软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(5):1454−1464 [doi: 10.13328/j.cnki.jos.005660]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种基于深度森林的恶意代码分类方法*

卢喜东[1], 段哲民[1], 钱叶魁[2], 周 巍[1]

[1](西北工业大学 电子信息学院,陕西 西安 710072)

[2](陆军炮兵防空兵学院 郑州校区,河南 郑州 450052)

通讯作者: 钱叶魁, E-mail: qyk1129@163.com

摘 要: 针对当前恶意代码静态分析方法精度不足的问题,将恶意代码映射为无压缩的灰度图像,然后根据图像变换方法将图像变换为恒定大小的图像,使用方向梯度直方图提取图像的特征,最后提出一种基于深度森林的恶意代码分类方法.实验中选择不同家族的多个恶意代码样本进行分类,验证了该方法的有效性,并且实验结果优于近期提出的 SPAM-GIST 方法.

关键词: 恶意代码分类;方向梯度直方图;深度森林

中图法分类号: TP311

中文引用格式: 卢喜东,段哲民,钱叶魁,周巍.一种基于深度森林的恶意代码分类方法.软件学报,2020,31(5):1454−1464. http://www.jos.org.cn/1000-9825/5660.htm

英文引用格式: Lu XD, Duan ZM, Qian YK, Zhou W. Malicious code classification method based on deep forest. Ruan Jian Xue Bao/Journal of Software, 2020,31(5):1454−1464 (in Chinese). http://www.jos.org.cn/1000-9825/5660.htm

Malicious Code Classification Method Based on Deep Forest

LU Xi-Dong[1], DUAN Zhe-Min[1], QIAN Ye-Kui[2], ZHOU Wei[1]

[1](School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China)

[2](Zhengzhou Campus, PLA Army Artillery Air Defense Academy, Zhengzhou 450052, China)

Abstract: Aiming at the problem of insufficient accuracy of current static classification method of malicious code, this study maps the malicious code into uncompressed gray-scale image. Then the image is transformed into a constant-size image according to the image transformation method, and the direction gradient histogram is used to extract the features of the image. Finally, a kind of malicious code classification method based on deep forest is proposed. Experiments on malicious code samples from different families verify the effectiveness of the proposed method and the results are superior to the recently proposed SPAM-GIST method.

Key words: malicious code classification; histogram of oriented gradient; deep forest

近年来,恶意代码的数量不断增长,严重威胁互联网环境的安全.据赛门铁克(Symantec)报告[1]指出,2010 年恶意代码语料库中总数为 2.86 亿,到 2014 年,恶意代码总数已经达到 17 亿;另据金山火眼(FireEye)于 2013 年 6 月的一项调查显示,2012 年,有 47%的受访组织遭受了恶意软件的威胁[2].可见,恶意代码数量迅速增长且威胁日益严重.

通过将恶意代码映射为图像后,属于同一家族的恶意代码图像具有视觉相似性,属于不同家族的恶意代码

* 基金项目: 西北工业大学研究生创意创新种子基金(ZZ2018020); 国家重点基础研究发展计划(973)(2013CB329104); 通信网信息传输与分发技术国家重点实验室基金

Foundation item: Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University (ZZ2018020); National Program on Key Basic Research Project of China (973) (2013CB329104); Fund of State Key Laboratory of Science and Technology on Information Transmission and Dissemination in Communication Networks

收稿时间: 2017-12-01; 修改时间: 2018-05-05, 2018-06-01; 采用时间: 2018-08-09

图像之间具有一定的差异性[3].根据恶意代码图像的这一特点,在实验中,可以提取恶意代码图像特征,然后使用分类算法对提取的特征进行分类.例如,Nataraj 等人[3−5]利用 Gabor 滤波器提取恶意代码图像的全局特征,该滤波器多尺度、多方向的特点能够提取图像的全局特征,最后使用最近邻算法对恶意代码进行分类.Kancherla 等人[6]将利用 Gabor 滤波器提取的恶意代码图像的全局特征与恶意代码图像的小波特征和强度特征结合在一起,然后使用结合在一起的特征对恶意代码进行分类.Han 等人[7]将恶意代码转化为图像,然后计算图像的熵值并映射到曲线上,最后,通过计算曲线的相似度来对恶意代码分类.Farrokhmanesh 等人[8]使用 3 种不同的哈希算法提取恶意代码灰度图的指纹;然后,计算图像指纹之间的距离对恶意代码分类.但是,这些方法不能解决恶意代码二进制文件重新分配的问题,并且在分类精度还有上升的空间.

针对上述问题,本文提出一种基于深度森林(deep forest,简称 DF)[9]的恶意代码分类方法(malicious code classification method based on deep forest,简称 MCDF),利用特征分类技术与图像特征提取技术,将恶意代码映射为灰度图像,并利用缩放算法将所有图像变换为统一的尺寸,使用 Gamma 校正法将图像标准化,计算图像的梯度并构建方向梯度直方图(histogram of oriented gradient,简称 HOG)[10];然后将块内方向梯度直方图进行归一化处理,收集所有块内的 HOG 特征结合成为最终 HOG 特征向量;最后将 HOG 特征向量转化为二维 HOG 特征矩阵,使用深度森林算法对二维 HOG 特征矩阵分类.

