

## 基于特征融合的恶意代码分类研究

张景莲<sup>1,2</sup>, 彭艳兵<sup>2</sup>

(1. 武汉邮电科学研究院, 武汉 430000; 2. 南京烽火天地通信科技有限公司, 南京 210000)

**摘 要:** 基于特征码匹配的静态分析方法提取的特征滞后于病毒发展, 且不能检测出未知病毒。为此, 从病毒反编译文件及其灰度图出发进行特征提取及融合, 采用机器学习中的随机森林(RF)算法对恶意代码家族进行分类, 提取恶意代码的操作码指令和灰度图纹理 2 个局部特征, 并将颜色直方图作为恶意代码的全局特征。实验结果表明, 融合恶意代码特征与 RF 算法可实现恶意代码家族的有效分类, 平均准确率达到 99.59%。

**关键词:** 恶意代码; 机器学习; 灰度图纹理; 颜色直方图; 随机森林

**中文引用格式:** 张景莲, 彭艳兵. 基于特征融合的恶意代码分类研究[J]. 计算机工程, 2019, 45(8): 281-286, 295.

**英文引用格式:** ZHANG Jinglian, PENG Yanbing. Research on malware code classification based on features fusion[J]. Computer Engineering, 2019, 45(8): 281-286, 295.

## Research on Malware Code Classification Based on Features Fusion

ZHANG Jinglian<sup>1,2</sup>, PENG Yanbing<sup>2</sup>

(1. Wuhan Research Institute of Posts and Telecommunications, Wuhan 430000, China;

2. Nanjing Fiberhome World Communication Technology Co., Ltd., Nanjing 210000, China)

**[Abstract]** The features extracted by the static analysis method based on feature code matching often lag behind the development of the virus and cannot detect unknown viruses. Therefore, the feature extraction and fusion are carried out from the virus decompilation file and their grayscale image. The Random Forest(RF) algorithm in machine learning is used to classify the malware code family, extract the two local features of the opcode instructions and grayscale texture of the malware code, and use the color histogram as the global feature of the malware code. Experimental results show that the fusion of malware code features and RF algorithm can achieve effective classification of malware code families, and the average accuracy rate is 99.59%.

**[Key words]** malware code; machine learning; grayscale texture; color histogram; Random Forest(RF)

DOI:10.19678/j.issn.1000-3428.0051790

### 0 概述

据微软统计, 2007 年全球约三分之一的计算机被安装了木马、后门等各种恶意软件, 恶意代码的快速变种及混淆手段<sup>[1]</sup>使其越来越难以识别, 对网络安全造成重大威胁<sup>[2]</sup>。2017 年全球爆发的大规模勒索病毒感染事件, 用户只要上网就可被攻击, 需支付赎金才能解密恢复文件, 攻击活动使得大量高科技设施、大型企业机构的正常活动遭到严重破坏, 甚至导致校园教学系统瘫痪。恶意代码变种的检测技术成为目前网络信息安全领域研究的热点<sup>[3]</sup>。当前的病毒分析方法主要有基于内容的静态分析方法和基于行为的动态分析方法<sup>[1,4]</sup>, 检测技术包括基于规则、基于特征码和基于算法的检测技术。

静态分析方法通过查看病毒二进制文件或反编

译文件, 确定其特征码进行准确查杀<sup>[3,4]</sup>, 然而恶意代码的各种变形使得反病毒软件很难再提取出一段代码作为恶意代码的特征码<sup>[1]</sup>, 一些复杂加壳<sup>[3]</sup>的样本不容易做反汇编, 而且精准的特征码匹配需要定期更新病毒库<sup>[3]</sup>, 无法应对迅速变种的代码, 因此静态方法的局限较大。动态分析方法需在实际或虚拟环境(如沙箱)下执行恶意代码<sup>[1]</sup>, 通过考察代码执行过程中样本对操作系统中各种资源的操作来构建并分析特征<sup>[5]</sup>, 其关键是保证恶意代码的有效执行, 而恶意代码的各种反检测技术使其具有反沙箱、反调试功能, 病毒在沙箱内运行时间短, 导致其行为不能被完全激发。基于规则的方法根据病毒行为编写规则, 与基于算法的检测方法存在共同的缺陷: 行为特征少, 需要混合动态和静态特征来构建检测系统<sup>[6]</sup>。

