基于特征权重的恶意软件分类方法

叶 彪[1,2],李 琳[1,2],丁 应[3],宋荆汉[4],万振华[4]
(1. 武汉科技大学 计算机科学与技术学院,湖北 武汉 430065;
2. 智能信息处理与实时工业系统湖北省重点实验室,湖北 武汉 430065;
3. 上海航天精密机械研究所,上海 201600;
4. 深圳开源互联网安全技术有限公司,广东 深圳 518000)

摘 要:近年来由于计算机和人们的工作生活结合得更加紧密,为保障信息安全,恶意软件分类的重要性与日俱增,但是现有的恶意软件分类方法大多都存在模型复杂、耗费时间长以及效果不突出等困境。为提高恶意软件分类效率,提出一个结合特征提取和卷积神经网络的恶意软件分类框架。针对目前恶意软件分类算法准确率低、处理时间慢等问题,引入并改进了 NLP 领域中的一种特征权重算法。通过计算操作码的特征权重,选取具有较大信息增益的操作码作为特征词,然后提取恶意样本的特征图,最后传入卷积神经网络进行训练和分类。实验结果表明,该方法在 big2015 数据集上的准确率为 99.26%,比基于 TFIDF 特征提取的方法略好,接近该数据集上的冠军方法,在不均衡类别上的分类表现优于基于频率的特征词选择的提取算法,并且在预处理时间上短于其他方法。

关键词:特征权重;特征提取;操作码;卷积神经网络;恶意软件分类
中图分类号:TP309 **文献标识码**:A **文章编号**:1673-629X(2022)11-0115-06
doi:10.3969/j.issn.1673-629X.2022.11.017

Malware Classification Method Based on Feature Weights

YE Biao[1,2],LI Lin[1,2],DING Ying[3],SONG Jing-han[4],WAN Zhen-hua[4]
(1. School of Computer Science and Technology,Wuhan University of Science and Technology,
Wuhan 430065,China;
2. Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System,
Wuhan 430065,China;
3. Shanghai Aerospace Precision Machinery Research Institute,Shanghai 201600,China;
4. Shenzhen Open Source Internet Security Technology Co.,Ltd.,Shenzhen 518000,China)

Abstract:In recent years,as computers and people's work and life have become more closely integrated,the importance of malware classification has increased day by day to ensure information security. However,most of the existing malware classification methods have difficulties such as complex model,long time-consuming,and inconspicuous effects. In order to improve the efficiency of malware classification,a malware classification framework combining feature extraction and convolutional neural network is proposed. Aiming at the problems of low accuracy and slow processing time of current malware classification algorithms,a feature weighting algorithm in the field of NLP is introduced and improved. By calculating the feature weight of the opcode,the opcode with greater information gain is selected as the feature words,then the feature maps of the malicious sample is extracted,and passed into the convolutional neural network for training and classification at last. Experimental results show that the accuracy of the proposed method on the big2015 dataset is 99.26%,which is slightly better than the method based on TFIDF feature extraction. It is close to the champion method on this dataset,and the classification performance on unbalanced categories is better than that based on frequency. The extraction algorithm for feature word selection,and the preprocessing time is shorter than other methods.

Key words:feature weight;feature extraction;opcode;convolutional neural network;malware classification

收稿日期:2021-12-08 **修回日期**:2022-04-11
基金项目:国家自然科学基金(61572381);湖北省教育厅项目(2020354);湖北省大学生创新创业训练计划项目(S202110488047)
作者简介:叶 彪(1998-),男,硕士研究生,研究方向为机器学习、信息安全;通讯作者:李 琳(1981-),女,博士,副教授,CCF 专业会员(J3887M),研究方向为人工智能、信息安全。

0 引 言

由于恶意软件带来的经济损失逐年上升,例如震网病毒 Stuxnet[1]、勒索病毒 WannaCry[2]。恶意软件检测和分类变得愈发重要。恶意软件分类大体的框架可以分为三个部分:数据处理、特征提取和分类器设计,其中特征处理和分类器设计是恶意软件分类的重要技术手段,数据处理是贯彻始终的方法。在特征提取的阶段,要尽可能筛选信息增益大的特征。特征提取的好坏直接影响分类的结果。近几年恶意代码静态特征提取主要基于字节序列、可阅读字符串、文件头部信息、熵、动态链接库等特征,然后使用机器学习或者深度学习对特征进行训练和分类。

