一种基于特征编码技术的恶意代码检测方法

丁 应,李 琳
(武汉科技大学 计算机科学与技术学院,湖北 武汉 430065)

摘 要:在对恶意代码进行检测和分类时,由于传统的灰度编码方法将特征转换为图像的过程中,会产生特征分裂和精度损失等问题,严重影响了恶意代码的检测性能。同时,传统的恶意代码检测和分类的数据集中只使用了单一的恶意样本,并没有考虑到良性样本。因此,文中采用了一个包含良性样本和恶意样本的数据集,同时提出了一种双字节特征编码方法。首先将待检测的PE文件特征编码为二进制数,从单个特征中取前两个字节,然后将所有字节转换为图像,最后通过卷积神经网络提取特征并在测试集上进行验证。实验表明,通过将待检测的PE文件的特征进行双字节编码处理,相对于同等条件下的灰度编码方法,其准确率从81.4%提升到92.82%。实验结果证明双字节特征编码方法能够有效地应用于恶意代码检测中。

关键词:双字节;特征编码;卷积神经网络;恶意代码;检测
中图分类号:TP309;TP181 **文献标识码**:A **文章编号**:1673-629X(2021)01-0131-06
doi:10.3969/j.issn.1673-629X.2021.01.024

A Method for Detecting Malicious Code Based on Feature Encoding Technology

DING Ying, LI Lin
(School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China)

Abstract: In the detection and classification of malicious codes, the traditional gray-scale coding method will produce feature splitting and accuracy loss in the process of converting features into images, which will seriously affect the detection performance of malicious codes. At the same time, the traditional malicious code detection and classification dataset only uses a single malicious sample and does not take into account benign samples. Therefore, we adopt a dataset including benign samples and malicious samples and propose a double byte feature encoding method. Firstly, the features of PE file to be detected are encoded as binary numbers, the first two bytes are taken from a single feature, then all bytes are transformed into images, and finally the features are extracted by convolutional neural network and verified on the test set. Experiments show that the PE file to be detected is double byte encoded, the accuracy rate is improved from 81.4% to 92.82% compared to the gray encoding method under the same conditions. The experimental results prove that the double-byte feature encoding method can be effectively applied to malicious code detection.

Key words: double-byte; feature encoding; convolutional neural network; malicious code; detection

0 引 言

早期的恶意代码由于没有采用复杂的加密算法对其进行加密,因此可以通过交叉匹配部分代码的方式来检测恶意代码。但是随着多态和变态(混淆)等现象的出现,恶意代码每经过一轮迭代就进行一次相同密钥的加密算法进行加密(多态),有的甚至使用不同密钥的加密算法进行加密(变态),使恶意代码变得极难检测,恶意代码分析师往往需要花费至少一周甚至更久的时间才能分析一个新出现的病毒属于哪个家族。最近的一份安全报告显示,虽然攻击者对开发移动端的恶意代码的兴趣越来越高,但是Windows系统依然是黑客攻击的首要目标。2019年至今,几乎每周都有重大网络事件发生,给受害者造成了巨大的经济损失,更严重的甚至会危害受害者的生命安全。

传统的恶意代码检测技术可以分为静态分析和动态分析两种方法。静态分析[1]的优势是可以在代码执行之前对其进行分析[2],并且可以在确切的位置挖掘出代码的敏感信息,通过这些信息可以判断该代码属于哪种类型的恶意代码(木马,蠕虫,后门),动态分析方法很难做到这些;而在动态分析[3]中,代码在执行

收稿日期:2020-03-10 **修回日期**:2020-07-12
基金项目:国家自然科学基金(61702383,61602350);湖北省教育科研项目(B2018554);国家级"大创计划"项目(201810488012)
作者简介:丁 应(1996-),男,硕士研究生,研究方向为恶意代码、深度学习;李 琳,博士,讲师,研究方向为人工智能、信息安全。

阶段会被实时监控[4](虚拟机,沙箱技术等),可以轻易地找到软件受感染的位置,但在静态分析中这是不可能的。

静态检测技术在对恶意代码进行分析时,需要借助恶意数据特征库。由于恶意数据特征库存在一段时间的更新周期,如果检测不及时,会存在漏检、误报等问题;并且静态检测技术对新型病毒也会显得心有余而力不足。