**基金项目:** 国家重点研发计划“现代服务业共性关键技术研发及应用示范”(2017YFB1400704)。

**作者简介:** 张景莲(1992—), 女, 硕士研究生, 主研方向为网络安全、计算机视觉; 彭艳兵, 高级工程师、博士。

**收稿日期:** 2018-06-12      **修回日期:** 2018-08-02      **E-mail:** yslm206@gmail.com

目前,机器学习理论被越来越多地运用于检测未知恶意代码的研究中。以恶意代码行为序列报告为基础,采用朴素贝叶斯和操作相似度窗口进行自动化恶意代码分类<sup>[4]</sup>以及基于恶意代码静态特征与栈式自编码分类模型结合的方法<sup>[7]</sup>均取得了不错的结果。在恶意代码检测中,特征描述算法是影响变种检测能力和检测效率的核心技术<sup>[3,5]</sup>。本文利用恶意代码反汇编文件提取不同层次的特征,基于恶意代码特征与机器学习算法对恶意代码进行分类研究。

## 1 恶意代码文件特征提取

本文从病毒反编译文件中提取恶意代码 2 种局部特征:Opcode N-gram 和灰度图纹理,并将颜色直方图作为恶意代码的全局特征。

### 1.1 Opcode N-gram

Opcode N-gram 在先前的恶意代码分类研究中已被证明是非常有效的特征<sup>[8-9]</sup>。N-gram 模型被称为一阶马尔科夫链,通过长度为  $N$  的滑动窗口收集一系列重叠的子字符串。模型基于以下假设:第  $n$  个词的出现只与它前面  $n-1$  个词有关。整句的概率就是各个词出现概率的乘积<sup>[8]</sup>。

恶意代码经 IDA Pro 反编译生成 PE 可执行文件(.asm 文件)和 16 进制字节码构成的字节文件(.bytes 文件)。PE 文件的节表程序段包括代码段(.text,有些文件是.CODE)、数据段(.data)、资源段(.rsrc)等<sup>[10-11]</sup>。代码段包含恶意代码的操作指令。从软件层面看,这些 Opcode<sup>[12]</sup>汇编指令及其组合是组成一个逻辑函数的最小单元,代表恶意代码的核心。已有研究表明,Opcode N-gram 比 Byte N-gram 能够更直观地描述二进制样本的功能性特征。基于此,本文选择对 PE 文件 Opcode 操作指令进行 N-gram 特征提取,并将其作为区分不同恶意代码家族的特征之一,算法具体如下:

#### 算法 1 Opcode N-gram 特征提取

输入 恶意代码.asm 文件,样本总数  $M$

输出 Opcode N-gram 特征向量  $X_i$ ,表示第  $i$  个恶意代码文件的第  $j$  个元组权重,  $i \leq M$

1. 提取  $M$  个.asm 文件程序段.text 或.CODE 中的所有操作码指令,形成序列。

2. 提取 N-gram 特征,如长度为  $k=4$  的序列[ mov, push, test, jnz]经  $n=3$  的窗口滑动后得到  $(k-n)+1$  个元组{ mov, push, test}, { push, test, jnz}。

3. 统计所有元组出现的次数,进一步特征选择选出特征集元组。

4. for  $\forall m \in M$  do

统计  $m$  个文件特征集元组在文件中出现的频次

end for

### 1.2 灰度图纹理特征提取

#### 1.2.1 恶意代码可视化

文献[13]利用 B2M 算法将恶意代码二进制文件转化为无压缩灰度图的原理如图 1 所示。

1000111001010  
1101100110010  
...

