基于多特征随机森林的恶意代码检测

李劭杰[1,2] 王 晨[2] 史 崄[2]

[1](武汉邮电科学研究院 湖北 武汉 430074)

[2](南京烽火软件股份有限公司 江苏 南京 210000)

摘 要 特征提取是恶意代码检测研究的重点内容,传统恶意代码特征提取以单一特征为主,恶意代码检测时效性差,特征提取滞后于病毒数量的发展。恶意软件源文件经过 IDA 反编译后生成. bytes 文件和. asm 文件,. asm 文件可以从两个角度提取特征。通过 N-Gram 算法提取文本特征,可以将. asm 文件图像化转化成为灰度图像。灰度图像的纹理特征通过灰度共生矩阵的不同参数来体现,颜色特征作为全局特征通过灰度直方图提取,最终结合随机森林算法进行分类。实验结果表明,多种特征相结合的检测方法,能极大程度地提高检验的准确率。

关键词 灰度图 灰度直方图 灰度共生矩阵 N-Gram 算法 随机森林

中图分类号 TP3 **文献标志码** A DOI:10.3969/j. issn. 1000-386x. 2020. 10. 053

MALICIOUS CODE DETECTION BASED ON MULTI-FEATURE RANDOM FOREST

Li Shaojie[1,2] Wang Chen[2] Shi Yin[2]

[1](*Wuhan Institute of Posts and Telecommunications*, *Wuhan 430074*, *Hubei*, *China*)

[2](*Nanjing Fiberhome Software Co.*, *Ltd.*, *Nanjing 210000*, *Jiangsu*, *China*)

Abstract Feature extraction is the focus of malicious code detection research. The traditional malicious code feature extraction is dominated by a single feature, which results in poor timeliness of malicious code detection. Feature extraction lags behind the development of virus number. The malware source file is decompiled by IDA to generate . bytes file and . asm file. The . asm file can extract features from two perspectives. The N-Gram algorithm was used to extract the text features. The . asm file was transformed into grayscale image. The texture features of grayscale image were reflected by different parameters of graylevel co-occurrence matrix, and the color features were extracted by gray histogram as global features. Finally, the classification was carried out by combining the random forest algorithm. Experimental results show that the detection method based on multi-feature can greatly improve the accuracy of the test.

Keywords Gray scale image Gray histogram Gray level co-occurrence matrix N-Gram algorithm Random forest

0 引 言

飞速发展的互联网技术推动人类社会不断进步,然而互联网技术的不断革新也促进了恶意代码的发展壮大,其危害从原来的计算机虚拟世界逐渐延伸到现实生活中。《2018 年我国互联网网络安全态势报告》显示[1],网络安全问题越来越尖锐,目前的网络安全问题已经从高技术人员为展现技术的恶作剧式攻击,转变成目的性极强、组织严密、技术高超的团伙式黑色产业链,攻击者的黑手不断向各个领域传导渗透。《Freebuff 2018 年度互联网安全报告》显示[2],恶意代码的种类和数量日渐庞大,危害日益严重。恶意代码在众多的网络威胁和安全事件中最具威胁性,在网络安全防护中已知的恶意软件攻击反而相对威胁较小,真正具有巨大破坏力的是潜在的病毒威胁。随着互联网和移动端的技术进步,恶意软件的攻击手段和攻击形式也发生了巨大的变化。例如 2018 年频发的勒索软件恶意