特征提取方法用于深度学习中,Schultz 等人[3]首次将 N-gram 提取方法引入恶意软件的特征提取工作。他们首先将恶意文件的字节转化为 16 进制,然后将其划分为多组字节序列,再利用 N-gram 进行提取,将结果放入多种分类器中进行训练,证明了这种方法比单纯的基于签名的方法更好。Zhang Hanqi 等人[4]提出了一种静态提取方法,将操作码转化为 N-gram 序列,针对勒索病毒取得了不错效果。Nataraj 等人[5]提出将恶意软件转化为灰度图像的方法,验证了图像作为中间表示的可行性。徐玄骥等人[6]以及杨春雨等人[7]使用多个特征进行融合的方法来检测恶意软件,该方法比单纯使用一种特征的效果要好。蒋永康等人[8]将恶意代码的汇编指令转化为图像矢量,该模型在微软 big2015 数据集上的交叉验证准确率达到了 97.87%。传统的特征提取方法没有考虑特征权重的思想,每个样本的特征向量都非常大,运行十分缓慢,因此需要引入特征权重对特征进行筛选。

在特征权重算法中,最经典的就是由 Salton 等人[9]提出的 TFIDF 算法,该算法由词频(TF)和逆文档频率(IDF)组成。它的主要思想是特征在文档中出现的频率越高,出现该特征的文档数越少,则特征对分类的区分度越大。该算法和 1-gram 提取进行结合可以有效降低特征维度、提高信息增益并减少计算时间。基于该算法的相关研究取得了丰富的成果,例如吴思慧等人[10]提出将 TFIDF 自注意层和长短期记忆网络结合使用,在垃圾短信识别上准确率更高,速度更快。但是在恶意软件检测与分类领域,相关文献十分稀少。并且该算法本身有一个缺陷,没有利用标签信息,是一种无监督的特征权重算法。

鉴于上述情况,该文提出了一种改进的特征权重方法,利用标签信息,提升了恶意软件分类的效率。贡献如下:改进了特征权重方法并将其引入恶意软件分类;从理论上解释了特征权重对特征提取的重要性;实验对比了不同模型的分类准确率和时间开销。

1 关键技术

1.1 N-gram 特征提取算法

N-gram 是 NLP(Natural Language Processing, NLP)中大词汇连续语音识别常用的一种语言模型[11],它的基本思想是统计一个特定长度(N)单词的频度,每 1 个单词块为 1gram。通过该方法可以提取特征,从而区分不同类别的语义信息,在恶意软件的特征提取阶段同样可以用到 N-gram 的方法。

常用的是二元的 Bi-gram 和三元的 Tri-gram。当对一个由 256 个字组成的样本进行 Bi-gram 提取时,最多将产生 65 536 个词语,进行 Tri-gram 提取时将产生 16 777 216 个词语,特征提取的时间占所有时间的 90% 以上。由于恶意样本通常是海量的,因此时间成本很重要。该文使用的 1-gram 是 N-gram 的一种特殊情况,此时滑动窗口 N 为 1,可以直接理解为统计单个词语的个数或者频度。该方法为先利用训练集产生一定量的特征词,然后根据特征词库对每个样本进行特征词个数统计,可以节约大量的计算时间。

1.2 特征权重算法

1.2.1 TFIDF 算法

传统的特征权重算法是词频-逆向文件频率算法 TFIDF(Term Frequency - Inverse Document Frequency),它是一种用于信息检索与数据挖掘的常用加权技术,源于 NLP 领域常用于挖掘文章中的关键词。TFIDF 有两层意思,一层是词频(TF),形容一个词语在文章中的出现次数。另一层是逆文档频率(IDF),与此相关的是包含一个词条的文档数,则一个单词的 TFIDF 值就是二者的乘积。

$$\mathrm{TFIDF} = \mathrm{TF} \times \mathrm{IDF} \tag{1}$$

逆文档频率思想认为出现特征的文档数越大,则该特征对分类的区分度越小,此方法可以突出重要词,抑制次要词[12]。但是此方法并不完善,没有利用到标签的信息,因此只需要将逆文档频率这个概念引入标签信息,便可以得到有监督的特征提取算法。