8 bit 一组转为  
无符号整型  
(0~255)

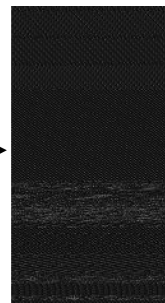


图 1 二进制文件转化为无灰度压缩图的过程

B2M 算法具体描述为:以二进制比特流<sup>[14]</sup>的方式读取恶意代码文件,将其转化为以字节为单元的十六进制向量,设定固定宽度  $width$  (一般是 2"),长度  $length$  由文件大小和宽度自动获得 ( $length = \text{文件长度}/width$ ),将向量排列为二维矩阵的形式,每个矩阵元素取值范围为 0~255。

灰度图像是无色彩的记录明亮信息的 2D 图像,灰度指黑白图像中点的颜色深度,范围为 0~255,黑色为 0,白色为 255。因此,由恶意代码文件转化而来的矩阵中每个元素都可以当作灰度图中的像素点,映射成如图 1 所示的无压缩灰度图像。

#### 1.2.2 灰度图纹理特征

尽管反检测技术使得恶意代码产生了大量变种,但变种的恶意代码很大程度上共用了祖先的大部分源代码<sup>[5]</sup>,同一家族的核心代码变化微小,恶意软件在代码结构上大同小异。利用 B2M 算法将不同家族的恶意代码文件转化为对应的灰度图像,结果如图 2 所示。

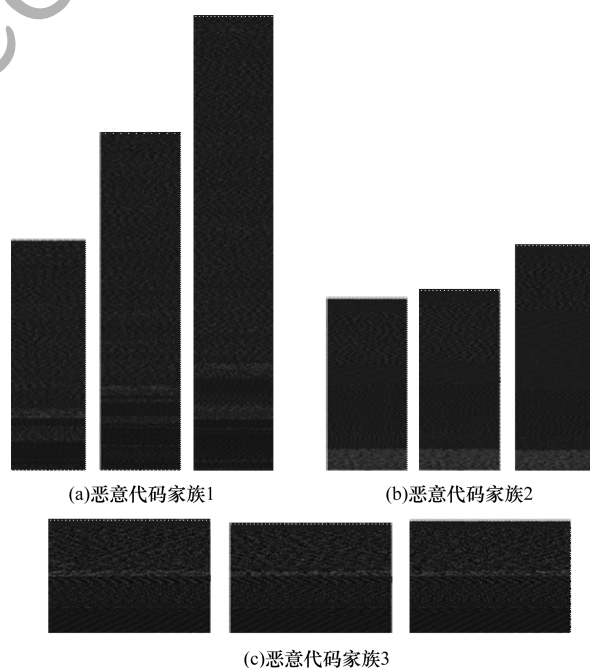


图 2 不同恶意代码家族的图像纹理

可以看出,同一家族的恶意代码图像整体上呈现相似性,不同家族代码呈现不同的纹理结构,有较明显的差别,恶意代码文件转化来的灰度图能够有效取代恶意代码文件本身,成为其另一种表达形式。

恶意代码 PE 文件不同程序段在其转化来的灰度图像中对应不同的纹理表现<sup>[15]</sup>,纹理特征由像素值排列分布反映差别。文献[14]中的图2详细描述了 PE 文件转换为图像后各程序段的对应位置,核心特征操作码所在的代码段(.text)在灰度图像结构的前段位置,本文经反复交叉验证选择一个合适的  $M$  值使其所在位置包含核心代码区,以灰度矩阵  $M$  个像素值的排列向量表征灰度图像的纹理特征,在进行训练时将该二维矩阵转换成一维向量的形式作为恶意代码的特征向量之一。

### 1.3 颜色直方图特征

Opcode N-gram 特征和灰度图纹理特征的前提是忽略文件中其他结构,从不同家族恶意代码的核心特征出发,依据 .text 段包含可执行代码,在已有的恶意代码检测研究中通常只重视核心代码部分而不考虑全局信息<sup>[7-8,10]</sup>,分析 PE 文件结构其他程序段(如 .rsrc)囊括了恶意代码执行模块的所有资源数据,如应用执行时需要的图表、菜单、位图等资源,.data 块包含其他外来 DLL 函数及数据信息、输入表等,不同的程序段携带恶意代码文件的不同信息,这些程序段属于恶意代码的非关键区,非关键区变异虽不会产生新的恶意代码变种,但会使恶意代码的识别更复杂从而导致杀毒软件漏查<sup>[5]</sup>,仅靠单一的局部特征势必造成信息的丢失而导致分类性能受到限制,因此找到一种能够充分表达恶意代码全局信息的方法尤为重要。

