一种应用于机器学习的恶意网页特征提取方法

张珂伟[1,2],郑世普[1,2],程永灵[1,2],王长帅[1,2]
(1.中电(海南)联合创新研究院有限公司,海南 澄迈 571924;
2.海南省 PK 体系关键技术研究重点实验室,海南 澄迈 571924)

摘 要:基于机器学习的恶意网页检测技术进行研究。目前流行的“特征码”“白名单”等方式,仅能够检测已知的恶意网页;机器学习方法,能够检测出未知的恶意网页,但在处理网页特征时要面临数据量大、复杂和繁琐的问题。提出一种哈希压缩的方法,用于处理网页的特征数据。该方法在保证检测模型的漏报率和误报率下可实现将 150 万的特征映射在 2 万的特征空间内,对提取出的特征数据运用 K 折交叉验证法训练多个传统机器学习模型和集成学习模型。并通过评估模型的检测效果,筛选出表现最好的分类检测模型。

关键词:机器学习;恶意网页检测;哈希压缩
中图分类号:TP181 **文献标识码**:A DOI:10.16157/j.issn.0258-7998.222907

中文引用格式:张珂伟,郑世普,程永灵,等. 一种应用于机器学习的恶意网页特征提取方法[J].电子技术应用,2022,48(12):122-127.
英文引用格式:Zhang Kewei,Zheng Shipu,Cheng Yongling,et al. A feature extraction method for malicious web pages applied on machine learning[J]. Application of Electronic Technique,2022,48(12):122-127.

A feature extraction method for malicious web pages applied on machine learning

Zhang Kewei[1,2],Zheng Shipu[1,2],Cheng Yongling[1,2],Wang Changshuai[1,2]
(1.CEC Joint Innovation Research Institute,Chengmai 571924,China;
2.Key Laboratory of PK System Technologies Research of Hainan Province,Chengmai 571924,China)

Abstract:Applied on machine learning, malicious web page detection technology is studied in this paper. At present, popular methods of “feature code” or “whitelist” can only detect known malicious web pages. The method of machine learning can detect unknown malicious web pages, but it has to face the problem of that the data is large, complex and tedious when processing web page features. In this paper, a Hash compression method is proposed. The method can map 1.5 million features into 20,000 feature space, and train multiple traditional machine learning models and integrated learning models using k-fold cross-validation method for extracted feature data. The best classification detection model will be selected by evaluating the detection effect of the model.
Key words:machine learning;malicious web page detection;Hash compression method

0 引言

PKS 体系是中国电子在 PK 体系的基础上,将“可信计算 3.0”技术融入到 CPU、操作系统和存储控制器中,形成了“三位一体”的“PKS”主动免疫防护。PKS 通过在核心层内生内置安全技术,最大限度地提升网络安全防护效果。本文基于 PKS“小核心大生态”理念,在基于 PKS 核心底座的基础上,通过提出一种网页特征提取方法,实现在增强层进一步提升系统安全的能力。

随着网络的迅速发展,网络攻击已经成为一个严重的问题。当前一些网络钓鱼、垃圾邮件、木马下载、恶意软件执行等攻击方式常常通过恶意网页作为传播中介。因此,检测恶意网页去阻止这些攻击,对维护网络安全具有非常重要的意义[1]。

当前恶意网页的检测方法主要包括静态特征检测和动态特征检测,两种检测方法都需要对网页特征进行提取。静态特征的提取方法是首先需要建立一个恶意网页特征库,对网页的源代码或 URL 链接等属性进行特征提取,将提取的特征在恶意网页静态特征库中进行比对,最终判断待检测网页是否为恶意网页[2-4]。动态特征检测方法是对恶意网页在运行时数据的下载动作、插件处理、访问网页等动态特征进行提取,通过分析行为结果对待检测网页进行检测[5-6]。

以上两种检测方法都存在一些缺点。静态特征检测方法中当恶意网页利用加密等方式伪装恶意内容后会导致检测准确率下降。动态特征检测方法在分析时往往需要“蜜罐”、虚拟内存等技术进行协助,进而增加了系统的运行负担,同时在分析过程中也存在被攻击者攻击的风险[7]。