收稿日期:2019 - 06 - 25。**李劭杰**,硕士生,主研领域:网络安全。**王晨**,高工。**史崄**,学士。

程序利用影响范围广的漏洞快速传播,形式和目的都具有多样性,对我国的政府、医疗和教育等机构造成了严重的损失。云平台给我们的生活带来极大的便利,我们将大量关乎国计民生、企业运营的系统平台部署到云服务器上,这也使云服务平台成为黑客攻击的重灾区。2018 年,云平台遭受的攻击超过其他各类型网络安全事件总比例的 50%,其中攻击手段以 DDoS 攻击、植入后门和篡改网站为主。另外恶意软件攻击越来越多开始针对工业控制系统并且带有极强的目的性,恶意软件 Trisis 就曾成功袭击我国某石油天然气工厂致使工厂停运。现在有越来越多的恶意嗅探器针对国内的工业设备、系统平台进行目的性极强的侦测,虽然目前仍未造成较大的损失,但仍需提高警惕。黑客攻击从原来原始的单一攻击手段演变到现在复杂多重伪装的安全事件,提醒我们恶意软件的检测防护工作时刻不能放松,否则会给我们的生活带来难以估量的损失。但是恶意代码在变种的过程中有许多内联性和相似性,多特征融合检测方法可以有效地发现潜在的恶意代码。现代恶意代码样本数据规模十分庞大,机器学习算法能有效克服这一困难,完成对庞大数据集的训练。随机森林以决策树为基础高度灵活且准确地对恶意代码完成分类,通过判断种类来确定样本是否已知,从而采取相应的安全措施,是一种值得尝试的方法。

1　相关研究

20 世纪 70 年代 Creeper 程序由 BNN Technologies 的开发人员 Bob Thomas 创建,标志着计算机病毒的概念由此产生。虽然这个程序初衷并非要破坏个人电脑,且该程序也是在数年之后才被认定为是病毒,但是其程序设计理念是当下恶意程序的蓝本。随着 Reaper 程序诞生,它能从系统中完美地切断 Creeper 并将其删除,因此 Reaper 程序也被称为第一个“杀毒软件”,就此拉开了恶意代码攻击与防护拉锯战的序幕,以至于众多组织机构投入大量的资金和精力到恶意代码的检测和防护当中。

机器学习是目前计算机技术的热门话题,而网络安全问题又是一个经久不衰的命题,两者互相渗透后也产生了意想不到的效果。崔鸿雁等[3]提出了机器学习中特征选择的方法,并针对五大类机器学习算法展开了深入探讨,罗列了多种情况,分析了各种算法的解决思路和适用场景以及当数据量不足时的解决办法。高程程等[4]提出了可以将木材表面抽象成灰度图像再通过灰度共生矩阵发现木材表面的纹理特征,这个构想可以应用于恶意代码的检测,因为恶意代码同样可以可视化为灰度图像。周绮凤等[5]提出了一种优化随机森林特征选择的方式,将随机森林的相似度矩阵看作一种特殊的核度量,提出相似性比率的转化量作为优先选择特征的重要指标,来选取随机森林的特征。随着信息技术的发展恶意代码也衍生出了种类繁多的变种,但是对于相同族类的恶意代码其核心代码段具有很高的传承性[6]。因此我们可以将这部分代码作文本相似比对,或者转化成图像进行比对。多分类器算法中随机森林算法(RF)是一个典型代表,作为一种重要和常用的数据挖掘技术,随机森林算法在各个行业和领域都有不错的分类和预测能力[7]。本文通过选取多角度特征与随机森林算法相结合,以达到对恶意代码完成分类的目的。

2　恶意代码文件特征提取

2.1　灰度图特征提取

2.1.1　恶意代码可视化

恶意代码可视化将单纯的文本拓展到了空间图像领域,可以更有效地分析恶意代码结构,为发现其潜在特征提供了新的思路。随后提出了恶意代码转换成灰度图像的想法,将恶意代码二进制文件利用 B2M 算法转换为未压缩灰度图像。将恶意软件做反汇编的预处理,得到. asm 后将其作为二进制位流读取,长度 length 由文件大小和宽度自动获得,向量以二维矩阵的形式排列,每个矩阵元素具有 0 到 255 的值[8]。灰度图像是无色彩的 2D 图像,可记录明亮的信息,因此,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里的像素点,映射成无压缩灰度图像。

