文章编号:1671-4598(2017)10-0198-04 DOI:10.16526/j.cnki.11—4762/tp.2017.10.050 中图分类号:TP311.1 文献标识码:A

# 基于函数调用图的 Android 恶意代码检测方法研究

李自清
(青海民族大学 物理与电子信息工程学院,西宁 810007)

**摘要**:随着移动互联网的迅猛发展和智能设备的普及,Android 平台的安全问题日益严峻,不断增多的恶意软件对终端用户造成了许多困扰,严重威胁着用户的隐私安全和财产安全;因此对恶意软件的分析与研究也成为安全领域的热点之一;提出了一种基于函数调用图的 Android 程序特征提取及检测方法;该方法通过对 Android 程序进行反汇编得到函数调用图,在图谱理论基础上,结合函数调用图变换后提取出的图结构和提取算法,获取出具有一定抗干扰能力的程序行为特征;由于 Android 函数调用图能够较好地体现 Android 程序的功能模块、结构特征和语义;在此基础上,实现检测原型系统,通过对多个恶意 Android 程序分析和检测,完成了对该系统的实验验证;实验结果表明,利用该方法提取的特征能够有效对抗各类 Android 程序中的混淆变形技术,具有抗干扰能力强等特点,基于此特征的检测对恶意代码具有较好地识别能力。

**关键词**:机器学习;Android 程序;函数调用图;图谱理论;特征提取

# Android Malicious Code Detection Method Based on Function Call Graph

Li Ziqing
(School of Physics and Electronic Information Engineering, Qinghai University for Nationalities, Xinin 810007, China)

**Abstract**: With the popularity of the rapid development of mobile Internet and smart devices, Android platform security issues become more and more serious, more malware caused a lot of trouble to the end user, a serious threat to the safety of the user's privacy and property safety. Therefore, the analysis and research of malware has become one of the hot topics in security field. An innovative practical feature extraction and detection of Android program scheme based on function call graph is proposed in this paper. On Android program disassembling function call graph is obtained by the method, which based on the spectral graph theory, combined with the function call graph transformation after extraction of graph structure and extraction algorithm to obtain a certain anti-interference ability of program behavior characteristics. On this basis, the prototype system is realized, and the system is verified by the analysis and detection of a number of malicious Android programs. The experimental results show that the features extracted by this method can effective against all kinds of Android application confusion deformation technology, has the characteristics of strong anti-jamming ability. Based on this feature detection of malicious code has better recognition ability. 。

**Keywords**: machine learning; Android program; function call graph; spectral graph theory; feature extraction

## 0 引言

Android 系统得到应用以来,以较低成本的开销、良好的用户体验和较高的开源性等优点获得了广泛好评。但由于简单的安全检查机制,引来无数恶意者对于 Android 应用市场进行恶意攻击。各类恶意软件对于 Android 平台的攻击包括恶意扣费、窃取隐私、消耗资源、远程控制、恶意传播等严重影响了用户的信息安全,对个人隐私、企业发展、国家安全都造成了严重的威胁和不可挽回的损失。如何检测 Android 恶意程序成了信息安全中不可忽视重要任务。冯本慧[1]指出传统的 Android 恶意程序检测技术过度依赖分析人员的经验,且无法检测未知恶意程序等问题,面对当前数目庞大的且层出不穷的 Android 恶意程序,分析效率低下。因此需要利用数据挖掘技术实现检测的自动化、智能化。而当前此方法的研究重点主要分为两个方面,一是不同分类算法的选择使用,不同分类算法的效率与精度各有不同,二是 Android 程序的特征以及特征选择的方法,不同特征描述的是恶意代码不同层面的信息 Android 程序特征提取技术在此背景下应运而生。

本文提出了一种基于函数调用图的 Android 恶意程序提取和检测技术,该方法在图谱理论基础上,结合函数调用图变换后提取出的图结构和提取算法,获取出具有一定抗干扰能力的程序行为特征,将提取出的特征输入决策树模型进行训练得到预测系统。图特征提取只需进行静态提取,不需要动态运行,且实现了检测的智能化与自动化,易于应用于未知恶意程序检测。实验结果表明该方法对常见的混淆变形后的恶意程序具有较好的鲁棒性和时效性。