动态检测技术在对恶意代码进行分析时,需要借助虚拟机、虚拟镜像、沙箱等技术,这不仅会造成资源的浪费,而且一些病毒甚至能判断自身所处的环境,在虚拟环境下并不表现出自身的特点,导致动态检测误报。

无论是动态检测技术还是静态检测技术,在进行检测时还存在着其他的局限性,比如只能用于某些安全公司的"自产自销"。如今,互联网高度集成发达,如果不能将自身资源分享出去,也是一种资源的浪费。

针对上述传统的恶意代码检测技术的不足,深度学习算法凭借其强大的特征提取能力自然而然地被研究者用来对恶意代码进行检测和分类。T. Kim 等人[5]将 RGB 编码技术和卷积神经网络结合起来对异常网络进行检测,通过将数据特征编码成彩色图像,然后使用卷积神经网络对生成的彩色图像进行分类,在三个不同的数据集上都取得了不错的准确率。韩晓光等人[6]通过把恶意代码映射为无压缩的灰度图片,然后对图片进行相应的处理,最后使用深度学习算法对其进行分类,准确率不高,而且过程过于复杂,单给研究者提供了思路。Baptista I[7]通过将恶意样本转换为灰度图像后,与无监督学习方法结合,对各种恶意文档或软件(pdf,txt,exe,doc 等)的分类都能达到较高的准确率。

由此可以看出,将深度学习算法与其他方法结合起来,然后应用于对恶意代码的检测和分类,其性能远远优于传统的恶意代码检测技术。但是上述文献中大多没有引入良性数据集,严格意义来讲,它们并不能算是对恶意代码的检测。因此,该文采用的数据集包含良性样本和恶意样本,然后将数据集上待检测的 PE 文件的特征进行双字节编码,转换为 8 × 8 的灰度图像,最后通过卷积神经网络提取图像特征并对测试集进行检测。实验结果表明,提出的双字节特征编码方法在 Ember 数据集上的准确率为 92.82%,与传统的灰度编码技术相比,提高了 11.42%。

1 相关工作

随着深度学习算法的不断发展,它们已经被广泛地应用于恶意代码的检测和分类当中。蒋晨等人[8]通过将深度神经网络应用于不同平台下恶意代码的分类,其结果表明该方法在 Windows 平台上的分类效果远高于安卓平台。A. I. Elkhawas 等[9]通过将待检测的 PE 文件的特征转换成三元组的形式,然后通过 SVM 算法对其进行分类,在 PE 文件的分类上具有非常好的表现。A. Utku 等[10]采用的决策树算法在移动端恶意代码的分类上的准确率可以达到 95% 以上。其他的深度学习算法[11-15],在检测其他的恶意事件中也具有良好的表现。有趣的是,当生成对抗网络技术出现时,研究者发现可以通过生成技术在恶意代码某些特定的部分填充一些字节,在不改变该代码功能的前提下来逃避深度学习方法对可执行软件的检测,进而把恶意样本误判为良性样本。而 Vineeth S. Bhaskara 等人[16]通过生成技术在已知恶意软件行为的可逆分布 RGB 图像表示上训练 GAN,编码 API 调用 n-gram 的序列和相应的项频率,生成的图像表示可以重新解码为底层 API,通过调用序列信息可以合成恶意软件。Z. Li 等人[17]提出了一种灰度编码方法,通过将入侵检测数据集的单个特征编码为 10 位二进制数(0b0000000000-0b1111111111),然后转换为灰度图像,并用深度学习算法对其进行分类,最终的准确率为 82%。由此可知,将各种特征处理方法与深度学习算法相结合,在恶意代码的检测和分类上的表现都优于传统方法。

EMBER 数据集是 Endgame 公司在 2018 年四月份发布的一个大型开源数据集。该数据集可以用于训练机器学习模型来检测静态 Windows 便携式可执行文件。它总共有 110 万条数据,其中训练集有 90 万条,恶意样本、良性样本以及未标注样本各 30 万条;测试集有 20 万条,恶意样本和良性样本各 10 万条。该数据集用 jsonlines 格式保存,并详细解释了各数据的含义。其父特征有 8 类,部分父特征又可以分解成子类特征,总共构成了 56 个特征(父特征和子类特征的集合);这些特征又可以分成数值特征和非数值特征。