近年来，随着人工智能的发展，为了解决以上两种传统检测方式的缺陷和局限性，机器学习被越来越多地应用于恶意网页检测。利用机器学习检测恶意网页主要包括训练和预测两个步骤[8-10]。训练是通过训练集训练出一个分类模型，预测是利用训练好的分类模型进行检测，检测出待测网页是否属于恶意。其中特征提取和模型选择很大程度上决定了机器学习最终检测效果的好坏。当前机器学习的恶意网页检测主要存在两个问题：第一，提取的特征具有一定的时效性，因此需要保持特征的更新与升级；第二，提取特征的复杂度越来越高，花费时间越来越长，因此需要降低特征提取的复杂度和缩短花费的时间[11]。

本文通过选择结构简单的特征和采用哈希算法降低特征维度去解决机器学习在恶意网页检测中存在的以上两个问题。

1 实验设计

1.1 实验流程图

本实验由数据获取、数据清洗、特征提取、特征降维、模型训练和模型评估等环节构成，流程图如图 1 所示。

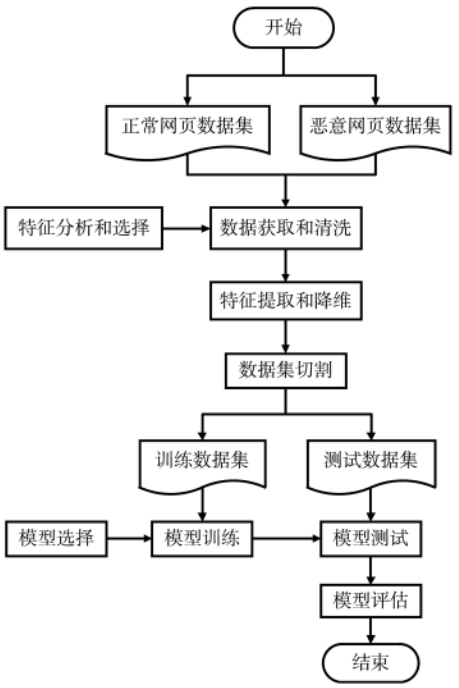

图 1 实验流程图

1.2 数据获取和清洗

1.2.1 数据获取

本实验中采用监督学习方式训练恶意网页分类模型，因此需要大量的正常 HTML 网页数据集和恶意 HTML 网页数据集。为了保证实验具有一定意义上的现实通用性，本文实验中正常网页数据的采集使用 PKS 体系的新八核 D2000 终端机设备，恶意网页数据来源于国外知名恶意代码数据库网站 VirtuTotal.com。

1.2.2 数据清洗

采集的网页数据集中包含一定数量的重复和缺失网页，经过去重和删除等处理后，最终得到需要的数据集。数据集中网页特征标签如表 1 所示。

表 1 网页特征标签("0"表示正常，"1"表示恶意)

网页特征	标签
正常网页特征	0
恶意网页特征	1

1.3 特征提取方法和哈希压缩方法

1.3.1 特征提取方法分析

网页的可读字符串即网页中包含的连续可打印的源代码字符串，可读字符串中包含丰富的信息，比如网址、加密特征、恶意函数、消息文本等。网页可读字符串是网页运行和展示的核心部分，通过对正常网页可读字符串和恶意可读字符串进行特征分析，发现恶意网页的可读字符串通常具有某些共性。基于这些共性可以对网页可读字符串进行提取，并作为分类模型的输入特征[12]。可读字符串举例如表 2 所示。

表 2 可读字符串举例

正常网页	恶意网页
document.MM_pgW=innerWidth	pp=window.open("bg.php",´_blank´,"height=20,
style="position:absolute	CwEk2akseFqqiu2FWhS3Om5otkRDLsdZ
name="adere"	var output = Aes.Ctr.decrypt(hea2t, hea2p,