## 1 Android 恶意程序特征提取与预测模型

### 1.1 Android 恶意程序特征提取技术分析

Android 字节特征提取技术是利用定长或者变长的 n-gram 滑动窗口滑动恶意代码文件所获得的序列信息。Reddy[2]使用此技术在文本分类中取得了很好的效果。字节序列作为特征,虽然可以获得恶意代码的编码风格等有用信息,但由于缺乏语义信息,对于加密混淆后的病毒检测效率低下。Altyeb[3]开始利用恶意代码的应用程序编程接口做为特征,由于恶意代

**收稿日期**:2017-03-29; **修回日期**:2017-04-13。
**基金项目**:教育部"春晖计划"合作科研项目(S2015037)。
**作者简介**:李自清(1975-),男,陕西人,硕士,讲师,主要从事计算机应用技术方向的研究。

码需要实现自身的功能需要借助操作系统提供的 API 完成，而 API 的调用序列可以表示恶意代码的行为以及涉及的语义信息。API 调用序列分为静态和动态的调用序列，静态调用序列不需要运行程序的前提下获得文件导入表或者反汇编文件中的 API 调用序列，动态调用序列即需要在虚拟机运行程序中利用调试等技术获得与系统交互的 API 调用序列。获得 API 调用信息作为特征，使用分类算法进行检测，最后达到了较高的精度。然而，由于恶意代码会隐藏导入表 API 的调用，使得无法获得全部 API 调用信息，最终导致静态 API 序列作为特征检测恶意代码的效率不高，但恶意代码需要完成自身功能及时隐藏导入表 API 调用，也会与操作系统中的 API 进行交互，因此 Zhang M[4]将可疑文件置于虚拟机中运行动态获得 API 序列，并计算 API 序列和正常文件的距离作为特征进行检测。动态获得 API 调用序列的特征进行检测技术中，要获得特征就必须运行恶意代码，导致开销过大，而对于某些能够检测到虚拟机存在环境的病毒无能为力，并且一些恶意代码采用了相关行为层的混淆技术，导致动态提取 API 调用序列失败。函数调用图是编译过程对程序中函数调用关系的一种静态描述，其中节点表示函数，边表示函数之间的调用关系，由于程序的功能性主要由库函数和系统调用来决定，因此函数调用图能为程序的实际行为提供静态的有效近似，是程序的结构化表示形式，对于基于源码或二进制码的局部软件变形具有鲁棒性，函数调用图通过 IDA Pro 这种成熟的交互式反汇编工具生成。因此部分研究人员从将研究重点转移到了提取函数调用图的特征作为程序特征图。刘星[5]提出用函数调用图的相似性距离度量两个代码的相似性来检测恶意代码，该方法考虑了恶意代码中函数的指令级信息以及函数之间的调用关系。杨帆[6]提出了基于二分图匹配的图编辑距离检测恶意程序，图编辑距离通过将一个图转换为另一个图需要编辑操作集合的最小数量来度量程序间的关系。赖兴瑞[7]提取出函数依赖图的最大公共子图作为中文 web 文本分类。颜克文[8]在安卓程序相似性比较中通过函数调用图转换为特征向量，以图特征向量之间相似性衡量程序相似性不失为一种办法，但是提取出的特征向量不具备相同维度，一次只能对少数程序进行比较。

因此，不论是图编辑距离、最大公共子图还是图同构方法在 Android 海量应用场景下计算代价非常大[9-11]，要对大量程序进行检测，其实现的时间复杂度和空间复杂度不具备可行性。

### 1.2　预测模型建立

Android 恶意程序特征提取与预测模型分为 3 个模块：特征提取模块、构造分类器模块、预测模块组成如下图 1 所示：

1）输入安卓程序被称为“训练数据”，每组训练数据有一个明确的标识或结果，如 Android 程序对应着“恶意程序”或“正常程序”[12]。在建立分类模型之前，需要对数据进行预处理即将 Android 程序进行反编译，从函数调用关系中提取出依赖特征、构造函数调用图的特征向量，接着建立一个学习过程，将处理后的特征向量输入分类模型中，将分类模型的预测结果与“训练数据”的实际结果进行比较，不断的调整预测模型，直到模型的预测结果达到一个预期的准确率。