本文的主要工作包括以下 3 点.

(1) 本文提出了通过提取恶意代码图像 HOG 特征来对恶意代码进行分类的方法.从恶意代码二进制文件映射到 HOG 特征矩阵,然后使用特征匹配算法对恶意代码分类,该方法在一定程度上可以确定未知恶意代码的类别.

(2) 提出深度森林的恶意代码分类方法,利用深度森林不同大小的窗口扫描 HOG 特征矩阵,在一定程度上可以解决二进制文件重新分配的问题.

(3) 利用 7 组恶意代码数据集对本文方法进行实验,验证分类的精确性和有效性.

1 相关工作

恶意代码特征提取及分类方法是本文的主要研究工作.首先需要对恶意代码进行分析,分析恶意代码具有的特征;然后研究恶意代码二进制文件的特征提取方法,使提取的特征能够尽量表征恶意代码的本质,最后基于提取的特征对恶意代码进行分类.因此,本文的相关研究工作主要有恶意代码分析、恶意代码特征提取、恶意代码分类方法这 3 个方面.

动态分析和静态分析为恶意代码分析领域中普遍采用的两种方法.动态分析所分析的代码就是实际执行的代码,在代码运行过程中进行分析,每次运行只能得到 1 条路径的行为,但是,一些恶意代码存在多条执行路径.如 Nataraj 等人[4]采用法医对照比较的方法,通过比较感染前和感染后系统的组合特征来分类恶意代码,改善了动态分析方法的单一执行路径问题.毛蔚轩等人[11]提出了一种主动学习的方法,利用很少的样本即可对未知样本进行检测.静态分析首先对可执行程序进行反汇编,并在此基础上提取代码的特征信息.例如,Wang 等人[12]提取操作码序列,并将操作码序列映射为图像,进而对恶意代码进行检测.该方法在检测速度和检测精度上取得了良好的效果,但易受代码混淆技术的干扰.Farrokhmanesh 等人[13]提出利用音频的方法分析恶意代码,该方法将恶意代码二进制文件转化为音频,然后从音频中提取恶意代码特征对恶意代码进行检测.该方法在检测速度和检测精度上取得了良好的效果,但不能解决垃圾代码插入的问题.Nataraj 等人[3,14]提出利用信号处理的方法分析恶意代码,该方法将不同家族恶意代码二进制文件转化为相等长度的向量,通过随机投影降维来检测恶意代码.该方法解决了代码顺序变换的问题,但易受恶意代码二进制文件重新分配的干扰.

特征的提取能力直接影响分类精度和分类效率,如何降低混淆技术的干扰,提取到恶意代码的本质特征,对恶意代码变种进行准确分类,是恶意代码研究的热点.将恶意代码映射为图像和信号并提取图像和信号的特征是一种新颖的特征提取方法,如 Nataraj 等人[3,14]将反汇编的恶意代码二进制文件映射为稀疏的信号,并从信号中提取恶意代码特征,解决了代码顺序变换的问题,但易受恶意代码二进制文件重新分配的干扰.韩晓光等人[15]

将恶意代码二进制文件映射为灰度图,通过提取图像的纹理指纹来描述恶意代码的特征.该方法可以有效地避免反逆向逻辑、反追踪以及其他常见的代码混淆方法,但对于加壳、混淆的恶意代码没有很好的效果.Han 等人[7]将反汇编的可执行文件映射为灰度图,然后计算图像每行像素的熵值并将熵值投影到曲线上,进而对恶意代码进行分类.这类算法虽然有很高的准确率,但是该方法不能解决反汇编文件重新分配或者添加大量冗余的问题.Zhang 等人[16]通过提取汇编码序列,并将汇编码序列转化为图像对恶意代码进行分类.该方法在精度上取得了良好的效果,但易受代码混淆技术的干扰.Arefkhani 等人[8]通过使用 3 种不同的哈希算法对恶意代码进行分类,利用哈希算法提取图像特征,其中,使用感知哈希算法取得了很好的分类效果,而另外两种方法在分类精度上不如感知哈希算法.

恶意代码分类是指将新出现的恶意代码划归到已知的恶意代码类型中,分类的主要工作就是分类算法的构建.传统的恶意代码分类方法大多使用机器学习算法,例如,Nataraj 等人[3−5]使用 K 近邻(K-nearest neighborhood,简称 KNN)算法对恶意代码进行分类,该算法是一种消极学习法,所以可以为不同的待分类实例构建不同的逼近函数.Wang 等人[12]将操作码序列映射为图像,然后使用支持向量机(support vector machine,简称 SVM)算法对恶意代码进行分类,该算法可以选择不同的核函数,生成不同的支持向量机.Yang 等人[17]利用随机森林(random forest,简称 RF)算法分类恶意代码,该算法是由一组决策树分类器(decision tree,简称 DT)组成的集成分类器,通过决策树的投票决定分类结果.近年来,研究者开始使用深度学习对恶意代码进行分类.如 Zhang 等人[16]利用卷积神经网络(convolutional neural network,简称 CNN)对经过处理得到的操作码序列图像分类,该方法用来处理具有网格结构的数据,例如时间序列数据和图像等.Davis 等人[18]使用循环神经网络(recurrent neural network,简称 RNN)分析恶意代码,该算法用来处理具有序列结构的数据.David 等人[19]将恶意代码动态行为转化维二值向量,使用深度置信网络(deep belief network,简称 DBN)对恶意代码进行分类,该算法是建立一个观察数据与标签之间联合分布的概率生成模型.