2 双字节特征编码与深度学习

在使用深度学习算法对恶意代码进行检测之前,首先需要将非数值特征转化为数值特征,然后将数值特征转换成二进制数,接着通过双字节特征编码技术将其转成灰度图片,最后使用深度学习算法对其进行检测。本章主要介绍了该文使用的 EMBER 数据集的特征、提出的双字节特征编码技术以及所使用的深度学习算法。

2.1 相关特征介绍

根据 EndGame 公司对 EMBER 数据集特征的描述筛选出如图 1 中的特征用来编码,包含 4 个父特征

和 35 个子类特征。具体描述如下:

label:标签,值为 1 代表恶意样本,0 为良性样本,-1 表示未标记,可用作半监督学习。

general:常规文件信息,主要包括的特征有文件大小、从 PE 头获得的基本信息、文件的虚拟大小、导入函数和导出函数、文件是否具有调试部分、线程本地存储、资源、重定位、签名以及符号数。

header:在 coff 标头中,主要包括目标主机(字符串)和图像特征列表(字符串列表);在 optional 标头中,特征有目标子系统(字符串)、DLL 特性(字符串列表)、magic 字符串(例如“PE32”)、主要和次要映像版本、链接器版本、系统版本和子系统版本、代码、标头以及提交大小。

strings:字符串中的特征包括简单的统计信息,至少五个可打印字符长度的可打印字符串(由 0x20 到 0x7f 范围内的字符组成)、字符串的数量、字符串的平均长度、所有可打印字符串中字符的熵;此外,字符串特征还包括以 C:\(不区分大小写)开头的可能表示路径的字符串数,以及 http://或 https://(不区分大小写)的出现次数、它们表示路径 URL、还有表示注册表项的 HKEY_的出现次数以及可能提供 Windows PE 删除程序或捆绑的可执行文件的短字符串 MZ 的出现次数。该数据集提供的是字符串的简单统计摘要而不是原始字符串的列表,可以尽量不泄露某些良性文件可能存在的隐私问题。


```
"label": 0,
"general": {
"size": 3101705,
"vsize": 380928,
"has_debug": 0,
"exports": 0,
"imports": 156,
"has_relocations": 0,
"has_resources": 1,
"has_signature": 0,
"has_tls": 0,
"symbols": 0},
"header": {
"coff": {
"machine": "I386",
"characteristics": ["CHARA_32BIT_MACHINE", "RELOCS_STRIPPED", "EXECUTABLE_
IMAGE", "LINE_NUMS_STRIPPED", "LOCAL_SYMS_STRIPPED"]
},
"optional": {
"subsystem": "WINDOWS_GUI",
"dll_characteristics": [],
"magic": "PE32",
"major_image_version": 0,
"minor_image_version": 0,
"major_linker_version": 7,
"minor_linker_version": 10,
"major_operating_system_version": 4,
"minor_operating_system_version": 0,
"major_subsystem_version": 4,
"minor_subsystem_version": 0,
"sizeof_code": 26624,
"sizeof_headers": 1024,
"sizeof_heap_commit": 4096}
},
"strings": {
"numstrings": 14573,
"avlength": 5.972071639333013,
"printables": 87031,
"entropy": 6.569897560341239,
"paths": 3,
"urls": 0,
"registry": 0,
"MZ": 51
},
}
```


图 1　单个 PE 文件的部分原始特征

图 1 和 2.1 节详细地描述了单个 PE 文件的部分原始特征以及特征的含义。

2.2　特征编码技术

本节主要介绍该文对数值特征采用的数据预处理技术以及提出的双字节特征编码方法,并将该方法相较于传统灰度编码方法的优点进行了简单的描述。

2.2.1　数据预处理

由图 1 可知数据集的子类特征包含两种,一种是非数值特征,比如 subsystem 和 dll_characteristics 等特征;另一种是数值特征,比如 numstrings 和 entropy 等特征。对于非数值特征,如果该特征中的子类特征的数目不超过 16,那么就可以使用独热编码的方式对其进行编码,例如 subsystem 的子类特征数目为 12,所以其编码方式为(0b000000000001,0b100000000000),即 12 个位置上每一个位置只有一位为 1,其余为 0。对于子类特征数目超过 16 的情况(如图 1 中的 dll_characteristics),直接对其进行类别编码并转换为数值特征。