2）构造分类器模块中，分类器采用梯度上升决策树实现，原理为：算法开始时，为每个样本赋上一个估计值，初始时每个样本的估计值都一样，在每一步训练中得到的模型（分类回归树），会是的数据点的估计有对有错，计算数据点估计值与实际值的残差的梯度，在每一次模型梯度减少的方向建立一个新模型，重复 $N$ 次（$N$ 由用户指定），会得到 $N$ 个简单的分类器（分类回归树），将 $N$ 个分类器组合起来（加权、或者进行投票等），得到一个最终的模型

3）病毒检测模块中，分类器中得到 $N$ 个简单分类器，用 $N$ 个分类器依次对未知病毒文件进行判断，每次判断病毒文件按照树的分支条件选择符合自己的叶子节点，计算每次分类器判断得到的叶子节点的信息增益，相加最多的为最终结果。

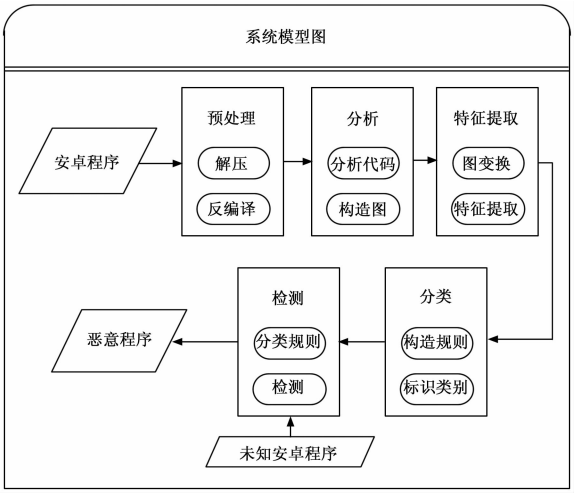


图 1　Android 恶意代码检测模型图

## 2　程序特征提取与检测技术

### 2.1　Android 程序预处理技术

Android 平台的应用程序包以 APK 格式为主，用户将 APK 文件直接传输到手机上即可进行安装。因此，Android 恶意程序的各种恶意行为，大多是通过在应用程序包中植入恶意代码或恶意组件来实现的。恶意攻击者将恶意程序注入并隐藏在安全应用程序中，经过重新打包，表现出同原安全应用程序一样的外在，导致程序使用者将其认定为安全应用程序下载，并安装到自己的 Android 手机上。当程序启动后，被注入到安全应用程序中的恶意程序就开始了自己的恶意行为，对用户信息安全构成了严重的威胁。在安全应用程序中注入恶意程序 主要由以下步骤构成，第一步对其进行反汇编；第二步在反汇编后的安全应用程序中注入恶意代码，加入的应用程序的功能和恶意攻击的代码的内容有关；第三步将被改写过的安全应用程序重新打包并签名。进行以上三步，一个 Android 恶意程序就形成了[13]。要提取函数调用图需在第三方网站上下载 APK 文件，对 APK 文件进行解压，META－INF 为存放签名信息文件 res 目录存放资源文件，classes.dex 是 java 源码编译后生成的 java 字节码文件，是最终用来被虚拟机 Dalvik 加载和运行的可执行 Android 文件。对 classes.dex 文件反编译，根据反编译的程序即可生成函数调用图，例如将一个手机照明软件反编译后提取出的部分函数调用图如图 2 所示。