本文提出的基于深度森林的恶意代码分类方法采用不同恶意代码家族反汇编后的二进制文件作为分类样本集,该方法结合了方向梯度直方图的特征提取能力和深度森林算法的分类能力,充分利用了恶意代码图像的空间相似性.此外,通过实验证实了本文提出的方法优于近期提出的 SPAM-GIST 方法[3].

2 MCDF 方法

本文提出的基于深度森林的恶意代码分类方法主要分为 3 个阶段:第 1 阶段为数据预处理阶段,包含第 2.1 节和第 2.2 节;第 2 阶段为 HOG 特征提取阶段;第 3 阶段为恶意代码分类阶段.首先需要将恶意代码二进制文件映射为无压缩的灰度图像,然后根据图像变换方法将图像变换为恒定大小的图像,使用方向梯度直方图提取图像的特征,最后利用深度森林算法对恶意代码进行分类.

2.1 恶意代码映射方法描述

为了提取基于图像的恶意代码特征,我们需要将恶意代码二进制文件映射为图片.对于给定的恶意代码二进制文件,读取 8 位二进制数转化为十进制整型(范围为 0~255),并将这些整型重塑为固定行宽的向量,最后生成一个二维数组,此二维数组的宽度和高度根据文件大小的不同而不同.最后将此数组可视化为一个灰度图像.映射后的图像被保存为无压缩的 PNG 图片.恶意代码映射为图像的流程如图 1 所示.不同恶意代码家族的恶意代码图片如图 2 所示.

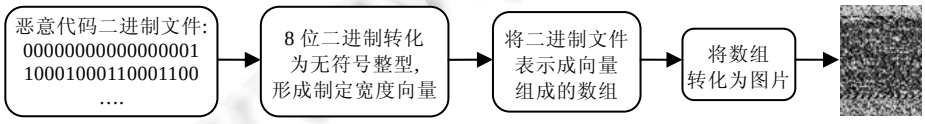
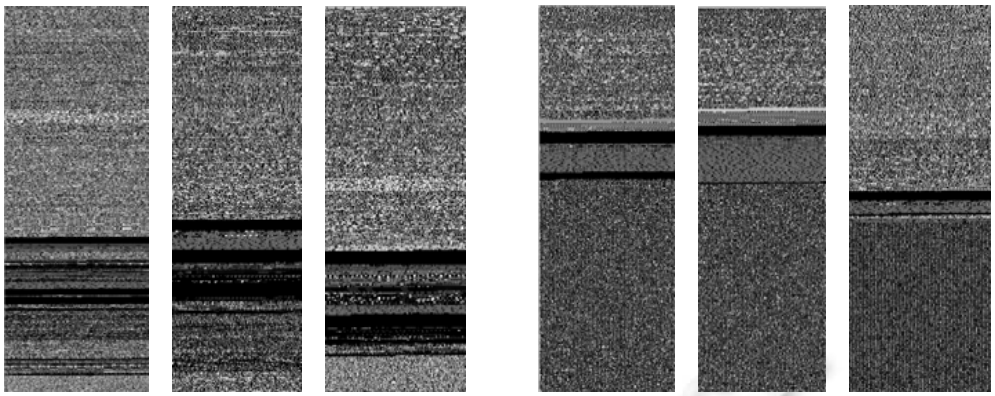


Fig.1 Malicious code mapping flow chart

图 1 恶意代码映射流程图



(a) Ramniit (b) Obfuscator_ACY

Fig.2 Malicious code images of different families

图 2 不同恶意代码家族的恶意代码图片

2.2 基于插值的图像缩放

为了提取维数相等的 HOG 特征向量,并且能够得到图像的全局特征,使用图像插值方法将所有恶意代码图像变换为固定大小的尺寸.为了能够尽量保留恶意代码图像的特征,使缩放后的图像具有较高的图像质量,在本文中采用双三次插值算法对恶意代码图像进行缩放.该算法选取恶意代码图像待插值点周围直接相邻的 4 个点和邻近的 12 个点,并使用基函数对此 16 个点的灰度值做三次插值运算.该算法需要选取插值基函数来拟合数据,通过如下公式构造基函数.

$$W(x)=\begin{cases}(a+2)|x|^3-(a+3)|x|^2+1, & |x|\leqslant 1\\ a|x|^3+5a|x|^2+8a|x|-4a, & 1<|x|<2\\ 0, & \text{其他}\end{cases} \tag{1}$$

双三次插值公式如下.

$$f(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3}f(x_i,y_j)W(x-x_i)W(y-y_j) \tag{2}$$

其中,(x,y)为恶意代码图像中待插值的像素点,$(x_i,y_i)(i,j=0,1,2,3)$为该像素点 4×4 邻域点.