1.2.2 改进的特征权重算法

首先在训练集上统计一定量每个类别中 1-gram 候选词。针对一个候选词 a 进行分析,计算该词在数据集中出现的概率:

$$\mathrm{PA} = \frac{N_a}{N_D} \tag{2}$$

其中,PA 为 a 在数据集中出现的概率,N_a 为 a 在数据集中出现的次数,N_D 为数据集的样本数量。

计算候选词 a 在一个类别 C 中出现的概率:

$$\mathrm{PB} = \frac{C_a}{N_C} \tag{3}$$

其中,C_a 为 a 在 C 中出现的次数,N_C 为 C 的样本数。

计算候选词 a 的特征权重:

$$\text{Weight}(a) = \frac{|\text{PA} - \text{PB}|}{\text{PA}} \tag{4}$$

通过上式计算出的 Weight(a) 即为候选词 a 的特征权重。信息熵(Entropy)是度量样本集合纯度最常用的一种指标,在决策树算法中广泛使用。假定样本集合 D 中第 C 类样本所占的比例是 $C_k(k = 1,2,\cdots,|y|)$,则 D 的信息熵定义为:

$$\text{Entropy}(D) = -\sum_{k=1}^{|y|} C_k \log_2 C_k \tag{5}$$

Entropy(D) 的值越小,则 D 的纯度越高。一般而言,特征词 a 的特征权重越大,那么使用 a 对数据集进行划分得到的纯度提升就越大,也就是说特征词 a 对分类起到的作用越大。计算每个类别中所有候选词的特征权重,然后将所有类别的特征权重进行合并去重,降序排序后取前面的候选词作为特征词库,通过这个特征词库对数据集进行特征提取。

1.3　卷积神经网络

卷积神经网络(Convolutional Neural Network, CNN)是基于前馈神经网络的一种深度学习模型,提供了一种端到端的学习模型,模型中的参数可以通过传统的梯度下降方法进行训练,经过训练的 CNN 能够学习到图像中的特征,并且完成对图像特征的提取和分类工作[13]。CNN 通过卷积和池化操作进行深度学习和参数约简,第一层为输入层,中间有若干隐藏层,最后一层为全连接层,输出分类结果。

同一个家族的恶意软件存在较大的相似性,因此将同一家族的 ASM 文件提取特征后转化为灰度图同样具有相似性,加上 CNN 对图片识别的出色效果,以及 CNN 的运算速度较快,因此选用 CNN 作为分类器。该文采用了 4 层 CNN 的结构,使用 3 * 3 的卷积核和 2 * 2的池化,使得分类更为快速准确;并在每一层中引入了 Relu,而不是整个网络只使用一层 Relu。

1.4　模型的总体框架

如图 1 所示,该方法的流程为:首先将数据分成训练集和测试集,在训练集中对所有样本执行特征权重算法,找出一定量的特征词,然后根据该特征词库对训练集和测试集进行特征提取,生成所有样本的特征图,将特征图传入 CNN 模型进行训练和分类,最后得出模型的分类结果。

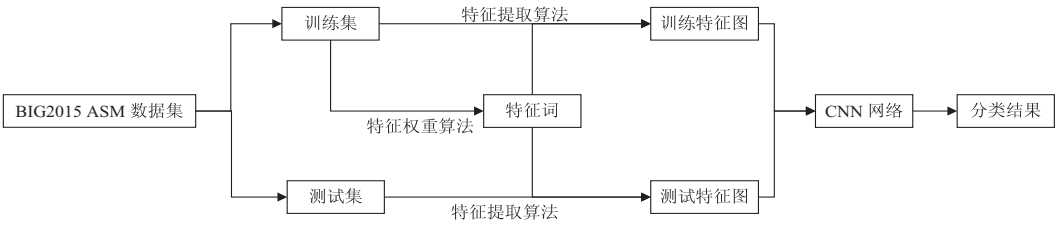

图 1　恶意软件分类系统框架

2　实验过程与结果分析

2.1　数据集

由于该文研究恶意软件的静态检测,因此选用的数据集为微软于 2015 年在 kaggle 平台发布的 big2015 数据集,该数据集已经成为恶意软件研究人员的基准数据集,并被大量引用[14]。该文使用的恶意软件表现形式是 ASM 文件,该文件是通过 IDA 工具反汇编生成的。big2015 中共有 21 741 个样本,但是只有 10 868 是有标签的样本,该数据集有 9 个恶意软件类别,每个类别的名称、数目和标签如表 1 所示。将 10 868 个有标签的数据按类别随机划分为训练集(80%)和测试集(20%)。

