APK 文件中包含工程属性文件、代码文件、应用程序资源文件和第三方库。其中对本文的方法比较重要的是 AndroidManifest. xml 文件和 classes. dex 文件。AndroidManifest. xml 包含用来描述应用的各个组件，包括构成应用的 Activity、服务、广播接收器、ContentProvider、声明应用必须具备的权限等，其中权限用来规定应用可以访问 API 中受保护的部分，可以作为判定安卓应用是否恶意的一个重要特征。classes. dex（dex：dalvik executable）文件是可以直接在 Dalvik VM 虚拟机上执行的文件执行格式，该文件含有 Java 源代码编译后的各个类，但是人工很难阅读，因此需要经过反编译处理后才能提取其中有用的信息。Dalvik 是 Google 公司在 Android4. 4 版本之前专门为 Android 操作系统设计的一个基于寄存器的虚拟机，在此之后 Google 推出 ART 作为新的虚拟机，但是运行的仍是 dex 格式的代码，因此对 dex 的分析不受影响。

典型应用安装包组成

asset	资源目录1：音频和视频资源
lib	so库存放位置，一般由NDK编译得到，常见于使用游戏引擎或JNI native调用的工程中
META-INF	存放工程一些属性文件，例如Manifest.MF
res	资源目录2：Java开发的Android工程使用到的除视频，音频以外的资源文件
AndroidManifest.xml	Android工程的基础配置属性文件
classes.dex	Java代码编译得到的Dalvik VM能直接执行的文件
resources.arsc	对res目录下的资源的一个索引文件，保存原工程中strings.xml等文件内容

图 1　典型应用安装包组成

1.2　典型恶意应用检测方法

现有的安卓恶意代码检测技术主要为基于静态检测技术和基于沙盒的动态行为检测技术，或者同时使用两者的混合检测技术。

一般的静态分析内容有签名信息分析、代码语义分析、控制流分析、数据流分析等。典型的静态检测方法在没有运行应用或者安装应用的情况下，通过对应用程序中的代码内容采用源代码的静态分析工具进行检测分析，然后汇总分析代码在执行过程中对资源的使用。秦中元等[3]提出一种多级签名匹配算法，以 MD5 哈希算法与反编译生成的 smali 文件为基础，生成 API 签名、Class 签名、Method 签名、APK 签名和已知恶意应用样本库进行对比。王兆国等[4]提出来一种抗混淆的 Android 应用检测方法，通过提取不同应用的某些特定文件内容特征，与作为对比标准的正版应用、已知恶意应用进行对比来识别恶意应用。王志强等[5]提出用控制流序列即调用某个类方法的序列和有限长度的 API 函数调用序列在一定程度上表征安卓应用程序的行为。

动态行为检测方法运用虚拟机环境（也称沙箱），将未知待检测的代码运行其中，通过检测相应软件是否产生恶意行为作为判断依据来判断是否是恶意代码[6]。同时代码加密和混淆不会轻易影响基于行为的检测方法，但是需要收集如函数调用、运行行为、文件使用等应用产生的大量相关动态运行信息。该类方法与静态检测方法相比，对未知的恶意应用检测效果方面优于基于特征代码的检测方式，可以减少漏报，但是速度较慢，其计算资源和时间是很难承受的，并且有些应用检测到沙箱运行后会针对性地改变其行为，躲避检测。

近年有研究人员采用收集应用 API 序列、调用函数序列，或者操作码应用朴素贝叶斯[7]、随机森林等机器学习算法进行检测，对未知恶意应用的检测取得了一定的效果。蒋晨等将恶意软件转化为灰度图像，通过深度卷积神经网络来提取图像纹理特征，从而检测恶意软件[8]。由静态分析方法速度较快，资源占用少的特点和卷积神经网络提取特征的能力，本文提出运用应用静态信息通过卷积神经网络的训练来识别其恶意行为逻辑，进而实现恶意应用的检测。