2.3 HOG特征提取

方向梯度直方图特征是用于物体检测的特征描述子,该方法从像素级别生成图像的梯度,然后将图像分割为多个细胞单元并统计每个细胞单元的梯度直方图,结合多个细胞单元形成块并计算每个块的 HOG 特征,最后收集所有块的特征组成图像的 HOG 特征.在 HOG 的计算过程中,采用由像素点到块、由块到 HOG 的特征提取方法可以很好地捕捉到图像的信息.此外,在计算梯度直方图时,只考虑各个像素点的梯度大小和方向而不考虑像素点间的位置关系,这样可以在一定程度上解决二进制文件重新分配的问题.

(1) Gamma 校正法标准化

为了提高恶意代码图像的对比度,同时抑制噪音的干扰,本文采用 Gamma 校正法对恶意代码图像进行颜色空间的标准化.Gamma 校正公式如下.

$$I(x,y)=I(x,y)^{Gamma} \tag{3}$$

(2) 图像梯度的计算

计算图像横坐标和纵坐标方向的梯度,并据此计算每个像素的梯度幅值和梯度方向.图像中像素点(x,y)的梯度为

$$G_x(x,y)=H(x+1,y)-H(x-1,y) \tag{4}$$

$$G_y(x,y)=H(x,y+1)-H(x,y-1) \tag{5}$$

其中,$H(x,y)$,$G_y(x,y)$,$G_x(x,y)$分别表示输入恶意代码图像中像素点(x,y)处的像素点坐标、垂直方法梯度和水平方向梯度.在本文中,使用[–1,0,1]梯度算子在恶意代码图像 x 方向进行卷积计算,使用$[1,0,-1]^T$梯度算子在 y 方向做卷积运算,分别得到恶意代码图像 x 方向梯度分量 $G_x(x,y)$和 y 方向的梯度分量 $G_y(x,y)$.然后,使用如下公式计算恶意代码图像的梯度幅值 $G(x,y)$和梯度方向$\alpha(x,y)$.

$$G(x,y)=\sqrt{G_x(x,y)^2+G_y(x,y)^2} \tag{6}$$

$$\alpha(x,y)=\tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \tag{7}$$

(3) 梯度方向直方图的构建

为了构建恶意代码图像梯度方向直方图,我们将图像分割成 $n\times n$ 个像素组成的细胞单元,采用 bin 个方向的直方图统计每个细胞单元的梯度信息,然后将细胞单元内每个像素的梯度幅值用梯度方向映射到直方图中固定的角度范围,即可形成每个细胞单元的恶意代码特征描述符.

(4) 块内归一化梯度直方图

结合空间上连通的几个细胞单元组合成块,将一个块内所有细胞单元的特征向量按一定方式串联起来,就得到该块的 HOG 特征,然后将每个块的 HOG 特征向量进行归一化.归一化公式如下.

$$V'=\frac{V}{\|V\|_2} \tag{8}$$

其中,V 和 V'为归一化前和归一化后 HOG 特征向量,$\|V\|_2$ 为特征向量 V 的 l_2-范数.由于这些块是互有重叠的,这就意味着每一个细胞单元的特征会以不同的结果多次出现在最后的特征向量中.我们将归一化后的块描述符称为恶意代码 HOG 描述符.

(5) HOG 特征收集

最后,收集恶意代码图像中所有块的 HOG 特征 V',结合成最终的特征向量 $f=\{V_1',V_2',...,V_n'\}$.图 3 所示为 HOG 特征提取流程.

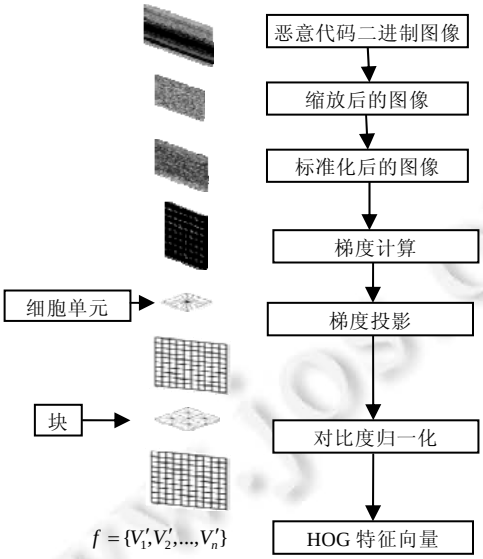


Fig.3 HOG feature extraction process

图 3 HOG 特征提取流程

2.4 恶意代码分类

前面我们已经提取恶意代码图像的HOG特征,现在需要通过算法对恶意代码进行分类,接下来我们分两部

分描述深度森林算法对恶意代码的分类过程.

- 第 1 部分从决策树的角度描述随机森林生成的过程,随机森林为组成深度森林的基本结构.
- 第 2 部分描述深度森林算法对恶意代码分类的具体过程.