在图像领域的研究中,图像颜色特征属于全局特征之一,RGB 颜色空间中 R、G、B 分别代表图像红绿蓝三通道,而灰度图只有一个通道,即传统的 RGB 中的 R、G、B 统一用 Gray 代替,形成新的颜色(Gray, Gray, Gray)。颜色直方图是描述颜色特征的基本表示方法之一,反映图像场景的颜色组成,即图像包含哪些颜色及这些颜色的概率。基于图2中不同恶意代码家族图像像素亮度的分布特性表现出明显的差别,采用颜色直方图作为恶意代码全局特征并验证其在分类中的表现。

颜色直方图在特征提取时首先对颜色空间进行量化,本文将灰度图以像素点为基本单元划分灰度子空间,分割为 0~255 共 256 个像素单元,遍历整幅图像统计落在每个单元像素点的像素数,表达式为:

$$H(K) = \frac{n_k}{N} (K=0,1,\dots,L) \quad (1)$$

其中,  $K$  表示图像像素点取值,  $L$  表示图像特征维数,  $n_k$  表示特征维数  $L$  取值为第  $k$  维时的像素个数,  $N$  是图像像素总数。

## 2 分类模型

在机器学习中分类器分为基于距离和基于决策。基于距离的分类器包括 K 近邻(K Nearest Neighbor, KNN)、支持向量机(Support Vector Machine, SVM)等,基于决策的分类器包括梯度提升分类器(Gradient Boost Classifier, GBC)和随机森林(Random Forest, RF)等,其中 RF 分类器是当前恶意代码检测中的热门算法<sup>[8]</sup>,实际上是一个基于 Bagging 算法的集成模型<sup>[16]</sup>:通过 Bootstrap 方式抽取数据独立训练多个基础模型(决策树(Decision Tree, DT)),由列采样随机使用一定的量特征值训练得到决策树,每棵树的分类能力不强,但在各自领域分类能力很强,多棵决策树组合起来对一个样本从各个角度进行分类,并将这些预测结果通过平均投票的方式给出最终结果,由弱分类器组合成强分类器,其分类性能通常高于单个分类器。

决策树分裂或特征选择的依据是:基尼系数和信息增益。基尼系数用来度量数据的不纯度,其公式为:

$$Gini(D) = 1 - \sum_i^c p_i^2 \quad (2)$$

其中,  $c$  表示数据集类别数量,  $p_i$  表示类别  $i$  的样本数量占有所有样本的比例,数据混合程度越高,基尼系数也越高。若数据集  $D$  被特征  $A$  分成  $k$  个  $D_j$  子集,则分裂后属性  $A$  划分子集的基尼系数为:

$$Gini_A(D) = \sum_{j=1}^k \frac{|D_j|}{|D|} Gini(D_j) \quad (3)$$

基尼系数增益<sup>[16]</sup>为:

$$\Delta Gini(A) = Gini(D) - Gini_A(D) \quad (4)$$

选择基尼系数增益值最大时的特征作为该节点的分裂条件。信息熵评价了信息的混乱程度,熵值越大信息的不确定性越大。节点信息熵为:

$$Info(D) = - \sum_{i=1}^m p_i \lg p_i \quad (5)$$

信息增益的计算公式与基尼系数大致类似,属性  $A$  划分子集的期望信息:

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} Info(D_j) \quad (6)$$

属性  $A$  划分子集的信息增益:

$$Gain(A) = Info(D) - Info_A(D) \quad (7)$$

选择具有最大信息增益的属性作为分裂属性。RF 需要考虑:1) 基于距离的分类器如 SVM、KNN 对具有复杂分布的特征难以进行线性划分;2) RF 可处理多特征的高维数据且不进行特征选择,依赖于 RF 的 2 个采样随机性,即随机有放回地从训练集中提取样本(bootstrap 有放回取样<sup>[16]</sup>)和随机无放回地从特征集中选取特征子集,从  $M$  个特征中选择  $m$  个特征( $m \ll M$ ),在每棵树进行分裂时,从这  $m$  个特征中选择最优特征,且这 2 种随机性可有效减小过拟合程度;3) 模型完成训练后能给出特征