2.1.2　灰度图纹理特征

尽管反检测技术产生了大量恶意代码变体,但变体恶意代码在很大程度上共享了祖先的大部分源代码。恶意软件在代码结构上大同小异,殊途同归。利用 B2M 算法将不同家族恶意代码文件转化为对应的灰度图像,图 1 被虚线分割开的 4 个部分:恶意代码 Adialer. C、Instantaccess、Lolyda. AA2 和 Swizzor. gen! I 生成的灰度图像。

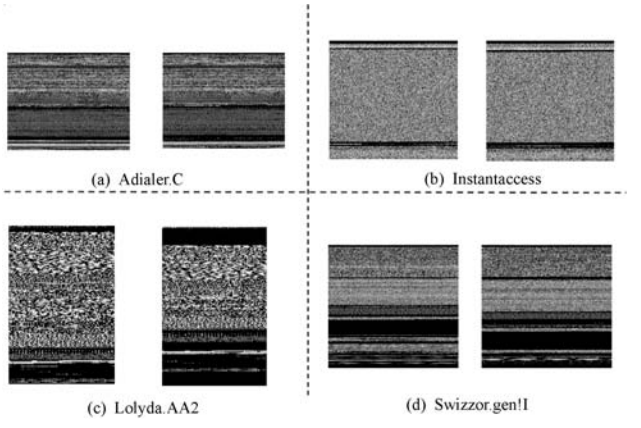
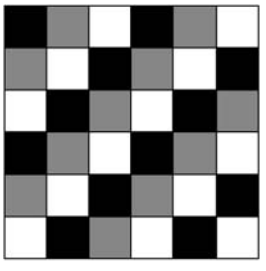


图 1 灰度图

由虚线分隔的 4 个区域代表 4 种恶意代码族,肉眼可以看出同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地取代恶意代码文件本身,成为其另一种表达形式。

2.1.3 灰度共生矩阵

灰度共生矩阵是通过空间特性来描述灰度图像的纹理特征,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着不同的纹理表现,纹理特征由像素值排列分布反映出差别。图 2 是一幅纹理像素的放大图以及其对应的灰度矩阵,可以看出灰度图有三个灰度级别,所以其灰度共生矩阵应为 3 ×3 的矩阵。



0	1	2	0	1	2
1	2	0	1	2	0
2	0	1	2	0	1
0	1	2	0	1	2
1	2	0	1	2	0
2	0	1	2	0	1

图 2 灰度像素图及矩阵

则图 2 中灰度像素图的灰度共生矩阵为:

$$\boldsymbol{P}_{\sigma}=\begin{bmatrix}0 & 10 & 10\\ 10 & 0 & 10\\ 10 & 10 & 0\end{bmatrix}\quad \sigma=(\pm 1,0) \tag{1}$$

归一化后的形式为:

$$\boldsymbol{P}_{\sigma}=\begin{bmatrix}0 & 1/6 & 1/6\\ 1/6 & 0 & 1/6\\ 1/6 & 1/6 & 0\end{bmatrix} \tag{2}$$

灰度共生矩阵通过灰度像素图及其灰度像素矩阵计算而得,灰度共生矩阵的一些特征值代表灰度像素图的部分特性。可以通过一些参数体现图像间清晰度的差异,观察其矩阵内部行和列的相似程度,以及灰度分布均匀度和纹理的粗细程度,还可以发现纹理是否存在一致性。这些值可以在一定程度上反映一幅灰度图像的特性,选用这些值作为灰度共生矩阵的特征,可以有效地筛选出相同类别的恶意软件。

2.1.4 灰度直方图

数字图像既具有空间分布特性又有统计特性,也是灰度图像的颜色特征,灰度直方图可以在一定程度上代表数字图像的全局特征。基于图 1 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,采用灰度直方图作为恶意代码全局特征并验证其在分类中的表现。恶意代码 Adialer. C 灰度直方图如图 3 所示。