恶意网页可读字符串的关键特征归纳如下：

(1)Object、Image、Embed、Script 等标签数量较多。

(2)嵌套的 URL 地址数量较多。

(3)Window.open、ocument.location、window.location、document.write 等方法调用次数较多。

(4)混淆代码多且长，并使用 escapte、unescapte、encrypt、decrypt 等函数进行加解密。

(5)JS 脚本代码长度过长，并且熵值较高。

1.3.2 特征提取方法实现

可读字符串特征提取的方法如下：

(1)利用正则表达式截取空格和波浪线编码之间的可读字符串，每个可读字符串作为一个维度特征。

(2)获取长度大于 20 的可读字符串，长度较短的可读字符串容易重复并且特征不明显，后期容易增加训练训练的负担，经过多次试验后，取字符串的长度为 20，不仅能够显著降低特征维度的数量而且能够达到满意的效果。

特征提取方法和执行结果如图 2 所示。

1.3.3 哈希压缩方法分析

将可读字符串作为特征虽然简单，但是提取后特征维度数量达到了 150 万左右。特征维度数量过多会增加

```
chars=r' -~' //空格到波浪线之间的unicode编码
min_length = 20 //匹配字符串特征的最小长度
string_regexp = '[%s]{%d,}' % (chars, min_length) //正则表达式
string_regexp=string_regexp.encode(encoding="utf-8") //
pattern = re.compile(string_regexp) //特征提取
```

```
b'<table border="0" width="100%" cellspacing="0"
cellpadding="1">'
b' <tr class="footer">'
b' <td class="footer"> '
```

图 2 特征提取方法和执行结果

模型的复杂度和训练时间。因此需要在尽量不损失原始特征表达能力的条件下，把原始的高维特征压缩成低维特征，最终本文采用特征哈希法进行了特征降维处理。

1.3.4 哈希压缩方法实现

哈希压缩方法通过哈希将高维特征映射到低维特征空间中，本实验中使用 sklearn 的 FeatureHasher 类来实现哈希压缩方法。

```
hasher=FeatureHasher(20000)
hashed_features=hasher.transform([string_features])
```

利用随机森林算法构建分类模型，经过不断调整输出的特征维度数量，观察模型的预测准确率，最终选择了较为合适的 20 000 作为输出维度特征数量，如图 3 所示。

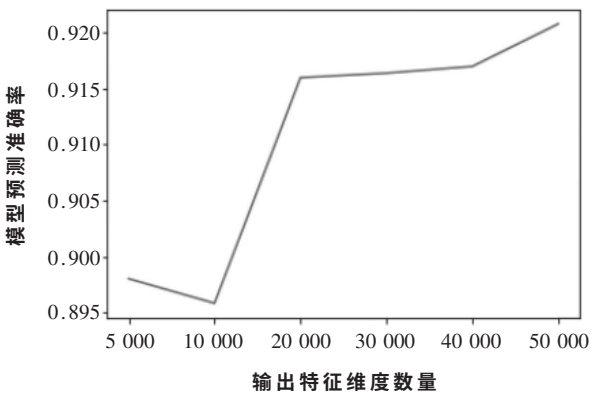

图 3 哈希法的输出维度数量对随机森林模型预测准确率的影响

1.3.5 哈希压缩方法降维处理

将可读字符串特征向量存储在字典中，并赋值为 1，采用哈希特征算法进行降维处理，得到输出维度是 20 000 的哈希特征向量。如图 4 所示。

```
[ 0.  0. -1. ...  0.  0.  0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[0. 0. 0. ... 0. 0. 0.]
[ 0.  0.  0. ...  0.  0. -1.]
[0. 0. 0. ... 0. 0. 0.]
```

图 4 部分降维后的特征维度

1.4 机器学习模型选择

实验中采用基于 Python 的 scikit-learn 库所提供的神经网络模型进行恶意网页分类检测。首先选取 6 种经典的机器学习算法，包括决策树、逻辑回归、朴素贝叶斯、支持向量机、K 近邻、神经网络。除了经典的机器学习算法以外，还选取了机器学习中的 3 种集成学习算法，包括随机森林、梯度提升树(GBDT)和 AdaBoost。最终对经典机器学习算法和集成学习算法进行了对比研究[13-15]。

2 实验结果

2.1 实验设备

本实验的开发环境硬件采用 PKS 体系的新八核 D2000 终端机设备，软件采用 Python 3.7-64 位的 spyder 3.3.6 版本。实验代码主要分为字符串特征提取、特征哈希降维、模型训练、模型评估等。