重要性评价,并且在训练过程中能够检测到特征间的影响。在实际训练中特征数量多,各个特征间可能不相关也可能相互依赖,在分类和预测时容易造成模型训练时间长、模型复杂、推广能力下降等问题,了解特征间的影响从而剔除不相关和多余特征,可达到减少特征数量,提高模型精度,减少运行时间的目的<sup>[16]</sup>;4)特征遗失仍可以维持准确度。Opcode N-gram和灰度图纹理特征都是对恶意代码局部的描述,会遗失其他信息,并且各种混淆技术会导致代码不全,但由于 RF 具有模型训练对缺失值不敏感的特点,因此对恶意代码特征有较好的包容性,可对抗部分混淆。

### 3 实验结果与分析

#### 3.1 实验环境

实验软硬件环境如表 1 所示,其中机器学习框架选择 Scikit-learn 是因为其具有广泛可靠的机器学习算法和良好的训练模型。

表 1 实验软硬件环境

| 硬件环境                                                 | 软件环境                                                 |
|------------------------------------------------------|------------------------------------------------------|
| CPU: Intel(R) Xeon(R)<br>E5620 2.4 GHz<br>RAM: 32 GB | 操作系统: Linux 64<br>框架: Scikit-learn<br>语言: Python 3.6 |

#### 3.2 Microsoft 数据集

本文数据集为微软公司在 Kaggle 微软恶意代码分类大赛中提供的数据集<sup>[17]</sup>,Microsoft 数据集包含 9 个恶意代码家族,每个恶意代码去除 PE 头,分别包含 2 个文件,IDA 反编译工具得到的 .asm 文件和 16 进制表示的 .bytes 文件。本文采用反编译工具生成的 .asm 文件,共 10 868 个样本,约 136 GB,详细信息如表 2 所示。

表 2 恶意代码数据集

| 恶意代码家族类别        | 样本数量  |
|-----------------|-------|
| Ramnit          | 1 541 |
| Lollipop        | 2 478 |
| Kelihos_ver3    | 2 942 |
| Vundo           | 475   |
| Simda           | 42    |
| Tracur          | 751   |
| Kelihos_ver1    | 398   |
| Obfuscator. ACY | 1 228 |
| Gatak           | 1 013 |

#### 3.3 单一特征分类性能比较

将 3 种特征的特征向量分别传输到 RF 分类器进行训练,数据集训练时均采用 Cross-Validation 交叉验证以避免陷入局部最小值造成过拟合的情形。随机按比例抽取 60% 数据作为训练集,其余作为测试集。

#### 3.3.1 Opcode N-gram 分类表现

数据集中不同恶意代码文件大小不一,从全部的恶意代码数据集中提取所有操作指令的 N-gram 数量过大,本文对此做了特征选择。文献[18]提出多种特征选择方法,在提取的所有特征中选择少量相关特征组成相关特征集,解决特征集过大的问题,在本文中选取所有恶意代码样本中操作码 N-gram 出现频次高于 500 次作为相关特征并对特征进行降维,分别选取  $N$  值为 {2,3,4,5,6} 进行不同元组的分类准确率对比,准确率公式为:

$$Accuracy = TP/N \quad (8)$$

其中,  $TP$  表示给定的测试数据集中,每一类恶意代码被正确分类的数目总和,  $N$  表示恶意代码测试样本总数。N-gram 不同元组的分类性能结果如图 3 所示。

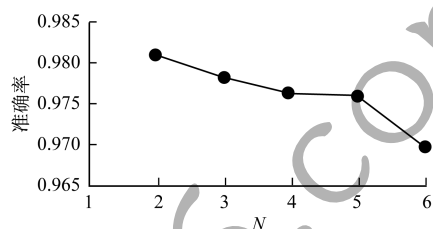


图 3 N-gram 不同元组的分类性能

由图 3 可知,  $N$  在取值为 {2,3,4} 时分类准确率在 97.5% 以上,  $N=2$  时的 Bi-gram 效果达到最佳,平均准确率达到 98.10%,高于 4 元组后分类准确率降低。

#### 3.3.2 灰度图纹理分类表现

本文选取像素点数目  $M$  值为 {1 000, 1 500, 2 000, 2 500, 2 700, 3 000} 进行不同实验的准确率对比,结果如表 3 所示。