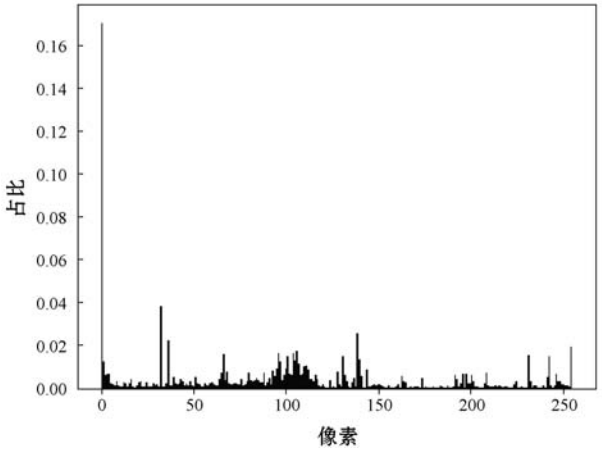


图 3 Adialer. C 灰度直方图

量化灰度图后方便提取其灰度分布的特征,直方图统计的数据和图像中亮暗的位置无关,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。

2.2 OpCode N-Gram 的特征提取

先从. asm 文件中获取操作码序列[9]。N-Gram 模型的第 n 个单词仅与其前 $n-1$ 个单词相关,整个句子出现的概率就等于各个词出现的概率乘积,可以快速地完成文本相似度比对。本文准备对 2-Gram 至 6-Gram 的特征进行简单的测试,由于 4-Gram 以上的测试需要庞大的语料且时间复杂度很高所以测试效果并不理想,准确率也低于 3-Gram 特征,所以本文决定采用出现次数超过 400 的 3-Gram 作为特征。

3 实 验

3.1 实验数据集

本文数据集由微软公司在 2015 年 Kaggle 的恶意代码分类大赛中提供,本文选用的恶意代码共 9 类,包

括 Adialer. C、Instantaccess、Lolyda. AA1、Lolyda. AA2、Lolyda. AA3、Yuner. A、VB. AT、Skintrim. N 和 Swizzor. gen!I[10]。从 35 GB 的 Kaggle 数据集中抽取2 000条数据进行实验,训练测试数据按照 6∶4的比例进行测试。

3.2　随机森林算法

随机森林算法有其独特的采样方式和分类形式,它采用"装袋"的方式训练数据,首先从原始样本集中抽取训练集进行 n 轮抽取得到 n 个训练集,然后使用抽取的 n 个训练集得到 n 个模型,最后 n 个训练集按照投票的方式得到分类结果。其中:f 表示特征;c 表示基于某种特征得到的分类结果。如图 4 所示。

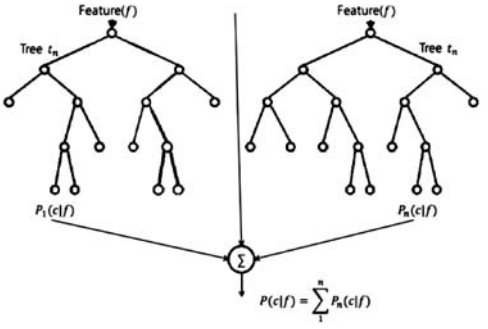


图 4　基础模型(决策树)

每棵树由列采样随机使用一定量特征值训练出来,并且每棵树的分类能力具有很强的针对性,将一个样本通过决策树按照不同的角度进行分类,并将这些预测结果通过投票的方式给出最终结果,将针对分类能力强的分类器整合,是集体智慧的体现,其分类性能往往高于单个分类器[10]。