2.2 评估方法

恶意网页检测是一个二值分类问题，评估指标主要分为功能指标和性能指标。功能指标主要用于对各类模型的分类效果进行评价，而性能指标主要用于对各类模型的效率进行评价。

功能指标主要包括以下 5 项：

(1)准确率(ACC，accuracy rate)：指模型中预测正确的结果在总样本中所占的比例，主要用于衡量检测模型总体的准确程度。

(2)精确率(P，precision)：指被正确分类的正样本占所有被分类为正样本的比例，主要用于衡量模型对于恶意样本的差准率。

(3)召回率(R，recall)：指在所有正样本中被正确识别为正样本的样本比例，主要用于衡量检测模型对于恶意样本的检测能力。

(4)误报率(FPR)=1-精确率：正常网页被识别为恶意网页的概率。

(5)漏报率(FNR)=1-召回率：恶意网页被识别为正常网页的概率。

性能指标主要包括运行时间，指模型在处理数据集时所耗费的时间。

本文实验中使用准确率、误报率、漏报率和运行时间去评价各种分类模型的检测效果，理想的检测模型是在运行时间较短的条件下，具有较高的准确率及较低的误报率和漏报率。

2.3 实验过程

考虑到机器设备性能以及训练数据集的规模，实验中采用 K 折交叉验证进行检测模型的评估，其中 K 值设为 3。部分参数采用 scikit-learn 库所提供的默认值，准确率、漏报率、误报率和运行时间用小数表示，并精确到 4 位。

2.3.1 第一次实验

实验目的是挑选功能和性能指标优秀的模型。本次

实验采用样本为 10 000 的数据集。其中正常网页包含 5 000 个，恶意网页包含 5 000 个网页，各占 50%。结果如表 3 和图 5 所示。

由图 5 得出，基于本文提取的特征向量条件下，传统机器学习中，逻辑回归、神经网络、决策树三个模型表现较好，其中逻辑回归模型的运行时间最短同时准确率最高，漏报率和误报率也较低。集成学习中，随机森林和 AdaBoost 模型表现较好，其中随机森林模型的运行时间最短，同时准确率最高，漏报率和误报率也较低。

2.3.2 第二次试验

实验目的是选择最优秀的模型。本次实验与上次实验最大的不同是提高了样本数据集数量，数据集共包含 90 000 个网页。其中正常网页包含 45 000 个，恶意网页包含 45 000 个网页，各占 50%。经过第一次实验的对比筛选，本次实验只选取在第一次实验中表现最好的两个模型逻辑归回和随机森林进行比较。结果如表 4 和图6、图 7 所示。

2.4 实验评估

表 4 第二次实验的恶意网页识别结果

分类模型	准确率	漏报率	误报率	消耗时间/s
逻辑回归	0.956 3	0.048 6	0.039 2	182.997 6
随机森林	0.938 0	0.029 5	0.088 6	26 871