(1) 随机森林生成

随机森林是由一组决策树分类器$\{h(X,\theta_k),k=1,\ldots,K\}$组成的集成分类器,其中,$K$ 表示随机森林中决策树的个数,$\{\theta_k\}$是服从独立同分布的随机向量.

生成随机森林的步骤如下.

1) 应用 bootstrap 方法有放回的从训练样本集中随机抽取 K 个新的自助样本集,并将未被抽到的样本组成 K 个袋外数据,然后使用抽到的样本构建 K 棵决策树.
2) 设有 n 个特征,则在每棵树的每个节点处随机抽取 $m(m\leqslant n)$个候选特征,并计算每个候选特征的基尼指数(*Gini_index*),然后,选择具有最小基尼指数值的特征进行节点分裂.
3) 当每棵树中的节点中只有一个类别或节点中的样本数小于最小分裂级数时,停止生长.
4) 将生成的 K 棵树组成随机森林,新的数据使用该随机森林进行分类,并将决策树的投票结果作为分类结果.

其中,基尼指数(*Gini_index*)的计算方法如下.

$$Gini(D)=1-\sum_{k=1}^{|y|}p_k^2 \tag{9}$$

$$Gini\_index(D,a)=\sum_{V=1}^{V}\frac{|D^V|}{|D|}Gini(D^V) \tag{10}$$

其中,

- 公式(9)为基尼值的计算方法,其中,$|y|$和 p_k 为数据集 D 的类别数量和每个类别占总数据集的比例.
- 公式(10)中,$Gini(D^V),|D|,|D^V|,a$ 分别为类别 V 的基尼值、样本总数、类别 V 的样本总数以及需要划分的特征.

完全随机森林生成过程与随机森林生成过程的第 2 步稍有差别,完全随机森林生成过程中每棵树随机选取一个特征作为分裂树的分裂节点,其他步骤和随机森林生成过程一样.

(2) 分类方法

深度森林是由多粒度扫描和级联森林两部分组成的多层结构,每层由相等数量的随机森林和完全随机森林组成.本文中随机森林在进行节点分裂时每次随机选取 $\sqrt{d}$ (d 为特征维数)个特征作为候选特征.

接下来,将恶意代码 HOG 特征向量转化为矩阵,并且分为训练集和测试集:训练集用于训练深度森林,测试集用来评估深度森林.

如图 4 所示,多粒度扫描阶段,使用滑动窗口 $W\times W$ 和扫描步长 S 扫描恶意代码训练集样本,将每个窗口中的特征作为多粒度扫描森林的输入,输出每个窗口中特征的概率向量;然后,将所有输出的概率向量按一定方式串联作为级联森林的输入.

如图 5 所示,级联森林的第 1 级将多粒度扫描的输出结果作为输入,在此后的每一级都将多粒度扫描后的概率向量和前一级级联森林的输出作为下一级的输入.在此过程中,级联森林每增加一级,就使用测试集样本对已经生成的整个深度森林进行测试,如果测试集的准确率小于前一层的准确率,深度森林停止生长,级联层不在增加;否则,继续增加级联层,直到测试集的准确率小于前一层的准确率为止.最后一级,对所有输出的概率向量取均值,输出具有最大概率那一类的标签,作为最终恶意代码预测结果.

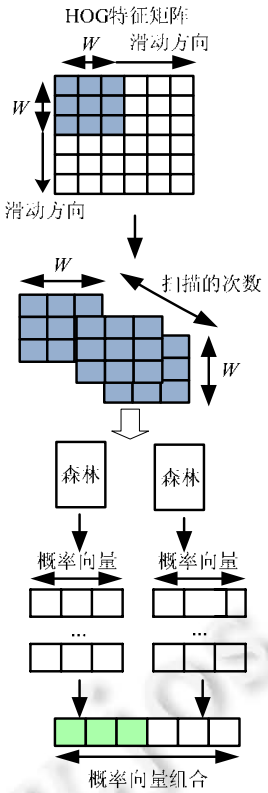


Fig.4 Multi-grained scanning

图 4 多粒度扫描

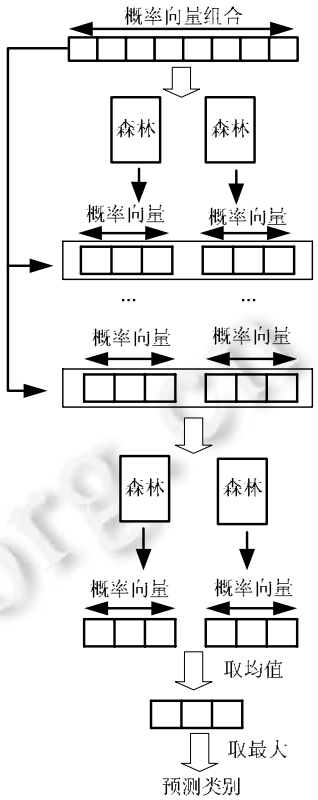


Fig.5 Cascade forest structure

图 5 级联森林结构

3 实验评价

3.1 实验环境和实验数据集

基于深度森林的恶意代码分类系统,运行的 CPU 为 Intel(R) Core(TM) i7-6800K 双核处理器,配置两块 4GB DDR3L 内存;操作系统使用 64 位 Windows10.恶意代码映射、HOG 特征提取以及分类使用 Python 语言,相关包为 Anaconda3-4.3.1-Windows-x86_64,包含实验过程中所有用到的包.