对于数值特征,使用 MIN-MAX 归一化的方法对其进行处理,转换成[0,1]范围的值,公式如下:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

其中,x_{norm} 为归一化后的数字,x 为原始数字,$x_{\min}$ 和 $x_{\max}$ 分别为该特征的最小值和最大值。

2.2.2　双字节特征编码方法

该文提出了一种双字节特征编码方法,是将单个的数值特征都编码为相等数量的字节(每个数值特征编码为两个字节),非数值特征根据子类特征数目分别进行独热编码或类别编码。而数值特征的编码规则如式(2):

$$\text{binstring} = \begin{cases} 0\text{b}0000000000000000 & \text{value} = 0 \\ \text{bin}(\text{value})[0:16] & 0 < \text{value} < 1 \\ 0\text{b}1111111111111111 & \text{value} = 1 \end{cases} \tag{2}$$

由式(2)可知,对于数值特征的子类特征,如果该值为 0.0 或 1.0,就将其编码为 0x0000(值为“0”)或 0xffff(值为“1”);如果该值在 0.0 到 1.0 之间,那么就将十进制小数转换为二进制小数并取前十六位,即每个特征代表两个字节。非数值特征的处理方式在 2.2.1 节中已作详细介绍。

将待检测的 PE 文件特征通过该文提出的双字节特征编码方法进行编码后,得到的字节的数量是确定的。假设 N 为数值特征,M 为非数值特征中子特征数大于 16 的特征,K 为非数值特征中子特征数小于 16 的所有特征经过独热编码后相加的数量。则该文提出的编码方法的字节数量为 $((M+N)\times 16+K)/8$,最

后得到的字节数目是 64,刚好可以转换为 8 × 8 的灰度图像。而传统的灰度编码方法经过相同处理后,最终得到的大小为 42.5 的字节,需要填充 6.5 个都为 0 的字节才能转换为 7 × 7 的灰度图像。图 2 为任意四个待检测的 PE 文件的特征值经过双字节特征编码后生成的图片(经过 1 228% 放大后)。

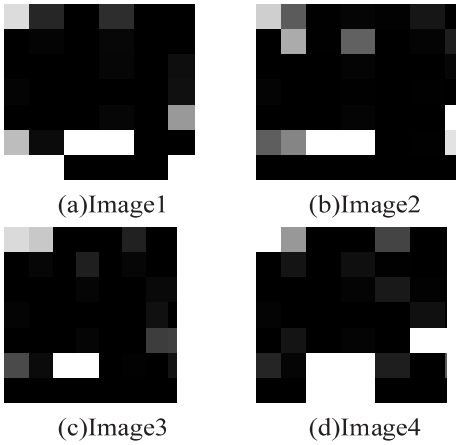

图 2 待检测 PE 文件的特征编码图片

该文提出的特征编码与灰度编码相比,有以下优点:

在编码时将每个数值特征以及子类特征中类别超过 16 的都编码为 2 个字节,而灰度编码只是将特征编码为 1.25 个字节。灰度编码在转换为图像的时候会造成特征分裂,降低准确率。

在精度方面,灰度编码的损失远远大于特征编码。比如灰度编码将特征值 0.400 1 和 0.499 9 都编码为 0b0001000000,损失值为 0.098。而双字节特征编码则分别将其编码为 0b0110011001101100 和 0b0111111111111001,是完全两个不同的值,且损失值小于 2^{-16},与灰度编码方法相比,损失值可以忽略不计。