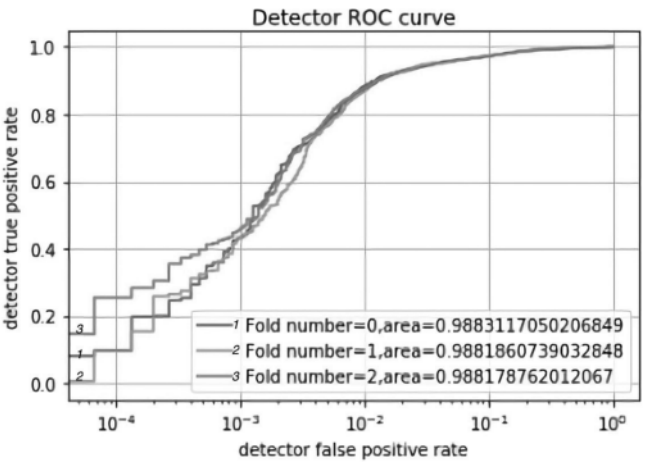

图 6 三次逻辑回归的 ROC 曲线

表 3 第一次实验中恶意网页识别结果

分类模型		参数设置	准确率	漏报率	误报率	运行时间/s
传统机器学习模型	决策树	默认	0.854 9	0.102 6	0.172 8	446.037 2
	逻辑回归	默认	0.913 6	0.098 9	0.075 9	10.255 7
	朴素贝叶斯	默认	0.741 5	0.241 6	0.266 2	29.846 6
	支持向量机	启用概率估计	0.678 5	0.635 2	0.001 6	8 976.765 4
	K 近邻	邻近数量 15	0.820 6	0.174 9	0.178 8	13 356.598 1
	神经网络	默认	0.880 1	0.089 6	0.141 4	244.210 6
集成学习	AdaBoost	SAMME 分类算法，学习率 0.8	0.874 4	0.216 9	0.038 8	2 634.402 8
	随机森林	子树数量等于 100	0.915 4	0.102	0.07	265.217 6
	GBDT	随机种子等于 10	0.864 4	0.233 1	0.043 6	7 865.769 3

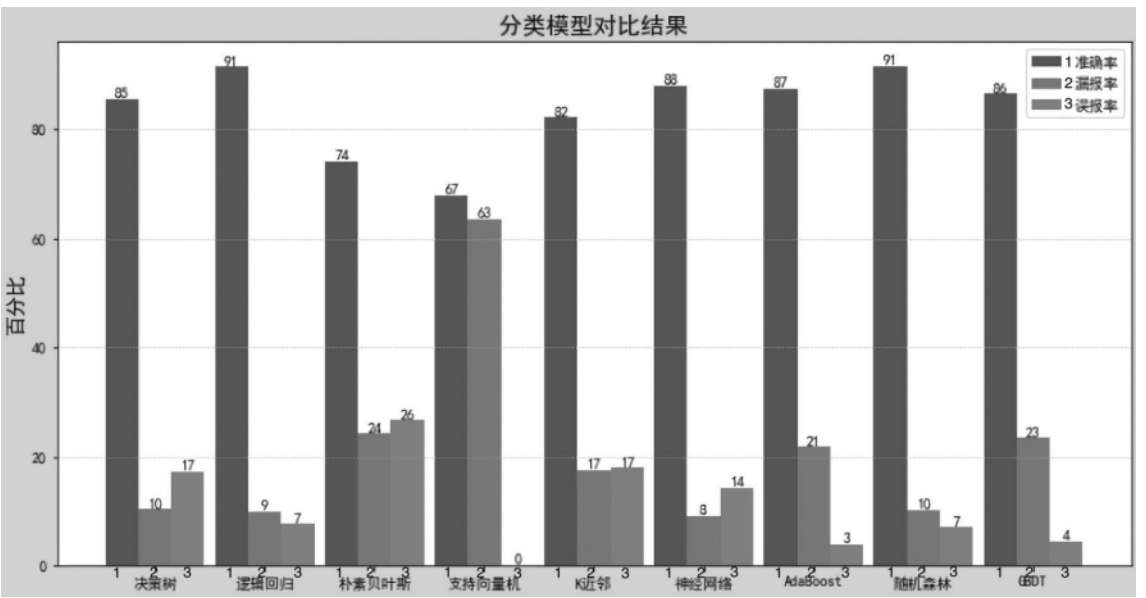

图 5 多种检测模型对比结果

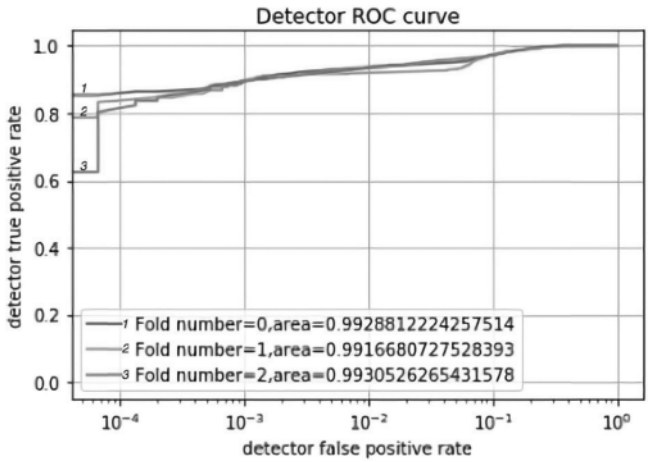

图 7 三次随机森林的 ROC 曲线

2.4.1 第一次实验结果评估

基于本文提取的特征，第一次实验中，传统机器学习模型在各项指标中从优到劣的排序如表 5 所示，综合各项指标能够得出逻辑回归模型对恶意网页的检测效果最好。集成学习模型在各项指标中从优到劣的排序如表 6 所示，综合各项指标可以得出随机森林模型对恶意网页的检测效果最好。