Gini 系数是我们选择随机森林特征的重要参考标准。特征选择依据于 Gini 系数的增益,Gini 系数用来计算样本不纯度的公式如下:

$$Gini(T) = 1 - \sum_{i}^{c} [p(i \mid T)]^2 \tag{3}$$

式中:c 表示数据集类别数量;i 表示第 i 种分类;计算样本数量占所有样本的比例,数据混合程度越高 Gini 指数也越高。若数据集 T 被特征 A 分成 n 个 T_a 子集,则分裂后属性 A 划分子集的 Gini 指数为:

$$Gini_A(T) = \sum_{a=1}^{n} \frac{|T_a|}{|T|} Gini(T_a) \tag{4}$$

其增益指数为:

$$\Delta Gini(A) = Gini(T) - Gini_A(T) \tag{5}$$

分类器采用随机森林算法,为了准确地进行分析和验证,本实验使用以下几个参数进行验证分析:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

式中:TP 表示为预测为真,实际为真;TN 预测为假,实际为假;FP 预测为真,实际为假;FN 为预测为假,实际为假。

本文之所以选择随机森林算法,是因为随机森林算法可以处理高维数据,在采样时有其独有的特点,随机抽样和又放回的抽样两个随机性可有效地减小过拟合程度,且其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,必然会遗失其他信息,各种混淆技术会导致代码不全,随机森林模型具有对缺失值不敏感的特点,所以其对恶意代码特征有很好的包容性,能够对抗部分混淆。

3.3　单一特征分类性能比较

3.3.1　基于灰度直方图的随机森林

将 3 种特征的特征向量分别输送到随机森林分类器进行训练,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,其余数据作为测试集。

由表 1 可知共有实例 1 524 例,其中 1 类实例共有 118 例,预测正确的有 99 例,错误预测成 3 类的共有 15 例,错误预测成 4、5、7 类的各有 2、1、1 个,2 类预测正确的 76 例无预测错误,3 类实例共有 119 例,预测正确有 103 例,错误预测共有 16 例。错误预测成 1、6、7 类的各有 2、3、11 个,以此类推。

表 1　混淆矩阵

实例 类别	1	2	3	4	5	6	7	8	9
1	99	0	15	2	1	0	1	0	0
2	0	76	0	0	0	0	0	0	0
3	2	0	103	0	0	3	11	0	0
4	0	0	0	105	0	1	8	5	0
5	0	7	0	0	357	0	3	0	0
6	0	0	2	0	0	349	0	0	1
7	0	0	0	2	0	0	133	6	3
8	0	0	1	0	0	1	0	150	0
9	0	0	2	0	4	1	0	0	60

由表 2 可知利用随机森林算法和灰度直方图获得

的特征值相结合其准确率为 93.96%，能有效地对恶意代码进行分类。

表 2 综合指标

实例类别	Precision	Recall	F1-score	Support
1	0.838 983	0.980 198	0.904 110	118
2	1.000 000	0.915 663	0.955 975	76
3	0.865 546	0.837 398	0.851 181	129
4	0.882 353	0.963 303	0.921 053	119
5	0.972 752	0.986 188	0.979 398	367
6	0.991 477	0.983 099	0.987 262	352
7	0.923 611	0.852 564	0.886 626	144
8	0.986 842	0.931 677	0.958 426	152
9	0.895 522	0.937 500	0.915 301	67
平均值	0.928 565	0.931 954	0.928 814	1 524

3.3.2 基于 OpCode 3-Gram 的随机森林

同样参照上文的方法可以发现，采用 OpCode N-Gram 的方式作为特征预测，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，本文对此进一步作特征选择。

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，结果如图 5 所示。经过测试发现与随机森林算法相结合后 3-Gram 特征的准确率和稳定性都要略优于其他两种特征，其准确率为 94.75%。对比灰度直方图特征，3-Gram 特征略强于灰度直方图的特征值的随机森林，所以其同样也能有效地对恶意代码进行分类。