表 3 图像纹理特征的分类效果

| 像素点数目 | 准确率     |
|-------|---------|
| 1 000 | 0.809 9 |
| 1 500 | 0.980 8 |
| 2 000 | 0.981 4 |
| 2 500 | 0.986 6 |
| 2 700 | 0.984 4 |
| 3 000 | 0.983 7 |

像素值个数  $M$  取值在 2 000 ~ 3 000 的范围内分类准确率处于平稳状态,最佳取值点是  $M=2 500$  的像素点位置,平均准确率达到 98.66%,可以推测这个范围包含了恶意代码的核心代码区。

#### 3.3.3 颜色直方图分类表现

为每个恶意代码图像构建颜色直方图,经过多次的训练得出平均准确率为 98.66%,与 Opcode N-gram 和灰度图纹理特征有相当的分类性能,此结果表明本文提出的颜色直方图特征可以较好地描述恶

意代码的全局信息并在家族分类上具有较好的表现。

### 3.4 特征融合分类性能比较

本文融合 3 个特征对恶意代码进行局部和全局不同层面的描述,其中各个单一特征的参数取多次训练的最佳点  $\{N=2, M=2\ 500\}$ ,特征向量合并

成新的特征向量输入 RF 分类器,进行多次训练后分类平均准确率达到 99.59%,并且最高准确率达到 99.79%,采用混淆矩阵和对数损失函数对该分类系统性能进行综合评估,混淆矩阵展示了本文分类系统下每个恶意代码家族被误判的情况,如表 4 所示。

表 4 恶意代码家族分类混淆矩阵

| 恶意代码家族类别       | Ramnit | Lollipop | Kelihos_ver3 | Vundo | Simda | Tracur | Kelihos_ver1 | Obfuscator.ACY | Gatak | 误判率      |
|----------------|--------|----------|--------------|-------|-------|--------|--------------|----------------|-------|----------|
| Ramnit         | 604    | 0        | 0            | 1     | 0     | 0      | 0            | 0              | 0     | 0.001 65 |
| Lollipop       | 0      | 999      | 0            | 0     | 0     | 0      | 0            | 0              | 0     | 0.000 00 |
| Kelihos_ver3   | 0      | 0        | 1 189        | 0     | 0     | 0      | 1            | 0              | 0     | 0.000 84 |
| Vundo          | 1      | 0        | 0            | 189   | 0     | 0      | 0            | 0              | 0     | 0.005 26 |
| Simda          | 1      | 0        | 0            | 0     | 16    | 3      | 0            | 0              | 0     | 0.200 00 |
| Tracur         | 6      | 0        | 0            | 0     | 0     | 301    | 0            | 0              | 0     | 0.019 54 |
| Kelihos_ver1   | 1      | 0        | 3            | 0     | 0     | 0      | 166          | 0              | 0     | 0.023 52 |
| Obfuscator.ACY | 0      | 0        | 0            | 0     | 0     | 0      | 0            | 495            | 0     | 0.000 00 |
| Gatak          | 0      | 0        | 0            | 0     | 0     | 0      | 0            | 6              | 366   | 0.016 12 |

对数损失函数是衡量分类算法对于各类别恶意代码是否有较高的分类准确率的函数,定义为:

$$\text{lb loss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \text{lb } p_{ij} \quad (9)$$

其中,  $N$  是恶意代码样本总数,  $M$  是恶意代码类别数,  $y_{ij}$  是样本  $i$  预测为第  $j$  类,预测值与实际相符时为 1,否则为 0,  $p_{ij}$  表示样本  $i$  属于第  $j$  类的概率。

从表 4 可知,组合特征 + RF 算法对恶意代码家族具有较好的区分能力,除了样本数较少的第 5 类,算法对每一类恶意代码家族误判率都较低 ( $<3\%$ )。通过对 9 个类别的样本进行交叉验证,分类平均准确率达到 99.59%,对数损失只有 0.025 33,与 3 个单一特征分类平均准确率相比,平均分类准确率提高了 1.12%,实验结果说明全局和局部特征结合可以有效地对不同恶意代码家族进行特征描述和分类。随机选取该算法的测试结果,混淆矩阵图如图 4 所示。