表 1　big2015 数据集中的样本类别及数目

Class	numbers	labels
Ramnit	1 541	0
Lollipop	2 478	1
Kelihos_ver3	2 942	2
Vundo	475	3
Simda	42	4
Tracur	751	5
Kelihos_ver1	398	6
Obfuscator. ACY	1 228	7
Gatak	1 013	8

2.2　模型训练

2.2.1　ASM 文件的特征提取

ASM 文件内有 push、mov、cmp 等操作码,还有各种其他英文字段,由于 ASM 文件中还有一些 16 进制字段,比如 BD、E3 等会被误判为英文单词,所以通过正则式方法只提取长度大于等于 3 的单词。

通过 python 代码提取的操作码按照频率从高到底排列,比如 dword、byte、segment 等。

在训练集上对每个类别分别提取 300 个频率高的候选词,然后对每个类别所有候选词进行第一节中的特征权重计算,合并去重,然后进行降序排列,取前 256 个候选词作为特征词库。表 2 显示了权重靠前的几个特征词的权重,虽然 dword 这个特征词出现频率非常高,但是它在所有类别里出现得很平均,因此特征

权重的值反而很小。endp 这个词的出现频率很低,但是它在各个类别中的分布不均匀,因此特征权重很高。

表 2 权重靠前的几个特征词的权重

Word	weight
Endp	7.95
Xchg	7.94
Size	7.93
Start	7.93
Ptr	7.92
hdc	7.91
lparam	7.91
segment	7.90
align	7.90

通过该特征词库对所有 ASM 文件进行特征提取,统计该特征中单词的出现次数,然后将数据进行取整并标准化。例如特征词库第一个词是 endp,第二个词是 xchg 等等,然后对于一个样本,经过统计,endp 出现了 100 次,xchg 出现了 150 次,则即可得到该样本的特征向量:

$$(100,150,\cdots) \tag{6}$$

为了消除同一个特征向量中不同特征间量纲的影响,提高数据之间的可比性,将数据统一到同一个量纲下。该文选用(max,min)标准化方法,先得到一个特征向量中的最大值 $x_{\max}$ 和最小值 $x_{\min}$,标准化的公式如下:

$$x' = 255 * \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{7}$$

将数据转化到 0 ~ 255 的区间,之所以选用这个区间,是因为该区间兼顾了机器学习的输入要求(数字过大会导致运算过于缓慢)、CNN 的输入要求、图像可视化要求(0 ~ 255 对应灰度像素的范围)。

最后将长度为 256 的特征向量转化为 16 * 16 的灰度图,作为单个文件的特征图,传入 CNN 进行训练。

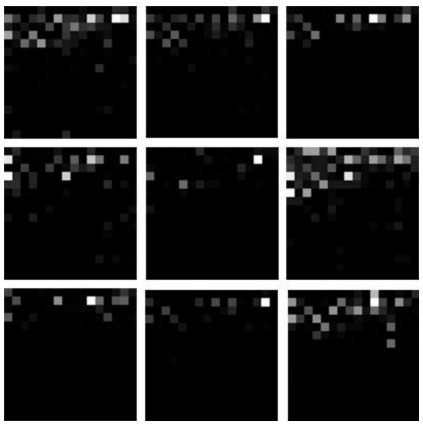

图 2 9 种类别的灰度图示例

如图 2 所示,每种类别的样本各选取一个产生一张灰度图,每张图就代表一个样本的最终特征图。观察这九张图,例如对比第一个像素,标签为 6 的样本明显比其他的要亮;对比第二个像素,标签 3 和 5 的样本明显比其他的要亮等等。特征提取的目的就是在有限的特征向量上尽可能提高不同类别的区分度,因此图 2 可以直观地体现出改进的特征权重算法的作用。