2.3 深度学习算法

CNN(convolutional neural network)模型的灵感来源于动物视觉皮层组织,该组织负责小范围内的光检测[18]。通常,CNN 由两种类型的层组成,分别称为提取特征的卷积层和用于特征映射的池化层,可以很好地提取特征,通过该特征可以轻易地识别出图像之间的区别,因此 CNN 在图像分类上具有卓越效果。在 2014 年,ImageNet 大规模视觉识别挑战赛(ILSVRC)竞赛的获胜者也是以 CNN 模型为架构的[19],在比赛中,它的错误率仅为 6.67%,与人类水平的性能几乎相同。由 2.1 节可知,该文的输入是 8×8 大小的灰度图片,而输出分别为 0(良性)或者 1(恶意),属于二分类问题。因此,适合用 CNN 模型来对待检测的 PE 文件进行检测。

在 TensorFlow 框架的基础上,把输入大小为 8 × 8 的灰度图片,经过一次 3 × 3 的卷积之后,得到 6 × 6 大小的灰度图片,通过使用 ReLU 激活函数可以有效防止梯度消失问题,并且计算量也比较小,然后使用 2 × 2 的最大池化层来减少参数误差造成的均值偏移,最后输出的图片大小为 5 × 5。由于此时的分类效果并不理想,因此在此基础上重复做了一次 3 × 3 的卷积和 2 × 2 的最大池化,得到输出为 2 × 2 大小的灰度图片。为了避免出现梯度消失和过拟合等问题,分别使用带有 ReLU 激活函数和 dropout 的全连层,最后使用 softmax 对数据集进行分类。图 3 为该文采用的 CNN 网络架构。

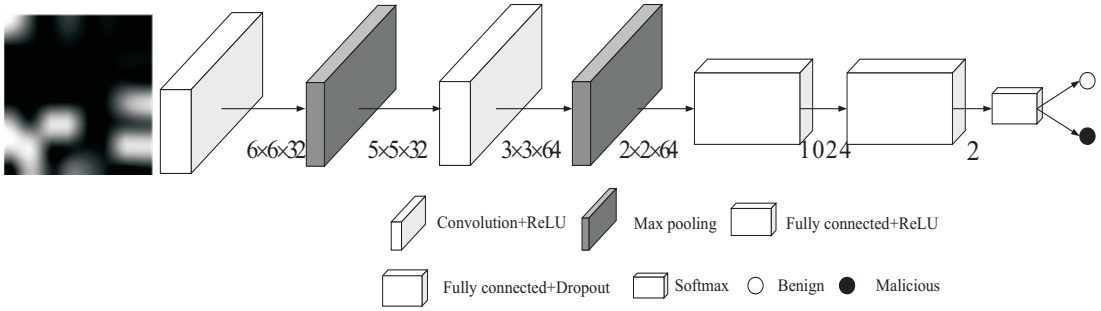

图 3 CNN 网络架构

3 实验结果分析

由于大部分开源数据集都只有恶性样本而没有良性样本(例如:BigData2015),因此该文选择 EndGame 公司在 2018 年四月份公布的包含两种样本的 EMBER 数据集;并通过准确率、查全率、查准率三种具有代表性的度量性能比较该文提出的双字节特征编码以及传统的灰度编码技术。

本节主要介绍了实验过程中使用到的数据集以及对实验结果的具体分析。

3.1 数据集

在 EMBER 数据集的基础上分别随机选取 30 万条良性样本和恶意样本,以形成 60 万条样本的新数据集。然后将该数据集以 9 ∶ 1 的大小随机划分为训练集和测试集,表 1 详细描述了该数据集训练集和测试集的大小。

表 1　数据集分布

数据集	正常样本	恶意样本	总数
训练集	270 000	270 000	540 000
测试集	30 000	30 000	60 000
总数	300 000	300 000	600 000

3.2　实验结果

通过采用如图 3 所示的卷积神经网络架构,对 3.1 节中划分的数据集进行实验,并且采用准确率(accuracy)、查全率(precision)、查准率(recall)和 F1 值度量两种方法的性能,以比较双字节特征编码方法与灰度编码方法的优劣性。其中查准率 P、查全率 R 和 F1 值分别定义为:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4}$$

$$\mathrm{F1-score} = \frac{2 \times P \times R}{P + R} \tag{5}$$

其中,TP 表示真正例,FP 表示假正例,FN 表示假反例。图 4 为双字节特征编码和灰度编码在 CNN 上的准确率曲线。

由图 4 可以看出,在同等情况下,当经过 100 个 epoch 之后,灰度编码方法的准确率仅为 81.4%,而该文提出的双字节特征编码方法的准确率达到 92.82%,比灰度编码提高了 11.42%。