2　应用关键信息选取及卷积神经网络

由第 1 节可知对于安卓恶意应用，APK 文件中包含可表征安卓应用恶意行为的信息，并可利用深度学习技术自动提取特征的能力检测恶意应用。下面将阐述所提方法需提取的静态信息及深度学习中卷积神经网络的基本结构。

2.1　关键特征信息选取

APK 文件中有很多可以用于恶意应用静态检测方法的特征信息，但是直接将不经处理的应用输入模型会给神经网络输入过多冗余信息，造成模型难以训练或者拟合效果差。因此需要选择适当的特征信息输入网络以实现较好恶意应用检测效果：

（1）权限类：Android 系统有一个权限控制系统，用户可在安装或者使用的过程中对某应用的权限进行合理控制，限定控制权限在用户手中，并且可在安装过程中提示用户从而开启应用功能所需的权限，在一定程度上可保护用户的隐私数据并保护系统安全。然而恶意应用会采用各种隐瞒策略诱骗用户打开超出程序功能所需的权限，通过统计分析发现，大量的恶意应用会使用敏感权限如短信相关权限、监控电话是否进行的权限和安装应用的权限[9]。大量

恶意应用会通过权限漏洞进行恶意行为：安装其它非授权应用、恶意消耗资费、偷取用户通讯录、相册文件、监控用户电话活动等等；

（2）API 类：现今很多正常应用也会存在过度授权现象，使得仅通过其权限的使用来判断恶意应用缺乏充分性。需要从另一个角度看应用的行为，Android 系统通过系统框架层提供了供应用使用的各种 API 去调取系统的各种资源，隐私数据也包括在其中。因此可以通过应用 API 的使用来表征应用的行为，即使恶意应用绕开了权限管理控制。但是传统的 API 特征只观察系统是否调用某个 API，用 0/1 标志出现与否得到特征向量，这样无法得出应用的整体行为。根据这个思路，本文考虑提取安卓应用 API 的使用顺序 API 以及它们之间的关联得到一种重要特征，用来识别应用的敏感操作；

（3）ACTION 类：Android 系统运行过程中会产生一些事件，应用如需获取这类消息，则需要在应用的 AndroidManifest. xml 文件中或应用代码中注册 ACTION 组件，如此应用便可收到其它应用产生的事件或者系统状态改变的事件从而监控它们的运行状态。而恶意应用常常为了实现某些隐藏功能而注册一些 ACTION 组件，比如检测到电池状态低的时候挂起程序以免引起怀疑的情形。

以往用到权限特征的方法只是简单地提取了权限特征，利用机器学习方法来检测应用程序[10]。本方法综合采用权限信息、API 类和 ACTION 类，可以一定程度避免输入冗余信息，降低模型运算量，同时不丢失区分恶意应用和正常应用的重要参考信息。

2.2 深度卷积神经网络

本文提出的方法将应用静态信息表示为文本形式，输入深度卷积网络模型即可从特征信息序列中学到指示恶意软件的功能[11]，从而使文本分类神经网络适于本方法。深度学习模型一旦经过训练，就可以有效地在 GPU 上执行，迅速的大量扫描检测应用。

深度学习算法的最大优势在于其逐层从数据中学习特征的特点，网络越深，学到的特征越抽象，即可以自动学到表征恶意应用的特征，这消除了大量的领域专业知识和手工特征提取的需要。神经网络模型通常采用反向传播（back propagation，BP）进行训练。BP 算法是“误差反向传播”的简称，为一种使用梯度下降法监督学习人工神经网络的方法。对于给定的人工神经网络和一个误差函数，该算法根据输出与标签计算出误差函数的值，之后计算误差相对于神经网络中每个权重的梯度，根据该梯度更新各个权重的值。通过算法可快速更新权重，即更新神经网络，经过大量数据训练后生成可快速检测恶意应用的模型。同时模型还具有可更新性，通过不断“投喂”新的恶意应用可实现对新型恶意应用变种的检测能力。