2.2.2 CNN 网络设计

通过对多种优化器进行实验对比,选用收敛速度和准确率较好的 Adam 优化器。训练批次为 128,训练轮次为 100,并使用变化学习率,设置初始学习率为 0.005,每经过 5 轮迭代,将学习率变为原 0.9 倍,以达到逐渐逼近最优值的目的。CNN 的架构参数如表 3 所示,采用的 CNN 总体上分为四层,输入为 16 * 16 的灰度图,其中前三层为卷积-激活-池化的组合重复三次,最后一层为全连接层,输出为 9 个类别。

表 3 CNN 架构参数

层 layers	输入 input	卷积核大小 kernal_size	特征图 feature map	输出 output
2D conv	1 * 16 * 16	3 * 3	16	16 * 16 * 16
Relu	16 * 16 * 16		16	16 * 16 * 16
maxPool2D	16 * 16 * 16	2 * 2	16	16 * 8 * 8
2D conv	16 * 8 * 8	3 * 3	64	64 * 8 * 8
Relu	64 * 8 * 8		64	64 * 8 * 8
maxPool2D	64 * 8 * 8	2 * 2	64	64 * 4 * 4
2Dconv	64 * 4 * 4	3 * 3	128	128 * 4 * 4
Relu	128 * 4 * 4		128	128 * 4 * 4
maxPool2D	128 * 4 * 4	2 * 2	128	128 * 2 * 2
Fully connected	128 * 2 * 2			9

2.2.3 实验环境

(1)处理器:Intel(R) Core(TM) i7-6700 CPU;

(2)内存:8 GB;

(3)硬盘:120 GB SSD+1TB HDD;

(4)集成开发环境:PyCharm Community Edition 2020.3 x64;

(5)Python 版本:3.8;

(6)深度学习框架:Torch 1.7.1。

2.3 评价指标

使用 accuracy(准确率)、precision(精准率)、recall(召回率)和 f1 进行模型性能的评估,其定义如下[8]:

记 S 为数据集中的样本数量,i 表示 S 中的第 i 个样本,y 表示预测值,Y 表示真实值,$l(x)$ 为指示函数,则:

$$\text{accuracy} = \frac{1}{|S|}\sum_{i=0}^{|S|-1} l(Y_i = y_i) \tag{8}$$

其次,定义:

$$P(A,B) = \frac{|A \cap B|}{|A|}, A \neq \varnothing \tag{9}$$

$$R(A,B)=\frac{|A\cap B|}{|B|},B\neq\varnothing \tag{10}$$

令 s 为 S 的子集,则:

$$\text{precision}=\frac{1}{|S|}\sum_{s\in S}P(y_s,Y_s) \tag{11}$$

$$\text{recall}=\frac{1}{|S|}\sum_{s\in S}R(y_s,Y_s) \tag{12}$$

$$\text{f1}=\frac{2\cdot\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}} \tag{13}$$

其中,accuracy 为模型预测正确占总量比例; precision 为预测正确中真正确的比例;recall 所有真正确中被正确预测的比例;f1 是 precision 和 recall 二者的加权数值。