本文使用的恶意代码数据集来自于 Microsoft 在 Kaggle 上的项目——Microsoft Malware Classification Challenge[20].本文选取 7 个类别的 9 929 条恶意代码二进制文件进行实验,基本信息见表 1.

Table 1 Malicious code dataset

表 1 恶意代码数据集

恶意代码类别	类别号	数量
Ramniit	0	1 513
Lollipop	1	2 470
Kelihos_ver3	2	2 936
Vundo	3	446
Kelihos_ver1	4	387
Obfuscator_ACY	5	1 166
Gatak	6	1 011

3.2 评价标准

本文采用准确率(*accuracy*)、宏 *F*1(*macro-F*1)两种评价指标评价深度森林算法对恶意代码的分类效果,计算公式如下.

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

$$P=\frac{TP}{TP+FP} \tag{12}$$

$$R=\frac{TP}{TP+FN} \tag{13}$$

$$macro\text{-}P=\frac{1}{n}\sum_{i=1}^{n}P_i \tag{14}$$

$$macro\text{-}R=\frac{1}{n}\sum_{i=1}^{n}R_i \tag{15}$$

$$macro\text{-}F1=\frac{2\times macro\text{-}P\times macro\text{-}R}{macro\text{-}P+macro\text{-}R} \tag{16}$$

其中,*TP*,*FP*,*FN*,*TN* 分别表示被分类器识别为正的正样本、被分类器识别为正的负样本、被分类器识别为负的正样本、被分类器识别为负的负样本.*macro-P*,*macro-R* 为宏查准率和宏查全率.

3.3 实验结果分析

在实验中,我们选用数据集的 80%作为训练集,20%作为测试集.在本文中,将恶意代码图像缩放为 128×128 个像素的图像,每个细胞单元为 16×16 个像素,采用 8 个方向的直方图来统计这 16×16 个像素的梯度信息,然后将每 4 个单元格形成一个大的、连通的块,形成 1 568 维的 HOG 特征向量,最后将特征向量转化为 49×32 的二维矩阵.其中,在数据处理阶段的 Gamma 值取 0.5.

(1) 方法比较

我们使用我们的 MCDF 方法与 SPAM-GIST[3]进行了比较.在 MCDF 中,多粒度扫描和级联结构都包含两个森林、每个森林包含 1 000 棵树以及每棵树的最小分裂级数为 10,扫描阶段窗口大小为 32×32、步长 1.在 SPAM-GIST 实验中,*K* 近邻(*K*-nearest neighbor,简称 KNN)分类算法的 *K*=3,采用 10 折交叉验证(tenfold cross validation)估计每个样本的类别.此外,使用 10 折交叉验证的 KNN 分类算法对 HOG 特征进行分类(MCKNN),并使用深度森林算法对文献[3]中 320 维的 GIST 特征进行分类(GIST-DF),其中,对 GIST 特征进行分类时,将 320 维特征向量转化为 20×16 的二维矩阵,深度森林扫描阶段窗口大小为 16×16,其他参数与 MCDF 实验相同.在本文中,4 组实验分别进行 10 次,取最好的结果作为最终结果,实验结果见表 2.

Table 2 Experimental results of MCDF and SPAM-GIST

表 2 MCDF 与 SPAM-GIST 实验结果

实验名称	分类器	*Accuracy* (%)	*macro-F*1 (%)
MCDF	深度森林	96.0	95.2
MCKNN	KNN	93.4	91.3
SPAM-GIST	KNN	95.4	94.4
GIST-DF	深度森林	95.6	95.1

(2) 分类器对比

为验证深度森林的分类性能,本文使用深度学习方法人工神经网络(artificial neural network,简称 ANN)和传统的机器学习方法随机森林(random forest,简称 RF)对本文提取的 HOG 特征向量进行分类.其中,ANN 包含 3 个隐藏层,每个隐藏层包含的节点个数为 1 000、500、100.随机森林由 1 000 棵决策树组成.实验结果见表 3.

Table 3 Experimental results of different classifiers

表 3 不同分类器的实验结果

分类器	*Accuracy* (%)	*macro-F*1 (%)
深度森林	96.0	95.2
ANN	94.5	92.1
RF	93.5	90.4

由表 3 可知,深度森林的分类性能优于 ANN 和 RF.

(3) 参数敏感性分析

为了验证深度森林不同参数对实验结果的影响,本文通过改变深度森林中多粒度扫描结构中窗口大小(*windows*)以及每个森林中树的数量(*mgsRFTree*)和最小分裂级数(*msmgs*)、级联结构中的森林数量(*csdRF*)和每个森林中树的数量(*csdRFTree*)和最小分裂级数(*mscsd*)对本文中的算法进行评价,每个参数设定下做 10 组实验,并对实验结果取平均值作为最终结果,实验结果如图 6 所示.