如图 2 所示，为本文所用的卷积神经网络结构。

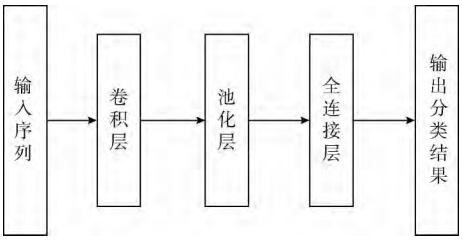

图 2 卷积神经网络组成

（1）输入层：模型的输入层由每个应用提取的特征信息的词向量组成的矩阵组成，输入矩阵大小为 N * K，其中 K 为词向量的长度，N 为序列的长度。

（2）卷积层：在输入层的基础上，使用卷积核进行卷积操作得到特征图。实验中使用的 3 种大小的卷积核，分别是 3 * K，4 * K，5 * K，K 表示词向量的长度。其中每种类型大小的卷积核有 100 个含有不同值的 Filter。每一个过滤器都能从输入的矩阵中抽取出一个特征图，在自然语言处理中称为文本特征。

（3）池化层：对特征进行进一步提取，实验中对特征图的池化操作方式是取最大值池化的方式，即将每个特征图向量中最大的一个值抽取出来，组成一个一维向量。

（4）全连接层：该层的输入为池化操作后形成的一维向量，经过激活函数输出，再加上 Dropout 层防止过拟合。最后，将所得向量 f 传递到多层感知器（MLP，包括完全连接的隐藏层和全连接的输出层）。带有隐藏层的 MLP 提取的特征之间的高阶关系可以被检测从而用于分类。

（5）输出分类结果：softmax 层会使正确的分类获得更大的概率，使错误的分类得到更小的概率，输出当前样本二分类结果。然后和样本标签做对比，通过 BP 算法进行误差反向传播调整神经网络的参数。

卷积神经网络不仅在图像领域有优异的效果，在文本分类上表现同样优异。对于文本卷积网络，和图像中的处理稍不同的是卷积核通常是对图像的一小块区域进行计算，卷积核的宽度不与输入矩阵相同，而在输入矩阵中每一行代表一个词的词向量，构成一句话的词向量作为输入。因此在处理文本时，每个卷积核就会覆盖连续的几个词，此时卷积核的宽度需与矩阵的词向量长度相同。通过这样的方式，卷积神经网络便可捕捉连续出现的多个词之间的特征，然后利用池化层减小表示空间的大小，以减少网络中的参数和计算量，最终通过全连接层实现对 APK 的分类。

3 基于深度学习的安卓恶意应用检测系统架构设计

根据上述思路，本文设计了安卓恶意应用检测系统框架，实现提出的深度卷积神经网络的安卓恶意应用检测方法，可验证提出的检测方法的有效性。

如图 3 所示，数据集获取模块获取检测系统所需的

应用素材，安卓应用特征提取及预处理模块对应用进行提取信息和加工，深度卷积神经网络训练模块将加工完成的序列信息输入网络进行训练，输出优化后的模型，最后采用生成的模型进行恶意应用检测验证效果。主要模块如下。