其中节点表示函数，边表示函数之间的调用关系，有向边也称弧，边的始点称为弧尾，终点称为弧头，一条弧表示弧尾的节点函数调用弧头的节点函数。

### 2.2　函数调用图特征提取技术

得到了函数调用图 $G$ 后需要提取图的特征向量，从函数

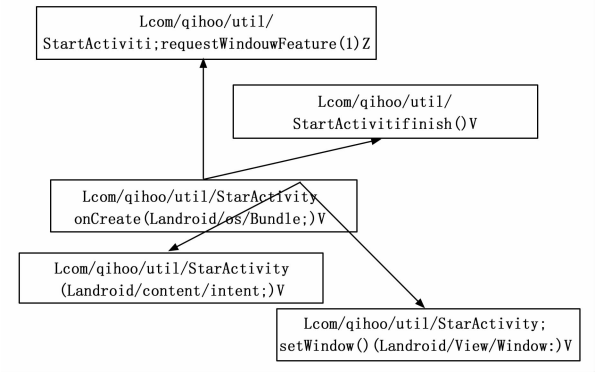


图 2 函数调用图

调用图中提取特征向量具体步骤如下：

1）一个图有 $G$ 顶点集 $V(G)=\{V_1, V_2, \cdots, V_n\}$ 和边集 $E(G)=\{eI_1, e_2, \cdots, e_n\}$，邻接矩阵 $\boldsymbol{A}(G)$ 是一个 $N\times N$ 阶矩阵，若 $G$ 的顶点 $V_iV_j$ 相邻，那么邻接矩阵 $\boldsymbol{A}(G)_{i,j}$ 的元素取值为 1，否则为零。

将函数调用图 $G$ 转化为邻接矩阵为 $G\rightarrow \boldsymbol{A}(G)$ 的矩阵，$N$ 为函数调用图的节点数。

2）转移概率：根据顶点的出度信息和顶点之间的调用关系计算顶点之间的转移概率，通过计算顶点之间的转移概率体现类依赖关系的调用概率。

$u$ 表示主调顶点，$v$ 表示被调顶点，$out$ 表示顶点的出度，$in$ 表示顶点的入度。转移概率为：

$$TP(u,v)=\frac{u,out}{u.out+v.in}\times\frac{v.in}{v.out+v.in}$$

计算转移概率矩阵 $\boldsymbol{A}(G)\rightarrow D(G)$

3）谱图论是一种常用的研究方法，其可以利用拉普拉斯矩阵来描述图的结构。通过分析 拉普拉斯矩阵及其特征值能够对图结构有更清晰的认识，特别是在很多情况下，需要提高拉普 拉斯矩阵的次小特征值 $\lambda 2$，以使图结构得到优化。为了更好地反映图的全局信息，可以对函数调用图做拉普拉斯变换，拉普拉斯矩阵是度矩阵和邻接矩阵的差。度矩阵是一个对角矩阵，其包含了每个顶点的度。在处理有向图时，根据应用来选择入度或出度。对转移概率矩阵做拉普拉斯变换 $D(G)$ $L(G)$。

4）矩阵特征值的集合称作图的谱。设 $A$ 是 $n$ 阶方阵，如果数 $\lambda$ 和 $n$ 维非零列向量 $x$ 使 关系式 $Ax=\lambda x$ 成立，那么这样的数称为矩阵特征值，非零向量 $x$ 称为 $A$ 的对应于特征值的特征向量，图的特征向量已被广泛应用以及被证实可以反映图的特性。由于邻接矩阵与点的标记有关。谱是一个图常量，当两个图的邻接矩阵有相同的特征集时，它们被称为谱相似。谱相似的图不必同构，但同构的图必谱相似，因此需要求出变换后的拉普拉斯矩阵 $L(G)$ 的特征值 $(\lambda_1\lambda_2\cdots\cdots\lambda_m)$ 以及特征值所对应的特征向量 $(\mu_1\mu_2\cdots\cdots\mu_1)$。

5）将特征向量按对应特征值大小排序，取前 $k$ 个特征向量 $(\mu_1\mu_2\cdots\cdots\mu_k)$。

6）图的谱特征选择谱系数夹角谱特征，对于图 $G$，目标是找出同序列图像顶点之间具有一致性的相似性，及不同序列图像的差异性。图的普特征已经被广泛证实可以反映图的特性，因此使用谱系数夹角特征作为图的特征向量提取。谱系数夹角定义为图中在特征向量空间的特征向量各分量之间的夹角的余弦值：

$$C(i,j)=\cos\theta(ij)=\frac{\xi(i)^T\xi(j)}{\|\xi(i)\|\|\xi(j)\|}$$

用谱系数夹角表达图特征，计算各特征向量之间的余弦夹：

$$v=(C(1,2),C(1,3)\cdots\cdots C(1,k),C(2,3)\cdots\cdots C(2,k)\cdots\cdots C(k-1,k))$$