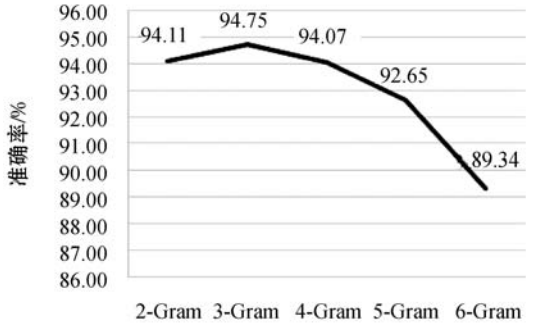


图 5 N-Gram 对分类准确率的影响

3.3.3 基于灰度共生矩阵的随机森林

利用随机森林算法和灰度共生矩阵获得的特征值相结合，本文选取灰度共生矩阵的对比度、相异性、同质性、能量和自相关系数作为灰度共生矩阵提取出的灰度图像特征。恶意代码的分类准确率为 96.01%，相较于其他两个单一特征准确率最高，具有良好的分类效果。

3.3.4 基于特征融合随机森林

将三个特征融合在一起，如表 3 所示。

表 3 融合特征综合指标

实例类别	Precision	Recall	F1-score
1	0.932 203	0.982 143	0.956 522
2	1.000 000	0.915 663	0.955 975
3	0.923 664	0.930 769	0.927 149
4	0.932 773	0.973 684	0.952 790
5	0.973 262	0.986 450	0.979 787
6	0.991 477	0.983 099	0.987 262
7	0.951 389	0.913 333	0.931 929
8	0.986 842	0.961 538	0.973 984
9	0.910 448	0.953 125	0.930 662
平均值	0.955 784	0.955 534	0.955 118

特征向量合并成新的特征向量输入随机森林分类器，进行多次训练后按照融合特征综合指标所示，根据图 6 所示三种特征融合后作为新的特征构成的随机森林，在精确率、召回率和 F1 参数上都有一定的提高，说明融合特征作为分类标准能取得较好的分类效果。

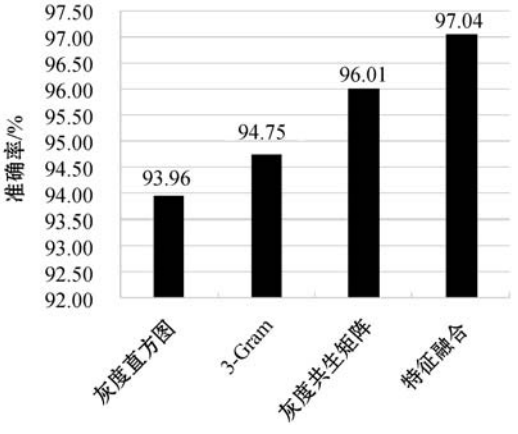


图 6 各特征准确率比较

与 3 个单一特征分类的平均准确度相比，融合特征的分类准确率为 97.04%，比单个特征中具有最高准确度的灰度共生矩阵高 1.03%，充分说明组合特征能够很好地对恶意代码进行准确分类。

4 结 语

从恶意代码反编译文件的核心代码出发提取 Opcode N-Gram 和灰度共生矩阵特征以及可以描述病毒文件的全局特征的灰度直方图，算法系统由特征提取、训练特征集、聚类 3 部分组成，实验结果表明融合后的特征可以有效清晰地描述恶意代码特征，从而判断其种类。本文对 9 类恶意代码样本进行实验验证，

结果表明灰度直方图的随机森林、灰度共生矩阵的随机森林、N-Gram 的随机森林,以及融合特征的随机森林均可以有效地进行恶意代码的分类,其中 3 种特征融合后与随机森林算法相结合其分类效果显著提升。目前更多的是静态特征融合,从代码生成的灰度图纹理和恶意代码文本两方面为切入点,取得不错的分类效果。下一步将融合一些动态特征,比如恶意软件的行为特征,观察其是否可以进一步提高恶意代码分类的准确率,从而进一步优化分类器。

参 考 文 献

[1] 国家互联网应急中心(CNCERT). 2018 年我国互联网网络安全态势报告[OL]. [2019 - 04 - 18]. https://www.freebuf.com/articles/network/201280.html.