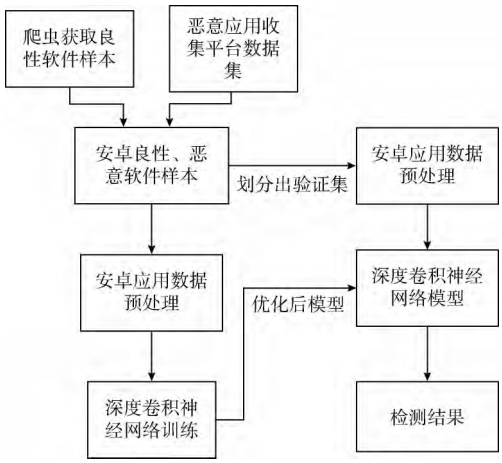

图 3　基于深度学习的恶意应用检测系统

3.1　数据集生成及预处理

利用爬虫方法从公开的市场获取良性应用样本，恶意应用样本则来自 VirusShare 平台的 2018 年度样本集，并随机划分 80%的样本为训练数据集，20%为验证集。之后对 APK 文件反编译，得到 AndroidManifest. xml 和 smali 文件，如图 4 从 AndroidManifest. xml 中提取出权限类、ACTION 类，从 smali 文件中提取 API 序列。将提取到的原始特征信息经过词向量编码后形成特征序列。

图 4　安卓应用特征提取

经过统计良性、恶意应用数据集中的安卓系统共有 135 种权限可以供应用申请，应用 API 经过统计分析共有 8432 个，然而恶意应用和良性应用数据集中的 API 按频率统计却有不同的情况。例如良性应用和恶意应用 API 频率统计中排名靠前的重合项很多，这是因为大多数应用都需要用到一些系统 API，如字符串类函数、文件类函数，但是这类 API 不具有较强的区分能力，因此本文的方法，采用在良性应用和恶意应用中频率差较高前 1500 个 API 作为所需的特征。

特征信息转化为序列的方式本文采用随机化初始化的词向量嵌入方法。在词嵌入之前往往采用 1-of-N Encoding（独热编码）的方法，这种编码方法会使得码字很长，若一共有 2000 个单词，就需要一个长度为 2000 的串进行编码，对计算资源是一种浪费。所以在本方法使用的特征信息不超过 2000 种的情况下，采用随机初始化为词向量的编码方式生成特征信息对应为词向量的词典，每类信息编码长度仅为 11 位。

最后将每个应用的权限类、API 类、ACTION 类信息根据统一随机初始化为词向量的编码方式进行编码，以此种编码方式将应用特征信息表示为二值化序列，从而便于输入神经网络进行训练。权限类、ACTION 类信息放到最前面，API 类信息在其后 3 种，3 种信息用英文句号分隔，然后每个 smali 文件都是同一个类的，将每个 smali 文件中的 API 放到用句号结尾，即每一类文件可作为一句话。如图 5 所示为某 APK 经过预处理后的 APK 文本的样式。

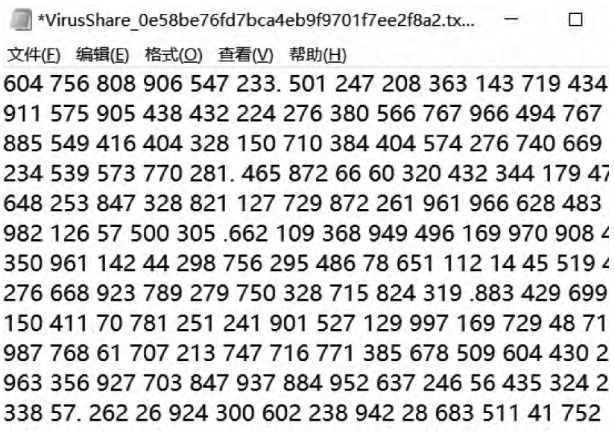

图 5　经过预处理后的 APK 文本

每一个样本的特征信息经过提取后按照权限信息、ACTION 信息和 API 信息的顺序进行排列，形成一个 n * m 大小的矩阵序列。

3.2　恶意应用检测模型优化

最后是卷积网络训练过程。由于预处理我们从中把含有权限信息和 API 类，ACTION 类序列的所有类连接在一起，用单个序列代表整个应用程序的特征信息。卷积神经网络的输入序列长度一般是固定的，然而不同的应用生成的序列长度是不一致的，此时需将较短的序列后面补充零值词向量。