2.4　实验结果

实验结果如表 4 所示,可见四个指标均在 99% 以上。

表 4　实验结果

项目	值/%
准确率(accuracy)	99.26
精确率(precision)	99.10
召回率(recall)	99.31
f1	99.19

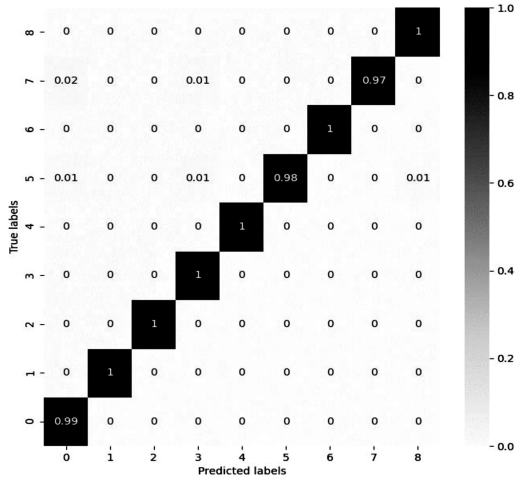

图 3　混淆矩阵表示的实验结果

分类器分类效果的混淆矩阵如图 3 所示,图中竖向 0 到 8 表示样本的真实标签(可以参考表 1),横向 0 到 8 表示该分类器的分类结果,图中任一点表示某一类测试数据的真实标签和测试标签相等的频率。该图像的数据已经进行了归一化,每一行的数据加起来是 1,代表该真实标签的数据总量为 1(由于只取两位有效数字,因此存在 0.01 的误差),右边的对比指示条代表正确率大小,颜色越深则越接近 1,颜色越浅则越接近 0。用混淆矩阵来评判分类器的表现,如果数据更加集中在对角线上,则代表分类器的效果越好,数据越分散则分类器效果越差。真实标签为 1、2、3、4、6、8 时,预测标签也分别是 1、2、3、4、6、8 的频率为 1,代表几乎所有标签为 1、2、3、4、6、8 的数据都被分类正确了。真实标签为 7 时,有 97% 的数据被分类器正确分类了,还有 3% 的数据被错误分类为 0、3 类。

2.5　对比实验

如图 4 所示,使用基于频率特征提取的方法进行分类,该方法先统计各个特征词的频率,并简单选取在全部样本中出现频率高的特征词,然后进行特征提取。对比图 3(文中方法)可以发现,该方法总体上效果还不错,但第四类的查全率比较低,只有 80% 的样本被正确分为第四类,而文中方法在第四类的查全率是 100%。简单的基于特征频率的特征提取方法只是关心所有样本的特征频率,只是取所有样本特征中频率靠前的特征,有些频率不高但是信息增益很大的特征容易被忽视。

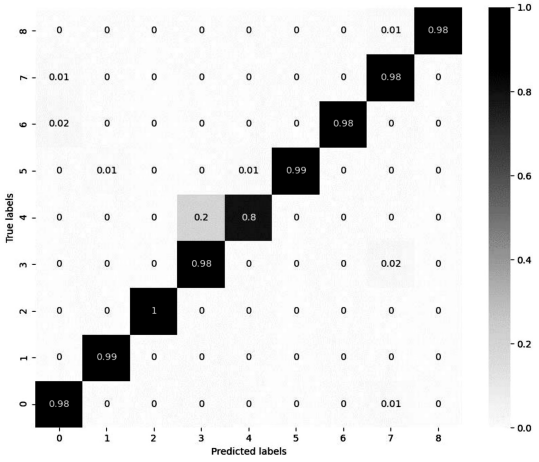

图 4　使用频率特征提取的实验结果

文中方法不再简单地关注特征词在所有样本中出现的概率,而是考虑信息增益的思想,只关注对分类有帮助的特征,因此明显优于基于频率特征提取的方法。

big2015 数据集上的分类效果对比如表 5 所示。

通过实验结果与其他参考文献实验结果的对比,第一行为冠军方法,第六行为使用 TFIDF 方法做的对比实验,最后一行为该文采用的方法。可以清晰地看到,文中方法在准确率上优于操作码频率提取方法、kNN、图像矢量这些自动提取方法,逼近该数据集上的冠军方法(准确率 99.83%),而且在特征的预处理、训练时间和预测时间上要明显优于冠军方法。该文的特征选择方法准确率比 TFIDF 方法略高,体现了文中方法对 TFIDF 提取方法的优越性,在预处理时间上,两者不分上下,因为二者都要经过特征提取和 CNN 学习的过程,只是权重策略算法不同而已。除此之外,还可以发现手动提取特征比自动提取特征的效果要好,预处理的时间要更短一些;诸如在特征提取阶段直接使用 CNN 进行自动提取的方法可以减少人工干预,因而

容错率更高,但效果和处理时间上没有手动提取的好。

表 5 big2015 数据集上的横向对比

Method	Accuracy/%	Pretreatment time/h	Training time/h	Prediction/s
BIG2015 Winner Solution[15]	99.83	72.00	1.0	13 649
Novel Features Extraction[16]	99.77	21.86	–	4 096
操作码频率[17]	98.50	–	–	–
Lempel-Ziv Jaccard Distance with kNN[18]	97.10	1.35	–	–
基于图像矢量的恶意代码分类模型[8]	97.87	0.23	1.70	5.11
RandomForest+ TFIDF	97.93	10	0.5	50
Method of this article	99.26	10	0.5	5