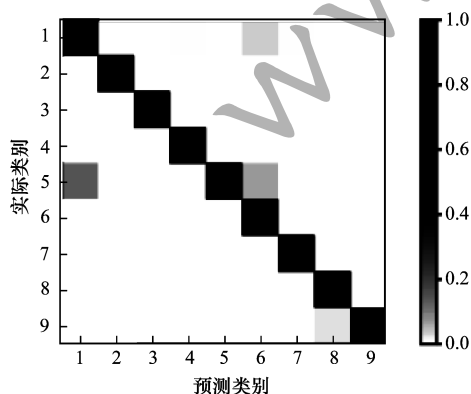


图 4 模型预测类别与实际类别的混淆矩阵

在图 4 中,纵轴代表 9 个恶意代码家族的实际类别,横轴代表预测类别。颜色越深表示预测为此类的概率越大,图中每一类的分类结果大部分都在对角线位置,表明实际类别被预测正确的概率较高,说明应用组合特征 + RF 算法使得每一类恶意代码家族都能被正确预测和归类。

### 3.5 性能比较与评价

为验证 RF、GBC、DT、逻辑回归 (Logistic Regression, LR) 分类器与不同特征组合算法的分类性能,将不同组合的特征向量传输到不同的分类器进行多次训练,结果如表 5 所示,与其他算法的结果对比如表 6 所示。由表 5 可知,RF 分类器在不同组合的特征上都有较好的分类表现,组合特征比单个特征有更好的分类效果。从所有特征组合来看,基于决策的分类器平均准确率高于基于距离的分类器,说明了基于距离的分类器对于分布复杂的特征难以进行线性划分,KNN 分类器在对组合有灰度直方图的特征训练时反而比单个特征分类效果差,说明灰度直方图具有较复杂的分布,主要原因是灰度直方图只取了颜色信息而忽视了位置信息导致特征向量的复杂分布,而 RF 因为 2 个随机性使得其能够处理高维数据且无需做特征选择,对复杂分布的特征集依然能够进行较好的训练和分类,组合特征能够获得更高的准确率。从表 5 和表 6 的对比结果可知,“颜色直方图 + Opcode N-gram + 灰度图纹理 + RF”的组合有最佳的分类效果,证明了本文提出的组合特征能够更加全面描述恶意代码的信息。

表 5 不同分类器在不同组合特征上的分类准确率

| 特征                            | RF      | GBC     | DT      | KNN     | LR      |
|-------------------------------|---------|---------|---------|---------|---------|
| 颜色直方图                         | 0.986 6 | 0.986 1 | 0.977 3 | 0.967 4 | 0.975 8 |
| Opcode N-gram                 | 0.981 0 | 0.982 4 | 0.975 2 | 0.963 7 | 0.968 7 |
| 灰度图纹理                         | 0.986 6 | 0.983 9 | 0.967 4 | 0.959 5 | 0.892 8 |
| 颜色直方图 + Opcode N-gram         | 0.990 4 | 0.992 7 | 0.985 5 | 0.961 7 | 0.895 8 |
| 颜色直方图 + 灰度图纹理                 | 0.994 3 | 0.992 4 | 0.986 7 | 0.957 6 | 0.897 2 |
| Opcode N-gram + 灰度图纹理         | 0.994 5 | 0.995 8 | 0.991 3 | 0.978 1 | 0.988 1 |
| 颜色直方图 + Opcode N-gram + 灰度图纹理 | 0.995 9 | 0.994 7 | 0.992 2 | 0.959 7 | 0.889 7 |
| 平均值                           | 0.992 3 | 0.989 7 | 0.982 2 | 0.963 9 | 0.929 7 |

表 6 算法分类性能对比结果

| 算法      | 特征                            | 分类器 | 准确率     |
|---------|-------------------------------|-----|---------|
| 文献[7]算法 | . bytes 文件 + . asm 文件         | SAE | 0.987 7 |
| 文献[8]算法 | N-gram                        | RF  | 0.886 0 |
| 本文算法    | 颜色直方图 + Opcode N-gram + 灰度图纹理 | RF  | 0.995 6 |