集成学习的思想是先训练若干个弱分类器，再通过某种串行或并行方式将这些弱分类器组合起来，从而达到提高预测准确率的效果。分别统计传统机器学习和集成学习中所有模型的各项指标平均值，如表 7 所示。通过结果对比能够证明，集成学习模型整体表现优于传统机器学习模型。

表 5 传统机器学习模型按性能指标排序

准确率(降)	漏报率(升)	误报率(升)	运行时间(升)
逻辑回归	神经网络	支持向量机	逻辑回归
神经网络	逻辑回归	逻辑回归	朴素贝叶斯
决策树	决策树	神经网络	神经网络
K 近邻	K 近邻	决策树	决策树
朴素贝叶斯	朴素贝叶斯	K 近邻	支持向量机
支持向量机	支持向量机	朴素贝叶斯	K 近邻

表 6 集成学习模型按性能指标排序

准确率(降)	漏报率(升)	误报率(升)	运行时间(升)
随机森林	随机森林	AdaBoost	随机森林
AdaBoost	AdaBoost	GBDT	AdaBoost
GBDT	GBDT	随机森林	GBDT

2.4.2 第二次实验结果评估

第二次实验中发现，随着样本数据集规模的增加，逻辑回归和随机森林两个检测模型在准确率、漏报率方面的性能都有所提升，但在误报率方面随机森林模型的性能反而下降。运行时间上，随机森林模型也消耗了更多的时间。逻辑回归和随机森林两次实验的结果如表 8 所示，逻辑回归和随机森林两次实验的比较如表 9 所示。

表 7 传统机器学习和集成学习的各项指标平均值

	准确率(平均)	漏报率(平均)	误报率(平均)	运行时间/s
传统机器学习	0.814 8	0.447 5	0.288 3	11 297
集成学习	0.884 7	0.184 0	0.050 8	3 587

表 8 逻辑回归和随机森林两次实验的结果

		准确率	漏报率	误报率	运行时间/s
第一次实验	逻辑回归	0.913 6	0.098 9	0.075 9	10.255 7
	随机森林	0.915 4	0.102	0.07	265.217 6
第二次实验	逻辑回归	0.956 3	0.048 6	0.039 2	182.997 6
	随机森林	0.938 0	0.029 5	0.088 6	26 871

表 9 逻辑回归和随机森林两次实验的比较

	准确率(提高)	漏报率(下降)	误报率	运行时间(倍)
逻辑回归	4.7%	50.9%	48.4%(降低)	16.7(增加)
随机森林	2.5%	72.5%	26.6%(提升)	103.9(增加)
逻辑回归	4.7%	50.9%	48.4%(降低)	16.7(增加)
随机森林	2.5%	72.5%	26.6%(提升)	103.9(增加)

2.5 实验结论

2.5.1 传统机器学习模型比较

逻辑回归和神经网络表现尚可，其中逻辑回归在准确率、误报率和运行时间上表现更好。而决策树、K 近邻、朴素贝叶斯和支持向量机则表现不佳。

2.5.2 集成学习模型比较

集成学习模型中所选的三个模型总体表现尚可，其中随机森林在准确率、漏报率和运行时间上表现更好。而 AdaBoost 和 GDBT 则相对较差。

2.5.3 逻辑回归和随机森林模型比较

参与训练的样本数据集增加 7 倍后，逻辑回归模型表现略优秀，其中准确率提高了 4.7%，漏报率下降 50.9%，误报率下降了 48.4%，运行时间增加了 16.7 倍。随机森林不仅运行时间增加了 103.9 倍，误报率反而提升了 26.6%。