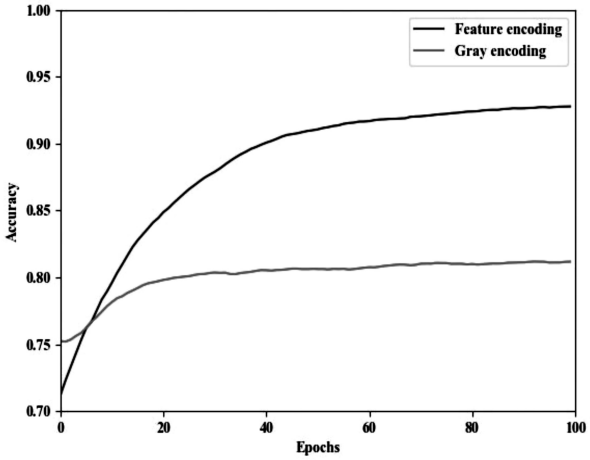

图 4　特征编码和灰度编码的准确率曲线

由表 2 可知,特征编码方法在查准率 P、查全率 R 和 F1 值上的表现都优于灰度编码;双字节特征编码在正常样本和恶意样本上的查全率分别为 93.14% 和 93.16%,而灰度编码分别为 81.39% 和 81.41%;双字节特征编码在正常样本和恶意样本上的查准率分别为 93.17% 和 93.14%,而灰度编码分别为 81.42% 和 81.38%;双字节特征编码在正常样本和恶意样本上的 F1 值为 93.15%,而灰度编码分别为 81.4% 和 81.39%。由此可知,该文提出的双字节特征编码的性能远优于传统的灰度编码方法。

表 2　度量性能

方法	正常样本			恶意样本		
	查全率/%	查准率/%	F1 值/%	查全率/%	查准率/%	F1 值/%
灰度编码+CNN	81.39	81.42	81.4	81.41	81.38	81.39
特征编码+CNN	93.14	93.17	93.15	93.16	93.14	93.15

4　结束语

针对传统的恶意代码检测技术存在漏报、误报以及资源浪费等的不足,结合卷积神经网络技术在图像分类上的优异表现,提出了一种新的双字节特征编码方法。该方法通过将待检测的 PE 文件的单个数值特征和非数值特征中子特征数目超过 16 的编码为 2 个字节,并将编码后的所有字节转换为灰度图像,然后使用卷积神经网络对其进行特征提取并检测。在带有标签的 EMBER 数据集中的数据上进行了实验,实验数据为 60 万条,其中良性样本和恶意样本数各 30 万条。实验结果表明,提出的双字节特征编码方法通过与卷积神经网络算法结合,在 6 万条测试集上的准确率 accuracy、查全率 P、查准率 R 和 F1 值的表现都优于传统的灰度编码方法。由此可知,提出的双字节特征编码方法比灰度编码方法更加有效。

提出的双字节特征编码与传统的灰度编码相比具有以下两个优点:

(1)由于提出的双字节编码是将特征编码为两个字节,因此在生成灰度图像时,不会造成特征分裂,保存了特征的完整性。

(2)在精度方面,提出的双字节编码方法的损失值仅为 2^{-16},远低于灰度编码的 2^{-10},因此在分类性能上大大提升。

提出的双字节特征编码方法存在三点不足:

(1)在对一些非数值特征里面的相关子类特征进行处理时,并没有根据该特征的特点进行对应编码。

(2)没有考虑父特征和子类特征的关联性,没有将父特征与其对应的子类特征进行结合编码。

(3)在深度学习算法方面,CNN 网络结构上的优化也存在不足,并且只使用了一种网络架构。

未来的工作可能会采用新的编码方法对非数值特

征里的子类特征进行编码,根据其特点进行对应编码;并且将父特征和子类特征进行结合编码,体现其关联性;并使用更加成熟的网络结构进行训练(例如:VGG-16,Inception-v4 等)。

参考文献:

[1] KANG H,JANG J W,MOHAISEN A,et al. Detecting and classifying android malware using static analysis along with creator information[J]. International Journal of Distributed Sensor Networks,2015,11(6):914-919.