#### 4 结束语

本文从恶意代码反编译文件的核心代码出发,提取了 Opcode N-gram 和灰度图像纹理 2 个局部特征,并提取颜色直方图描述病毒文件的全局特征,算法由特征提取、训练特征集、聚类三部分组成。实验结果表明,融合后的特征可以有效描述恶意代码的局部和全局特征,实现恶意代码的家族分类,从而证明合理提取和选择有效特征是机器学习算法在实际应用中的关键,此外验证了恶意代码转化为图像是一个有效的形式。目前深度学习的推进使得神经网络在图像领域得到快速的发展,通过各级神经元对特征进行层层抽象,无需人为干预即可提取和筛选特征并挖掘隐藏特征,因此下一步将利用深度学习的神经网络对恶意代码的灰度图像进行训练,实现恶意代码特征的自动提取及分类。

#### 参考文献

- [1] LEE J, JEONG K, LEE H. Detecting metamorphic malwares using code graphs[C]//Proceedings of ACM Symposium on Applied Computing. New York, USA: ACM Press, 2010:1970-1977.
- [2] TIAN R, BATTEN L, VERSTEEG S, et al. Function length as a tool for malware classification [C]//Proceedings of the 3rd International Conference on Malicious and Unwanted Software. Washington D. C., USA: IEEE Press, 2008:369-378.
- [3] 王蕊,冯登国,杨轶,等. 基于语义的恶意代码行为特征提取及检测方法[J]. 软件学报, 2012, 23(2): 378-393.
- [4] 朱克楠,尹宝林,冒亚明,等. 基于有效窗口和朴素贝叶斯的恶意代码分类[J]. 计算机研究与发展, 2014, 51(2):373-381.
- [5] 韩晓光,曲武,姚宣霞,等. 基于纹理指纹的恶意代码变种检测方法研究[J]. 通信学报, 2014, 35(8): 125-136.
- [6] ISLAM R, TIAN R H, BATTEN L M, et al. Classification of malware based on integrated static and dynamic features[J]. Journal of Network and Computer Applications, 2013, 36(2):646-656.
- [7] PARK Y, REEVES D, MULUKUTLA V, et al. Fast malware classification by automated behavioral graph matching[C]//Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligent Research. New York, USA: ACM Press, 2010:156-165.
- [8] 王卫红,朱雨辰. 基于 N-Gram 与加权分类器集成的恶意代码检测[J]. 浙江工业大学学报, 2017, 45(6): 604-632.
- [9] 张东,张尧,刘刚,等. 基于机器学习算法的主机恶意代码检测技术研究[J]. 网络与信息安全学报, 2017, 3(7):25-31.
- [10] 任卓君,陈光. 熵可视化方法在恶意代码分类中的应用[J]. 计算机工程, 2017, 43(9):167-171.
- [11] BAT-ERDENE M, PARK H, LI H, et al. Entropy analysis to classify unknown packing algorithms for malware detection [J]. International Journal of Information Security, 2016, 16(3):1-22.
- [12] DING Yuxin, DAI Wei, YAN Shengli, et al. Control flowbased opcode behavior analysis for malware detection[J]. Computers and Security, 2014, 44:65-74.
- [13] OJALA T, PIETIKINEN M, MAENPAA T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7):971-987.
- [14] NATARAJ L, KARTHIKEYAN S, JACOB G, et al. Malware images: visualization and automatic classification [C]//Proceedings of the 8th International Symposium on Visualization for Cyber Security. New York, USA: ACM Press, 2011:4-10.

(上接第 286 页)

- [15] CONTI G, SHUBINA B S. A visual study of primitive binary fragment types [EB/OL]. [2018-05-07]. [https://www.researchgate.net/publication/265356241\\_A\\_Visual\\_Study\\_of\\_Primitive\\_Binary\\_Fragment\\_Types](https://www.researchgate.net/publication/265356241_A_Visual_Study_of_Primitive_Binary_Fragment_Types).
- [16] 崔鸿雁, 徐帅, 张利锋, 等. 机器学习中的特征选择方法研究及展望[J]. 北京邮电大学学报, 2018, 41(1): 1-9.
- [17] Kaggle. Microsoft malware classification challenge

(BIG 2015) [EB/OL]. [2018-05-07]. <https://kaggle.com/c/maleware-classification/>.

- [18] YANG Y, PEDERSEN J O. A comparative study on feature selection in text categorization[C]//Proceedings of International Conference on Machine Learning. New York, USA: ACM Press, 1997: 412-420.

编辑 陆燕菲