3 结束语

在人们的工作生活和计算机产品的联系日益紧密的今天,恶意软件的识别和分类工作是人们能享受到信息时代便利和数据安全的保障。在应对诸如零日漏洞等问题上,恶意软件分类效率是至关重要的,准确率和检测时间就是效率的体现。所提方案优化了传统特征权重算法,选出了信息增益高的特征词,因而增加了分类的效率;通过结合 CNN 网络模型,完成了恶意软件的高效分类工作。但一些模型参数尚有调整优化空间,例如特征权重公式和 CNN 框架均未达到最优,且未利用字节码数据集,这些问题都是以后的优化方向。

参考文献:

[1] YANNAKOGEORGOS P A, LOWTHER A B. Conflict and cooperation in cyberspace: the challenge to national security [M]. [s. l.]: CRC Press, 2013.

[2] HAN J W, HOE O J, WING J S, et al. A conceptual security approach with awareness strategy and implementation policy to eliminate ransomware[C]//Proceedings of the 2017 international conference on computer science and artificial intelligence. Jakarta: Association for Computing Machinery, 2017: 222-226.

[3] SCHULTZ M G, ESKIN E, ZADOK F, et al. Data mining methods for detection of new malicious executables[C]// Proc of 2001 IEEE symposium on security and privacy. Oakland: IEEE, 2000: 38-49.

[4] ZHANG H, XIAO X, MERCALDO F, et al. Classification of ransomware families with machine learning based on N-gram of opcodes[J]. Future Generation Computer Systems, 2019, 90: 211-221.

[5] 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 (Viz Sec'11). New York: ACM, 2011: 1-7.

[6] 徐玄骥,张智斌. 基于多维度特征的 Android 恶意软件检测方法[J]. 通信技术,2021,54(5):1240-1245.

[7] 杨春雨,徐 洋,张思聪,等. 基于静态特征融合的恶意软件分类方法[J]. 计算机工程与应用,2021,57(15):147-155.

[8] 蒋永康,吴 越,邹福泰. 基于图像矢量的恶意代码分类模型[J]. 通信技术,2018,51(12):2953-2959.

[9] SALTON G, BUCKLEY C. Term-weighting approaches in automatic text retrieval[J]. Information Processing & Management, 1988, 24(5): 513-523.

[10] 吴思慧,陈世平. 结合 TFIDF 的 Self-Attention-Based Bi-LSTM 的垃圾短信识别[J]. 计算机系统应用,2020,29(9):171-177.

[11] 张家旺,李燕伟. 基于机器学习算法的 Android 恶意程序检测系统[J]. 计算机应用研究,2017,34(6):1774-1777.

[12] 苏林萍,刘小倩,陈 飞,等. 基于 N-Gram 和 TFIDF 的 SQL 注入检测方法[J]. 计算机与数字工程,2021,49(6):1177-1181.

[13] 李彦冬,郝宗波,雷 航. 卷积神经网络研究综述[J]. 计算机应用,2016,36(9):2508-2515.

[14] LI L, DING Y, LI B, et al. Malware classification based on double byte feature encoding[J]. Alexandria Engineering Journal, 2022, 61(1): 91-99.

[15] XIAOZHOU W J L, XUEER C. Say no to overfitting[EB/OL]. (2015-05-01)[2018-08-05]. https://github.com/xiaozhouwang/kaggleMicrosoftMalware/blob/master/Saynotooverfitting.pdf.

[16] AHMADI M, ULYANOV D, SEMENOV S, et al. Novel feature extraction, selection and fusion for effective malware family classification[C]//Proceedings of the sixth ACM conference on data and application security and privacy. New Orleans: Association for Computing Machinery, 2016: 183-194.

[17] 任卓君. 基于深度学习的恶意代码可视化检测及分类研究[D]. 上海:东华大学,2020.

[18] RAFF E, NICHOLAS C. An alternative to NCD for large sequences, Lempel-Ziv Jaccard distance[C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. Halifax: Association for Computing Machinery, 2017: 1007-1015.