函数调用图提取特征向量步骤如图 3 所示。

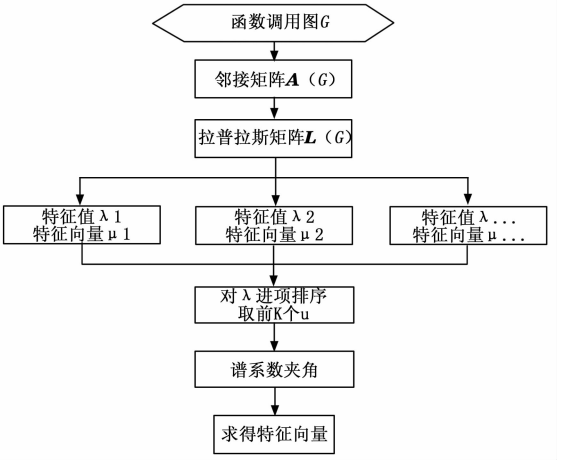


图 3 特征提取流程图

### 2.3 Android 程序预测技术

#### 2.3.1 构造分类器技术

依照上述方法提取出 APK 样本函数调用图的特征向量，每个样本取出函数调用图的特征向量，构造分类器。构造分类器的本质是使用某种算法对已知类别数据集进行训练并得到一个分类模型，该数据集是由多个属性组成的特征向量，其中包含类别标记，训练结果输出一组可以对未知 型的数据进行预测分类的判定规则。

本技术的分类器采用梯度上升决策树方法（Gradient Boosting Decicion Tree）实现，原 理为：算法开始时，为每个样本赋上一个估计值，初始时每个样本的估计值都一样，在每一 步训练中的到的模型（分类回归树），会是的数据点的估计有对有错，计算数据点估计值与实际值的残差的梯度，在每一次模型梯度减少的方向建立一个新模型，重复 $N$ 次（$N$ 由用户指定），会得到 $N$ 个简单的分类器（分类回归树），将 $N$ 个分类器组合起来（加权、或 者进行投票等），得到一个最终的模型。构造分类器方法如图 4 所示。

$N$ 个简单的分类器采用分类回归树构造，具体步骤如下：

1）输入 $X$ 和 $Y-p$ 的梯度 $g$

2）遍历 $X$，选择任意特征的特征值 $X(ij)$，计算所有用特征值划分后的信息不纯性度（可 以选择 GINI 指数、双化指数、有序双化指数），信息不纯度越大代表信息当前 X 包含的病毒 种类越杂乱，找到信息不纯度最小时的 $X(kl)$。

3）求出 $X(il)\{i: 0\sim m-1\}$ 中所有大于 $X(kl)$ 的值 $d(k)\{k: 1, 2\}$，将 $X$ 的所有的 $d(k)$ 行 组成左子树数据集 $LX$，对应的 $g$ 为 $Lg$，将剩下的行组成右子树数据集 $RX$，对应的病毒种类为 $Rg$。

4）若划分后的 $LX$，$RX$ 的数量是否小于用户设定数量，返回 1）继续寻找 $X(kl)$，$r$ 若信息不纯性度量是小于一定值

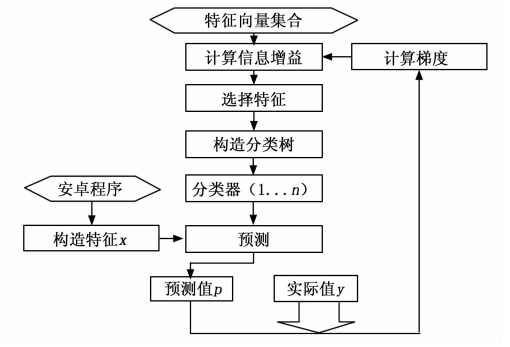


图 4　迭代分类器构造图

则执行 5），否则执行 7）。

5）返回当前 $X$ 对应的 $g$ 的平均值为分类树的叶节点。

6）选择 $X$（$kl$）为分裂结点，节点的左子树是大于特征值的数据集 $LX$，$Lg$，节点的右子树 小于等于特征值的数据集 $RX$，$Rg$。

7）将左子树的数据集 $LX$，$Lg$ 做为新数据集 $X$，$g$，执行 2）。

8）将右子树的数据集 $RX$，$Rg$ 作为新数据集 $X$，$g$，执行 3）。

2.3.2　预测技术

分类器中得到 $N$ 个简单分类器，用 $N$ 个分类器依次对未知病毒文件进行判断，每次判断病毒文件按照树的分支条件选择符合自己的叶子节点，计算每次分类器判断得到的叶子节点的信息增益，想加最多的为最终结果