卷积网络训练之前需要进行参数初始化，初始设置见表 1。

其中学习率是对权值修改的幅度，值越大对各层网络的权值的修改幅度越大，网络收敛速度越快。之后将应用特征信息形成的序列输入文本卷积神经网络，通过误差反向传播算法自动更新网络权重，对网络进行微调，自动学习特征，生成分类模型。

表 1 卷积网络参数初始化设置

参数	值	说明
Input	N * K	N * K 大小的矩阵
Kernel	3 * K,4 * K,5 * K,100	卷积核大小、个数
activation	ReLU	激活函数，用于非线性变换
η	0.001	学习率
Epoch	25	迭代次数

3.3 恶意应用检测系统验证

把经过参数优化的模型用验证集进行检测、记录、分析检测结果，并和传统机器学习方法对比。

相比于现有的恶意软件检测方法，所提方法可以克服动态检测方法时间长、依赖于特定工具的缺点。本方法只需要少量的数据预处理工作，即可自动学习数据的特征表示，相比于传统机器学习方法可以在更大的数据集上使用，并且通过向系统输入标记过的新样本，还可自动更新检测系统。

4 实验结果及分析

本文所用的数据集分别来自 360 应用市场和 VirusShare 平台，良性样本和恶意样本分别为 2160 个和 1342 个，分别随机划分其中的 80%作为训练集，20%作为验证集。实验平台配置见表 2。

表 2 实验平台配置

实验环境	环境配置
操作系统	windows 10
内存	16 GB
CPU	I5-8400
GPU	英伟达 GTX1080
深度学习框架	Tensorflow

本文中的良性、恶意应用数据集，根据样例的真实类别与深度神经网络预测类别的组合划分为真正类（TP）、假正类（FP）、真反类（TN）、假反类（FN）4 种情形。分类结果见表 3。

表 3 分类结果混淆矩阵

实际	预测	
	正例	反例
正例	TP(真正例)	FN(假反例)
反例	FP(假正例)	TN(真反例)

TP：真正类，恶意应用被预测为恶意应用

FN：假负类，恶意应用被预测成良性应用

FP：假正类，良性应用被预测成恶意应用

TN：真负类，良性应用被预测成良性应用

根据以上分类情形，便于评价检测效果的准确率，漏报率和误报率的定义如下所示

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F_A = \frac{FN}{TP + FN}$$

$$M_R = \frac{FP}{FP + TN}$$

本文的数据预处理中的反编译工具为 Apktool，深度学习框架为 Tensorflow，编程语言为 Python。经过深度学习模型训练后，采用数据集的 20%进行检测，得到实验结果见表 4。

表 4 同基于随机森林算法的方法的比较

指标	基于深度学习	随机森林
准确率/%	94.2	92.84
漏报率/%	9.3	—
误报率/%	3.2	—

本实验结果和采用机器学习的随机森林算法[12]的结果对比来看，本方法的精确度略高于随机森林算法，这是因为本方法将权限、API 类、ACTION 类等信息合理运用，采用适宜的特征信息向量化方式，运用深度学习可获得应用深层特征的能力，实现了较好检测的效果。

5 结束语

本文提出了一种基于深度学习的安卓恶意应用检测方法，创建了一个较大的包含恶意、良性应用的数据集，通过提取应用较少的静态信息，运用深度学习算法的能力表征应用的行为，实现了高效迅速的检测未知恶意应用。本文的方法相比于动态检测方法也可节省较多的时间，但是漏报率较高，下一步将研究在更大的数据集上进行实验，不同良性、恶意应用样本的比例下，降低漏报率，进一步提高检测效果。

参考文献：

[1] Bhat P, Dutta K. A Survey on various threats and current state of security in android platform [J]. ACM Computing Surveys, 2019, 52 (1): 21.

[2] LI Jianghua, QIU Chen. Survey of Android malware detection methods [J]. Application Research of Computers, 2019, 36 (1): 1-7 (in Chinese). [李江华，邱晨. Android 恶意软件检测方法研究综述 [J]. 计算机应用研究，2019，36 (1)：1-7.]