[2] 奇虎 360. 中国互联网安全报告[EB/OL]. 2018 - 08 - 02. https://www.freebuf.com/articles/paper/179295.html.

[3] 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.

[4] 高程程,惠晓威. 基于灰度共生矩阵的纹理特征提取[J]. 计算机系统应用,2010,19(6):195 - 198.

[5] 周绮凤,洪文财,杨帆,等. 基于随机森林相似度矩阵差异性的特征选择[J]. 华中科技大学学报(自然科学版),2010,38(4):58 - 61.

[6] 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.

[7] Breiman L. Random forest[J]. Machine Learning,2001,45:5 - 32.

[8] Nataraj L,Karthikeyan S,Jacob G,et al. Malware images:visualization and automatic classification[C]//8th International Symposium on Visualization for Cyber Security. ACM,2011.

[9] 戴逸辉,殷旭东. 基于随机森林的恶意代码检测[J]. 网络空间安全,2018,9(2):70 - 75.

[10] Kaggle[OL]. https://kaggle.com/c/maleware classification/.

(上接第 322 页)

[9] Rndic N, Laskov P. Practical evasion of a learning-based classifier: A case study[C]//2014 IEEE Symposium on Security and Privacy. IEEE,2014:197 - 211.

[10] 张思思,左信,刘建伟. 深度学习中的对抗样本问题[J]. 计算机学报,2019,42(8):1886 - 1904.

[11] Papernot N, McDaniel P, Wu X, et al. Distillation as a defense to adversarial perturbations against deep neural networks[C]//2016 IEEE Symposium on Security and Privacy (SP). IEEE,2016:582 - 597.

[12] Carlini N, Wagner D. Defensive distillation is not robust to adversarial examples[EB]. arXiv:1607.04311,2016.

[13] Graese A, Rozsa A, Boult T E. Assessing threat of adversarial examples on deep neural networks[C]//2016 15th IEEE International Conference on Machine Learning and Applications(ICMLA). IEEE,2016:69 - 74.

[14] Shaham U, Yamada Y, Negahban S. Understanding adversarial training: Increasing local stability of supervised models through robust optimization[J]. Neurocomputing,2018,307:195 - 204.

[15] Zhang F, Chan P P, Biggio B, et al. Adversarial feature selection against evasion attacks[J]. IEEE Transactions on Cybernetics,2016,46(3):766 - 777.

[16] Bhagoji A N, Cullina D, Mittal P. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers[EB]. arXiv:1704.02654,2017.

[17] Biggio B, Corona I, Fumera G, et al. Bagging classifiers for fighting poisoning attacks in adversarial classification tasks[C]//International workshop on multiple classifier systems. Springer, 2015:350 - 359.

[18] Xu W L, Qi Y J, Evans D. Automatically evading classifiers[C]//Proceedings of the 2016 Network and Distributed Systems Symposium,2016:21 - 24.

(上接第 327 页)

[12] Dauphin Y N,Fan A,Auli M,et al. Language modeling with gated convolutional networks[EB]. arXiv:1612.08083,2016.

[13] 冉鹏,王灵,李昕,等. 改进 Softmax 分类器的深度卷积神经网络及其在人脸识别中的应用[J]. 上海大学学报(自然科学版),2018,24(3):352 - 366.

[14] Krawczyk B. Learning from imbalanced data:open challenges and future directions[J]. Progress in Artificial Intelligence,2016,5(4):221 - 232.

[15] Lin T Y,Goyal P,Girshick R,et al. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2018,42(2):318 - 327.

[16] Simonyan K,Zisserman A. Very deep convolutional networks for large-scale image recognition[EB]. arXiv:1409.1556,2014.

[17] Kinga D P,Adam J B. Adam:A method for stochastic optimization[C]//International Conference on Learning Representations (ICLR),2015.

[18] Tavallaee M,Bagheri E,Lu W,et al. A detailed analysis of the KDD CUP 99 data set[C]//2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. IEEE,2009:1 - 6.