## 3　实验与分析

### 3.1　实验数据来源

实验非恶意程序样本主要从 Google Play 上下载，包括游戏娱乐、工具类、系统管理类 等程序。虽然据报道称 Google Play 也存在会出现恶意软件的可能性，但一经发现便会被立刻下架，我们认为在所有软件市场中 Google Play 的审核是相对严格的，因此我们假定下载 的程序为正常非恶意程序。虽然常有 Android 恶意软件的报道，但并没有公开的恶意软件样本库，主要通过平时在虚拟机上运行手机报毒的恶意软件作为恶意程序样本。

### 3.2　实验结果分析

随机选取参与训练的良性程序和恶意程序各 50 个提取 10 维特征后输入分类模型进行分类实验，再选择未参与训练的新数据进行检测，进行五轮实验，其中选择 7 项标准来评测实验结果。

$TP$ 为真阳性，即检测正确的恶意样本数量；

$FP$ 为假阳性，即检测错误的良性样本数量；

$FN$ 为假阴性，即检测错误的恶意样本数量；

$TN$ 为真阴性，即检测正确的良性样本数量；

$TPR$ 为真阳率，$TPR=TP/P=TP/(TP+FN)$ 表征一种命中率；

$FPR$ 为假阴率，$FPR=FP/N=FP/(FP+TN)$ 表征一种错误命中率；

$ACC$ 为检测正确所有样本所占总样本量的比值。实验结果如表 1 所示。

表 1　样本检测结果表

| | 序号 | $TP$ | $FN$ | $TN$ | $FP$ | $TPR$/% | $FPR$/% | $ACC$/% |
|---|---|---|---|---|---|---|---|---|
| 训练样本数 | 1 | 34 | 16 | 29 | 21 | 68 | 42 | 63 |
| | 2 | 40 | 10 | 35 | 15 | 80 | 30 | 75 |
| | 3 | 30 | 20 | 33 | 17 | 60 | 34 | 63 |
| | 4 | 32 | 18 | 31 | 19 | 64 | 38 | 63 |
| | 5 | 33 | 17 | 40 | 10 | 66 | 20 | 73 |
| 新样本数 | 1 | 21 | 29 | 44 | 6 | 42 | 12 | 65 |
| | 2 | 22 | 28 | 32 | 18 | 44 | 36 | 54 |
| | 3 | 37 | 13 | 22 | 28 | 74 | 56 | 59 |
| | 4 | 40 | 10 | 32 | 18 | 80 | 36 | 72 |
| | 5 | 35 | 15 | 31 | 19 | 70 | 38 | 66 |

由以上模拟实验可以看出，训练样本的预测效果相对较好，对于新出现的样本具有一定的检测能力，但不够稳定和理想，原因可能是输入训练模型的样本数量偏少，另外如果能按软件的分类如游戏、娱乐、工具、系统 管理等来进行训练与检测也许效果会更好，因为不同类型的软件所需要的调用的函数以及权限集合也有所不同。

### 3.3　修正实验

因此扩大样本容量，以及按软件的分类进行训练，随机选取游戏娱乐、音乐软件、聊天 220 工具、办公软件、下载工具类别各 50 个软件进行训练，再选择未参与训练的新数据进行检测，实验结果如表 2 所示。

表 2　样本检测结果表

| | 类别 | $TP$ | $FN$ | $TN$ | $FP$ | $TPR$/% | $FPR$/% | $ACC$/% |
|---|---|---|---|---|---|---|---|---|
| 训练样本数 | 游戏 | 37 | 13 | 35 | 15 | 74 | 30 | 72 |
| | 音乐 | 40 | 10 | 35 | 15 | 80 | 30 | 75 |
| | 聊天 | 41 | 9 | 9 | 11 | 82 | 22 | 80 |
| | 办公 | 35 | 15 | 33 | 17 | 70 | 34 | 68 |
| | 下载 | 33 | 17 | 40 | 10 | 66 | 20 | 73 |
| 新样本数 | 游戏 | 34 | 16 | 44 | 6 | 68 | 12 | 78 |
| | 音乐 | 29 | 21 | 32 | 18 | 58 | 36 | 61 |
| | 聊天 | 37 | 13 | 28 | 22 | 74 | 44 | 65 |
| | 办公 | 39 | 11 | 31 | 19 | 78 | 38 | 70 |
| | 下载 | 30 | 20 | 27 | 13 | 60 | 26 | 57 |