2.5.4 逻辑回归和随机森林 ROC 曲线比较

由 ROC 曲线得处结论，即在解决恶意网页检测的二分类问题上，在不考虑性能指标即运行时间的条件下，逻辑回归和随机森林的整体表现区别不大。

3 结论

本文根据真实的样本数据集构建了恶意网页检测模型，基于本文所提取的特征，对多种不同的模型在恶意网页检测问题上进行了研究。

实验结果表明虽然集成学习整体优于传统的机器学习模型，但逻辑回归模型综合表现最优，在性能指标最优的情况下，功能指标中的准确率、漏报率和误报率

分别达到了95.63%、4.86%和3.92%。遗憾的是在现实工作环境中，如果检测模型的误报率高于1%，那么这个模型是不能被应用的，因此本文提出的检测模型暂时无法应用于实际工作。

在本文的第一次实验中能够观察到神经网络模型的整体表现也非常优秀，所以在第二次实验中原本增加了神经网络模型的训练，但由于机器性能问题，导致48小时后依然没有得到结果，因此神经网络模型没有出现在第二次实验的比较结果中。

本文所使用的样本数据集规模依然偏小(9万条数据集)，因此在训练过程中容易出现过拟合问题。但可以预期，随着训练样本数据的增加和机器设备性能的提高，后期能够进一步提高模型的检测指标，下一步工作不仅重新开展神经网络模型的训练，而且也将增加在恶意网页检测中深度学习模型的训练，最终将深度学习检测模型移植到PKS体系中。期望通过本文的研究内容对于增强PKS体系未来大生态应用场景的安全产生一定的积极作用。

参考文献

[1] 李泽宇，施勇，薛质.基于机器学习的恶意URL识别[J].通信技术，2020，53(2)：427-431.

[2] KOLTER J，MALOOF M.Learning to detect malicious executables in the wild[C]//Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining，470-478.

[3] TIAN R，BATTEN L，VERSTEEG S.Function length as a tool for malware classification[C]//Proceedings of the 3rd International Conference on Malicious and Unwanted Software，2008：57-64.

[4] SIDDIQUI M，WANG M C，LEE J.Detecting Internet worms using data mining techniques[J].Journal of Systemics，Cybernetics and Informatics，2009：48-53.

[5] ZOLKIPLI M F，JANTAN A.An approach for malware behavior identification and classification[C]//Proceeding of 3rd International Conference on Computer Researchand Development，2011：191-194.

[6] FIRDAUSI I，LIM C，ERWIN A.Analysis of machine learning techniques used in behavior based malware detection[C]//Proceedings of 2nd International Conference on Advances in Computing，Control and TelecommunicationTechnologies(ACT)，2010：201-203.

[7] 叶自谦.一种结合Kafka和Spark-streaming的大规模快速恶意网页识别方法的设计与实现[D].南京：南京邮电大学，2019.

[8] HOU Y T，CHANG Y，CHEN T，et al.Malicious Web content detection by machine learning[J].Expert Systems with Applications，2010，37(1)：55-60.

[9] LIN S F，HOU YT，CHEN C M，et al.Malicious webpage detection by semantics-aware reasoning[C]//The Eighth International Conference on Intelligent Systems Design and Applications，2008：115-120.

[10] HOU Y T，CHANG Y，CHEN T，et al.Malicious Web content detection by machine learning[J].Expert Systems with Applications，2010，37(1)：55-60.

[11] 王正琦，冯晓兵，张驰.基于两层分类器的恶意网页快速检测系统研究[J].网络与信息安全学报，2017，3(8)：48-64.

[12] 陈维.恶意软件识别方法研究与应用[D].成都：电子科技大学，2017.

[13] 陈远，王超群，胡忠义，等.基于主成分分析和随机森林的恶意网站评估与识别[J].数据分析与知识发现，2018，2(4)：71-80.

[14] 孙博文，黄炎裔，温俏琨，等.基于静态多特征融合的恶意软件分类方法[J].网络与信息安全学报，2017，3(11)：68-76.

[15] 王松.基于学习的恶意网页智能检测系统[D].南京：南京理工大学，2011.

(收稿日期：2022-04-22)

作者简介：

张珂伟(1979-)，男，硕士，工程师，主要研究方向：网络安全、机器学习、PKS体系。

郑世普(1980-)，男，硕士，助理研究员，主要研究方向：信号处理、科技管理。

程永灵(1979-)，男，本科，工程师，主要研究方向：PKS产品及行业解决方案。

扫码下载电子文档

版权声明