[2] MOSER A,KRUEGEL C,KIRDA E. Limits of static analysis for malware detection[C]//Twenty-third annual computer security applications conference (ACSAC 2007). Miami Beach,FL,USA:IEEE,2007:421-430.

[3] AL-DUJAILI A,HUANG A,HEMBERG E,et al. Adversarial deep learning for robust detection of binary encoded malware[C]//2018 IEEE security and privacy workshops (SPW). San Francisco,CA,USA:IEEE,2018:76-82.

[4] EGELE M,SCHOLTE T,KIRDA E,et al. A survey on automated dynamic malware-analysis techniques and tools[J]. ACM Computing Surveys,2012,44(2):1-42.

[5] KIM T,SUH S C,KIM H,et al. An encoding technique for CNN-based network anomaly detection[C]//2018 IEEE international conference on big data (big data). Seattle,WA,USA:IEEE,2018:2960-2965.

[6] 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.

[7] BAPTISTA I,SHIAELES S,KOLOKOTRONIS N. A novel malware detection system based on machine learning and binary visualization[C]//2019 IEEE international conference on communications workshops (ICC workshops). Shanghai,China:IEEE,2019:1-6.

[8] 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] ELKHAWAS A I,ABDELBAKI N. Malware detection using opcode trigram sequence with SVM[C]//2018 26th international conference on software,telecommunications and computer networks (SoftCOM). Split,Croatia:IEEE,2018:1-6.

[10] UTKU A, DOĜRU Ī A, AKCAYOL M A. Decision tree based android malware detection system[C]//2018 26th signal processing and communications applications conference (SIU). Izmir:[s. n.],2018:1-4.

[11] DAHL G E,STOKES J W,DENG L,et al. Large-scale malware classification using random projections and neural networks[C]//2013 IEEE international conference on acoustics,speech and signal processing. Vancouver,BC,Canada:IEEE,2013:3422-3426.

[12] SANTACROCE M,KORANEK D,KAPP D,et al. Detecting malicious assembly with deep learning[C]//NAECON 2018-IEEE national aerospace and electronics conference. Dayton,OH:IEEE,2018:82-85.

[13] KARA I,AYDOS M. Static and dynamic analysis of third generation cerber ransomware[C]//2018 international congress on big data,deep learning and fighting cyber terrorism (IBIGDELFT). ANKARA,Turkey:[s. n.],2018:12-17.

[14] TOBIYAMA S,YAMAGUCHI Y,SHIMADA H,et al. Malware detection with deep neural network using process behavior[C]//2016 IEEE 40th annual computer software and applications conference (COMPSAC). Atlanta,GA,USA:IEEE,2016:577-582.

[15] KRISHNAVENI S,SATHIYAKUMARI K. SpiderNet:an interaction tool for predicting malicious web pages[C]//2014 international conference on information communication and embedded systems (ICICES2014). Chennai,India:IEEE,2014:1-6.

[16] KOLOSNJAJI B,DEMONTIS A,BIGGIO B,et al. Adversarial malware binaries:evading deep learning for malware detection in executables[C]//2018 26th European signal processing conference (EUSIPCO). Rome,Italy:IEEE,2018:533-537.

[17] LI Z,QIN Z,HUANG K,et al. Intrusion detection using convolutional neural networks for representation learning[C]//International conference on neural information processing. Guangzhou,China:[s. n.],2017:858-866.

[18] LIU W,WANG Z,LIU X,et al. A survey of deep neural network architectures and their applications[J]. Neurocomputing,2017,234:11-26.

[19] SZEGEDY C,LIU W,JIA Y,et al. Going deeper with convolutions[C]//2015 IEEE conference on computer vision and pattern recognition (CVPR). Boston,MA,USA:IEEE,2015:1-9.