由以上模拟实验可以看出，将数据按类别分别训练样本后，由于不同类型的软件所需要的权限集合与组合有所不同，因此检测效果相对之前较好，学习到的模式与规律在各种类型的恶意代码中是基本稳定的，对新出现的恶意代码也有较好的检测能力。

## 4　结论

本文提出了 Android 恶意程序特征提取与检测新方案，该方案采用静态分析技术，基于 Android 程序恶意代码本身的函数调用顺序及程序结构特征，将恶意代码抽象为图结构，提取出图结构特征放入迭代决策树系统进行训练，得到预测模型来实现对未知恶意程序的有效判断。主要特点如下：1）通过图相似性对比转化为图特征向量相似性对比，提高了匹配效率，有效识别恶意代码的变种，时间效率高，不依赖于人工分析；2）用图的谱夹角表达图特征，既表达了空间各点的连接关系，同时不受空间尺度的影响；3）应用决策树技术实现检测的自

（下转第 205 页）

率的不断增加而随之不断下降，但是在另一方面可以看出虽然剪裁攻击对于水印的恢复质量产生了影响，但是其并没有造成致命性的破坏，水印的质量依然可以满足需求。这证明所提方案针对于剪裁攻击具有相当好的健壮性。

### 4.2　实验方案的不足

该通信方案结合扩频水印技术设计了安全通信方案，虽然可以实现对信息安全传输的目的需求，但是依然存在一些有待改进的地方。

1）该方案采用水印技术对信息进行隐藏，水印算法是多种多样的，这里只采用了小波算法做水印隐藏，没有对其他算法进行相应的分析，这导致了该方案的片面性。

2）为了能够满足信噪比的要求采用了扩频技术进行相应的处理，但是这样做对信号的处理性能提出了更高的要求，同时也给算法复杂度带来了挑战，降低了运行效率。

## 5　结论

本文基于扩频数字水印技术，结合小波变换算法针对战场对于信息安全的高要求设计了基于扩频数字水印技术的安全通信方案。提出了首先将待加载的水印信息转换为一维二进制信息，随后对其进行采样调制进行水印信息的预备工作，然后使用秘钥产生 $m$ 序列的扩频编码技术对其进行扩频编码，这一方法使得水印信息的安全性获得了极大的提高。随后通过小波变换对水印进行嵌入与提取操作，控制合适的水印能量使得水印信息不可被感知然后经过通信系统的传输接收，这同时保护了水印信息难以收到破坏性攻击的侵入。最后通过仿真实验证明该方法是有效的，可以实现信息的隐藏，同时其对于破坏性攻击具有一定的防护。该通信方案针对于保密传输拥有很高的安全性，同时对于剪裁攻击也具有很好的鲁棒性，在实际应用中具有很广阔的前景。

**参考文献：**

[1] 孙　锐，孙　洪，赵晓岚. 基于量化的扩频水印技术 [J]. 通信技术，2002 (2)：63－66.

[2] 龚　阳，崔　琛，王　津，等. 基于扩频通信的无线抄表系统设计与实现 [J]. 计算机测量与控制，2016，24 (12)：237－240.

[3] 周利军. 数字图像水印的扩频实现 [J]. 红外与激光工程，2000，29 (5)：27－31.

[4] 张　东，倪江群，李大捷. 基于 GSM 模型的扩频水印安全性分析 [J]. 自动化学报，2009，35 (7)：841－850.

[5] 孙圣和，陆哲明. 数字水印处理技术 [J]. 电子学报，2000，28 (8)：85－90.

[6] 程兴国，高　升. 信息隐藏与数字水印 [J]. 湖北工业职业技术学院学报，2004，17 (2)：65－67.

[7] 兀旦晖，郑恩让. 基于混沌 Logistic 和 Arnold 二次加密的图像水印算法研究 [J]. 计算机测量与控制，2017，25 (4)：193－196.