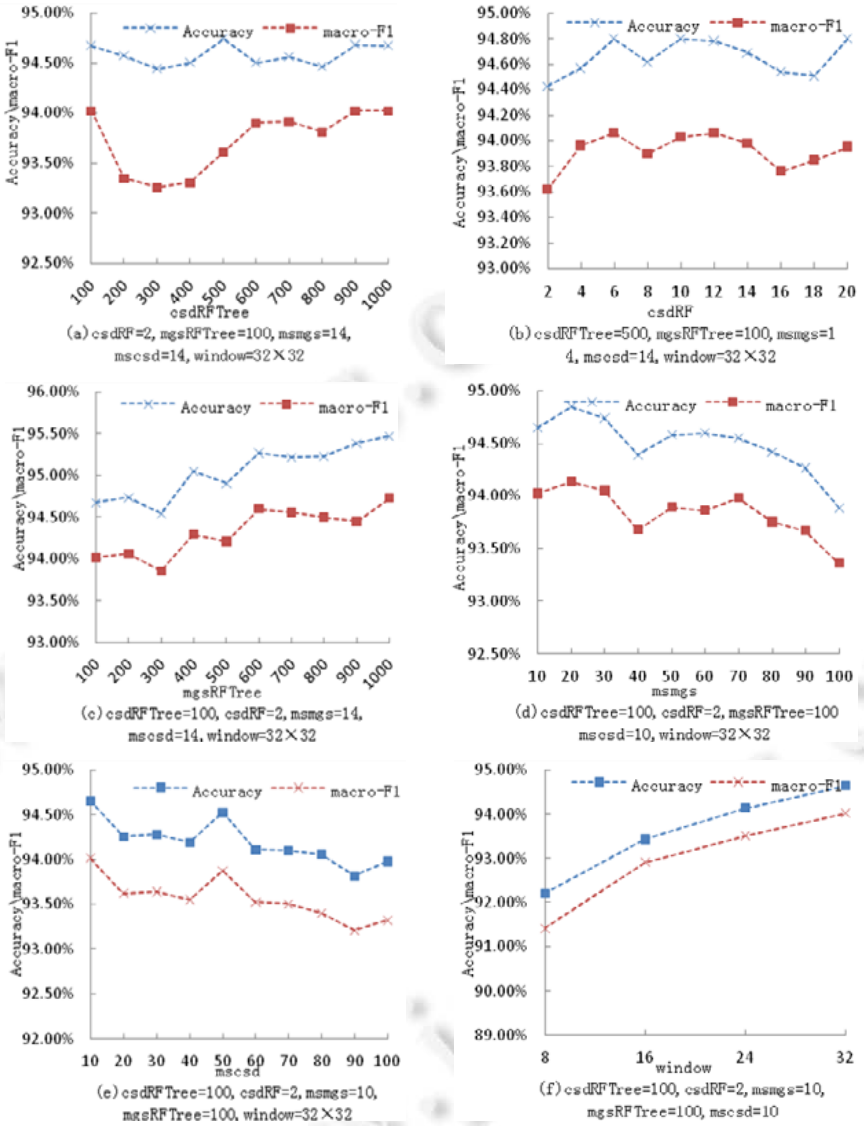


Fig.6 Experimental results of different parameters

图 6 不同参数实验结果

由图 6(a)、图 6(b)可以看出,当改变参数 *csdRFTree*、*csdRF* 时,对实验结果几乎没有影响;由图 6(c)、图 6(f)可以看出,当 *mgsRFTree*、*window* 增加时,算法的整体分类性能变得越来越好;由图 6(d)、图 6(e)可以看出,随着 *msmgs*、*mscsd* 的增加,算法的整体分类性能变得越来越差.

3.4 讨 论

实验中,选择多个家族的恶意代码样本进行测试,验证了本文提出的分类方法的有效性.本文通过提取恶意代码图像的 HOG 特征来对恶意代码进行分类.该特征的提取过程中,首先从像素级别计算图像的梯度,然后在细胞单元中统计梯度直方图,在块中计算每个块的 HOG 特征,最后收集所有块的特征组成图像的 HOG 特征.这种由点到局部、由局部到整体的方法可以很好地捕捉到图像的信息,在统计细胞单元内的梯度直方图时,只考虑各个像素点的梯度大小和方向而不考虑像素点间的位置关系,这样可以在一定程度上解决二进制文件重新分配的问题.此外,HOG 特征是由各个块中的局部特征组成,因此,HOG 特征向量中隐含了该块的空间位置关系,当深度森林的扫描窗口对特征扫描时,每个窗口中包含多个块内的特征,每个 HOG 特征向量分割为多个片段,这样可以提高表征学习能力和注意到更长的上下文信息,提高分类准确率.