[3] QIN Zhongyuan, WANG Zhiyuan, WU Fubao, et al. An-

droid malware detection based on multi-level signature matching [J]. Application Research of Computers, 2016, 33 (3): 891-895 (in Chinese). [秦中元，王志远，吴伏宝，等. 基于多级签名匹配算法的 Android 恶意应用检测 [J]. 计算机应用研究，2016，33 (3)：891-895.]

[4] WANG Zhaoguo, LI Chenglong, GUAN Yi, et al. An anti-obfuscation method for detecting similarity of Android application [J]. J Huazhong Univ of Sci & Tech (Natural Science Edition), 2016, 44 (3): 60-64 (in Chinese). [王兆国，李城龙，关毅，等. 抗混淆的 Android 应用相似性检测方法 [J]. 华中科技大学学报（自然科学版），2016，44 (3)：60-64.]

[5] WANG Zhiqiang, ZHANG Yuqing, LIU Qixu, et al. Algorithm to detect Android malicious behaviors [J]. Journal of Xidian University, 2015, 42 (3): 8-14 (in Chinese). [王志强，张玉清，刘奇旭，等. 一种 Android 恶意行为检测算法 [J]. 西安电子科技大学学报，2015，42 (3)：8-14.]

[6] XU Xin, CHENG Shaoyin, JIANG Fan. Malware dynamic analysis cloud [J]. Computer System and Application, 2016, 25 (3): 8-13 (in Chinese). [徐欣，程绍银，蒋凡. 恶意软件动态分析云平台 [J]. 计算机系统应用，2016，25 (3)：8-13.]

[7] ZHANG Guoyin, QU Jiaxing, LI Xiaoguang. Way of Android malicious behavior detection based on Bayesian networks [J]. Computer Engineering and Applications, 2016, 52 (17): 16-23 (in Chinese). [张国印，曲家兴，李晓光. 基于贝叶斯网络的 Android 恶意行为检测方法 [J]. 计算机工程与应用，2016，52 (17)：16-23.]

[8] JIANG Chen, HU Yupeng, SI Kai, et al. Malicious file detection method based on image texture and convolutional neural network [J]. Journal of Computer Applications, 2018, 38 (10): 2929-2933 (in Chinese). [蒋晨，胡玉鹏，司凯，等. 基于图像纹理和卷积神经网络的恶意文件检测方法 [J]. 计算机应用，2018，38 (10)：2929-2933.]

[9] ZHAO Kai, A research of malware detection on feature selection algorithm [D]. Changsha: Hunan University, 2016 (in Chinese). [赵凯. Android 恶意应用检测中特征选择算法的研究 [D]. 长沙：湖南大学，2016.]

[10] LU Zhengjun, FANG Yong, LIU Liang, et al. Android malicious behavior detection method based on context information [J]. Computer Engineering, 2018, 44 (7): 150-155 (in Chinese). [卢正军，方勇，刘亮，等. 基于上下文信息的 Android 恶意行为检测方法 [J]. 计算机工程，2018，44 (7)：150-155.]

[11] LIU Tengfei, YU Shuangyuan, ZHANG Hongtao, et al. Recurrent neural networks and convolutional neural networks for text classification [J]. Computer Engineering & Software, 2018, 39 (1): 64-69 (in Chinese). [刘腾飞，于双元，张洪涛，等. 基于循环和卷积神经网络的文本分类研究 [J]. 软件，2018，39 (1)：64-69.]

[12] SONG Xin, ZHAO Kai, ZHANG Linlin, et al. Research on Android malware detection based on random forest [J]. Netinfo Security, 2019, 19 (9): 1-5 (in Chinese). [宋鑫，赵楷，张琳琳，等. 基于随机森林的 Android 恶意软件检测方法研究 [J]. 信息网络安全，2019，19 (9)：1-5.]