[8] 钮心忻，杨义先，NIUXin－Xin，等. 基于小波变换的数字水印隐藏与检测算法 [J]. 计算机学报，2000，23 (1)：21－27.

[9] 傅德胜，孙文静，张小飞. 基于混沌特性的小波数字水印技术及实现 [J]. 计算机科学，2008，35 (6)：246－250.

[10] 徐祗军，吴晓娟，杜会斌，等. 扩频数字水印与军事通信 [J]. 军事通信技术，2005 (4)：58－60.

[11] 胡　鹏. 基于正交扩频码和 HVS 的 DCT 域图像数字水印技术 [J]. 信息安全与通信保密，2008 (7)：80－82.

[12] 刘忠英，许金勇，柳永祥. 频谱水印技术与军事应用分析 [J]. 现代军事通信，2013 (1)：45－48.

[13] 刘勇顺. 数字水印技术在军事信息安全中的应用 [J]. 现代电子技术，2006，29 (19)：64－66.

[14] 靳小晖. 音频信息隐藏技术在军事通信中的实际运用 [J]. 信息通信，2015 (5)：178－178.

[15] 康　芳，谭　薇，杨森斌. 信息隐藏技术及其在军事通信领域的应用研究 [J]. 现代电子技术，2008，31 (23)：97－99.

(上接第 201 页)

动化与智能化，由于学习到的模式与规律在各种类型的恶意代码中是基本稳定的，对新出现的恶意代码也有很好的检测能力。

目前实验的恶意代码库规模较小，特征提取的维数较小，图特征提取没有对冗余信息进行处理，这是下一步需要解决和完善的问题。

**参考文献：**

[1] 冯本慧. 基于数据挖掘于机器学习的恶意代码检测技术研究 [D]. 长沙：中南大学，2013.

[2] Nagaprasad S，Reddy T R，Reddy P V，et al. Empirical evaluations using character and word N－grams on authorship attribution for Telugu text [M]. Intelligent Computing and Applications，Springer India，2015：613－623.

[3] Shubair A，Ramadass S，Altyeb A A. kENFIS：kNN－based evolving neuro－fuzzy inference system for computer worms detection [J]. Journal of Intelligent & Fuzzy Systems Applications in Engineering &Technology，2014，26 (4)：1893－1908.

[4] Zhang M，Duan Y，Yin H，et al. Semantics－ AwareAndroid Malware Classification Using Weighted Contextual API Dependency Graphs [A]. ACM [C]. 2014：1105－1116.

[5] 刘　星. 恶意代码的函数调用图相似性分析 [J]. 计算机工程与科学，2014：1－3.

[6] 杨　帆. 基于图编辑距离的恶意代码检测 [J]. 武汉大学学报，2013：1－3.

[7] 赖兴瑞. 基于最大公共子图的中文 Web 文本分类研究 [D]. 厦门：厦门大学，2011.

[8] 颜克文. 基于图特征向量的安卓程序相似性检测算法研究 [D]. 湘潭：湘潭大学，2014.

[9] Seo S H，Gupta A，Mohamed S A，et al. Detecting mobile malware threats to homeland security through static analysis [J]. Journal of Network and Computer Applications，2014，38：43－53.

[10] Jang J，Woo M，et al. Towards automatic software lineage inference [A]. Proceedings of the 22nd USENIX conference on Security. USENIX Association [C]. Berkeley，CA，USA：USENIX Association，2013：81－96.

[11] Grace M，Zhou Y，Zhang Q，et al. RiskRanker：Scalable and Accurate Zero－day Android Malware Detection [A]. Proceedings of the 10th International Conference on Mobile Systems，Applications and Services (MobiSys'12) [C]. 2012：4－20.

[12] Arp D，Spreitzenbarth M，Hübner M，et al. Drebin：Efficient and Explainable Detection of Android Malware in Your Pocket [A]. Proceedings of the 21th Annual Network and Distributed System Security Symposium (NDSS'14) [C]. 2014.

[13] Peng H，Gates C，Sarma B，et al. Using Probabilistic Generative Models for Ranking Risks of Android Apps [A]. Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS' 12) [C]. 2012：2－5.