4 总 结

本文提出了一种基于深度森林的恶意代码分类方法,通过结合图像分析技术与恶意代码分类技术,将恶意代码二进制文件映射为无压缩的灰度图片,然后根据图像变换方法将图像变换为恒定大小的图像,使用方向梯度直方图提取图像的特征,并将这些特征作为恶意代码的本质特征,最后使用深度森林算法对恶意代码进行分类.实验中,对不同家族的恶意代码样本进行测试,验证了本文提出方法的有效性,并且优于近期提出的 SPAM-GIST 方法.在下一步工作中,我们将利用深度学习对恶意代码进行分析,进一步提高分类精度.

References:

[1] Corporation S. Internet security threat report 2013. 2014. https://www.facebook.com/Symantec

[2] InfoRiskToday. The need for speed: 2013 incident response survey, FireEye. 2016. http://www.inforisktoday.in/surveys/2013-incident-response-survey-s-18

[3] Nataraj L, Manjunath BS. SPAM: Signal processing to analyze malware. IEEE Signal Processing Magazine, 2016,33(2):105−117.

[4] Nataraj L, Yegneswaran V, Porras P, *et al*. A comparative assessment of malware classification using binary texture analysis and dynamic analysis. In: Proc. of the ACM Workshop on Security and Artificial Intelligence. ACM, 2011. 21−30.

[5] Nataraj L, Karthikeyan S, Jacob G, *et al*. Malware images: Visualization and automatic classification. In: Proc. of the Int'l Symp. on Visualization for Cyber Security. ACM, 2011. 1−7.

[6] Kancherla K, Mukkamala S. Image visualization based malware detection. In: Proc. of the Computational Intelligence in Cyber Security. IEEE, 2013. 40−44.

[7] Han KS, Lim JH, Kang B, *et al*. Malware analysis using visualized images and entropy graphs. Int'l Journal of Information Security, 2015,14(1):1−14.

[8] Arefkhani M, Soryani M. Malware clustering using image processing hashes. In: Proc. of the Machine Vision and Image Processing. IEEE, 2016. 214−218.

[9] Zhou ZH, Feng J. Deep forest: Towards an alternative to deep neural networks. 2017. http://xueshu.baidu.com/usercenter/paper/show?paperid=637f02600a538dc721ff4c3213ce2b7a&site=xueshu_se&hitarticle=1

[10] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2005). IEEE, 2005. 886−893.

[11] Mao WX, Cai ZM, Tong L. Malware detection method based on active learning. Ruan Jian Xue Bao/Journal of Software, 2017, 28(2):384−397 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5061.htm [doi: 10.13328/j.cnki.jos.005061]

[12] Wang T, Xu N. Malware variants detection based on opcode image recognition in small training set. In: Proc. of the IEEE Int'l Conf. on Cloud Computing and Big Data Analysis. IEEE, 2017.

[13] Farrokhmanesh M, Hamzeh A. A novel method for malware detection using audio signal processing techniques. In: Proc. of the Artificial Intelligence and Robotics. IEEE, 2016. 85−91.

[14] Nataraj L, Karthikeyan S, Manjunath BS. SATTVA: SpArsiTy inspired classificaTion of malware VAriants. In: Proc. of the ACM Workshop on Information Hiding & Multimedia Security. ACM, 2015.

[15] Han XG, Qu W, Yao XX, *et al*. Research on malicious code variants detection based on texture fingerprint. Journal on Communications, 2014,35(8):125−136 (in Chinese with English abstract).

[16] Zhang J, Qin Z, Yin H, *et al*. IRMD: Malware variant detection using opcode image recognition. In: Proc. of the IEEE Int'l Conf. on Parallel and Distributed Systems. IEEE, 2017. 1175−1180.

[17] Yang M, Wen Q. Detecting Android malware by applying classification techniques on images patterns. In: Proc. of the IEEE Int'l Conf. on Cloud Computing and Big Data Analysis. IEEE, 2017. 344−347.

[18] Davis A, Wolff M, Soeder DA, *et al*. Recurrent neural networks for malware analysis. U.S. Patent 9,495,633, 2016-11-15.

[19] David OE, Netanyahu NS. DeepSign: Deep learning for automatic malware signature generation and classification. In: Proc. of the 2015 Int'l Joint Conf. on Neural Networks (IJCNN). IEEE, 2015. 1−8.

[20] BIG 2015. http://www.kaggle.com/c/malware-classification

附中文参考文献:

[11] 毛蔚轩,蔡忠闽,童力.一种基于主动学习的恶意代码检测方法.软件学报,2017,28(2):384−397. http://www.jos.org.cn/1000-9825/5061.htm [doi: 10.13328/j.cnki.jos.005061]

[15] 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.通信学报,2014,35(8):125−136.

卢喜东(1991－),男,陕西靖边人,硕士,主要研究领域为恶意代码分析,机器学习.

钱叶魁(1980－),男,博士,副教授,主要研究领域为网络信息安全.

段哲民(1953－),男,博士,教授,博士生导师,主要研究领域为数据采集,信号处理.

周巍(1979－),男,博士,副教授,CCF 专业会员,主要研究领域为视频压缩及其 VLSI 设计.